IBM iDoctor for IBM i
7.4 Documentation

(Also covers latest changes to 7.2 and 7.3)

IBM iDoctor for IBM i Development Team

28 January 2022

Abstract

Provides in-depth coverage of all major GUI functions for all components at 7.1, 7.2, 7.3 and 7.4. This document also covers the server-side portion of the iDoctor tools such as the various commands used for collecting and analyzing performance data.

Changes

28 Jan 2022 – Updated latest slides/content to match build 1523

18 Feb 2021 – Replaced YouTube links with MediaCenter links

16 Dec 2019 – Initial creation (based on 7.3 version)

Table of Contents

1 Introduction

1.1 Product Overview

IBM iDoctor for IBM i (iDoctor) is a suite of performance tools used by IBM and customers to collect and analyze performance data in order to quickly solve performance problems on IBM i. The tools may be used to monitor overall system health at a high level or for analyzing performance details within job(s), disk unit(s) and/or programs collected. iDoctor includes many drill-down options to assist you with the most logical next step listed first.

At 7.4 IBM iDoctor for IBM i includes the following components:

Job Watcher
Collection Services Investigator
Disk Watcher
Plan Cache Analyzer
PEX Analyzer
FTP GUI
Power Connections
QMGTOOLS (green screen only)

Note: Collection Services Investigator, Plan Cache Analyzer and Disk Watcher are included with a Job Watcher license. The iDoctor license for Job Watcher is a different offering than the Job Watcher feature included with the Performance Tools LPP (licensed program product, PT1). PT1 is not required to run iDoctor.

All components require IBM i 7.1 or higher with the required PTFs for each installed. The required PTFs are listed on the iDoctor website for each release.

Server build updates are no longer available at 7.1, but the client is still tested at that release and can be used.

1.2 Updates Summary

This PDF identifies the major changes with each release and can be found here:
https://public.dhe.ibm.com/services/us/igsc/idoctor/iDoctorUpdatesSummary.pdf

1.3 FAQs

The iDoctor Frequently Asked Questions document is available here:
https://public.dhe.ibm.com/services/us/igsc/idoctor/iDoctorFAQs.pdf

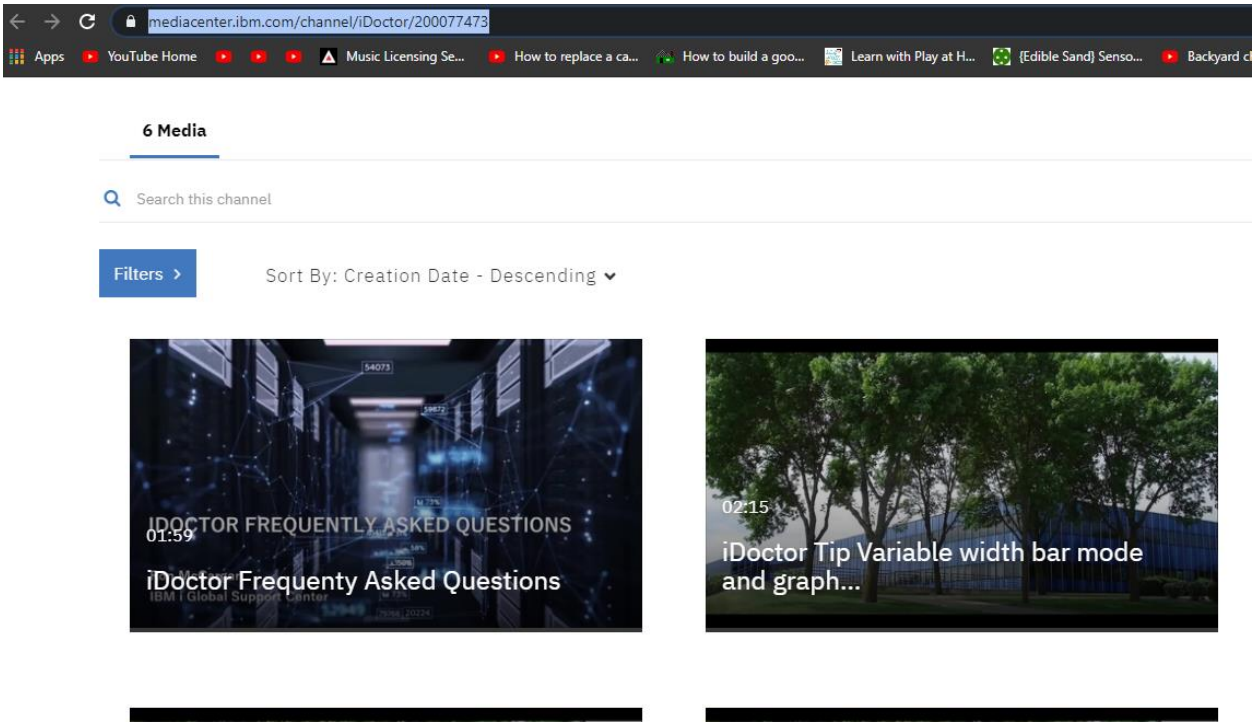
1.4 Videos

A few helpful videos on iDoctor usage can be found on the IBM Systems MediaCenter channel.

IBM Systems MediaCenter:
https://mediacenter.ibm.com/category/IBM+i/172218552

iDoctor videos:
https://mediacenter.ibm.com/channel/iDoctor/200077473

This is a place where you can find usage tip videos on the iDoctor tools suite as well as other topics related to IBM I support.



1.5 iDoctor Base Support/QIDRGUI Library

iDoctor has server and client components. QIDRGUI library contains functions/programs/commands needed for the GUI to function properly. Library QIDRGUI must be installed to use any of the iDoctor components with the GUI. In some cases, the library is also necessary when running iDoctor commands in other libraries (like QIDRPA/STRPACOL) because it contains several common objects.

This library also contains all the iDoctor stored procedures which are needed to perform many tasks needed by the GUI. When the GUI starts up it will often check the stored procedures in this library and see if they need to be updated and will attempt to update them via the GUI using create procedure SQL statements. This allows the user in many cases to use the latest client without updating the iDoctor server builds. This process does require that the user has authority to replace these *PGMs in QIDRGUI using SQL. The alternative to this is to install the latest server builds which also will apply the latest stored procedures.

1.6 IBM iDoctor for IBM i Job Watcher

Job Watcher returns real-time information about all jobs, threads and/or LIC tasks running on a system (or on a selected set of jobs/threads or tasks). The data is collected by a server job, stored in database files, and displayed on a client via the iDoctor GUI. Job Watcher is similar in sampling function to the system commands WRKACTJOB and WRKSYSACT in that each "refresh" computes delta information for the ending snapshot interval. Refreshes can be set to occur automatically, as frequently as every 100 milliseconds. The data harvested from the jobs/threads/tasks being watched is done so in a non-intrusive manner (like WRKSYSACT).

This data is summarized to show high-level overviews of system performance over time. From these overview charts a user can select a time period of interest and drill down. The drill down graphs from the

overview charts into rankings graphs to show the job/thread experiencing the highest amount of work for the desired statistic. From the rankings graphs, users can select one or more job/threads to show how they performed over time.

The biggest advantage to Job Watcher for performance analysis over other tools is its extensive use of wait buckets. These buckets consist of waits that are generally considered good or bad, and seeing the bad ones on a graph like seize contention makes it easy to identify problem areas for further investigation.

The information harvested by Job Watcher includes:

- Standard WRKSYSACT type info: CPU, DASD I/O breakdown, DASD space consumption, etc.
- Some data previously only seen in Collection Services: "real" user name, seize time, breakdown of what types of waits (all waits) that occurred.
- Some data not available anywhere else in real time: details on the current wait (duration, wait object, conflicting job info, specific LIC block point id), 1000 level deep invocation stack including LIC stack frames.
- SQL statements, host variables, communications data, activation group statistics
- Classic JVM statistics (6.1 and earlier releases only)
- J9 JVM statistics

Job Watcher is available for trial evaluation or purchase via this website. A license for Job Watcher includes:

- Job Watcher software (licensed by system serial number via an access code)
- Collection Services Investigator software
- Disk Watcher software
- Electronic defect support for the software for the term of the contract
- No charge updates to the software for the term of the contract

The IBM Redbook for Job Watcher provides many examples for the use of Job Watcher. This Redbook is available through the following link: http://www.redbooks.ibm.com/abstracts/sg246474.html

Note: This Redbook was written in the V5R3 timeframe (March 2005).

1.6.1 iDoctor Job Watcher vs PT1 (PDI) Job Watcher

At 6.1 and higher the PT1 LPP offers a Job Watcher GUI in the System Director Navigator web interface called Performance Data Investigator.

For the most part besides the obvious presentation differences all the functionality provided in the web interface is included with iDoctor. For simplicity, here is a list of key functions provided in iDoctor Job Watcher **not** included with the web version:

- Time range graphs (ability to adjust the Time interval size used for graphing)
- Monitors (24 x 7 collection of data)
- Collection scheduling
- PTF checking
- Collection Summary analysis and improved graphing functions as a result
- Create Job Summary analysis to add up totals for the desired jobs across collection(s)
- Call Stack Summary analysis
- Long Transactions analysis
- Situational Analysis
- Dynamic legend (drag/drop, add/remove fields)
- Much faster tables/graphs and better flexibility.
- alternate views (quick toggles to other graph types)
- Collection search
- Call stack reports

- Report Generator – loads graphs and captures screenshots in batch
- Send data to IBM support
- Feature rich SQL editor
- Synchronized tables beneath the graphs
- SQL tables comparison wizard

Additional differences are described here (Note: As of 2015):
http://public.dhe.ibm.com/services/us/igsc/idoctor/JWComparison.jpg

1.7 Collection Services Investigator

Collection Services Investigator provides the user with the ability to analyze the performance database files produced by Collection Services. Collection Services is like Job Watcher in the statistics collected, but the primary difference is the interval size in Collection Services is usually much longer (5-15 minutes vs 5-15 seconds in Job Watcher). Collection Services Investigator can be used to analyze wait statistics, CPU, and I/O activity. Some types of communications reports are also provided.

Collection Services Investigator also includes a function that analyses multiple collections at once for the desired jobs for the purpose of comparing total I/Os, CPU, waits, etc for the collections being analyzed. This is useful when comparing the performance impact of batch runs from one day to the next.

CSI includes support to analyze external storage DS6K/DS8K device cache statistics as well as link and rank statistics for these devices.

1.8 Disk Watcher

Disk Watcher provides the user with the ability to collect either a statistical summary of disk performance data or a trace of all disk I/O events that occur on a system. The trace mode is recommended as it provides more options for analyzing the data and determining potential disk problems.

The Disk Watcher GUI provides many graphs with drill downs for each mode of collection (statistical or trace). Using Disk Watcher the user can take a trace and summarize the trace data into an interval size desired for the purpose of easily graphing the statistics at either or broad or detailed level.

1.9 Plan Cache Analyzer

Plan Cache Analyzer provides the ability to collect and analyze snapshots of the system's SQL Plan Cache. It is designed to complement the features already available in IBM i Navigator for analyzing the Plan Cache by providing several graphs and drill-down options not available there.

The plan cache is a repository that contains the access plans for queries that were optimized by SQE.

1.10 PEX Analyzer

The PEX Analyzer component is specifically geared towards pinpointing issues affecting system and application performance. This is the most advanced of the iDoctor components offering the most detailed reporting for disk operations, CPU utilization, file opens, MI programs, wait states, DASD space consumption and much more.

1.11 iDoctor FTP GUI

The FTP GUI is a general-purpose tool for viewing libraries and objects on the. Files can be transferred to/from a system using this tool.

When connected to an IBM i system it allows the user to work with either the IFS or the libraries on the system. If working with libraries and objects, many additional options are available to show object properties or view data within tables or construct SQL queries over the data they contain.

This is a free component offered “as-is”.

1.12 Power Connections

Power Connections allows users to connect to VIOS or HMC systems from the iDoctor GUI interface. The connections in iDoctor have been segregated into two sections (IBM i and non IBM i.) The Power Connections View in the GUI represents the non-IBM i connections that have been defined within iDoctor.

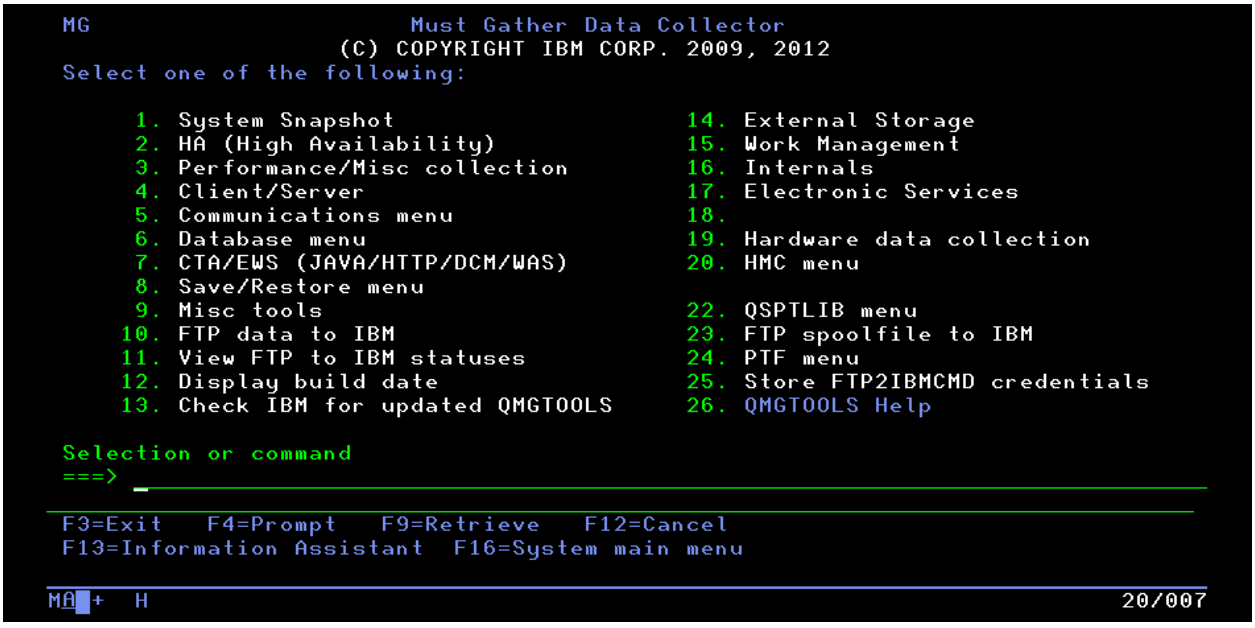
Power Connections can be used to analyze nmon statistics or browse the file systems on VIOS or HMC. When analyzing nmon the data can be optionally transferred to an IBM I for analysis using SQL DB2 databases and graphs.

The Power Connections support also includes scripts and functions to automatically analyze and transfer data to a central repository (currently IBM i only) for analysis.

Power Connections is a free tool offered “as-is”.

1.13 QMGTOOLS (Must Gather Tools)

Must Gather Tools (QMGTOOLS library) is a set of tools to assist individual IBM support teams collect data for issues concerning their products. For example, the HA menu contains a set of programs that will collect specific items (job logs, SST macro output, VLOGs, and so on) to assist the technical representative in debugging a problem related to High-Availability. A user will install this tool and use the menu shown below to collect data for the specific problem:



Must Gather Tools main menu

2 Installation

This chapter includes information about the following:

- IBM i installation requirements
- PC installation requirements
- Ports needed for GUI use
- Installing and uninstalling IBM iDoctor for IBM i

Note: For PTF requirements visit the iDoctor website
http://public.dhe.ibm.com/services/us/igsc/idoctor/html/ptfs.html

2.1 IBM i Requirements

- IBM i 7.1 or higher
- The required PTFs listed at http://public.dhe.ibm.com/services/us/igsc/idoctor/html/ptfs.html for the applicable download page must be loaded and applied.
- The user profile performing the installation must have *SECOFR user class and special authorities *ALLOBJ and *SECADM.
- The following host servers (identified by the SERVER parameter values on the STRHOSTSVR command) need to be running on the server: *DATABASE, *RMTCMD, *SIGNON, *SRVMAP
- System value QALWOBJRST must be *ALL or (*ALWSYSSTT and *ALWPGMADP)
- **Note:** If English is not installed as the primary language, the user profiles used to connect to the server with should set their CCSID parameter value to 37 for best results.
- Power Connections requires OpenSSH option 1 (5733SC1) for the disk mapping function.
- Please ensure that if user profile QIDOCTOR is used, its CCSID is not set to 65535.

2.2 PC Requirements

- Windows 7 or higher: (**Note**: Many users run iDoctor via Citrix/Linux KVM, Mac Parallels or Remote Desktop.)
- Must install with administrator level authority. Administrative level authority is not required to use iDoctor however once it is installed.
- IBM i Access for Windows or IBM i Access Client Solutions (IBM i ACS) **with the Windows application package add-on**. This is 2nd required install step when using IBM i ACS. Download from here: http://ibm.biz/IBMi_ACS
- 8+ GB of RAM
- Pentium 4 or higher
- Microsoft Visual Studio 2012 Update 4 (or higher) redistributable package (the x86 version) must be installed. Links are provided on our download pages.
- .NET framework 4.5 or higher (required for Power Connections only)
- Java 1.5 or higher (required for Power Connections only)
- SQLite (required for Power Connections only. Only needed if you do not have an IBM i available and you wish to analyze non-IBM I performance data.)

Note: Some virus scanners or even Windows itself may prevent the opening of our downloads. In some cases, you may get errors and need to temporarily disable these in order to open the download image.

Warning: The following IBM i Access for Windows service pack levels do not work properly with iDoctor and if installed, iDoctor usage will be disabled:
1) V5R3 or earlier versions
2) V5R4 GA or SI20465
3) 7.1 GA or SI36916 or SI37895. You must use a newer SP level or use ACS with the Windows add-on installed instead.

For more information on iDoctor GUI install requirements visit this iDoctor FAQ page.

2.2.1 Ports needed for GUI access

This section lists the various ports needed for GUI connections on the server. This information was taken from the ibm.com support document TCP/IP Ports Required for IBM i Access and Related Functions.

The iDoctor GUI utilizes the following server functions listed below: **Sign-on** 8476 (9476)**, License Management** 8470 (9470)**, Remote Command** 8475 (9475)**, Database** 8471 (9471)**, Data Queue** 8472 (9472) and **FTP** 21, (990.)

Technote (troubleshooting

Problem(Abstract)

This document provides information on which TCP/IP ports are required to have access when using IBM i Access Client Solutions or IBM i Access for Windows.

Resolving the problem

The following table lists the ports that IBM i Access and related functions use for communication with the IBM i OS System:

PC Function	Server Name	Port Non-SSL	Port SSL
Server Mapper	as-svrmap	449	---
License Management	as-central	8470	9470
Database Access	as-database	8471	9471
Data Queues	as-dtaq	8472	9472
IFS Access using System i Navigator	as-file	8473	9473
Network Printers	as-netprt	8474	9474
Remote Command	as-rmtcmd	8475	9475
Signon Verification	as-signon	8476	9476
Telnet (5250 Emulation)	telnet	23	992
Navigator for i (web)	as-nav	2004	2005
HTTP Administration	as-admin	2001	2010
POP3 (MAPI)	pop3	5010	---
Management Central	as-mgtc >	5555 and 5544	5566 and 5577
Ultimedia Services	as-usf	8480	9480

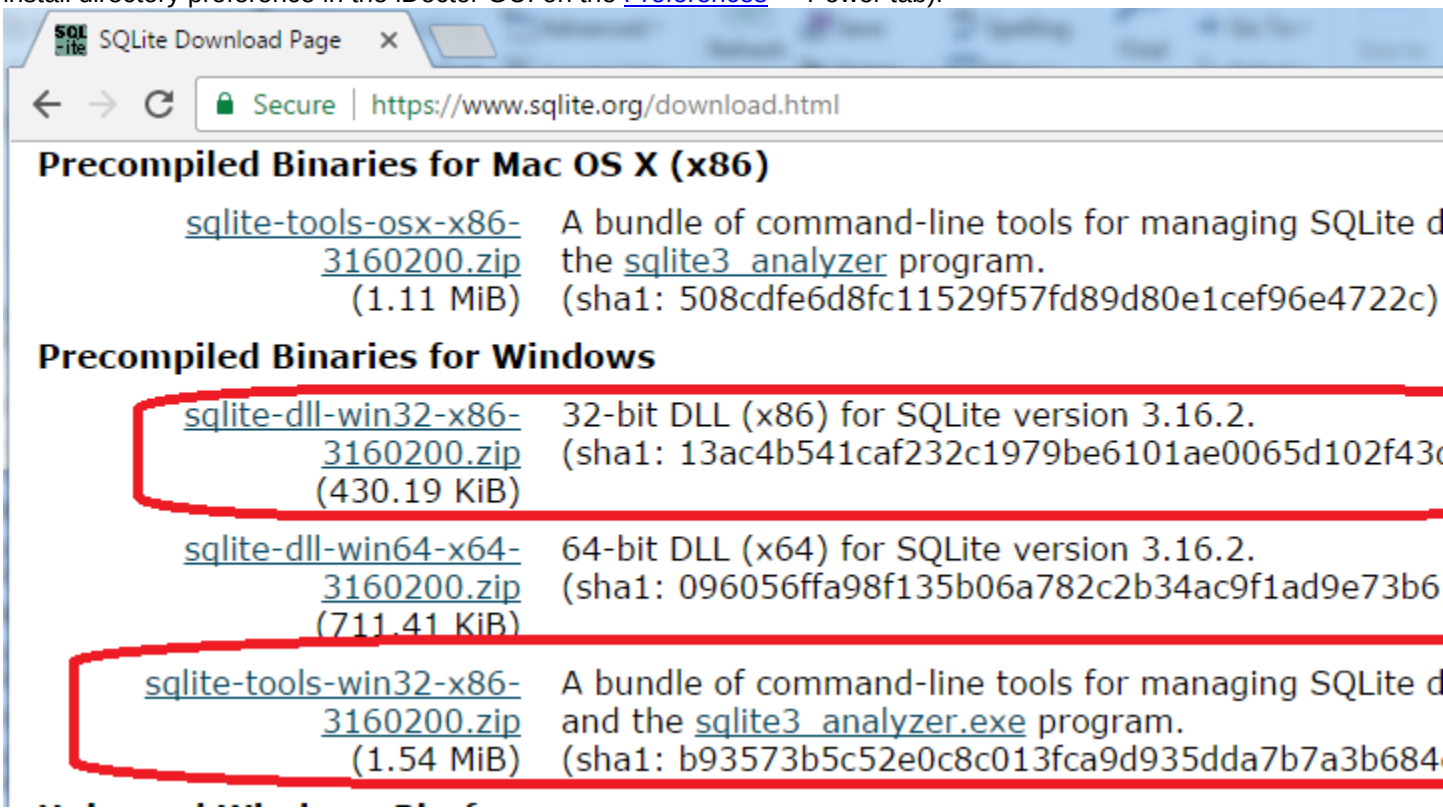
DRDA	DRDA	446	---
DDM	DDM	447	448
NetServer	netbios >	137	---
NetServer	CIFS	445	---
NetServer	netbios >	139	---
Service Tools Server	as-sts	3000	---
DHCP Monitor	---	---	942
RUNRMTCMD	REXEC	512	---

If any of the above ports are restricted using a firewall or any other mechanism, IBM i Access or related functions may fail to operate. For assistance with configuring ports or working with a firewall beyond the above information, contact the firewall provider or obtain a consulting agreement.

2.3 SQLite

If you do not have an IBM i available and you wish to install SQLite in order to analyze performance data on the PC then visit the SQLite website at https://www.sqlite.org/download.html

On their website, you will need to download the sqlite 32-bit dll and the bundle of command-line tools that includes the command-line shell. Unzip both packages to the C:\sqlite directory (or change the sqlite install directory preference in the iDoctor GUI on the Preferences -> Power tab).



SQLite website download page

2.4 Installation

Note: You can install the tools before installing the PTFs if desired. However, it is recommended to install the PTFs before collecting performance data. Use the Fix Central website if you need help installing PTFs.

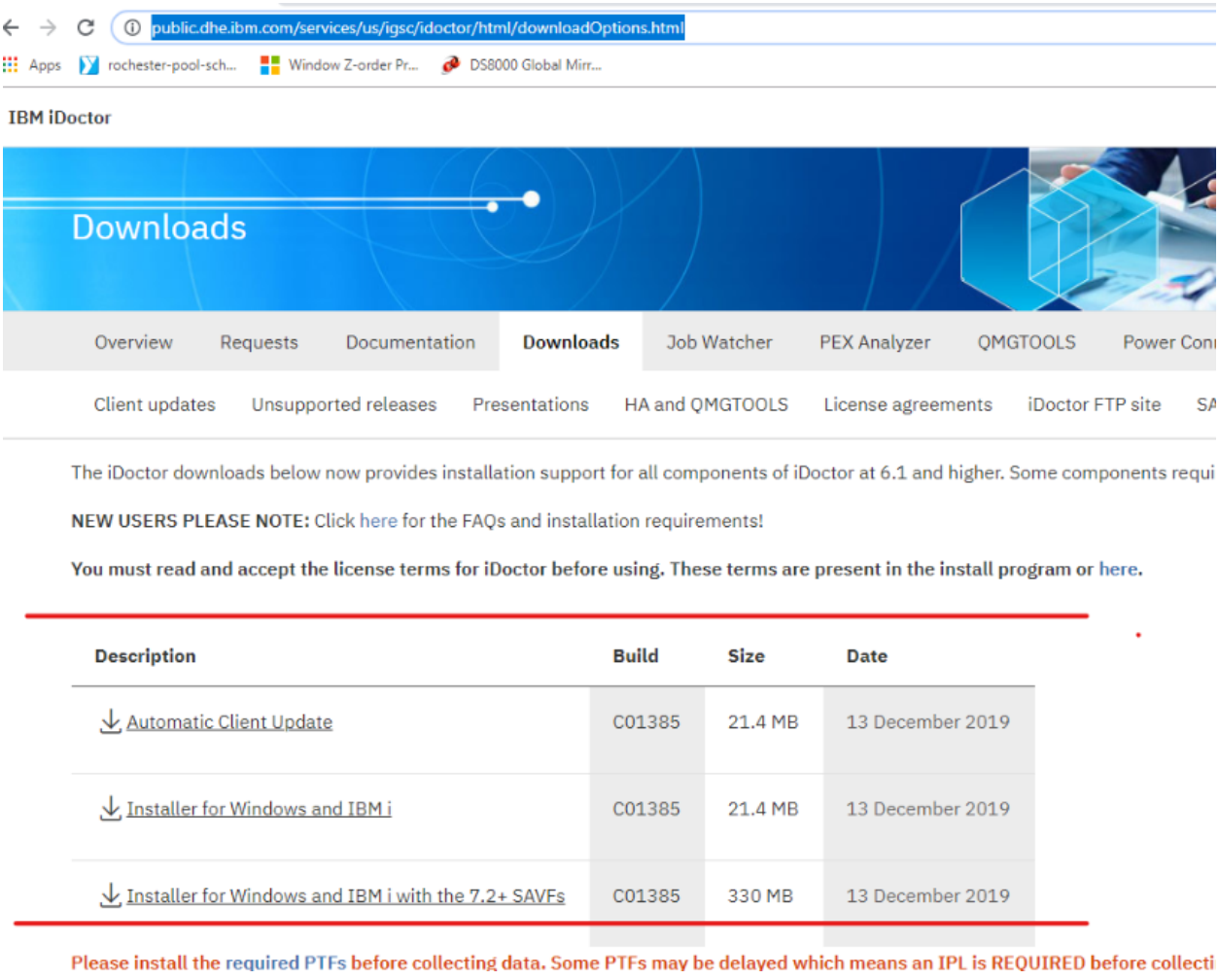
After installation, you will have the following new libraries on your server depending on the components installed:

QIDRGUI - GUI (Base) support
QIDRWCH - Job Watcher
QIDRPA - PEX Analyzer
QMGTOOLS – Must Gather Tools (**Note:** this is a green screen only component!)
QPLANCACHE - Plan Cache Analyzer

Also, you will find that directory /QIBM/ProdData/iDoctor will exist in the IFS after installation.

2.4.1 Install options

On the https://public.dhe.ibm.com/services/us/igsc/idoctor/html/downloadOptions.html page you will find several download options for installing iDoctor:



Install options available:

1. **"Automatic client update"** – This option can be used if you only want to install the GUI and do not need to install the server-side piece of iDoctor. **Note:** Keep in mind that installing client updates may not always work if your server builds are very old. Also the iDoctor GUI will attempt to load newer versions of programs on your server at startup into library QIDRGUI. Do not install client updates if you have restrictions in place on production systems that do not allow program objects to be replaced. **Tip:** This method is also used after installing the GUI and starting it up. iDoctor will connect to our website and will ask you if you wish to update your client if a new build is available. You can enable/disable this check under Preferences -> Confirm

2. **"Installer for Windows and IBM I"** – This option is used if you have sufficient network access to download save files (SAVFs) from our FTP site; ftp://public.dhe.ibm.com/services/us/igsc/idoctor/web The installation will download the required SAVFs automatically during installation and then send them to your IBM i for you**.**

3. **"Installer for Windows and IBM i with the 7.2+ SAVFs"** – This option can be used if you are unable to download the SAVFs with the install program or need to do the installation manually.

4. You can also install server builds using the iDoctor GUI. First connect to an IBM i and on the iDoctor components window press the **Check for new server builds** button. This option does require you to have full admin-level authority to your PC however. **Note:** This option may not work if you have not installed iDoctor in a long time since our processed may have drastically changed.

   The next section will present an example of installing iDoctor using option 2 above.

2.4.2 Installer for Windows and IBM i installation example

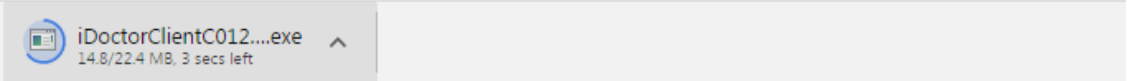
This section will show an example of using the "Installer for Windows and IBM i" download option. This is the recommend install method for new users.

Also see the following video to walk through installing iDoctor using this technique.
https://mediacenter.ibm.com/media/Installing+iDoctor+on+an+IBM+i/1_t18tia7i/200077473

Step 1 Click the link below from our website to download the latest install image:
http://public.dhe.ibm.com/services/us/igsc/idoctor/html/downloadOptions.html

Description	Build	Size	Date
Automatic Client Update	C01242	22.4 MB	12 September 2016
Installer for Windows and IBM i	C01242	22.4 MB	12 September 2016
Installer for Windows and IBM i with the 7.1+ SAVFs	C01242	269 MB	12 September 2016

Step 2 Depending on the web browser installed there will be different prompts/warnings before being able to open the installer. You could also save the file to your PC and open it there instead.

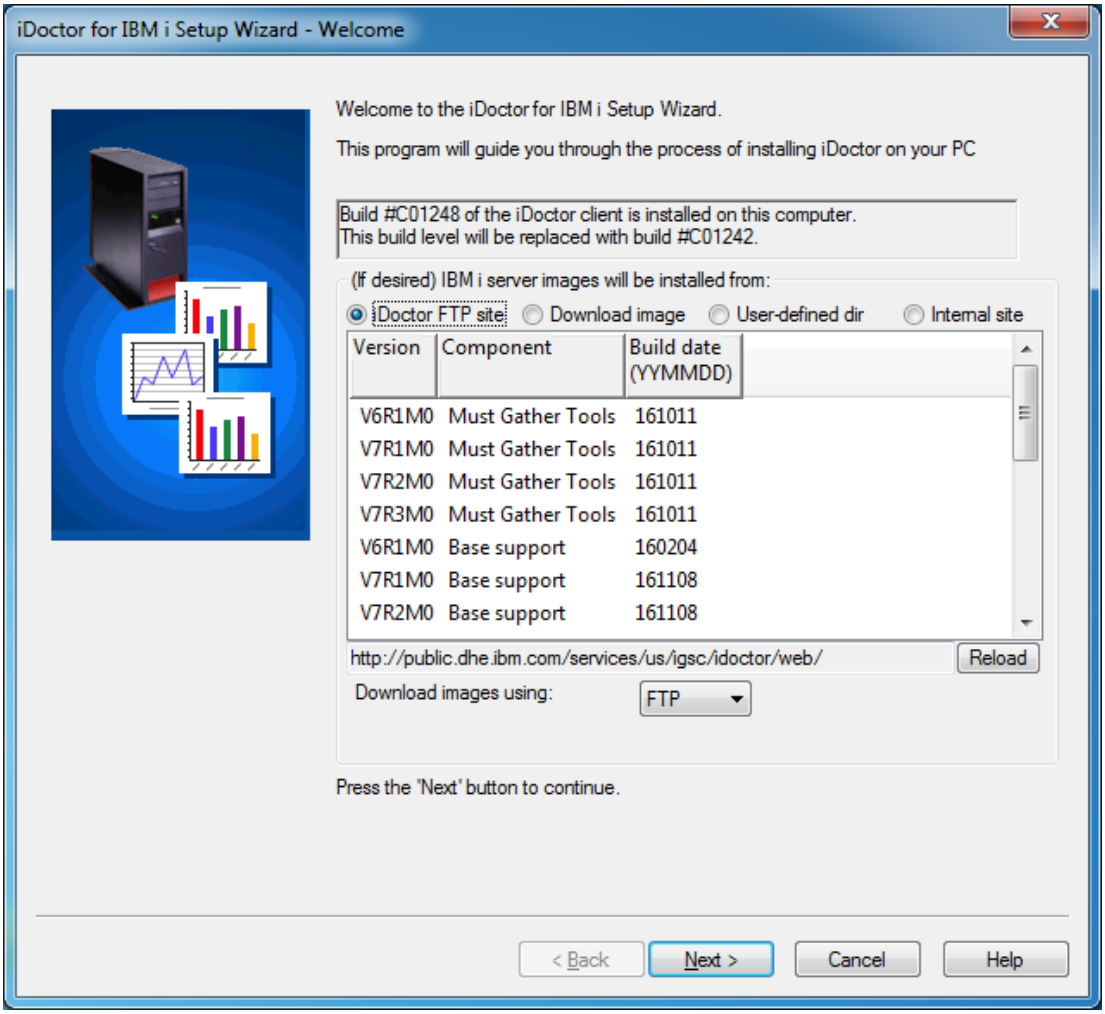


Note: Some virus scanners or even Windows itself may prevent the opening of our downloads. In some cases you may get errors and have to temporarily disable these in order to open the image.
Step 3: This screen identifies the version of iDoctor installed on the PC (if found) and indicates where the server image save files will be installed from. The following options are shown:

- iDoctor FTP site: Most users should use this to download the SAVFs from our website.
- Download image: Use this option if you picked the install image where the SAVFs are bundled.
- User-defined dir: Advanced users can use this option if you need to install using SAVFs in a directory on your PC (or network drive.)
- Internal site: For IBMers only.

Note: The **"Download images using"** option can be FTP or HTTP. Switch this to HTTP if you have trouble downloading the SAVFs to your PC from our FTP site.

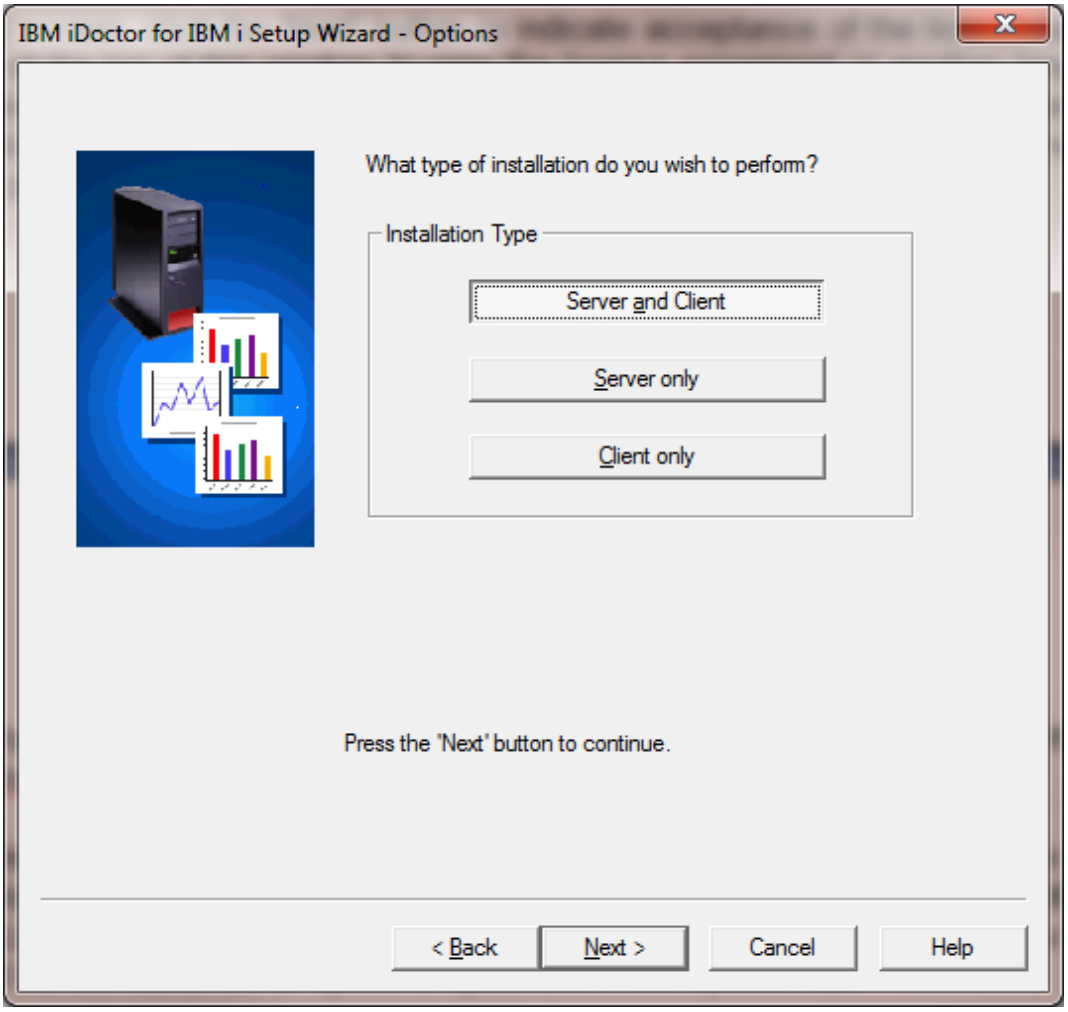
Click the iDoctor FTP site option and click 'Next'.



Step 4 On the next screen, click the 'Accept All' button then the 'Next' button to indicate acceptance of the license agreements. Use the View button next to each component name to view the desired license agreement.

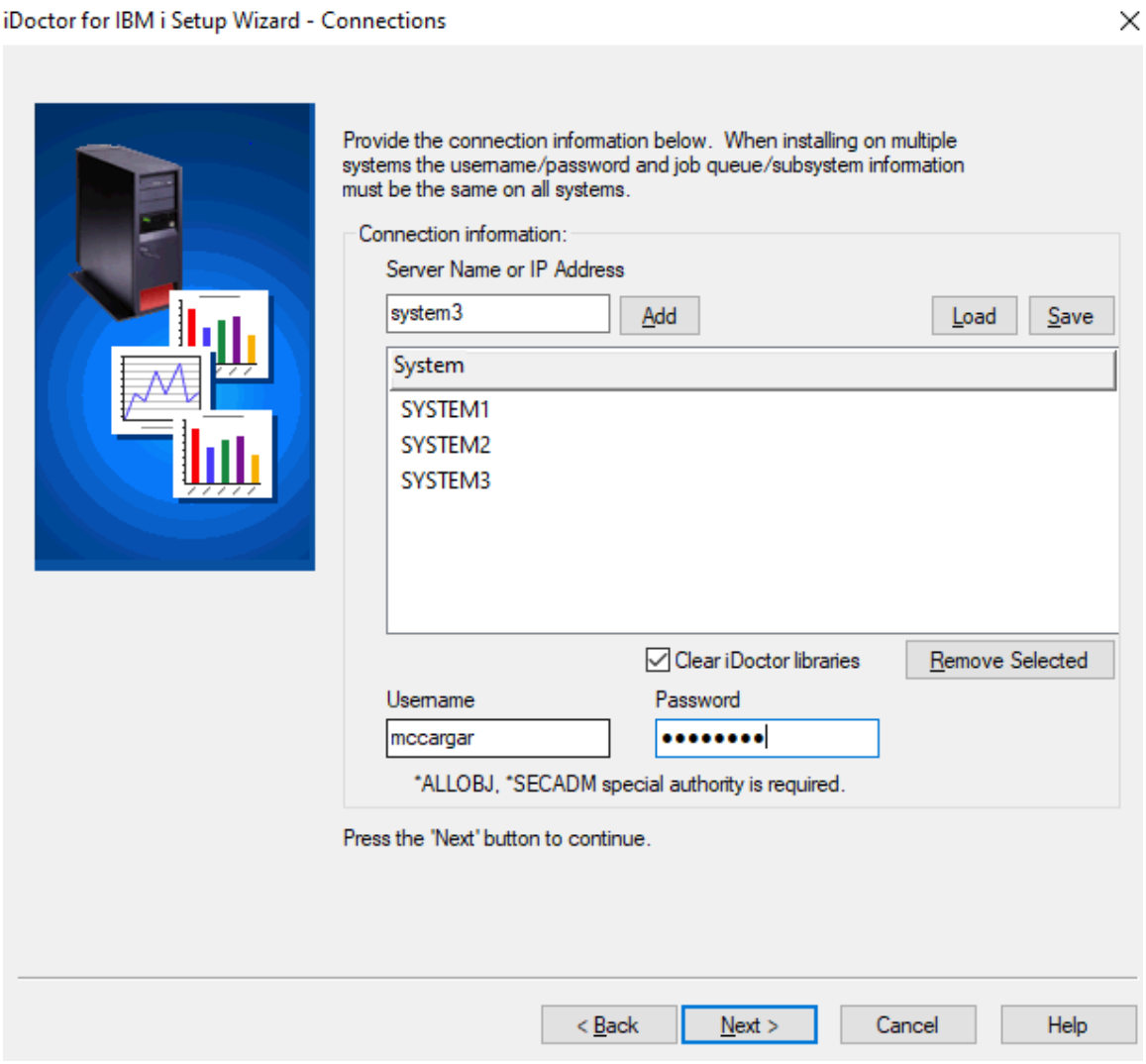
These agreements apply to both the iDoctor client and server code installed. Even if you are only using a trial version, the agreements apply to all users.

Step 5 Select the type of installation to perform. This screen allows you to choose whether to install the server side of iDoctor, the client side of iDoctor or both.

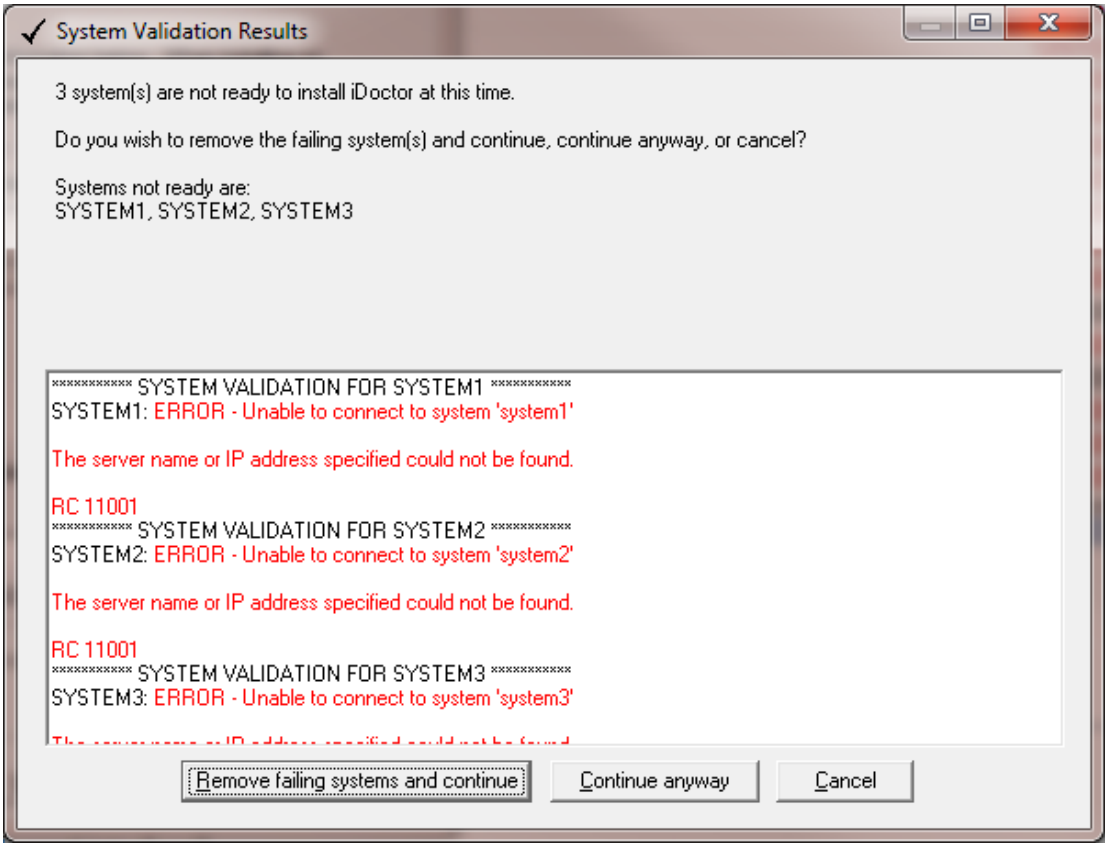


Step 6 If you are installing the server side of iDoctor you will see a screen asking for the connection information to use to access the server(s). The user profile must have the user class authority of *SECOFR and *ALLOBJ, *SECADM special authorities. If installing on multiple systems the user profile and password must be the same on all systems specified in the system list.

If desired, you may use the Save and Load buttons to save a system list to a text file or load one into this interface that was previously saved. This file is a simple list of system names or IP addresses with each entry on a separate line.



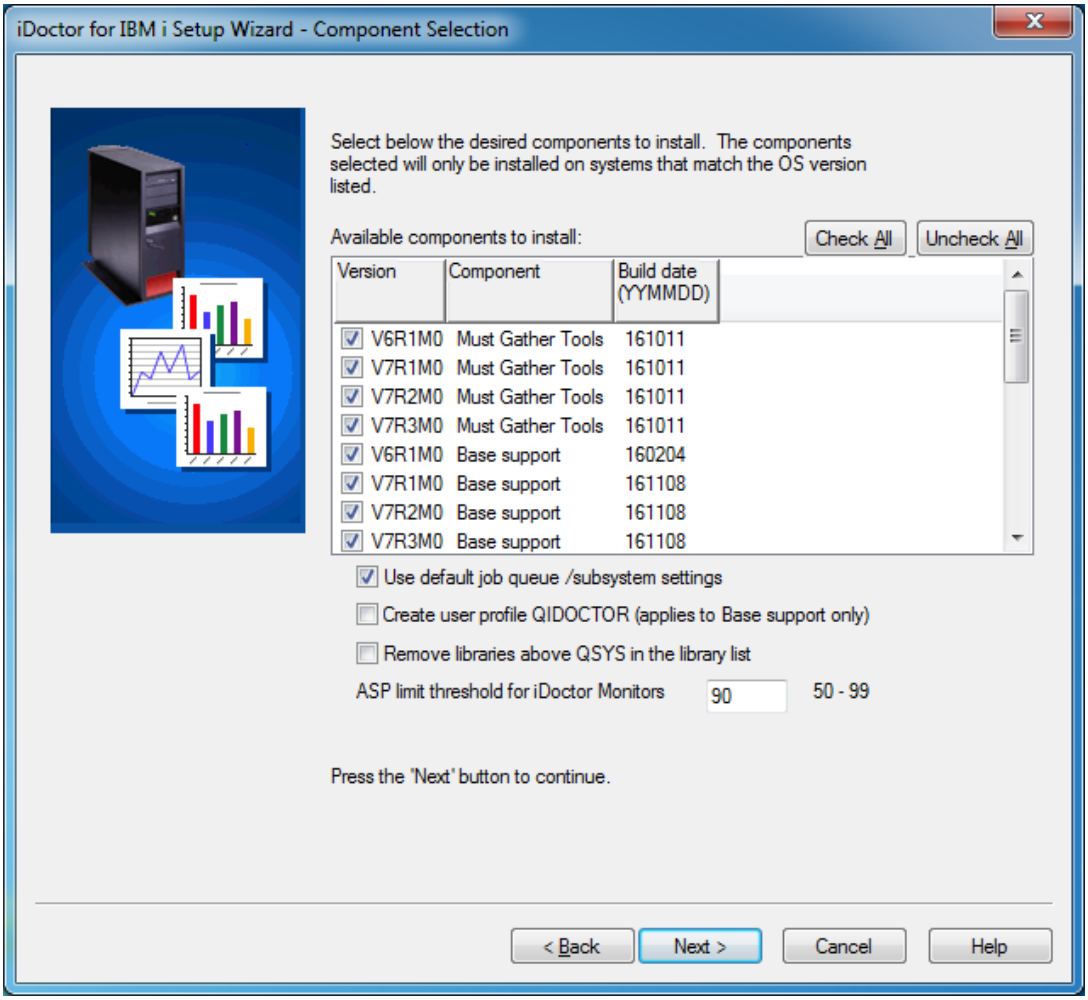
When clicking ‘Next’, system validation will occur, which means that each system will be checked to make sure they are ready for iDoctor to be installed. If any errors occur a screen such as the following will be shown that allows you to view the problems and take corrective action.



Step 7 The Component Selection screen lists the components to be installed. Note that using Disk Watcher or Collection Services Investigator requires that both the Job Watcher and Basic support options are installed.

Note: Base support is required in order to use any of the component.

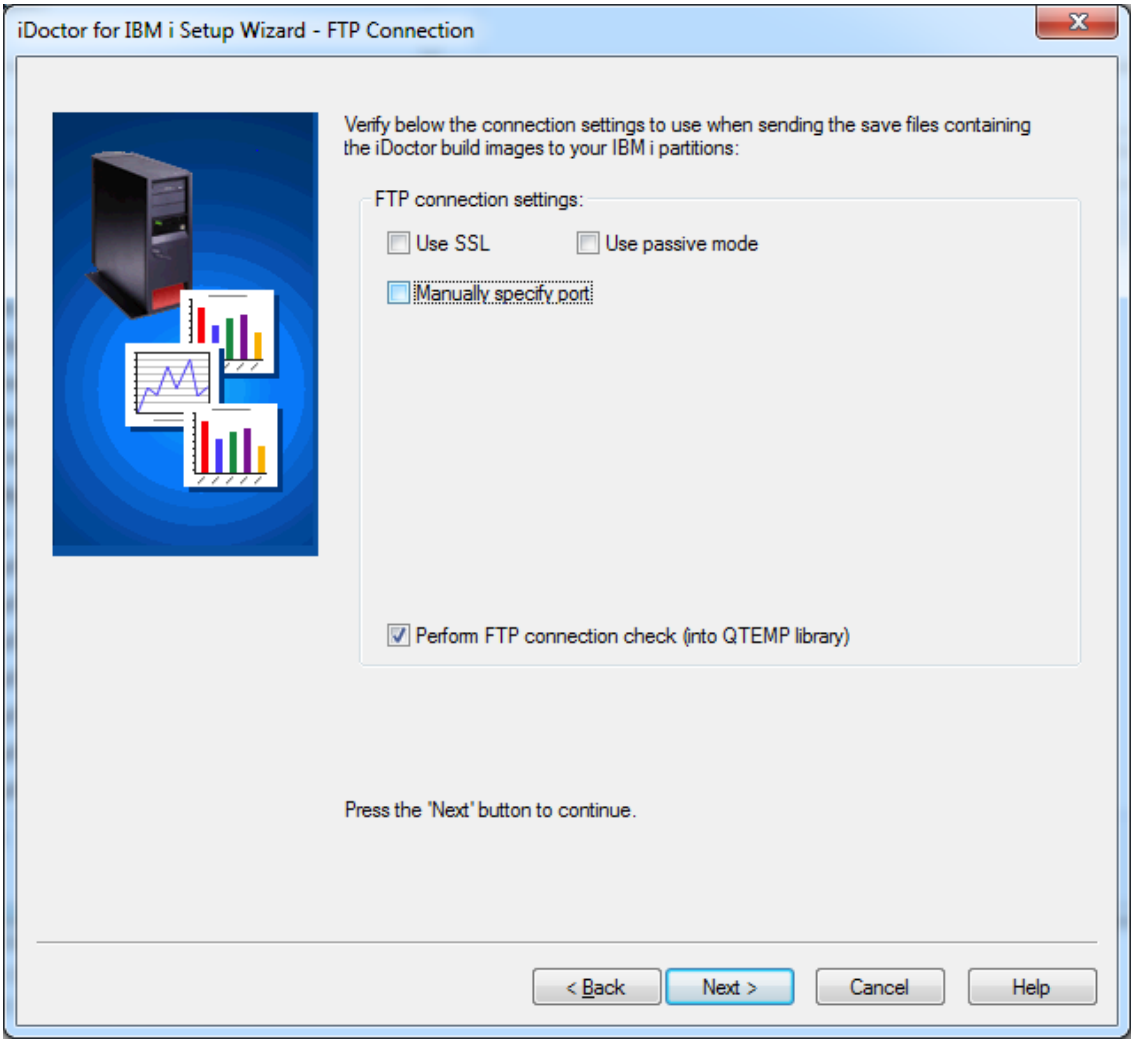
Click 'Next' again on the Component Selection screen, using the default options.



Step 8 The next page gives you the option to specify which type of FTP connection should be used when performing the install. Only in unusual circumstances should anything other than the defaults be used on this page. However, if installing over a VPN connection and “Use passive mode” does not work, try unchecking that option.

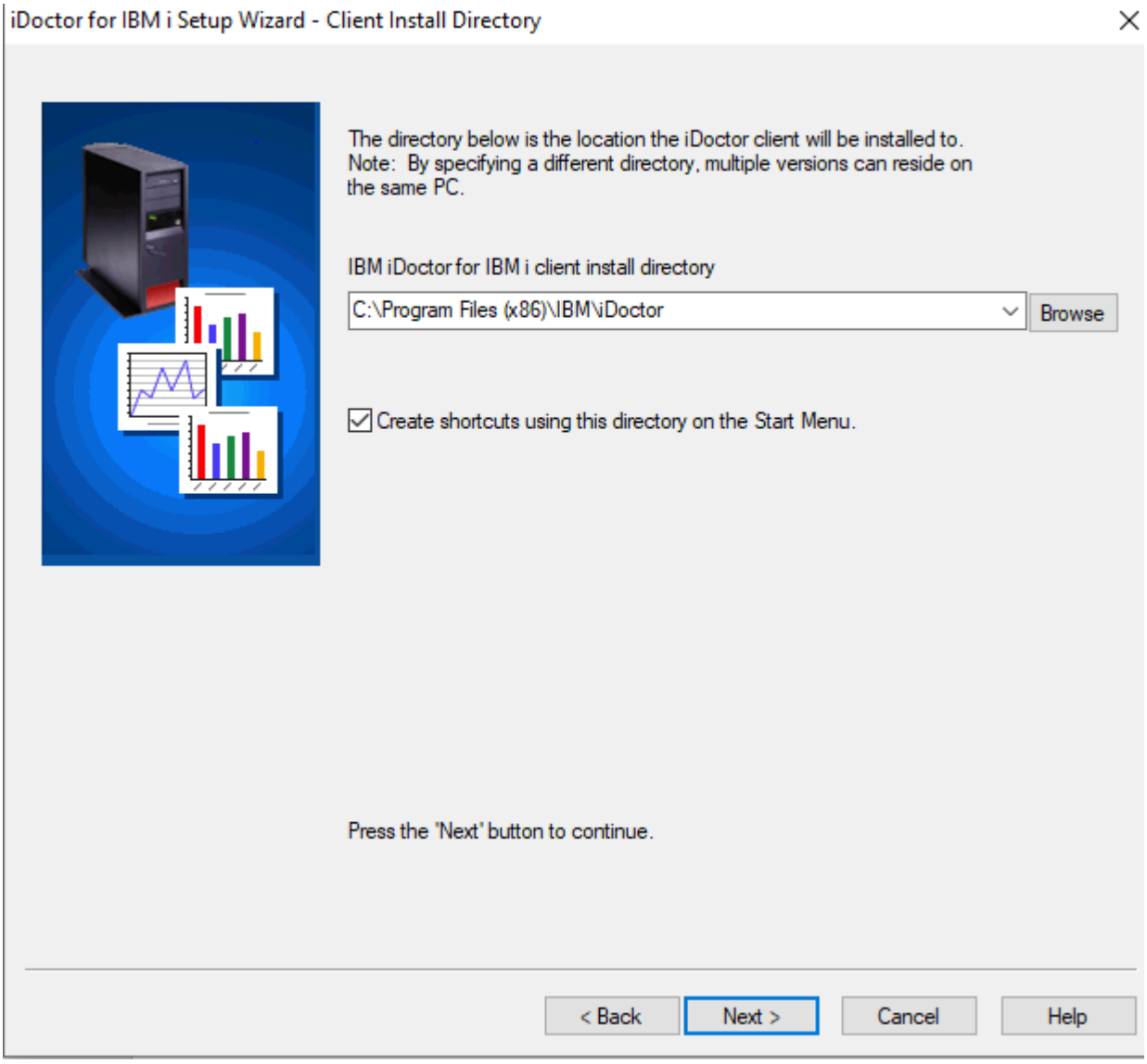
Also if you have SSL FTP you can check the box to attempt to use that method instead. If needed you may specify the FTP port to use with the Manually specify port option.

Clicking 'Next' on this screen will verify that the FTP connection is working between the PC and the server specified (or the 1st server specified if installing on multiple systems). Uncheck the option “Perform FTP connection check” to skip FTP verification.

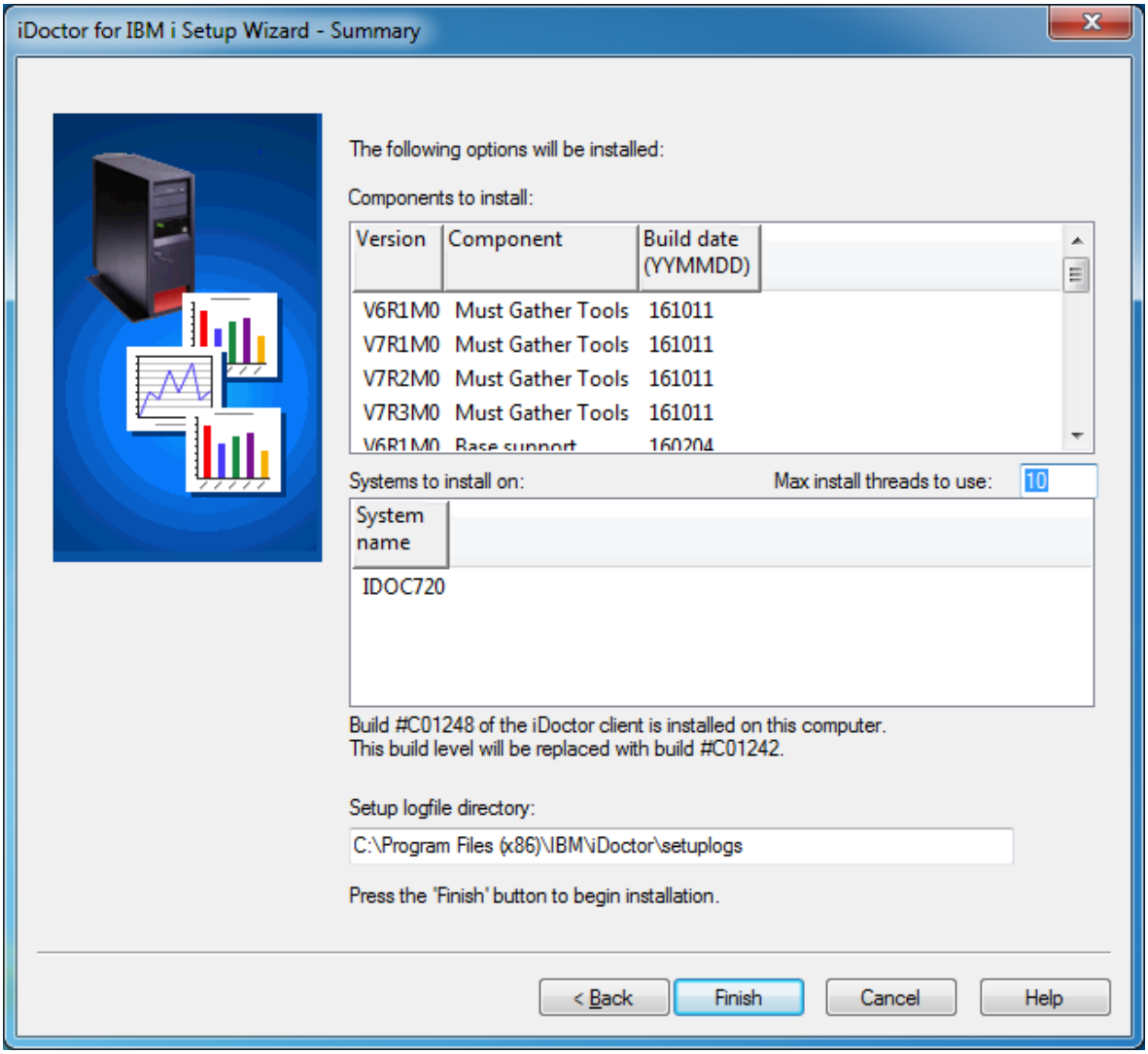


Step 9 This screen allows you to specify the directory the iDoctor client should be installed to.

Note: Installing multiple instances of the iDoctor client into different directories is now supported and possible if needed. Uncheck the create shortcuts using this directory on the Start menu option on this screen if desired. You will need to make multiple shortcuts manually if you are using multiple versions of the iDoctor GUI.



Step 10 A summary of your selections appears on the final screen.



Step 11 Clicking the 'Finish' button will copy all of the files and run the commands necessary to install the server and/or client portion for the selected components. The server portion of the installation will take a few minutes.

After the install completes the setup log file (for each component installed) will be viewable if needed through the install program. To do this, first select the desired system and component from the progress window to view the setup log for, and then click the Display setup log button.

If any errors occur that you are unable to resolve on your own, send the log file to idoctor@us.ibm.com for assistance. The log file directory is listed at the bottom of the last panel of the Wizard and can be modified.

Note: In some cases, certain component installations may fail. This can occur because of rare timing issues with our multithreaded install process. Typically selecting the failed component and using the **"Reinstall selected"** will succeed in installing the component on the 2nd attempt. Should problems persist, contact us however. If you wish to avoid this issue, you can set the "Max install threads to use" value to 1 although the install will take much longer especially on multiple LPARs.

2.5 Manual install steps

This section describes the steps to install iDoctor on a system running IBM i 7.2 or higher. Using the GUI install wizard is preferred in most instances but if this is not possible, these steps are the only other way to get iDoctor on the desired system.

NOTE: These steps DO NOT apply to 7.1 and earlier releases. Please see the applicable older version of the documentation for those instructions.

Requirements:

1. The user profile performing the installation must have *SECADM, *ALLOBJ special authorities. Otherwise all objects may not be restored properly.

2. The system value QALWOBJRST must be set to *ALL or certain objects will not be installed properly. Temporarily change this system value if necessary, in order to perform this install. Remember to change it back to what it was when you started after the manual install has completed.

3. FTP on your IBM i must be enabled and ready for use. These steps use the Windows command prompt FTP command which does NOT support SSL FTP on port 990. If you require this, then you will need to get the SAVFs onto your system using another method of your choice.

2.5.1 Windows command prompt FTP method

Step 1 Download the SAVFs that you intend to install from either
http://public.dhe.ibm.com/services/us/igsc/idoctor/iDocInstallFull.exe (all of them) or
http://public.dhe.ibm.com/services/us/igsc/idoctor/web/ (pick and choose desired ones.)

Step 2 Examine the directory where you created/saved the SAVFs to ensure that you have the required save files.

Each set of .savf files are used to install a particular iDoctor component as follows:

BA*.savf = Library QIDRGUI (GUI support library)
JW*.savf = Library QIDRWCH (Job Watcher and all subcomponents)
PA*.savf = Library QIDRPA (PEX Analyzer)
MG*.savf = Library QMGTOOLS (Must Gather Tools)

Note: From this point on some steps or commands issued will indicate which component they apply to. BA, JW, MG or PA. The GUI steps are required to install any of the components.

Step 3 Upload the files (matching the system's VRM) to the server. This can be done manually or by using the uploadSAVFsToIBM.bat file. You can modify this batch file to best suit your needs.

Uploading the SAVFs using the batch file "UploadSAVFstoIBMi.bat"

If using the batch file, you will pick the library to send the save files to and they will be created in save files named BAVxRy, JWVxRy, PAVxRy and MGVxRy.

Uploading the SAVFs manually

If you don't use the batch file, then the following commands are needed to send the data to the IBM i. In this example it is assumed that library QIDRINST is used to store the save files.
Note: *Please replace VxRy with V7R1, V7R2, etc.*

Run these commands from a green screen session to the IBM i:

DLTLIB QIDRINST

CRTLIB QIDRINST

CRTSAVF FILE(QIDRINST/BAVxRy)

(JW)
CRTSAVF FILE(QIDRINST/JWVxRy)

(PA)
CRTSAVF FILE(QIDRINST/PAVxRy)

(MG)
CRTSAVF FILE(QIDRINST/MGVxRy)

Now open a Windows Command prompt and issue the following commands in order to send the required save files to the server. **Note: Please replace values like <SystemName> with the system you need to FTP the save files to and VxRy with V7R1, V7R2, etc. depending on the OS release you are installing.**

Note: If your system requires secure SSL FTP on port 990 then you will need to download and use software that allows this. These steps are only for unsecure FTP using Windows Command Prompt FTP command.

CD C:\iDoctor\iDoctorInstall

FTP <SystemName>

BIN

Cd QIDRINST

Put BAVxRy.SAVF

(JW)
Put JWVxRy.SAVF

(PA)
Put PAVxRy.SAVF

(MG)
Put MGVxRy.SAVF

Step 4 From a green screen window, run the following commands:

CHGSYSVAL QALWOBJRST *ALL
- you may want to note the existing value first so you can change it back when done

(BA/GUI)
DLTLIB QIDRSBA

DLTLIB QIDRGUI

RSTLIB QIDRSBA *SAVF SAVF(QIDRINST/BAVxRy) MBROPT(*ALL) ALWOBJDIF(*ALL)

QIDRSBA/INSTIDOCBA
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server Base install." is returned

QIDRSBA/INSTSBSD JOBQ(QGPL/QIDRJW) CRTJOBQ(*YES) SBSD(QSYS/QIDRJW) CRTSBSD(*YES) POOLID(1)

(JW)
DLTLIB QIDRSJW

DLTLIB QIDRWCH

RSTLIB QIDRSJW *SAVF SAVF(QIDRINST/JWVxRy) MBROPT(*ALL) ALWOBJDIF(*ALL)

QIDRSJW/INSTIDOCJW
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server Job Watcher install." is returned by the command.

(PA)
DLTLIB QIDRSPA

DLTLIB QIDRPA

RSTLIB QIDRSPA *SAVF SAVF(QIDRINST/PAVxRy) MBROPT(*ALL) ALWOBJDIF(*ALL)

QIDRSPA/INSTIDOCPA
- verify that the message "CPF9898 - SUCCESSFUL iDoctor server PEX Analyzer install." is returned by the command.

(MG)
DLTLIB QMGTOOLS

RSTLIB QMGTOOLS *SAVF SAVF(QIDRINST/MGVxRy) MBROPT(*ALL) ALWOBJDIF(*ALL)

Step 5 From a green screen window, run the following commands:

DLTLIB QIDRINST

DLTLIB QIDRSBA

DLTLIB QIDRSJW

DLTLIB QIDRSPA

Note: The required PTFs listed on the website must also be loaded and applied for the applicable component and OS release before collecting performance data.

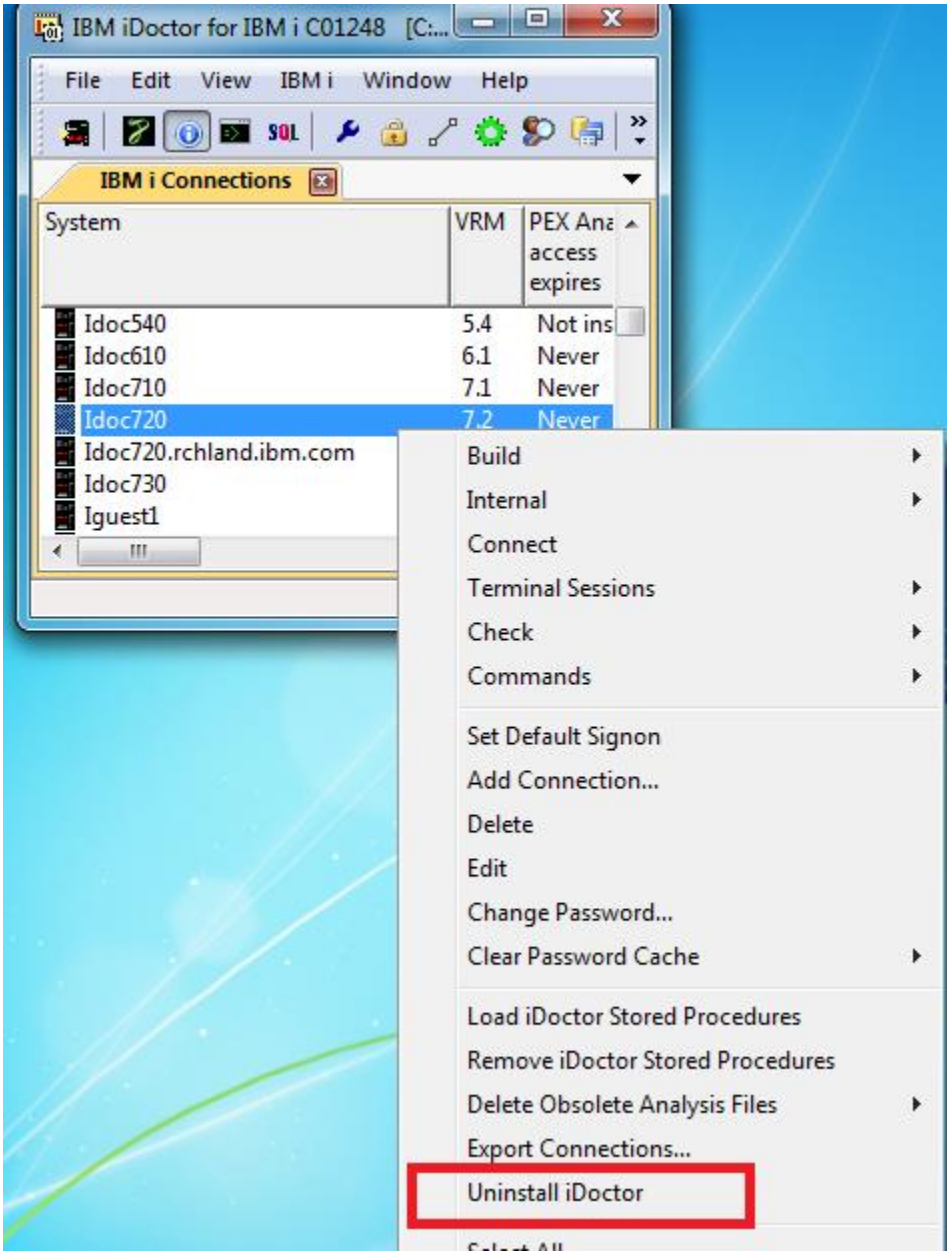
If you are unable to complete the manual installation successfully contact idoctor@us.ibm.com for assistance.

2.6 Uninstall

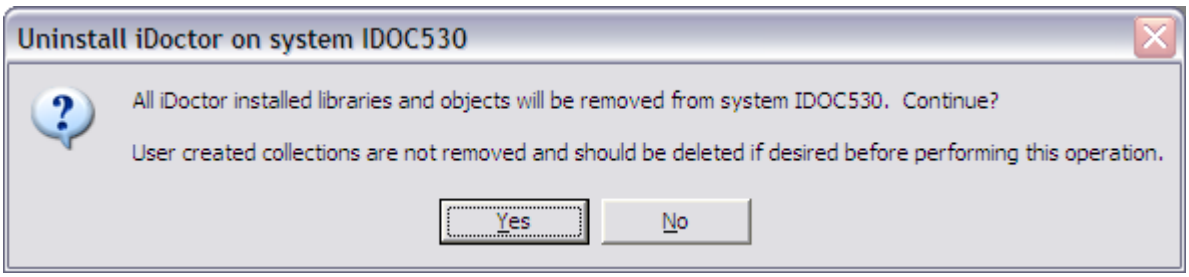
2.6.1 Server Uninstall (via GUI)

To remove iDoctor from an IBM i, several objects and libraries created during the installation must be removed.

From the GUI, within the IBM i Connections View, you may right-click the desired system to remove iDoctor from and then pick the “Uninstall iDoctor” menu option.



You will be asked to confirm this before continuing:



A CL program called QIDRUNINST is called which performs the actual uninstall of the iDoctor server-side objects and the status of that program and job log is available in the GUI from the view shown at the bottom of the screen (just right-click the view and use the appropriate menu to display the job log).

2.6.2 Server Uninstall (via green screen)

If you don't have the GUI installed, then simply find *PGM QIDRUNINST in QIDRGUI

Assuming it exists, then copy that program to QTEMP and then call it.

CRTDUPOBJ OBJ(QIDRUNINST) FROMLIB(QIDRGUI) OBJTYPE(*PGM) TOLIB(QTEMP)

CALL PGM(QTEMP/QIDRUNINST)

This program upon completion may complain about 1 or more objects not being removed. This is typically normal because several objects may not actually even exist. But to check that the only way is to review the log manually and ensure everything you wish to remove has been deleted. Any locks of course would prevent that.

If that program does NOT exist, then delete all libraries named QIDR* and that will remove 99% of iDoctor from your system. Of course, keep in mind this may also delete user data you may or may not care about in these libraries so be careful with this. If you need further assistance with this contact idoctor@us.ibm.com.

2.6.3 GUI Uninstall

To remove the iDoctor GUI uninstall it just like any other GUI application on Windows.

On Windows 7 or higher, use the Control Panel -> Programs and Features option, then look for the IBM iDoctor for IBM i program entry in order to remove it.

If you need to run the iDoctor GUI uninstall program in a batch process, then call the iDoctorUninstall.exe with -silent command line parameter like this:
iDoctorUninstall -silent

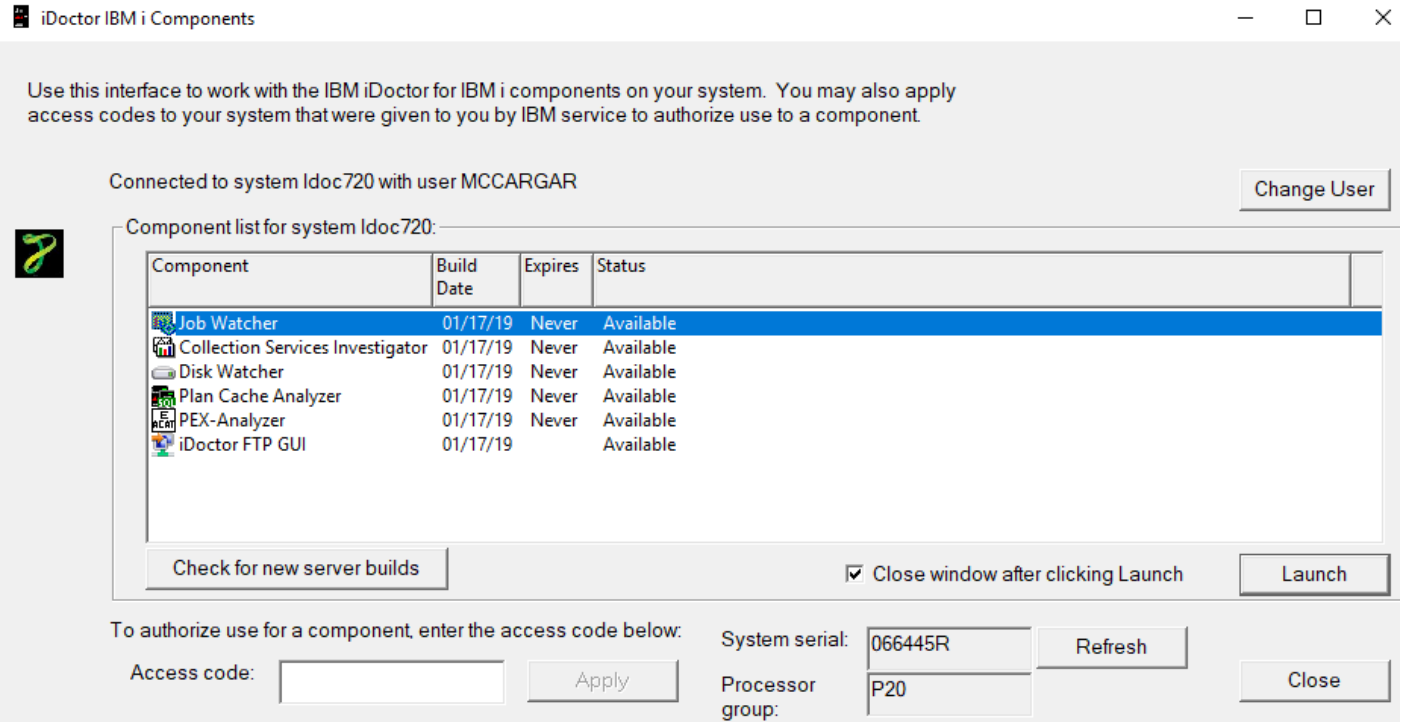
This will hide all message boxes and prompting. The uninstall program is installed in the same directory where iDoctor was installed (specified at install time).

2.7 Applying access codes

There are three ways to authorize use of iDoctor:

Option 1 Provide the access code using the GUI.

1. Make a connection in the IBM i Connections View to your IBM i. Right-click it, then use the Add connection menu and define the new connection.
2. Connect to the system (double-click it.)
3. In the components window there is an Access code text box at the bottom of the screen that can be used to apply keys. Copy and paste into there and press Apply (one key at a time.)

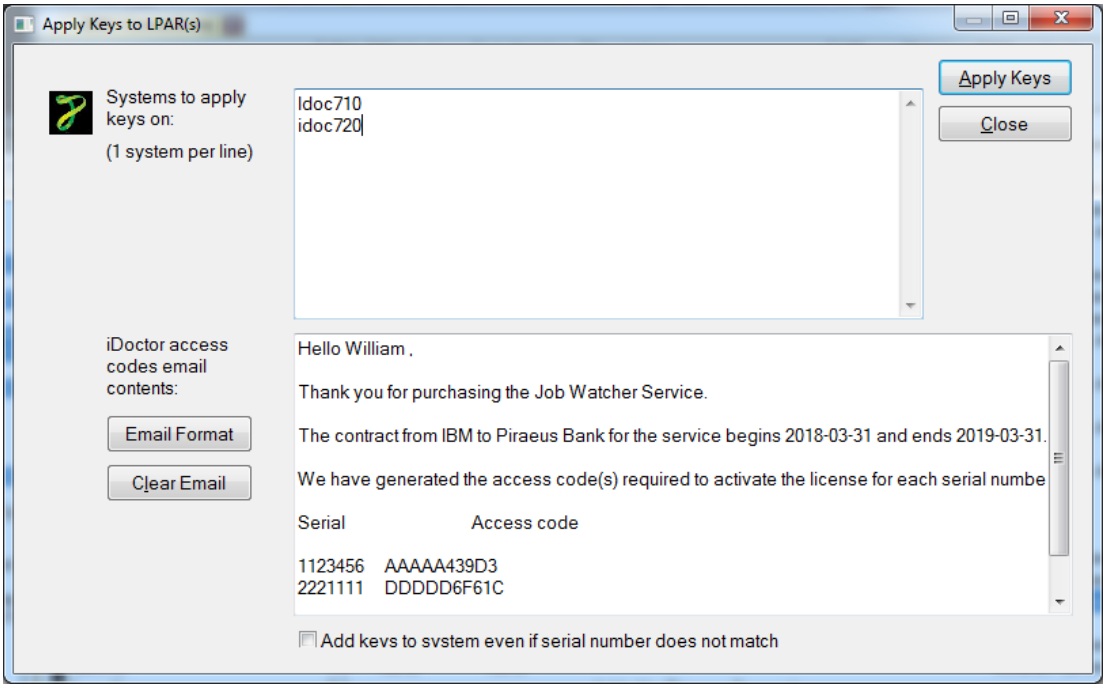


Option 2 By using the green screen command QIDRGUI/ADDPRDACS.

1. Open an interactive session to the server:
2. Type QIDRGUI/ADDPRDACS and press F4
3. Copy/paste (or type in) the access code that you were given by IBM Support.
4. Press enter.

Option 3: Use the **Apply Keys** menu from the IBM i Connections View.

1. Select one or more LPARs to apply keys on and right-click and use the Apply Keys menu.
2. Copy and paste the email received from IBM into the bottom text box of this screen. If you did not receive the keys directly from IBM you can request them using the email address idoctor@us.ibm.com or press the Email Format button to see the format that the email needs to be in order for this function to work correctly.
3. Press the Apply Keys button.

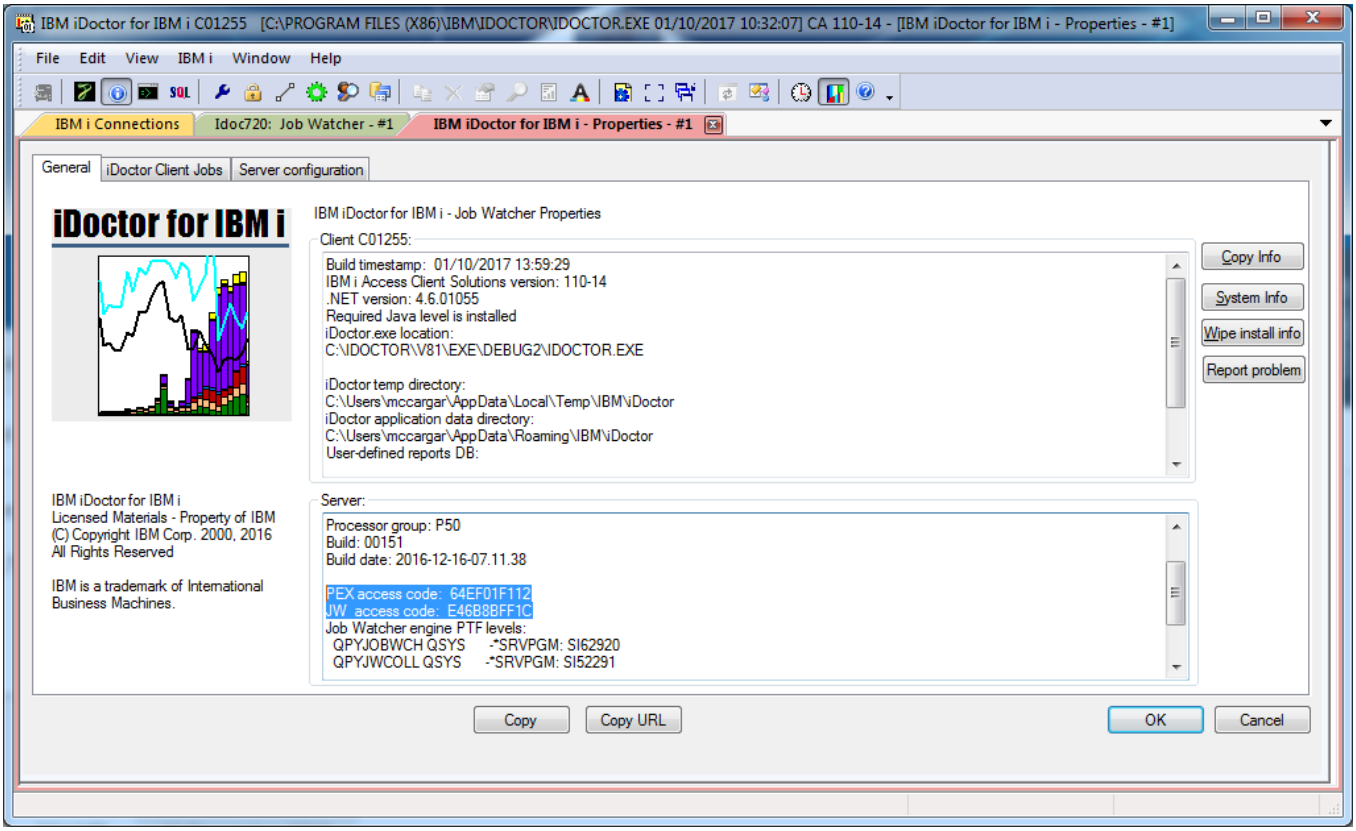


You are now authorized to use IBM iDoctor for IBM i. If you are using the tool with an evaluation code, your access will expire 45 days from the date it was created. **Note:** if you already had a trial code previously and request a 2nd one, the duration of the code you receive may be less than 45 days at the discretion of IBM.

2.8 Viewing access codes

To view existing access codes that have already been applied to an IBM i:

1) Open an IBM i component under the IBM i Connections View.
2) Right-click the component icon (i.e. Job Watcher or PEX Analyzer icon at the top of the tree) and use the Properties menu.
3) The applied keys are listed under the Server section of the window.



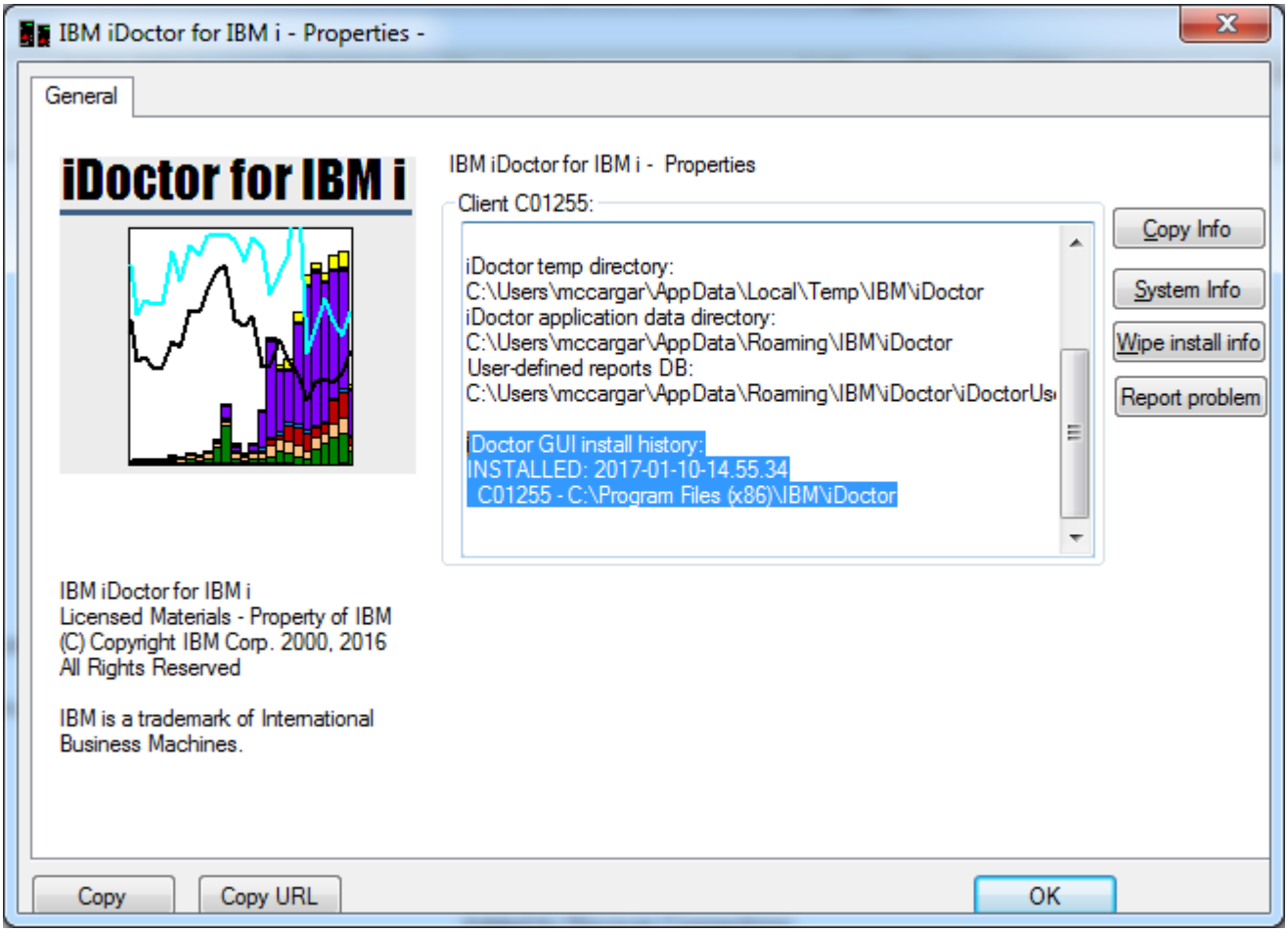
iDoctor Properties - General

2.9 GUI install history

In order to view a history of GUI client builds installed on the PC view the application properties and look at the bottom of the data in the client section. Click the help icon on the main window toolbar or use the Help -> About menu or use the component -> Properties menu to access the application properties

This information includes the date the build was installed, the build number and the location it was installed to.

Note: This information is only removed if the user presses the Wipe install info button on the application properties screen. It is not removed by uninstalling the iDoctor GUI.



iDoctor Properties - General

2.10 PTF Installation

Visit the Fix Central website if you need assistance with installing PTFs.

3 iDoctor for Performance Analysis

This chapter covers the components of performance and how they can be analyzed using the iDoctor tools.

3.1 Components of Performance

Note: Some images and wording in this chapter have been taken from Chapter 1 of the Job Watcher Redbook. You can find the original/complete text at
http://www.redbooks.ibm.com/redbooks/pdfs/sg246474.pdf

Several different components may affect performance on the server. These components consist of:

- CPU
- Disk I/O
- Other (seizes, locks, gates)

A further breakdown of CPU components might look like Figure 3-1.

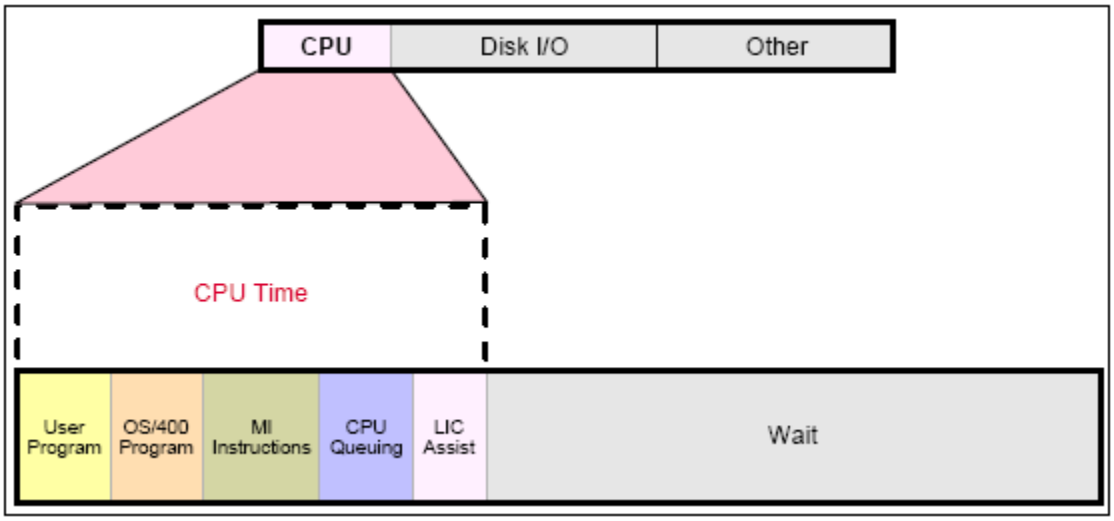


Figure 3-1 CPU components

Note: LIC assist tasks are defined as tasks running in the background doing work on behalf of a requesting job. Examples are DBL3xxx tasks that handle asynchronous disk I/Os on behalf of jobs, load/dump tasks (LD component) on behalf of save/restore tasks, and on behalf of other LIC tasks.

MI in these figures and elsewhere to refer to Machine Instruction–level instructions (below the operating system level), which is referred to in more current documentation as the Technology Independent Machine Interface (TIMI).

A further breakdown of Disk I/O components might look like the following figure:

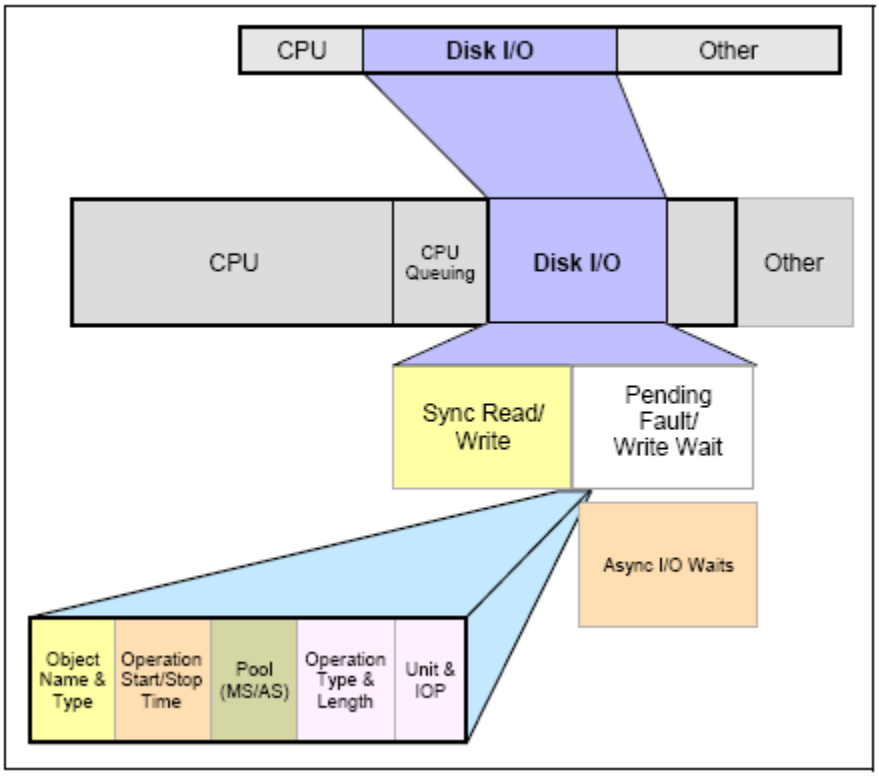


Figure 3-2 Disk components

Pending (page) Fault/Write *wait* applies when the system has determined to write some main storage data/program code to disk and there is a wait to actually perform that write to disk. A fault is the result of needing to bring new data or instructions into main storage when there is not enough available space, so least-used data or instructions have to be written to disk first.

Note that asynchronous I/O can also become momentarily synchronous (*Async I/O Waits*) because, for example, the target disk is already busy performing previously issued I/O operations.

A further breakdown of the Other components of performance might look like this:

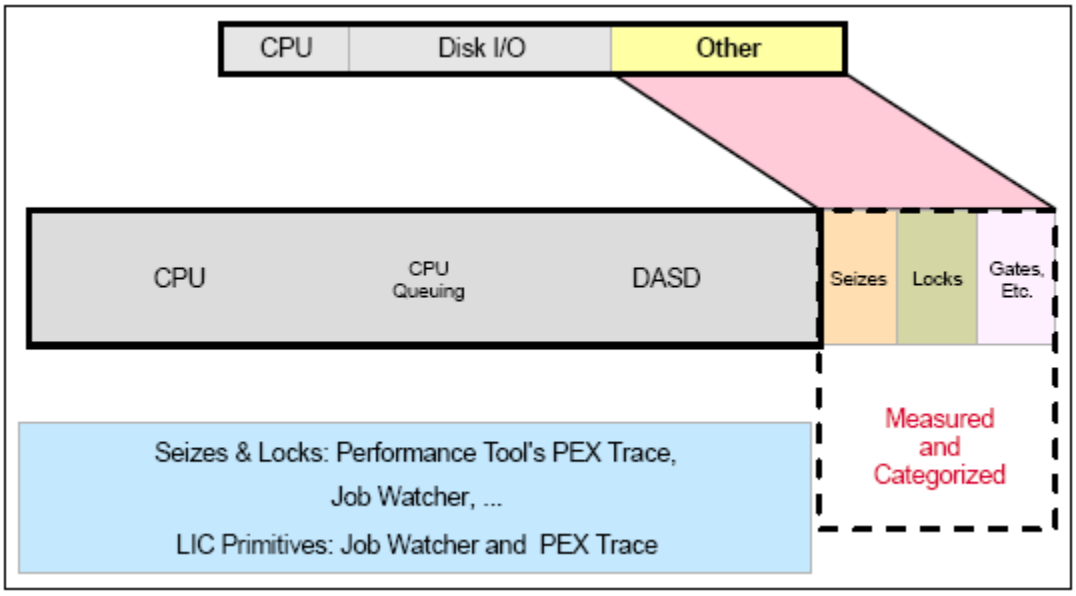


Figure 3-3 Other components

Locks include database record locks and object locks. These locks occur above the MI in the operating system.

Object lock contention is the conflict between threads involving objects. The OS will frequently obtain locks to perform many common operations like: opening a file, creating/deleting options, moving objects to another library, and changing object ownership.

Seizes and gates (a low-level synchronization function) are LIC-level lock/unlock or seize/release mechanisms. Job Watcher and PEX Analyzer provide more details, including the job/thread waiting on an object, the job/thread holding the object, and the object name and type. Job Watcher measures and categorizes gates (wait and related information).

Note: Seizes and gates occur below the MI (TIMI) and do not time out. Locks occur above the MI and will eventually time out. Job Watcher can show very long seizes and gates well in the iDoctor GUI. In rare situations, if a system has a large numbers of jobs in seize contention it is not uncommon for these seizes to last for minutes without any CPU use occurring.

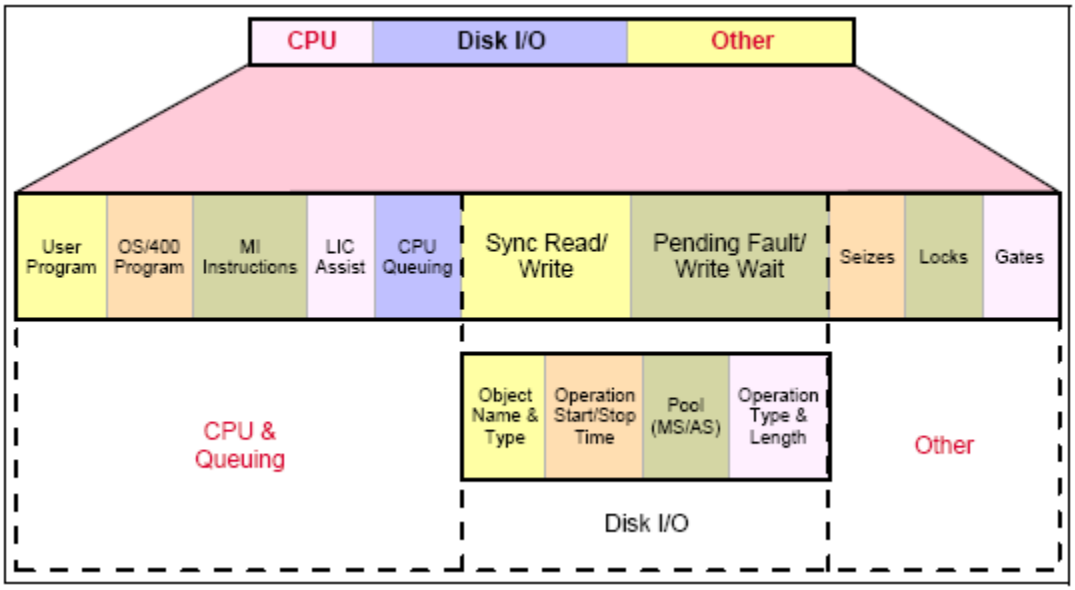


Figure 3-4 Summarization of Performance Components

Figure 3-4 represents a moderately detailed summarization of the content in the preceding three figures on CPU, Disk I/O, and Other.

One or several of these components can be a bottleneck and a problem for some transactions. While the thought of replacing the entire system with a newer, faster model may be an alternative, it might not be the most effective or efficient use of time and resources. With the help of the proper tools, performance can be improved by diagnosing these bottlenecks. By properly balancing system resources, all jobs can run at their optimal level with minimal resource conflicts. Job Watcher , Disk Watcher and PEX Analyzer measure and present all of these components. The strengths of each iDoctor component and how they may be used to help you balance your system resources is covered in the next sections.

3.2 Job Watcher

Job Watcher provides information about all 3 components of performance: CPU, Disk I/O and Other (waits/seizes/gates/locks). It is a tool best used for wait analysis but it can also produce useful graphs for CPU and Disk I/O analysis as well.

Job Watcher consists of:

- Tools for collecting data
- Tools for analyzing and viewing the collected data

Job Watcher returns near-real-time information about a selected set of jobs, threads, or licensed internal code (LIC) tasks. It is similar in sampling function to the system commands WRKACTJOB and WRKSYSACT in which each refresh computes delta information for the ending snapshot interval. In the Job Watcher, these refreshes can be set to occur automatically, even as frequently as every 5 seconds. Better yet, Job Watcher harvests the data from the jobs/threads/tasks being watched in a manner that does not affect other jobs on the system while it is collecting.

Job Watcher collected data includes the following:

- Standard WRKSYSACT type information
  - CPU
  - DASD I/O breakdown
  - DASD space consumption
  - Current user profile
  - And more
- Expanded details about types of waits and object lock/seize conditions
- Last run SQL statements syntax
- Program/procedure call stack, up to 1000 levels deep

- Job Watcher also includes some data not available anywhere else *in real time*:
  - Seize time (this includes objects being locked at the operating system level and objects being seized at the microcode level)
    - Holder and waiter job and thread
    - Specific LIC block point ID (also referred to as ENUM)

  - A breakdown of all other waits that occurred
    - Details about the current wait:
    - Duration of the wait
    - Object being waited for
    - Conflicting job/thread/task information
    - Specific LIC block point ID

- Call stacks
- SQL statements, host variables, prepared statement arrays, open cursors
- Communications data
- Activation group statistics
- J9 JVM statistics

The data created by the tool is summarized in many different types of reports and graphs via the iDoctor GUI. The client provides a quick picture of what is happening either system wide or on a per-thread basis when multiple different threads are being analyzed.

The GUI provides the flexibility to select one or more jobs or a time range and drill down for the details while the collection is in progress or after it has ended.

A typical situation for deciding to use Job Watcher is for a job that is taking a long time to run but is using hardly any CPU resource and disk I/Os are not particularly excessive. Job Watcher is an excellent tool to help you determine job waits, seizes, and other types of contention. Identifying why a job or multiple jobs or threads are not doing anything when they should be, is a primary situation to demonstrate a key set of Job Watcher capabilities.

Tip: Run Job Watcher when your system is running normally, and you are not having performance problems. This gives you a health check of your system so that when it does have performance problems, you have a baseline to make comparisons to. You might want to run Job Watcher over your system for a

period in the morning, afternoon, and possibly during the evening when you might have more batch-type jobs running.

You could also run Job Watcher over specific key jobs on your system when they are running well for baseline-comparison purposes.

3.3 Collection Services Investigator

Collection Services Investigator provides information about all 3 components of performance: CPU, Disk I/O and Other (waits/seizes/gates/locks). It is a tool best used for wait analysis (ideally in conjunction with Job Watcher) but it can also produce useful graphs for CPU and Disk I/O analysis as well.

Collection Services Investigator provides the user with the ability to analyze the performance database files produced by Collection Services. Collection Services is like Job Watcher in the statistics collected, but the primary difference is the interval size in Collection Services is usually much longer (5-15 minutes vs 5 seconds in Job Watcher).

Collection Services is a no-charge part of IBM i that continually collects sets of performance categories on the system. The data is collected into management collection objects that must be converted into database files before analysis with Collection Services Investigator can be performed. Either use the CRTPFRDTA command or the Collection Services functions within IBM Systems Director Navigator or System i Navigator to create these database files.

Collection Services does not provide the following metrics that Job Watcher collects:

- Call stacks
- SQL statements
- Activation groups
- JVM statistics
- Object waited on
- Holding thread

Despite this Collection Services is useful for a big picture look at how a system has been performing over longer periods of time than what is typically collected by Job Watcher (days/weeks instead of minutes/hours).

Collection Services Investigator does provide many types of data not found in Job Watcher (or some other tools):

- TLBIEs
- Multiple LPAR stats
- Disk graphs
- Memory pool graphs
- External storage statistics
- Save/Restore
- IASP bandwidth estimations
- Disk/hardware configuration

3.4 Disk Watcher

Disk Watcher provides the user with the ability to collect either a statistical summary of disk performance data or a trace of all disk I/O events that occur on a system. The trace mode is recommended as it provides more options for analyzing the data and determining potential disk problems.

The Disk Watcher GUI provides many graphs with drill downs for each mode of collection (statistical or trace). Using Disk Watcher the user can take a trace, then summarize the trace data into an interval size desired for the purpose of easily graphing the statistics at either or broad or detailed level.

Disk Watcher is available at releases V5R4 and higher.

3.5 PEX Analyzer

PEX Analyzer assists the user with the analysis of PEX (Performance Explorer) data. PEX is a component of IBM i. The server-side of PEX Analyzer includes a command STRPACOL that simplifies the process of creating a collection by wrapping the IBM i commands ADDPEXDFN, STRPEX and ENDPEX into one step.

The client side of PEX Analyzer includes many graphing and query capabilities that allow a user to quickly identify performance bottlenecks.

This component is available for a trial evaluation or purchase via this website.

3.6 Must Gather Tools

Must Gather Tools is designed to assist IBM support personnel quickly and easily capture data needed to solve problems of a variety of types. This is a green screen menu driven tool.

After restoring the QMGTOOLS library, add the library to your library list and use the GO MG command to start using this interface.

3.7 Performance Analysis Using the iDoctor GUI

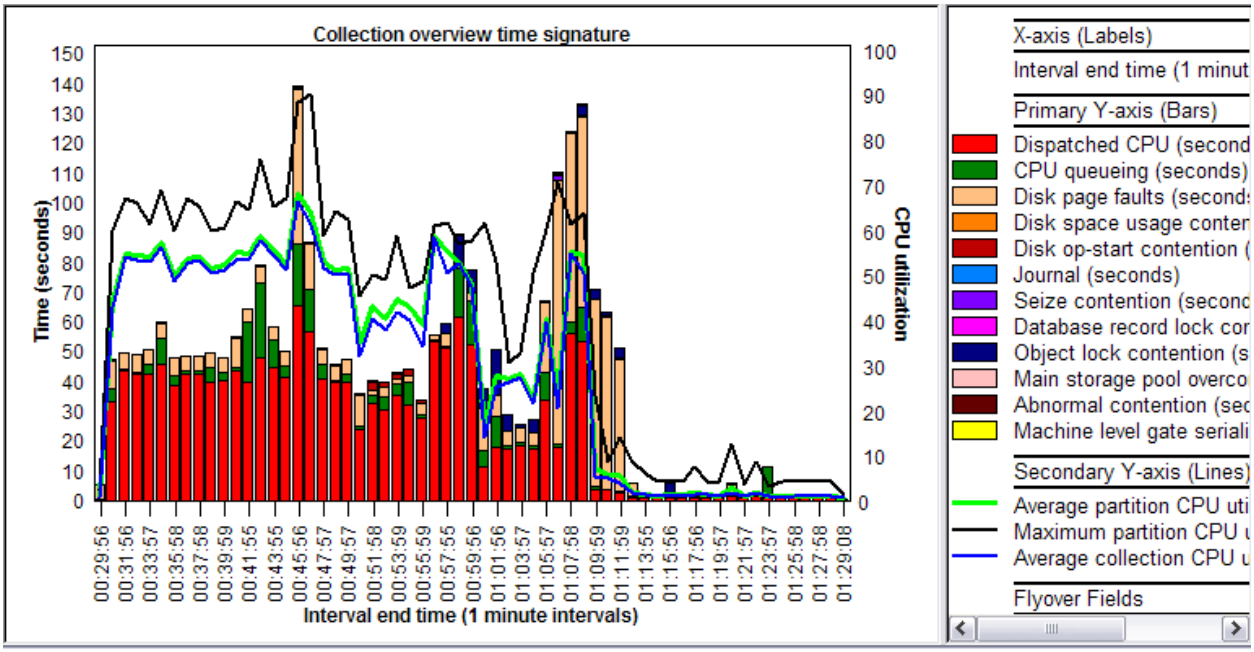
All iDoctor components provide a similar GUI experience when analyzing performance data. Performance data resides within a collection. Each component in iDoctor shows the libraries and collections that exist on the current system that contain data of the desired type (Job Watcher displays Job Watcher collections). The collections contain the graphs and tables available categorized based on the different types of statistics available in the data.

The hierarchy of these elements looks like this:

System/Component
- Libraries
  - Collections
    - Report folders
      - Reports (graphs and tables)

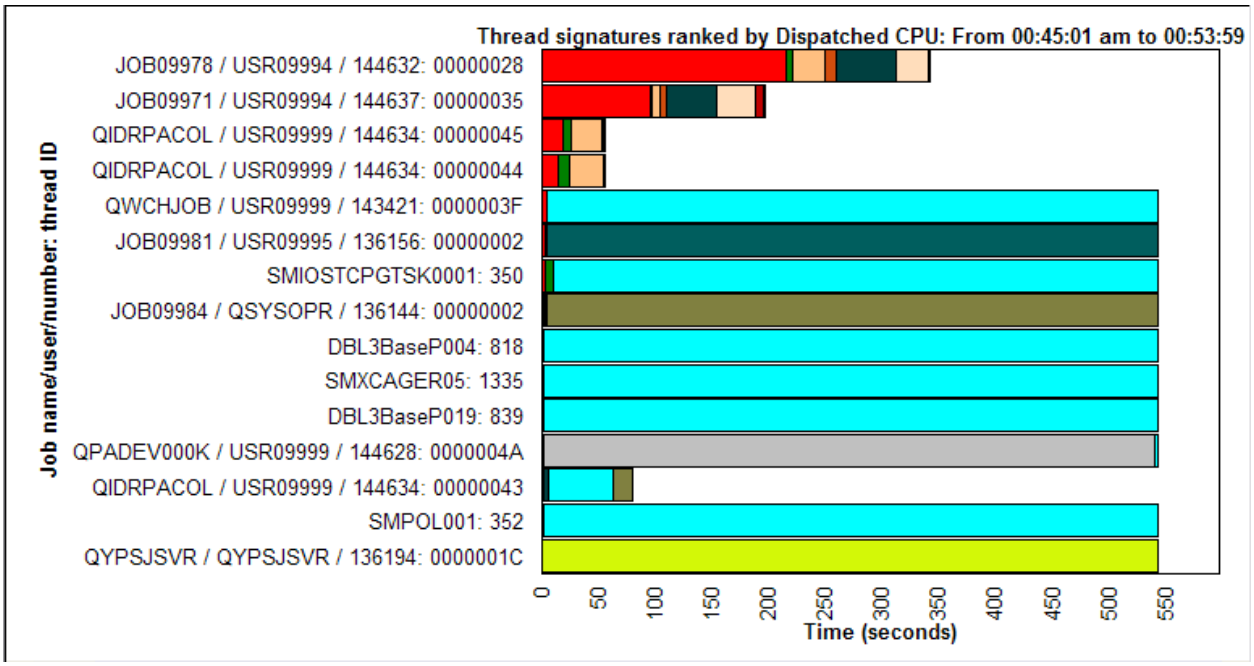
Graphs in iDoctor generally consist of one of three distinct types:

Collection-wide by time interval
These graphs show some set of statistics across the entire collection over time. The Time interval size (collection interval size, 30 secs, 1 min, etc) is configurable by clicking on the clock icon on the toolbar. This gives the effect of zooming in or out by showing more or less detail.



Collection-wide by time interval graph example

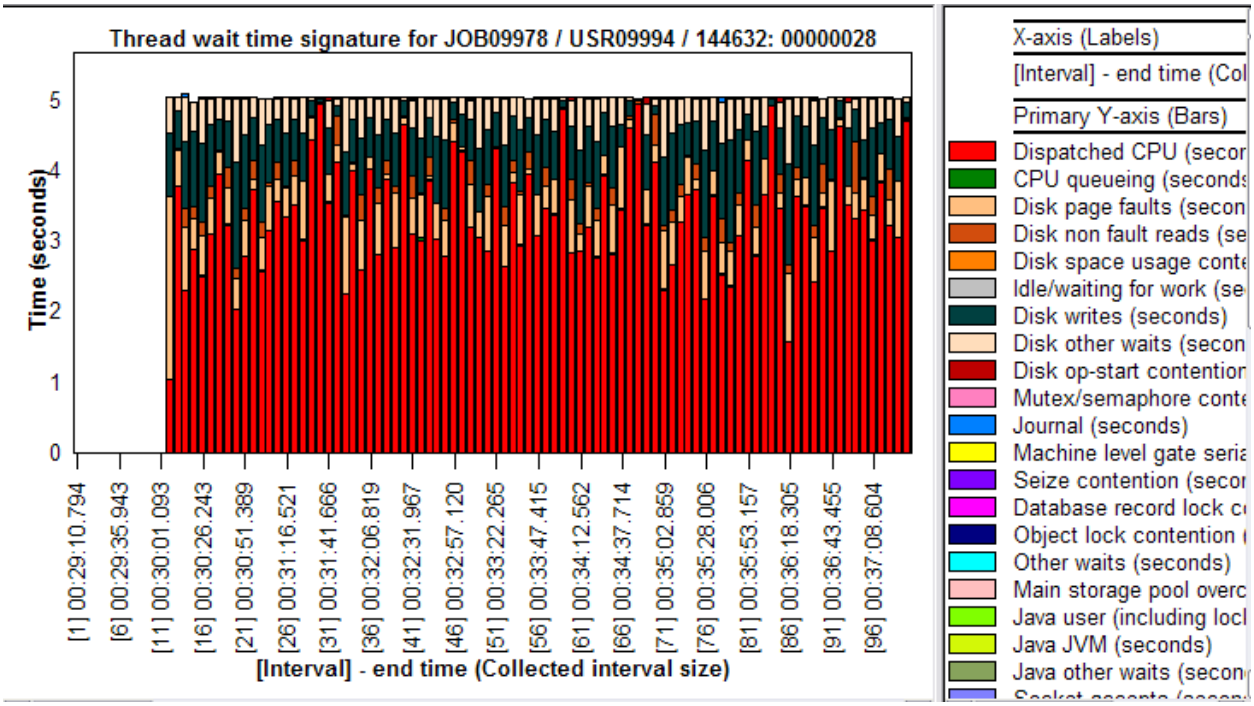
Rankings graphs (such as thread, job, disk unit or pool)
These graphs rank objects by statistics for the desired time period. The time period could be the entire collection or the selected time interval that was drilled into from the previous graph type.



Rankings graph example

Selected object(s) by time interval
These graphs are usually a drill down from the previous graph after selecting one or more objects. They are very similar to the collection-wide by time interval graphs with the key difference that they only show data for the objects selected.

This graph often provides the ability to drill up to the previous graph for the selected time period. This gives the user the ability to make comparisons with other objects (threads or disk units for example).



Selected object(s) by time interval graph example

4 The iDoctor GUI

This chapter covers the iDoctor client and major functions provided in all components. Some common functionality is also covered separately in additional chapters (i.e. Power, Data Viewer.)

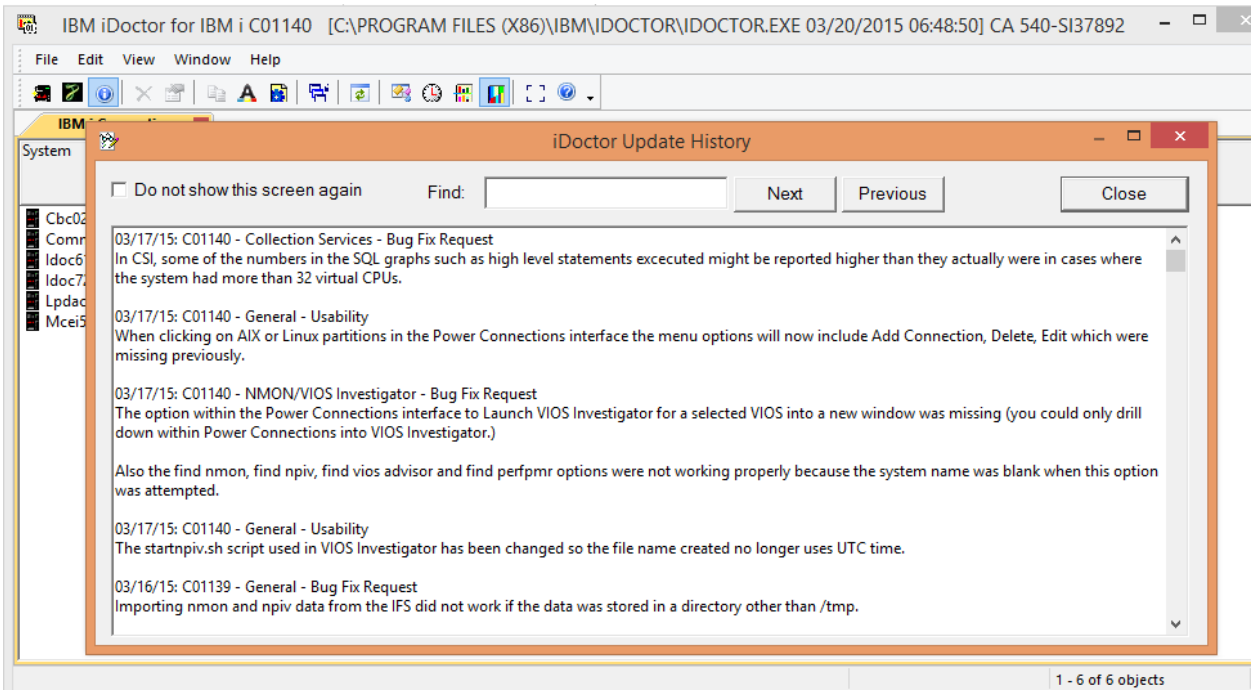
Note: This chapter also covers the IBM i command panes (WRKOBJ Pane, WRKOBJLCK Pane, WRKLNK Pane, etc) which can be used on any IBM i system and do not require an access code.

4.1 Starting iDoctor

iDoctor may be launched in one of several ways:

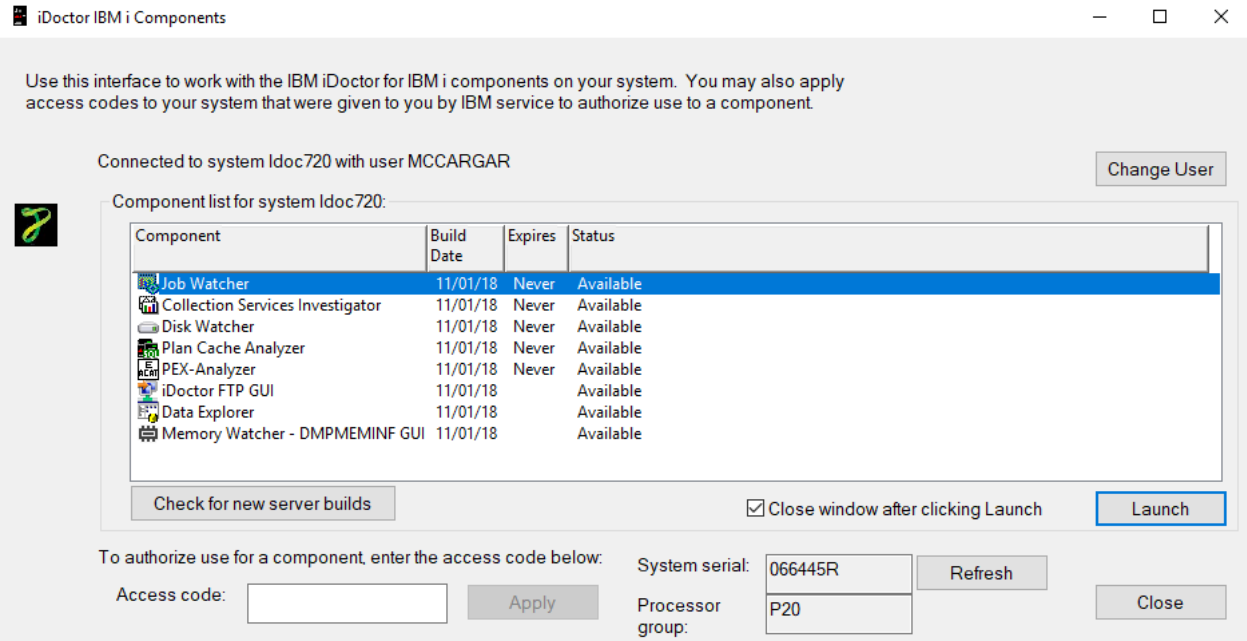
1. via the IBM iDoctor for IBM i desktop icon
2. On Windows 7; Start -> Programs -> IBM iDoctor for IBM i -> IBM iDoctor for IBM i) or
3. On Windows 8/10: Start -> IBM iDoctor for IBM i
4. By double-clicking an iDoctor created .idr (session) file.
5. By passing an idoctor:// URL string to a web browser. This works currently with Edge, Firefox or Chome (latest version tested, it did not work with old versions of Chrome!). To generate this string, look for a “Copy URL” button or menu where this option is available.

The iDoctor Main Window will initially show a list of recent changes made in the iDoctor Update History window.



Pressing Close on this screen will return you to the Main Window where you can define a connection or pick an existing connection on the IBM i Connections View in order to begin looking at data.
Right click in the IBM i Connections View and use the Add Connection menu to add a connection or double-click on a system name shown to connect.

Next you will be prompted for your username and password and then shown the iDoctor components window that displays the components available:



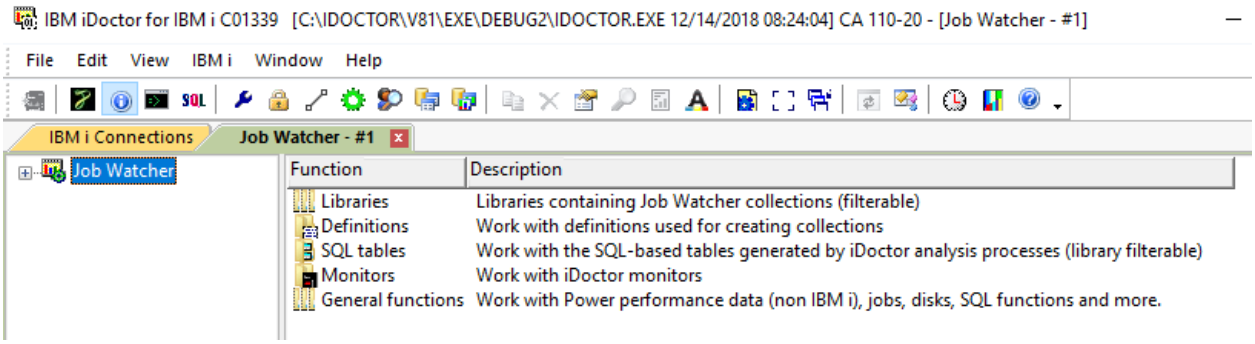
If the status indicates the component is not available due to a missing access code you can enter the access code at the bottom of this window.

If the status message indicates the access code is invalid, these are the possible reasons:

1. Serial number provided to IBM was incorrect.
2. The wrong OS level of the iDoctor server code is installed. This can happen (for example) if you have iDoctor 5.4 installed and then upgrade the system to 6.1 without also updating the iDoctor server code.
3. The access code entered was for the wrong component (you asked for PEX but really wanted Job Watcher)
4. Access code generation error (on IBM's side). This could be due to an administrative error, website problem, etc.

Note: If the iDoctor components window does not list the component as "Available" but lists a message about the client and server build levels not being up to date, you can still continue to launch the component anyway. But in this situation some functionality may not work correctly. If problems occur, then it's typically recommended to download the latest version and install it on both client and server.

Assuming the desired component is listed as "Available", double-clicking on it will open the component view for it. Component views display all reporting options available for the desired system and component. Each view can be resized, tiled or manipulated at runtime. Multiple component views may be in use within the same Main Window as desired. To launch a second component view you would either need to revisit the Connections View and double-click the desired system a second time or simply uncheck the option "close window after clicking Launch" on the iDoctor components window.



Job Watcher component view example

4.2 iDoctor and Internet connectivity

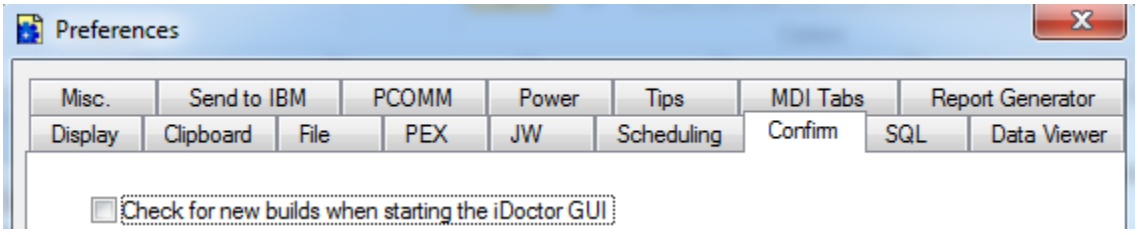
This section describes the functions performed behind the scenes at startup that require an Internet connection to be successful. If one is not available or blocked by a firewall on the PC, then these functions will NOT be performed.

4.2.1 Automatic client updates

When the iDoctor client starts, it will check if a newer client build is available. If one is available, you will be asked if it should be downloaded and installed.

By picking yes, the latest client will be downloaded, the current client will end, and the default web browser will be started to download the latest client update. After installation completes, the iDoctor GUI will be launched again.

Note: This check can be disabled by going under Preferences -> Confirm and unchecking the 1st option shown.

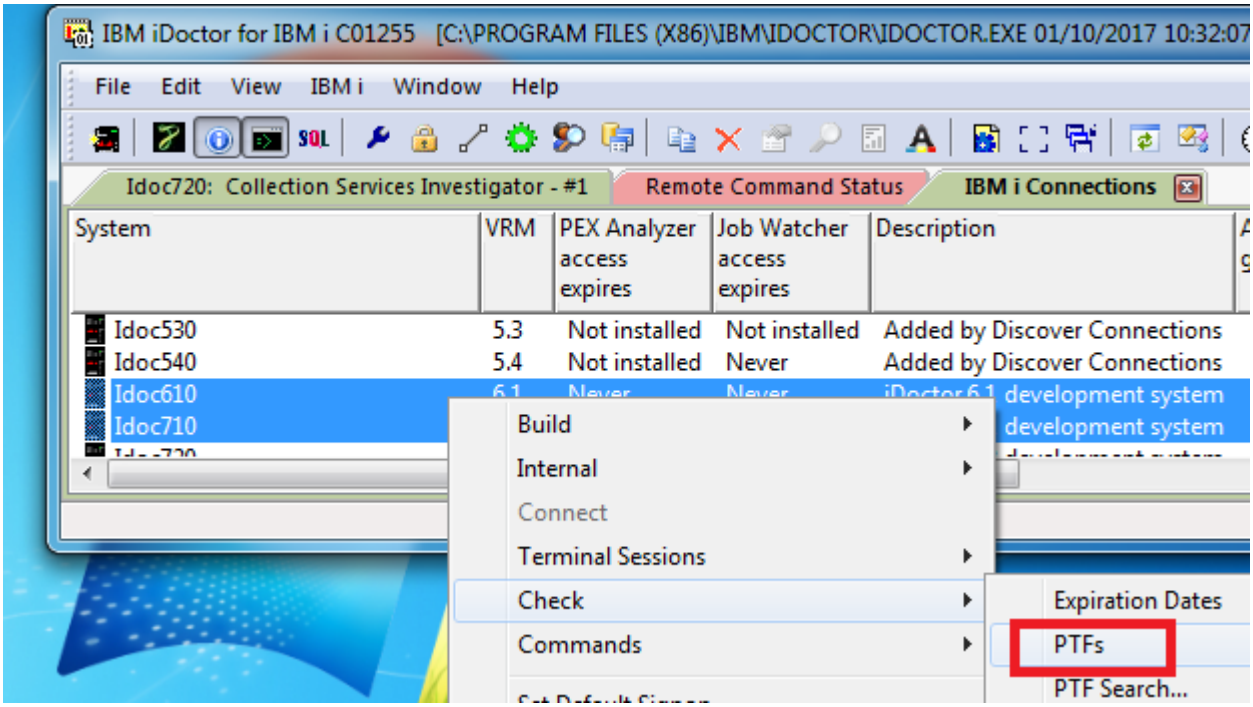


4.2.2 Automatic server PTF checking

When the iDoctor client starts, it will attempt determine the latest required PTFs for the components of iDoctor. These are used when attempting to start a collection to verify that the required PTFs have been installed on the server before continuing. If the required PTFs are not available, you will be shown a list of the missing ones.

You can override this checking and continue with collection anyway but it is generally not advised to do so without consulting first with IBM or at least checking that the PTFs missing are not associated with a system failure/crash.

Tip: You can also check PTF levels for 1 or more systems by selecting them from IBM i Connections View, right-clicking and using the Check PTFs menu.



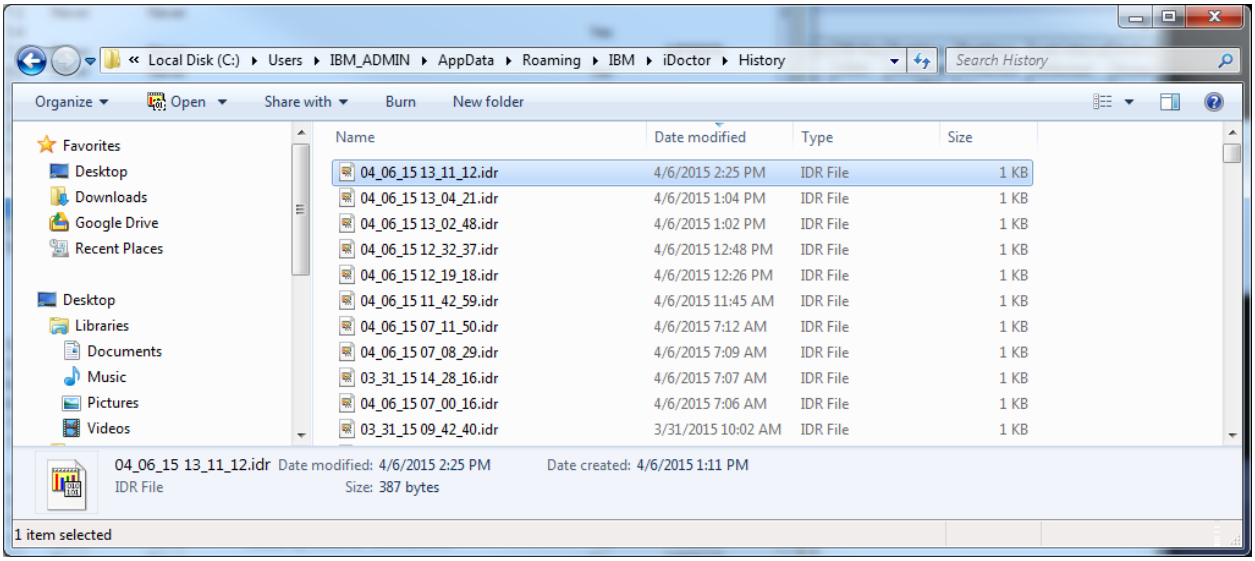
4.3 Sessions

Each time the GUI starts it will keep track of all the windows and reports that are **currently** opened and save them to a session file. These files are named *.idr and use the current PC date/time when the iDoctor session was started as part of the filename.

Session files are saved to the iDoctor history folder. The easiest way to see this location is to use the File -> Open iDoctor Session… menu option. Here is an example on Windows 7: C:\Users\IBM_ADMIN\AppData\Roaming\IBM\iDoctor\History

4.3.1 The current session

The current session file will be located in the iDoctor history folder and is the most recently created file. Sort by the modified date column in Windows in order to find it.



iDoctor history folder

4.3.2 Opening

Opening a previously saved session will destroy all views and Data Viewers you currently have opened and replace them with views and reports that were saved in the iDoctor session file. To perform this action, use the File -> Open iDoctor Session… menu.

Keep in mind that the session file does not store the actual performance data. If data has been deleted where it previously existed, then those reports will not be available, and the session should be resaved to a new file.

4.3.3 Saving

Saving an iDoctor session to a file of your choice can be accomplished using the File -> Save iDoctor Session as… menu option. The current session is also saved automatically each time you open/close a view or Data Viewer.

Note: If you wish to avoid having Data Viewer reports removed from the current session file when you need to shut down iDoctor, then close the Main Window while the Data Viewer windows are still open. From the Main Window you can click the X in the top right corner or use the File -> Exit menu.

4.3.4 Restore Previous iDoctor Session

This action can be performed by using the File -> Restore Previous iDoctor Session menu. Using this option will open the 2nd most recent iDoctor session from the history folder. The most recent iDoctor session is the current one.

4.4 MDI Tabbed Styles

The MDI Tabbed style interface was a new change to iDoctor's GUI design as of January 2015.

3 options for controlling the style used are available:

1) None – this is a classic Windows MDI without tabs
2) Standard – Provides tabs and allows users to tile and cascade but you cannot create groups of MDI tabs to compare with other tabs.
3) Grouped – Tabs **cannot** be tiled or cascaded but you can create groups of MDI tabs in order to make comparisons.

4.4.1 None

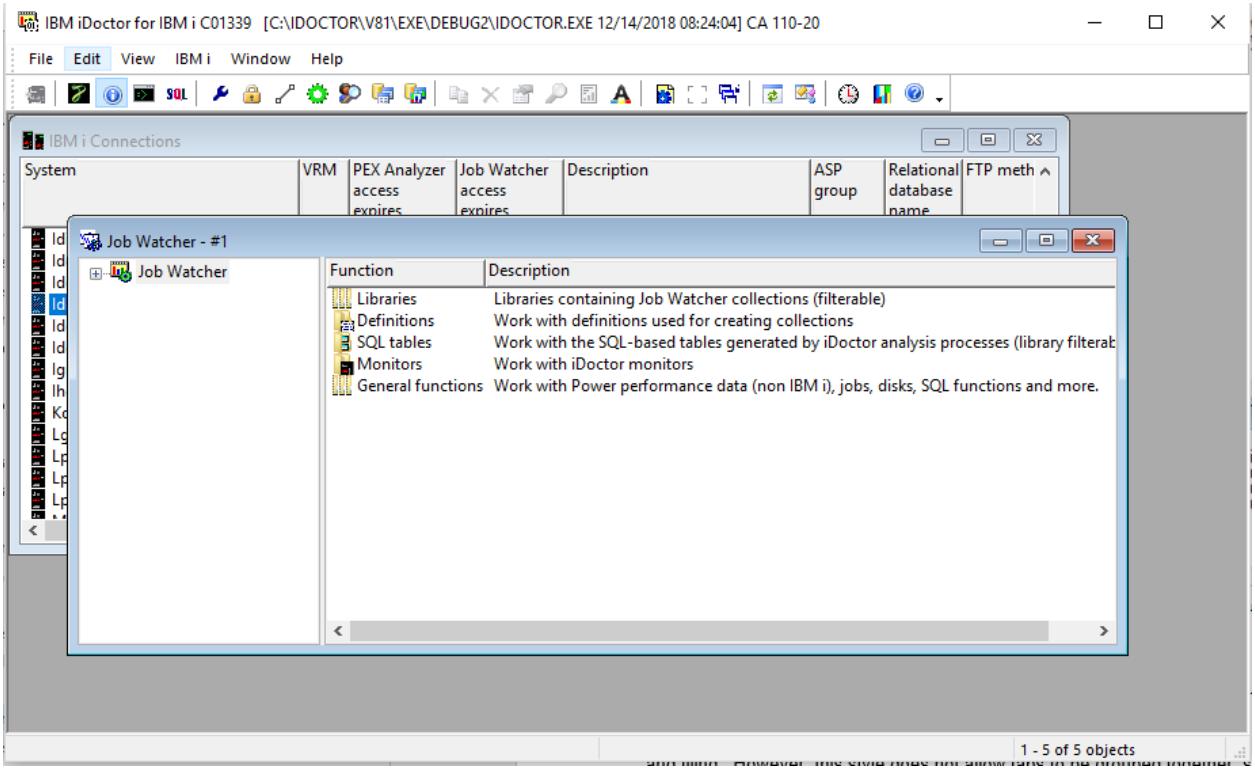
This style is the classic Windows MDI style where the frame window contains one or more child windows that can be individually moved/resized, cascaded or tiled horizontally or vertically. This setting is most like what was previously used in iDoctor before 2015.

If "None" is used, then some newer features are not available:

1) The View -> Full Screen option.
2) You will not be able to group several views/tabs together to make comparison with another set of views/tabs.

However, when using "None" you will be able to use the Windows -> Cascade, or Windows -> Tile menus to rearrange the views shown.

An example of the "None" MDI Tabbed Style in use is:

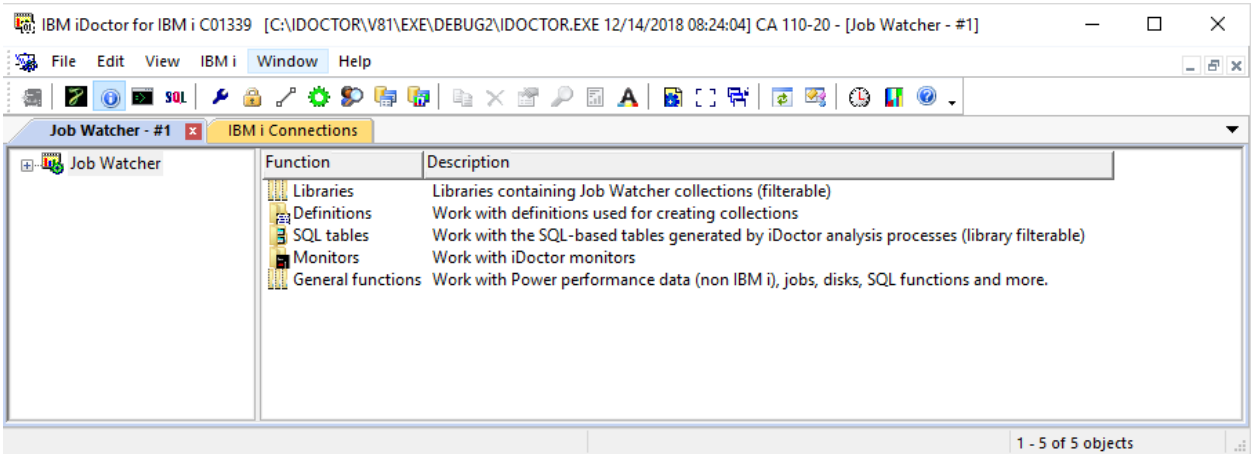


Main Window with the MDI Tabbed Style set to None

4.4.2 Standard

The Standard MDI style combines the benefits of using tabs with the classic MDI features of cascading and tiling. However, this style does not allow tabs to be grouped together, so it will be more difficult to make comparisons with another set of tabs when using this setting. You can also right-click anywhere on a tab to get additional options such as closing the tab or moving to another tab.

An example of the “Standard” MDI Tabbed Style in use is:



Main Window with the MDI Tabbed Style set to Standard

4.4.3 Grouped

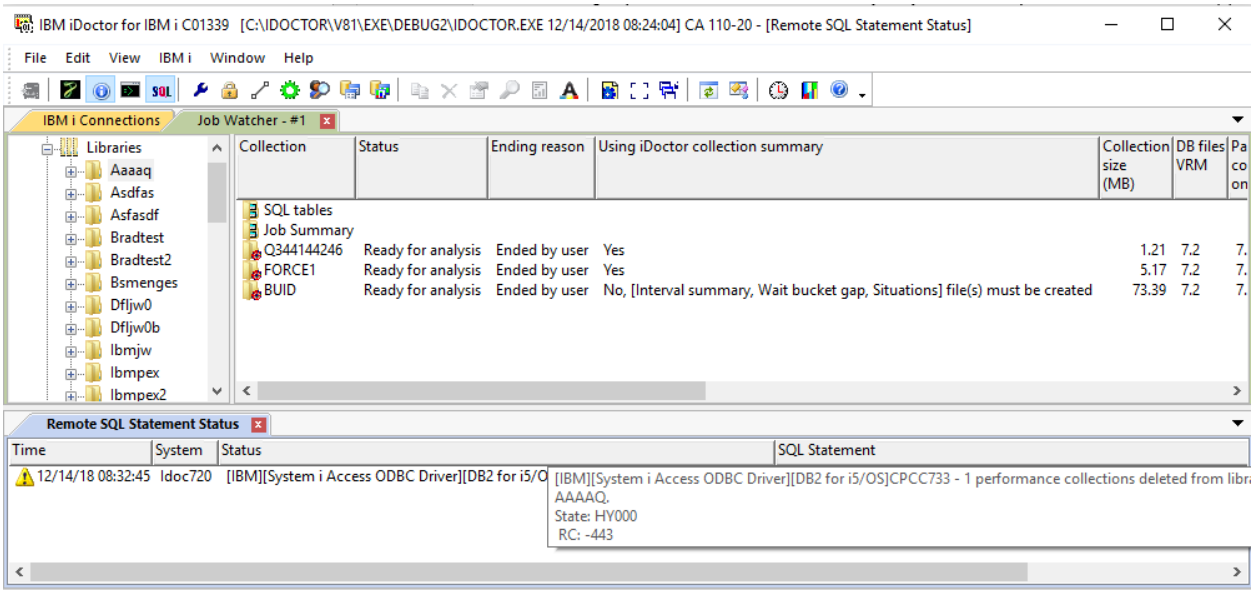
The Grouped MDI style allows users to drag and drop tabs/views order in order to create 1 or more groups of tabs. This makes it relatively easy to make comparisons of 1 or more view(s) vs another set of views.

To drag and drop a tab simply left click on a tab and hold the mouse down and move the mouse pointer to the desired part of the window to create a new tabbed group in (top, bottom, left or right.) Once in the correct location Windows will display a shaded rectangle around the area where the new tab will appear. At this point release the left mouse button and the tab will be moved to this location.

Grouped mode does not allow users to Tile or Cascade the views shown within.

Note: Within IBM, this mode is not available for users running IBM supplied Linux (RedHat) laptops running a Windows 7 KVM and also may not work/be available in other similar environments. When this mode is enabled it crashes the KVM.

An example of the “Grouped” MDI Tabbed Style in use is:

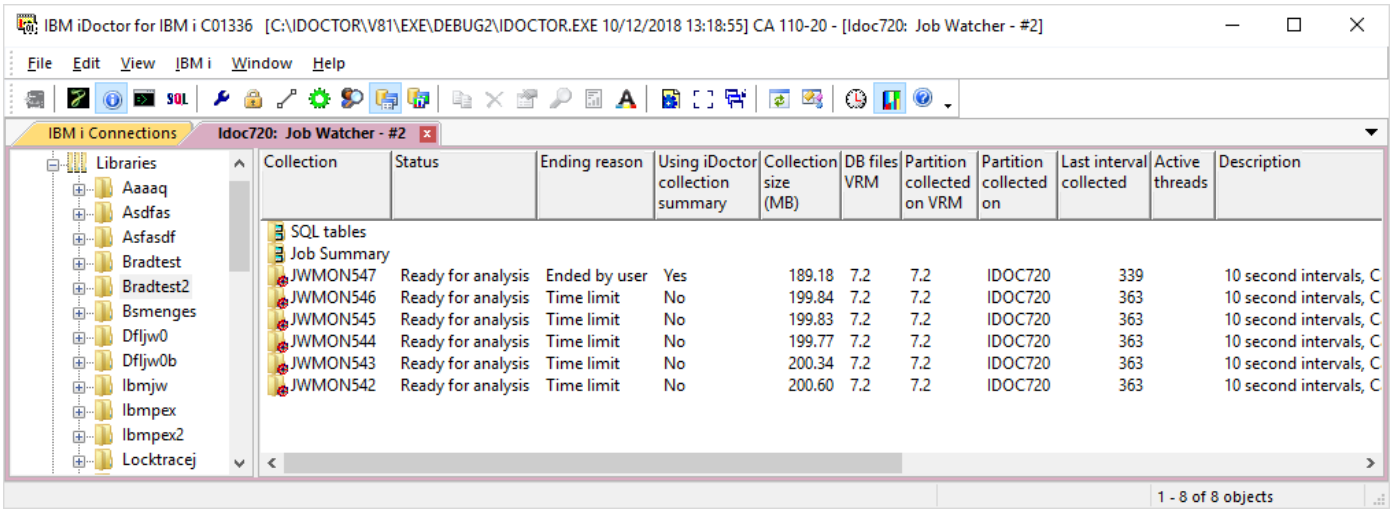


Main Window with the MDI Tabbed Style set to Grouped

4.5 The Main Window

The Main Window displays the various component views as well as some additional views such as the IBM i Connections View. Each of these views within the Main Window will be discussed in greater detail in the next sections. This section will discuss the general use of the Main Window within IBM iDoctor for IBM i.

The heart of the IBM iDoctor for IBM i application is the interface shown below. All components are displayed within this GUI each within a separate "tree/list" view. Each view can be resized, tiled or manipulated at runtime. You can have as many component (a.k.a "tree/list") views up at one time as you wish.



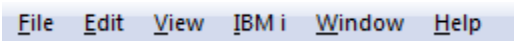
Main Window Example

4.5.1 Toolbar

	Add a connection to either the IBM i Connections View or Power Connections View depending upon which view is open and has focus. This option will only be enabled when one of the connection views has current focus.
	Shows or hides the Power Connections View.
	Shows or hides the IBM i Connections View.
	Shows or hides the Remote Command Status View. This view will show the status of CL commands running on behalf of the iDoctor GUI's requests.
	Shows or hides the Remote SQL Statement Status View. This view will show the status of SQL statements running on behalf of the iDoctor GUI's requests.
	Displays/hides the WRKOBJ Pane. This option lets you work with objects on the desired system.
	Displays/hides the WRKOBJLCK Pane. This option lets you work with the any locks that exist on the specified object.
	Displays/hides the WRKLNK Pane. This essentially lets you work with files / folders in the IFS under the specified location.
	Displays/hides the WRKACTJOB Pane. Unlike the command this view lets you work with active or inactive jobs on the system (to view spool files of completed jobs.)
	Displays/hides the WRKUSRPRF Pane. This lets you search/view/modify the user profiles found on the desired system.
	Displays/hides the TABLES Pane. This lets you search the desired system for physical files/SQL tables/view/etc.
	Displays/hides the Graph Search Pane. This lets you search the iDoctor report database for graphs and reports of interest and determine and what releases they are available and what special files or PTFs are required for each.
	Copy the selected rows or cells from a list view to the clipboard in text format. This allows you to quickly copy data shown in iDoctor list views into Notepad or other applications that work with text. **Note:** You can also press Ctrl+C from a list view to copy text from it.
	Deletes the selected objects (connections, libraries, collections, etc)
	Displays the property pages for the selected object (library, collection, etc)
	This option will open the Collection Search function for the currently selected performance collection. (PEX, CS, DW, JW, etc) This will let you search for something specific within the desired collection such as a job name.
	This option will open the Report Generator interface for the currently selected performance collection. This is used to build several graph/reports from the data at once and saving lists of reports into a reusable collection of favorites to be used repeatedly.
	Displays the Set Font window which allows you to control the font used in iDoctor tree and list views.
	Displays the iDoctor Preferences interface.
	Displays the current view in full screen mode. You can press escape to exit full screen mode.
	This button will display the Window Manager which lets you work with a list of all tabs/views that are opened. This lets you find and activate the desired view/window or close one or more views quickly.
	Refresh the selected list view or selected branch in a tree. **Note:** This is not the same as refreshing everything on the screen. In some cases, you may need to click on the folder above the current one to refresh the desired objects.

	Opens an empty Data Viewer for the system you are currently working with.
	Toggles the **default time range graph interval size**. The iDoctor default value for this is 1-minute intervals. This setting is changeable in the Data Viewer after the graph is opened using the clock icon there. The larger the interval size, the smaller number of bars produced in the graph and the more time that can be shown on a single graph page. This applies to all components that show data over time. **Note:** If the data was collected at an interval size greater than the current default time range interval size specified, (such as 15-minute intervals in Collection Services) the data will be shown at the collected interval size since it cannot be broken down further.
	This button enables or disables the Situational Analysis background colors across all graphs/components. If graphs are currently open, you may need to click on the desired graph after toggling this button on/off in order to see the change take effect on the graph.
	This option displays the properties for iDoctor. This button performs the same action as the Help -> About menu.
	This button is used to customize the toolbar. This lets you add or remove buttons, reset the toolbar to its original state and more. Changes are saved to the Windows registry and apply only to the current Windows user.

4.5.2 Menu Options



The tables below outline the different types of menu operations that may be performed within the Main Window of the iDoctor GUI.

File Menu	Description
Open iDoctor Session…	This option allows a user to open a previously saved iDoctor session file (*.idr.) When choosing this option you will be prompted for the file name to open and if continued all views in the Main Window and Data Viewers will be closed and then the previous session's state will be restored.
Save iDoctor Session As	Use this option to save the current state of all open views, tables and graphs to a session file.
Restore Previous iDoctor Session	This option can be used to open the last previously used iDoctor session.
Add Connection	This option will display the Add Connection window in order to add a new IBM i or Power connection depending on which view (IBM i connections or Power Connections) is currently active. If neither is active then this menu option will be disabled.
Open New Data Viewer	Opens an empty Data Viewer.
Close	This will close the active view within the Main Window.
Set User-Defined Reports Database	This option allows a user to load/use another user's iDoctor user-defined reports/graphs that they have previously created. When saving user-defined reports these are saved into the specified database. This can either be an IBM i library or a local database on the PC (MDB file). To find the current user-defined reports DB settings, either use this menu option or see the application properties (Help -> About menu) and then look for the "User-defined reports DB" location.
Migrate User-Defined Reports Database	This option is used to migrate the user-defined reports DB to the latest format. With latest GUI builds this is necessary in order to avoid problems with using user-defined reports.
Exit	Exits the application. All open windows including Data Viewers will be closed down.

Edit Menu	Description
Copy	Copies the current selection from the active view to the clipboard. This is only enabled when the active view is a list view or text in a textbox.
Delete	Deletes the current selection.
Select all	If the current view is a list view or text box, this option will select the entire contents.
Find	This option displays the Find Window which can be used to find the next or previous text in a list view.
Find Next	Use this option (or press F3) to look for and select the next occurrence of the text last entered on the Find Window.
Find Previous	Use this option (or press Shift+F3) to look for and select the previous occurrence of the text last entered on the Find Window.
Search	This option will open the Collection Search function for the currently selected performance collection. (PEX, CS, DW, JW, etc) This will let you search for something specific within the desired collection such as a job name.
Generate Reports	This option will open the Report Generator interface for the currently selected performance collection. This is used to build several graph/reports from the data at once and saving lists of reports into a reusable collection of favorites to be used repeatedly.
Launch Workload Estimator	This option will launch the IBM Workload Estimator web interface (tool) for the currently selected Collection Services collection.
Set Font	Displays a window allowing you to change the font used for the list views in the IBM iDoctor for IBM i application.
Preferences	Displays the preferences window letting you work with user preferences. These settings are stored in the PC's registry.
Wait Bucket Preferences	Displays a window letting you work with preferred colors, patterns and wait buckets to show in iDoctor wait bucket graphs.
Column Search	This option allows a user to search all of the iDoctor graph/report databases (.mdb files) at once for a specific column name in any of the reports. This function is mainly intended to be used by iDoctor development to consolidate duplicate columns across multiple databases into iDocUI.mdb.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Increase Windows GDI limit	On Windows 7, 8 and 10 this option will allow a user to open more graphs (4X) than normally you would be able to before running out of Windows (GDI) objects (a type of memory structure). After using this option, you will need to reboot your PC. On other releases of Windows this option may not do anything. As a work-around you can also try opening up multiple instances of iDoctor.
Restore Windows GDI limit	This option will restore the Windows GDI limit to the Windows default.

View Menu	Description
Customize	This option is used to customize the menu and toolbar options shown. You can control how they appear, remove/add options or reset them to their iDoctor-shipped defaults as desired.
Full Screen	This menu will display the Main Window maximized without a menu bar or toolbar in order to use all available space. You can press ESC or use the Close Window button to get out of this mode. **Note:** This option will be disabled if the MDI tabbed style is None.
MDI Tabbed Style	Use this option to change the current MDI tabbed style being used. There are 3 styles of MDI tabs available in iDoctor: 1) None – this is a classic Windows MDI without tabs 2) Standard – allows users to tile and cascade but you cannot create groups of MDI tabs to compare with other tabs. 3) Grouped – Tabs cannot be tiled or cascaded but you can create groups of MDI tabs in order to make comparisons. A Preferences option is also available which will open up the Preferences for the MDI tabs.
Application Look	Use this option to change the current application look setting used by iDoctor. The available options are various types of color schemes and Windows OS releases.
Status Bar	This menu will either show or hide the status bar. If the status bar is already visible there will be a checkmark next to the menu.
Refresh Selected	This menu will refresh the currently selected portion of a tree/list view. If a tree item is selected and this menu is clicked, everything underneath the tree item, including the tree item will have its data refreshed. If the list has focus and this menu is clicked, the entire list will be refreshed.
Power Connections View	Shows or hides the Power Connections View. If the view is already open there will be a checkmark next to the menu.
IBM i Connections View	This menu will either show or hide the IBM i Connections View. If the view is already open there will be a checkmark next to the menu.
Remote Command Status View	This menu will either show or hide the Remote Command Status View. If the view is already open there will be a checkmark next to the menu. This view displays the status of long running remote commands such as copying a collection or sending a collection to another system.
Remote SQL Statement Status View	This menu will either show or hide the Remote SQL Statement Status View. If the view is already open there will be a checkmark next to the menu. This view executes SQL statements used to run stored procedures or user-defined SQL statements from the SQL Editor using the Execute in batch option.
Work with Objects	Displays/hides the WRKOBJ Pane. This option lets you work with objects on the desired system.
Work with Object Locks	Displays/hides the WRKOBJLCK Pane. This option lets you work with the any locks that exist on the specified object.
Work with Object Links	Displays/hides the WRKLNK Pane. This essentially lets you work with files / folders in the IFS under the specified location.
Work with Active Jobs	Displays/hides the WRKACTJOB Pane. Unlike the command this view lets you work with active or inactive jobs on the system (in order to view spool files of completed jobs.)

IBM i Menu	Description
Work with Objects	Displays/hides the WRKOBJ Pane. It lets you browse/search the objects on an IBM i.
Work with Object Locks	Displays/hides the WRKOBJLCK Pane. This option lets you work with the any locks that exist on the specified object.
Work with Tables	Displays/hides the TABLES interface. This lets you search the desired system for physical files/SQL tables/view/etc.
Work with Object Links	Displays/hides the WRKLNK Pane. This essentially lets you work with files / folders in the IFS under the specified location.
Work with Jobs	Displays/hides the WRKACTJOB Pane. Unlike the command this view lets you work with active or inactive jobs on the system (in order to view spool files of completed jobs.)
Work with User Profiles	Displays/hides the WRKUSRPRF Pane. This lets you search/view/modify the user profiles found on the desired system.
Delete all Spool Files for current user	This option will prompt you and then if confirmed remove all spool files for the currently signed on user profile for the current system.

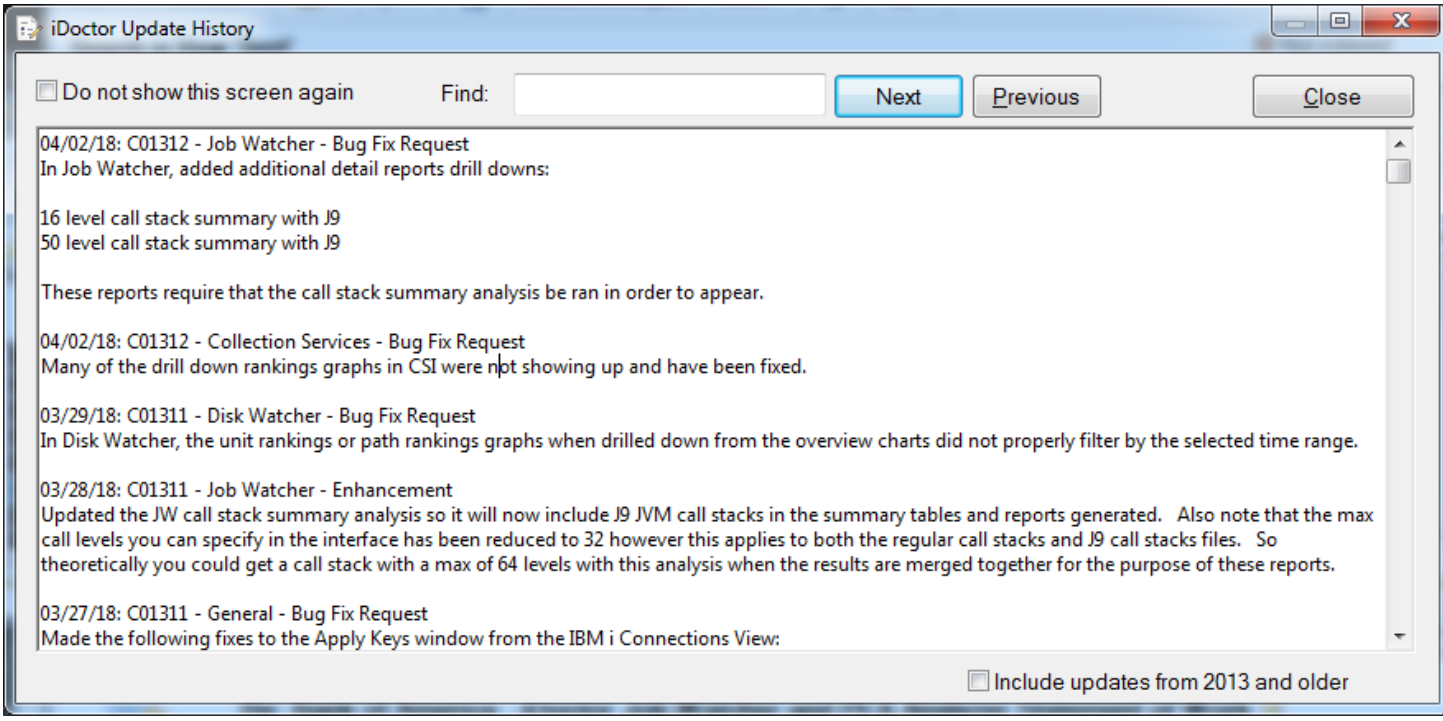
Window Menu	Description
Cascade	Use this menu to rearrange all views in the Main Window in an overlapping sequence starting in the upper left corner of the window. **Note:** This option is not shown when the MDI tabbed style is set to Grouped.
Tile Horizontally	Use this menu to rearrange all views in the Main Window such that each view will have an equal distribution of the available height in the Main Window. The views will not overlap each other. **Note:** This option is not shown when the MDI tabbed style is set to Grouped.
Tile Vertically	Use this menu to rearrange all views in the Main Window such that each view will have an equal distribution of the available width in the Main Window. The views will not overlap each other. **Note:** This option is not shown when the MDI tabbed style is set to Grouped.
Close All	This option will close all open tabs/views.

Note: The Window menu also dynamically contains a list of all open views within the Main Window for easy access to them.

Help Menu	Description
Update History	Shows the Update History window.
iDoctor Videos	Launches your web browser and takes you to iDoctor videos on IBM MediaCenter .
IBM i Support Center Videos	Launches your web browser and takes you to the IBM i Systems Support MediaCenter page
IBM iDoctor for IBM i website	Launches your web browser and takes you to the iDoctor website.
IBM iDoctor for IBM i downloads	Launches your web browser and takes you to the iDoctor download page.
IBM iDoctor for IBM i documentation	Launches your web browser and takes you to the documentation.
About IBM iDoctor for IBM i	This displays version information for the IBM iDoctor for IBM i client.

4.5.3 Update History

This window is displayed when iDoctor is started by default to show all the recent changes.



iDoctor Update History

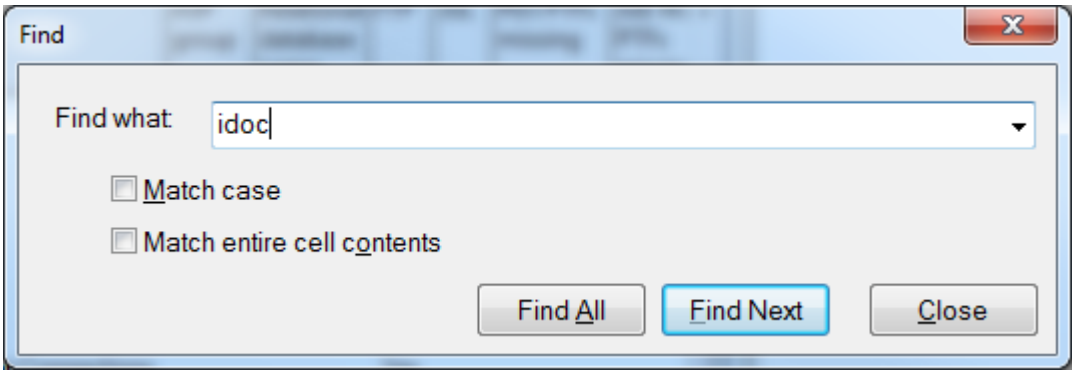
The window provides a find textbox that lets you search for the iDoctor function of interest. You may also click the do not show this again checkbox. If you click that checkbox and then want to see the Update History window again, just use the Help -> Update History menu.

Note: For more details on iDoctor updates also visit the iDoctor community.

4.5.4 Find Window

This window is shown whenever a user does a Find operation on an active list or table view.

Use the Edit -> Find menu (or Ctrl+F) to show this option.

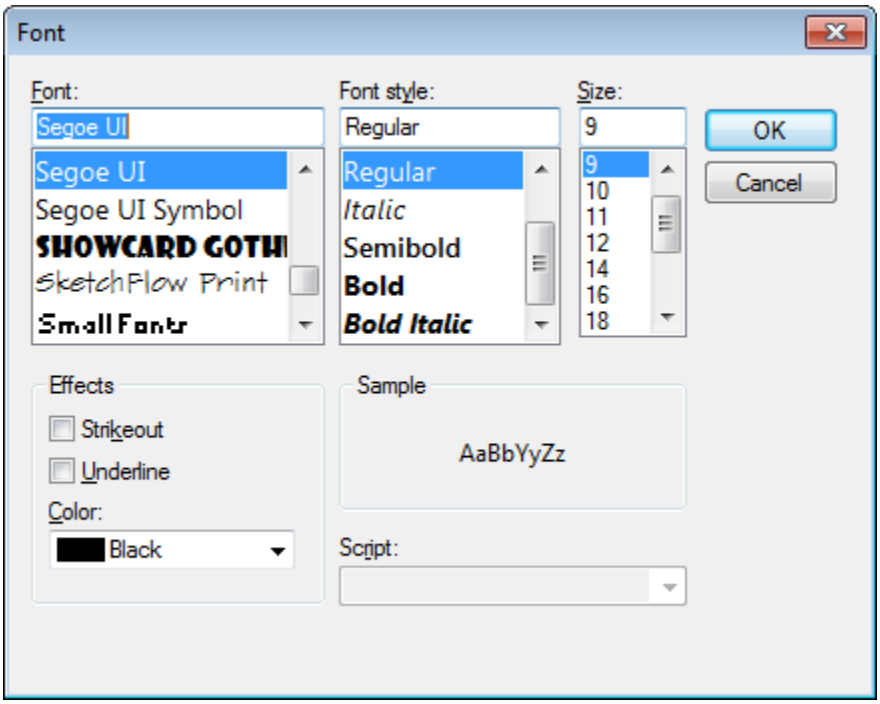


Find Window

After providing a search term to look for you can close the window and use the Edit -> Find Next (F3) or Edit -> Find Previous (Shift+F3) menu to look for the next/previous occurrences without needing to have this window visible.

4.5.5 Set Font

Another feature of iDoctor is the ability to customize the font used. The Set Font dialog provides the user with this flexibility. To change the font, use the Edit->Set Font menu from the Main Window or Data Viewer (or right-click on an active Table View and use the Set Font... menu). In addition to table views this font is used in all tree/list views and list views elsewhere in the application.



Font window

4.5.6 Wait Bucket Preferences

The Wait Bucket Preferences window allows a user to work with desired colors and patterns to use when graphing the wait buckets in iDoctor. The Wait Bucket Preferences are accessible via the Edit -> Wait Bucket Preferences menu from the iDoctor Main Window.

Note: Any changes made to this interface will not immediately take effect on already open graphs. You must first refresh the list of collections within the desired collection library and then open the desired graph to see any changes made to the Wait Bucket Preferences.

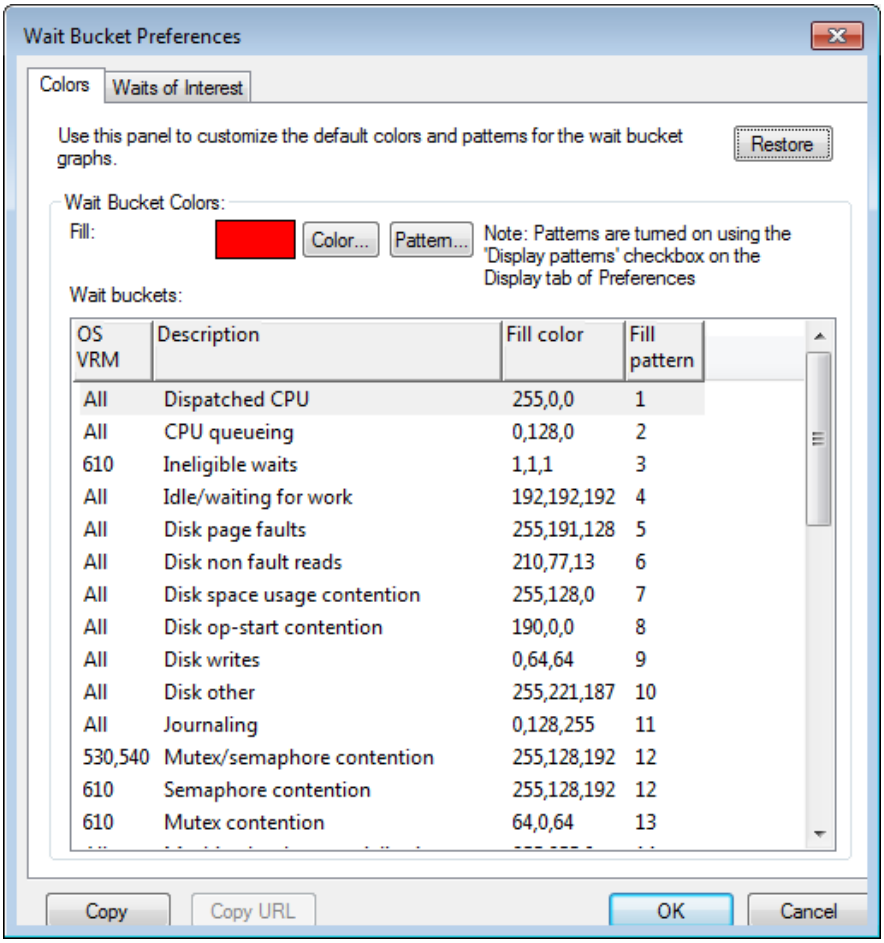
See the next sections for information on each page in this interface.

4.5.6.1 Colors

The Colors page allows the user to change the default colors for any desired wait bucket. In some cases, the wait bucket description only applies to a specific OS VRM and this VRM is listed in the 1st column.

This panel also allows a user to specify the pattern to use if the Display patterns preference is enabled.

An example of this interface is shown below:



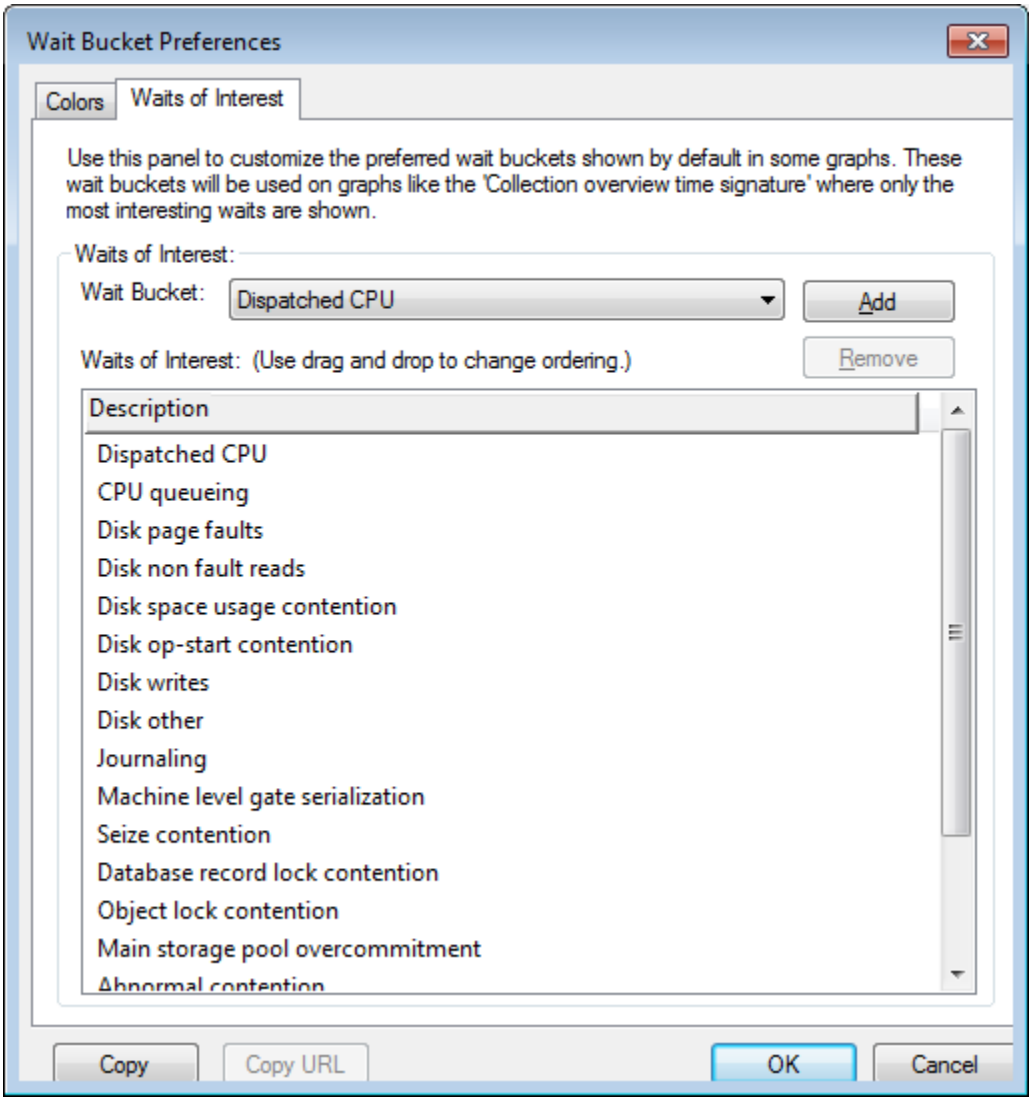
The options available on this page are summarized below:

Options	Description
Color… button	Allows you to modify the color for the selected wait bucket.
Pattern… button	Allows you to modify the pattern for the selected wait bucket.
Restore button	This button will discard any changes made to the wait bucket colors, patterns and waits of interest. The IBM-supplied defaults will be used.
List of buckets	List of wait buckets and the VRM, color and pattern that currently applies to it.

4.5.6.2 Waits of Interest

The Waits of Interest page lets you pick which wait buckets to display on the wait bucket graphs in iDoctor. It's important to only add wait buckets that will be helpful in solving performance problems. Therefore, it is unwise to add wait buckets to the list where jobs are frequently spending most of their time idle.

An example of this interface is shown below:



The options available on this page are summarized below:

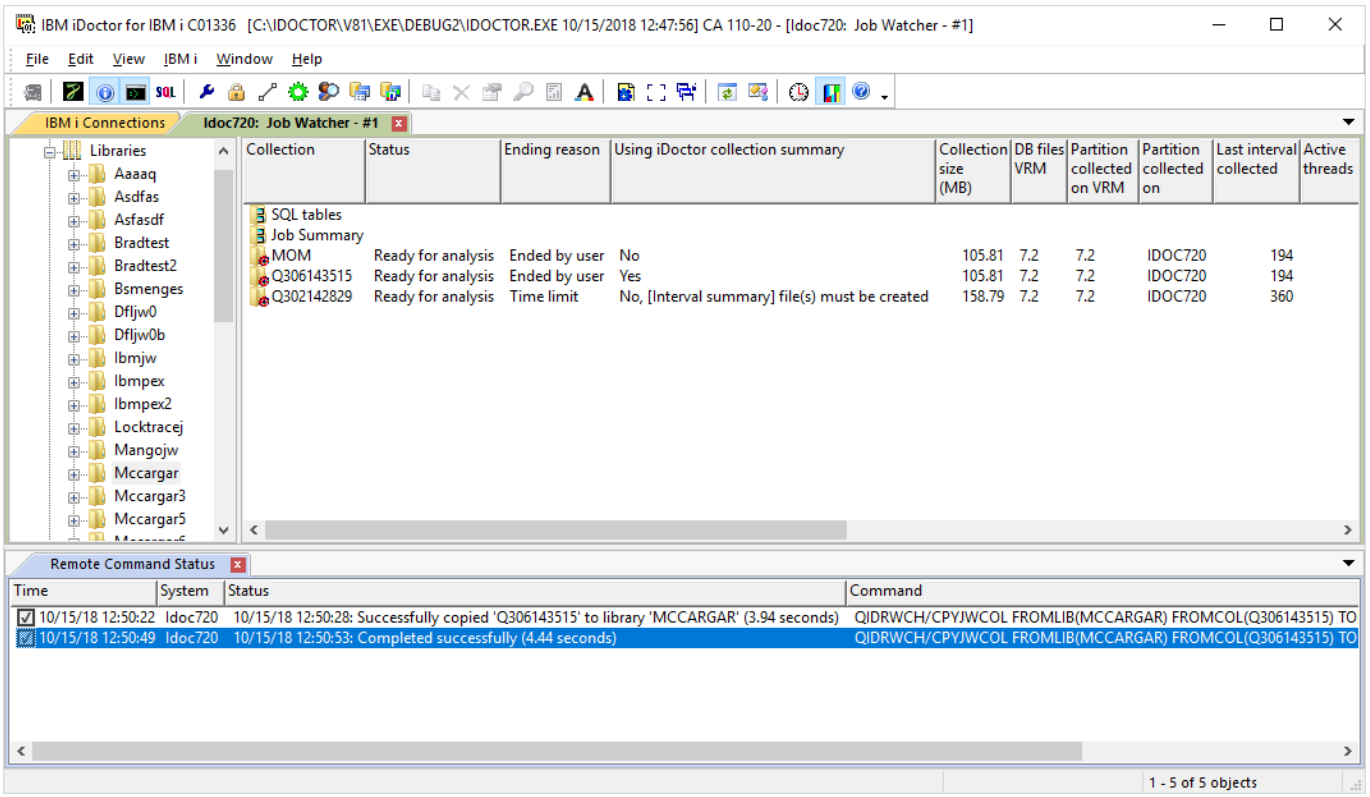
Options	Description
Wait bucket drop down list	List of possible wait buckets to add to the list below.
Add button	Adds the selected wait bucket from the drop down list to the list (if not already in the list)
Remove button	Removes the selected buckets from the list.
Waits of Interest list	List of Waits of Interest. These are the default wait buckets to display on iDoctor graphs. **Note:** Keep in mind that not all of these waits will be displayed in the graph's legend, because the graph legend (in the Y axis) only contains buckets that experienced values greater than zero.

4.5.7 Remote Command Status View

The Remote Command Status view shows you the status of certain remote commands being executed on a system. This allows you to perform lengthy operations like copy objects or delete files or data collections without tying up the GUI.

Depending on the function being used you will see one or more commands in the remote command status view. As each command completes you will immediately see its result or error message in the view.

You can also close this window and reopen it later while commands are being executed to periodically check the status of the commands issued. Use the View -> Remote Command Status menu on the Main Window to reopen it.



Remote Command Status View (within the Main Window) displaying the status of collections being copied in Job Watcher.

The following actions may be taken in the Remote Command Status View by selecting one or more entries and then right-clicking:

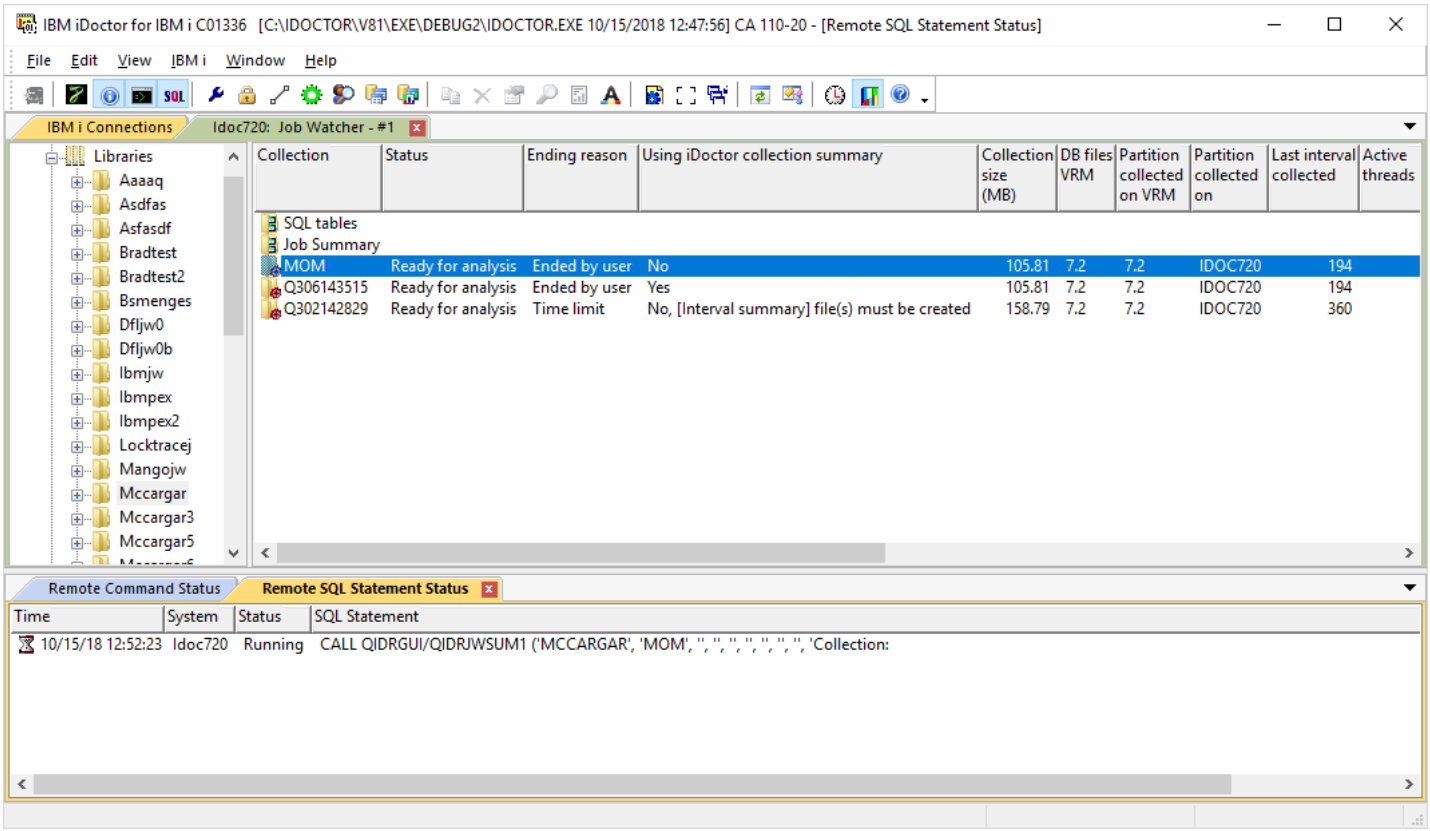
Popup Menu	Description
Show Job Log	Shows the job log for the selected system. Each system uses a different QZRCSRVS job to process the commands executed.
View Job in WRKACTJOB View	This option is used to open either the current QZRCSRVS job behind this view or if the command is a SBMJOB command this will allow you to see the submitted job in the WRKACTJOB interface
Copy Selected Commands to Clipboard	Copies all command strings listed in the entries selected to the Windows Clipboard.
Rerun commands	Allows you to rerun the selected command(s) either on the same LPAR or other LPAR(s).
Add Command	Displays a window where you can provide your own CL command(s) to run on this system (and/or other systems.)
Remove/Cancel Selected	Use this menu to remove all selected remote command entries from the view. If actively running commands are selected the QZRCSRVS job will end (and be recreated if needed) in order to cancel the command.
Remove/Cancel All	Use this menu to remove all remote command entries from the view. If actively running commands are selected the QZRCSRVS job will end (and be recreated if needed) in order to cancel the command.

4.5.8 Remote SQL Statement Status View

The Remote SQL Statement Status view shows you the status of SQL statements (usually stored procedure calls) that are running on the system.

Depending on the function being used you will see one or more statements in this view. As each statement completes you will immediately see its result or error message in the view.

You can also close this window and reopen it later while commands are being executed to periodically check the status of the statements issued. Use the View -> Remote SQL Statement Status View menu on the Main Window to reopen it.



Remote SQL Statement Status View (within the Main Window) displaying the status of collections being deleted in Job Watcher.

The following actions may be taken in the Remote SQL Statement Status View by selecting one or more entries and then right-clicking:

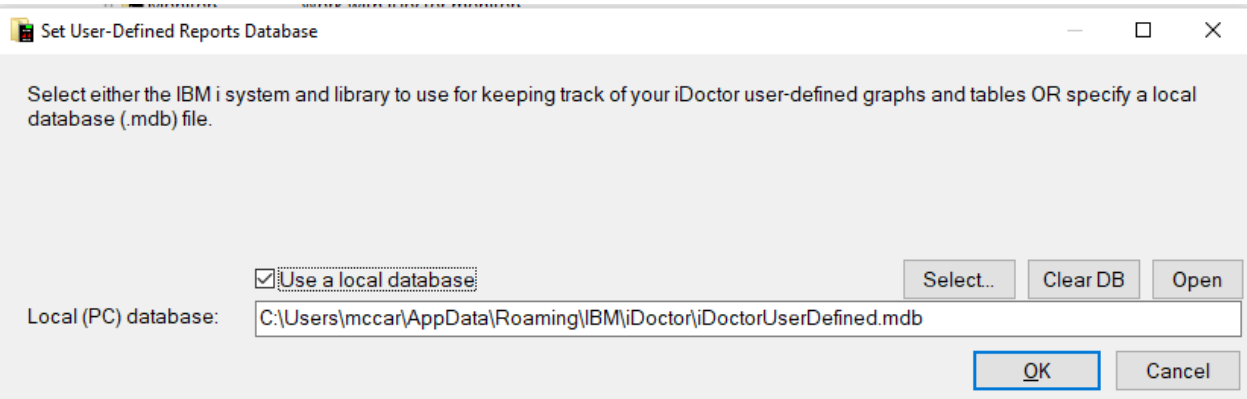
Popup Menu	**Description**
Show Job Log	Shows the job log for the selected system. Each system uses a different QZDASOINIT job to process the statements executed.
Copy Selected SQL Statements to Clipboard	Copies all SQL Statements for the entries selected to the Windows Clipboard.
Add SQL Statement	Displays a window where you can provide your own SQL Statements(s) to run on this system (and/or other systems.) Multiple statements may be provided at once. Use a semicolon at the end of each statement.
Remove/Cancel Selected	Use this menu to remove all selected SQL Statements from the view. If actively running SQL statements are selected the SQLCancel API will be issued to cancel the request.
Remove/Cancel All	Use this menu to remove all SQL Statements from the view. If actively running SQL statements are selected the SQLCancel API will be issued to cancel the request.

4.5.9 Set User-Defined Reports Database

Note: You can access this option either from the Main Window menu **File -> Set User-Defined Reports Database**, or from any IBM i component view right-click the component (root) icon and use the **Set User-Defined Reports Database** menu**.**

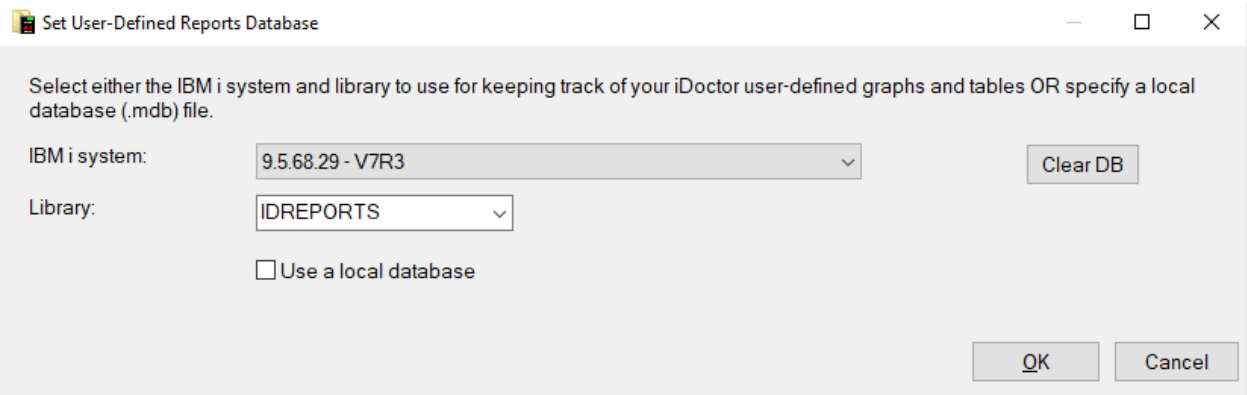
This window allows the user to specify the database where iDoctor user-defined graphs and tables are accessed and stored. You can use this option to access another user's iDoctor reports they have defined against their data or your data.

The database used can be either an IBM i library or an MS Access database file residing on the PC or a network drive. If using the IBM i library option, you will need to be on iDoctor server builds from Nov 2017 and newer.



Set User-Defined Reports Database window with a local database selected

GUI element	Description
Local PC database	This identifies the location of the MS access mdb file on the PC to use as the user-defined reports database.
Select…	This button will open a prompt to allow you to change the current user-defined reports database.
Clear DB	This option will delete all data in the iDoctor user-defined reports database. **Note:** No performance data is removed, just the definitions that define the user-defined reports.
Open	This option will open in Microsoft Access the current database (if Access is installed on the PC.)
Use a local database	This box must be checked to specify a MS access database to use instead of an IBM i library.



Set User-Defined Reports Database with an IBM I library selected

GUI element	Description
IBM i system	This identifies the system to use for the iDoctor user-defined reports database.
Library	This identifies the library on the IBM I system to use for the iDoctor user-defined reports database.
Clear DB	This option will delete all data in the iDoctor user-defined reports database. Note: No performance data is removed, just the definitions that define the user-defined reports.
Use a local database	This box must be checked to specify a MS access database to use instead of an IBM i library.

The easiest way to share your reports with other users is to use the same library on a shared IBM i. If all users configure their Set User-Defined Reports database, then those users will be able to easily share the same reports that they create with iDoctor.

When the window is first opened the current setting for the user-defined reports DB will be shown.

If using an IBM i library as the database, these are the tables that will exist on the library making up the database.

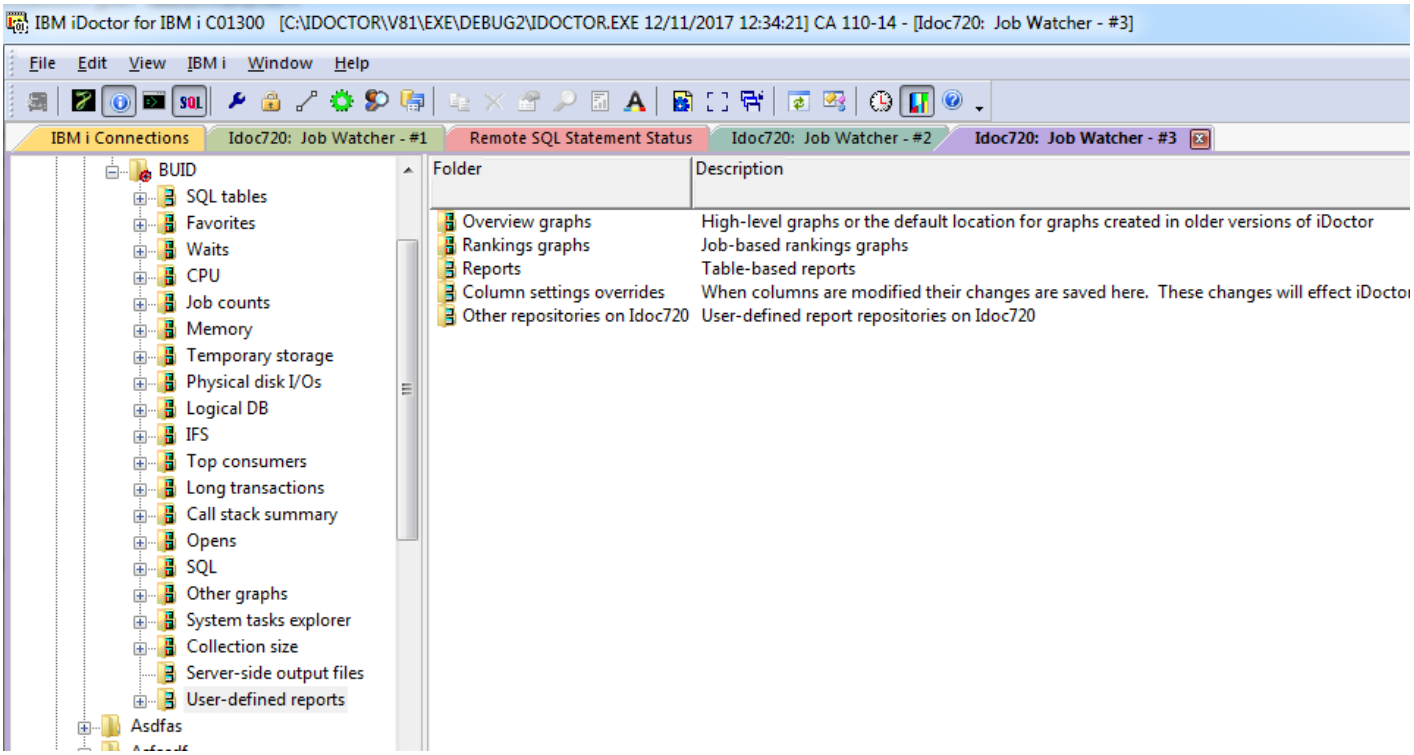
COLUM00001 (this is really the COLUMNDESCS SQL table which is typically named COLUM0001)
QAIDRCATS - this defines the folders
QAIDRGPH - graph definitions
QAIDRRGEN - report generator lists
QAIDRSQL - SQL statements (for graphs and/or reports)

You can move these tables to other libraries/systems to backup or reuse them as needed.

4.5.10 User-Defined Reports folder

When viewing collections in most iDoctor components an option called “User-defined reports” will be shown. This option allows you to show reports from your user-defined reports database but applied to the current collection. This folder contains both graph and table views.

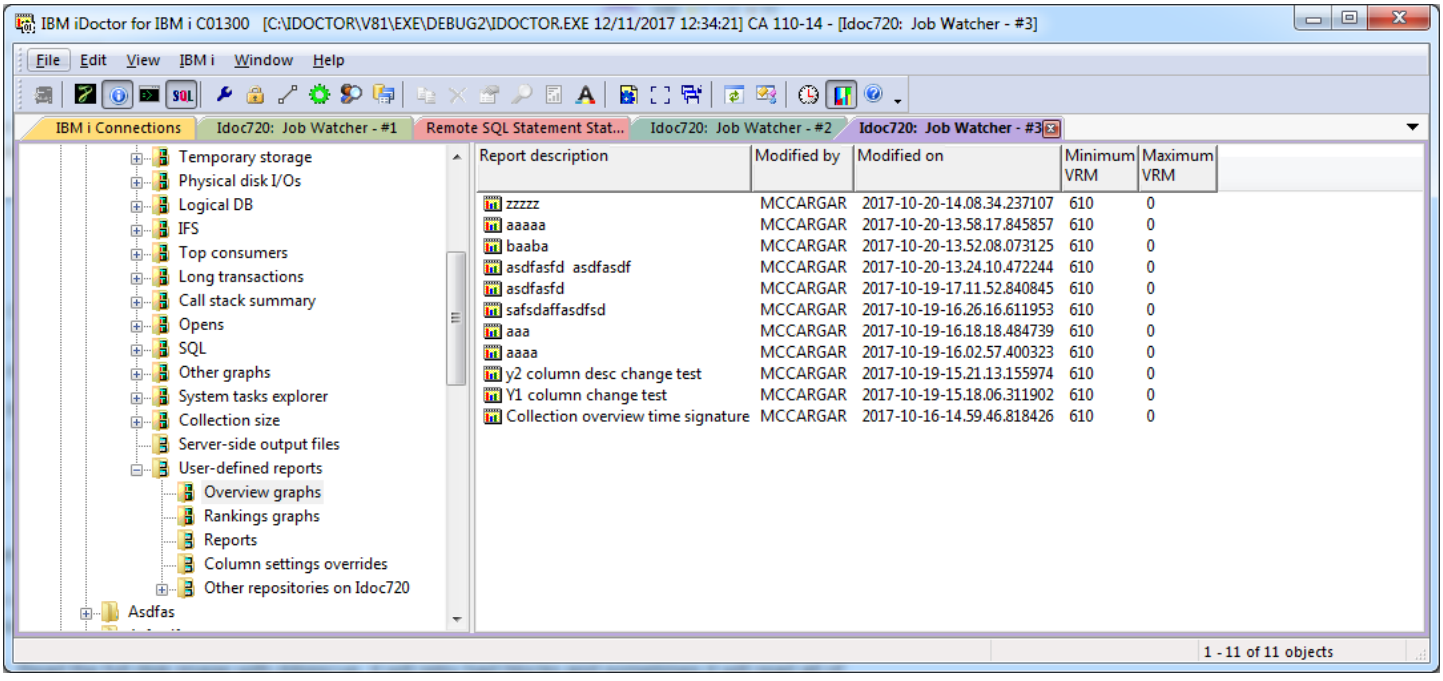
Typically, users will see something like the following:



User-defined reports folder for a Job Watcher collection

4.5.10.1 Overview graphs

This folder typically contains user-defined graphs that are time based. If you had created user-defined graphs in old versions of iDoctor they will be migrated here by default.



A collection's User-defined reports -> Overview graphs folder

Double-click a graph to open it. You may also right-click the graph and use the Properties option to view and change some of the settings for the graph such as the SQL statement used.

4.5.10.2 Rankings graphs

Use this option to save user-defined graphs that rank data (such as jobs or disks.)

4.5.10.3 Reports

Use this option to store any table views. If you had created user-defined tables in old versions of iDoctor they will be migrated here by default.

4.5.10.4 Column settings overrides

This folder contains each column that has been modified in iDoctor by any user using the currently defined user-defined reports database. Whenever a user modifies a graph or table column using the Edit column interface they will be saved here. These overrides apply to iDoctor supplied reports and user-defined reports for any where the column short name in the SQL statement matches the values in this list.

If you delete a column from this view, then the iDoctor-defined settings for this column (colors and description) are used instead (assuming the same column name exists in the iDoctor reports.)

4.5.10.5 Other repositories

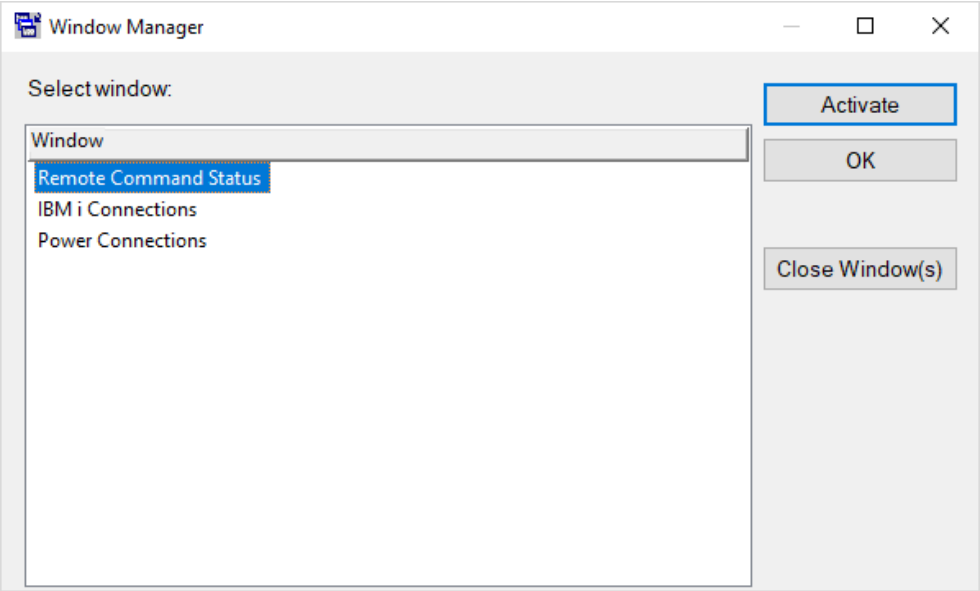
This option allows you to work with any other repositories (libraries) found on the current IBM i you are using. Only libraries will appear that contain the iDoctor repository SQL tables such as QAIDRGPH or QAIDRSQL.

Note: Additional folders can appear based on what the user has defined in the database. When saving a user-defined graph or table you can specify the folder the report appears under and you can create a new sub folder to store it into as well.

4.5.11 Window Manager

The Window Manager is an option found by pressing the  button on the Main Window or Data Viewer toolbars. The window allows a user to see a list of all windows (or views) open and to pick the desired one to activate.

Tip: You can double-click a window in the list to activate it.



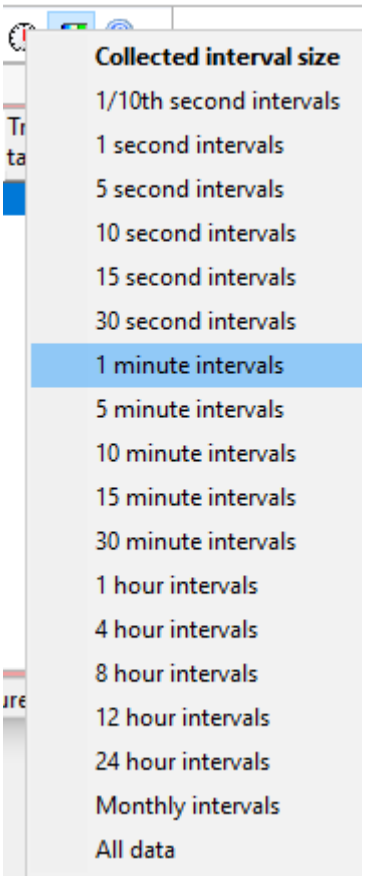
Window Manager

Option	Description
Activate	This will activate the selected window from the Select window list. You can also double-click a window to activate it.
OK	Closes the Window Manager window.
Close Window(s)	Closes all views/windows selected in the Select window list.

4.5.12 Time interval size (clock icon)

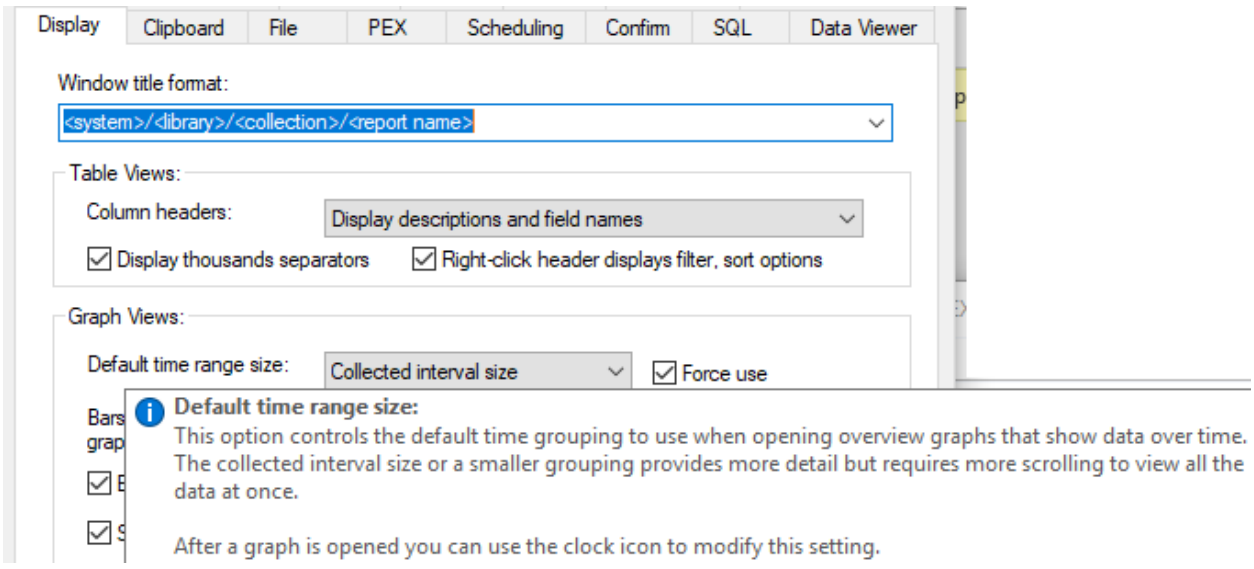
The clock icon on the Main Window toolbar allows a user to modify the default time range grouping on all iDoctor time-based graphs. Many groupings are available and make it easier to get the big picture view of thousands of records of data more quickly rather than scrolling through it or trying to graph thousands of points on a single screen (often not feasible!)

To enable this option, press this  button and a list of options will be shown. Picking the desired grouping will change the current default value for the time interval size. The current default setting will be shown in bold font.



Clock icon menu

Tip: This preference can also be set using the **Preferences -> Display -> Default time range size** setting.



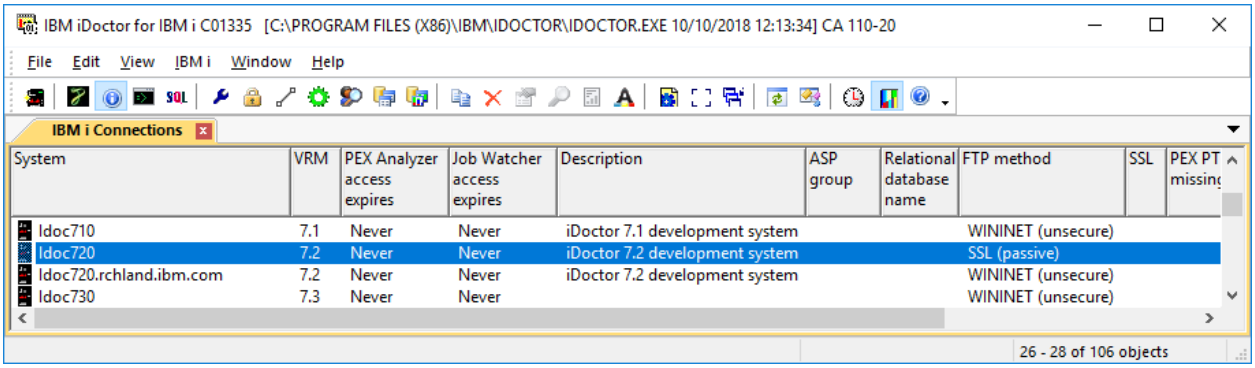
Preferences -> Display -> Default time range size

4.6 IBM i Connections View

The IBM i Connections view allows you to work with all the connections defined to IBM i systems created via IBM i Navigator or iDoctor. You can easily add or remove connections to other systems through this view. The primary purpose of this view is to provide a quick and easy way to launch the iDoctor components for any system desired.

The list of connections shown is for the currently active environment (as defined in IBM i Navigator). You can change the currently active connection environment by right-clicking on the list and choosing the Change Environment... menu.

Note: You cannot create new IBM i connection environments with iDoctor. You must use IBM i Navigator to do that.



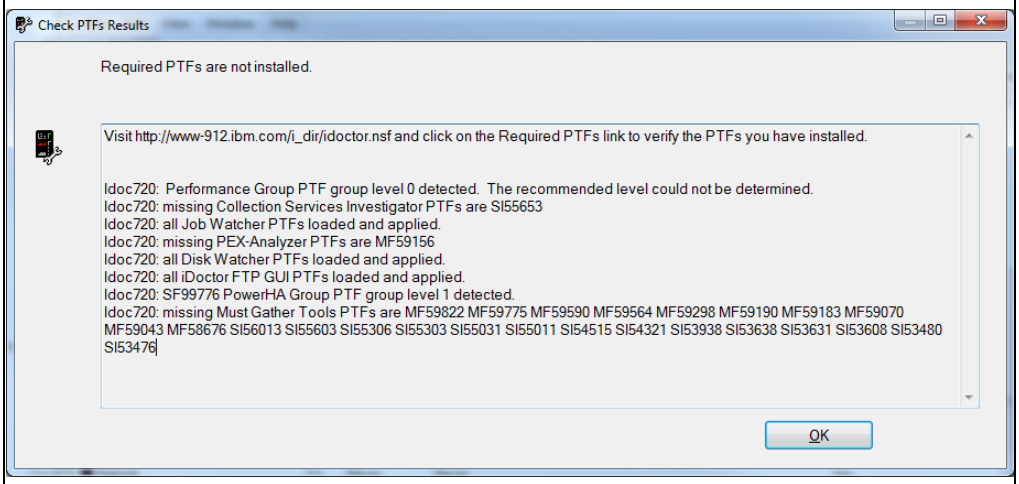
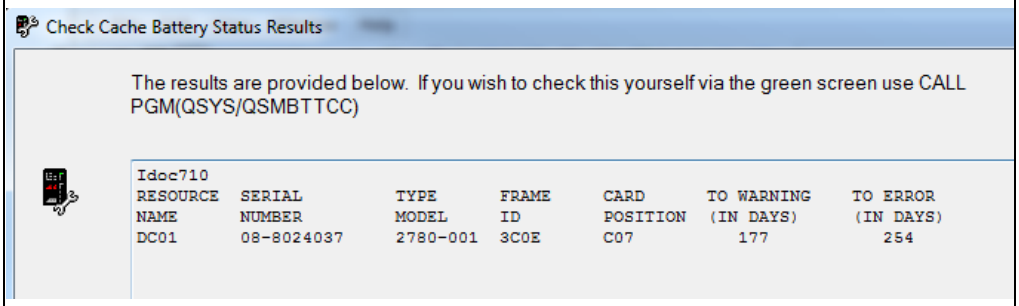
Main Window displaying the IBM i Connections View

The list contains several columns. **NOTE:** All values shown are based on the last known connection made to the system and may not reflect current system settings.

Column	Description
System	System/partition name or IP address.
VRM	The version/release of the IBM i.
PEX Analyzer access expires	Last day when PEX Analyzer will be available (unless a new access code is supplied by IBM.)
Job Watcher access expires	Last day when Job Watcher and its subcomponents will be available (unless a new access code is supplied by IBM.)
Description	An optional description given to the connection within iDoctor.
ASP group	The name of the IASP group to use when making the connection. This is a required field if you wish to work with libraries created on IASPs. The value given should match the value supplied after running the command: WRKDEVD DEVD(*ASP)
Relational database name	The name of the relational database to use when making the connection to the system. This is primarily used when connecting to a system with an IASP (in the ODBC connection for the QZDASOINIT job). The value may or may not be same as the ASP group value depending on how the system is configured. The relational database name is listed in the Database component of IBM i Navigator.
FTP method	This indicates the preferred FTP method to use for each connection.
SSL	This setting indicates if SSL connections should be made for ODBC and IBM i Access remote command / program calls.
PEX PTFs missing	The list of required PEX PTFs not yet installed. This value is updated only when the PTFs are checked by the GUI.
Job Watcher PTFs missing	The list of required Job Watcher PTFs not yet installed. This value is updated only when the PTFs are checked by the GUI.
Disk Watcher PTFs missing	The list of required Disk Watcher PTFs not yet installed. This value is updated only when the PTFs are checked by the GUI.
IP Address	The last known IP address of the system. To update this right-click the list and use the Check -> IP Address popup menu.

The following menu options are available in the IBM i Connections View:

Popup Menu	Description
Connect	Connects to the selected system and displays the iDoctor components window with the status of each component installed on the system. You will be prompted to signon if a connection has not yet been established to the system.
Terminal Sessions – Launch PCOMM	If IBM Personal Communications (PCOMM) is installed and the preferences on the IBM i Access page are configured correctly, then this option will launch a PCOMM session for the selected system. The following are required in order for this function to work correctly. 1) IBM Personal Communications must be installed. 2) In the IBM i Access Preferences, the PCOMM directory must point to where IBM Personal Communications is installed, the sessions directory must point to where your (*.ws) session files for each system are located. 3) The system name listed in iDoctor must match the system name used for the .ws file name. **Note:** This option will not work with IBM Personal Communications 6.0.11 (6.0 MR11) and presumably earlier releases. The green screen session would appear to hang due to a bug with IBM Personal Communications when it is being launched from another program. To resolve this there are 2 options either: 1) Use the Help -> Detect and Repair... menu within a PCOMM session and then reboot the PC. (recommended) 2) Within the PC's BIOS under CPU disable Intel Hyper Threading Technology. (not recommended)
Terminal Sessions – Launch Putty (SSH)	If Putty is installed and preferences on the Power page are configured correctly to allow iDoctor to know its location, then this option will attempt to launch Putty using an SSH connection to the selected system. **Note:** You may need to run this command on the IBM I before being able to use this option by starting the SSHD server. QSYS/STRTCPSVR SERVER(*SSHD)
Terminal Sessions – Launch Putty (Telnet)	If Putty is installed and preferences on the Power page are configured correctly to allow iDoctor to know its location, then this option will attempt to launch Putty using a telnet connection to the selected system. **Note:** You may need to run this command on the IBM I before being able to use this option by starting the Telnet server. QSYS/STRTCPSVR SERVER(*TELNET)
Terminal Sessions – Start SSH Server	This option will run command QSYS/STRTCPSVR SERVER(*SSHD) on the desired system to start the SSHD server.
Terminal Sessions – End SSH Server	This option will run command QSYS/ENDTCPSVR SERVER(*SSHD) on the desired system to end the SSHD server.
Terminal Sessions – End all PCOMM sessions	This option will kill all open instances of IBM Personal Communications regardless of whether they were started by iDoctor or not.
Terminal Sessions – End all Putty sessions	This option will kill all open instances of Putty regardless of whether they were started by iDoctor or not.
Check - expiration dates	This option checks the access code expiration dates of all systems selected and updates the applicable columns in the list of connections.
Check – *PTFs*	This option checks the required PTF levels for all iDoctor components on the desired system(s). The Performance Group PTF level will be checked as well as the required PTFs for Job Watcher, PEX, Collection Services and Disk Watcher. Upon selecting this option a window is shown that describes any PTFs missing for each system by component:

	 **Note:** PTFs for Power HA are no longer checked by this option.
Check – PTF Search	This option allows you to search for one or more PTFs on the system and display their status.
Check – Cache Battery Status	This option will check the cache battery status for all applicable resources on all selected systems. The results of this check will look something like this:  If you wish to perform this check manually then run the following command: CALL PGM(QSYS/QSMBTTCC)
Check – Cache Battery Status (verbose)	This option is the same as the previous one but more details are provided in the output.
Check – FTP Connections	This option will check each of the types of FTP Connections available with iDoctor to the selected system and indicate which methods work and which do not.
Check – FTP Connections (verbose)	This is the same as the previous option but also provides additional debug information.
Check – Connection (show ping results)	This option will ping the selected system and displays the results.
Check – IP Address	This option will ping the selected system in order to determine the current IP address.
Commands	These options will open the desired pane for the selected IBM i and command picked from the list.
Set Default Signon	This option will set the default user name and password for making all types of iDoctor connections. This user name and password will be used on any connections where the “set specific user ID” option has not been set yet.
Add Connection	Use this menu to add an IBM i connection to the IBM i Connections View.
Apply Keys…	The Apply Keys Window allows you to apply access codes to one or more selected LPARs using the email containing the access codes from iDoctor support.

Delete	This will delete the selected connection(s) from the IBM i Connections View.
Edit	This option allows you to modify the selected connection's settings.
Change Password…	Use this option to change the password on the selected system.
Clear password cache	iDoctor will store encrypted passwords on the PC if the save password checkbox is left checked when signing on to any system. This option will clear this cache for either all systems or all selected systems depending on the submenu selected.
Export connections	Use this option to create a Windows registry file that contains a list of all your IBM I connections. This file can be used to restore all your connections at a later time or to another system.
Uninstall iDoctor	This option removes all iDoctor server libraries and objects. After running this option, you can view the results (job log) from the Remote Command Status View. If you wish to run this process outside of the GUI then execute the following commands: CRTDUPOBJ OBJ(QIDRUNINST) FROMLIB(QIDRGUI) OBJTYPE(*PGM) TOLIB(QTEMP) CALL PGM(QTEMP/QIDRUNINST) **Note:** No performance data created by iDoctor is deleted using this option. If this is desired, clear the desired performance data from the system first before uninstalling iDoctor.
Select all	Selects all connections in the list.
Change Environment...	This menu lets you change the currently active environment as desired. Each environment represents a list of connections. The environments are created using IBM i Navigator.

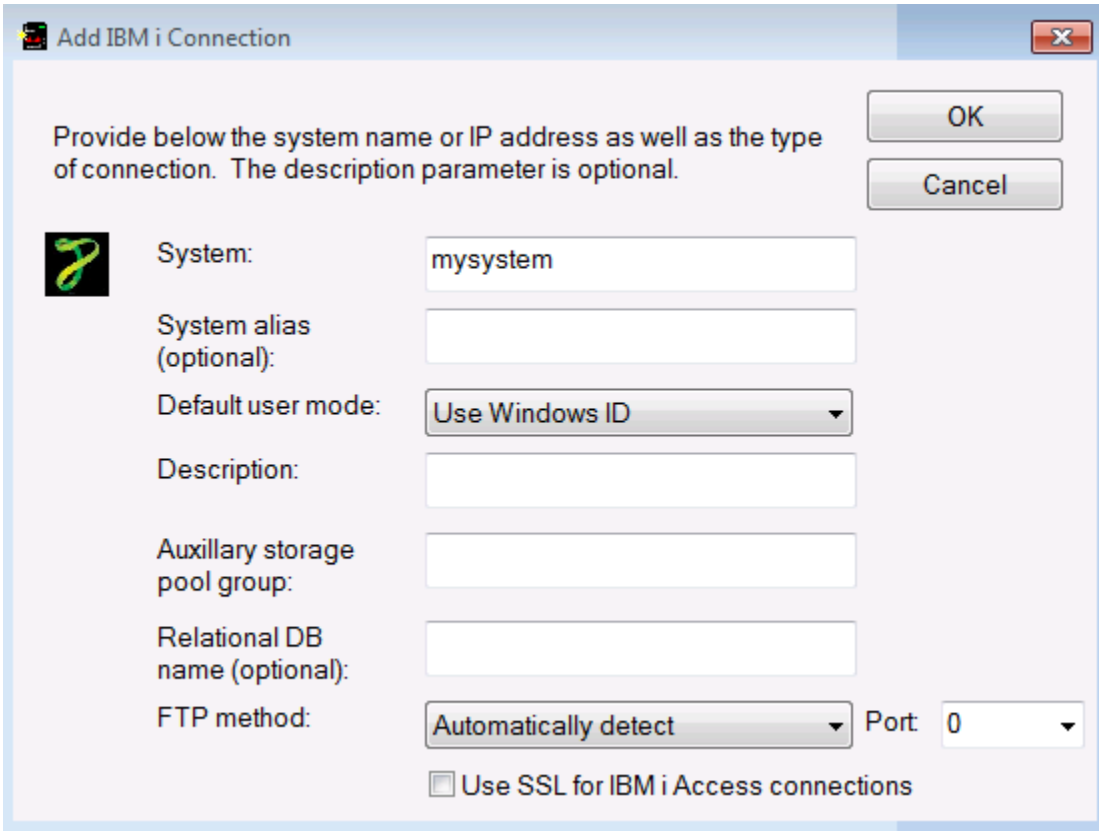
4.6.1 Add/Edit IBM i Connection

This window allows a user to add (or edit) a connection to the list.

Simply provide the system name or IP address, the default user mode an optional description and click OK to register the system on your PC and add it to the list.

The auxiliary storage pool group should normally be left blank but if you have IASPs and want to look at data stored in libraries created using an IASP, then you can specify the ASP group name and Relational database name to see them.

An example of this interface is:

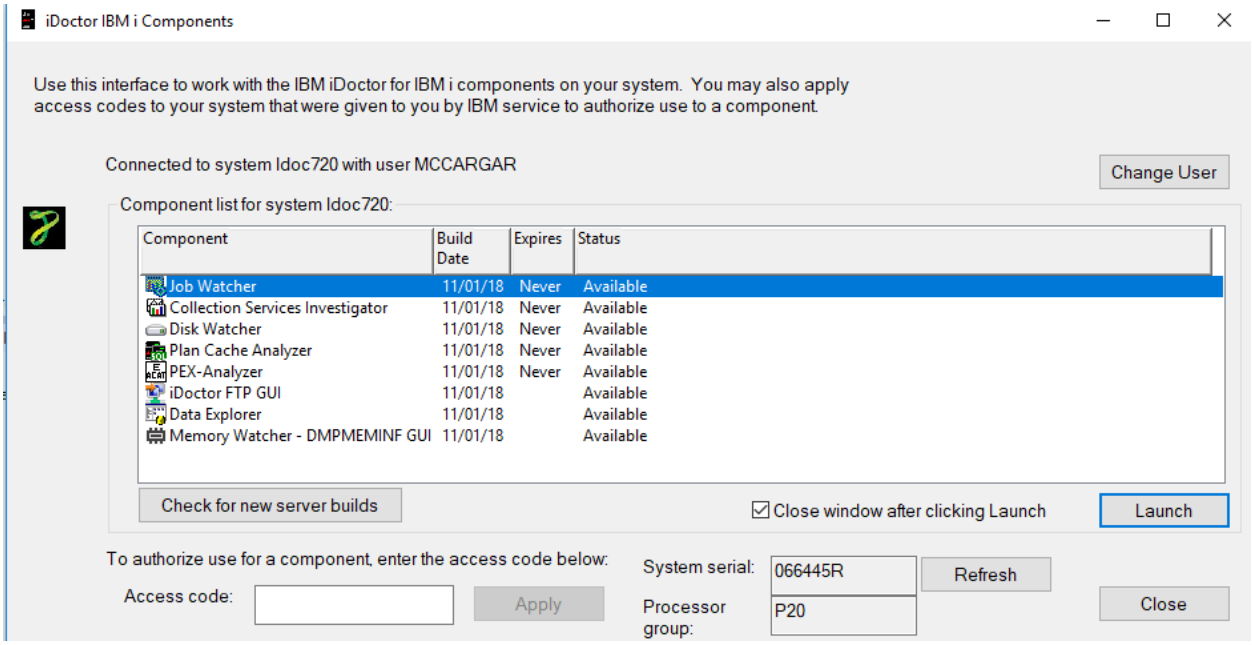


The options available on this screen are described in the following table:

Option	Description
System	System/partition name or IP address.
System alias (optional)	This option (if specified) will display this value as the system name in graph titles or report titles in the Data Viewer.
Default user mode	Indicates how the value for the user name to use when making connections to this system will be determined. The options are: Use Windows ID Set specific user ID (a text box will appear where you can enter this value) Prompt every time
Description	An optional description given to the connection.
Auxillary storage pool group	The name of the IASP group to use when making the connection. This is a required field if you wish to work with libraries created on IASPs. The value given should match the value supplied after running the command: WRKDEVD DEVD(*ASP)
Relational DB name	The name of the relational database to use when making the connection to the system. This is primarily used when connecting to a system with an IASP (in the ODBC connection for the QZDASOINIT job). The value may or may not be same as the ASP group value depending on how the system is configured. The relational database name is listed in the Database component of IBM i Navigator.
FTP method	This option lets you pick which FTP method is preferred when connecting to this system. “Automatically detect” is the default option but several options are available. The Port value defaults to 0 but this just means the default port for the type of connection will be used. WININET is 21, SSL (passive) is 990, etc. Specific a specific method and port if desired.
Use SSL for IBM i Access connections	This setting indicates if SSL connections should be made for ODBC and IBM i Access remote command / program calls.

4.6.2 iDoctor IBM i Components Window

The components window provides the status of the iDoctor components installed on the system selected from the IBM i Connections View.



iDoctor IBM i Components Window

This window allows a user to launch a component, change the user signed on to the system or apply an access code. After applying an access code the component list will refresh to indicate any changes in status (i.e. Not Authorized -> Available)

Note on applying accessing codes: The serial number listed here is for your convenience and verification. If the system serial number has changed, use the “Refresh” button to update the value shown. This button will also refresh the processor group value shown from its last retrieved value.

PTF checking for each component does not take place through this interface. PTFs are checked when creating a collection or via the Check PTFs menu from the IBM i Connections View.

Use the Check for new server builds button in order to download and install the latest server builds. You can use this option instead of using the iDoctor install program.

The options available on this screen are described in the following table:

Option	Description
Change user button	This button can be used to change the current user profile used to connect to this system, without needing to go back to the IBM i Connections View.
Component list	This contains the list of all components that either are available or could be installed on the desired system. The list provides the component name, build date, access code expiration and status for each component. In some cases, if the server builds are older than the client you may see a message regarding this and the functions that may be broken if the server builds are not updated. Using the tool in this state is up to you but cannot be supported. **Note:** IBMers need to use the iDoctor internal components enabler file in order to be able to see internal components in this list.
Check for new server builds button	This button allows a user to install the latest server builds automatically without needing to revisit the iDoctor website and download and install them. **Note:** You can only use this option if you have admin level authority to your PC.
Close window after clicking Launch	Uncheck this option if you wish to launch/open a component in the background and then perform other options on this screen before continuing.
Launch button	This button will launch the selected component(s) from the list.
Access code	The access code can be entered into this box and then press the Apply button to have it take effect. You should immediately see the status and expiration date change for the component(s) the access code applies to. **Note:** The Access code will only be correct and work if the system serial number supplied to IBM is correct and the appropriate release of iDoctor on the server is installed.
System serial number	This value provides the last known system serial number value retrieved for the current system. Press the Refresh button to ensure the value shown is still valid.
Processor group	This value provides the last known processor group value retrieved for the current system. Press the Refresh button to ensure the value shown is still valid.

4.7 Power Connections View

This view is now covered in section 7 Power -> Power Connections View.

4.8 Preferences

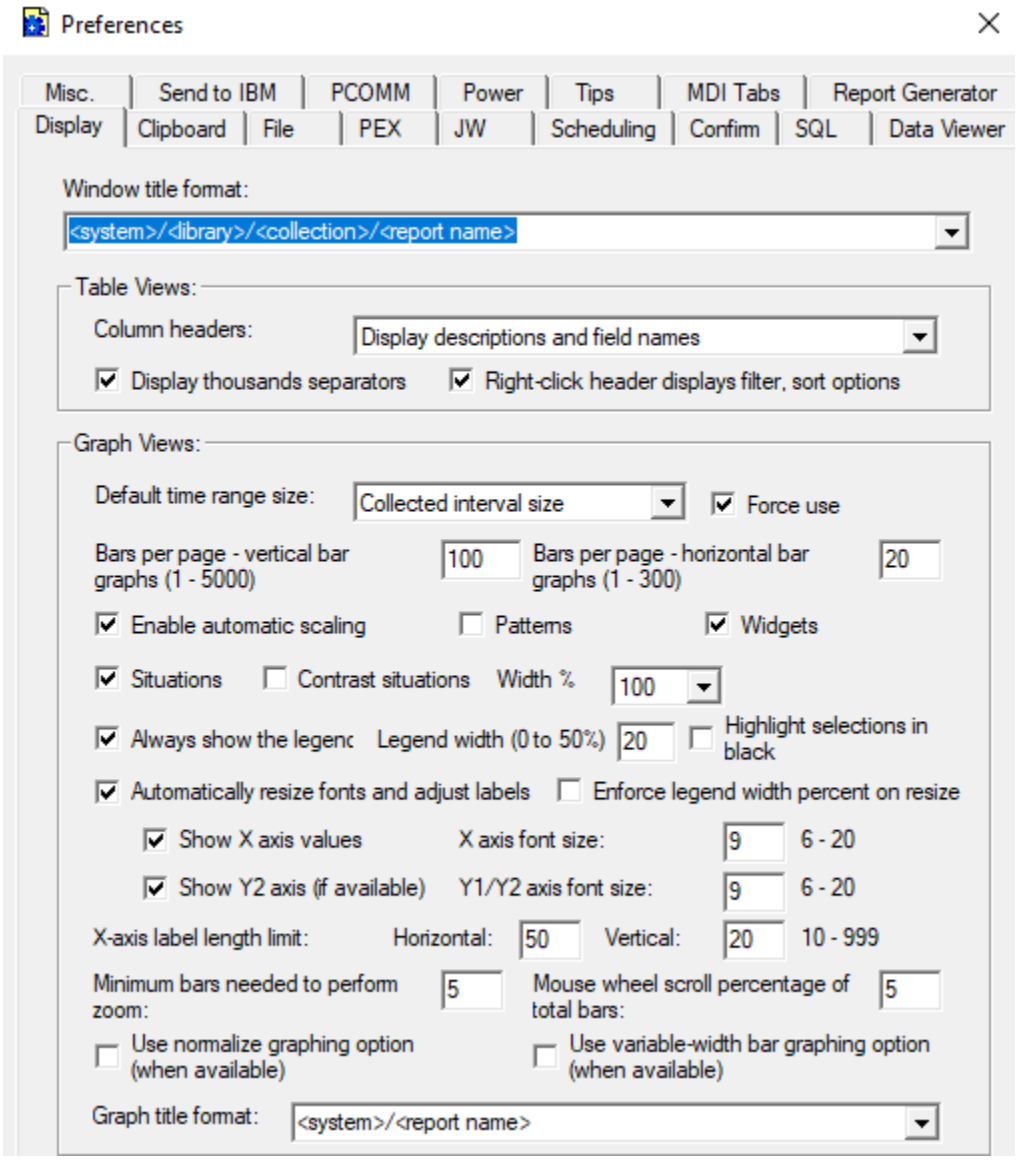
The Preferences window allows a user to work with the customizable options in the IBM iDoctor for IBM i client. Several different categories of options are available, and each category is presented on a different page.

The Preferences window is accessible via the Edit -> Preferences menu in the Data Viewer or from the iDoctor Main Window. See the next sections for information on each page in the Preferences window.

4.8.1 Display

The Display page on the Preferences window lets the user work with options that effect the visible presentation of table or graph views in the IBM iDoctor for IBM i client.

An example of this interface is shown below:



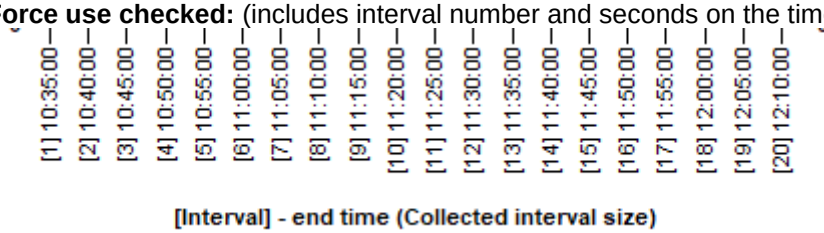
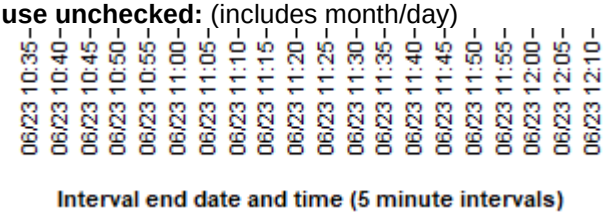
Preferences - Display

The options available on this page are:

Windows title format: Use this option to identify how the titles of iDoctor reports should be named. The dropdown list contains several different possible name formats. Other possible titles are available by modifying the value in the list and including any of the tabs listed below in <>.

<system> - The current system the data resides on
<collection system> - The name of the system the collection was created on (if known).
<library> - Library name for the collection
<library desc> - Description for the library
<collection> - Collection name
<collection start> - time the collection started (if known)
<collection end> - time the collection ended (if known)
<report name> - The report description.

Table Views Options	**Description**
Column headers	This drop down lets the user determine if field descriptions or short (SQL generated) field names or both should be displayed in the column headings for of all table views. The choices are: - Display field names - Display descriptions - Display descriptions and field names **Note:** Field names will be displayed if the descriptions are not available within the file being viewed.
Display thousands separators	This option will display thousands separators (as commas) for numeric fields in the table views. (i.e. 1000 will be displayed as 1,000)
Right-click headers displays filter, sort options	If checked, right-clicking columns headers in tables shows a menu with options instead of sorting the table in descending sequence.

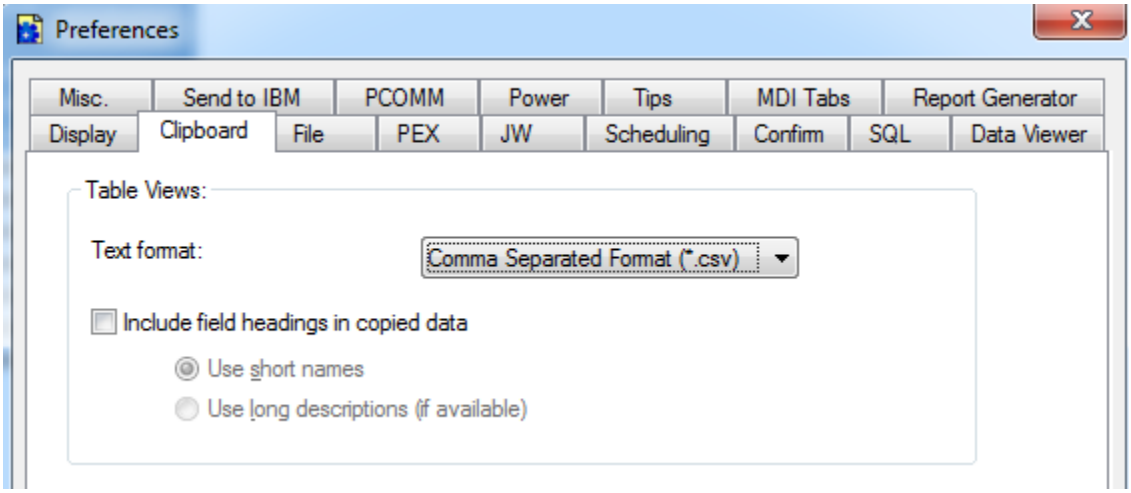
Graph View Options	Description
Default time range size	This option allows the user to control the default time range grouping for all time interval graphs in iDoctor. This option can be used to summarize many thousands of intervals in the data into a smaller set of bars in order to graph all of the data onto a single screen. For example, if you have 1000 1 second intervals in the data, and you pick 1 minute intervals as the time range size, you will end up with a graph of 17 bars summarized together instead of 1000 (the collected interval size).
Force use	This checkbox is only available if the default time range size is set to the Collected interval size. If checked (the default), then the collected interval size option is used instead of the minimum detected interval size (such as 5 min intervals in a CSI collection.) This mainly causes slight differences in the look of the X-axis label depending on if checked or not. **Force use checked:** (includes interval number and seconds on the timestamp)  **Force use unchecked:** (includes month/day) 
Bars per page - vertical	Indicates how many bars (or points if the graph is a line graph) should be displayed per page in a graph. Up to 5000 bars per page are allowed although realistically you should get this number down to perhaps 300-500 max. **Note:** Unless your monitor can display 5000 pixels it will be impossible to physically display that many bars/points. This also greatly consumes GDI objects which means you won't be able to open very many graphs at a time per iDoctor session.
Bars per page - horizontal	Indicates how many bars (or points) should be displayed per page in a horizontal bar graph. Up to 300 bars per page are allowed.
Enable automatic scaling	Indicates if the graph should automatically resize the scale on the Y-axis each time the current position in the graph changes. If this option is turned off the scale will be fixed based on the maximum and minimum values of the first page of the graph when it is opened.
Patterns	When checked, graph patterns or hatchings will be displayed to fill bars instead of solid colors. The graph patterns are configurable using the Primary Y-axis panel of the graph definition interface.
Widgets	Indicates if widgets (shapes) will be added to points on lines shown on the Y2-axis of graphs.
Situations	This preference indicates if situation background colors will be displayed on the graph. You can also control this option by using the  button on the Main Window toolbar.
Contrast situations	If checked, CSI and JW 'situations' will be displayed on the graph as the opposite of the Y1 setting for patterns.

	If patterns are in use, then solid background colors are used for situations. If solid colors are used for the bar graph then situations will be patterns.
Width %	This determines how wide each situation background is drawn as a percentage of each bar's width. 50% must be used if you wish to display a max of 2 situations per bar. 25% is used to display a max of 4 situations per bar. Note: In the case when each bar is only 15 pixels or less wide then this setting is ignored.
Always show the legend	Indicates if the graph legend should always be shown when the graph is first opened. If checked this will override the option in some IBM-supplied graph definitions that indicates the graph legend should not be shown.
Legend width percent	Indicates what percentage of the graph window the legend should consume by default. For example, if this value were 50%, graphs would be displayed with the graph on the left and the graph legend on the right with equal size.
Highlight selections in black	When checked, and multiple selections are made in graphs, they will all be shown using a black color. If this option is unchecked, then the 1st selection will be highlighted in black and the 2nd, 3rd, etc selections will use different colors.
Automatically resize fonts and adjust labels	This option controls whether or not the fonts and labels should be automatically resized and adjusted (recommended on).
Enforce legend width percent on resize	Check this option if you want to have width percentage of the graph legend shown be automatically adjusted each time the graph view window is resized. **Note:** If this option is checked then adjusting the percentage of the legend that is shown manually is not possible.
Show X-Axis values	Indicates if labels for the X-Axis values should be displayed.
X-axis font size	Indicates the font size to use for values on the X-Axis. The higher the number the larger the font will appear.
Show Y2-Axis (if available)	Indicates if the Y2-Axis (the secondary Y-Axis) should be displayed. This axis is not used on all graphs.
Y1/Y2 axis font size	Indicates the font size to use for values on the Y-Axis. The higher the number the larger the font will appear.
X-axis label length limit	Indicates the maximum number of characters to include in X-axis labels. Different values are given for horizontal or vertical bar graphs.
Minimum bars needed to perform zoom	This option can be used to change how many bars are needed to perform a zoom operation. If this value is set to a small number, then it will be more likely that the user will accidentally perform a zoom.
Mouse wheel scroll percentage of total bars	This option is used to change how much of the graph to scroll when the mouse wheel is used.
Use normalize graphing option	Indicates if the graph normalize option should be used when the graph is first opened. This option divides each time value by the interval's duration to provide a flattening effect to bar heights in the graph. This option is only available for the vertical bar time range graphs.
Use variable-width bar graphing option	Indicates if the graph variable-width bar option should be used when the graph is first opened. This option draws longer duration intervals with wider bars. This option is only available for the vertical bar time range graphs.
Graph title format	Controls how the title of the graph should appear which can differ from the Window title format if desired.

4.8.2 Clipboard

The Clipboard page on the Preferences window lets the user work with the 'Copy to Clipboard' options available for table views in iDoctor.

An example of this interface is shown below:



Preferences - Clipboard

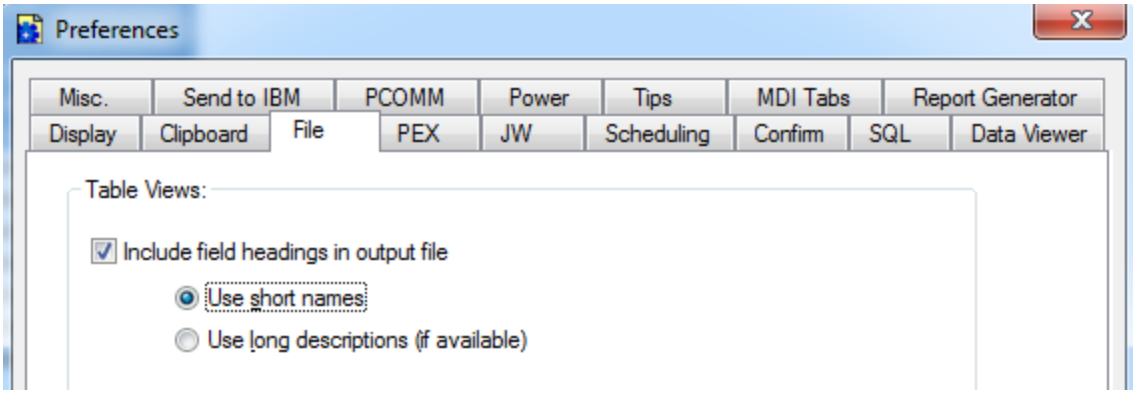
The options available on this page are summarized in the tables below:

Table Views Option	**Description**
Text format	Select the desired text format when copying records or cell selections to the clipboard. The possible choices are: comma separated, tab separated and rich text format.
Include field headings in copied data	Check this option to indicate that field headings should be included as the first record of data when copying data to the clipboard. If this option is checked you can choose to use short field names or long descriptions for the copied output.

4.8.3 File

The File page on the Preferences window lets the user work with options related to creating output files from a table view's data.

An example of this interface is shown below:



Preferences - File

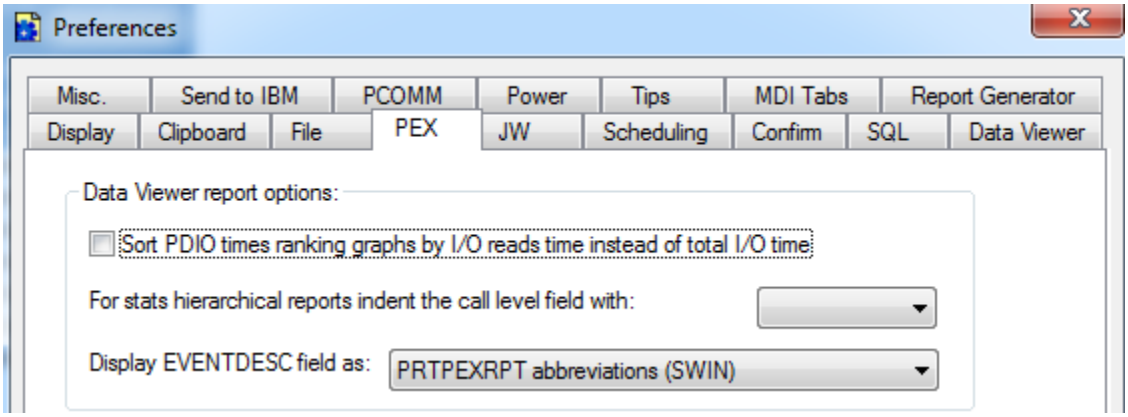
The options available on this page are:

Table Views Option	Description
Include field headings in output file	Check this option to indicate that field headings should be included as the first record of data when generating the output file. If this option is checked the user may choose to use short field names or long descriptions in the output file. To generate an output file use the File -> Save -> View As... menu for an active Table View in the Data Viewer.

4.8.4 PEX

The PEX Analyzer (PEX) page on the Preferences window lets the user work with options related to the PEX Analyzer component of IBM iDoctor for IBM i.

An example of this interface is shown below:



Preferences - PEX

The options available on this page are:

Option	Description
Sort PDIO ranking graphs by:	In the Physical Disk I/Os (PDIO) analysis the ranking graphs provide a parameter when sorting the results. This setting is used to control which field the results are to be sorted by (all in descending sequence.)
For stats hier indent call level field with	This option affects the Call Level field for a Statistical hierarchical report. The call level will be indented with the character selected in the drop-down list. The default is a single space for each call level (i.e. 5 spaces are used for 5 call levels)
Displays EVENTDESC field as	Indicates how to display the PEX event descriptions (field EVENTDESC) shown in several reports in the taskswitch analysis. An event such as 3, 8 could be described as either 3-8 or *PMCO or Base events – Performance Measurement Counter Overflow.

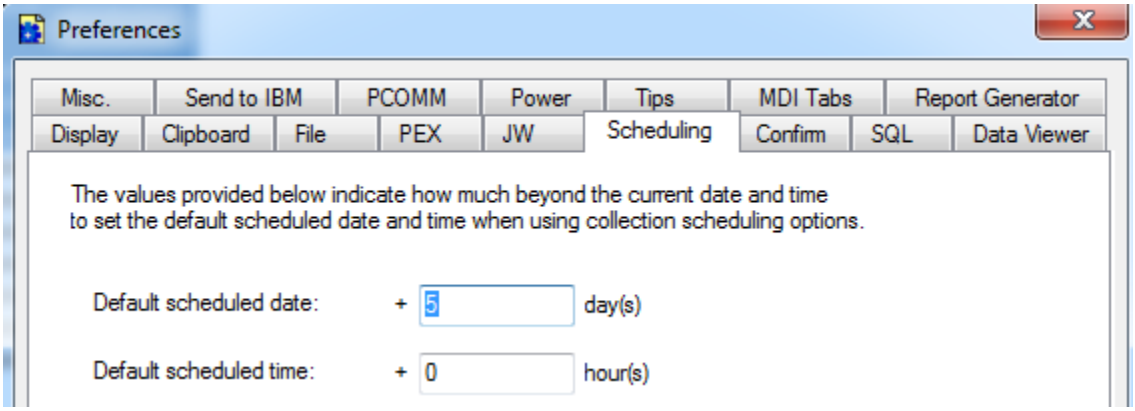
4.8.5 JW

The Job Watcher (JW) page on the Preferences window has been removed as of client 1337 and higher.

4.8.6 Scheduling

The scheduling page allows you to define the default start time when scheduling collections in iDoctor. The default is 1 week from the current date and time.

An example of this screen is the following:



Preferences – Scheduling

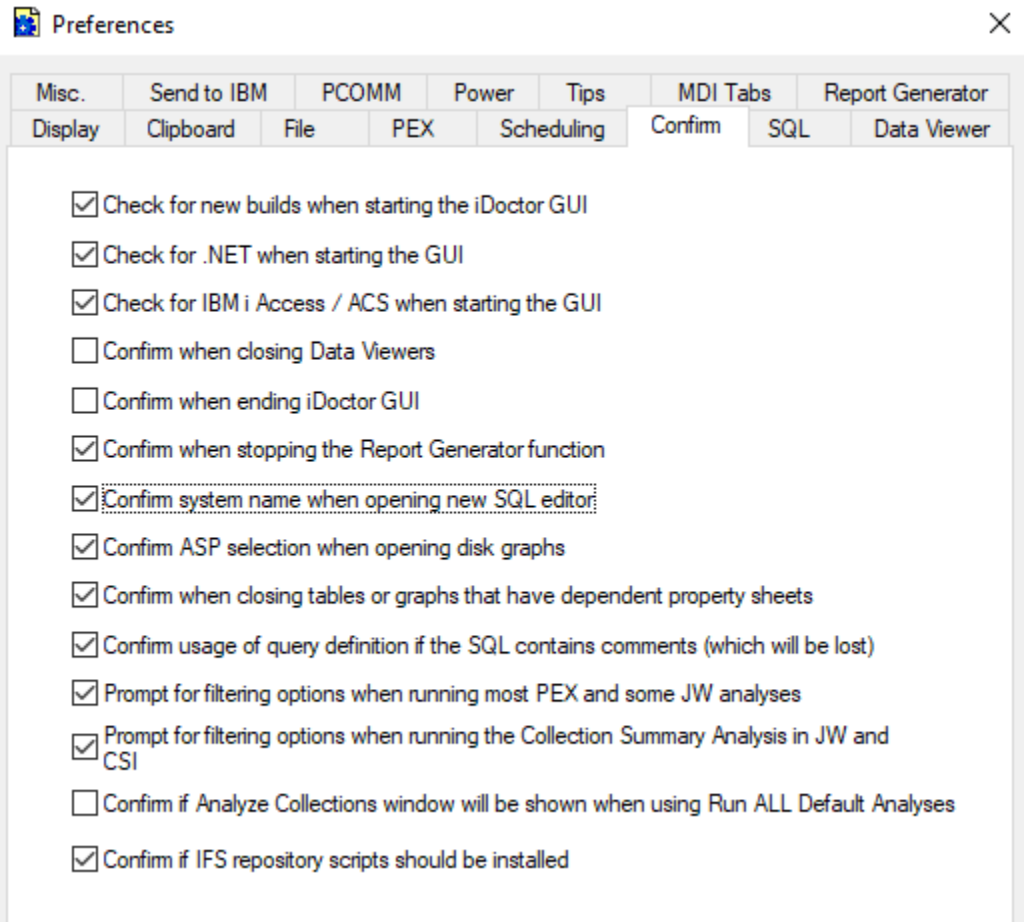
These options apply to several data collection Wizards in iDoctor when clicking the Schedule button on the Options screen.

The options available on this page are:

Options	Description
Default scheduled date	Indicates the number of days ahead (relative to the current time of the system you are connecting to) to set the default scheduled date/time.
Default scheduled time	Indicates the number of hours ahead (relative to the current time of the system you are connecting to) to set the default scheduled date/time.

4.8.7 Confirm

This page contains a set of preferences to control whether you are prompted for confirmation before performing various actions in the GUI.



Preferences - Confirm

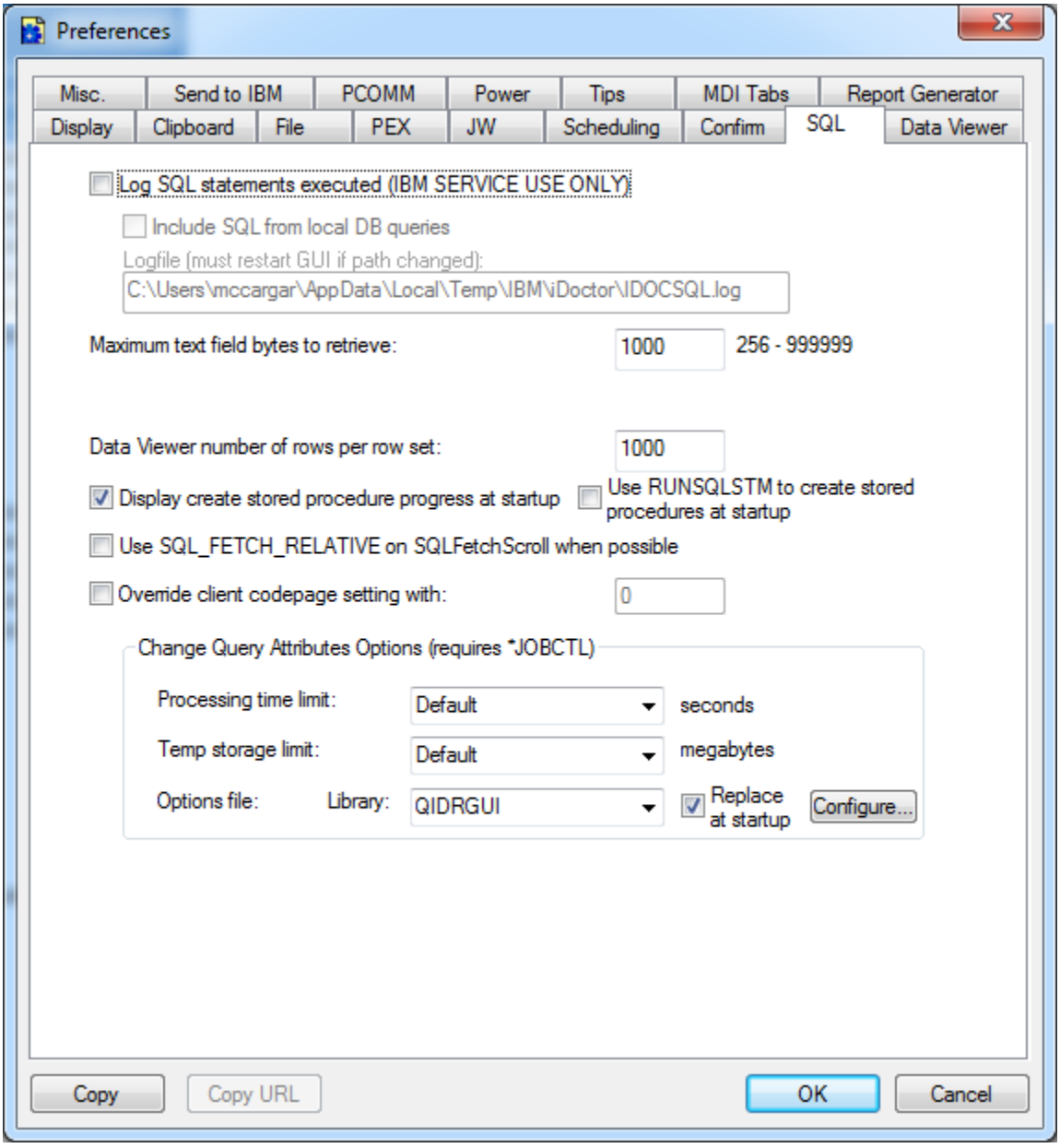
The options available on this page are:

Options	Description
Check for new builds…	This option will check for a new build each time the GUI is started and if a new version is available you will be prompted to download and install it. This feature will only work if a connection to the internet is available.
Check for .NET when starting the GUI	This option will check for the required level of .NET when starting the GUI. If not found, then you will be prompted to install it. If the required level does not exist, then some functions in the Power Connections interface will not work.
Check for IBM I Access / ACS when starting…	This option will check for the required level of either IBM i Access for Windows or System i Access Client Solutions (ACS) when starting the GUI. **Note**: If using ACS then installing the System i Access Client Solutions Windows Application Package is REQUIRED!
Confirm when closing Data Viewers	Indicates if the user should be warned before closing a Data Viewer. If unchecked and a Data Viewer is closed all views within it are shut down without confirmation.
Confirm when ending iDoctor GUI	Indicates if the user should be warned before closing the iDoctor application. If unchecked and the application is ended (close main window or Use File ->Exit menu) then all Data Viewers and views within them are shut down without confirmation.
Confirm when stopping the Report Generator function	Indicates if the user should be warned before stopping the reporting generator function while it is in progress. Closing the Data Viewer the reports are being loaded into is the method for stopping this function.
Confirm system name when opening new SQL editor	When opening a new SQL Editor, the default action is to prompt the user for the desired system to open the SQL Editor for. By unchecking this option the system will default to whichever system the user is currently working with.
Confirm ASP selection when opening disk graphs	If checked and opening disk graphs in CSI, DW or PEX that contain data from multiple ASPs, you will be prompted for the desired ASP to view the data for. From this window you will be able to select either a specific ASP or all of them if you desire. Typically for time interval and disk unit ranking graphs performance experts prefer to segregate the data ASP rather than averaging data together across the various ASPs.
Confirm when closing tables or graphs that have dependent property sheets	This option indicates if you should be prompted when closing a graph or table that has child windows opened associated with it that also must be closed at the same time.
Confirm usage of query definition if SQL contains comments	This option will prompt you if the current SQL statement contains comments and you wish to use the Query Definition interface instead of the SQL editor to modify the query. The comments and formatting are lost when using the Query Definition interface.
Prompt for time filtering options when running most PEX and some JW analyses	If checked, the user will be presented with a screen to allow them to filter the time range to include in most PEX Analyzer analyses and some in Job Watcher.
Prompt for filtering options when running Collection Summary…	If checked the user will see a screen providing filtering options when running the Collection Summary analysis. This can be useful if you only want to focus the graph output of the initial “Collection Overview” graphs to certain jobs.
Confirm if Analyze Collections window will be shown when using Run ALL default analyses	When using the Analysis option Run ALL default analyses this option will indicate if the Analyze Collection window will be shown first so the user can pick and choose which individual analyses to run.
Confirm if IFS repository scripts	If checked, you will be notified if the IFS scripts are out of date and will be asked if the latest version should be installed. Depending on the FTP settings and

should be installed	environment this option may fail so this option is provided to avoid this message.

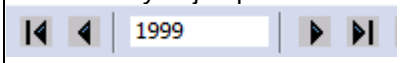
4.8.8 SQL

This page contains a set of preferences related to SQL statement processing, ODBC connection settings or the CHGQRYA defaults on the IBM i to use.



Preferences - SQL

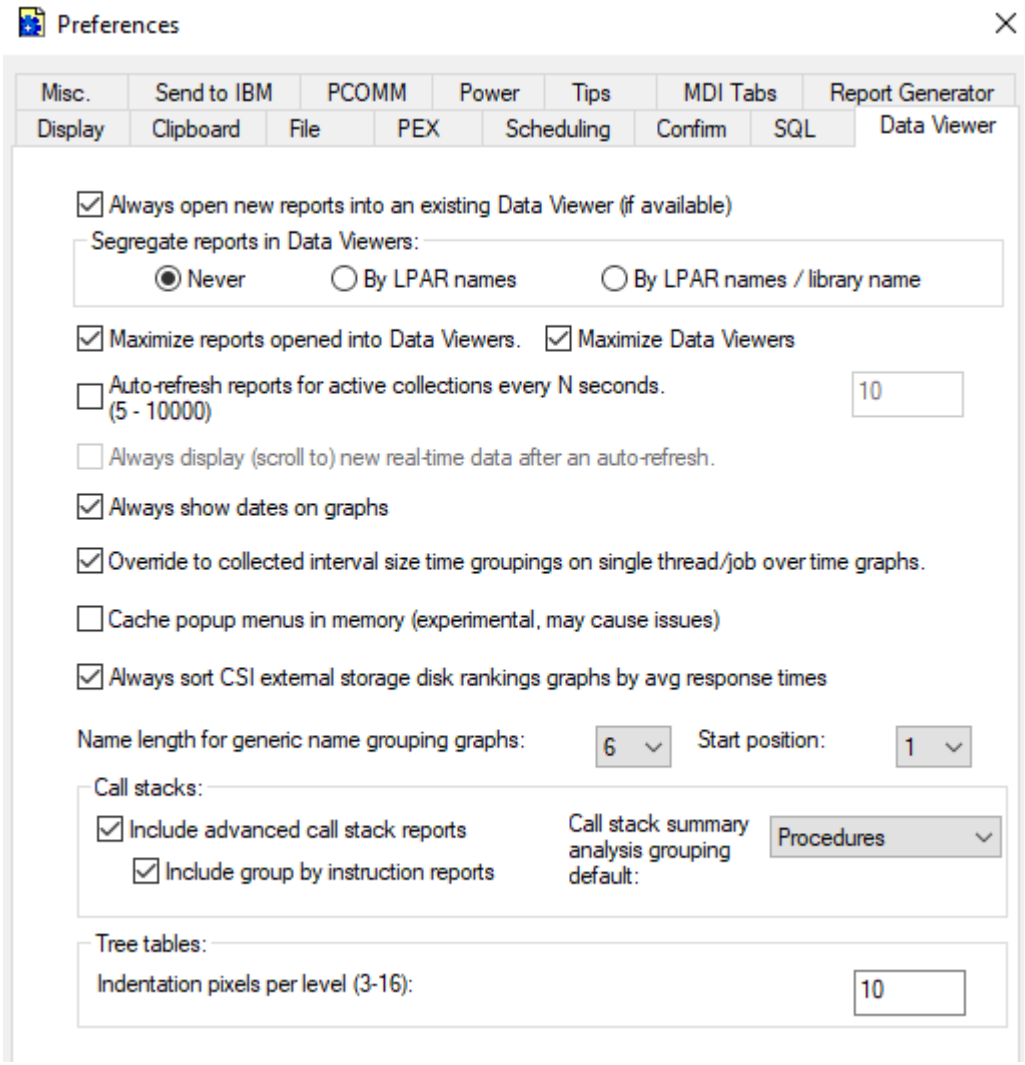
The options available on this page are:

Options	Description
Log SQL statements executed	Indicates if SQL statements and any errors should be logged to the log file shown. Other debug information is also included. This option is turned off by default but you may be directed to turn it on by IBM support. **Note:** You should not leave this option on unless necessary as it may cause crashing if you open multiple graphs/tables at once.
Include SQL from local DB queries	Indicates if SQL statements from the iDoctor component databases should be included in the log. Usually this is not desired or needed.
Log file	The location of the SQL statements log file created by iDoctor. This file is cleared every time the GUI restarts if the logging option is turned on.
Maximum text field bytes to retrieve	This value indicates the maximum amount of data that should be loaded from a single cell (column/row) of data. If the true value exceeds this number of characters then the data will be truncated. **Note:** Making this value very large will slow down report building.
Data Viewer number of rows per row set	This indicates the number of rows per block of rows in iDoctor. On the toolbar of the Data Viewer toolbar are controls that look like this. These controls let you jump around in the data you are currently looking at.  This value indicates how many rows of data will be skipped each time the inner buttons above (forward/backward) are pressed. **Note:** The outer buttons will take you to the beginning/end within the data.
Display Create Stored Procedure SQL script progress	Indicates if the user should be able to view the progress of stored procedures being created when connecting to a system.
Use RUNSQLSTM to create stored procedures at startup	This option is used to speed up the creation of any needed stored procedures when the GUI starts by building a list of SQL statements and having it run on the server via the RUNSQLSTM command. Otherwise these statements will be processed one at a time in the GUI either in the Remote SQL Statement Status view or within the GUI thread.
Use SQL_FETCH_RELATIVE	This is for IBM debug use only. It is used to improve performance when scrolling through large data sets, but may not work on older versions of IBM i Access for Windows.
Override client code page setting with	Check the box only in rare cases when the iDoctor GUI is unable to translate API calls or data sent/received using IBM i Access for Windows or IBM i Access Client Solutions. Work with support for assistance with this setting.
Change query attributes options	These options control additional advanced preferences for how the query should be ran. *JOBCTL special authority is required in order to use them.

4.8.9 Data Viewer

The Data Viewer tab in the Preferences window lets the user work with options only related to the Data Viewer window within iDoctor.

An example of this interface is shown below:



Preferences - Data Viewer

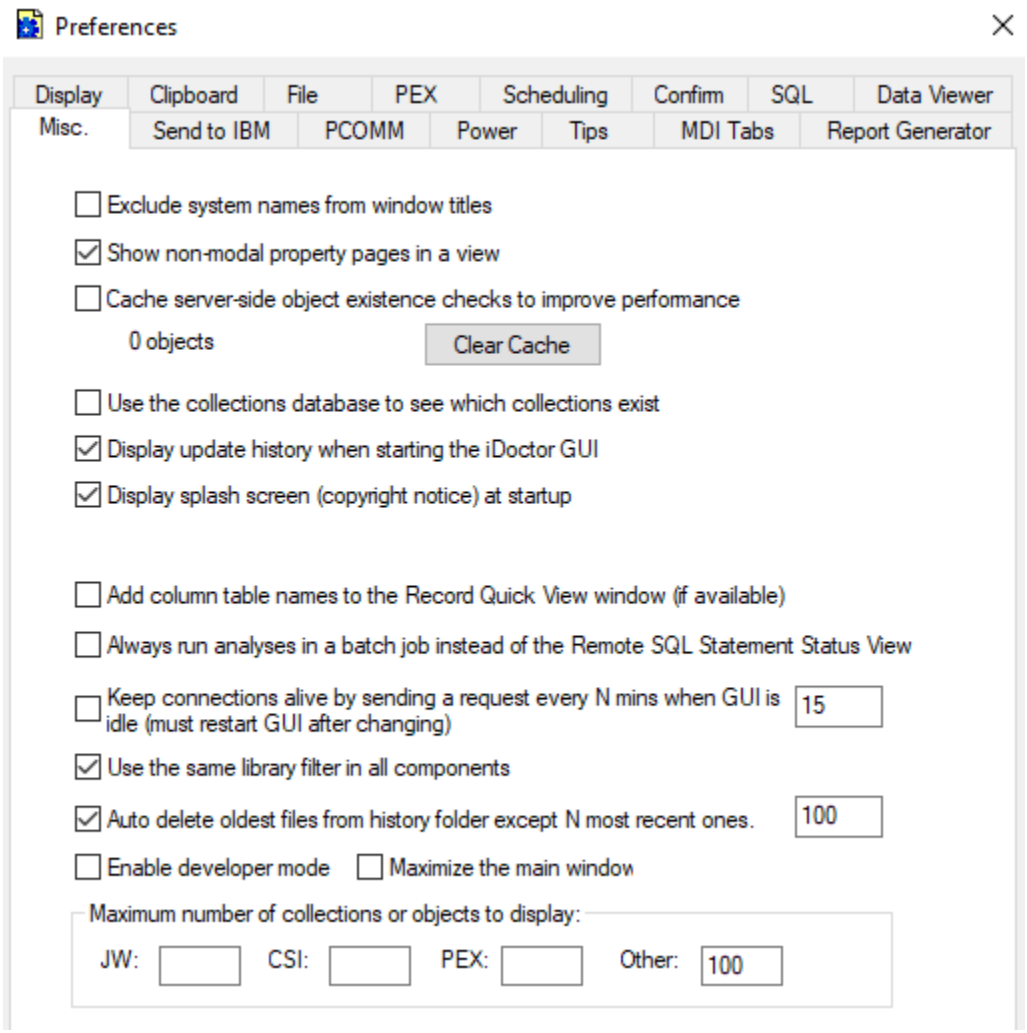
The options available on this page are:

Options	Description
Always open new reports into an existing Data Viewer	If checked and opening iDoctor tables/graphs, an existing Data Viewer will be used if one is available.
Segregate reports in Data Viewers	3 options are provided to control which Data Viewer a newly opened report will go into (if not otherwise specified): 1. **Never** - Use this option if you do not wish to group your reports by LPAR names. 2. **By LPAR names** - Use this option if you wish to group your reports by LPAR names. Two LPAR names will be shown in this format [current system / collected on system]. 3. **By LPAR names / library name** - Use this option if you wish to group your reports by LPAR name and library name. Two LPAR names will be shown in this format [current system / collected on system]. The library name on the current system where the data resides will also be shown.
Maximize reports opened into Data Viewers	When checked every view opened into a Data Viewer is maximized.
Maximize Data Viewers	This option indicates if Data Viewers should be automatically maximized when they are opened.
Auto refresh reports for active collections every N seconds	This option lets the user specify how often to auto refresh reports in the Data Viewer that are over currently active collections. **Note:** only the report with the current focus will get refreshed every N seconds.
Always display (scroll to) new data after refresh	This option indicates that after an auto refresh occurs the scrollbar should be adjusted to scroll to the end of the table or graph. This can be useful if new data is consistently being added to the end of the report.
Always show dates on graphs	If checked, then the time periods of the collection will always be shown on the graph. For time-based vertical bar graphs, the month/day will be included on the X-axis. For ranking graphs the start/end timestamps are shown at the bottom of the graph.
Override to collected interval size time grouping	Normally the overtime graphs will honor the default time range size preference. However, if this option is checked, when opening a graph showing a single job/thread over time, the time range will always be set to the collected interval size. **Note:** This option is checked by default.
Cache popup menus in memory	This option is used to improve performance when building drill down menus in the Data Viewer, but in some cases could issues.
Always sort CSI external storage disk ranking graphs by avg response times	In CSI the external storage disk ranking graphs provides an option to always sort by average response times if this option is checked.
Name length for generic name grouping graphs	This option can be used to indicate how many characters of the name to use for the generic name graphs shown in iDoctor (PEX, CSI and JW) and also the start position within the name to use. This option can apply to job names, disk resource names, etc.
Call stacks – include advanced call stack reports	This option indicates if additional call stack reports should be shown when viewing the call stack summary reports in JW, PEX. These reports provide additional columns not shown in the default views.
Call stacks – include group by instruction reports	This option indicates if additional call stack reports that group by instruction (instead of procedure) should be shown when viewing the call stack summary reports in JW, PEX.
Call stack summary analysis grouping	This option is used to control which type of grouping should be done by default when running the call stack summary analysis in JW or PEX. The choices are to

default	group by procedure or instruction. Grouping by instruction will result in fewer call stacks grouped together but provides more granularity.
Tree table indentation pixels per level	This option indicates how much indentation is used in the tree tables per level. The value provided is in pixels.

4.8.10 Misc.

This page contains a set of preferences to control some miscellaneous features.



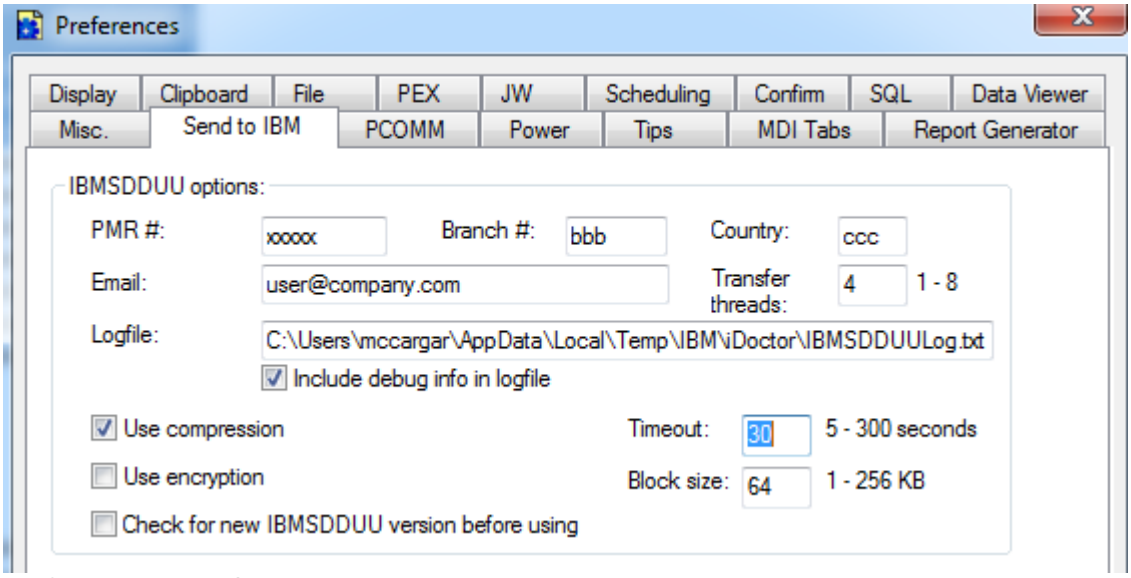
Preferences - Miscellaneous

The options available on this page are:

Options	Description
Exclude system names from window titles	This option is used to remove system names from the window titles shown in iDoctor.
Show non-modal property pages in a view	Indicates if property pages should be shown in a view. (Recommended) If this is not used property pages for interfaces such as call stacks will appear outside of the bounds of the Data Viewer window and drill down options from the call stack will appear behind the call stack (or inside of the Data Viewer).
Cache server-side object existence checks	This option allows you to keep track of all CHKOBJs performed by the GUI and not to perform them again until the cache is cleared in order to improve performance. Some interfaces will clear the cache for you in order to avoid misleading results but you may still experience misleading behavior if objects were created/deleted since the cache was cleared. It's recommended to not use this option unless the connection to the server is poor.
Use the collections database to see which collections exist	If this option is checked it can be used to speed up the display of collections in CSI, JW, DW and PEX. The downside to using this option is the data shown is cached in the collections database and may not always be up to date.
Display update history	Indicates if the iDoctor Update History screen should be shown when starting the iDoctor client. This panel lists the most recent changes made to iDoctor.
Display splash screen	Indicates if the iDoctor splash screen with copyright notice is shown when iDoctor is started.
Add column table names to the Record Quick View window (if available)	This option will add the table names to the Record Quick View window to make it easier to tell from which table the column is derived from.
Always run analyses in a batch job	This option will cause the Analyses -> Run Analysis XYZ menu options to always run the analysis in a batch job instead of the Remote SQL Statement Status View. If you are working with large collections this may be preferred.
Keep connections alive	This option can be used if connections are regularly dropped on your network to keep your iDoctor connections alive by sending small requests over all iDoctor connections at the specified time interval (in minutes.)
Use the same library filter in all components	This option is used in the Library Filter window to control if the same library for a given system will be used or not or if different library values will be remembered and used for each component.
Auto delete oldest files from history folder except from N most recent ones	This value indicates how many files in the iDoctor history/session folder will be retained. Each time iDoctor is ran a file is added to this folder. The location of this folder can be found by using the File -> Open iDoctor Session menu from the main window.
Enable developer mode	This option is intended by used by iDoctor development only.
Maximize the Main Window	This option if checked will always maximize the iDoctor Main Window when the GUI starts.
Maximum number of collections to display	This option is used to control how many collections are returned when displaying a list of collections in a library or other type of containing folder. This is most useful in cases where you have performance delays building lists of collections and only need to see the top N collections.

4.8.11 Send to IBM

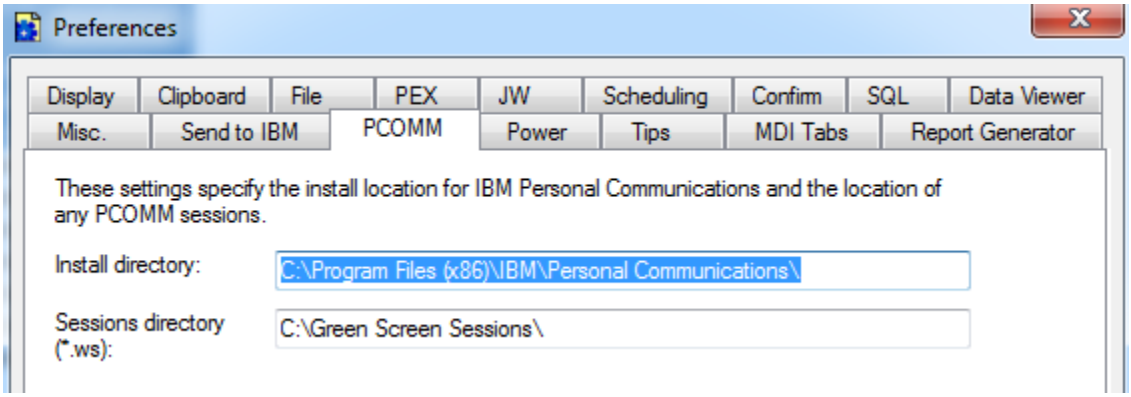
This page contains a set of preferences to control how data will be sent to IBM. These settings apply to data transfer functions. IBMSDDUU is used (optionally) to send data to IBM.



Preferences – Send to IBM

4.8.12 PCOMM

This page contains a set of preferences related to IBM Personal Communications (PCOMM.)



Preferences – PCOMM

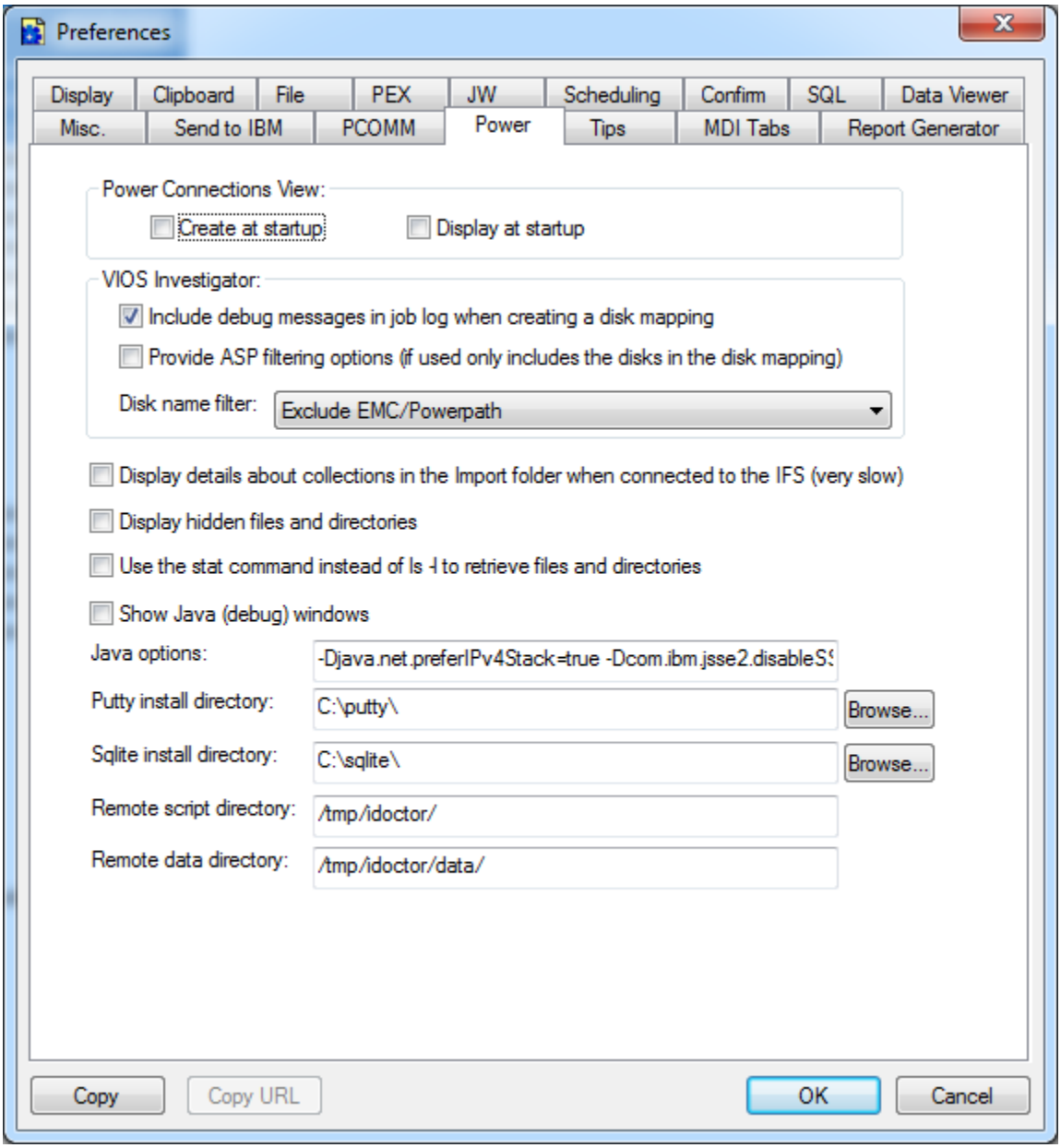
The options available on this page are:

Options	Description
Install directory	This is the directory where IBM Personal Communications is installed. If not correct, you may need to manually indicate this.
Sessions directory	This entry is the directory where *.ws files are stored on the PC. **Note:** This setting is used by the IBM i Connections View when opening PCOMM sessions for desired systems.

4.8.13 Power

The Power page on the Preferences window lets the user work with options that only apply to non IBM i systems (HMC, VIOS, AIX.)

An example of this interface is shown below:



Preferences - Power

The options available on this page are:

Power Connections View	**Description**
Create at startup	If checked, then the Power Connections View will be created when the iDoctor GUI starts.
Display at startup	If checked, then the Power Connections View will be initially shown at startup rather than displaying the IBM i Connections View when the iDoctor GUI starts.

VIOS Investigator	Description
Include debug messages in job log when creating a disk mapping	When running the VIOS to IBM disk mapping this option can be used to add extra information to the logs for IBM support purposes.
Provide ASP filtering options	When disk graphs are opened, indicates if the ASP filtering options will be shown to the user. This means a window will be shown allowing the user to select the desired ASP(s) to graph before the graph is opened. **Note:** NMON data collected on a VIOS will sometimes contain disks that are not used and not included by the LPAR the disk mapping was created at. Therefore using this option will either exclude these disks or (if disabled) include them.
Disk name filter	This option allows the user to select whether all disks will be shown on the disk graphs or only EMC/Powerpath will be shown or EMC/Powerpath will be excluded.

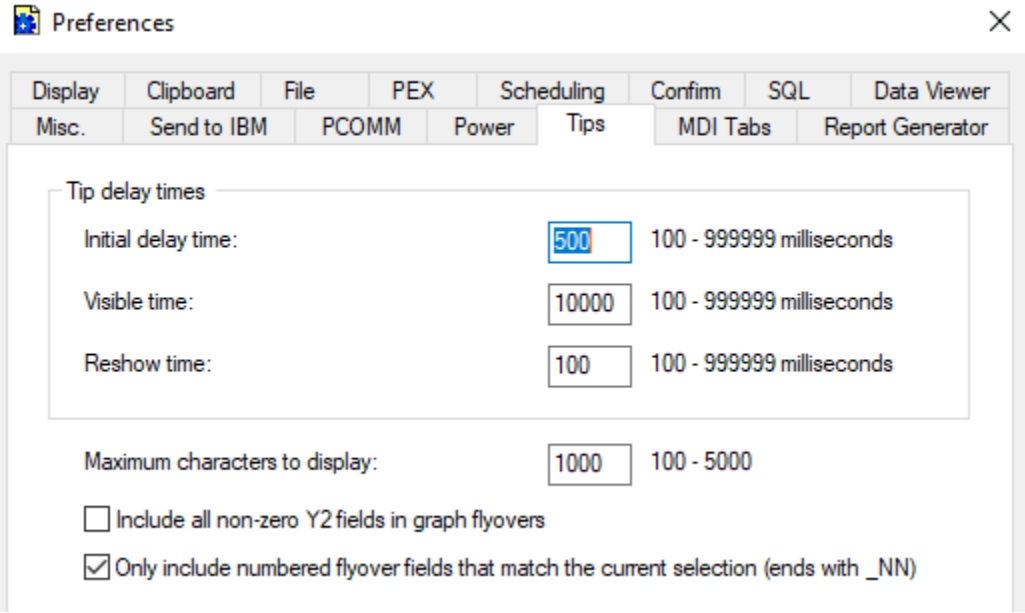
Option	Description
Display details about collections in the Import folder when connected to the IFS	When using an IBM i component under the General functions -> Power -> NMON -> Import the contents shown may optionally include details about each individual nmon file. This involves scripting that needs to open each file and is therefore much slower when enabled especially when ran on IBM i.
Display hidden files and directories	When listing directories on AIX, VIOS, Linux, etc, this option controls whether or not hidden files and directories are displayed.
Use the stat command instead of ls –l to retrieve files	Use this option to control whether or not the stat command will be used when listing directories and files.
Show Java debug windows	Java is used under the covers by the GUI to communicate via SSH to HMC, VIOS and Linux. This option allows you to either see these Java windows and commands being issued (or not.)
Java options	These options are added to each java command/session started by the iDoctor GUI. In some environments, the SSH connections may fail if these default settings are not used: -Djava.net.preferIPv4Stack=true -Dcom.ibm.jsse2.disableSSLv3=false **Note: In order to debug the SSH connections add the following to this field: -Dcom.ibm.ssh.trace=true**
Putty install directory	This path should indicate the directory where Putty has been installed.
Sqlite install directory	This is the location where Sqlite has been installed. If no IBM i is available then Sqlite on the PC is used as the analysis database by default.
Remote script directory	This directory indicates where any iDoctor scripts should be installed to. This applies to VIOS, AIX and Linux servers.
Remote data directory	This directory indicates the default location where iDoctor created data should be stored.

4.8.14 Tips

The Tips page on the Preferences window lets the user work with preferences related to tooltips that appear in dialogs, wizards and property pages.

Note: Tables and lists use tracking tooltips by design that are shown immediately when needed and these settings do not apply to them.

An example of this interface is shown below:



Preferences – Tips

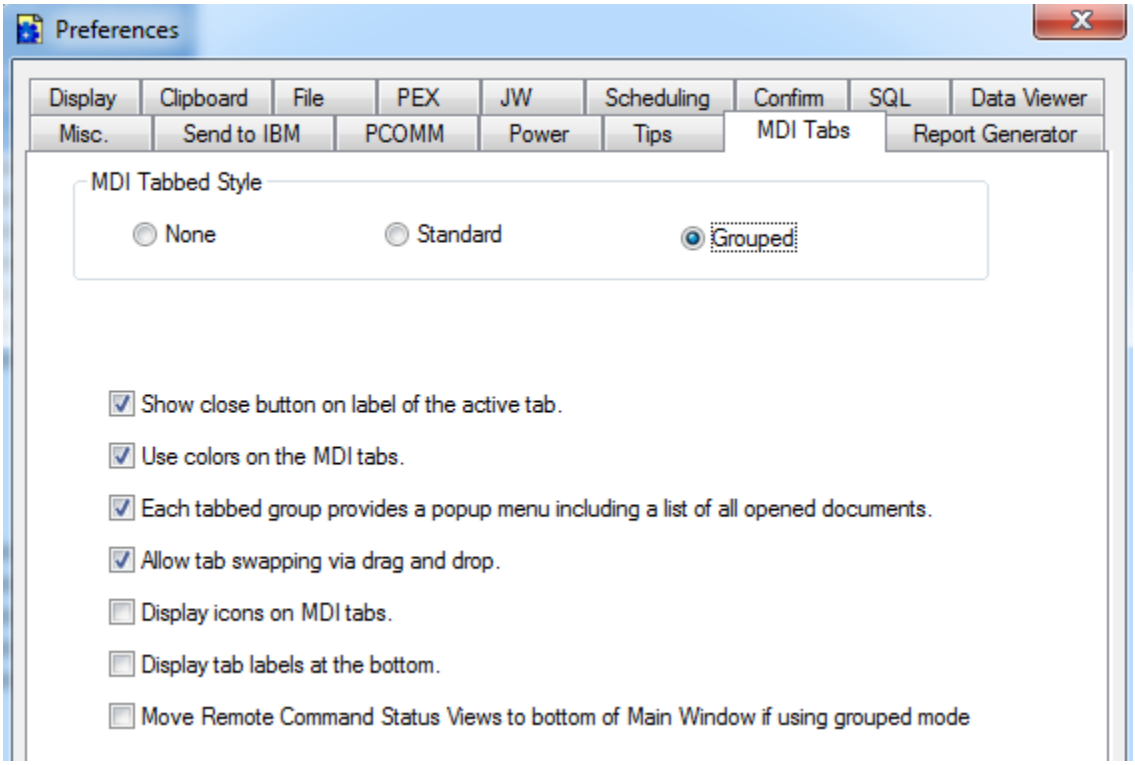
The options available on this page are:

Tip delay times	Description
Initial delay time	The amount of time (in milliseconds) the mouse pointer must remain stationary before showing a tooltip/flyover in a window for the first time.
Visible time	The amount of time (in milliseconds) the tooltip window will remain visible if the mouse pointer remains stationary.
Reshow time	The amount of delay time (in milliseconds) before showing subsequent tooltips.
Maximum characters to display	This is the maximum number of characters that should be displayed in an iDoctor tooltip.
Include all non-zero Y2 fields in graph flyovers	If checked, when viewing graph tooltips, then all Y2 fields (that are > 0) will be displayed. Otherwise only the Y2 field that has current focus will be shown in the tooltip/flyover.
Only include numbered flyover fields that match the current selection	Include only numbered flyover fields (fields that are named XYZ_01, XYZ_02, etc) that match the current selection in graph flyovers. i.e. If enabled and field ABC_02 is selected, then XYZ_02 is shown and not XYZ_03.

4.8.15 MDI Tabs

The MDI Tabs page on the Preferences window lets the user work with preferences related to the MDI Tabs style interface.

An example of this interface is shown below:



Preferences – MDI Tabs

The options available on this page are:

MDI Tabbed Style	Use this option to change the current MDI tabbed style being used. There are 3 styles of MDI tabs available in iDoctor: • None – this is a classic Windows MDI without tabs • Standard – allows users to tile and cascade but you cannot create groups of MDI tabs to compare with other tabs. • Grouped – Tabs cannot be tiled or cascaded but you can create groups of MDI tabs in order to make comparisons.

Note: The rest of these options do not apply if the MDI Tabbed Style is set to None.

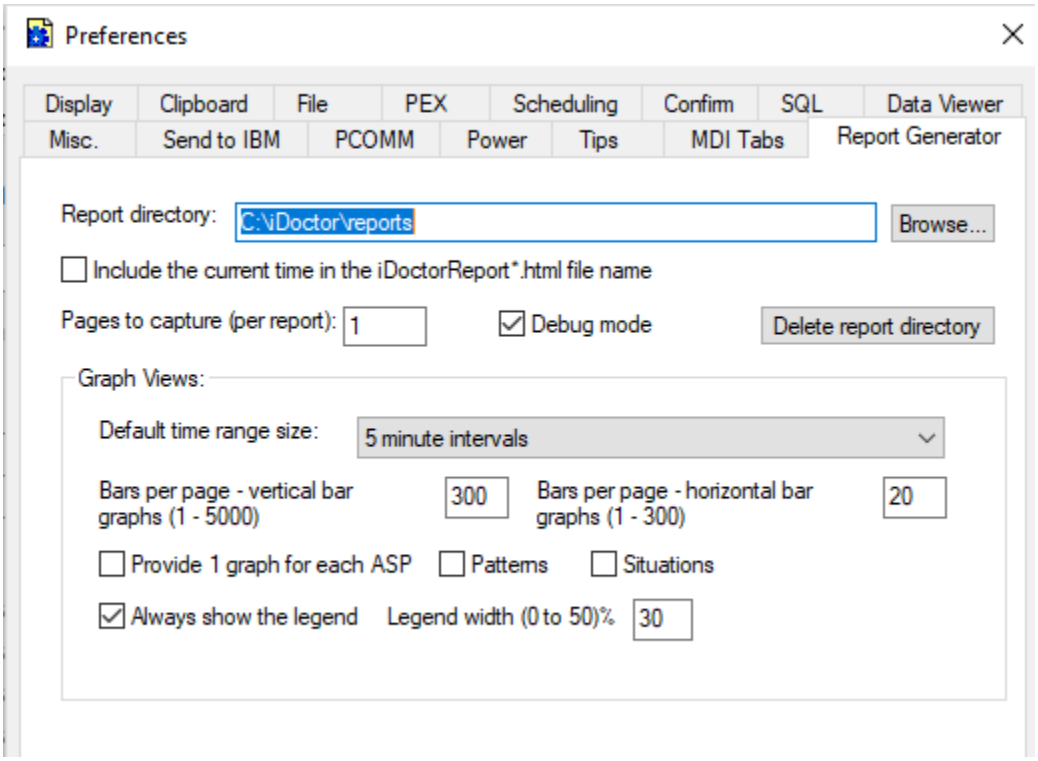
Option	Description
Show close button on label of the active tab	If checked, then the close button will be visible on the active tab. If unchecked then the close button will be placed at the far right-side of the tabbed group.
Use colors on the MDI tabs	If checked, then automatically assign a color to each tab opened.
Each tabbed group provides a popup menu	If checked, then a popup menu to show all opened views in the tabbed group will be available on the right-side of each tabbed group. If unchecked, then a left and right arrow buttons can be used to navigate through the open views.
Allow tab swapping via drag and drop	If checked, then allow tabs to be reorganized within a tabbed group via drag and drop.
Display icons on MDI tabs	If checked, then icons will be displayed on each tab.
Display tab labels at the bottom	If checked, then the tabs will be displayed at the bottom of each tabbed group rather than at the top.

Move Remote Command Status Views to bottom of Main Window if using grouped mode	This option will automatically move the Remote Command Status View and Remote SQL Statement Status View to the bottom of the main window if checked when grouped MDI tabbed style is enabled. If this option is unchecked then the tab will be created when needed and occupy the entire visible area of the Main Window.

4.8.16 Report Generator

This tab is used to control preferences related only to the iDoctor Report Generator. You can access this function by right-clicking on most types of collections in iDoctor and using the Generate Reports menu.

An example of this interface is shown below:



Preferences – Report Generator

The options available on this page are:

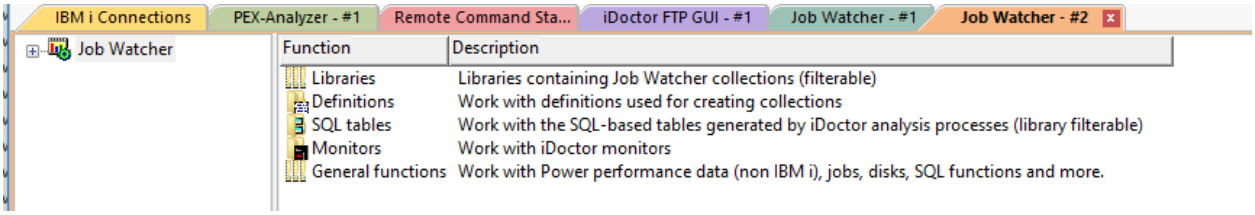
Option	Description
Report directory	This indicates the location where iDoctor Report Generator files should be generated. This consists of an HTML file for each report and several images (JPG files.)
Include the current time in the iDoctorReport*.html file name.	If checked, this option indicates that the current time will be added to the report file generated.
Pages to capture (per report)	This is the number of pages to capture per graph or table. If the amount of data in the report cannot be shown on a single page and this value is greater than 1, this means the data will be scrolled and an additional screenshot taken N times.

Debug mode	This is intended for use by IBM only.
Delete report directory	This will remove the report directory from the PC.

Graph Views	**Description**
Default time range size	This option indicates the default time range grouping to use for the report generator graphs. You may wish to use a larger value here than you would normally so more data from the collection will be summarized and visible on a single page/graph.
Bars per page – vertical bar graphs	This is the max number of bars to show on vertical bar graphs. If you wish to show more data on a single graph another option is to increase the default time range size to a larger value.
Bars per page – horizontal bar graphs	This is the max number of bars to show on horizontal bar graphs. If you wish to have the labels visible next to each bar, then it is best to keep this value fairly small (10-40.)
Provide 1 graph for each ASP	This option is used for disk graphs that contain multiple ASPs in the data. If checked for those types of graphs then a screenshot will be generated for each ASP, per report selected.
Patterns	This option uses hatchings in addition to colors to aid those with color-blindness.
Situations	This option indicates if the situational analysis background colors will be displayed on the graphs or not.
Always show the legend	This option simply indicates if the legend shall be shown or not.
Legend width percent (0 to 50)	This option indicates how much (percentage-wise) of the graph window the graph legend shall consume. It only applies if the “Always show the legend” option is checked.

4.9 Component Views

Component views are the primary means of working with any of the IBM iDoctor for IBM i components. You can have as many component views open within a Main Window as desired.



Job Watcher Component View

Component views look and feel consistently across the various components. The tree represents the hierarchy of options available within the component you are using.
Under the libraries folder, you will find all libraries on the system that contain data for the component you are working with. Under libraries you can find collections, and the reporting options available within.
Your current selection in the tree is always displayed in the list portion of the tree/list.

Tip: Because of the tendency to deal with large amounts of data and a desire to have the client perform optimally (reduce network traffic, etc.), refresh has been implemented in a way unlike most other applications. The refresh toolbar button  or menu will refresh only the contents of the selected tree branch. For example, if a library is selected in the tree, only the contents of the library will be refreshed, not the list of libraries in the tree. Refreshing the list of libraries would require selecting the folder above the list of libraries (typically the Libraries folder.)

4.9.1 Menu Options

All component folders (root folder such as Job Watcher, Collection Services Investigator) in iDoctor provide the following menu options:

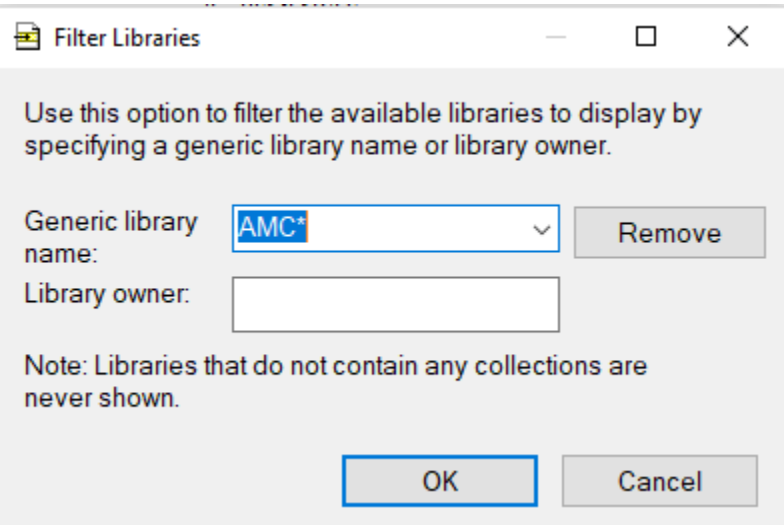
Menu	Description
Explore	Displays the contents of the root folder in the right pane of component view.
Filter libraries…	This option allows you to filter the libraries shown in the Libraries or SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of the list if the system contains many libraries containing collections (and/or SQL tables).
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.

Set User-Defined Reports Database	This option allows the user to view/modify the currently used user-defined reports database. The database can either be an MS Access file or a library on an IBM i. The database stores the information needed to build the user-defined tables and graphs shown in iDoctor.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Work with iDoctor scheduled jobs	This option is a shortcut to the General functions –> Work management -> Scheduled jobs folder. It shows all the iDoctor created scheduled jobs that exist on the current IBM i system.
Collections database	The iDoctor collection database identifies all collections on the system and can be used to facilitate the drill down from one component to another in some situations.
Properties	Use this menu to display version information for the current component installed on the current system. The build level of the GUI is also displayed here.

Additional options will be shown depending on the component.

4.9.2 Filter libraries

All IBM i component views in iDoctor offer an option to filter the list of libraries shown within the "Libraries" folder based on a generic name. Using the Filter libraries… menu from the "Libraries" folder will display the following interface:



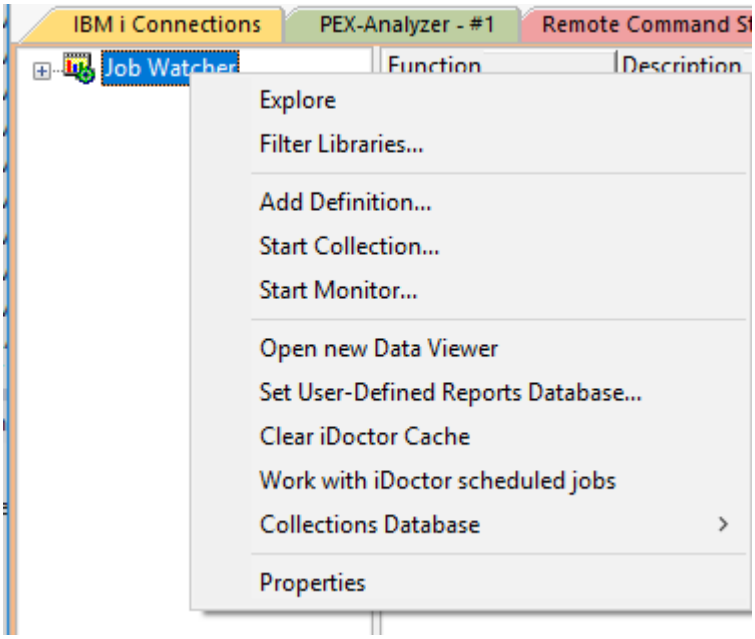
Filter Libraries window

GUI element	Description
Generic library name	This value must either contain a generic library name such as RON* or All libraries. You may select All libraries using the arrow after changing the filter to a generic name. Previous values entered into this box can be viewed within the drop-down list.
Library owner	The user profile of the library's owner. Leave blank to include all values.
Remove	Removes the current entry from the generic library list of values.

Note: The library filter applies to all IBM i components by default. However, if you wish to use a different library filter for each iDoctor component then see the Preference on the Miscellaneous tab called “Use the same library filter in all components” and uncheck it.

4.9.3 Properties

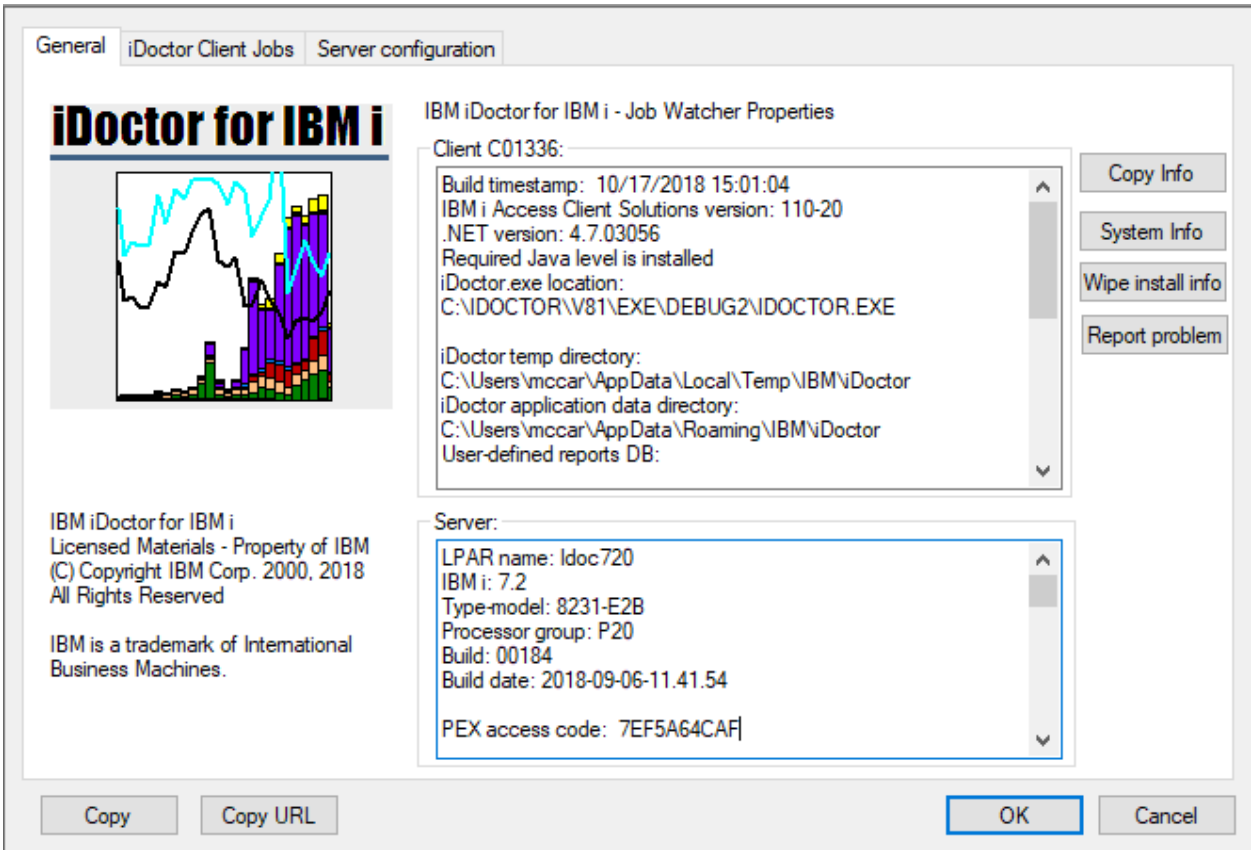
Each component view has a property page available by right clicking on the component icon and choosing the Properties... menu. The component icon is either the 'PEX Analyzer', 'Job Watcher', 'Heap Analyzer' etc depending on the component view you are working with. These properties pages offer high-level configuration settings such as: the build levels, configuring iDoctor job run priorities or listing any missing PTFs.



Job Watcher component menu options

4.9.3.1 General

An example of the General property page for Job Watcher is shown below:



Job Watcher component properties – General page

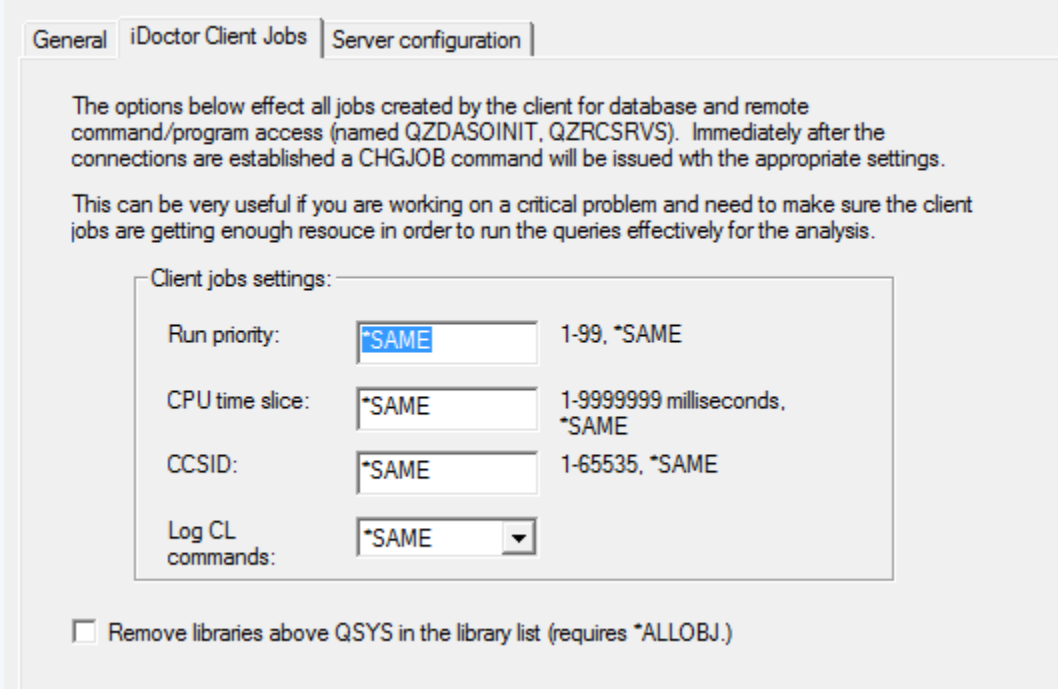
The following information is supplied within the General page of this window:

Client Version Information	**Description**
Client Cnnnnn	The client build number installed is listed near the top of this window. **Note:** Unlike the server build numbers, the client build numbers don't restart at 1 when new versions of IBM i are released.
Build timestamp	The date/time the build was produced.
IBM i access	The installed VRM and service pack level of either: 1) System i Access for Windows 2) IBM I Access Client Solutions **with the windows application package!**
.NET version	The version of .NET installed. If not updated to the required levels, then some of the FTP functions will not work.
Java version	Information about the level of Java installed. If not installed, then the user will be unable to use the SSH connections required in the Power Connections component.
iDoctor exe location	The directory and filename for the iDoctor GUI application.
iDoctor temp directory	The directory where temp files and some log files created by iDoctor are stored.
iDoctor application data directory	Files needed by iDoctor are stored in this directory. Also the Sessions are stored by default to the History sub directory.
User-defined reports DB	This value displays the location of the user-defined reports database.
Copy Info	This option will copy the data on this window to the clipboard as text.
System Info	This option will display the Windows System Information utility.
iDoctor GUI install history	A list of all iDoctor builds installed on the PC. This can be used to tell which build level was previously installed. This information is only removed if the user presses the Wipe install info button.
Wipe install info	This will remove information from the Windows registry that provides the iDoctor GUI install history.
Report problem	This will open the default email program on the PC to send an email to idoctor@us.ibm.com for support purposes. This provides debug information automatically in the email generated and is recommended!

Server Version Information	**Description**
LPAR name	The system that the current component view is connected to.
IBM i	The version and release of IBM i on the system.
Type-model	The type and model of the system.
Processor Group	The processor group of the system.
Build	Build number of this component installed on the server side. **Note: We now prefer to reference server builds by date instead of build number when contacting support.**
Build timestamp	The date/time the server build was produced. This value is shown in yyyy-mm-dd-hh.mm.ss format.
PEX access code	The last PEX Analyzer access code applied on this system.
JW access code	The last Job Watcher access code applied on this system.
PTF levels	This lists information about the JW PTFs installed (if working with Job Watcher), or the PEX PTFs installed (if working with PEX Analyzer).
Stored procedure versions	This provides a list of iDoctor stored procedures and their versions installed in the QIDRGUI library.

4.9.3.2 iDoctor Client Jobs

The following is an example of the iDoctor Client Jobs page:



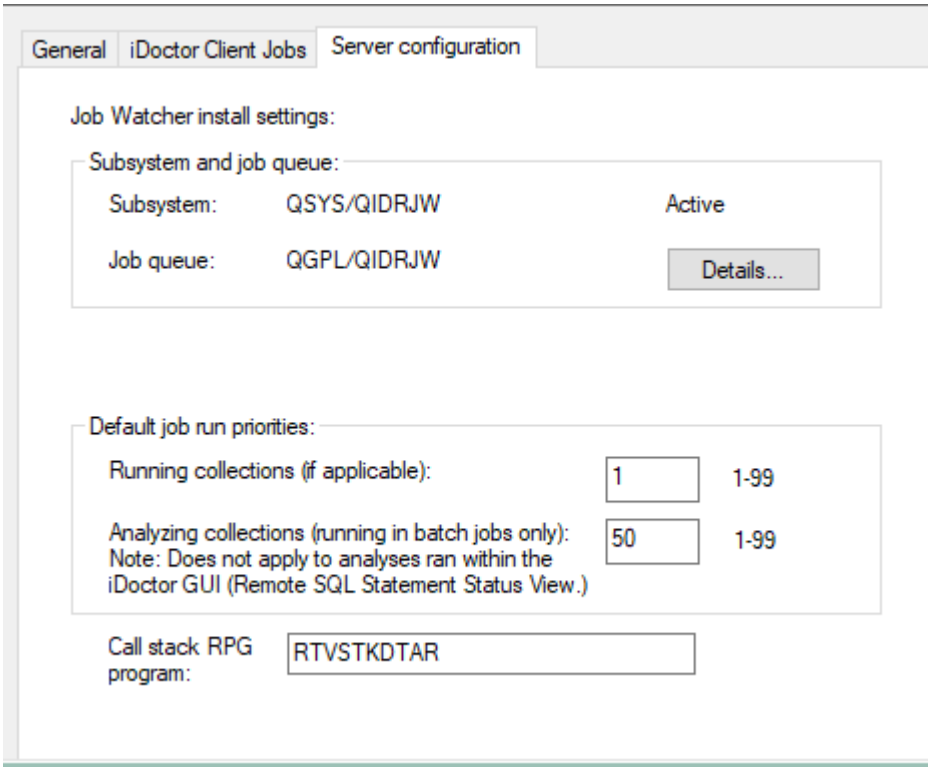
Job Watcher component properties – iDoctor client jobs page

This page lets you set the run priority and CPU time slice of all iDoctor client jobs. You can increase the run priority of the jobs that execute SQL statements that perform real-time analysis using the iDoctor GUI. This should only be set by advanced users and does require that the user profile you are connecting to the system with has *JOBCTL special authority. You must shut down the client and restart in order for any changes made on this screen to take effect.

Client job settings	Description
Run priority	Effects the run priority of all QZDASOINIT and QZRCSRVS jobs created by the iDoctor GUI. After the connections are started, the client will attempt to issue a CHGJOB command to adjust its run priority.
CPU time slice	Specifies the maximum amount of processor time (in milliseconds) given to each thread in the job before other jobs on the system are given an opportunity to run.
CCSID	The CCSID the job(s) should run under. For some analyses in PEX Analyzer CCSID 65535 must be used in order for them to run correctly (this can cause problems on DBCS system however).
Log CL commands	Indicates if CL commands should be logged to the job log or not (when possible.)
Remove libraries above QSYS in the library list	Use this option if you have other libraries above QSYS that have unexpected implementation of IBM i commands causing the GUI functions to fail. This option requires *ALLOBJ authority.

4.9.3.3 Server configuration

The following is an example of the Server configuration page.



Job Watcher component properties – Server configuration page

The subsystem and job queue used for batch jobs created by iDoctor is shown on this page.

If any of the required PTFs are not installed, they will be listed on this screen. It's not recommended to run collections until these PTFs are installed.

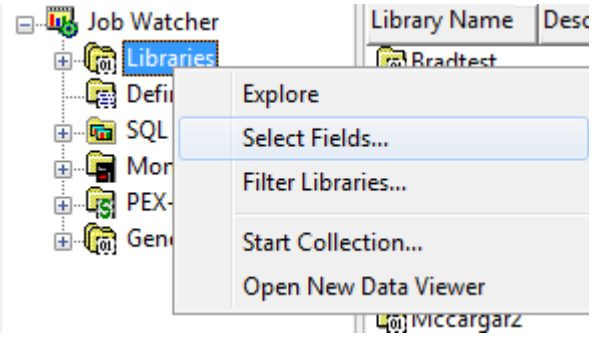
The default run priorities used when running collections and analyzing collections are shown and may be modified if desired from this page.

Note: The analyzing collections value only applies when running the analyses in batch instead of a QZDASOINIT client job. Use the iDoctor client jobs tab to affect the priority of the analysis process at those releases.

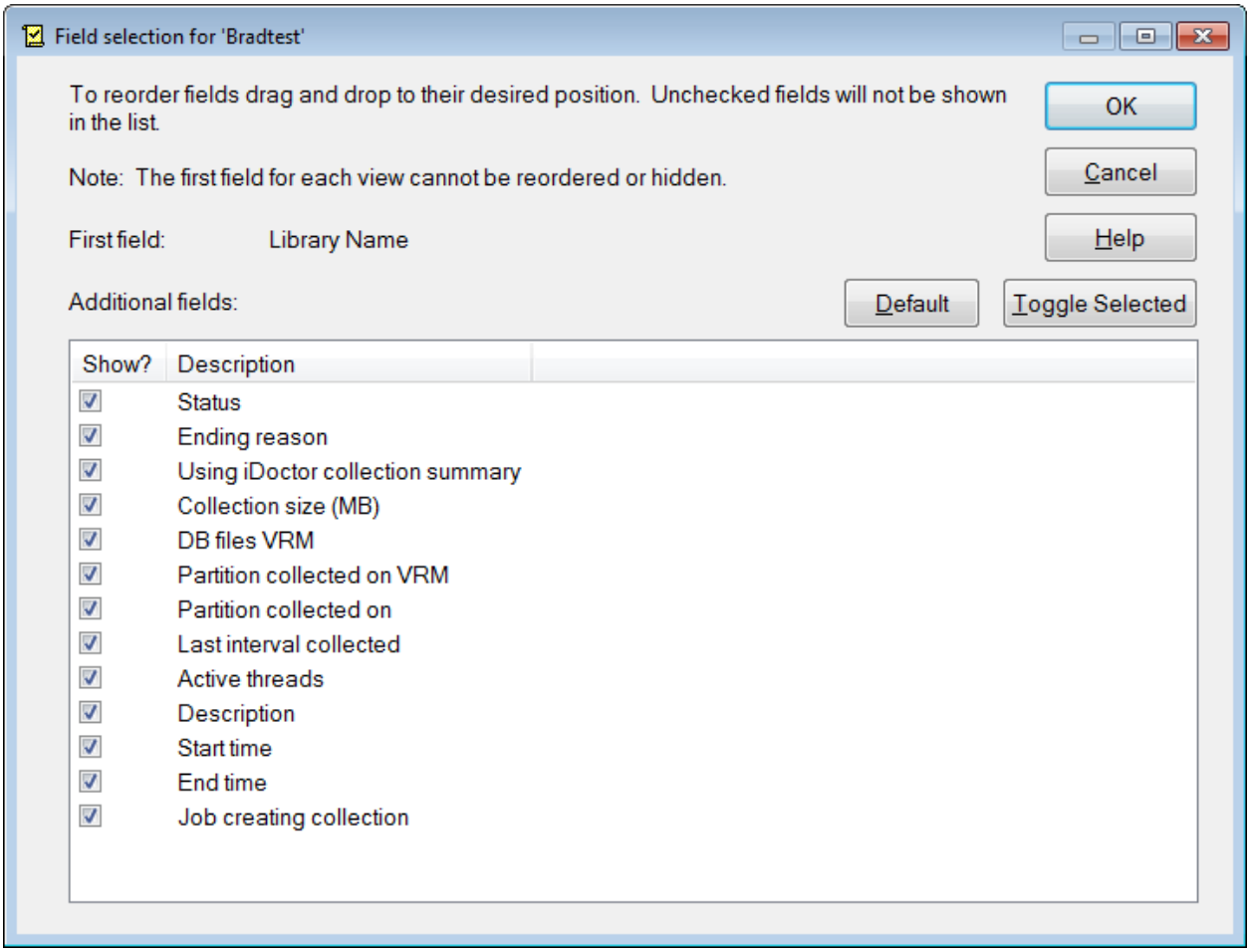
Thee call stack RPG program value is mainly for IBM internal use and effects how the Job Watcher call stacks are generated. For more information contain idoctor@us.ibm.com.

4.9.4 Field Selection Window

The Field Selection Window is a generic way to work with the fields shown in the list portion of a tree/list view. This window is available via the 'Select fields...' menu from any objects that has field selection enabled. Not all folders in the tree have field selection enabled; only those that have many available fields to display.



Using the Field Selection Window for the list of collections in a Job Watcher library



Field Selection Window

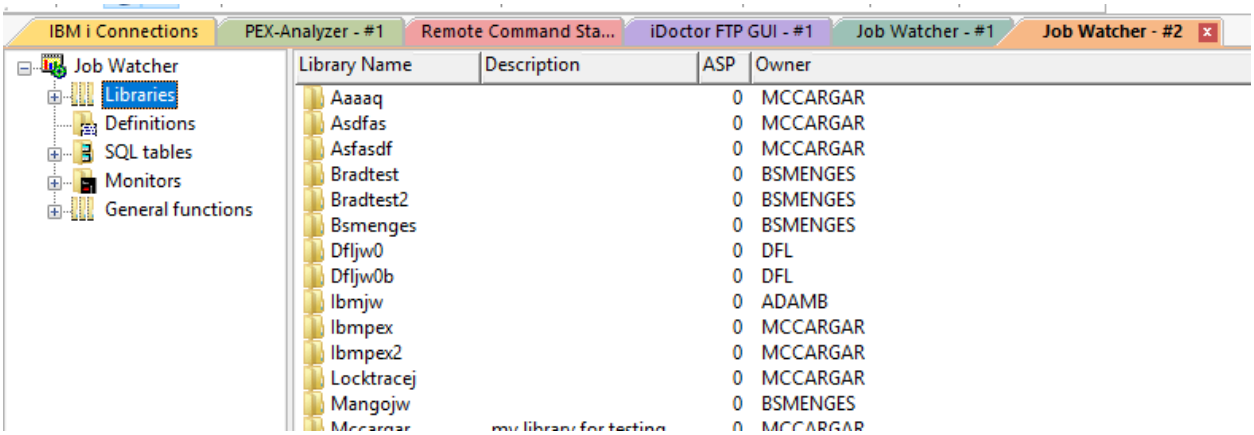
Interface	Description
First field	Lists the first field in the list. It cannot be changed.
Default button	Discards all changes and reorders the list of fields to the IBM-shipped defaults.
Toggle Selected button	Hides or shows the selected fields in the list by toggling the checkbox.
Additional fields list	List of available fields to include. You can press the space bar or click the Toggle Selected button to check/uncheck the box for the selected fields. Use drag and drop to reorder the fields in the list.

Any changes you make are saved to your PC's registry and reused the next time you open the view you are working with. To restore to the iDoctor-ship default ordering click the "Default" button. The "Toggle

Selected" button is a fast way to toggle the show checkbox for several selected fields in the list at once. To select multiples hold down the ctrl or shift key while clicking your mouse on entries in the list.

4.10 Libraries

Most components in iDoctor contain the Libraries folder. This folder displays all libraries on the system that contain applicable data for the component you are working with. The list of libraries can be filtered using the Filter libraries… menu found by right-clicking the Libraries folder.



Job Watcher Libraries Folder

All libraries have detailed properties and a set of menu options available. This section will discuss each of the library property pages in IBM iDoctor for IBM i as well as all the menu options for a library.

4.10.1 Menu Options

A library folder in iDoctor has the following menu options available by right-clicking on the library:

Menu	Description
Explore	Show the collections within the library.
Select fields...	Displays the Field Selection Window. This allows you to configure and reorder the fields that are displayed when showing the list of collections within a library.
Analyses -> Run analysis	Provides a list of available analyses you can run against all collections in all selected libraries. **Tip:** If you wish these to run in batch job rather than a QZDASOINIT job use the Preference -> Misc. -> Always run analysis in a batch job.
Copy URL	Creates a web browser link to the component and library that can be accessed later, or sent to another user.
Copy...	Allows you to copy the library's contents into a new library or into an existing one.
Save...	This option lets you save the library's contents into a save file on the server.
Transfer to...	Allows a user to create a save file of a library and transfer it to another system, to IBM or to the PC.
Clear	This option clears a library (deletes all objects in the library).
Delete	Deletes the library.
Rename	Renames the library.
Properties	Displays the property pages for the library.

Depending on the component, a library folder may have a menu option available to start a collection in the desired library.

4.10.2 Run analysis (menu)

This option when used on a library, will kick off the desired analysis on every collection found in the selected library.

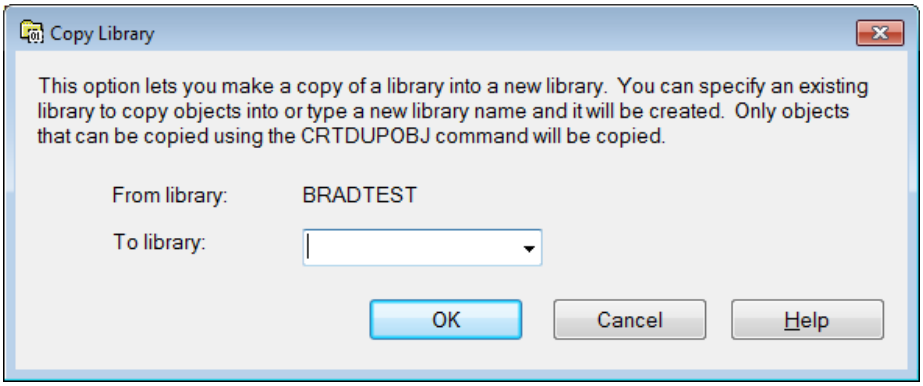
4.10.3 Copy URL

Use this option to copy and paste a URL into an email or instant messaging program to allow another user to access the same interface currently being viewed. Using this option from a library, will bring up the iDoctor GUI on the PC and open the component and library that was being viewed when this option was taken.

The URL generated by this option starts with idoctor:// and tells your web browser to launch iDoctor and perform the desired action.

4.10.4 Copy…

A library may have its contents copied into a new library or into an existing library by using the Copy... menu available by right-clicking on a library within IBM iDoctor for IBM i. This option is an interface over the CPYLIB command. The progress of the library being copied may be viewed using the Remote Command Status View.

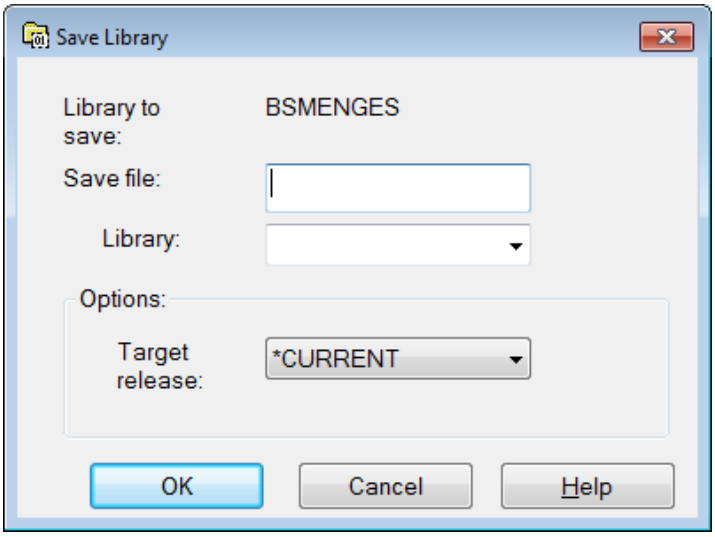


Copy Library Window

Option	Description
From library	Displays the name of the library to be copied.
To library	The name of the library that will receive the contents of the from library. By clicking the down arrow you can choose from a list of all libraries on the system.

4.10.5 Save…

A library's contents can be saved using the Save... menu available by right-clicking on a library within IBM iDoctor for IBM i. This option is an interface over the SAVLIB command. This interface is restricted to saving the library to a save file and is missing some of the advanced options on the command. The progress of the library being saved may be viewed using the Remote Command Status View.



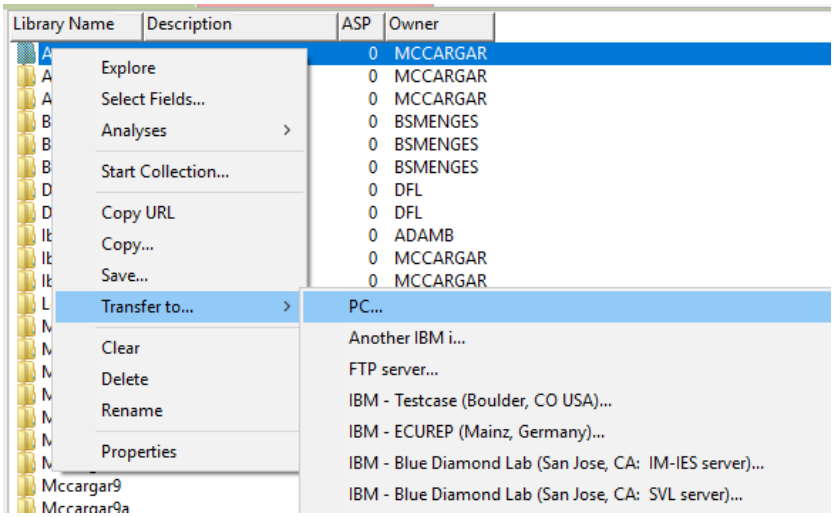
Save Library Window

Note: This interface will automatically specify high compression on the SAVLIB command.

Option	Description
Library to save	The name of the library to be saved.
Save file/library	The name of the save file and library to save the contents of the library into. If the save file doesn't exist it is created. If the save file does exist, you will be asked for confirmation before continuing.
Target release	Specifies the release of the operating system on which you intend to restore and use the object.

4.10.6 Transfer

This transfer options for libraries consist of the following choices:

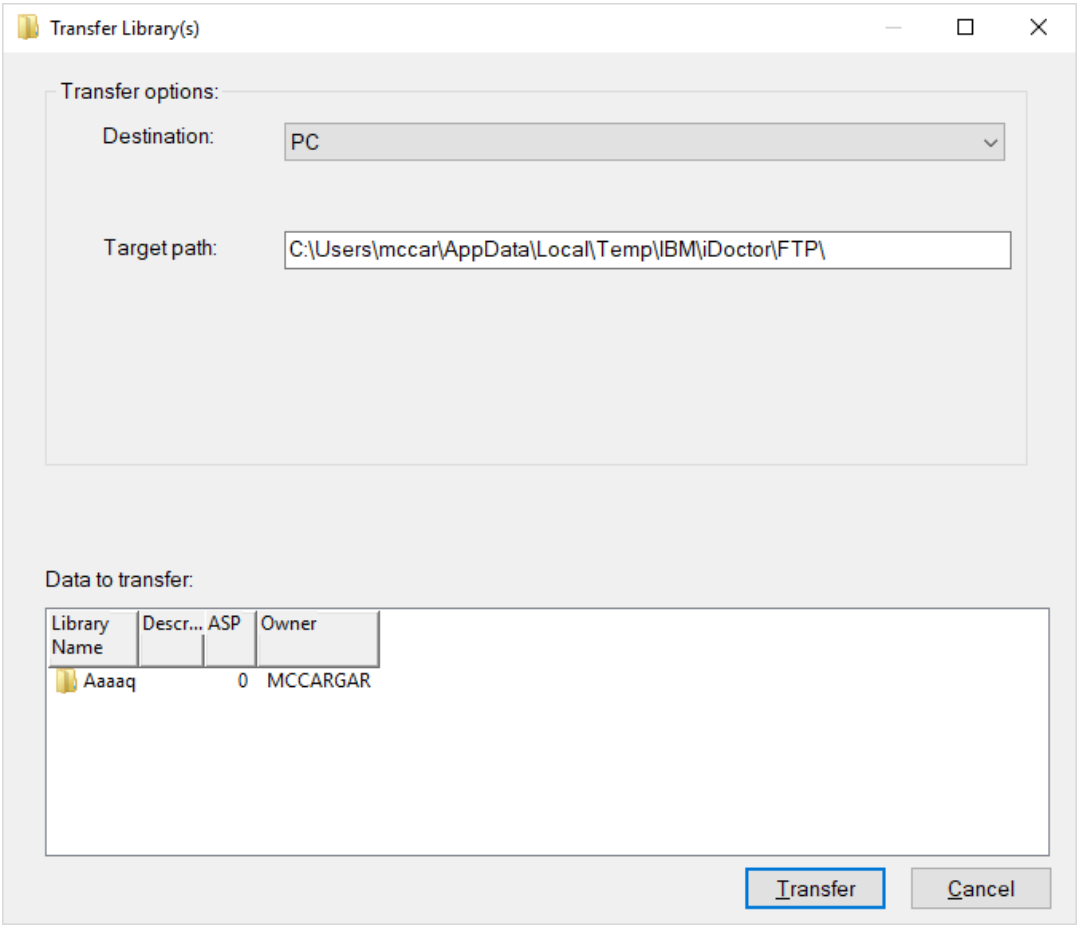


Library Transfer to menu in Job Watcher

Picking one of these will take you to a Transfer window with this selection premade.

4.10.6.1 Transfer to PC

This option will transfer one or more libraries to the specified location on the current PC.

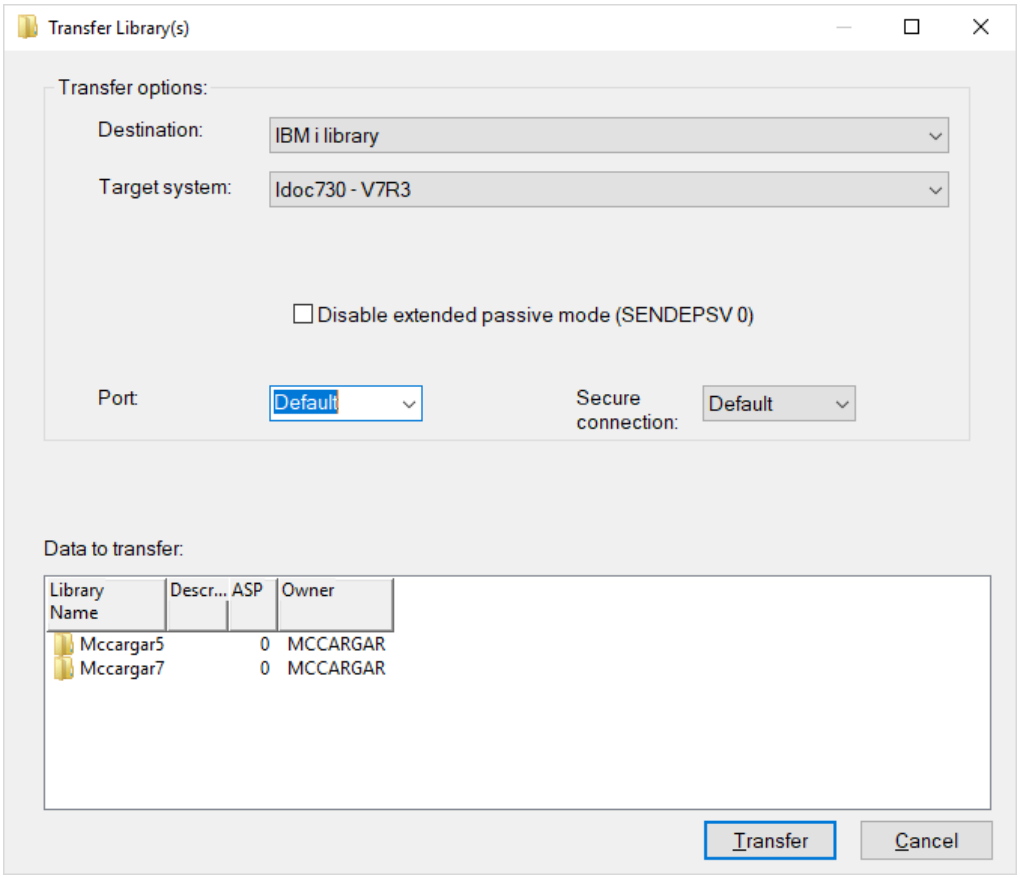


Transfer Library(s) to the PC

Option	Description
Destination	This indicates where you will be sending data to.
Target path	The directory on the PC to save the library(ies) to. They will be sent to this path where each library is a save file in this directory. **Note:** You can also specify a savf name on the PC here (that doesn't match the library name), but then this option would only work for a single library at a time.
Data to transfer	This is the list of libraries to transfer

4.10.6.2 Transfer to another IBM i

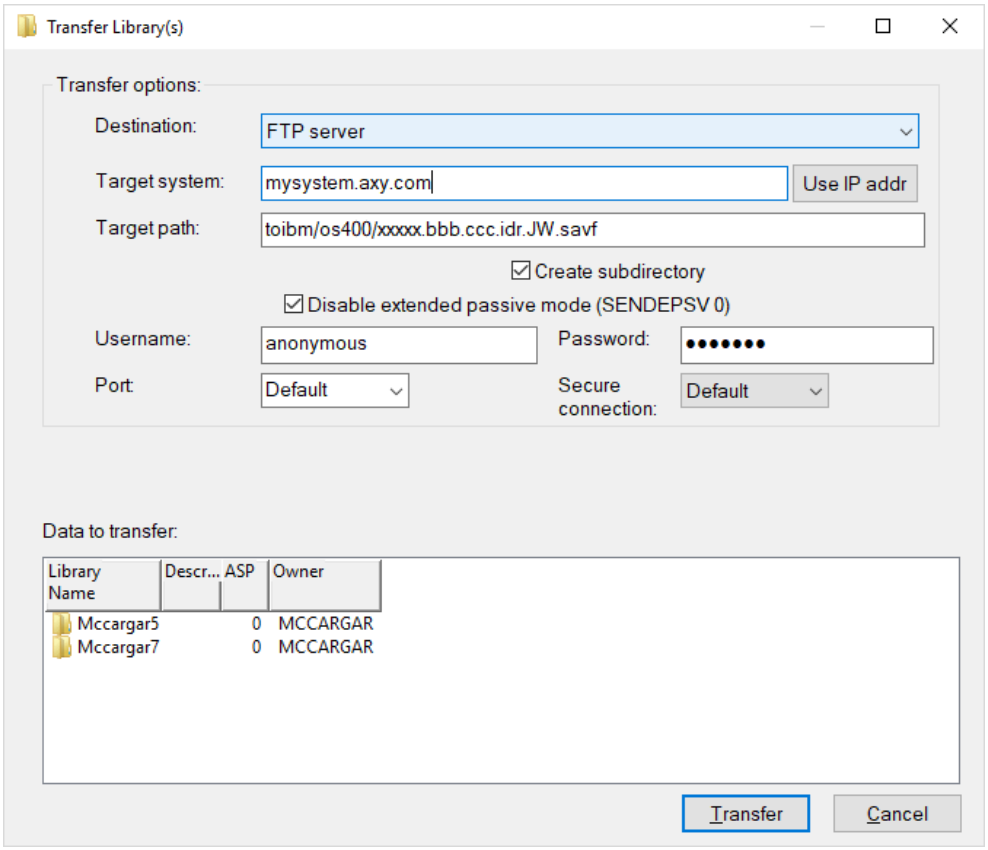
This option allows you to transfer one or more libraries to another IBM i and have them automatically restored.


Transfer Library(s)
Transfer options:
Destination:
IBM i library
Target system:
Idoc730 - V7R3
Disable extended passive mode (SENDEPSV 0)
Port:
Default
Secure connection:
Default
Data to transfer:
Library Name
Descr...
ASP
Owner
Mccargar5
0
MCCARGAR
Mccargar7
0
MCCARGAR
Transfer
Cancel

Option	Description
Destination	This indicates where you will be sending data to.
Target system	The IBM i to save and restore the library(ies) to.
Disable extended passive mode	Disables a setting in the FTP transfer of the libraries. In some environments this is required. Libraries are sent directly from the current IBM i to the target system (IBM i). You can read more about this setting here: https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzaiq/rzaiqsendepsv.htm
Port	The FTP port to use for the transfer. (1-65535 are valid) **Default:** 21 **Secure:** 990 **Note:** This parameter is passed down to the PORT parameter on the FTP command on the IBM i.
Secure connection	Specifies the type of security mechanism to be used for protecting information transferred on the FTP connection (which includes the password used to authenticate the session with the FTP server). Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are compatible protocols which use encryption to protect data from being viewed during transmission and verify that data loss or corruption does not occur. **Default**: If the PORT parameter specifies Secure or 990, Implicit is used; otherwise, None is used. **Implicit:** The FTP client immediately attempts to use TLS/SSL when connecting to the specified FTP server (without sending an AUTH subcommand to the server). If the server does not support implicit TLS/SSL on the specified port, or the TLS/SSL negotiation fails for any reason, the connection is closed. **SSL:** After connecting to the specified FTP server, the FTP client sends an AUTH (authorization) subcommand requesting a TLS/SSL protected session. If the server supports TLS/SSL, a TLS/SSL negotiation performed. If the server does not support TLS/SSL or the TLS/SSL negotiation fails, the connection is closed. **None:** The FTP client does not use encryption when connecting to the specified FTP server. **Note:** This parameter is passed down to the SECCNN parameter on the FTP command on the IBM i.
Data to transfer	This is the list of libraries to transfer

4.10.6.3 Transfer to FTP server

This option allows you to transfer one or more libraries to a directory on another system. The libraries are sent by default as 1 savf per library having the same name as the library (i.e. libname.savf.)



Transfer to FTP server

Option	Description
Destination	This indicates where you will be sending data to.
Target system	The system to send the library(ies) to.
Use IP addr(ess)	This will change the target system specified to its IP address if the iDoctor GUI is able to determine it. In many situations it is better to transfer by IP address depending on the network/system configuration because the target system name may not be known from the IBM i you are using. If that is the case the connection will fail.
Target path	The location on the system to send the SAVFs to.
Create subdirectory	This option (if checked) will create on the target system any subdirectories required by using the target path specified. Otherwise the transfer will fail if the location does not already exist or the user lacks the ability to create directories.
Disable extended passive mode	Disables a setting in the FTP transfer of the libraries. In some environments this is required. Libraries are sent directly from the current IBM i to the target system. You can read more about this setting here: https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzaiq/rzaiqsendepsv.htm
Username	The user name to connect (via FTP from the current IBM i) to the target system with.
Password	The password to connect (via FTP from the current IBM i) to the target system with.
Port	The FTP port to use for the transfer. (1-65535 are valid) **Default:** 21 **Secure:** 990 **Note:** This parameter is passed down to the PORT parameter on the FTP command on the IBM i.
Secure connection	Specifies the type of security mechanism to be used for protecting information transferred on the FTP connection (which includes the password used to authenticate the session with the FTP server). Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are compatible protocols which use encryption to protect data from being viewed during transmission and verify that data loss or corruption does not occur. **Default**: If the PORT parameter specifies Secure or 990, Implicit is used; otherwise, None is used. **Implicit:** The FTP client immediately attempts to use TLS/SSL when connecting to the specified FTP server (without sending an AUTH subcommand to the server). If the server does not support implicit TLS/SSL on the specified port, or the TLS/SSL negotiation fails for any reason, the connection is closed. **SSL:** After connecting to the specified FTP server, the FTP client sends an AUTH (authorization) subcommand requesting a TLS/SSL protected session. If the server supports TLS/SSL, a TLS/SSL negotiation performed. If the server does not support TLS/SSL or the TLS/SSL negotiation fails, the connection is closed. **None:** The FTP client does not use encryption when connecting to the specified FTP server. **Note:** This parameter is passed down to the SECCNN parameter on the FTP command on the IBM i.
Data to transfer	This is the list of libraries to transfer

4.10.6.4 Transfer to IBM – Testcase / Transfer to IBM - ECUREP

These options are used to send your library of data to IBM for analysis. Typically, you will need a PMR # to associate this data with.

It is possible that in the future these options will require a secure FTP connection, but as of this writing (Oct 19th, 2018) both do not. Keep in mind that transferring data over an unsecure connection could risk your password being compromised.

The options for these modes are the same as the previous section (Transfer to FTP server), except there is also an option to transfer using IBMSDDUU. This is a secure transfer mechanism written in Java that may optionally be used if desired.

4.10.6.5 Transfer to IBM - Blue Diamond Lab

These options are used to send your library of data to IBM for analysis. You will need a PMR # to associate this data with and a secure FTP connection method.

The same settings are used on this screen as those in the previous section.

4.10.7 Clear

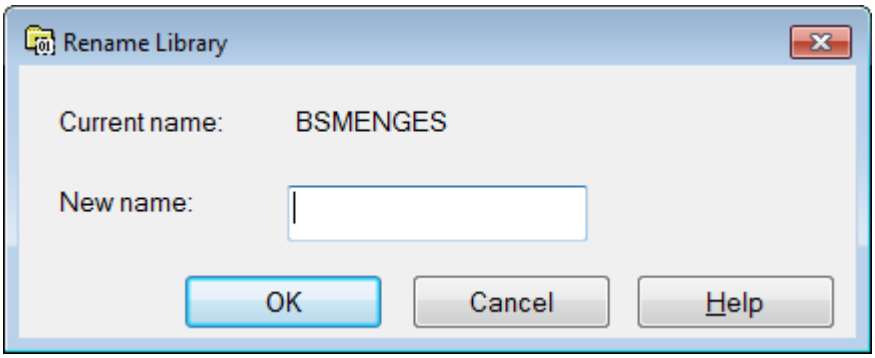
A library's contents may be cleared using the Clear... menu available by right-clicking on a library. This option will initiate (after confirmation) a CLRLIB command over the specified library. The progress of the library being cleared may be viewed using the Remote Command Status View.

4.10.8 Delete

A library may be deleted using the Delete... menu available by right-clicking on a library. This option is an interface over the DLTLIB command. The progress of the library being deleted may be viewed using the Remote Command Status View.

4.10.9 Rename

A library may be renamed using the Rename... menu. This option is an interface over the RNMOBJ command.



Rename Library Window

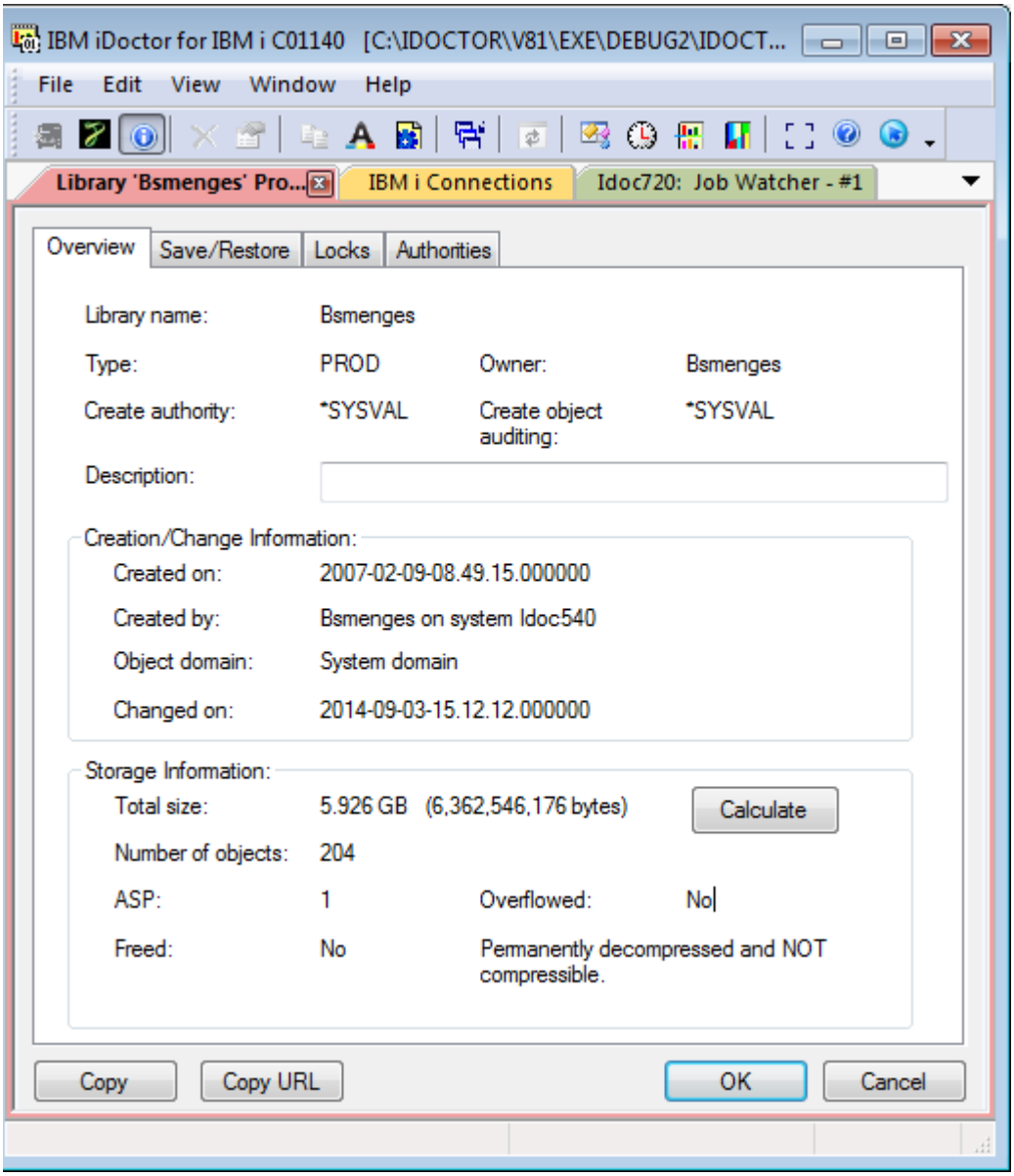
Option	Description
Current library name	The name of the library to be renamed.
New library name	The name to replace the current library name.

4.10.10 Properties

The library property pages are accessible by right-clicking on a library and choosing the Properties menu. The next section discusses the library properties pages.

4.10.10.1 Library Properties – Overview

The overview property page for libraries displays basic information about the library, including the total size of all objects in the library.



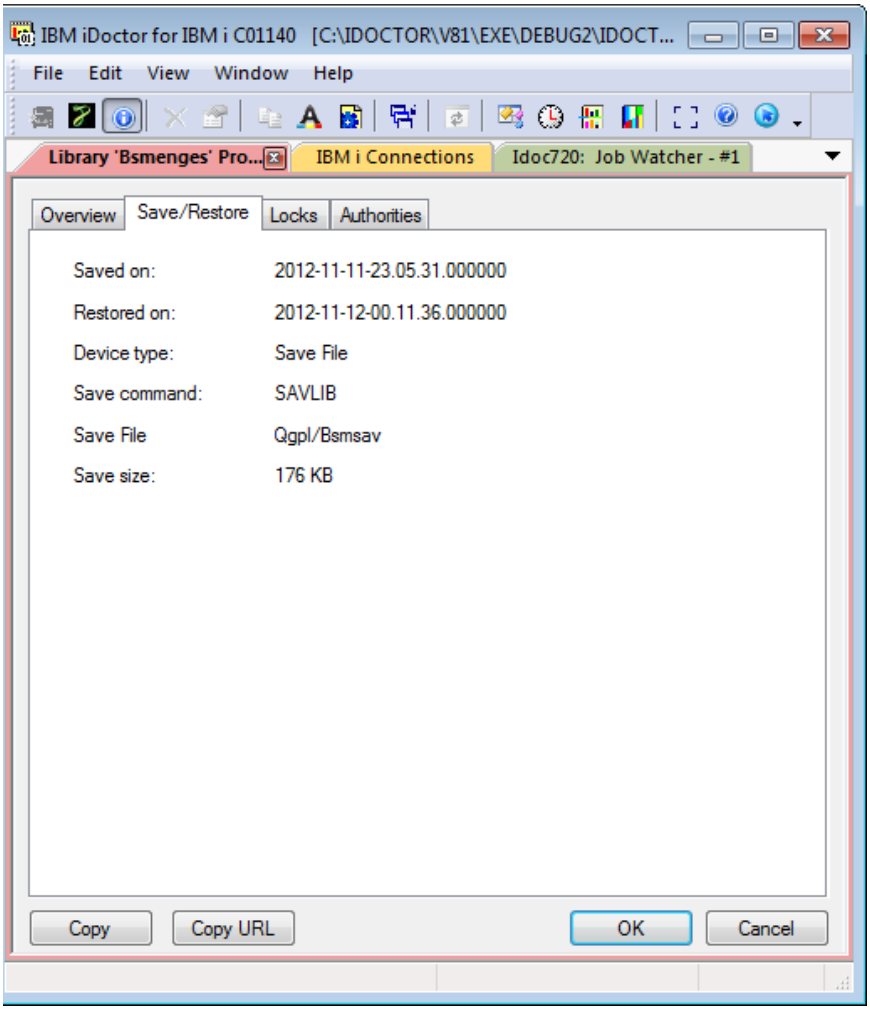
The following information is listed on this page:

Option	Description
Library Name	Name of the library.
Type	Indicates the libraries type. **PROD** The library is a production library. Database files in production libraries cannot be opened for updating if a user is in debug mode and requested that production libraries be protected. **TEST** The library is a test library. All objects in a test library can be updated during test. See the STRDBG command for more details.
Owner	The name of the user profile which owns the library.
Create Authority	The default public authority used when an object is created into a library. This authority is given to the following users: - Users who do not have specific authority to the object. - Users who are not on the authorization list. - Users whose user group has no specific authority to the object. The valid values are: ***ALL** The user can perform all authorized operations on an object created in this library. ***CHANGE** The user can read the object description and has read, add, update, and delete authority to an object created in this library. ***EXCLUDE** The user is prevented from accessing an object created in this library. ***SYSVAL** The default authority for an object created in this library is determined by the value specified by the QCRTAUT system value. ***USE** The user can read the object and its description but cannot change them for an object created in this library. **Authorization list name** The name of the authorization list that secures an object created in this library. The default public authority is taken from the authorization list, and the public authority for the object is specified as *AUTL.
Create Object Auditing	The auditing value for objects created in this library. The valid values are: ***ALL** All change or read access to the object is logged. ***CHANGE** All change access to the object by all users is logged. ***NONE** Use or change access to the object is not logged (no audit entry is sent to the security journal). ***SYSVAL**

	The value specified in the system value QCRTOBJAUD is used. ***USRPRF** The user profile of the user who accesses the object is used to determine if an audit record is sent for this access. The OBJAUD parameter of the Change User Auditing (CHGUSRAUD) command is used to turn auditing on for a specific user.
Description	Library description. You can change this value if you wish.
Created On	The date and time the library was created.
Created By	The name of the user who created the library and the system it was created on.
Object Domain	The domain that contains the object. The possible values are user domain or system domain.
Changed On	The date and time the library was changed.
Total Size	Total size of all objects in the library including the library itself. Click the **Calculate** button to compute this value. **Note:** This calculation can take a long time (minutes or worse) depending on the number of objects and members in the library. The GUI will appear to freeze during this time as well.
Object Count	Total number of objects in the library.
ASP	Auxillary Storage Pool: A number indicating the identifier of the auxiliary storage pool from which storage space for the library was allocated.
Overflowed	Indicates if the object has overflowed the auxiliary storage pool it resides in.
Freed	Indicates the storage status of the object (Yes/No). If the storage status is freed, then the object is suspended, otherwise the object is not suspended.
Object Compression	Indicates the compression status of the object.

4.10.10.2 Library Properties – Save/Restore

The save/restore property page displays information about how and when the library was last saved or restored.



The following information is listed on this page:

Option	Description
Saved On	The date and time the library was last saved.
Restored On	The date and time the library was last restored.
Device Type	The type of the device to which the library was last saved. Valid values are: **Blank** The library was not saved. **Diskette** The library was saved to diskette. **Optical** The library was saved to optical. **Save file** The library was saved to a save file. **Tape** The library was saved to tape.
Save Command	The command used to save the object.
Label	The file label used when the object was saved. This value is not shown if the library was not saved to tape, diskette, or optical. The value of this field corresponds to the value specified for the LABEL or OPTFILE parameter on the command used to save the object.
Save File	Displays the library and name of the save file.
Volume ID	The tape, diskette, or optical volumes that are used for saving the library.
Save Size	Displays the size of the save file.
Save Sequence Number	The tape sequence number assigned when the library was saved on tape. If the library was not saved to tape, this value is not displayed.

4.10.10.3 Library Properties – Locks

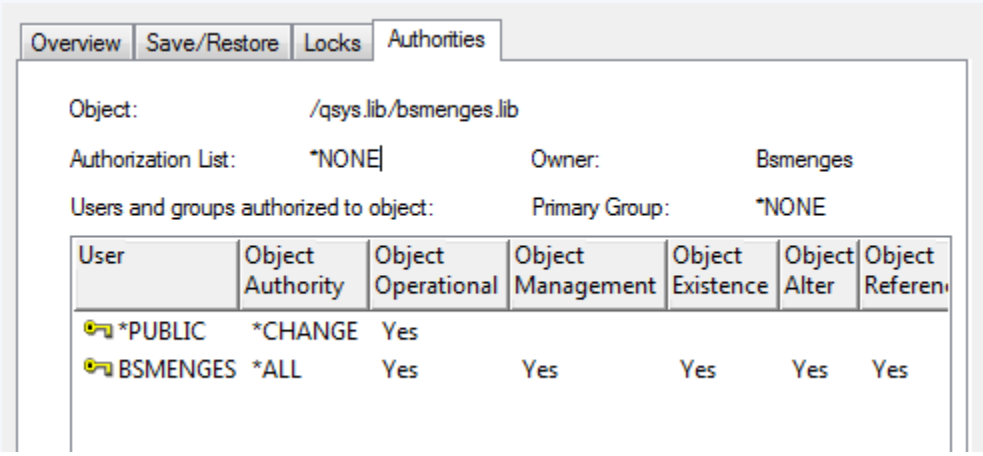
The locks property page for libraries provides an interface similar to the Work Object Lock (WRKOBJLCK) command. This page will tell you which jobs (if any) have a lock on the library.

The following information is shown for each job in the list.

Option	Description
Job	The simple job name of the job that issued the lock request.
User	The user name of the job that issued the lock request.
Number	The number of the job that issued the lock request.
Lock	The lock condition for the request. The possible values are: ***SHRRD** Lock shared for read. ***SHRUPD** Lock shared for update. ***SHRNUP** Lock shared no update. ***EXCLRD** Lock exclusive allow read. ***EXCL** Lock exclusive no read. ***NONE** Lock entry has a null value and is used to select display of lower-level locks.
Status	The status of the lock. The possible values are: **HELD** The lock is currently held by the job. **WAIT** The job is waiting for the lock. **REQ** The job has a lock request outstanding for the object.
Scope	Specifies whether the lock is scoped to the job or scoped to the thread.
Thread ID	Specifies the thread that is associated with the lock. If a held lock is job scoped, the field is blank. If a held lock is thread scoped, the field contains the identifier for the thread holding the lock. If the lock is requested, but not yet available, this field contains the identifier of the thread requesting the lock.

4.10.10.4 Library Properties – Authorities

The Authorities property page shows a list of users that have authority to the library and the users' authorities. This interface is similar to the DSPOBJAUT command.



The following information is shown on this page:

Option	Description
Object	The name of the object for which information is being displayed.
Authorization List	The name of the authorization list that is used to secure the named object. The value, *NONE, indicates that no authorization list is used in determining authority to the object.
Owner	The name of the user profile which owns the library.
Primary Group	The name of the user profile that is the primary group for the library. The primary group can be changed using the Change Object Primary Group (CHGOBJPGP) command.
User List	Displays each user authorized to the library and their detailed authorities to it.

The following information is shown for each user in the list.

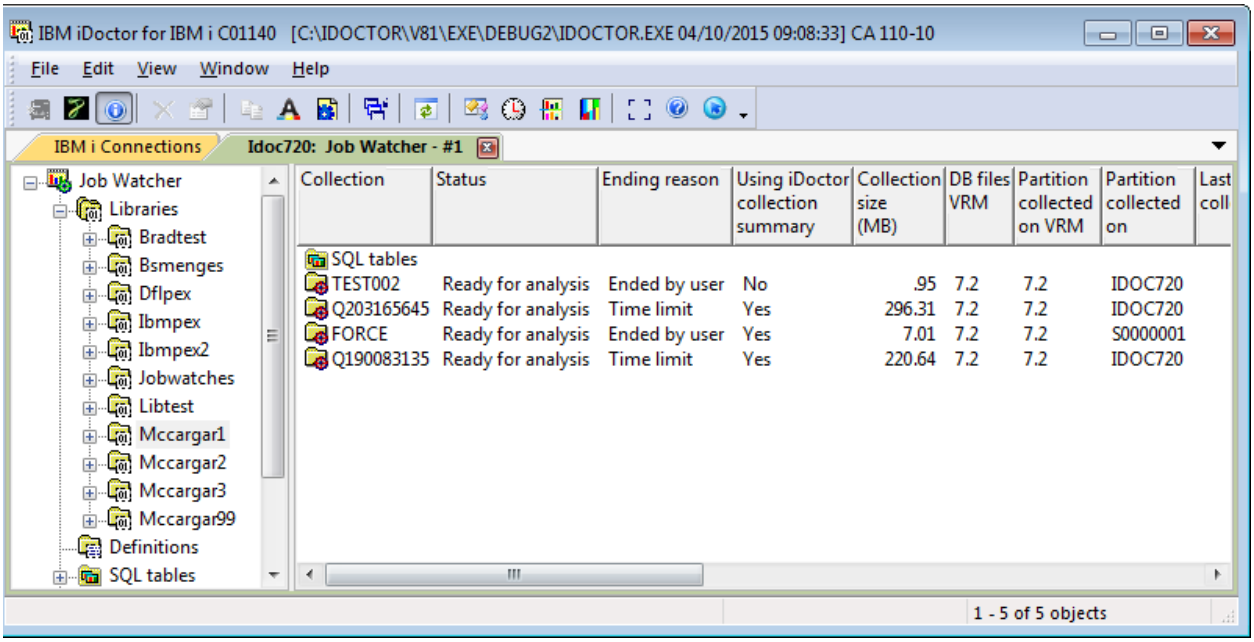
Option	Description
User	The names of users who are authorized to use the object. The value *PUBLIC is used to indicate the authorities of users who are not specifically named and are not in the object's authorization list.
Group	A group from which the user receives authority.
Obj Authority	The user's authority to the object. This field contains one of the following values: ***ALL** The user has all object (operational, management, existence, alter, and reference) and data (read, add, update, delete, and execute) authorities to the object. ***CHANGE** The user has object operational and all data authorities to the object. ***USE** The user has object operational and data read and execute authorities to the object. ***EXCLUDE** The user has none of the object or data authorities to the object, or authorization list management authority to the authorization list. ***AUTL** The public authority for the object comes from the public authority on the authorization list securing the object. This value can only be returned if there is an authorization list securing the object and the authorized user is *PUBLIC. **USER DEF** The user has some combination of object and data authorities that do not relate to a special value. The individual authorities for the user should be checked to determine what authority the user has to the object.
Obj Opr	Object operational authority provides authority to look at the object's attributes and to use the object as specified by the data authorities that the user has to the object.
Obj Mgmt	Object management authority provides authority to specify security, to move or rename the object, and to add members if the object is a database file.
Obj Exist	Object existence authority provides authority to control the object's existence and ownership.
Obj Alter	Object alter authority provides authority to change the attributes of an object, such as adding or removing triggers for a database file.
Obj Ref	Object reference authority provides authority to specify the object as the first level in a referential constraint.
Data Read	Read authority provides authority to access the contents of the object.
Data Add	Add authority provides authority to add entries to the object.
Data Update	Update authority provides authority to change the content of existing entries in the object.
Data Delete	Delete authority provides authority to remove entries from the object.
Data Execute	Execute authority provides authority to run a program or search a library or directory.

4.11 Collections

This section describes interface options for collections which are available for any of the components.

Collections exist under a Monitor, Library or within the Browse Collections interface in iDoctor.

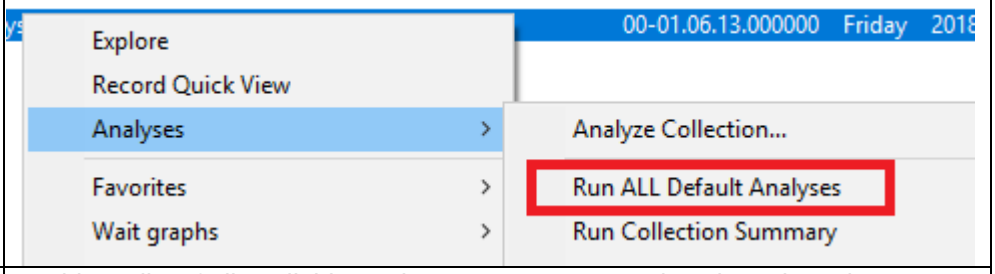
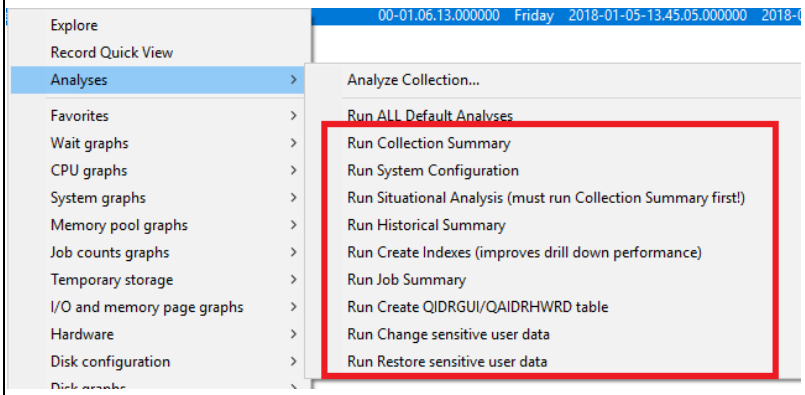
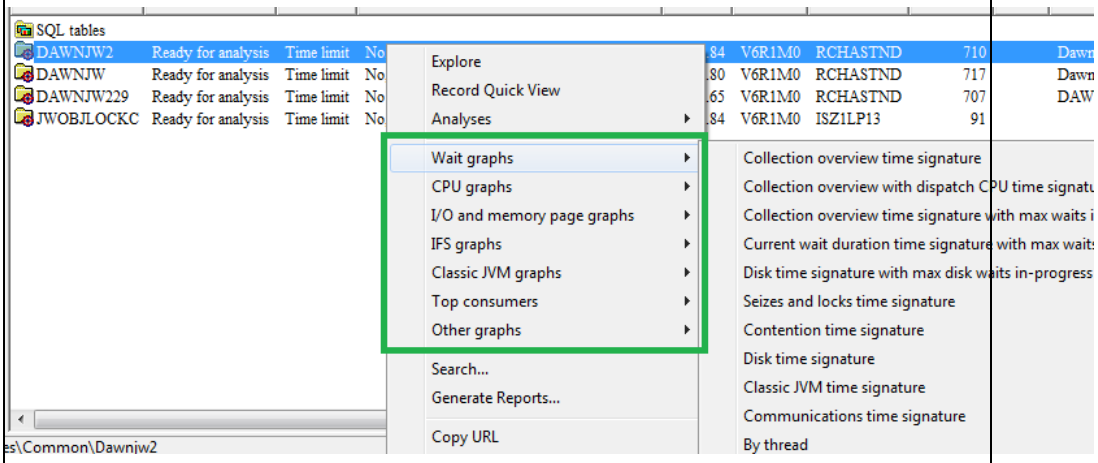
Below is an example of a list of collections in a library within Job Watcher:



4 collections in library in the Job Watcher component

4.11.1 Menu Options

A collection in iDoctor has the following menu options available (right-click):

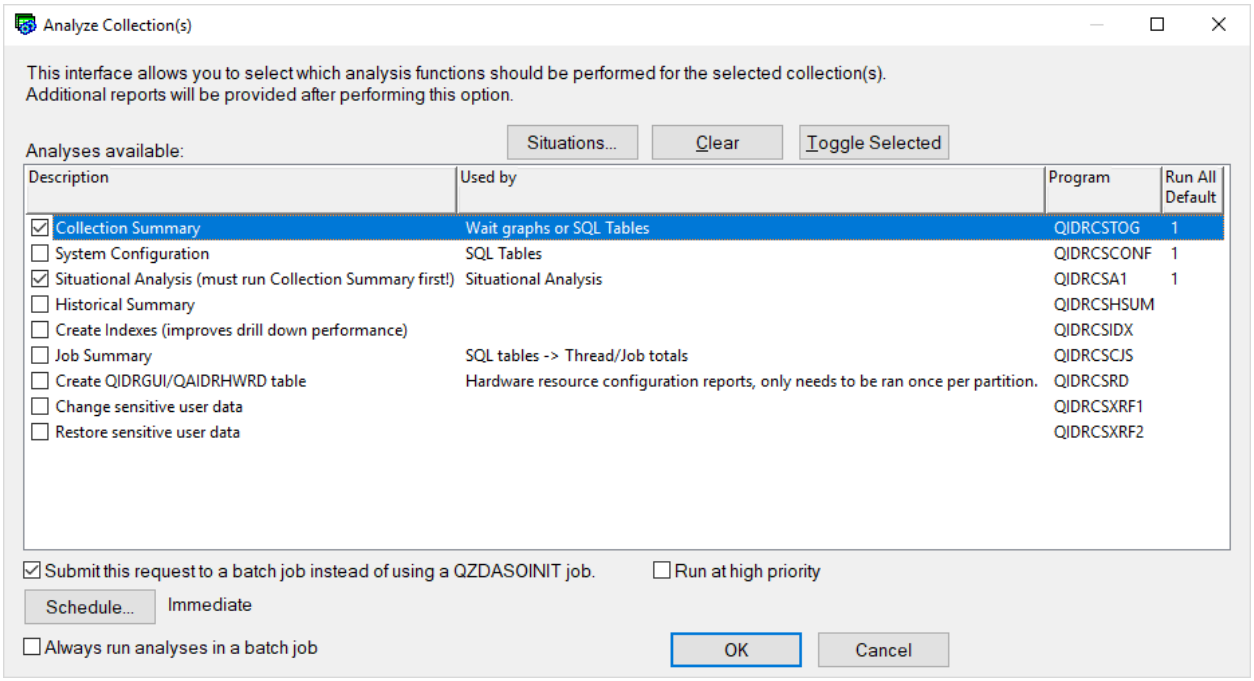
Menu	Description
Explore	Show the contents of the collection.
Record Quick View	Lists the information about the selected collection(s) vertically in a new window.
Analyses -> Analyze Collection…	This interface lets you see more details about each analysis and decide which ones to run on your collection. It also provides options for editing which situations to run when using the Situational Analysis in the CSI and JW components.
Analyses -> Run ALL default analyses	In some of the components this will run the "default" set of analyses on the selected collections. This list of "default" analyses varies by component and can be viewed by using the previous option "Analyses -> Analyze Collection" and looking for the "Run All Default" column in the list of analyses. 
Analyses -> Run analysis	Provides a list of all available analyses you can run against the selected collection(s.) By picking the desired one the analysis will run immediately (unless a prompt is required.)  **Tip:** If you wish these to run in batch job rather than a QZDASOINIT job use the Preference -> Miscellaneous -> Always run analysis in a batch job.
Report menu options	Most collections will provide menu options that allow a user to open graphs or tables by simply right-clicking the desired collection(s) and picking a report. 

	Note: You can also select multiple collections by using the Ctrl or Shift keys on the keyboard along with your mouse clicks and this allows you to graph multiple collections at once for many reports in iDoctor. If it's available, you will be prompted and asked if you wish to combine the collections into a single graph / report or not.
Graph Job(s)	Use this option to graph one or more jobs over time in Job Watcher or Collection Services Investigator.
Search	Some components allow a search capability. This typically will give you different types of data to look and a list of results that match the search. From the search results you will be able to drill down to retrieve more detail.
Generate Reports…	This option can be used to build a report of the desired set of graphs or reports. The report consists of a screenshot of each graph along with its title and collection information. The reports are built into a HTML page and displayed in a web browser when completed.
Copy URL	Creates a link to the component, library and collection that can be accessed later, or sent to another user.
Copy...	Allows you to copy the collection(s) to another location.
Delete	Deletes the selected collection(s).
Rename	Rename the selected collection.
Save...	This option lets you save the collection(s) into a save file on the server.
Split…	Divides a large collection into 1 or more smaller collections based on time filtering you specify. This option is only available in the Job Watcher and PEX Analyzer components.
Transfer to...	Allows a user to create a save file of the selected collection(s) and transfer it to another system, the PC or to IBM.
Properties	Displays the property pages for the collection.

4.11.2 Analyze Collection(s) Window

Use this option to be presented with a list of possible analyses to run against the selected collections. You can select one or more analyses to run by checking the checkbox next to each. In the Job Watcher and Collection Services Investigator components, a Situations button will exist that allows you to configure options related to the Situational Analysis.

If you want the analyses to run in a batch job instead of a QZDASOINIT job, then check the checkbox called "Submit this request to a batch job…". Doing so is usually desired if the collections have large amounts of data or if the analyses are expected to take a long time to run as this will free up the Remote SQL Statement Status View for other actions you may wish to take.

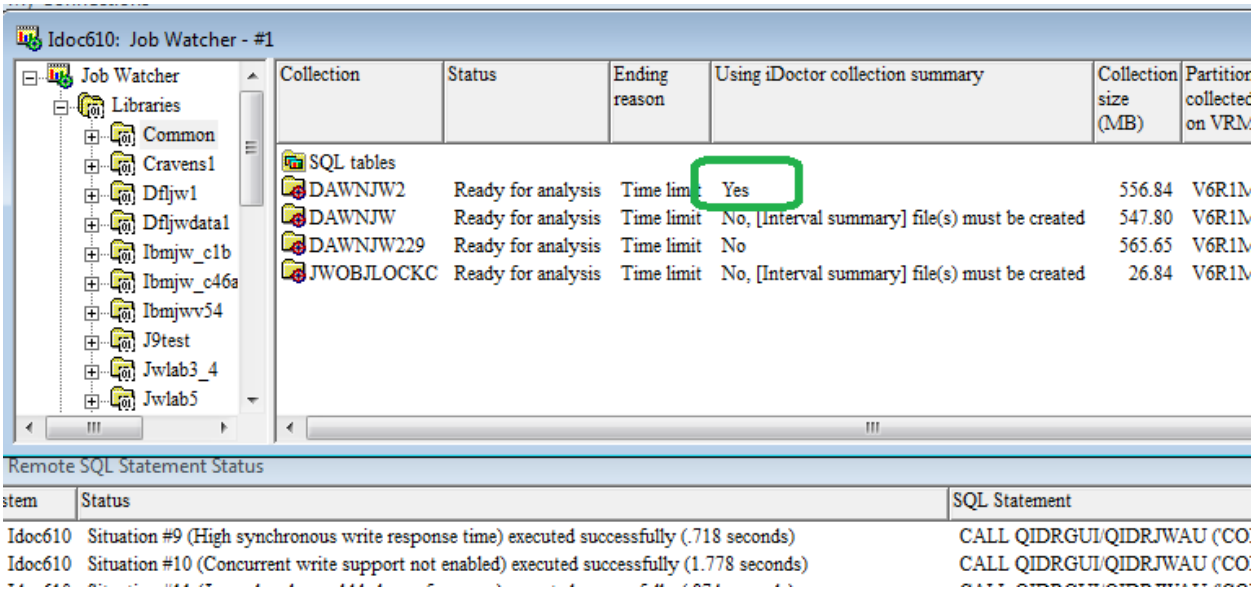


Analyze Collections Window

Option	Description
Situations…	This button opens the interface to work with the situations found in the Job Watcher or Collection Services Investigator components.
Clear	This button will uncheck all selections in the list.
Toggle selected	The selected analyses will either be checked/unchecked (i.e. toggled) from their current values.
Analyses available	This is the list of possible analyses to run. It contains these columns: **Description:** The name of the analysis. Typically, under the SQL Tables folder you will find folder of the same name which allows you to work with the data generated by these analyses. **Used by:** This describes where in the GUI the data created by this analysis is used. **Program:** The name of the program object in the QIDRGUI library. This is the SQL stored procedure that behind each analysis. These programs all have a text description which is a number such 005. This is the version number of the program. **Run all default:** If this value is set to a 1 then the analysis will be executed when using the Run ALL default analyses menu option found when right-clicking a collection.
Submit this request to a batch job instead of using a QZDASOINIT job	This option will run the desired analyses in a batch job on the IBM i using a file sent to the IBM i's IFS with the SQL statements to run and the RUNSQLSTM command. The job name used will be one of the following depending on the component identifier: Collection Services: QIDRCSSUM Job Watcher: QIDRJWSUM PEX Analyzer: QIDRPASUM Etc. **Note:** This option requires a valid FTP connection to the IBM i. Your FTP connection settings can be configured in the IBM i Connections View.
Run at high priority	This will submit the batch job to run the SQL statements at priority 1. **Note:** You probably do not want to use this option on production systems as in some cases the SQL can be intensive.
Schedule	This button is used to specify in the analysis should run now or later.
Always run analyses in a batch job	This option effects the same preference on the Misc tab. If checked then the next time this screen is used the "Submit this request to a batch job…" will be prechecked for you.

After pressing OK, the Remote SQL Statement Status View window will show calls to several stored procedures that are used to create the analysis tables. These tables will appear under the SQL Tables folder under the library and collection.

After the analyses are complete it's important to refresh (F5) the library to ensure that all new data is recognized by the GUI and it used in the reports shown. If a Collection Summary analysis was ran the "Using iDoctor collection summary" flag may also change from "No" to "Yes." This will typically cause additional reports to shown as well.

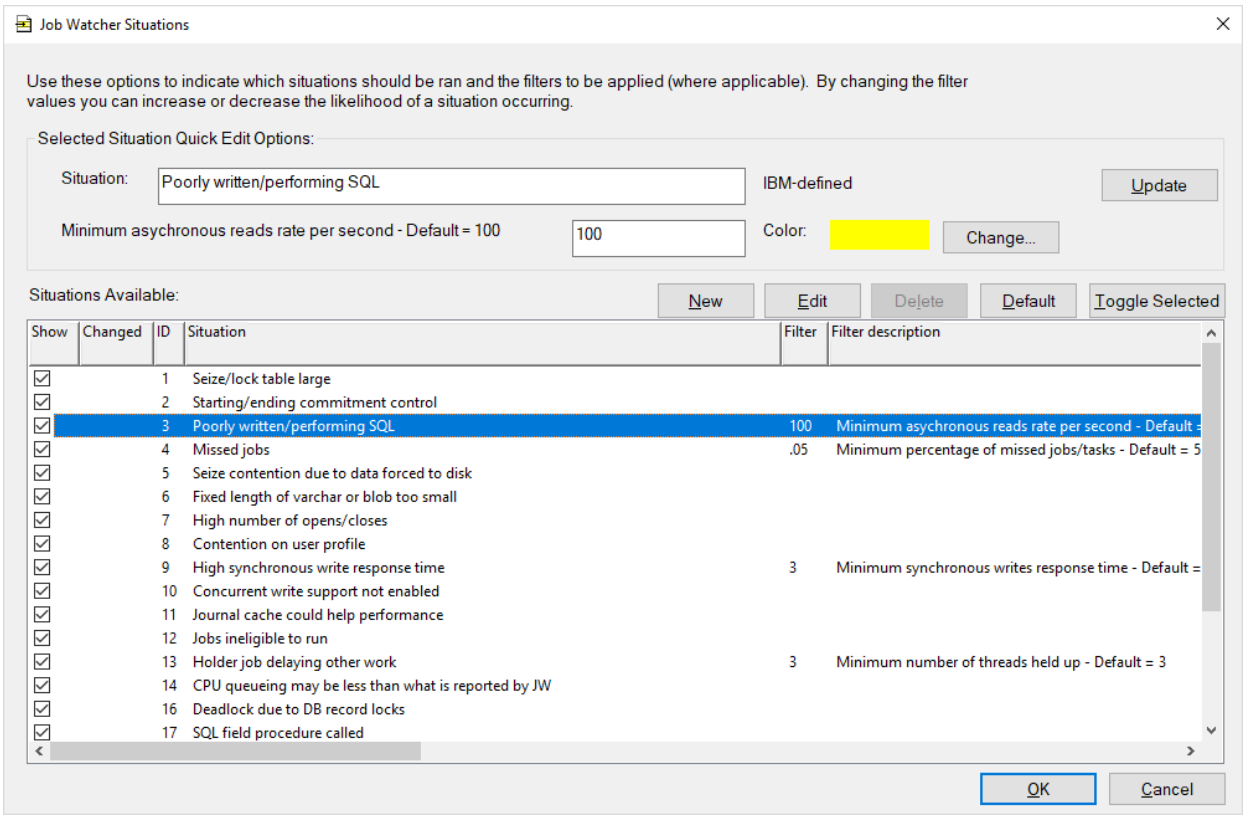


Using iDoctor collection summary flag changed to Yes after analyses complete

4.11.2.1 Situations Window

The Situations Window allows the user to control parameters used by the IBM defined situations or you may also create new user-defined situations to run against the data in your collections. Using this interface, you may also control which situations should be ran.

An example of this interface is:



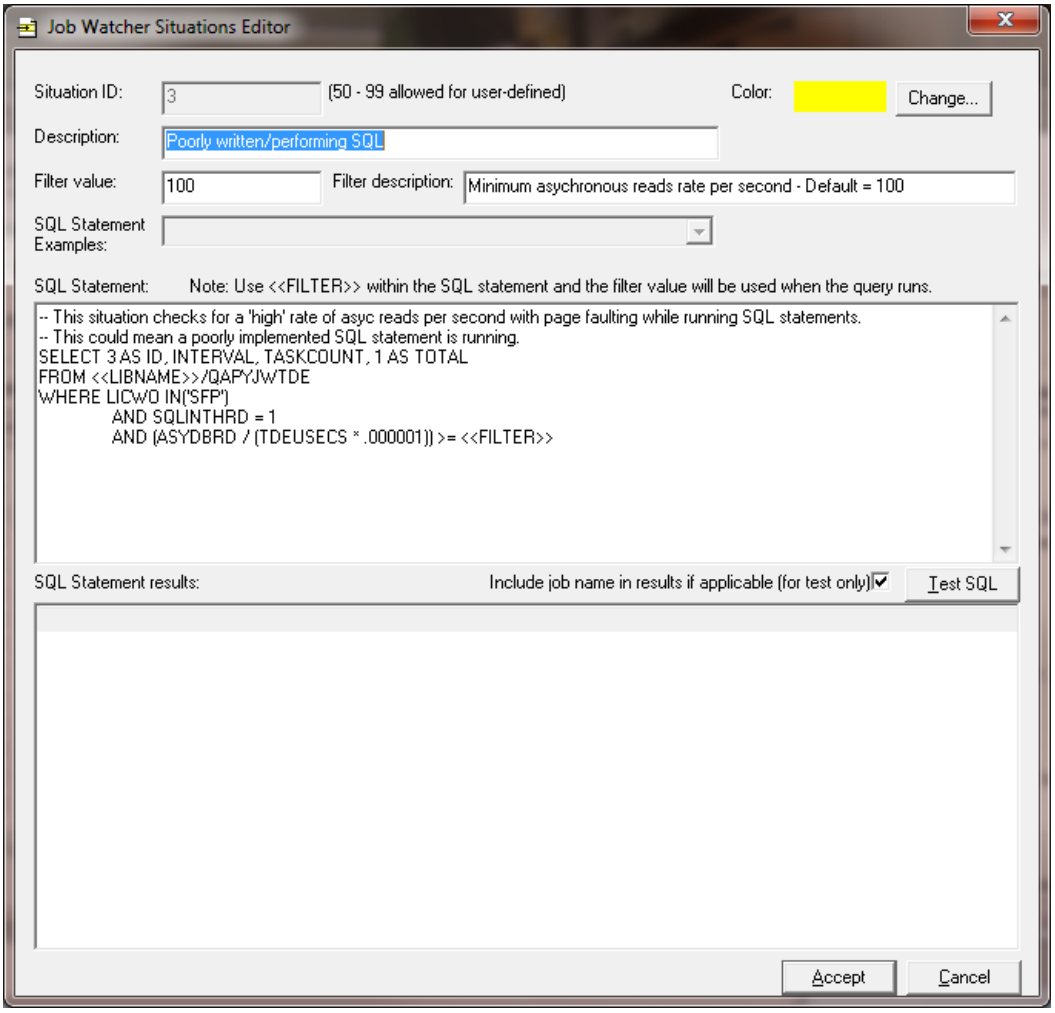
The options on the interface above is described in the table below:

Option	Description
Selected situation quick edit options	This area contains controls that allow you to modify some of the attributes for the selected situation such as name, filter value and color.
Situation text box	This field allows the user to modify the name of the situation.
Update button	This button will save any changes made within the Selected situation quick edit options frame to the selected situation in the list. These changes are saved in the windows registry.
Filter value text box	The filter value text box lets you modify the filter's value to use. The filter value replaces the <<FITLER>> parameter marker within the SQL statement.
Color change button	Changes the situations color shown as the background color when graphed. If multiple situations occur in a time period, then the color is always red.
Situations available list	This list contains all the IBM-defined and user-defined situations. The following columns are provided: **Show:** The show checkbox/column can be used to avoid running certain situations if desired. **Changed:** If the user has made changes to a situation this column will contain “Yes”. **ID:** The ID number is used to uniquely identify each situation. **Situation:** Name of the situation. **Filter:** Some situations have a <<FILTER>> parameter in the SQL statement. This is the value to use for that parameter and helps indicate if the situation should occur. **Filter description:** Describes the (optional) filter used by this situation. **Color**: Identifies the color of the situation in RGB format (0-255,0-255,0-255). **IBM-defined:** This column indicates if the situation is IBM-defined or user-defined. Typically, the situation ID will be >=50 for user-defined situations. **SQL:** Shows the SQL statement for the situation. It may be modified by pressing the Edit button.
New button	The new button displays the Situations Editor window which allows you to create your own situation.
Edit button	The edit button displays the Situations Editor window and fills in the information for the current situation.
Delete button	This button lets you delete the currently selected user-defined situations. IBM-defined situations cannot be removed.
Default button	This button removes all changes made to the IBM-defined situations, remove all user-defined situations and restores them to their original (shipped-default) state.
Toggle selected button	This button changes the checked state of all selected items in the list.

4.11.2.2 Situations Editor Window

The Situations Editor window is used to create a new situation or modify advanced settings of an existing one. It allows the user to modify a situation to suit their individual needs. Situations are built from a special SQL statement that meets certain characteristics.

An example of this window looks like this:



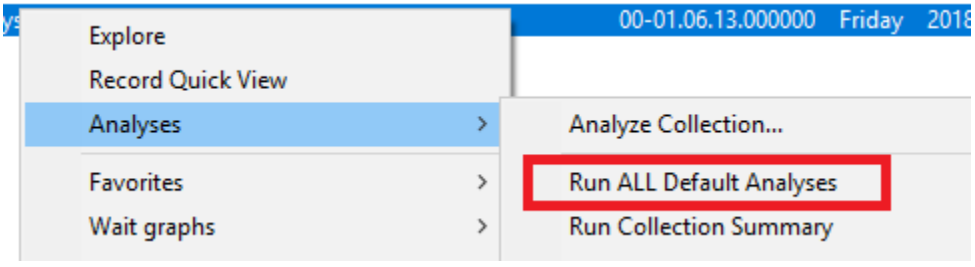
Job Watcher Situations Editor

Option	Description
Situation ID	The situation ID must be unique and needs to be between 50-99 for user-defined situations.
Color change button	Changes the situations color shown as the background color when graphed. If multiple situations occur in a time period, then the color is always red.
Description text box	This field allows the user to modify the name of the situation.
Filter value text box	The filter value text box lets you modify the filter's value to use. The filter value replaces the <<FITLER>> parameter marker within the SQL statement. **Note:** This field is optional and can be blank.
Filter description	The filter description describes the filter that has been included in the SQL statement. **Note:** This field is optional and can be blank.
SQL Statement examples	The SQL statement examples drop down box contains a list of all IBM-defined situations. Selecting one of these situations replaces the current SQL statement shown. These examples contain comments and should help you get a better idea on how to create your own situation to suit your needs.

SQL Statement	The SQL statement that performs the testing to see if the situation has been satisfied in the data. The tables should be referred to using <<LIBNAME>>/QAPYJWTDE syntax where <<LIBNAME>> is a parameter marker replaced at runtime with the current library and QAPYJWTDE is the desired file. **Note:** Aliases will be automatically created for you to point to the current collection member and do not need to be referred to here. The outer select must contain the following 4 fields (in this order): 1. ID = situation ID (User-defined situations are numbered 50+. IBM-defined situations are 1-49.) 2. INTERVAL = interval number when the situation occurred 3. TASKCOUNT = unique identifier for the job/task. Use a value of 0 if the situation applies to the entire collection and not a specific job or task. 4. TOTAL = The number of occurrences of this situation for this job/task and interval. If the situation does not apply to any specific job or task, then a value of 1 should be used.
Include job name in results	This option will modify the SQL statement slightly under the covers to display the Job name and thread ID associated with each taskcount found. Because Job name and thread ID are not returned in the situation analysis table they are only shown here for test purposes.
Test SQL	This button executes the current SQL statement shown above against the current collection. If any results are found, they will be shown in the SQL Statement results list. For testing purposes, use the Test SQL button and the “Include job name in results checkbox” to see the jobs in your test collection that match your situation before using. In this way you can modify the SQL Statement to control verbosity to best suit your needs.
SQL Statement results	This list contains the result set returned by running the SQL Statement shown above.
Accept button	Accepts all changes made and closes the window, returning to the Situations Window.

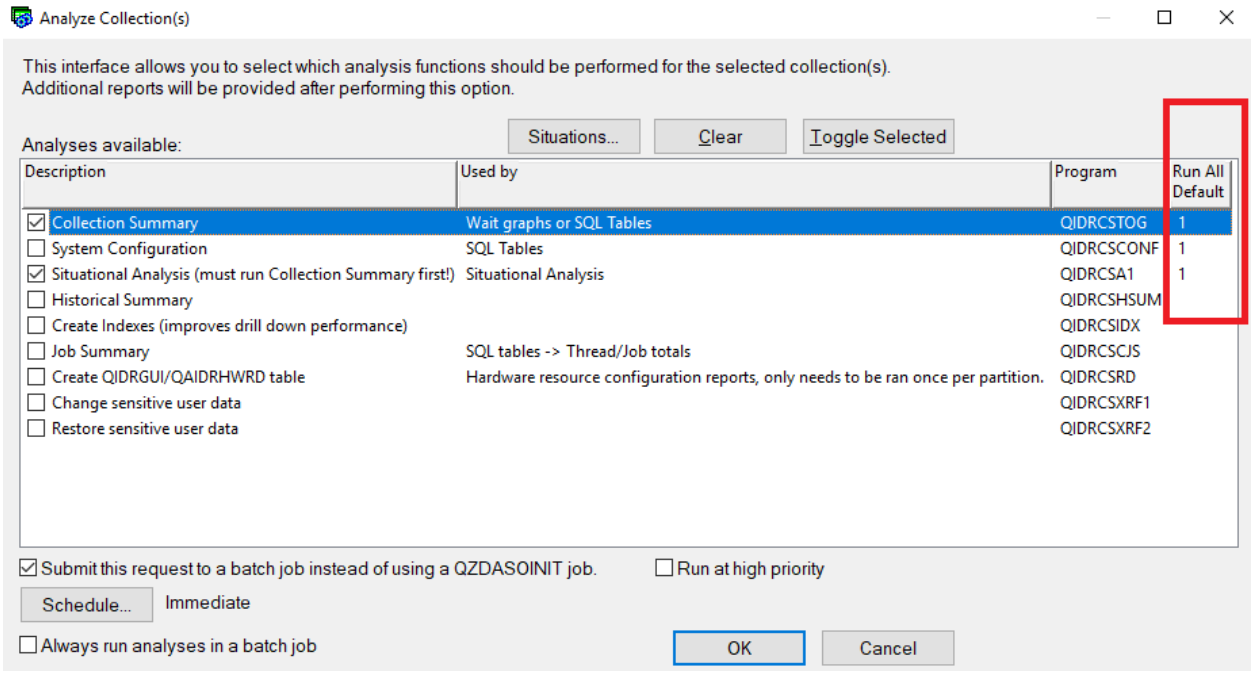
4.11.3 Run ALL Default Analyses

In some of the components this will run the “default” set of analyses on the selected collections.



Collection menu -> Analyses -> Run ALL Default Analyses

This list of “default” analyses varies by component and can be viewed by using the menu option “Analyses -> Analyze Collection” and looking for the “Run All Default” column in the list of analyses.



Analyze Collection(s) Window -> Run All Default column example

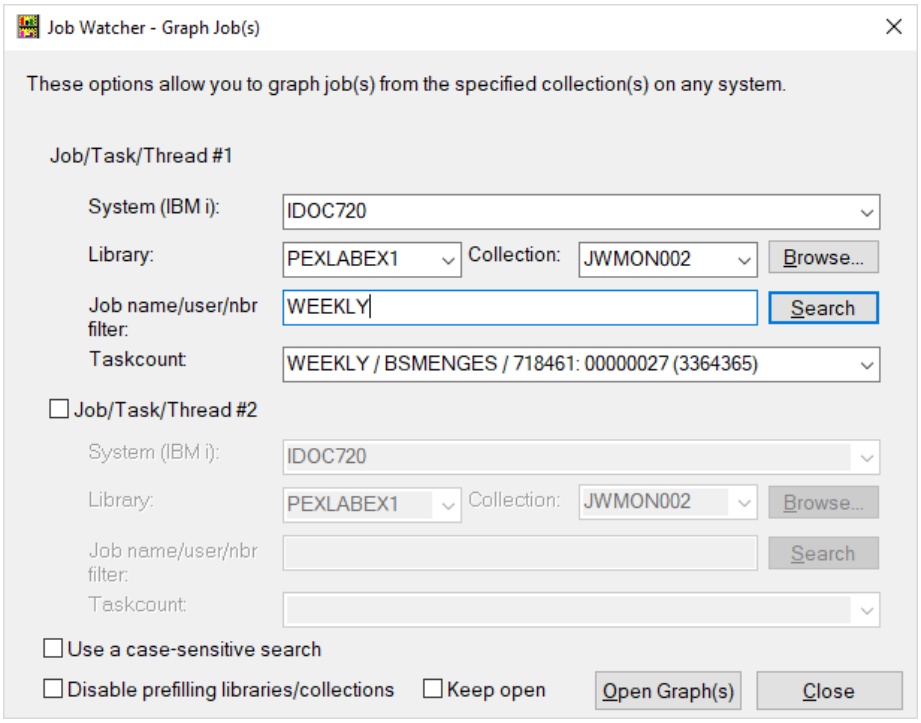
4.11.4 Analyses -> Run analysis menu

This option (by default) when used on a collection, will kick off the desired analysis in the Remote SQL Statement Status View. If the submit to batch preference is used instead, then you must wait until the batch job created finishes.

When complete additional reports will become available (after refreshing the component view).

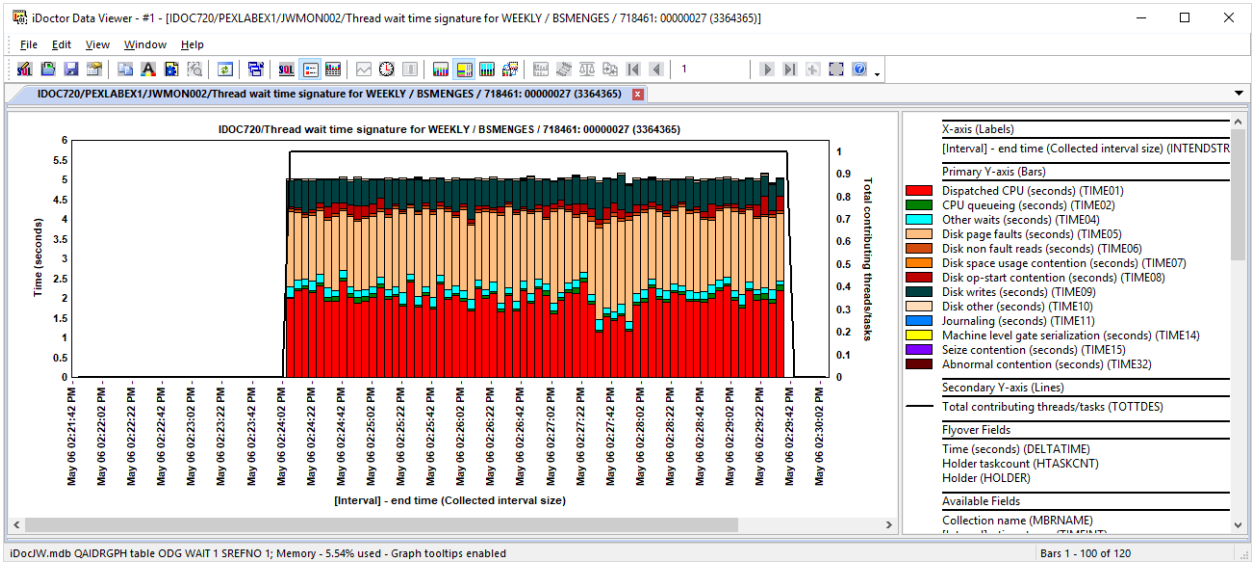
4.11.5 Graph Job(s)

This interface is found only in the Job Watcher and Collection Services Investigator and is used to graph the desired job in any collection over time. The wait bucket time signature graph for a single job/thread is always shown by this interface. This option also allows a user to graph and compare 1 job with another job on the same system or any system and collection they wish.



Graph Job(s) Window

Option	Description
System (IBM i)	The name of the system the collection is located on you wish to graph.
Library	The library of the collection.
Collection	The name of the collection to graph.
Job name/user/nbr filter	This field allows you to enter part of the job name. When pressing the Search button all matches will be shown. **Note:** If you press the Search button without entering anything in the list, then all jobs in the collection will be returned which could be extremely slow if there are thousands of them so it's best to filter by something first!
Taskcount	The taskcount is the unique identifier of the job and/or task. This is the value in parentheses returned after performing a search. **Note:** If you know the actual taskcount you can enter it here instead of performing a search by name.
Use a case-sensitive search	This option will need to be used in some cases where you need to find certain system tasks which have lower-case characters in their names. All job names use all upper-case names and a case-sensitive search is unnecessary.
Disable prefilling libraries/collections	If you are experiencing slowdowns on this interface when trying to select the library and collection using the drop-down list, you can check this box so the GUI no longer tries to determine the available data. **Tip:** If you have very large numbers of collections and/or libraries with data on your system, then this might be required.
Keep open	Check this box if you wish to keep this interface open after pressing the Open Graph(s) button. This will let you open several different graphs at once into a Data Viewer more easily before reviewing them.
Open Graph(s)	Opens the desired graphs into a Data Viewer. There are preferences in the Preferences -> Data Viewer tab that can be used to configure which Data Viewer these graphs appear into. **Note:** If the Keep open checkbox is NOT checked, then this window is closed after pressing this button.



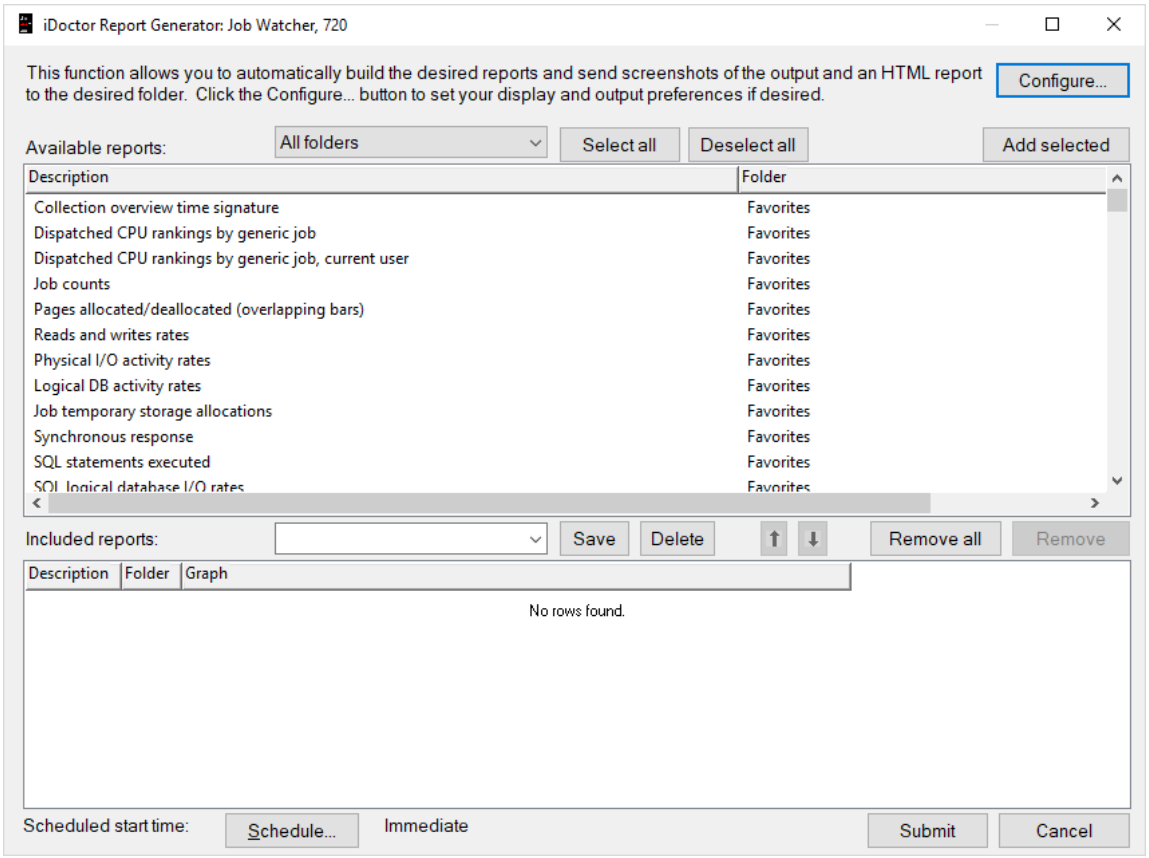
Graph Job(s) Example graph

4.11.6 iDoctor Report Generator

This interface allows a user to generate an HTML report for the desired graphs or tables. The user can select the reports of interest, and the GUI will automatically open each report, capture a screenshot and then build an HTML page showing all the reports in a single window for review. This feature is intended to allow a user to easily save a series of graphs for comparison with other collections or for a consultant to present their findings to a client.

In most components this feature is accessible by right-clicking a collection and choosing the Generate Reports… menu option.

An example of this interface is:



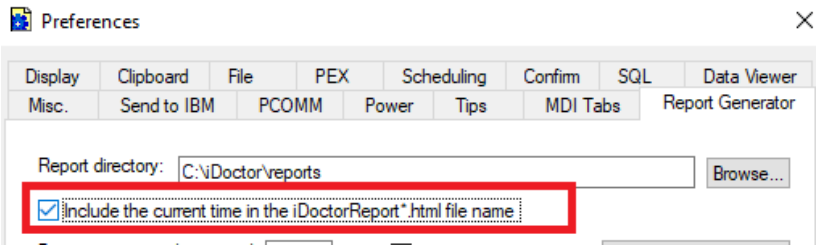
iDoctor Report Generator

To use this interface, select the desired reports from the list of available reports and press the “Add Selected” button to add them to the “Included reports” list. To save the list of included reports for future use, enter a name in the drop-down box next to the Save list button and press the Save button.

The Configure button at the top of the screen will take you to the Preferences for the Report Generator window. These preferences control various aspect to how the reports generated will look.

After pressing Submit, each table or graph will be opened into the Data Viewer capturing a screenshot for each one. If you wish to cancel this process, close the Data Viewer while the Report Generator is running. After all reports are loaded an HTML file is built to show all the screenshots captured into a web page. This page will be opened with the default web browser installed on the PC. Depending on how many reports are selected this could take a long time. You can also use the Schedule button at the bottom of the window to have the graphs/reports be opened later.

The following table describes the options available on the iDoctor Report Generator window:

Option	Description
Configure	This button opens the Preferences for the iDoctor Report Generator. This controls where the reports are generated and how they will look.
Folders drop-down	This drop-down list provides a list of all primary folders that appear under a collection. By selecting one of these you will be able to filter the list of available reports to a smaller number.
Select all	This button will select all reports in the Available reports list.
Deselect all	This button will unselect all reports in the Available reports list.
Add selected	Adds the selected reports from the Available reports list to the Included reports list.
Available reports list	The list of reports available. The list of reports will vary based on the folder selected, the component being used and the VRM of the collection.
Included reports list	This is the list of reports to be opened by the Report Generator.
Saved reports drop down list	This is a list of saved lists of reports that are available. Selecting a name from the list will update the list of reports selected to the ones indicated in the list. You can define a new list by typing a name into the drop-down box and pressing the Save button. These lists are saved into the User-defined reports database which allows them to be reused by other users. **Note:** Saved report lists you created are also visible under the Favorites folder (under a collection) in Job Watcher and Collection Services Investigator.
Save	Saves the current selection of reports to the name given in the drop-down list.
Delete	Press this button to remove the saved list of reports shown in the drop-down list.
Remove all	Removes all reports from the included reports list.
Remove	Removes the selected reports from the Included reports list.
Schedule	This option allows you to schedule the reports to be generated later using the Windows task scheduler. Options are available to have this occur repeatedly. If scheduling multiple runs then you should use the preference on the Report Generator tab to include the current time in the html output file name to avoid having the data be overwritten each time the process runs.  The PC must be on and the current user must be still logged in at the indicated date/time for this to work.
Use newest collection	This option appears only when the scheduling option is used. If this option is checked then when the report is generated it will use the newest collection in the library rather than the one used to launch the Report Generator interface originally.
Submit	This will either schedule the generation of reports or kick it off right now. Each report will be opened in the Data Viewer one at a time and a screenshot captured. A final report is opened in the default web browser when the process finishes.

4.11.7 Copy URL

Use this option to copy and paste a URL into an email or instant messaging program to allow another user to access the same interface currently being viewed. Using this option from a collection, will bring up the iDoctor GUI on the PC and open the component, library and collection that was being viewed when this option was taken.

The URL generated by this option starts with idoctor:// and tells your web browser to launch iDoctor and perform the desired action.

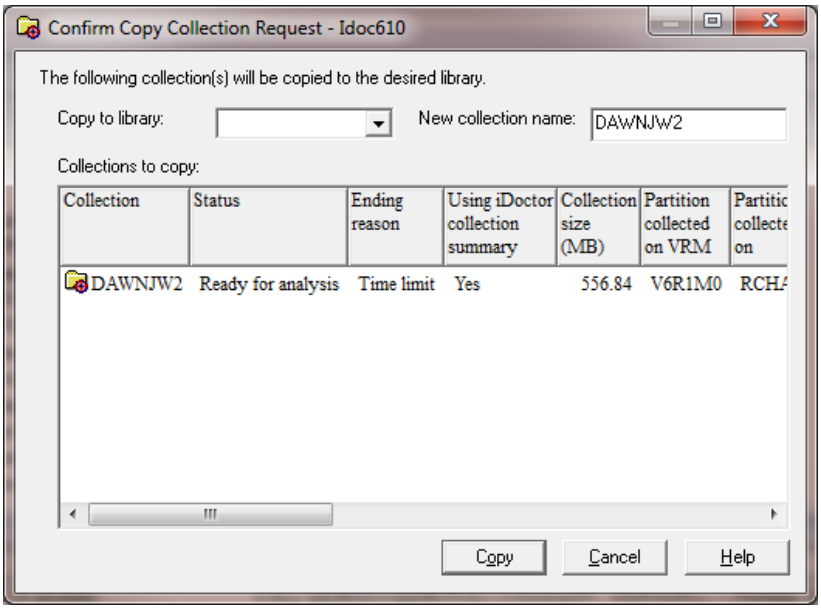
Note: This feature works with the latest versions of Edge (IE), Firefox or Chrome (as of this writing 10/19/2018.)

4.11.8 Copy

A collection can be copied by using the Copy... menu found by right clicking on a collection within the component view.

This option will execute the appropriate iDoctor collection copy command depending on the type of collection selected. Copying a collection that is still running is not allowed. Multiple collections can be copied at the same time if desired to another library.

Note: Be sure to only copy a collection to a library with existing performance data of the same type if the collection match the same IBM i release.



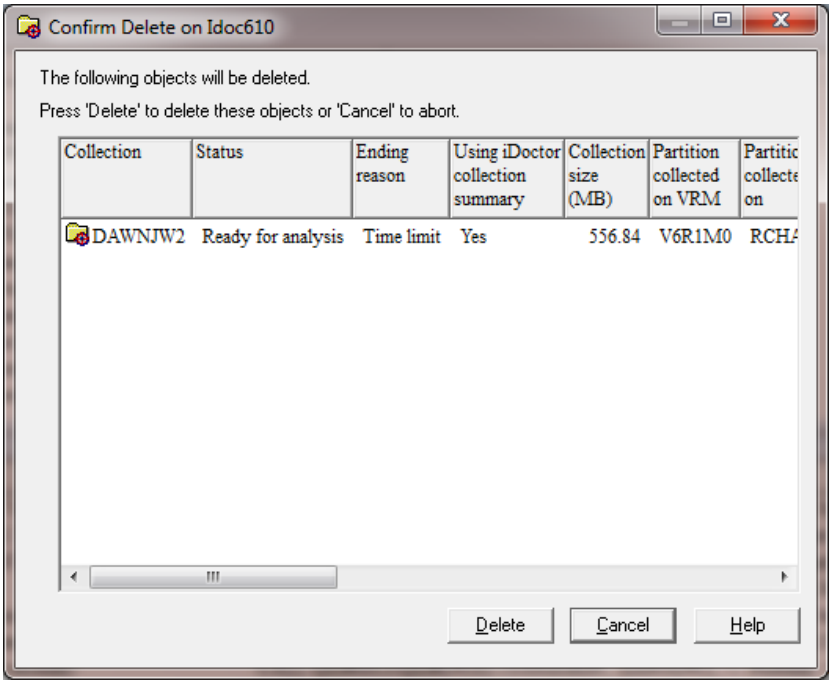
Confirm Copy Collection Request Window

Option	Description
Copy to library	The library to copy the collections to. Either type in the library name or click the arrow to display a list of all libraries on the system.
New collection name	This option allows you to copy a collection to a new name in the specified library. It will only be visible if a single collection has been selected.
Collections to copy	The list of collections to be copied to the library specified.

4.11.9 Delete

A collection can be deleted by using the Delete... menu found by right clicking on a collection within the component view.

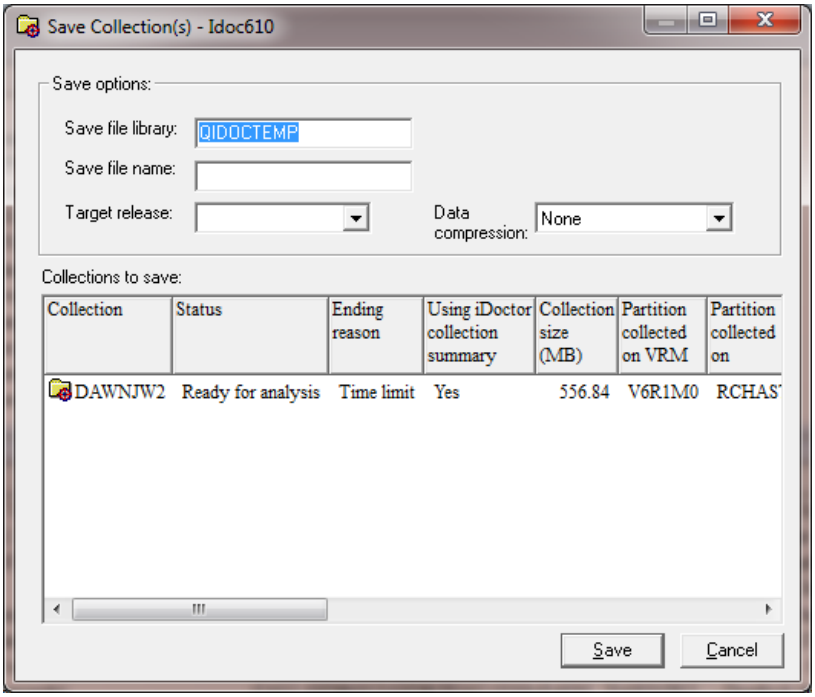
This option will execute the appropriate iDoctor delete collection command depending on the type of collection selected. This option is not allowed if the collection is running.



Confirm Delete Window

4.11.10 Save

One or more collections can be saved by using the Save... menu found by right-clicking the desired collection(s.)



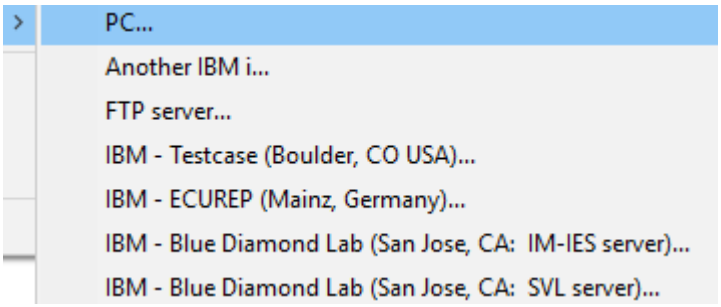
Save Collection(s) Window

Option	Description
Save file library	The library the save file will reside in.
Save file name	The save file name.
Target release	The OS VRM of the system you intend to restore this save file on.
Data compression	The type of compression to use when performing the save.
Collections to save list	The list of collections or monitors to save to the specified save file.

4.11.11 Transfer to…

A collection can be transferred to the PC or another system by using the Transfer to... menu found by right clicking on a collection within the component view. This option is only available for collections that are no longer running.

The transfer options for collections consists of the following choices:



Collection Transfer to menu in Job Watcher

Picking one of these will take you to a Transfer window with this selection premade.

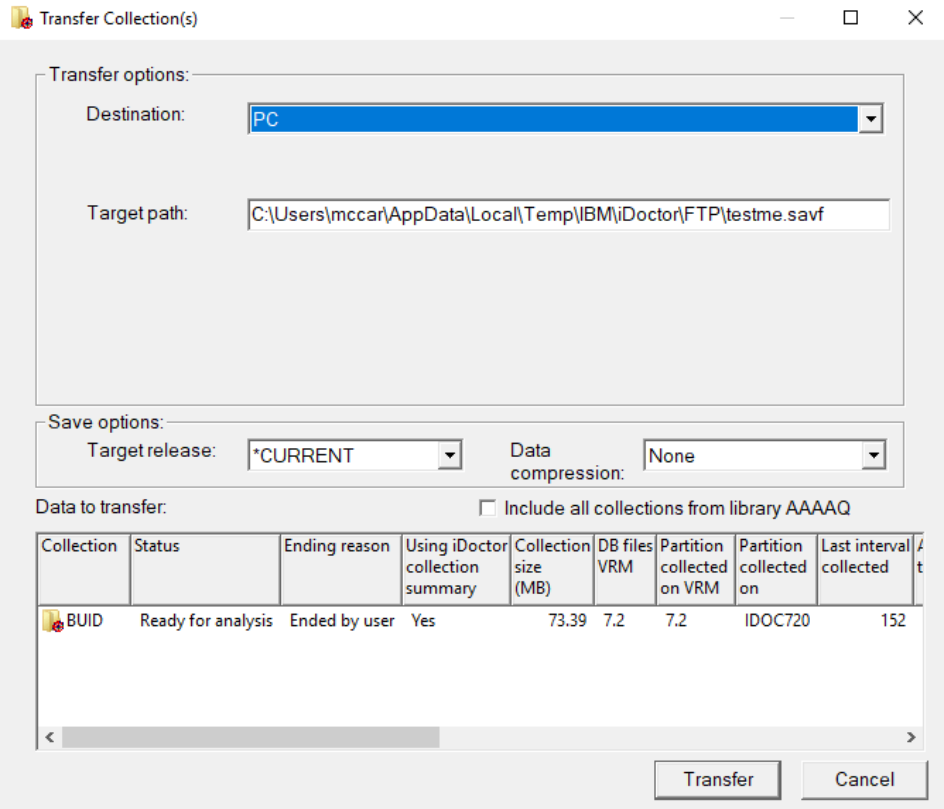
After reviewing the Transfer window and pressing the Transfer button, a validation step takes place to help detect any problems before continuing. Any errors that occur will be shown to the user. Afterwards commands will be issued on the local and remote servers to save, send and optionally restore the data. These commands will be shown in the Remote Command Status View.

In the Remote Command Status View, if an error occurs during the FTP part you can right-click the failing part and use the Show Job Log or Show History Log options. Another option for debugging is to open an SQL Editor (either in iDoctor or STRSQL) and issue the following query:

SELECT * FROM QIDRGUI/FTPLOG

4.11.11.1 Transfer to PC

This option will transfer one or more collections to the specified location on the current PC.

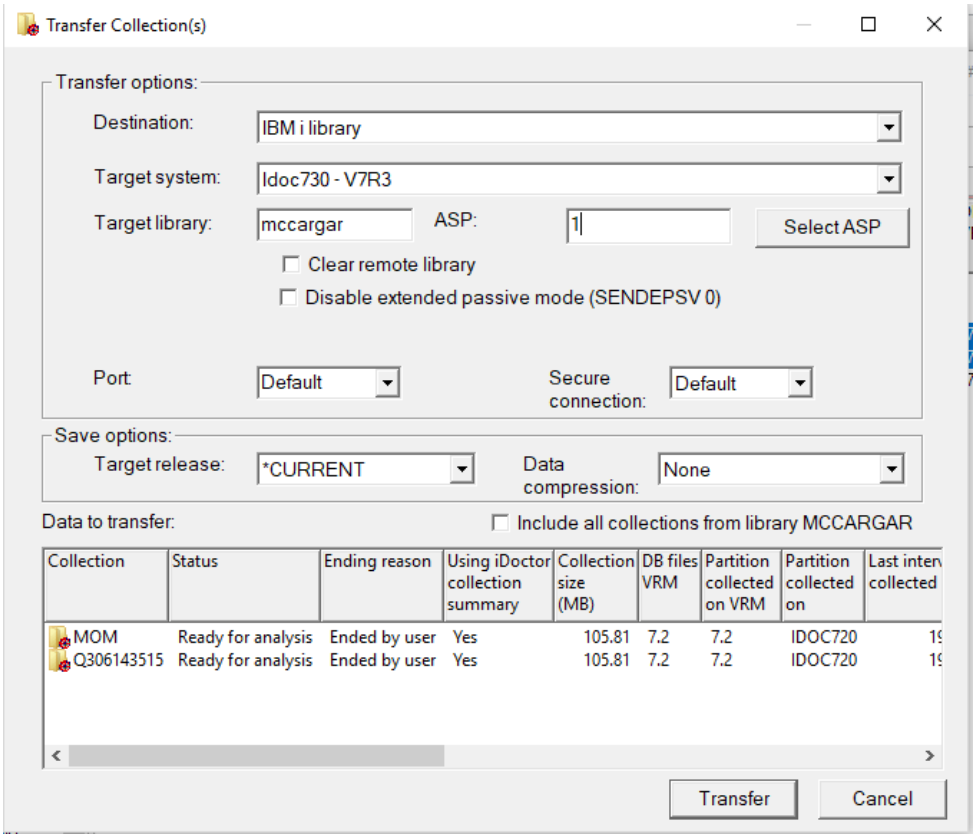


Transfer Collections Window (to PC)

Option	Description
Destination	This indicates which location you will be sending data to.
Target path	The directory and filename on the PC to send the collection(s). Multiple collections are saved into a single save file.
Target release	The IBM i release of the system the collection will be restored to.
Data compression	Whether or not the save file should use data compression. If slow network connections are used this is recommended to reduce FTP transfer times.
Include all collections from library XYZ	This option lets you include ALL collections from the library of the collection you opened this option from.
Data to library	This is the list of collections to transfer

4.11.11.2 Transfer to another IBM i

This option allows you to transfer one or more collections to another IBM i and have them automatically restored.

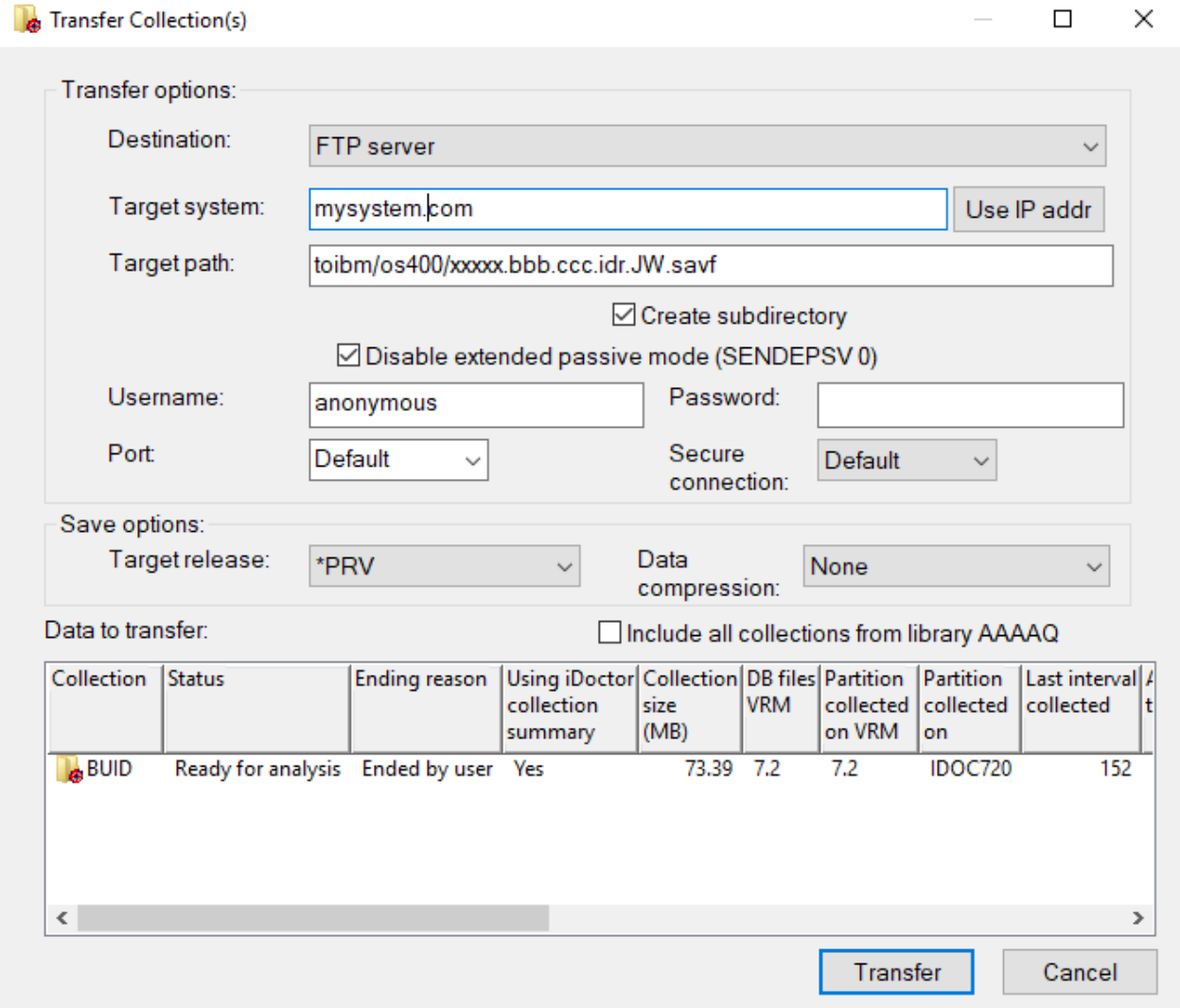


Transfer Collections Window (to another IBM i)

Option	Description
Destination	This indicates where you will be sending data to.
Target system	The IBM i to save and restore the collection(s) to.
Target library	The library on the target system to restore the collection(s) to.
ASP	The ASP number to use when creating the target library (if the library does not exist.)
Select ASP	This shows a list of ASPs on the target system to select from.
Clear remote library	This will cause the specified target library to be cleared before the transfer occurs. In some cases, this is necessary if the target library contains data from a previous IBM i release.
Disable extended passive mode	Disables a setting in the FTP transfer of the data. In some environments this is required. Collections are sent directly from the current IBM i to the target system (IBM i). You can read more about this setting here: https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzaiq/rzaiqsendepsv.htm
Port	The FTP port to use for the transfer. (1-65535 are valid) **Default:** 21 **Secure:** 990 **Note:** This parameter is passed down to the PORT parameter on the FTP command on the IBM i.
Secure connection	Specifies the type of security mechanism to be used for protecting information transferred on the FTP connection (which includes the password used to authenticate the session with the FTP server). Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are compatible protocols which use encryption to protect data from being viewed during transmission and verify that data loss or corruption does not occur. **Default**: If the PORT parameter specifies Secure or 990, Implicit is used; otherwise, None is used. **Implicit:** The FTP client immediately attempts to use TLS/SSL when connecting to the specified FTP server (without sending an AUTH subcommand to the server). If the server does not support implicit TLS/SSL on the specified port, or the TLS/SSL negotiation fails for any reason, the connection is closed. **SSL:** After connecting to the specified FTP server, the FTP client sends an AUTH (authorization) subcommand requesting a TLS/SSL protected session. If the server supports TLS/SSL, a TLS/SSL negotiation performed. If the server does not support TLS/SSL or the TLS/SSL negotiation fails, the connection is closed. **None:** The FTP client does not use encryption when connecting to the specified FTP server. **Note:** This parameter is passed down to the SECCNN parameter on the FTP command on the IBM i.
Target release	The IBM i release of the system the collection will be restored to.
Data compression	Whether or not the save file should use data compression. If slow network connections are used this is recommended to reduce FTP transfer times.
Include all collections from library XYZ	This option lets you include ALL collections from the library of the collection you opened this option from.
Data to transfer	This is the list of libraries to transfer

4.11.11.3 Transfer to FTP server

This option allows you to transfer one or more collections to a directory on another system. The collection(s) are combined into a single save file and sent the path/filename specified.



Transfer Collections Window (to FTP server)

Option	Description
Destination	This indicates where you will be sending data to.
Target system	The system to send the collection(s) to.
Use IP addr(ess)	This will change the target system specified to its IP address if the iDoctor GUI is able to determine it. In many situations it is better to transfer by IP address depending on the network/system configuration because the target system name may not be known from the IBM i you are using. If that is the case the connection will fail.
Target path	The location on the system to send the SAVFs to.
Create subdirectory	This option (if checked) will create on the target system any subdirectories required by using the target path specified. Otherwise the transfer will fail if the location does not already exist or the user lacks the ability to create directories.
Disable extended passive mode	Disables a setting in the FTP transfer of the collection(s). In some environments this is required. Collections are sent directly from the current IBM i to the target system. You can read more about this setting here: https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzaiq/rzaiqsendepsv.htm
Username	The user name to connect (via FTP from the current IBM i) to the target system with.
Password	The password to connect (via FTP from the current IBM i) to the target system with.
Port	The FTP port to use for the transfer. (1-65535 are valid) **Default:** 21 **Secure:** 990 **Note:** This parameter is passed down to the PORT parameter on the FTP command on the IBM i.
Secure connection	Specifies the type of security mechanism to be used for protecting information transferred on the FTP connection (which includes the password used to authenticate the session with the FTP server). Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are compatible protocols which use encryption to protect data from being viewed during transmission and verify that data loss or corruption does not occur. **Default**: If the PORT parameter specifies Secure or 990, Implicit is used; otherwise, None is used. **Implicit:** The FTP client immediately attempts to use TLS/SSL when connecting to the specified FTP server (without sending an AUTH subcommand to the server). If the server does not support implicit TLS/SSL on the specified port, or the TLS/SSL negotiation fails for any reason, the connection is closed. **SSL:** After connecting to the specified FTP server, the FTP client sends an AUTH (authorization) subcommand requesting a TLS/SSL protected session. If the server supports TLS/SSL, a TLS/SSL negotiation performed. If the server does not support TLS/SSL or the TLS/SSL negotiation fails, the connection is closed. **None:** The FTP client does not use encryption when connecting to the specified FTP server. **Note:** This parameter is passed down to the SECCNN parameter on the FTP command on the IBM i.
Target release	The IBM i release of the system the collection will be restored to.
Data compression	Whether or not the save file should use data compression. If slow network connections are used this is recommended to reduce FTP transfer times.
Include all	This option lets you include ALL collections from the library of the collection you

collections from library XYZ	opened this option from.
Data to transfer	This is the list of collections to transfer

4.11.11.4 Transfer to IBM – Testcase / Transfer to IBM - ECUREP

These options are used to send your data to IBM for analysis. Typically, you will need a PMR # to associate this data with.

It is possible that in the future these options will require a secure FTP connection, but as of this writing (Oct 19th, 2018) both do not. Keep in mind that transferring data over an unsecure connection could risk your password being compromised.

The options for these modes are the same as the previous section (Transfer to FTP server), except there is also an option to transfer using IBMSDDUU. This is a secure transfer mechanism written in Java that may optionally be used if desired.

4.11.11.5 Transfer to IBM - Blue Diamond Lab

These options are used to send your data to IBM for analysis. You will need a PMR # to associate this data with and a secure FTP connection method.

The same settings are used on this screen as those in the previous section except the Port value defaults to secure when using these options.

4.11.12 Server-side output files

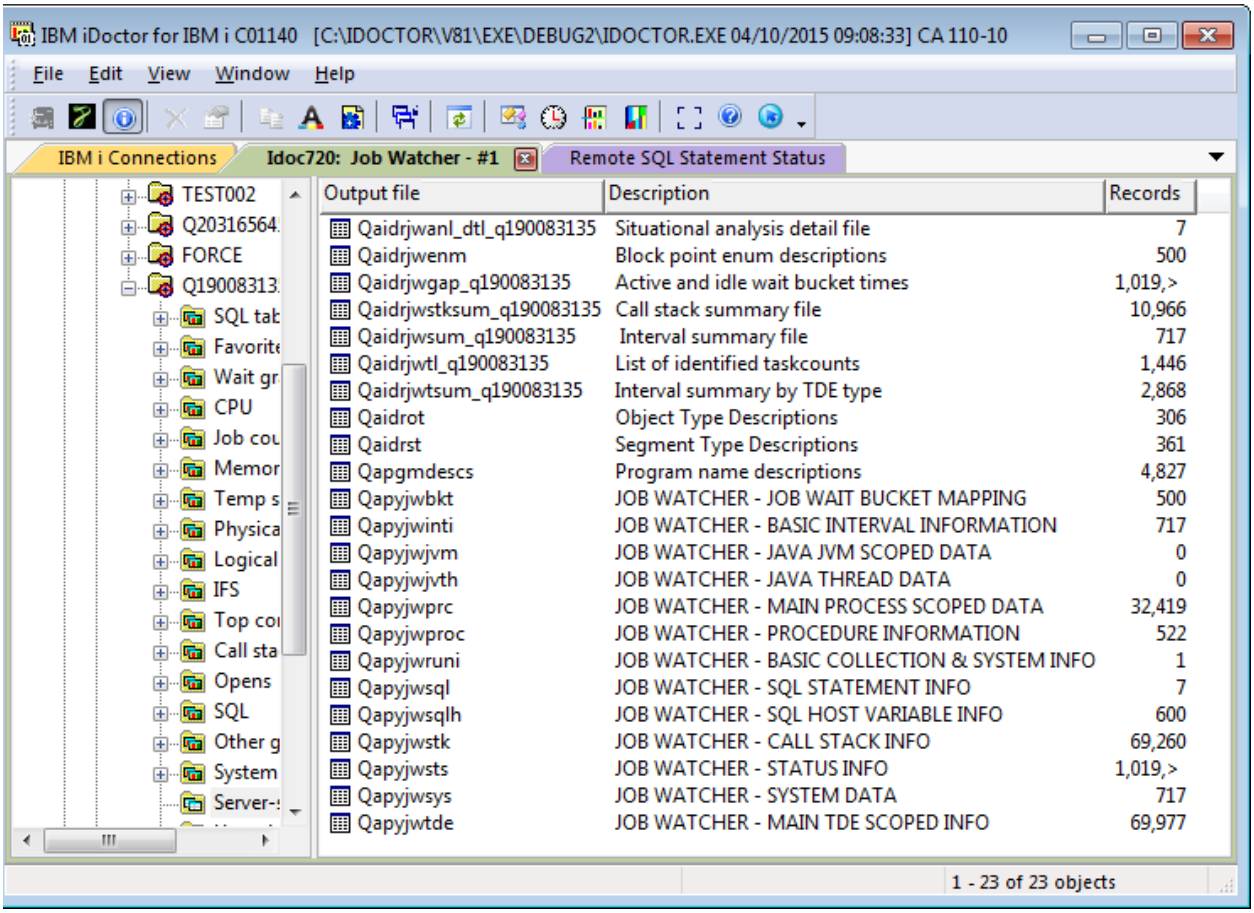
Most collections provide a folder called Server-side output files. This provides access to a list of tables applicable to the current collection. This list contains both iDoctor created files and files created by IBM i performance data collection mechanisms.

Each IBM i physical file name and (long) SQL table name is provided in this view.

You can right-click this folder and use the Select fields… menu to configure the list of fields shown in this list.

Tip: The number of records found in each table is also shown, which if 0 can help indicate a problem in some situations. This view is also used to tell if certain expected tables do not exist (and presumably did not get transferred successfully)

Note: Not all tables are shown in this view. Additional tables that were created by iDoctor analyses are shown under the SQL Tables folder.

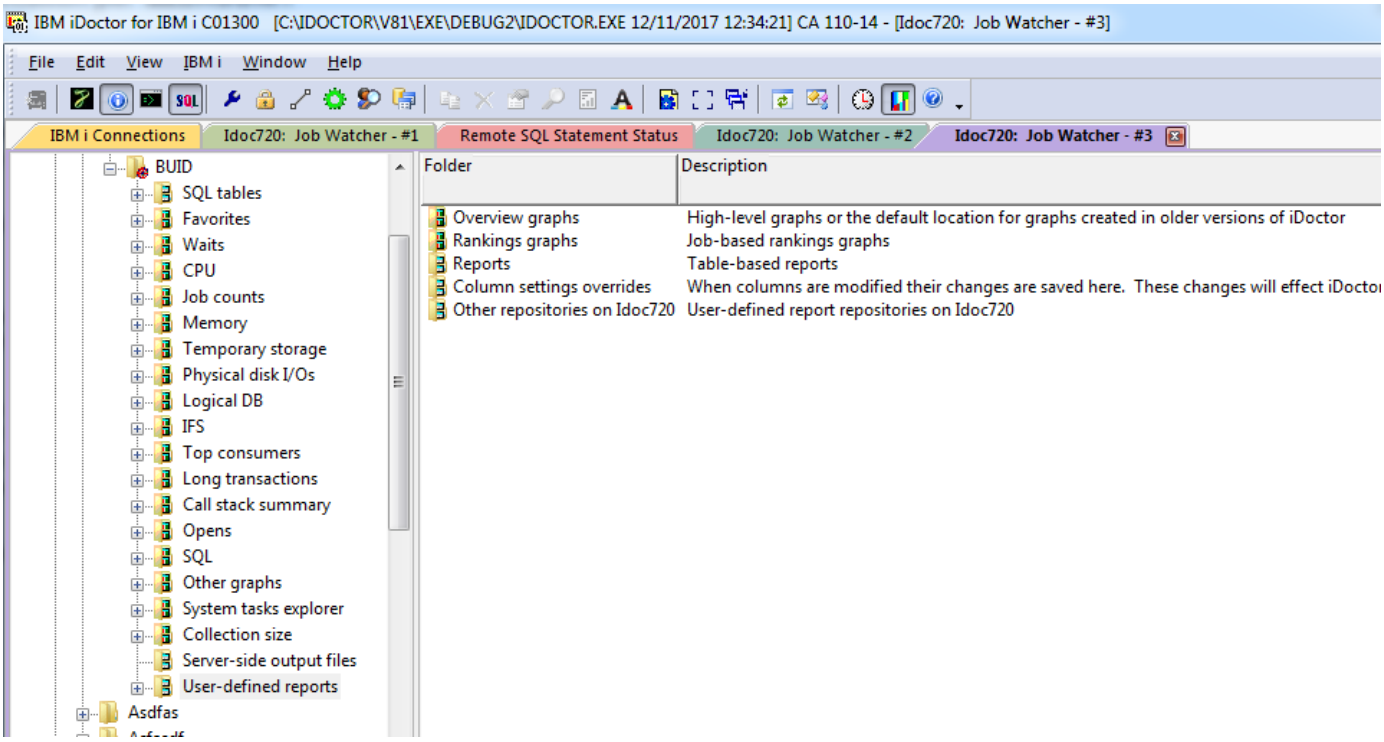


Server-side output files example in Job Watcher

4.11.13 User-Defined Reports folder

When viewing collections in most iDoctor components an option called “User-defined reports” will be shown. This option allows you to show reports from your user-defined reports database but applied to the current collection. This folder contains both graph and table views.

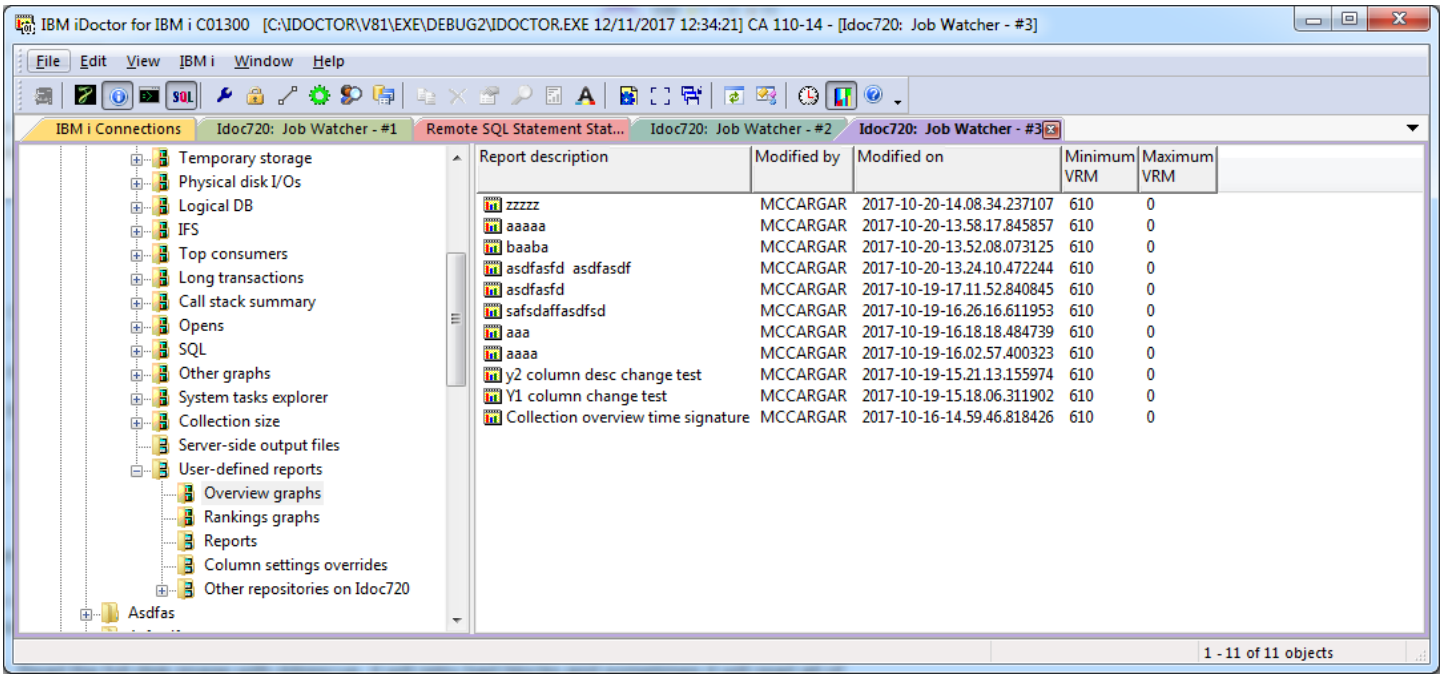
Typically, users will see something like the following:



User-defined reports folder for a Job Watcher collection

4.11.13.1 Overview graphs

This folder typically contains user-defined graphs that are time based. If you had created user-defined graphs in old versions of iDoctor they will be migrated here by default.



Job Watcher collection -> User-defined reports -> Overview graphs

Double-click a graph to open it. You may also right-click the graph and use the Properties option to view and change some of the settings for the graph such as the SQL statement used.

4.11.13.2 Rankings graphs

Use this option to save user-defined graphs that rank data (such as jobs or disks.)

4.11.13.3 Reports

Use this option to store any table views. If you had created user-defined tables in old versions of iDoctor they will be migrated here by default.

Column settings overrides – This folder contains each column that has been modified in iDoctor by any user using the currently defined user-defined reports database. Whenever a user modifies a graph or table column using the Edit column interface they will be saved here. These overrides apply to iDoctor supplied reports and user-defined reports for any where the column short name in the SQL statement matches the values in this list.

If you delete a column from this view, then the iDoctor-defined settings for this column (colors and description) are used instead (assuming the same column name exists in the iDoctor reports.)

Other repositories – This option allows you to work with any other repositories (libraries) found on the current IBM i you are using. Only libraries will appear that contain the iDoctor repository SQL Tables such as QAIDRGPH or QAIDRSQL.

Note: Additional folders can appear based on what the user has defined in the database. When saving a user-defined graph or table you can specify the folder the report appears under and you can create a new sub folder to store it into as well.

4.11.13.4 Graph/Report Menu Options

Each user-defined graph or report in these folders has the following menu options when right-clicked:

Menu	Description
Open Graph / Table	Opens the selected report in a new or existing data viewer depending on the sub-menu option taken.
Edit	Opens the selected report into the SQL Editor. The SQL statement will not be ran until requested by the user. This is most useful if the queries are long running and you wish to modify them before execution.
Delete...	Removes the selected user-defined reports from the user-defined reports database.
Properties	Displays details about the current user-defined graph definition or query definition that can be modified such as the SQL statement or fields shown on the graph.

4.11.13.5 Graph Definition Properties

User-defined graphs have properties pages that let you change several settings with how the graph looks. **Note:** You can also change these same settings by opening the graph and saving the user-defined graph there instead.

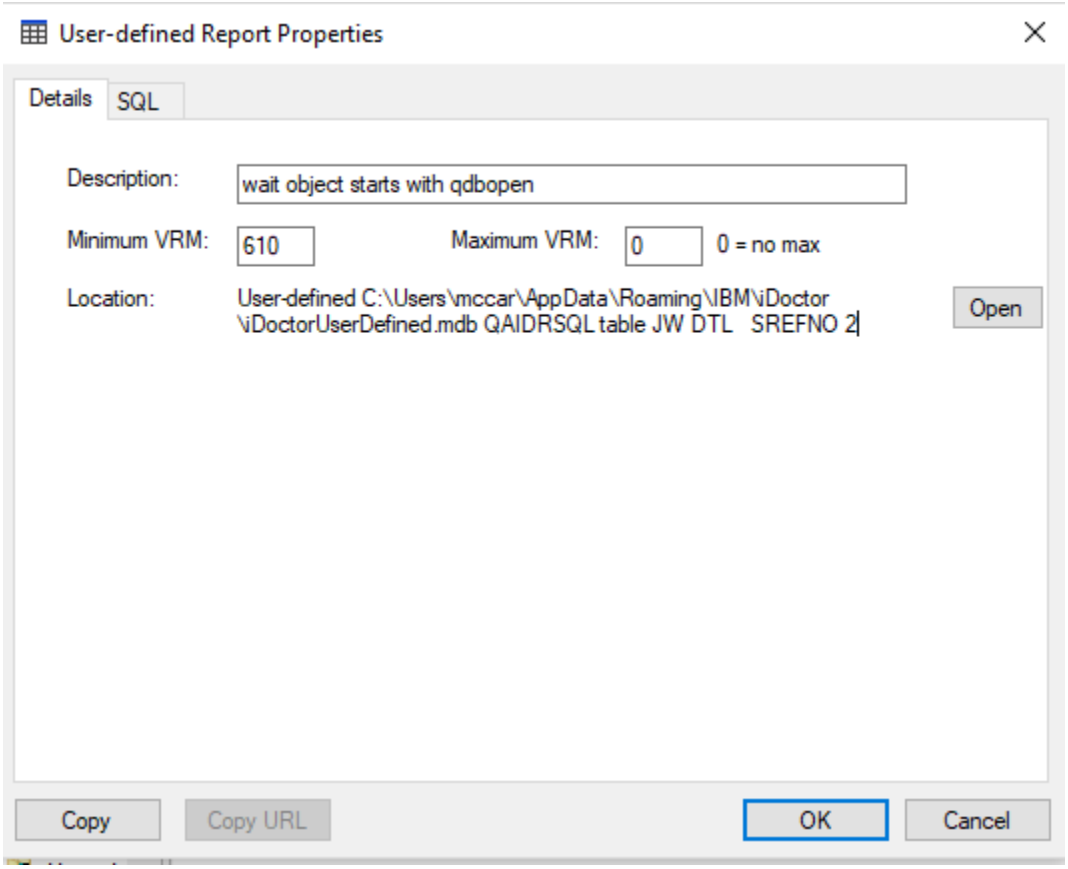
These properties for user-defined graphs are mostly identical to those for iDoctor supplied graph definition properties. The main differences are you can edit user-defined graph definitions and the SQL statement and your changes will be saved.

4.11.13.6 Query Definition Properties

User-defined reports have properties pages that let you change several settings with how the graph looks. **Note:** You can also change these same settings by opening the table and resaving the report as a user-defined report instead.

4.11.13.6.1 Details

The details page lets you change the title for the report or modify the IBM i VRM levels that the report should apply to.

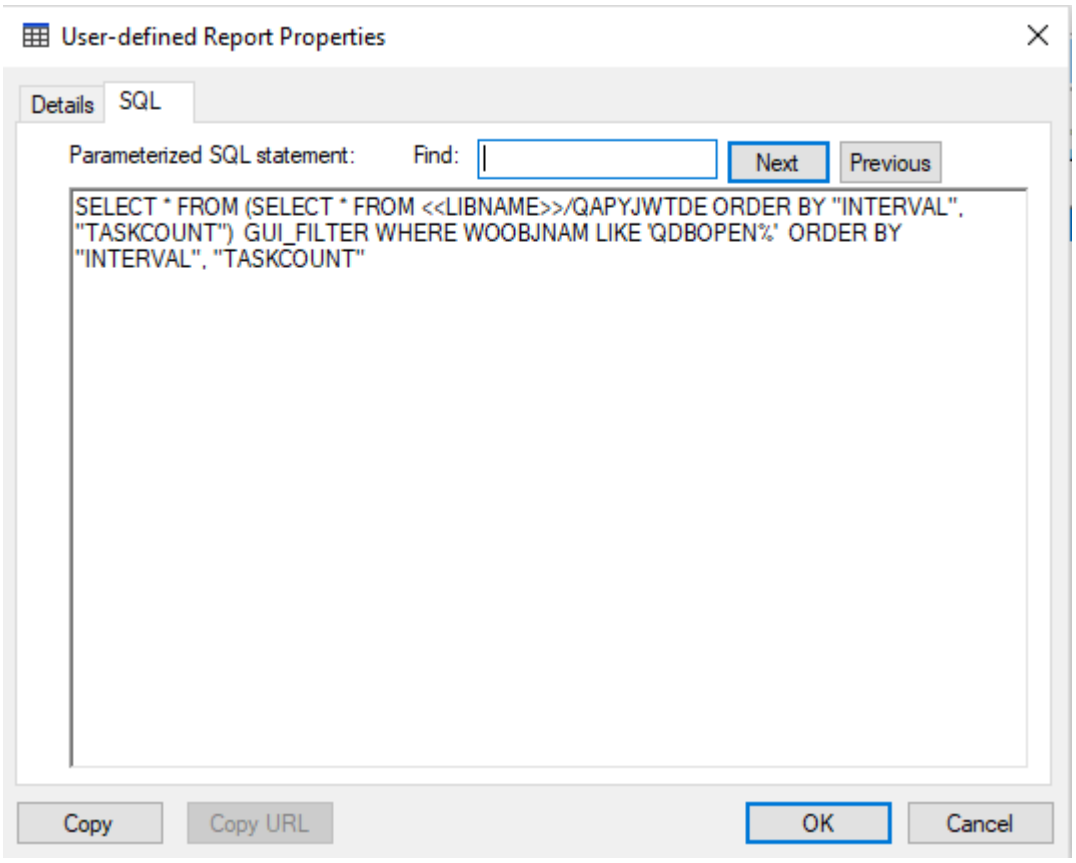


User-Defined Report Properties – Details

Option	Description
Description	The name to give the user-defined report.
Minimum VRM	The minimum IBM i release in nnn format such as 610, 710, 720, 730, etc. The minimum release currently functions at is V6R1 (i.e. 610)
Maximum VRM	The maximum IBM I release in nnn format this report should appear at. Use a value of 0 if no max.
Location	Identifies where this report exists in the user-defined reports database. User-defined reports are located within the database (either the .mdb file or IBM I library) in table QAIDRSQL. Within table QAIDRSQL the following columns are used in the example in the screenshot above: SIDCOMP = JW (component identifier) SQRYCAT = DTL (folder identifier – also see table QAIDRCATS) SREFNO = 2 (unique report identifier within this component and folder)
Open	Opens the user-defined reports database. **Note:** If using a local .mdb database, the QAIDRSQL table is not opened. You will need to open the table from the Tables section of MS Access manually and find the record using the information in the Location field above.

4.11.13.6.2 SQL

The SQL statement tab lets you change the parameterized SQL statement behind the report.

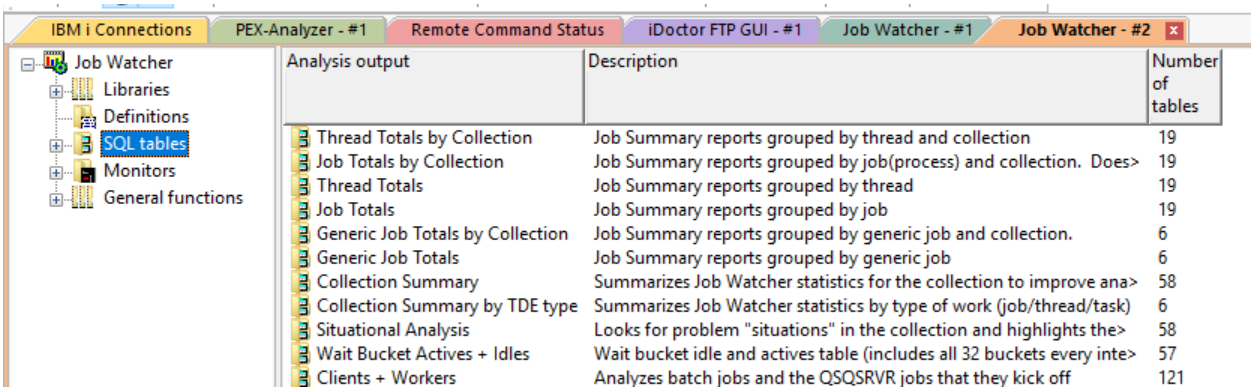


User-Defined Report Properties - SQL

Option	Description
Find	Lets you find the entered text within the SQL statement by pressing the Next or Previous button.
Parameterized SQL statement	This is the SQL statement including parameters such as: <<LIBNAME>> = library name <<MBRNAME>> = collection/member name **Tip:** If parameters are included in the SQL statement but are not known to iDoctor then you will be prompted to enter a value when running the report by the Change SQL Parameters window.

4.12 SQL Tables

The SQL tables folder in iDoctor is used to manage and work with the SQL tables generated by iDoctor analyses. This view organizes each type of SQL table into its own folder called "Analysis Output". This allows the user to more easily merge, graph and compare results from these tables by selecting the ones of the same type across different libraries and collections.



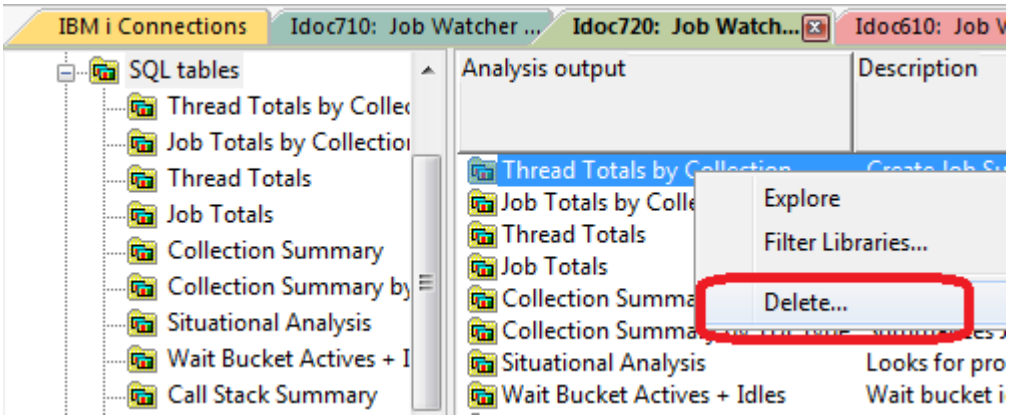
SQL tables folder within the Job Watcher component

Some analyses generate more than 1 type of report. In those cases, you may see an Analysis Output folder for each report type for the same analysis.

The SQL tables interface is also available under each library shown under the Libraries folder, and under each collection. This filters down the SQL tables to only include those in the current library (and/or collection.)

4.12.1 Analysis output menu

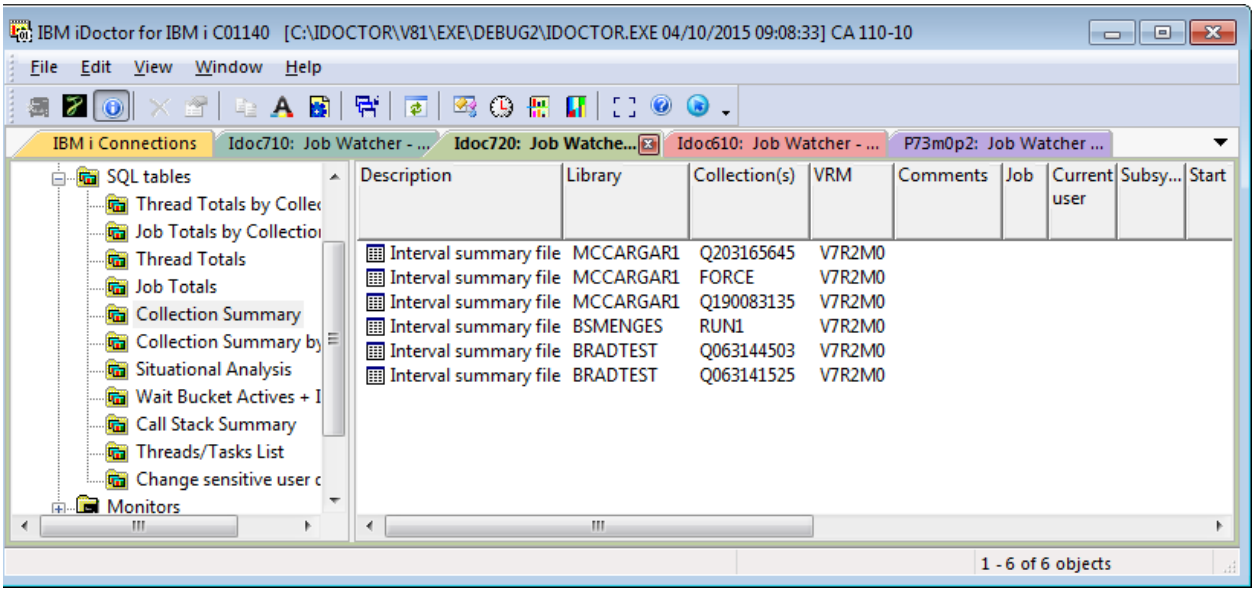
The SQL tables can be cleaned up (deleted) if desired by right-clicking the selected analysis output folders and using the Delete... menu. You can also select multiple folders when doing this action if desired.



Example of deleting all tables for a specific type

4.12.2 Tables

Within each analysis output folder will be a list of SQL tables found on the system that match the output folder you are working with.



List of SQL tables within the Collection Summary Folder

The list of SQL tables contains the following columns:

Column	Description
Description	Description for the SQL table
Library	The library (schema) the table resides in.
Collection(s)	The collection the data in the table applies to. In some cases, multiple collections will be listed.
VRM	The OS VRM of the system the collection was created on.
Comments	This is a user defined comment attached to the SQL table. This can be changed by right-clicking the table and using the Edit comment menu.
Job	The job filtering used when creating the SQL table (if applicable).
Current user	The current user profile filtering used when creating the SQL table (if applicable).
Subsystem	The subsystem filtering used when creating the SQL table (if applicable).
Start	The start time filtering used when the analysis was created.
End	The end time filtering used when the analysis was created.
SQL table name	The full SQL table name.
Change date	The date the table was last changed.
Physical file	The system-generated physical file object name (10 characters max) for the SQL table.

4.12.2.1 Menu options

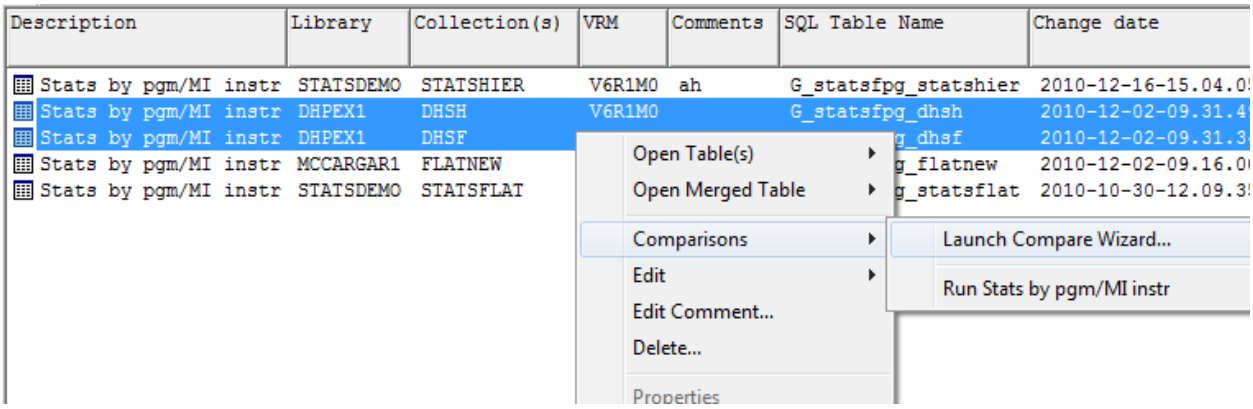
Right-clicking one or more SQL tables shows the following menu options:

Menu	Description
Open Table(s)	Opens the desired SQL tables in the Data Viewer.
Record Quick View	Lists the information about the selected SQL tables vertically in a new window.
Various graph/reporting options	In some cases, options may be shown to produce graphs or tables over the current SQL table's data.
Comparisons -> Launch Compare Wizard	This menu launches the SQL tables comparison wizard interface. This wizard allows the user to build reports over the SQL tables to do summarizations and comparisons.
Comparisons -> Run XYZ	Any IBM-supplied definitions that exist and are applicable to the current table are shown under the Launch Compare Wizard menu.
Open merged table	This option will be a report that combines all the selected table's data into 1 report. The data is simply UNIONed together and is not summarized.
Create merged table…	This option allows you to build a new table from the contents of all selected tables. You will be prompted for the name and library for the new table.
Edit comment	This option allows the user to modify the comment for the given SQL table.
Delete	This option lets the user delete the selected SQL table(s).
Properties	Displays property information for the SQL table.

4.12.3 SQL Tables Comparison Wizard

This wizard is launched by selecting 1 or more SQL tables right-clicking and using the Comparisons -> Launch Compare Wizard... menu. The wizard is designed to allow the user to build their own summarizations and comparisons for the desired tables. Several different types of reports can be generated.

Note: To use the Detailed report type (which compares 2 tables), only 2 tables can be selected.



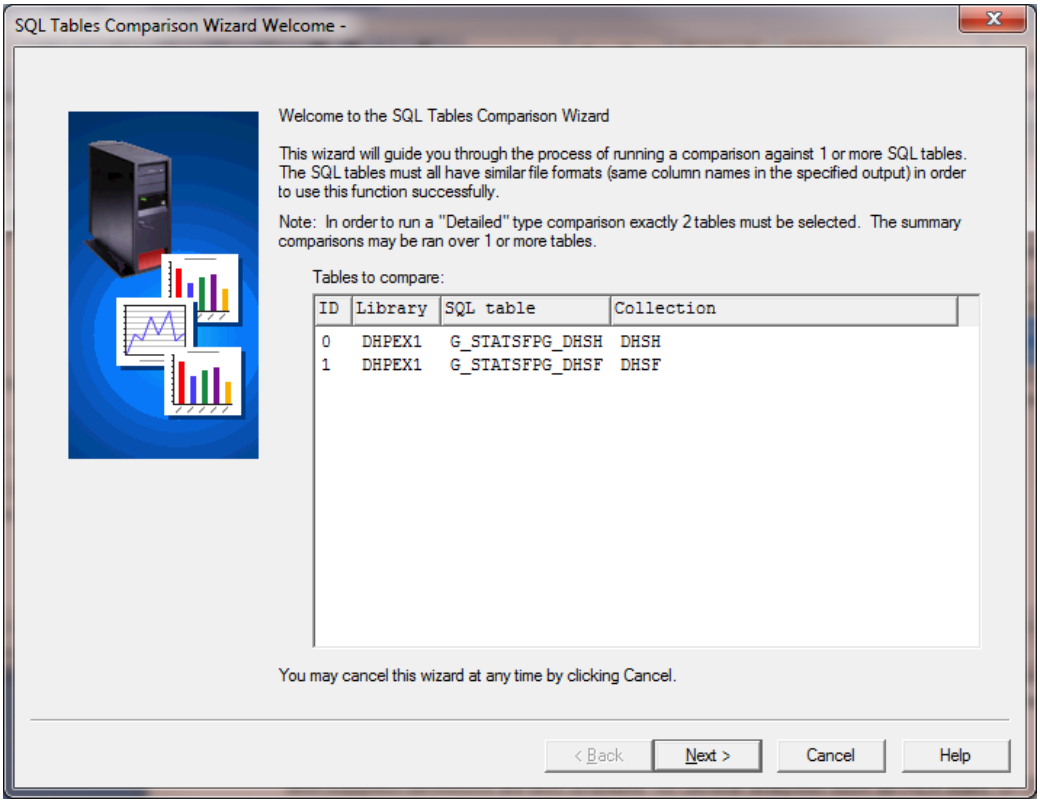
4.12.3.1 Definitions

The Wizard allows the user to save their changes into a Comparison definition. These definitions are only visible on the definition selection panel of the wizard and can be removed there. User-defined definition are stored in the file UserComparisons.mdb in the iDoctor install directory on the PC. If desired this file can be moved to another PC in order to utilize any comparison definitions. Currently there is no built in import/export function to do this.

IBM-supplied definitions are also available for several analyses such as PEX stats, or CSI system configuration.

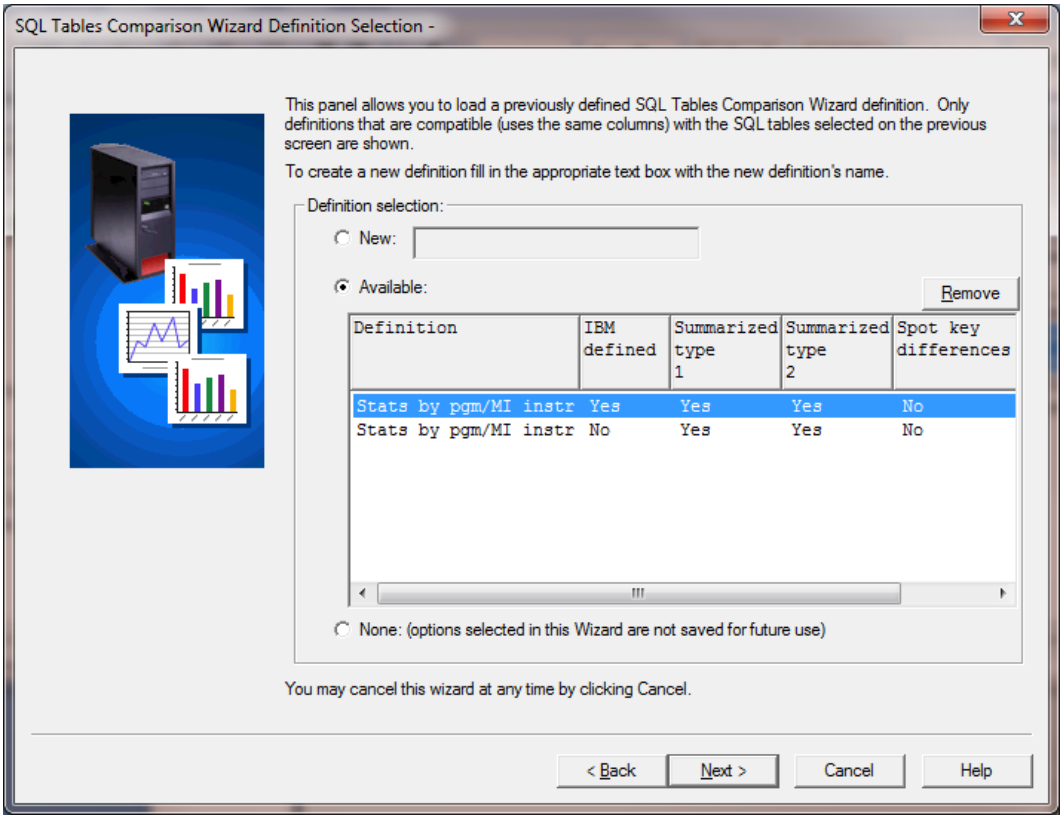
4.12.3.2 Welcome

The Welcome screen confirms the tables to compare. Note: Note in this example Library = schema and ID is automatically assigned and is used to simplify filtering on the Filter panel (shown later).



4.12.3.3 Definition Selection

This panel allows the user to indicate if an existing definition should be used or if a new one should be created. The final panel of the Wizard also allows you to modify the definition name



The list of definitions shows the definition name, whether the definition is IBM defined or user-defined and the report types generated by each.

4.12.3.4 (Report) Type Selection

This panel allows the user to indicate which types of reports to generate.

The report type options are described below:

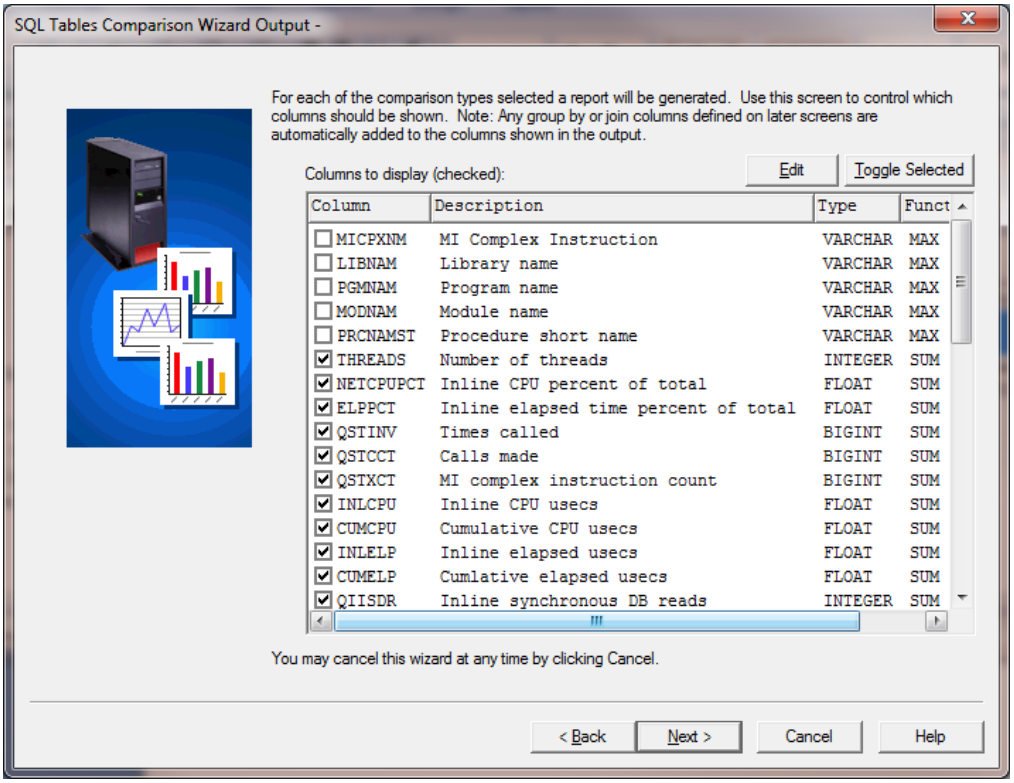
Option	Description
Summarized: 1 row per table	Builds a report that summarizes the data in each SQL table into 1 row. If this option is used the output, sort and filter panels will be shown.
Summarized: grouped by desired columns	Builds a report that summarizes the data in each SQL table by grouping on the desired columns. If this option is used the output, group by, sort and filter panels will be shown.

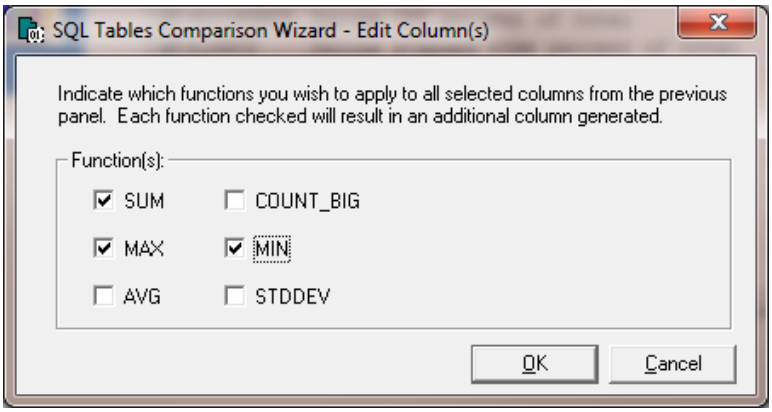
Differences	Builds a report that highlights differences between a set of tables. Typically this option is only used if dealing with data where the values in the tables are mostly the same and you only want to identify what has changed. It works well if you are comparing system configurations across many collections/runs and want to ensure that the environment is the same If this option is used the output, sort and filter panels will be shown.
Detailed	Builds a report that joins to tables together on the desired columns in order to make a detailed comparison between data in 1 table with data in the 2nd table. Delta values are given between the (numeric) output columns specified between the 2 tables. If this option is used the output, join criteria, sort and filter panels will be shown.
Detailed: Compare 1st to 2nd only	This option builds a detailed report where the delta values are only specified for table A - table B.
Detailed: Compare both ways, UNION results together	This option builds a detailed report where the delta values are shown for both directions (table A - table B and table B - table A). The results are unioned together into the same report.
Detailed: Compare both ways, separate reports	This option builds a detailed report where the delta values are shown for both directions (table A - table B and table B - table A). The results are shown in 2 different reports.
Detailed: Add delta percentages	This option will add an additional column after each delta column generated showing the percentage of the delta value within the absolute value of all deltas for the column. Because this option uses a WITH statement that calls the same subquery many times this option could be very slow.

4.12.3.5 Output

The Output panel allows the user to determine the columns to show from the SQL tables in the reports generated. If one of the summarized type of reports is generated then the list shown in this window will contain a FUNCTION column that specifies how that column should be aggregated when generating the summary report. Multiple functions can be applied if desired (each function generates an additional column in the output.). Select the desired columns to modify and press the Edit button to do this.

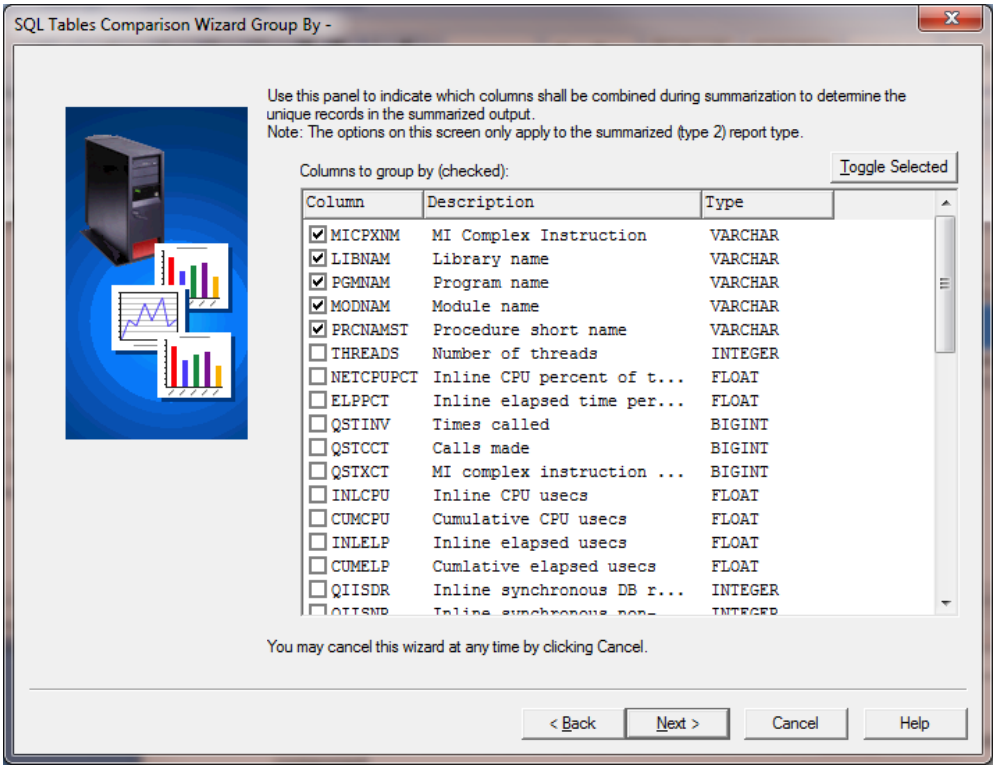
Note: Any group by or join columns are automatically added to the report and do not need to be selected on this screen.





4.12.3.6 Group By

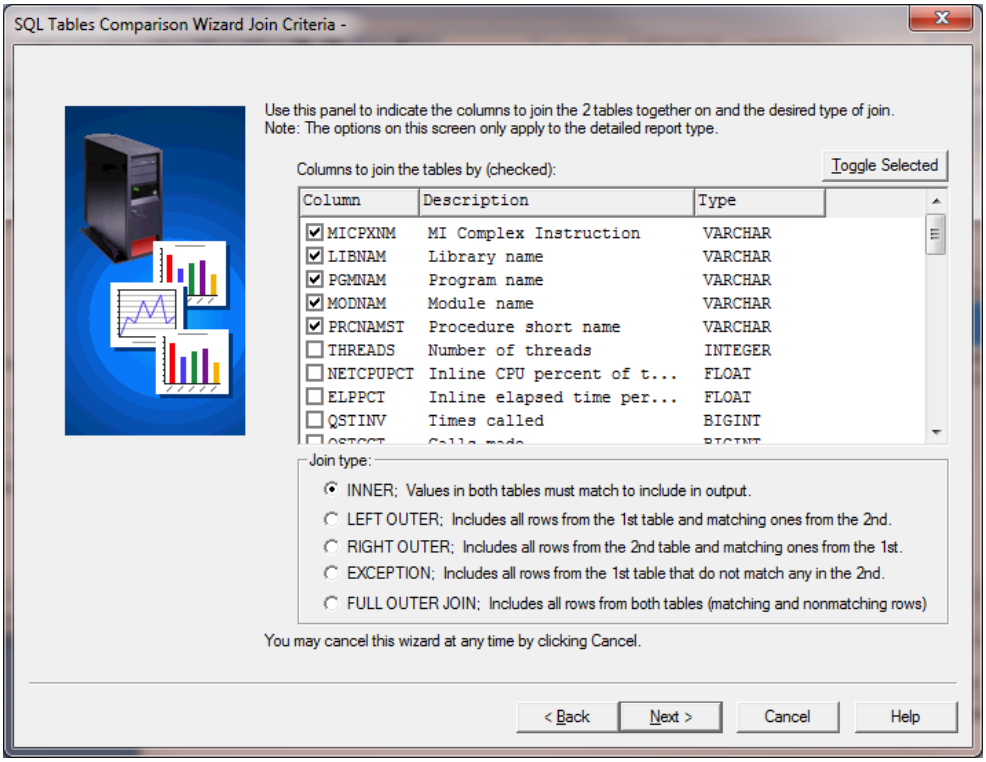
This panel provides the user with the capability to indicate which columns the data should be grouped on when doing the summary report. This means that the report will contain 1 record for each unique combination of the group by columns indicated.



4.12.3.7 Join Criteria

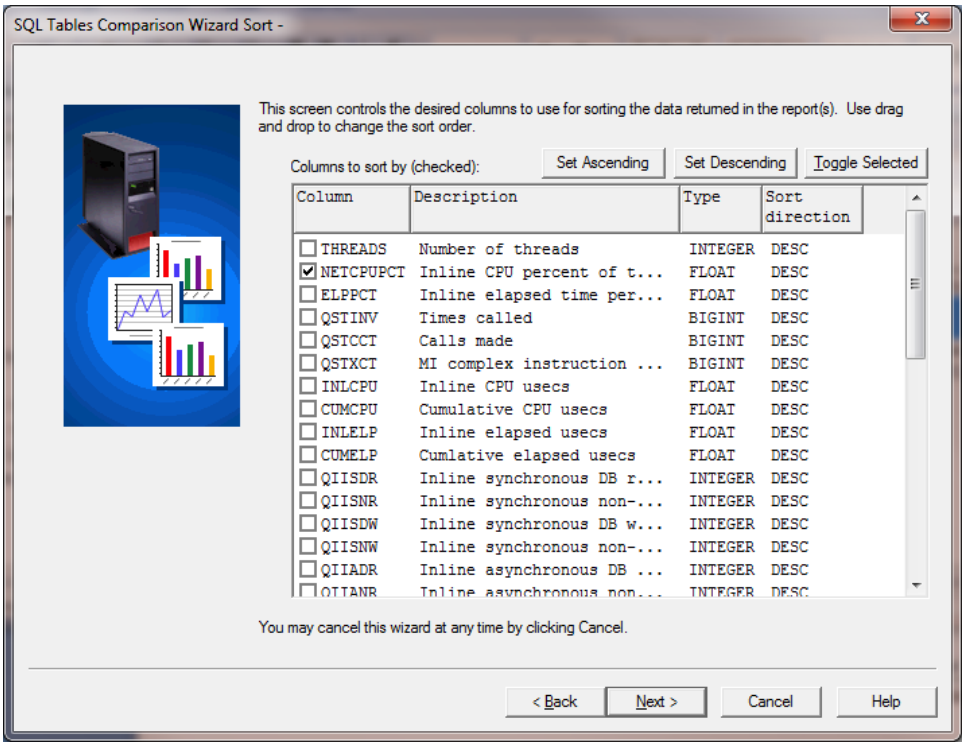
The Join criteria page lets the user specify how the 2 tables should be joined for the Detailed report. Typically these fields should represent the column that give uniqueness to each table's data.

The panel also gives the user the ability to custom to the type of join performed which controls things like whether data found in one table but not the other is included in the detailed report.



4.12.3.8 Sort

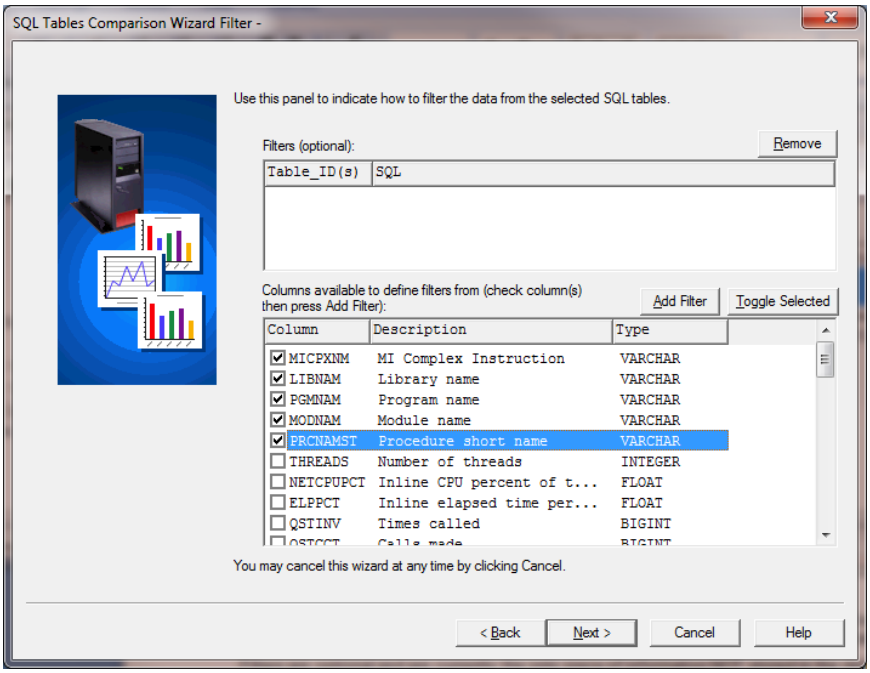
The sort panel lets the user control the order by for any of the report types generated. To change sort direction, simply select the desired columns and press the appropriate button at the top of the page.



4.12.3.9 Filter

The filter panel of the Wizard gives the user the power to do subsetting of the data before the summarization or comparison is performed. Data can be filtered from one table or all tables.

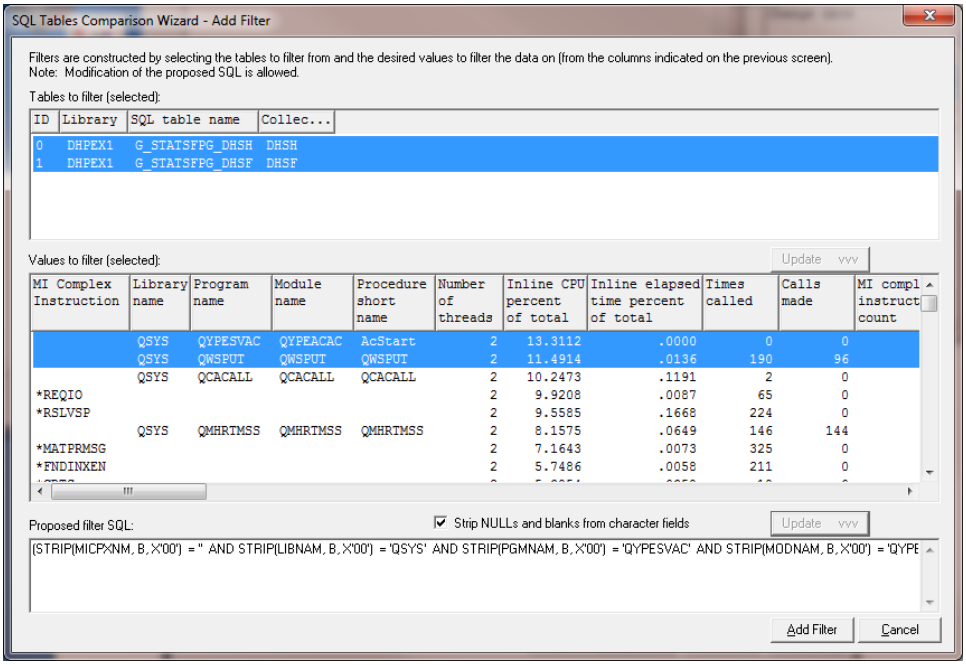
Filters are optional and are currently the only piece of information NOT stored in the definition. They are not stored in the definition because filters typically change often based on the problem at hand.



To add a filter, select the desired column(s) to filter on and press the Add Filter button.

This will generate a report in a new window showing all possible combinations of the selected columns to filter on. From this window you can select the desired values and press the 2nd update button. To control which tables are included in the report, you can deselect one or more tables and press the 1st update button to rebuild the report.

After pressing the 2nd update button, the SQL to use to perform the actual filtering is shown. This can be modified to suit individual needs (such as using LIKE or IN syntax, etc).

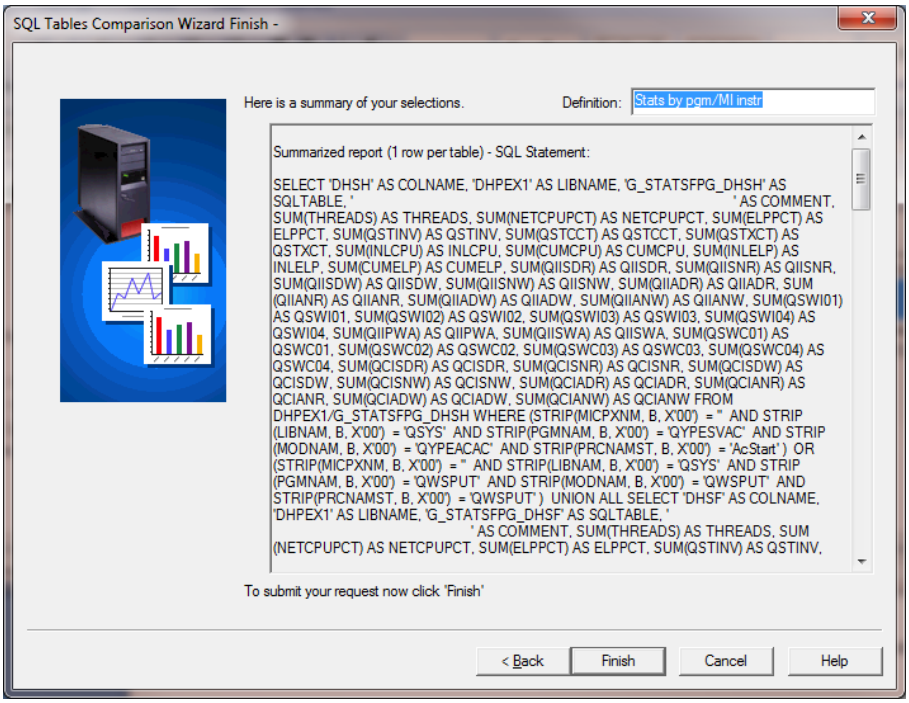


If a filter being defined should only apply to some of the tables this can be controlled by unselecting tables at the top of the window and rebuilding the list of values to filter.

Pressing the Add Filter button will close the popup window and return to the Wizard, adding the new filter information to the top of the window.

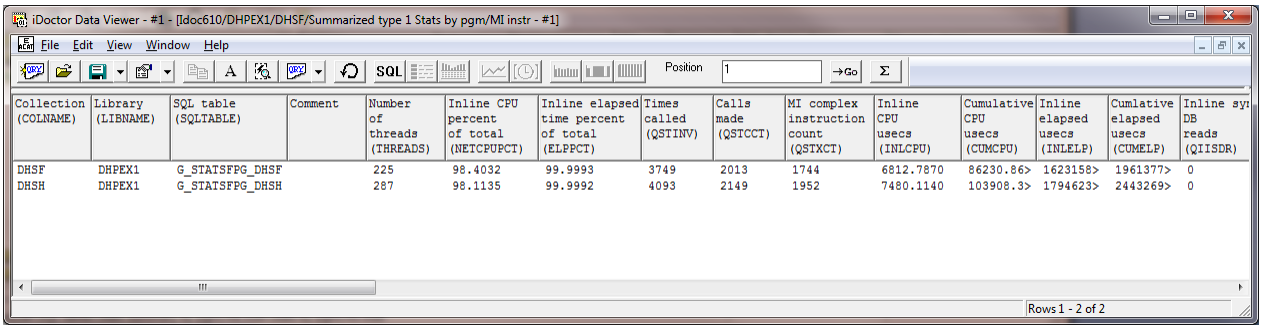
4.12.3.10 Finish

The last panel of the Wizard shows the SQL statements that have been generated by the Wizard for each report type and also gives the user the option to modify the name of the definition to save changes to.

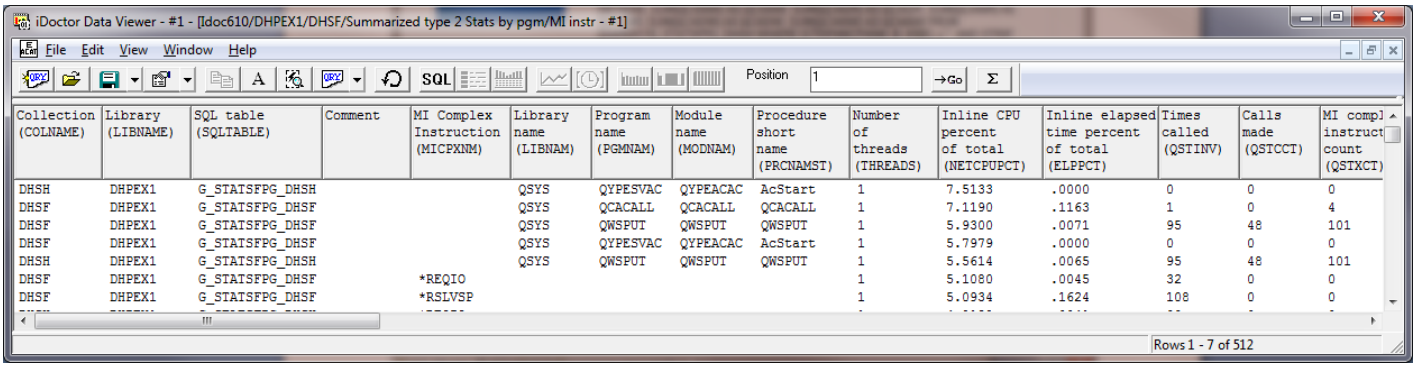


4.12.3.11 Reports

After Finish is pressed in the wizard the reports generated by the Wizard are shown in the Data Viewer.



Collection (COLNAME)	Library (LIBNAME)	SQL table (SQLTABLE)	Comment	Number of threads (THREADS)	Inline CPU percent of total (NETCPUPCT)	Inline elapsed time percent of total (ELPPCT)	Times called (QSTINV)	Calls made (QSTCCT)	MI complex instruction count (QSTXCT)	Inline CPU usecs (INLCPU)	Cumulative CPU usecs (CUMCPU)	Inline elapsed usecs (INLELP)	Cumlative elapsed usecs (CUMELP)	Inline syn DB reads (QIISDR)
DHSF	DHPEX1	G_STATSFPG_DHSF		225	98.4032	99.9993	3749	2013	1744	6812.7870	86230.86>	1623158>	1961377>	0
DHSH	DHPEX1	G_STATSFPG_DHSH		287	98.1135	99.9992	4093	2149	1952	7480.1140	103908.3>	1794623>	2443269>	0



Collection (COLNAME)	Library (LIBNAME)	SQL table (SQLTABLE)	Comment	MI Complex Instruction (MICPXNM)	Library name (LIBNAM)	Program name (PGMNAM)	Module name (MODNAM)	Procedure short name (PRCNAMST)	Number of threads (THREADS)	Inline CPU percent of total (NETCPUPCT)	Inline elapsed time percent of total (ELPPCT)	Times called (QSTINV)	Calls made (QSTCCT)	MI compl instruct count (QSTXCT)
DHSH	DHPEX1	G_STATSFPG_DHSH			QSYS	QYPESVAC	QYPEACAC	AcStart	1	7.5133	.0000	0	0	0
DHSF	DHPEX1	G_STATSFPG_DHSF			QSYS	QCACALL	QCACALL	QCACALL	1	7.1190	.1163	1	0	4
DHSF	DHPEX1	G_STATSFPG_DHSF			QSYS	QWSPUT	QWSPUT	QWSPUT	1	5.9300	.0071	95	48	101
DHSF	DHPEX1	G_STATSFPG_DHSF			QSYS	QYPESVAC	QYPEACAC	AcStart	1	5.7979	.0000	0	0	0
DHSH	DHPEX1	G_STATSFPG_DHSH			QSYS	QWSPUT	QWSPUT	QWSPUT	1	5.5614	.0065	95	48	101
DHSF	DHPEX1	G_STATSFPG_DHSF		*REQIO					1	5.1080	.0045	32	0	0
DHSF	DHPEX1	G_STATSFPG_DHSF		*RSLVSP					1	5.0934	.1624	108	0	0

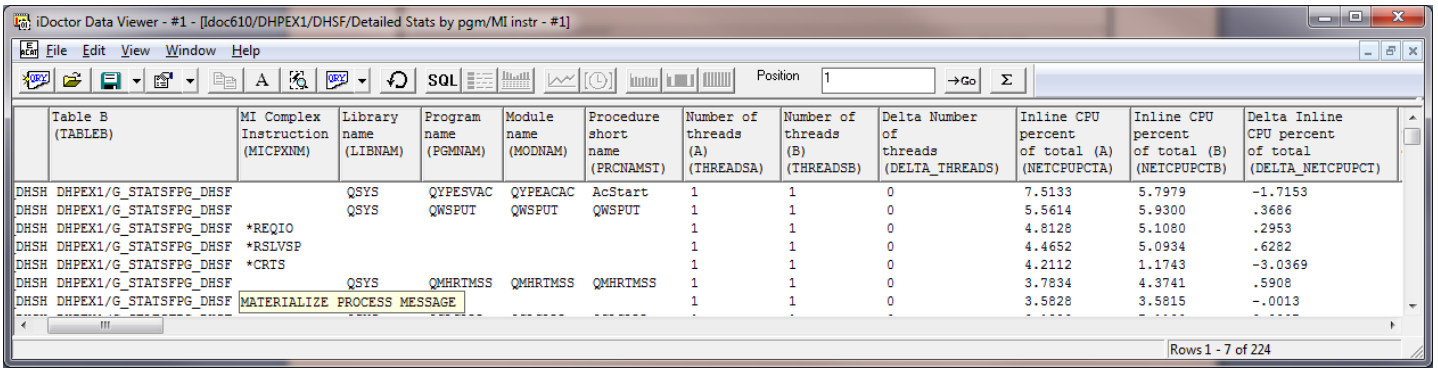


	Table B (TABLEB)	MI Complex Instruction (MICPXNM)	Library name (LIBNAM)	Program name (PGMNAM)	Module name (MODNAM)	Procedure short name (PRCNAMST)	Number of threads (A) (THREADSA)	Number of threads (B) (THREADSB)	Delta Number of threads (DELTA_THREADS)	Inline CPU percent of total (A) (NETCPUPCTA)	Inline CPU percent of total (B) (NETCPUPCTB)	Delta Inline CPU percent of total (DELTA_NETCPUPCT)
DHSH	DHPEX1/G_STATSFPG_DHSF		QSYS	QYPESVAC	QYPEACAC	AcStart	1	1	0	7.5133	5.7979	-1.7153
DHSH	DHPEX1/G_STATSFPG_DHSF		QSYS	QWSPUT	QWSPUT	QWSPUT	1	1	0	5.5614	5.9300	.3686
DHSH	DHPEX1/G_STATSFPG_DHSF	*REQIO					1	1	0	4.8128	5.1080	.2953
DHSH	DHPEX1/G_STATSFPG_DHSF	*RSLVSP					1	1	0	4.4652	5.0934	.6282
DHSH	DHPEX1/G_STATSFPG_DHSF	*CRTS					1	1	0	4.2112	1.1743	-3.0369
DHSH	DHPEX1/G_STATSFPG_DHSF		QSYS	QMHRTMSS	QMHRTMSS	QMHRTMSS	1	1	0	3.7834	4.3741	.5908
DHSH	DHPEX1/G_STATSFPG_DHSF	MATERIALIZE PROCESS MESSAGE					1	1	0	3.5828	3.5815	-.0013

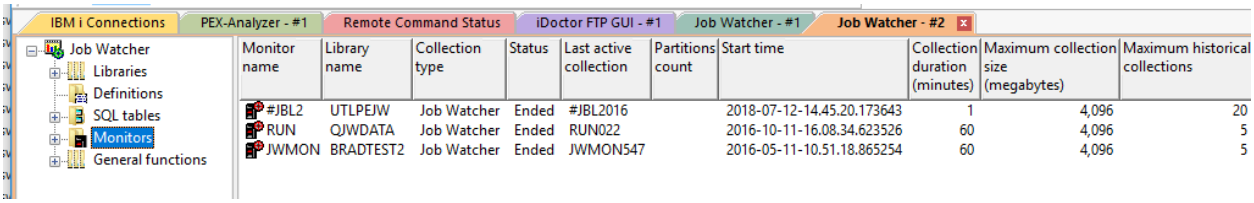
4.13 Monitors

The Job Watcher, Disk Watcher and PEX Analyzer components provide an option to start and work with monitors. Monitors allow the user to continuously collect Job Watcher, Disk Watcher and/or PEX data.

Monitors run continuously storing only the most recent number of collections desired. Monitors will run until ended manually by the user or when ended via a scheduled job. Monitors can be held and released if the user wishes to stop collecting data, and then continue collection again later. Monitors can also be scheduled to start and end at the desired times.

Once a monitor has been started and ended, it must be restarted using the Restart Monitor option. You cannot use the Start New Monitor option to restart an existing monitor.

The Monitors Folder is provided in iDoctor to allow the user to work with the monitors that exist on the current system.



Monitors Folder

The fields shown in this view are as follows:

Field	Description
Monitor name	The name of the monitor. Monitor names cannot be greater than 7 characters. The collections within the monitor use the monitor name concatenated with 001 through 999.
Library name	The library name the monitor's collections reside in.
Collection type	Either Job Watcher, Disk Watcher or PEX Analyzer.
Status	Indicates if the monitor is currently running or if it has ended.
Last active collection	Provides the collection name in the monitor for either the last active collection or the collection currently running.
Partitions count	If the monitor is collecting data simultaneously over multiple partitions this field indicates the number of partitions data is being collected for.
Start time	The date and time when the monitor started.
Collection duration (minutes)	The duration of each collection in minutes. It's generally not recommended to create a single collection greater than an hour since the larger the collection, the more time consuming it will be to run the queries to analyze the data. On some systems with large numbers of jobs, you may want to set the collection duration to 30 minutes or less.
Maximum collection size (megabytes)	This option indicates the maximum collection size for each collection created within the monitor. This value only applies to Disk Watcher and Job Watcher.
Maximum historical collections	This parameter indicates how many collections the monitor should contain at 1 time. As time progresses and the maximum is reached, the oldest collections are replaced as new collections are added.
Definition	The name of the definition used when the monitor was started.
Description	A description given to the monitor when it was started or restarted.
Monitor job	The job name that started and ended the collections in the monitor.
ENDPEX data option	This option applies only to PEX monitors and can be one of the following: 1) **Create DB files** - The data is dumped into the PEX DB files when each collection ends 2) **Create *MGTCOL** - The PEX data is dumped into a PEX *MGTCOL object when each collection ends 3) **Suspend** - The PEX data is not dumped and the collection will move to suspended status. After the desired maximum historical collections have been created, the PEX monitor will end. At that point the data must be dumped to database files or *MGTCOL objects manually using either the ENDPEX command or the Active collections folder within PEX Analyzer.

The following options are available when right clicking on one or more monitors in the list:

Popup Menu	Description
Explore	This option allows you to view the collections that are contained within the selected monitor.
Select fields…	Allows the user to modify the fields shown in the list of collections. These are found within the monitor by expanding it.
Start New Monitor	Opens the Start iDoctor Monitor Wizard to create a new monitor.
Restart Monitor	Opens the Start iDoctor Monitor Wizard to restart the selected monitor. This option is only enabled if 1 monitor is selected.
Hold/Release	This option allows the selected monitor to be held. If held the active collection will be ended immediately and no more collections will be started until the monitor is released.
End immediately	This option will end the monitor and all active collections defined within immediately.
End after current collection	This option will end the monitor once the current collection running completes.
Delete	This option will remove the monitor and all collections contained within them from the system.
Save	Use this option to save all collections within the monitor to a save file.
Transfer to	Use this option to save and then transfer all collections within the monitor to another system.
Display job log	Displays the job log for the selected monitor. This option will only appear if 1 monitor is selected and the job log exists.

4.13.1 Start iDoctor Monitor Wizard

This section describes the interface used when starting (or restarting) an iDoctor monitor.
Monitors for Job Watcher, Disk Watcher and PEX can be started at the same time using this interface if desired.

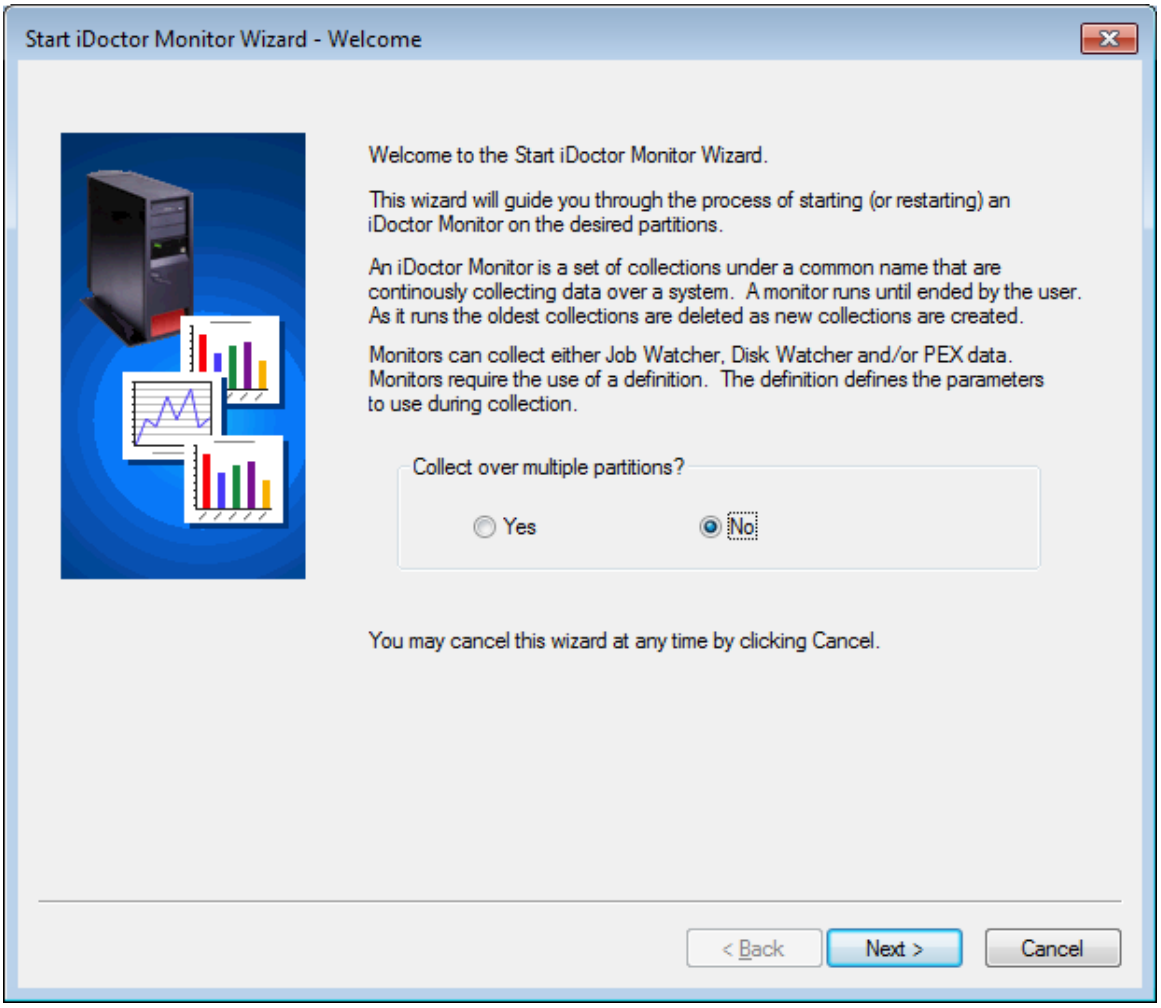
When restarting a monitor, the parameters that were used to last start the monitor are preloaded into this interface.

4.13.1.1 Welcome

The Welcome page introduces the user to the Monitor.

Depending on the client/server build level installed, you may have the option to start the monitor over multiple partitions. Selecting Yes on this screen will allow you to pick the partitions to collect data on.

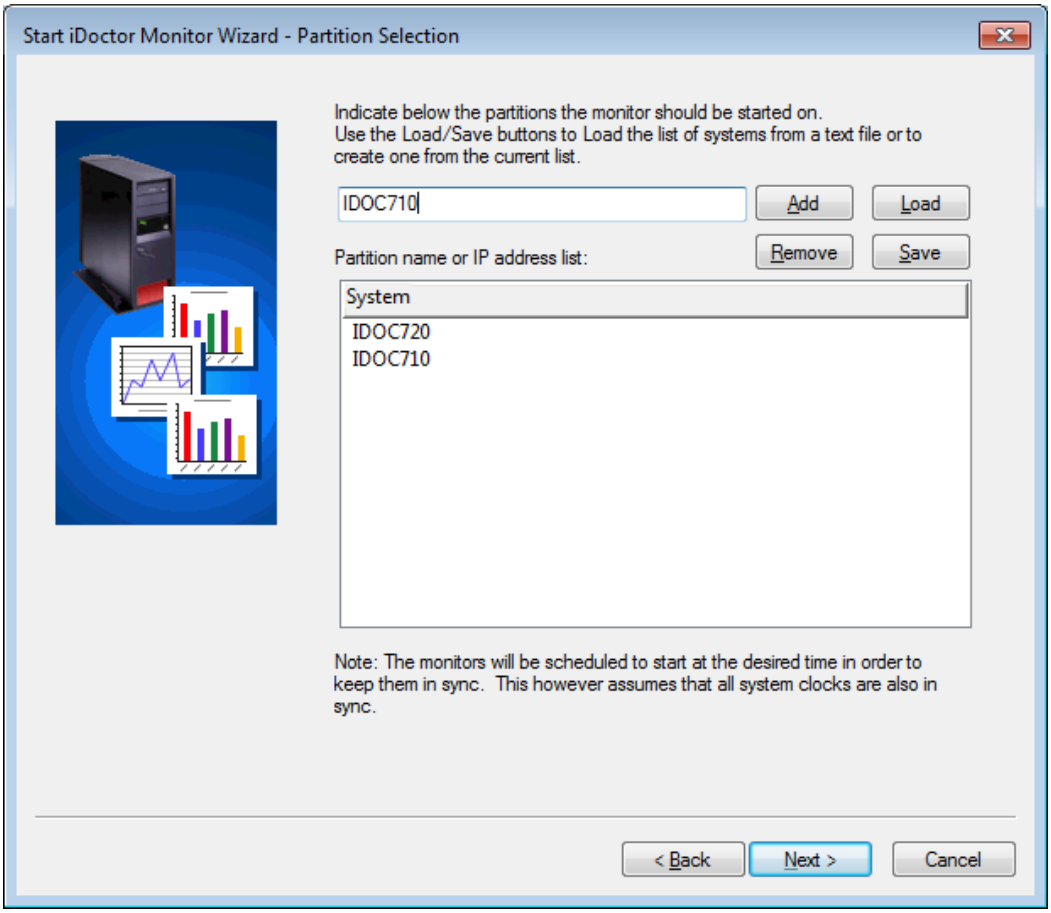
Tip: In order to make it easier to analyze the data, the system clocks on the partitions used should be in sync.



Start iDoctor Monitor Wizard - Welcome

Option	Description
Collect over multiple partitions	Select Yes if you want to collect the monitor across multiple partitions.

Pressing Next on the Welcome page shows the Partition Selection screen.
The Partition Selection screen shows the user the list of partitions that should be used to collect the monitor on.

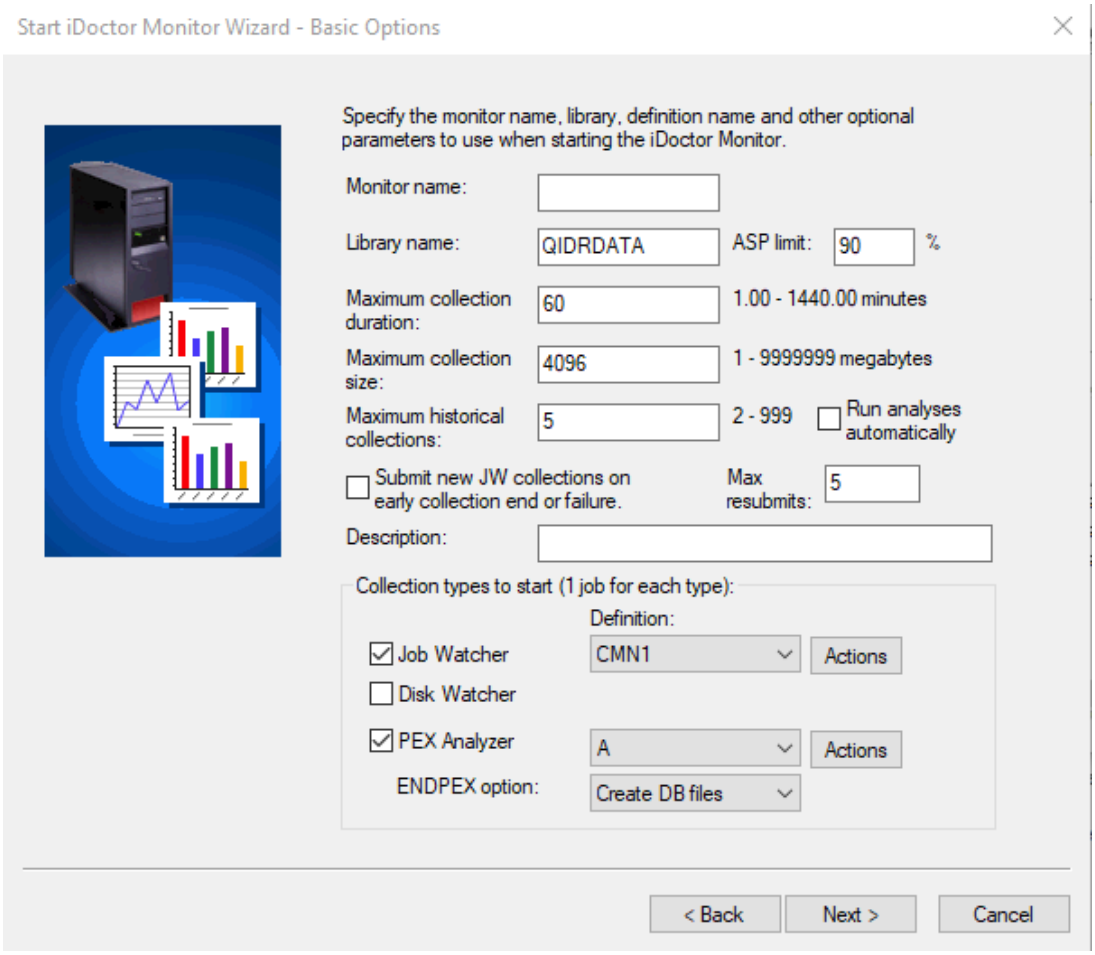


Start iDoctor Monitor Wizard – Partition Selection

Option	Description
Add	Adds the partition name or IP address in the text box to the list.
Remove	Removes the selected partitions from the list
Load	Loads a list of partition names from a text file. The file should have a partition name or IP address with nothing else in it on each line.
Save	Saves the current list of partitions to a text file that can be used later via the Load option.
Partition list	The list of partitions that the monitor(s) will collect data on.

4.13.1.2 Basic Options

This page allows the user to enter the parameters like the monitor name, library, and the type(s) of monitors to create.



Start iDoctor Monitor Wizard – Basic Options

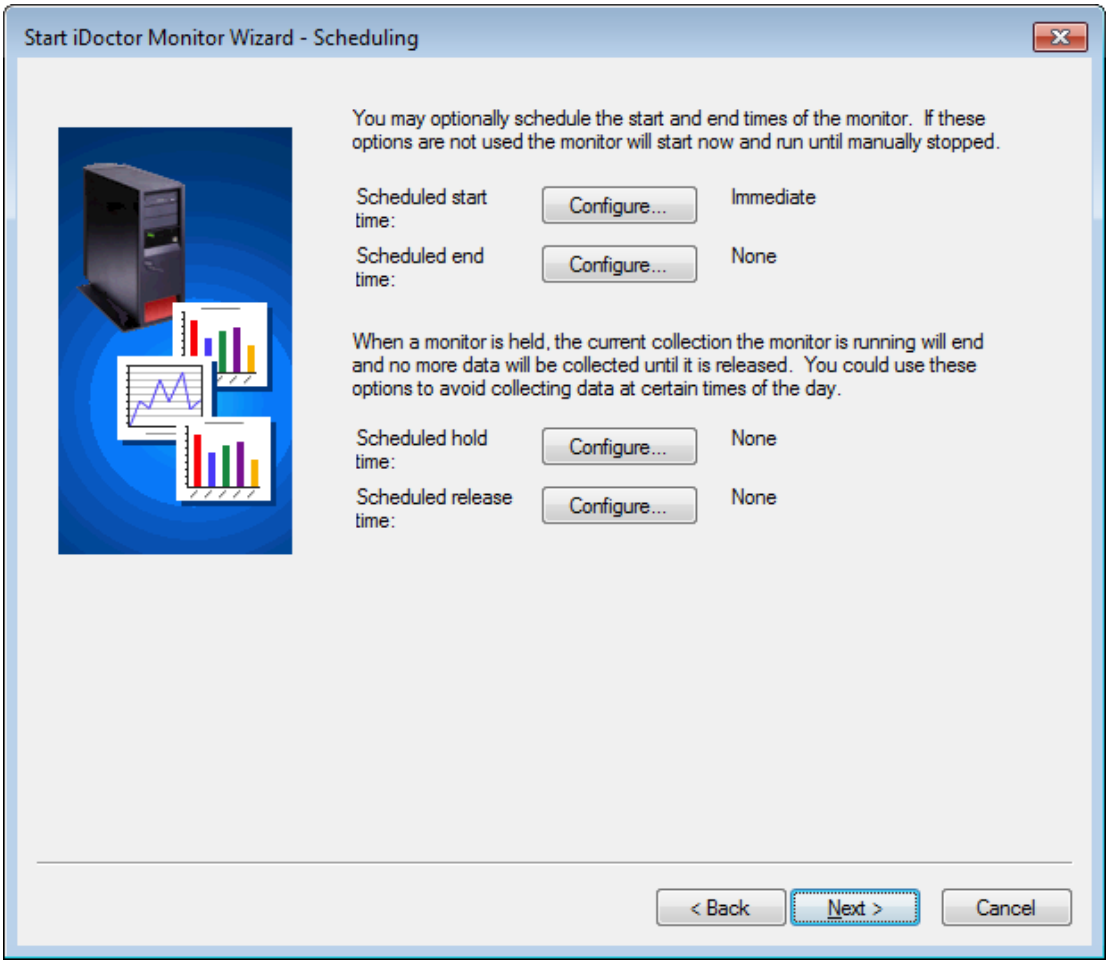
The following section lists the parameters available on this interface:

Option	Description
Monitor name	The name of the monitor. Monitor names cannot be greater than 7 characters. The collections within the monitor use the monitor name plus 001 through 999.
Library	The library name the monitor's collections should reside in.
ASP limit	This value indicates the maximum allowed ASP percentage used. If while the monitor is running this value is exceeded the monitor will end. **Note:** The ASP checked is the same as the ASP that the library resides in.
Maximum collection duration	Indicates how long each collection should run for (in minutes). **Tip:** Ensure that the definitions specified would allow the collection to run for at least this long to avoid having gaps in the monitor data where no data is being collected.
Maximum collection size	This parameter indicates the maximum size to allow for each collection in the monitor. If the size is exceeded, then the collection will stop and there will be a gap in the collection data until the monitor starts the next collection in the sequence. **Note:** This parameter only applies to Job Watcher and Disk Watcher Monitors. For PEX monitors you will need to change this value in the PEX definition instead..
Maximum historical collections	This parameter indicates how many collections the monitor should contain at 1 time. As time progresses and this maximum is reached, the oldest collections are replaced as new collections are added.
Run analyses automatically	If checked, the Run ALL default analyses option will be used. All default analyses will be executed for each collection after it completes. This is NOT all analyses but only a select few that are most commonly needed. **WARNING:** In some situations, this can be very resource intensive and typically should not be used on production systems.

Submit new JW collections on early collection end or failure	This optional parameter indicates if the Job Watcher monitor should attempt to submit a new collection if it's detected that the current collection has ended prematurely (for any reason). If this option is enabled, a new collection will be submitted up to the maximum specified by the max resubmits parameter if the current collection has stopped running. Use caution when using this option; your collection may have ended early because of disk space limits.
Max resubmits	The parameter indicates the number of times collections will be resubmitted if the previous field is enabled.
Description	A description given to the monitor.
Collection types to start	The user can collect Job Watcher, Disk Watcher and/or PEX. If multiple choices are selected a different monitor job is started one for each collection type.
Definition	This list provides the definitions available on the current system to pick from of the applicable type.
Actions	The definition actions include: **View** – Displays the Properties interface for the selected definition. **Change** – Displays the Add Definition Wizard with the selected definition's parameters filled into the interface. **New** – Displays the Add Definition Wizard in order to create a new definition. **Reload IBM-supplied definitions** – Runs a stored procedure to ensure that the IBM-supplied definitions are loaded and up to date.
ENDPEX option	The ENDPEX option is only applicable to PEX Analyzer monitors. It indicates how the collections generated by the monitor should be handled by providing 3 options: 1) **Create DB files** - The data is dumped into the PEX DB files when each collection ends 2) **Create *MGTCOL** - The PEX data is dumped into a PEX *MGTCOL object when each collection ends 3) **Suspend** - The PEX data is not dumped and the collection will move to suspended status. After the desired maximum historical collections have been created, the PEX monitor will end. At that point the data must be dumped to database files or *MGTCOL objects manually using either the ENDPEX command or the Active collections folder within PEX Analyzer.

4.13.1.3 Scheduling

This page allows the user to determine how to when the monitor should be started/ended, held or released. To run the monitor right away, click Next.



Start iDoctor Monitor Wizard – Scheduling

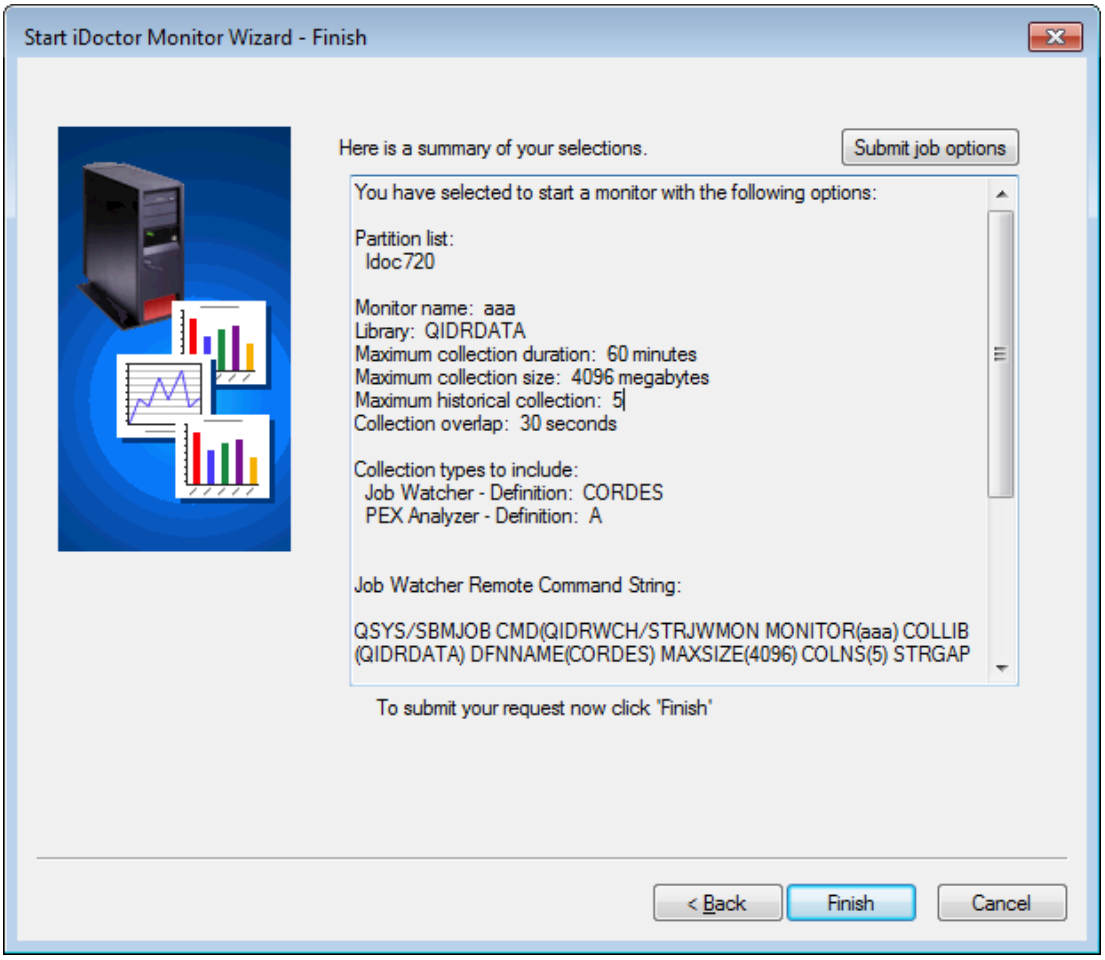
The following section lists the parameters available on this interface:

Option	Description
Scheduled start time	Use this option to schedule the monitor to start later..
Scheduled end time	Use this option to schedule the monitor to end at a desired date and time.
Scheduled hold time	Use this option to hold the monitor at the desired date and time.
Scheduled release time	Use this option to release the monitor (assuming it's in a held state) at the desired date and time.

4.13.1.4 Finish

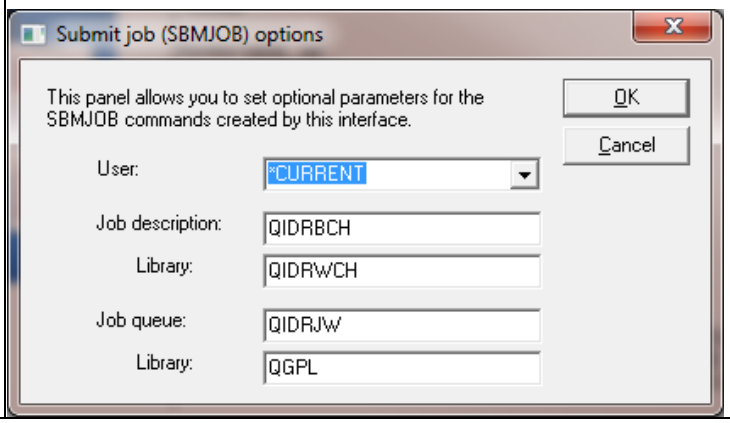
This screen provides a summary of the monitor that will be started/restart on the current system.

For your convenience the remote commands that will be executed to start the monitor(s) are listed at the bottom of this page.



Start iDoctor Monitor Wizard – Finish

The following section lists the parameters available on this interface:

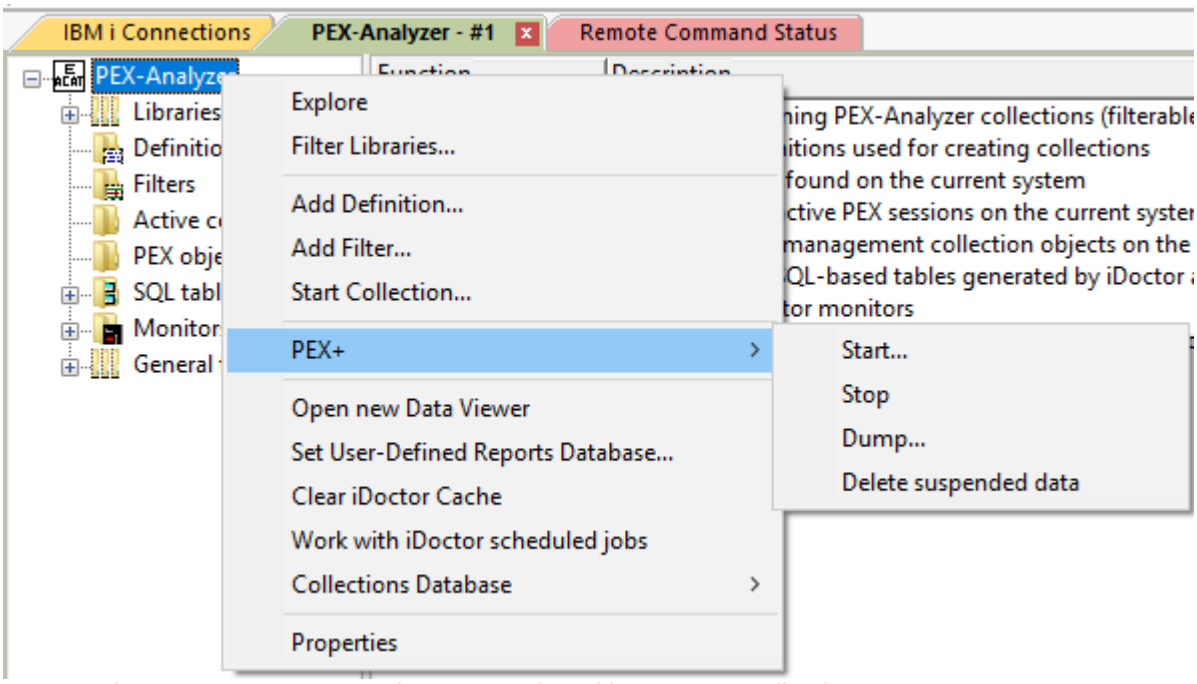
Option	Description
Submit job options	This button allows you to change parameters on the SMBJOB commands used to start the monitors. 

4.14 PEX+

PEX+ collections are a new type of “super collection” produced by the QMGTOOLS/STRPEX_ command.

This option allows a user to collect multiple types of performance data simultaneously and tie them together within one collection. The available types of data that may be included in a PEX+ collection are: Collection Services, PEX, Job Watcher, trace connections and XSM SST macro dumps.

PEX+ collections are started and stopped in PEX Analyzer from the PEX Analyzer icon. Right-click this folder and use the PEX+ menu options:



PEX Analyzer -> PEX+ menu options to start/stop/dump super collections

Note: To view the components of a PEX+ super collection, use the applicable iDoctor components for the desired data type. No interface currently exists to browse all components of a PEX+ collection as a single entity.

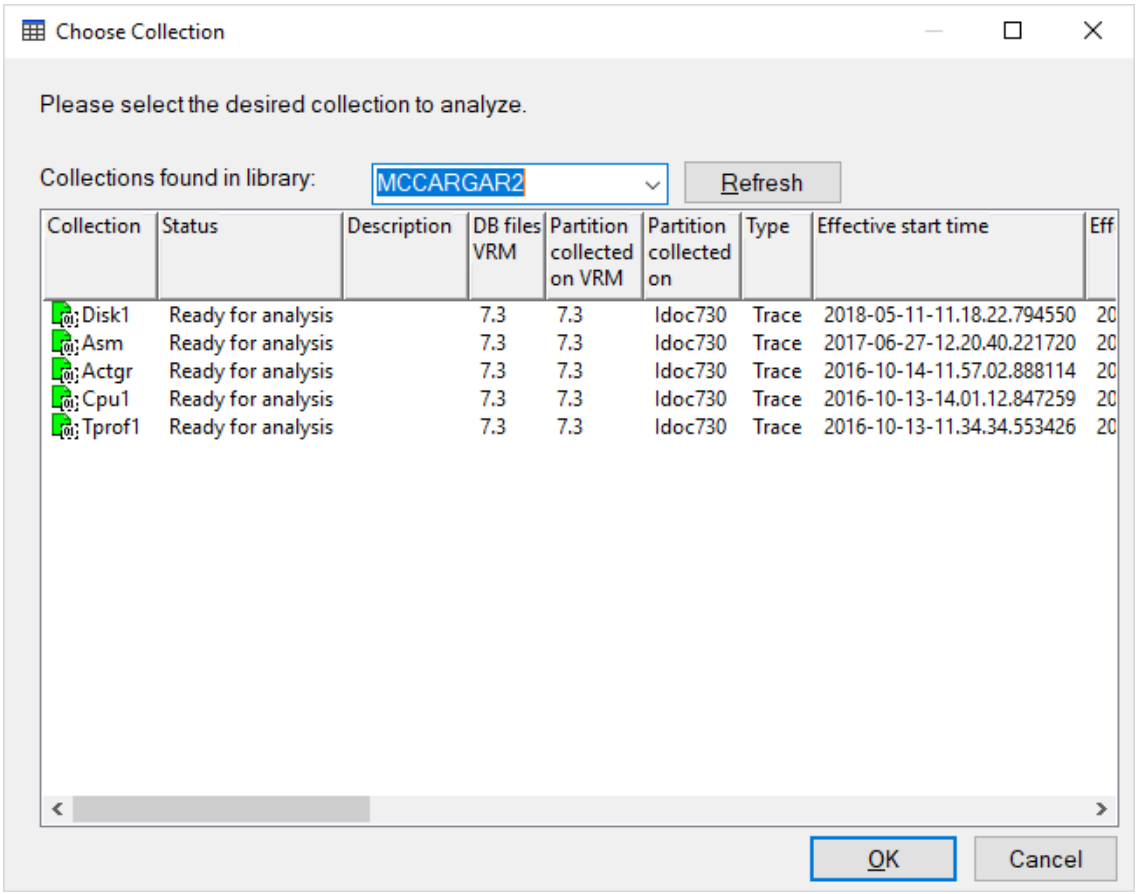
4.15 Choose Collection

The Choose Collection interface allows a user to select the desired collection to work with. It is shown to the user in two cases:

1. The required collection database member does not exist for one or more tables in the current SQL statement.

2. In the Data Viewer -> SQL Editor the user has chosen the Choose Collection or DB members menu option.

An example of this interface is:



Choose Collection Window

Option	Description
Collections found in library drop-down	This lets you pick a different library to use for the collection to open.
Refresh	Rebuilds the list of collections in the current library specified.
Collections found	The list of collection to choose from. Pick the desired one and press the OK button to continue.

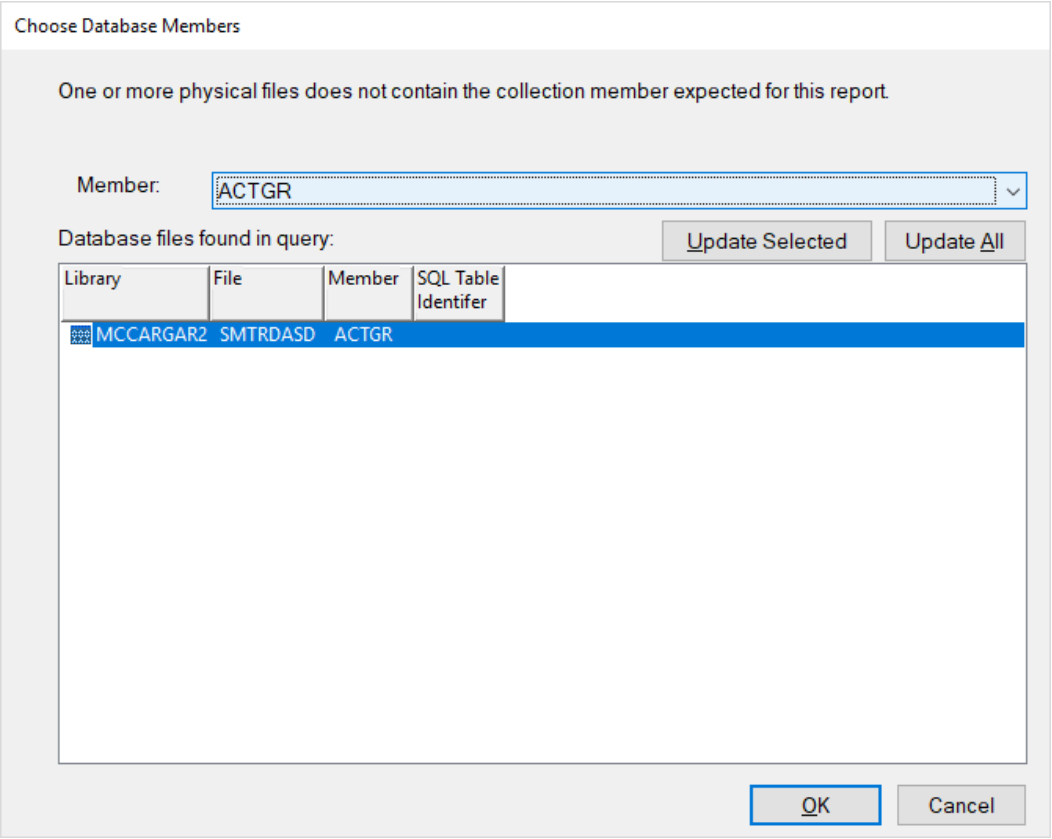
4.16 Choose Database Members

The Choose Database Members interface allows a user to select the desired members to work with. It is shown to the user in two cases:

1. The expected database member does not exist for one or more tables in the current SQL statement.

2. In the Data Viewer -> SQL Editor the user has chosen the Choose Collection or DB members menu option.

An example of this interface is:



Choose Database Members

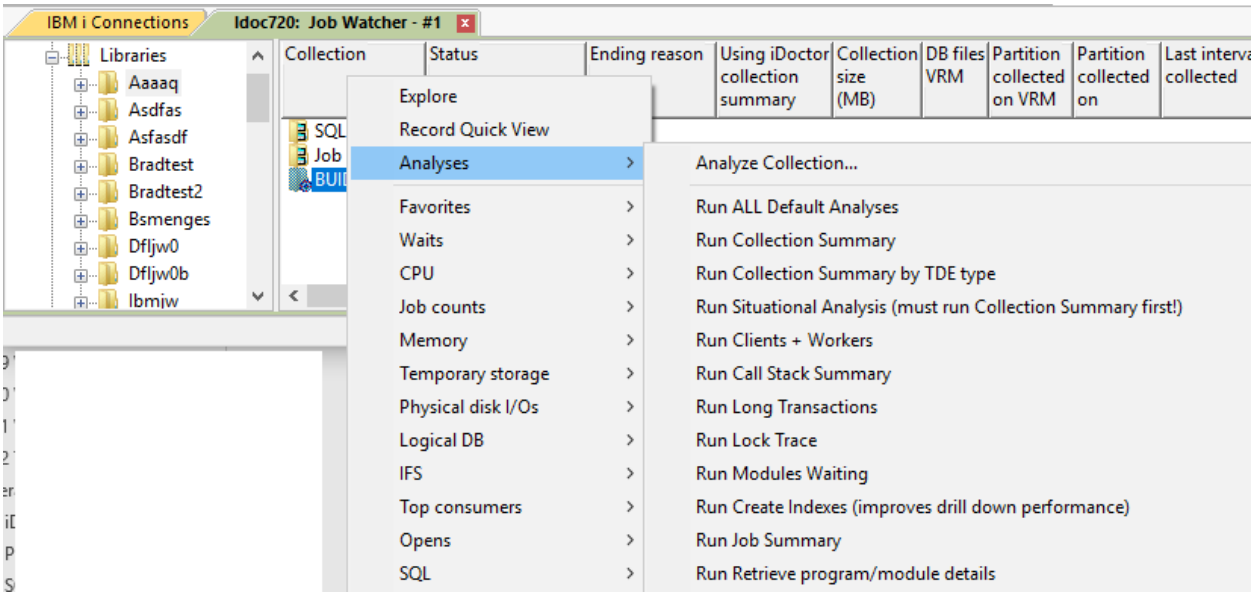
Option	Description
Member drop-down	This is a list of all members found in the currently selected file in the list.
Update Selected	This button will update the list of all **selected** files found to use the member name specified in the drop-down list.
Update All	This button will update the list of **all** files found to use the member name specified in the drop-down list.
Database files found in query	This is a list of all database files found in the current SQL statement where more than 1 member exists.

4.17 Analyses

Analyses in iDoctor are used to process performance data to either summarize the data in some manner or dig deeper and look for specific information for a given performance problem.

Analyses can be found in the main iDoctor components: Job Watcher, PEX Analyzer, Disk Watcher and CSI (Collection Services Investigator).

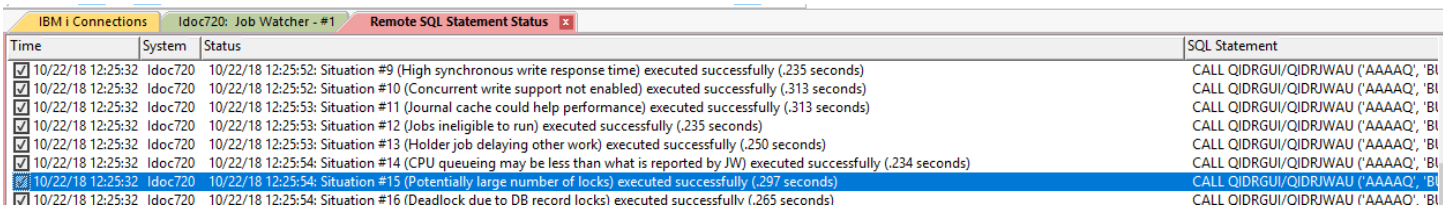
Analyses are initiated by right-clicking collection(s) (or library(ies)) and using the Analyses menu and picking either the Analyze Collection... menu (which shows a window to pick and choose from any available analyses) or pick one of the available "fast path" analysis options such as "Run Collection Summary", "Run Situational Analysis", etc.



A list of available analyses for a collection in Job Watcher.

Tip: The Analyses menu is also available if you select multiple collections and right-click. This will run the desired analyses on all the selected collections.

Analyses in iDoctor are SQL stored procedures. When an analysis runs it runs in the Remote SQL Statement Status View at the bottom of the main window. This window shows the progress of the analysis and from there completion or error messages will be shown. If an error is shown, you should right-click the error and use the Display job log menu to view the job log and try to determine the error. If you are unable to determine the reason for the error, send the job log and related information about the component, collection you are trying to analyze to iDoctor@us.ibm.com.



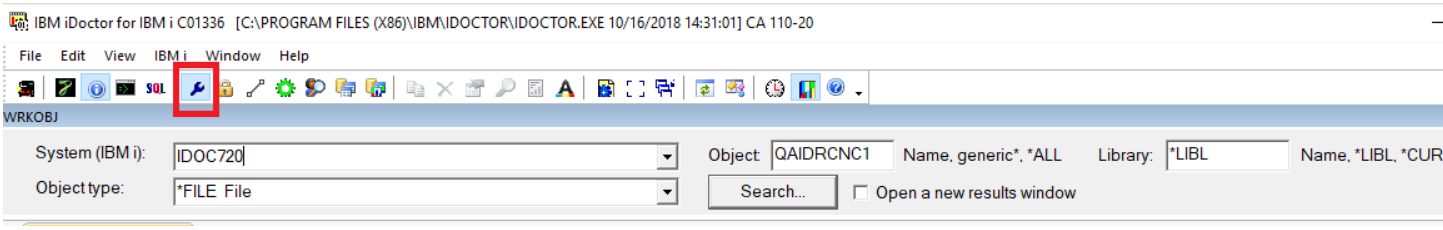
Remote SQL Statement Status View

After an Analysis is complete, the tables it creates can be accessed under the SQL Tables folder found in iDoctor under the component icon or under each library within the Libraries folder. Reports are often available from the SQL tables generated by the analysis. In many cases additional reporting options are also available under the collection, by right-clicking the collection and accessing a new menu option applicable to the new analysis.

4.18 WRKOBJ Pane

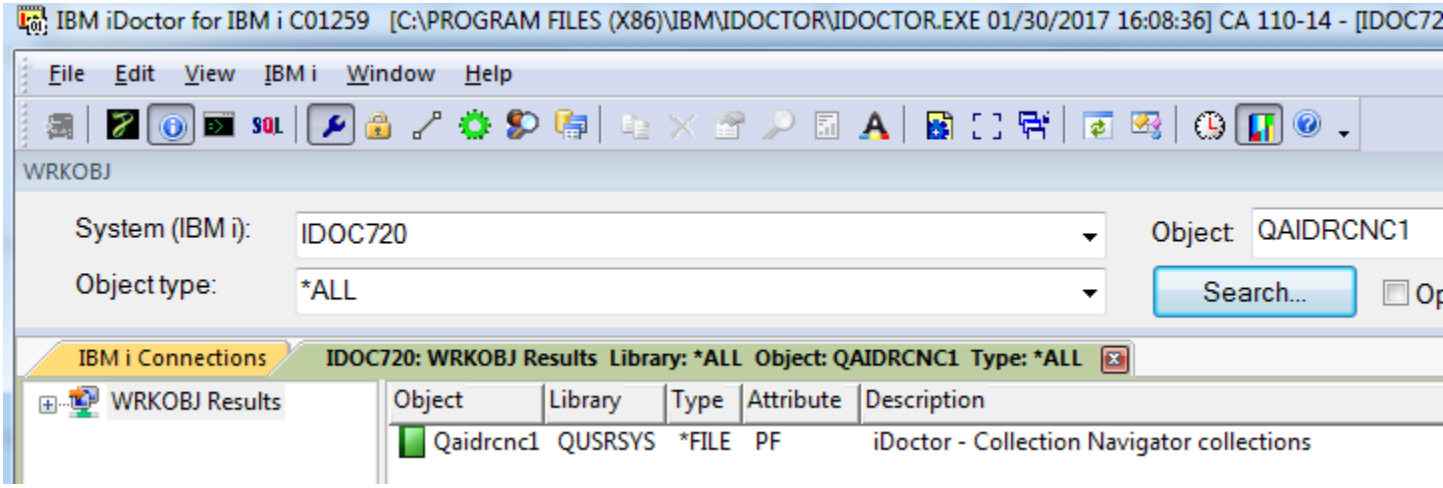
This pane is an interface for the WRKOBJ (Work with Objects) command and is accessible by clicking this icon  in the toolbar of the Main Window. It allows you to browse the objects and libraries on an IBM i based on the search criteria of your choice.

When pressing this button, the pane will appear near the top of the Main Window. Pressing the toolbar button again will dismiss the pane.



WRKOBJ Pane Example

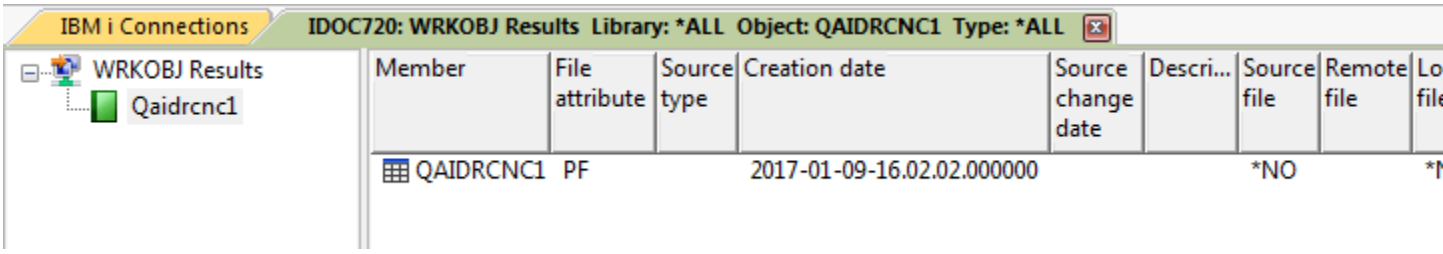
A user can specify the system, object or generic object name and library as well as the object type. Clicking search will display the results in a new window.



WRKOBJ Results View

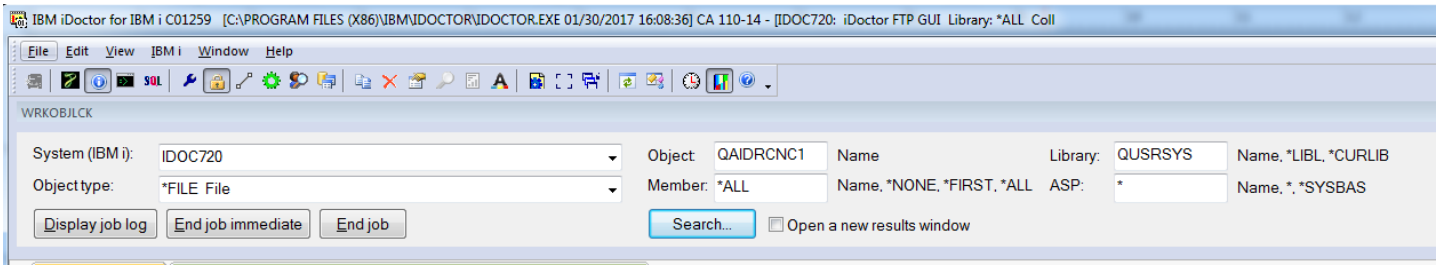
Some object types provide additional functionality and where applicable you will be able to drill down further and find more details about an object.

For example, physical files can be expanded to view the members within them. And the members can be opened as a new report in the Data Viewer.



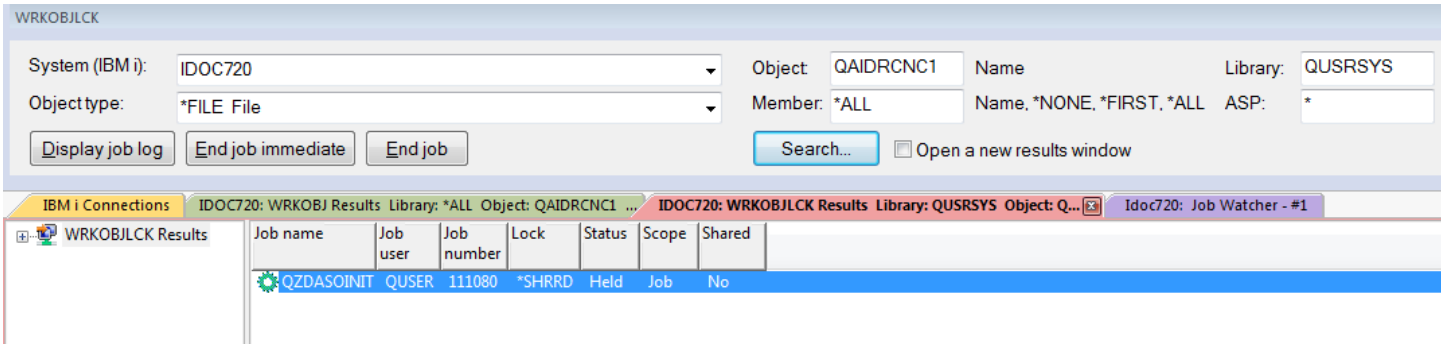
4.19 WRKOBJLCK Pane

This pane is an interface for the WRKOBJLCK (Work with Object Locks) command and is accessible by clicking this icon  in the toolbar of the Main Window. It allows you to see the jobs that have locks on the specified object and perform actions against those jobs. When pressing the button the pane will appear near the top of the Main Window. Pressing the toolbar button again will dismiss the pane.



WRKOBJLCK Pane Example

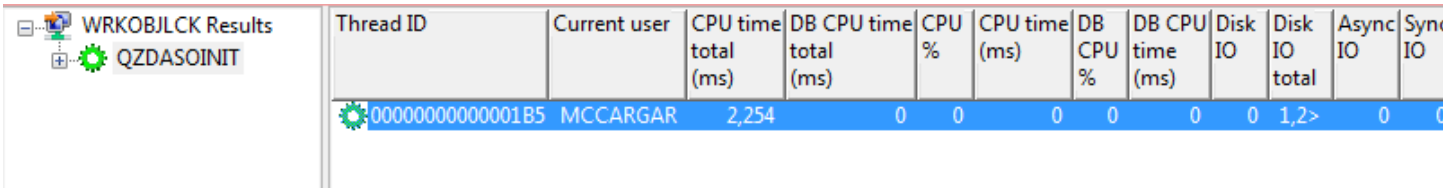
A user can specify the system, object, library as well as the object type to check. The member name (for files) and/or ASP may also be specified. Clicking search will display the results in a new window.



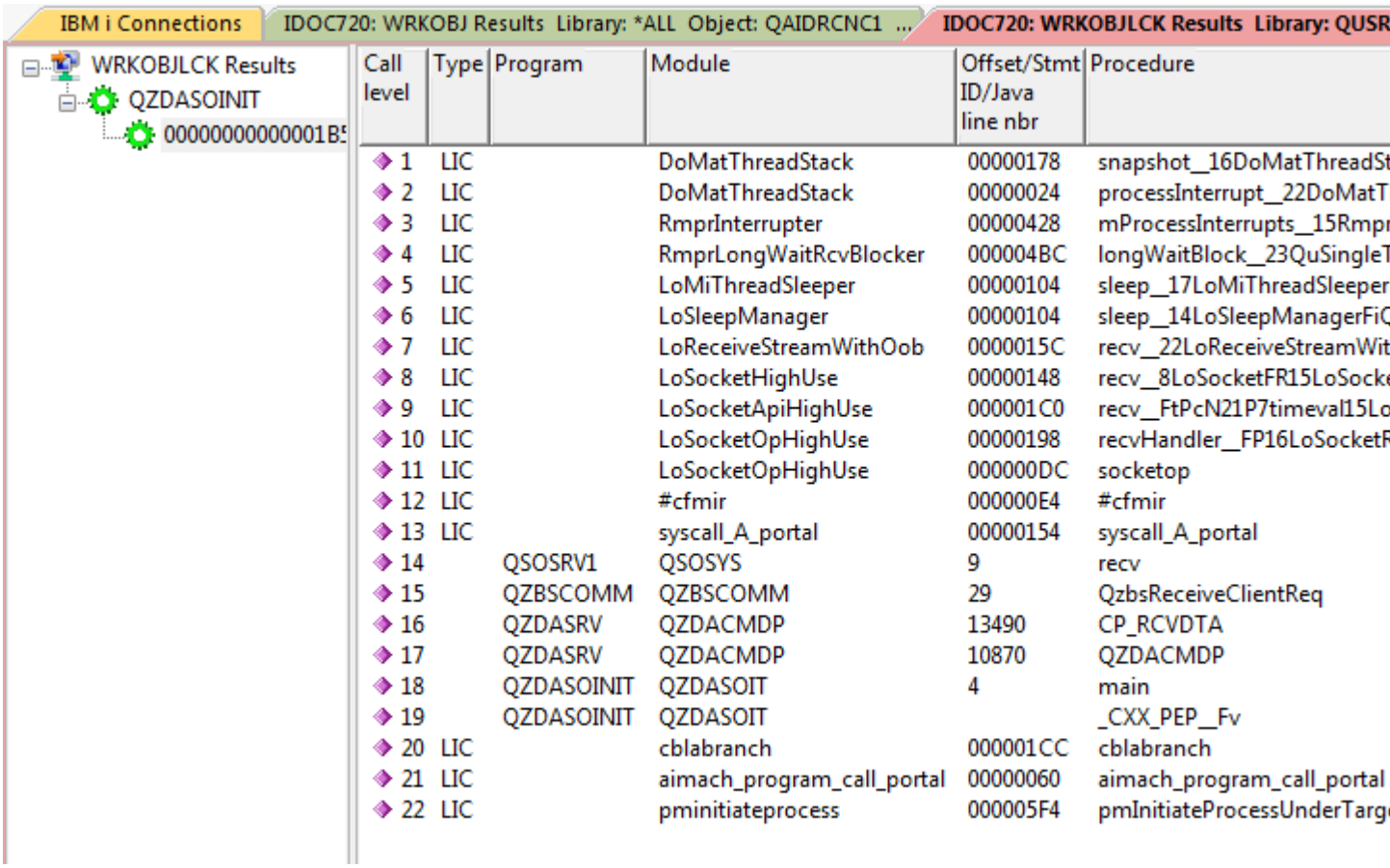
WRKOBJLCK Results View

By selecting a job and then pressing the desired button on the WRKOBJLCK Pane (End job, etc.) that action will be performed.

If any locks are found, then each job causing those locks will be listed. Jobs can be expanded to reveal the threads within them and expanding a thread reveals the call stack.



List of threads within a QZDASOINIT job in iDoctor

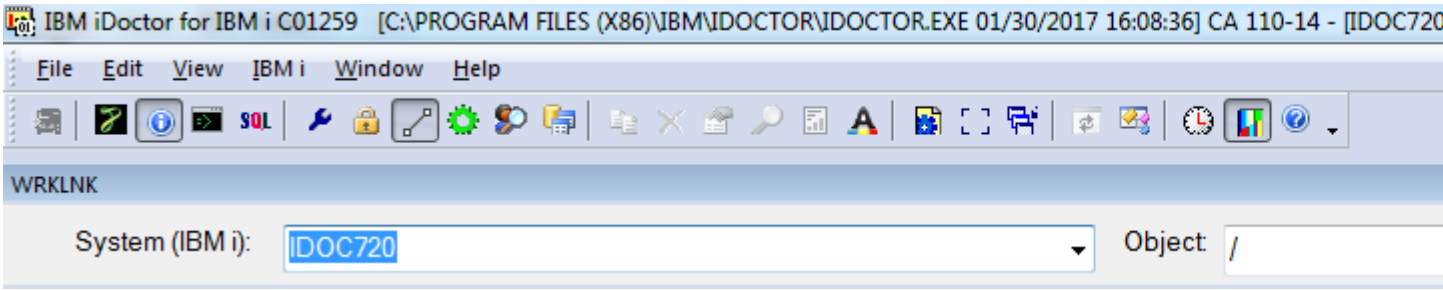


Call stack for a QZDASOINIT job/thread in iDoctor

For more information on the job/thread/call stack options in this view see the WRKACTJOB Pane section.

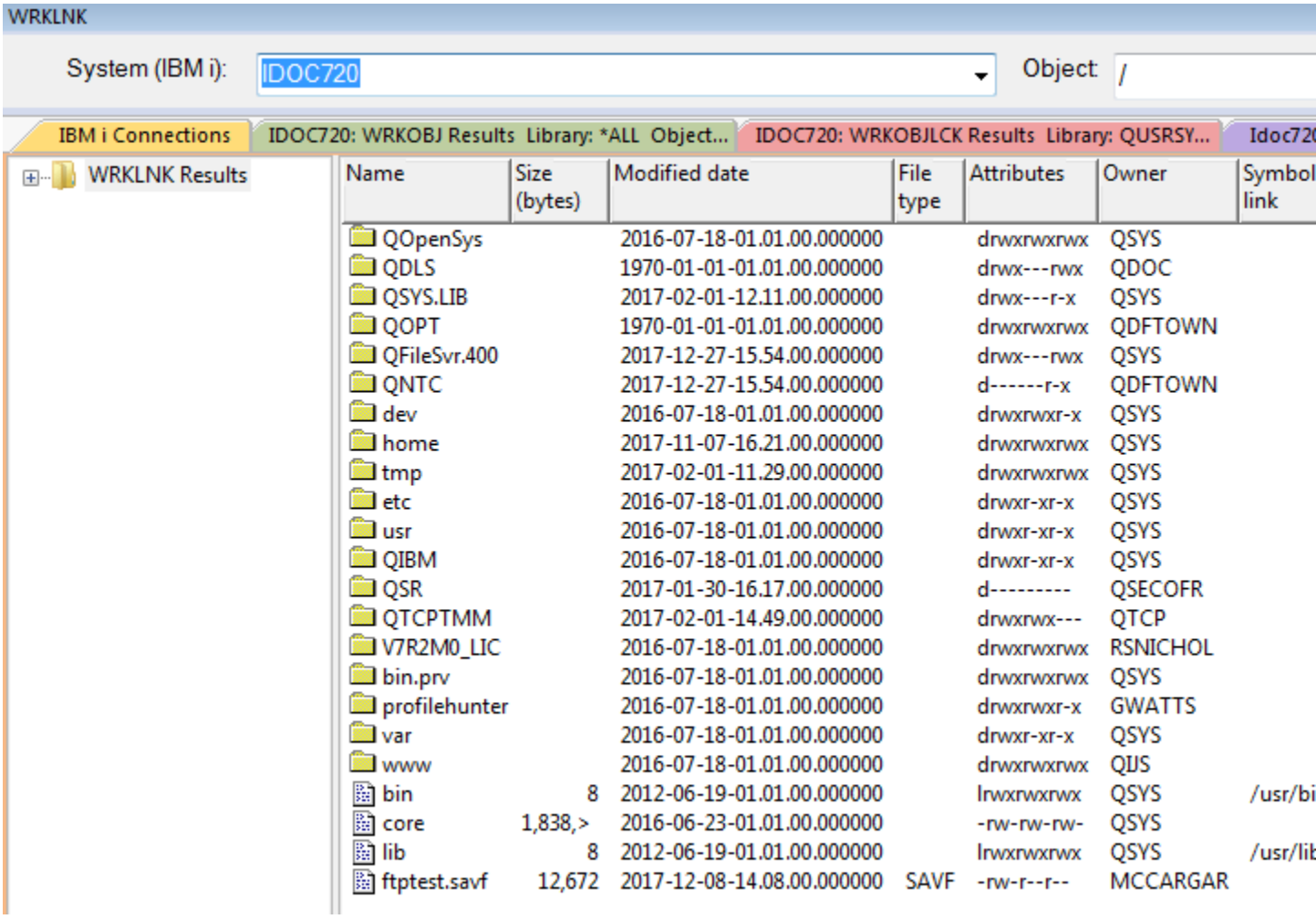
4.20 WRKLNK Pane

This pane is an interface for the WRKLNK (Work with Object Links) command and is accessible by clicking this icon in the toolbar of the Main Window. It allows you to browse the directories and files in the IFS that exist in the location you specify. When pressing the button, the pane will appear near the top of the Main Window. Pressing the toolbar button again will dismiss the pane.



WRKLNK Pane Example

A user can specify the system and the object (IFS path) to search. Clicking search will display the results in a new window.



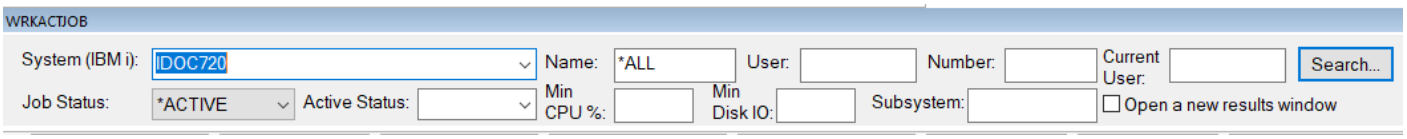
WRKLNK Results View

From this view a user can work with the files and directories shown in several ways. Files may be downloaded or opened/viewed. You may also upload additional files to the system by right-clicking one of the folders and using the Upload… menu.

4.21 WRKACTJOB Pane

This pane is like the interface for the WRKACTJOB to show the active jobs on the system but also provides the capability to view the job logs or spool files for jobs that have ended. For active jobs you can view the threads within each job and the call stacks for each thread. Options are also available to kick off performance collections or define definitions based on the selected active jobs.

The interface is accessible by clicking this icon  in the toolbar of the Main Window.

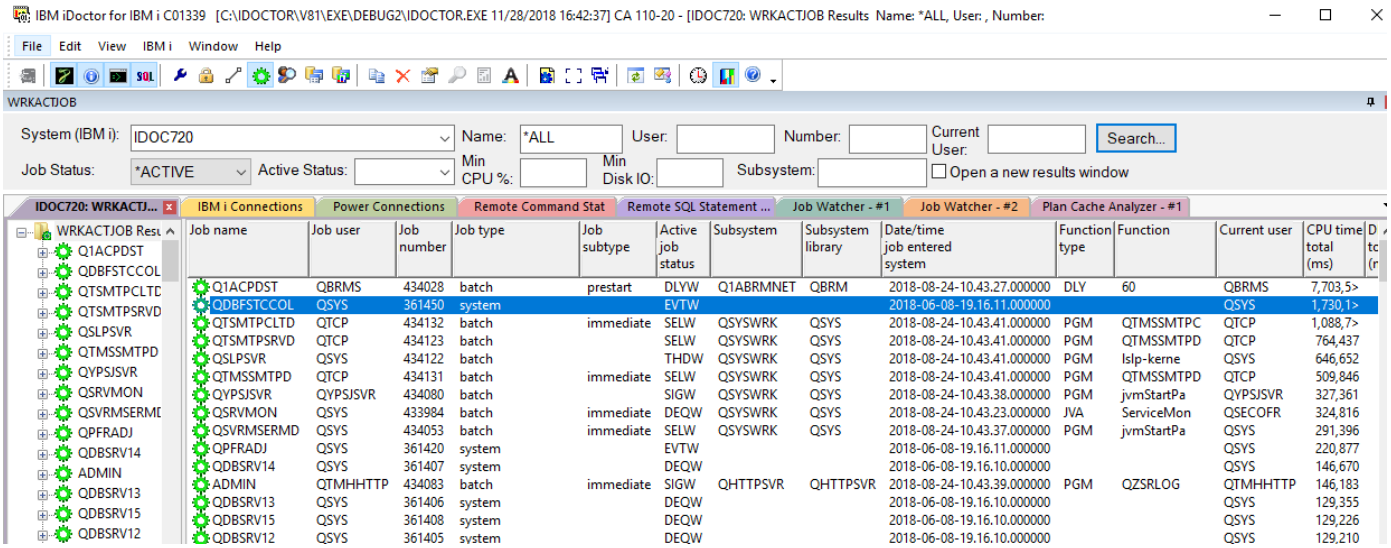


WRKACTJOB Pane Example

The following table describes the options available in this interface:

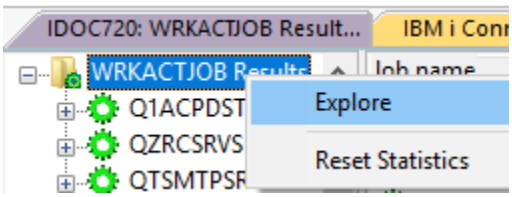
Option	Description
System (IBM i)	The name of the IBM i system to perform a search on. The values in the drop-down list is built from the list of systems within the IBM i Connections View.
Name	This is the job name or generic job name or *ALL to perform the search on.
User	This is the user name or generic user name (or *ALL) to use when performing the search.
Number	The exact 6-digit job number to include in the results or blank (or *ALL) for all.
Current User	The exact current user profile to include in the results or blank (or *ALL) for all. **Note:** This option only appears if the Job Status drop down value is *ACTIVE.
Job Status	This option allows you to indicate the types of jobs to include in the results based on their status. The option used effects the columns returned in the list. ***ACTIVE** – Only active jobs are returned. This will include some performance metrics. ***ALL** – All jobs, active or not will be returned in the list. ***JOBQ** – Only jobs waiting on a job queue will be shown in the list. ***OUTQ** – Only completed jobs containing spool file output are shown. You can expand these jobs to view these spool files.
Active Status	These indicates the type of active job status to further filter the results. You can pick one and only of these types in the list. For more information see the help text for the WRKACTJOB command's Status column. **Note:** This option will only appear if Job Status is *ACTIVE or *ALL.
Min CPU %	This is the desired minimum CPU % to use for filtering the results in the list. Only jobs having a "CPU %" column value greater than or equal to this number will be shown. **Note:** This option only appears if the Job Status drop down value is *ACTIVE.
Min Disk IO	This is the desired minimum Disk IO value to use for filtering the results in the list. Only jobs having a "Disk IO" column greater than or equal to this number will be shown. **Note:** This option only appears if the Job Status drop down value is *ACTIVE.
Subsystem	This is the subsystem name to use to filter the results by. **Note:** This option only appears if the Job Status drop down value is *ACTIVE.
Open a new results window	If checked a new WRKACTJOB search results view will be opened each time the search button is pressed. **Note:** By checking this option it makes the ability to get performance statistics updated on each search no longer usable.
Search	This will perform the job search based on the filters given on this pane. If job status is ***ACTIVE** then clicking this button multiple times will update the performance statistics each time the button is pressed.

4.21.1 Active jobs



WRKACTJOB Results View Example #1

Tip: This interface collects performance statistics for active jobs only. Each refresh or press of the search button will update these metrics. To reset the performance statistics counters then right-click the WRKACTJOB Results folder and use the Reset Statistics menu option.



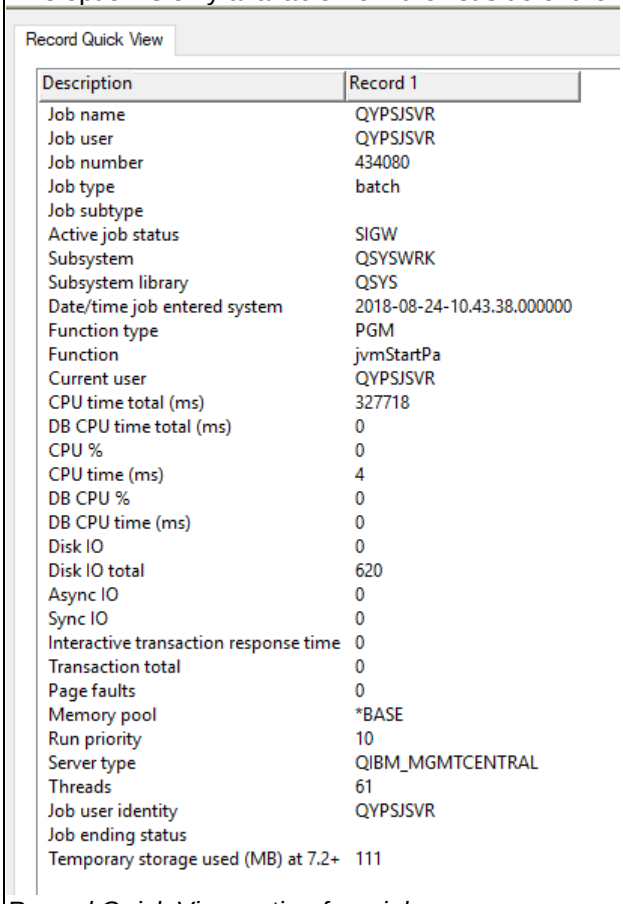
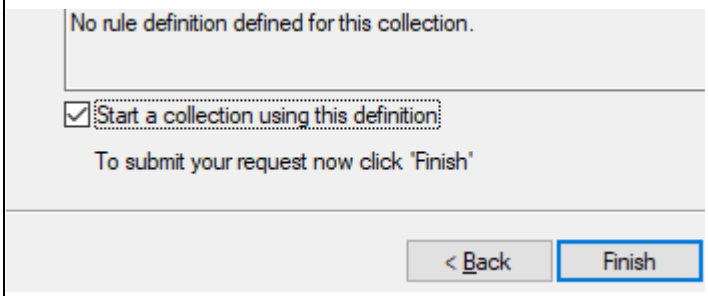
WRKACTJOB Results Folder Menu Options

Note: The columns displayed in the results will vary based on the job status value selected when performing the search.

The results folder contains a row for every job found on the system matching the filters. The following table describes the data shown in this folder if the Job Status option is *ACTIVE:

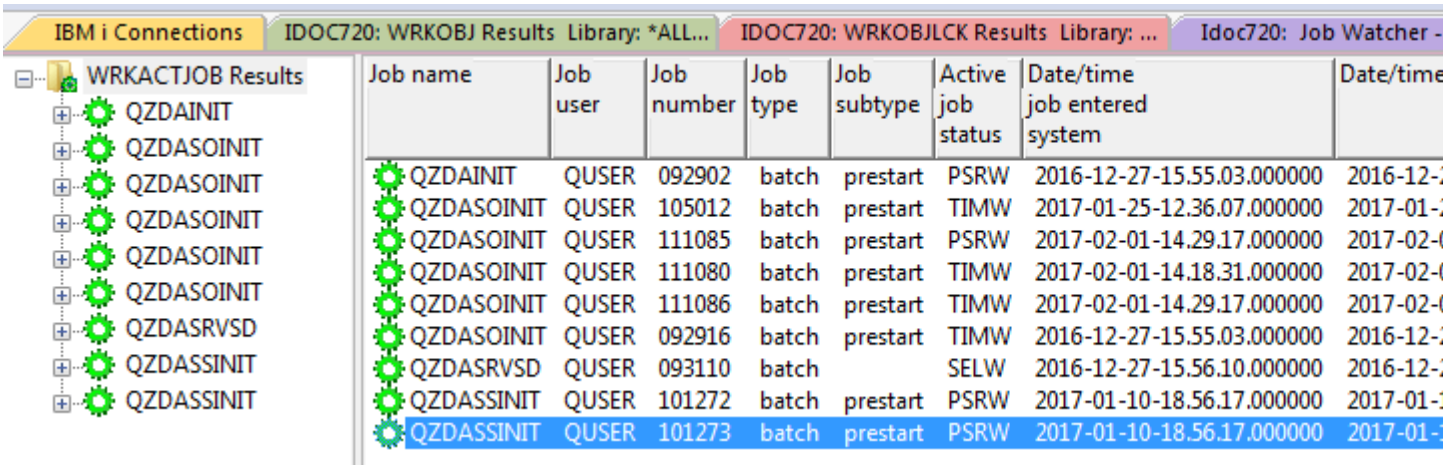
Field	Description
Job name	The job name
Job user	The job user name
Job number	The 6-digit job number.
Job type	This indicates the job type such as batch, system, autostart, etc.
Job subtype	This is the job subtype.
Active job status	The active job status. For more information see the help text for the WRKACTJOB command's Status column.
Subsystem	The name of the subsystem the job is running in.
Subsystem library	The library name for the subsystem the job is running in.
Date/time job entered system	This is the date and time when the job entered the system.
Function type	The current number of threads found within the job.
Function	The last high-level function initiated by the initial thread. This field is blank when a logged function has not been performed. The field is not cleared when a function is completed.
Current user	The current user profile the job.
CPU time total (ms)	Total CPU time consumed by the job (in milliseconds).
DB CPU time total (ms)	Total DB CPU time consumed by the job (in milliseconds).
CPU %	The percent of processing unit time attributed to this job over the elapsed time compared to the measurement time interval. For an uncapped partition using shared processors, this percentage can be greater than 100 percent.
CPU time (ms)	The CPU time used by the job during the elapsed time.
DB CPU %	The percent of processing unit type attributed to DB operations over the elapsed time compared to the measurement time interval.
DB CPU time (ms)	The DB CPU time used by the job during the elapsed time (in millseconds).
Disk IO	The number of disk I/Os that occurred for the job during the elapsed time.
Disk IO total	The total disk I/Os for the job.
Async IO	The number of asynchronous disk I/Os for the job during the elapsed time.
Sync IO	The number of synchronous disk I/Os for the job during the elapsed time.
Interactive transaction response time	The average interactive transaction response time (in milliseconds) for the life of the job.
Transaction total	The total number of interactive transactions during the life of the job.
Page faults	The number of page faults for the job during the elapsed time.
Memory pool	The name for the memory pool the job is running in.
Run priority	This is the run priority for the job. The smaller the value the higher priority the job has.
Server type	An identifier given to help identify the type of server job in more detail.
Threads	The number of threads the job contains. Expanding the job will show this number of threads.
Job user identity	The user profile name by which the job is known to other jobs on the system. The job user identity is used for authorization checks when other jobs on the system attempt to operate against the job.
Job ending status	The status of the job that is currently trying to end (if any.)
Temporary storage used (MB) at 7.2+	This value is the amount of temporary storage currently used by the job (in megabytes.) This column only applies when connected to a 7.2 or higher system.

By right clicking on an active job there are several options available:

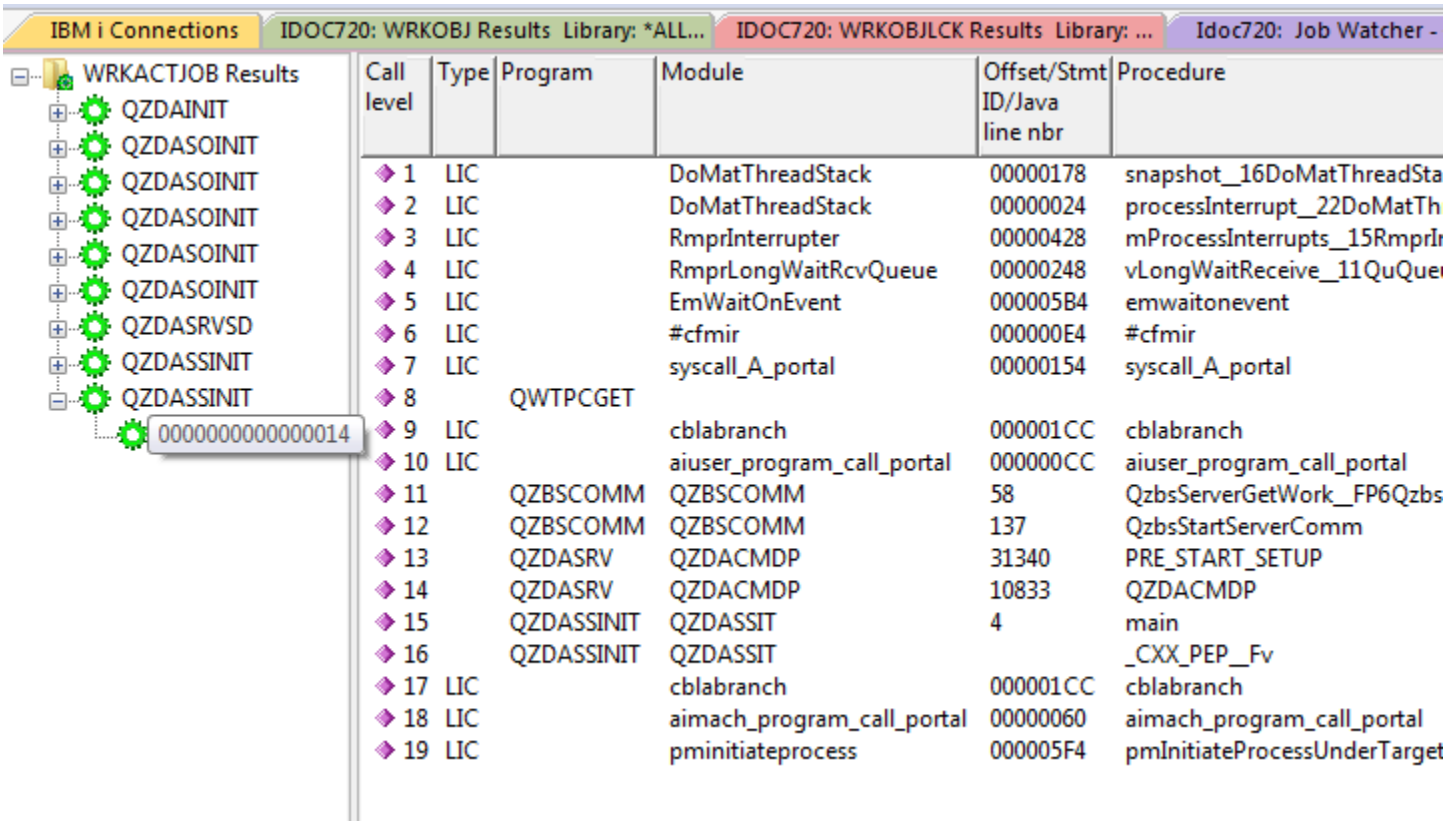
Popup Menu	Description
Explore	Displays the list of threads found within the job. The list of threads contains performance statistics that are updated on each refresh of the view. You can reset these statistics by using the Reset Thread Statistics menu option.
Record Quick View	Displays a vertical list of fields for the selected job in a new window. This option is only available from the list side of the view.  *Record Quick View option for a job*
Select fields…	This option allows you to select the fields to display for the list of threads within the job.
Add Job Watcher Definition	Defines a new Job Watcher definition that collects data only for the selected jobs using the Add Job Watcher Definition Wizard. One the final screen of the wizard be sure to check the box to indicate that you wish to start the collection now (if desired.)  *Add Job Watcher Definition Wizard – Finish screen*
Start PEX Collection	Launches the Start PEX Collection Wizard and preselects the selected jobs in the interface so that the collection only contains data for these jobs.
Add PEX Definition	Defines a PEX definition that will collect data over the selected jobs. You will need to indicate within the wizard if you want to have a *STATS or *TRACE type PEX

	collection.
Display Job Log	Displays the job log for the selected job.
End Job	This option will end the selected jobs using the OPTION(*CNTRLD) DELAY(30) parameters.
End Job Immediate	This option will end the selected jobs immediately.
Reset Thread Statistics	This option resets the collected thread level statistics for the selected job. This applies to the threads shown below the job when it is expanded in the tree.
Properties	This option displays the active job property pages for the selected job. This contains additional details about the job such as the call stack, and locks currently present.

When expanding an active job, the list of threads is shown within it. And expanding a thread will display its call stack.



WRKACTJOB Results View Example #2

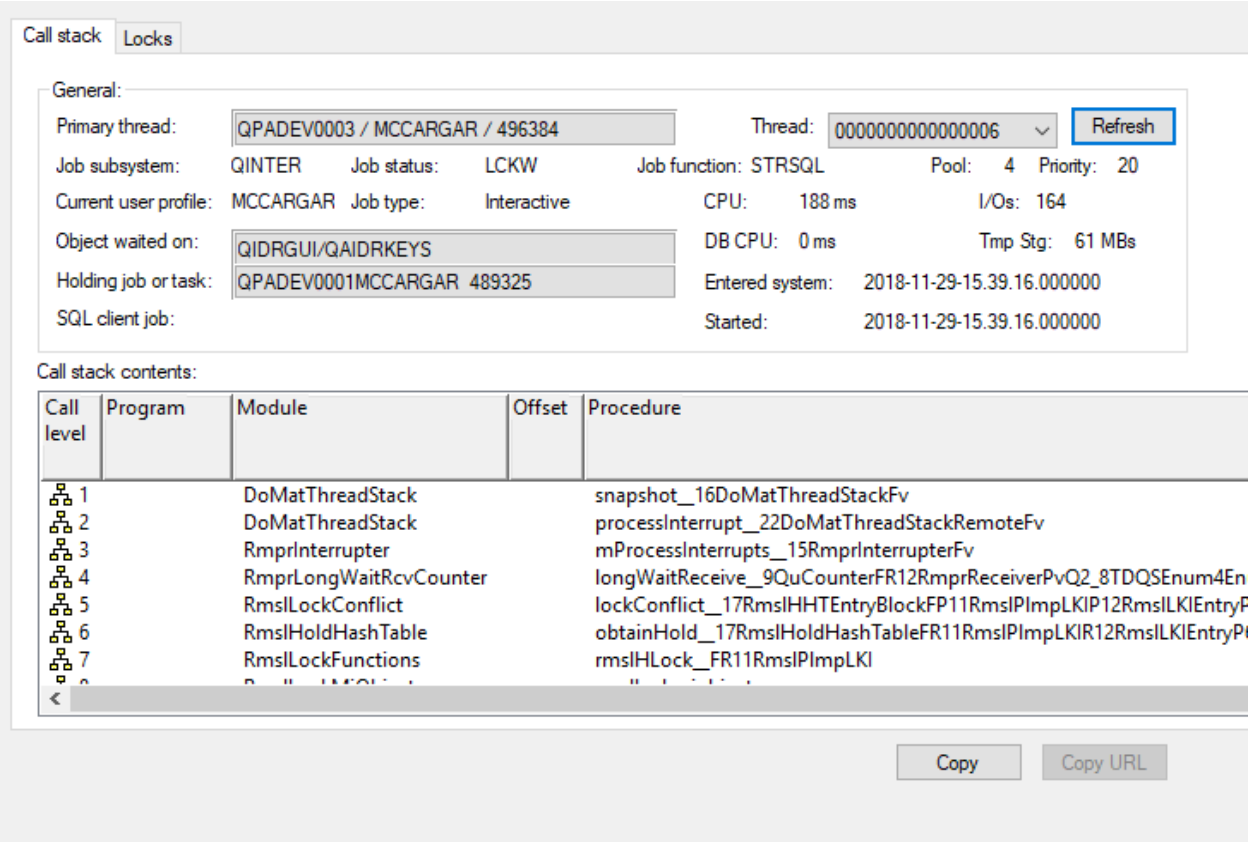


Call Stack within the WRKACTJOB Results View

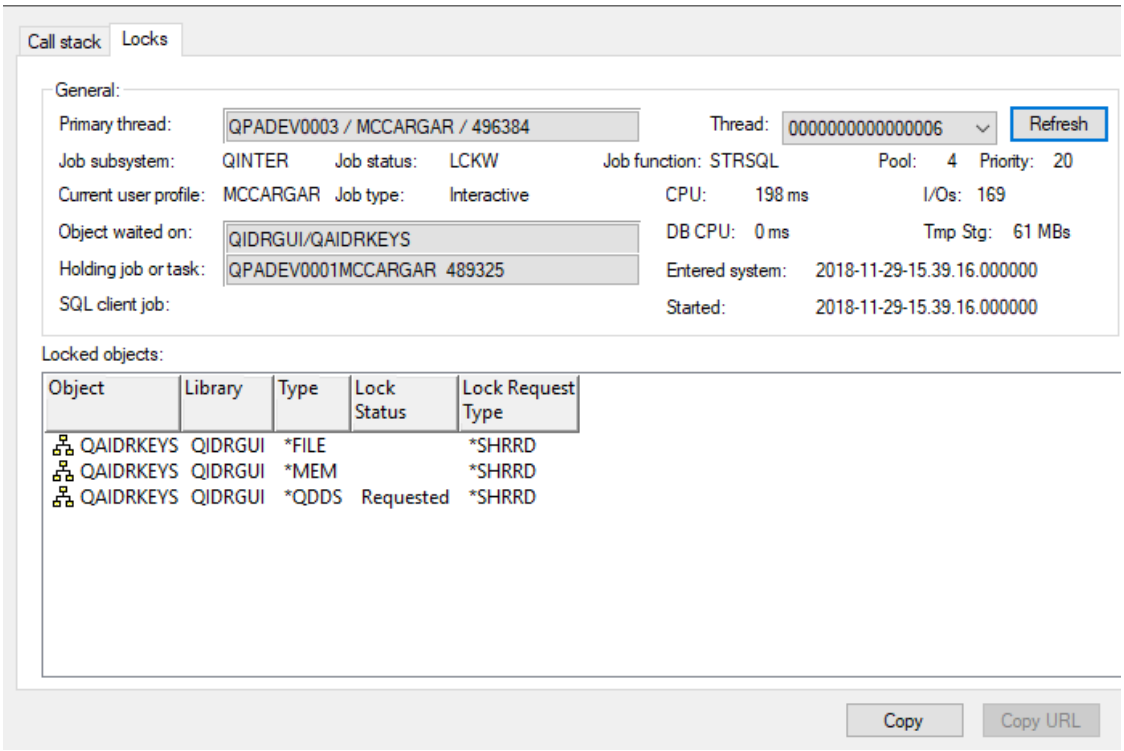
4.21.1.1 Properties

The properties for an active job consist of general information about the job at the top of the window for all tabs and additional information that varies for each individual tab selected.

These screens are designed to be like the ones found in Job Watcher for a given job/interval. They provide the wait object and holder information if one exists for the current job.



Active job properties – Call stack



Active job properties - Locks

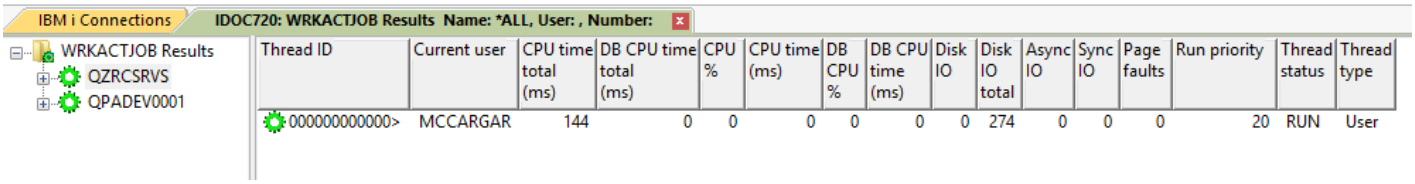
4.21.1.2 Threads

Expanding a job within the WRKACTJOB Results folder displays the list of threads the job contains along with performance data from the 1st snapshot and the last snapshot (refresh) of the current view.

Performance data shown in this list like CPU and IOs are based on the time between the initial refresh of this view and the most recent one. You can also reset these statistics by right-clicking the job folder and using the Reset statistics menu.

Expanding a thread allows you to easily see its current call stack.

This interface looks like the following:



WRKACTJOB Results – Threads for a Job

The folder contains a row for every thread found within the selected job from the tree. The following table describes the data shown in this folder:

Tip: Right-click the job and pick the Select fields menu to rearrange or add/remove fields from this list.

Field	Description
Thread ID	Unique identifier for the job's thread.
Current user	The user profile that the thread for which information is being retrieved is currently running under. This name may differ from the user portion of the job name.
CPU time total (ms)	Total CPU time consumed by the job (in milliseconds).
DB CPU time total (ms)	Total DB CPU time consumed by the job (in milliseconds).
CPU %	The percent of processing unit time attributed to this thread over the elapsed time compared to the measurement time interval. For an uncapped partition using shared processors, this percentage can be greater than 100 percent.
CPU time (ms)	The CPU time used by the thread during the elapsed time.
DB CPU %	The percent of processing unit type attributed to DB operations over the elapsed time compared to the measurement time interval.
DB CPU time total (ms)	Total DB CPU time consumed by the job (in milliseconds).
Disk IO	The number of disk I/Os that occurred for the job during the elapsed time.
Disk IO total	The total disk I/Os for the job.
Async IO	The number of asynchronous disk I/Os for the job during the elapsed time.
Sync IO	The number of synchronous disk I/Os for the job during the elapsed time.
Page faults	The number of page faults for the job during the elapsed time.

Run priority	The current run priority for the job.
Thread status	The current status of the thread. The status of a thread may be one of the following values: *Blank* The status of the thread is unknown. *CMTW* The thread is waiting for the completion of save-while-active checkpoint processin job. This wait is necessary to prevent a partial commitment control transaction fro saved to the media. *CNDW* The thread is waiting for a condition. *DEQA* The thread is waiting for completion of a dequeue operation in the pool activity lev *DEQW* The thread is waiting for completion of a dequeue operation. For example, a serve for work by waiting for a dequeue operation *EVTW* The thread is waiting for an event. *HLD* The thread is in a job that is being held. *HLDT* The thread is being held. *INEL* The thread is ineligible and not currently in the pool activity level. *JVAA* The thread is waiting for completion of a Java program operation in the pool activi *JVAW* The thread is waiting for completion of a Java program operation. *LCKW* The thread is waiting for a lock. *LSPA* The thread is waiting for a lock space to be attached while in a pool activity level. *LSPW* The thread is waiting for a lock space to be attached. *MTXW* The thread is in a mutex wait. A mutex is a synchronization function that is used t multiple threads to serialize their access to shared data. *RUN* The thread is currently running in the activity level. *SELW* The thread is in a select wait. More information on the select() function is in the S chapter in the System API Reference, SC41-5801. *SEMW* The thread is waiting for a semaphore. A semaphore is a synchronization functio to allow multiple jobs or threads to serialize their access to shared data.

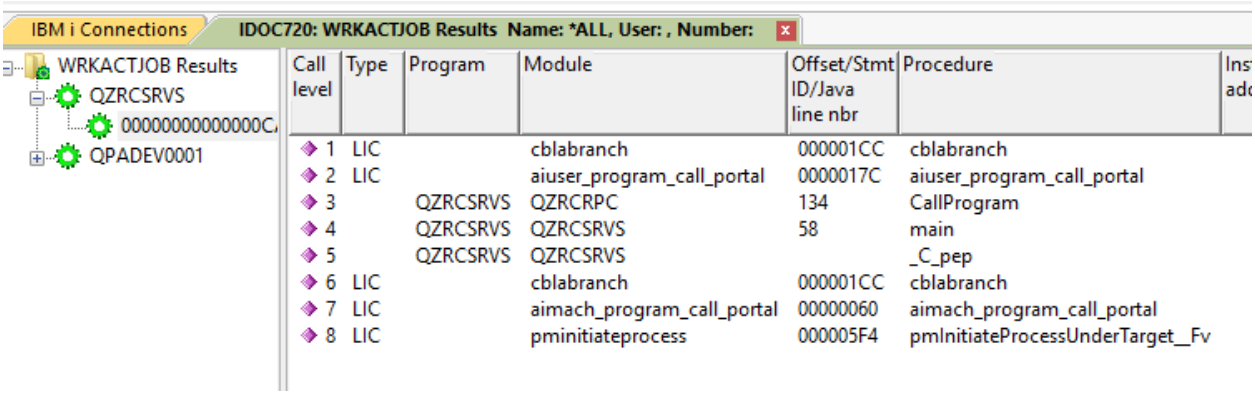
	SIGS The thread has been held by a signal. *SIGW* The thread is waiting for a signal. *THDW* The thread is waiting for another thread to complete an operation. *TIMA* The thread is waiting, in the activity level, for a time interval to end. *TIMW* The thread is waiting for a time interval to end.
Thread type	The thread type indicates how the thread was created. If this field is requested for a job, the value for the initial thread of the job will be returned. The type of a thread may be one of the following values: User - The thread was created either as the initial thread of the job or explicitly by the application. System - The thread was created by an operating system function.

By right clicking on a thread there are several options available:

Popup Menu	Description
Call Stack	Displays the current call stack for the selected thread.
Record Quick View	Displays a vertical list of fields for the selected thread in a new window. This option is only available from the list side of the view.
Select fields…	This option allows you to select the fields to display for the call stacks within the thread.

4.21.1.3 Call Stacks

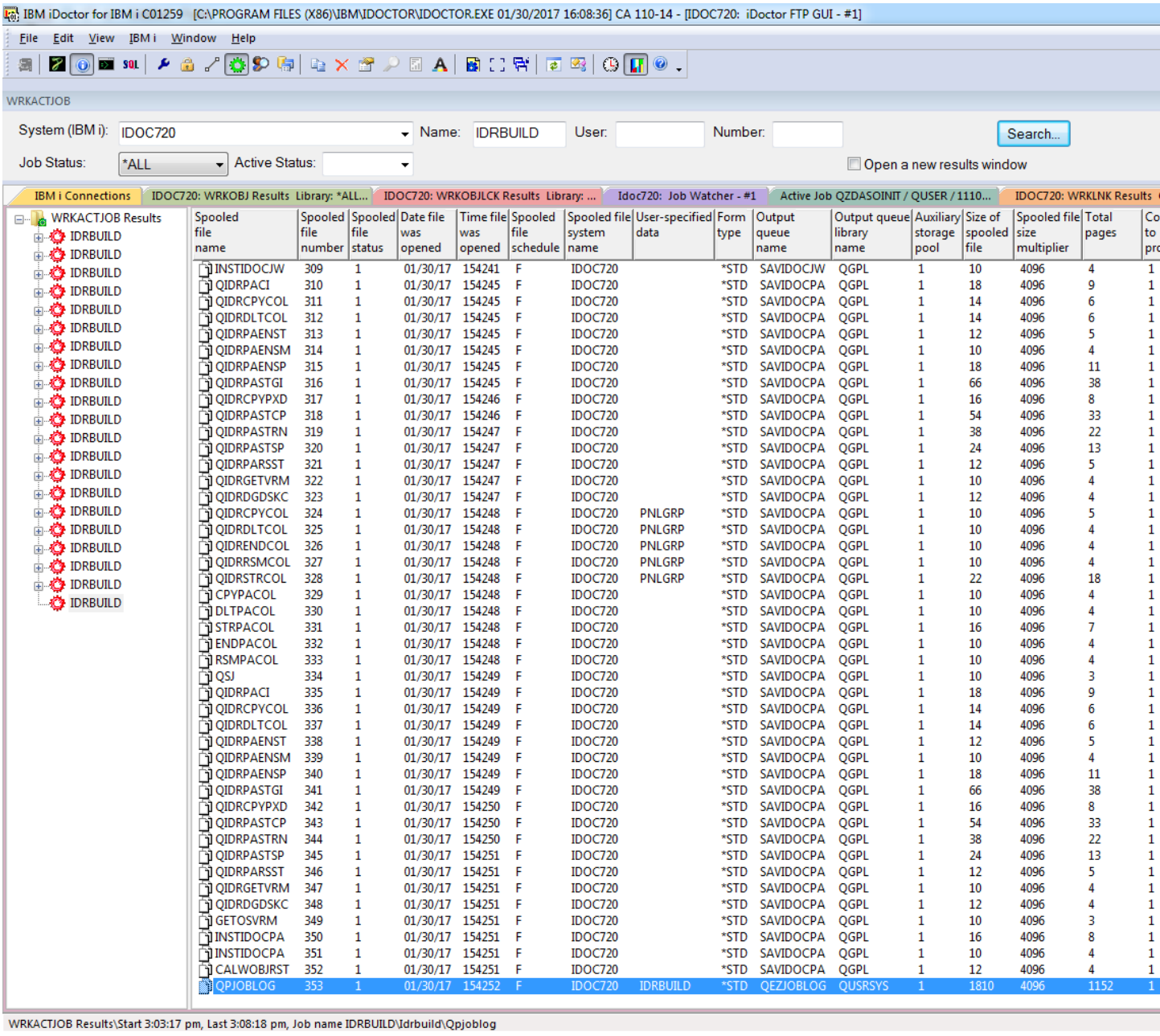
Expanding a thread within a job displays the current call stack for the thread.
The call stack shows both LIC and above the MI programs/procedures/modules.



WRKACTJOB Results – Call Stack for a Thread

4.21.2 Completed jobs

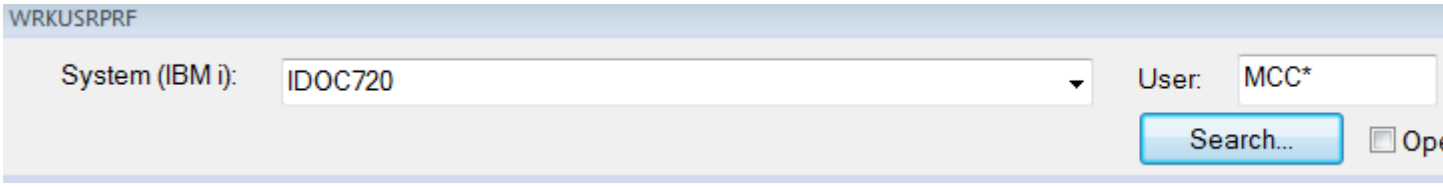
Expanding a completed job in the list will display a list of any spool files they contain.



Spool files within a completed job in the WRKACTJOB Pane

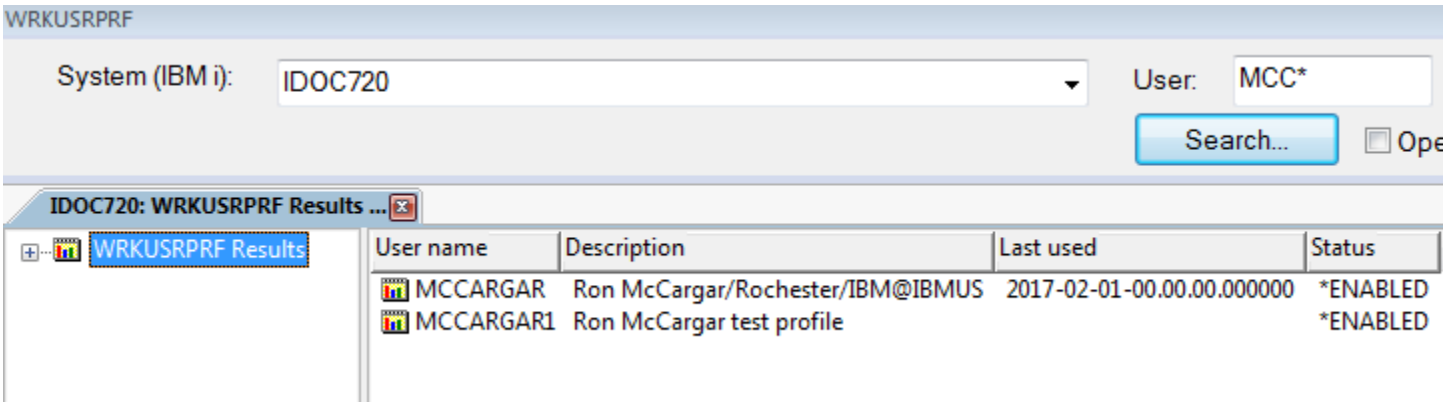
4.22 WRKUSRPRF Pane

This pane is an interface for the WRKUSRPRF (Work with User Profiles) command and is accessible by clicking this icon  in the toolbar of the Main Window. It allows you to browse the user profiles that exist on the IBM i and make some modifications as needed. When pressing the button the pane will appear near the top of the Main Window. Pressing the toolbar button again will dismiss the pane.



WRKUSRPRF Pane Example

After specifying the system name and user name criteria, clicking search will display the results in a new window.

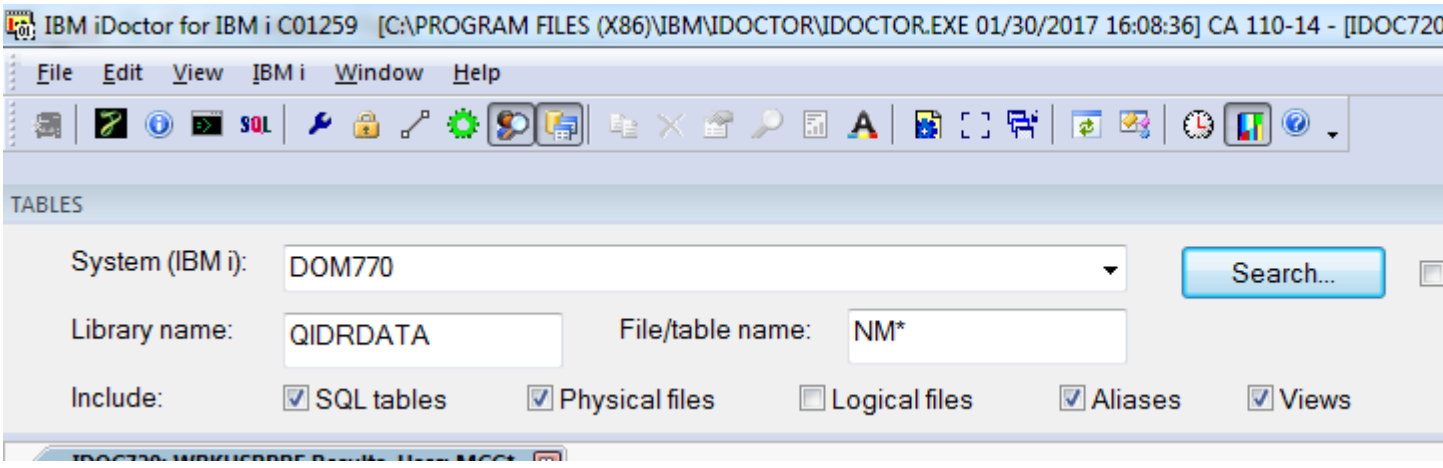


WRKUSRPRF Results View

From this view a user may create additional user profiles, change, delete or copy user profiles. Right-click a user profile for more options.

4.23 TABLES Pane

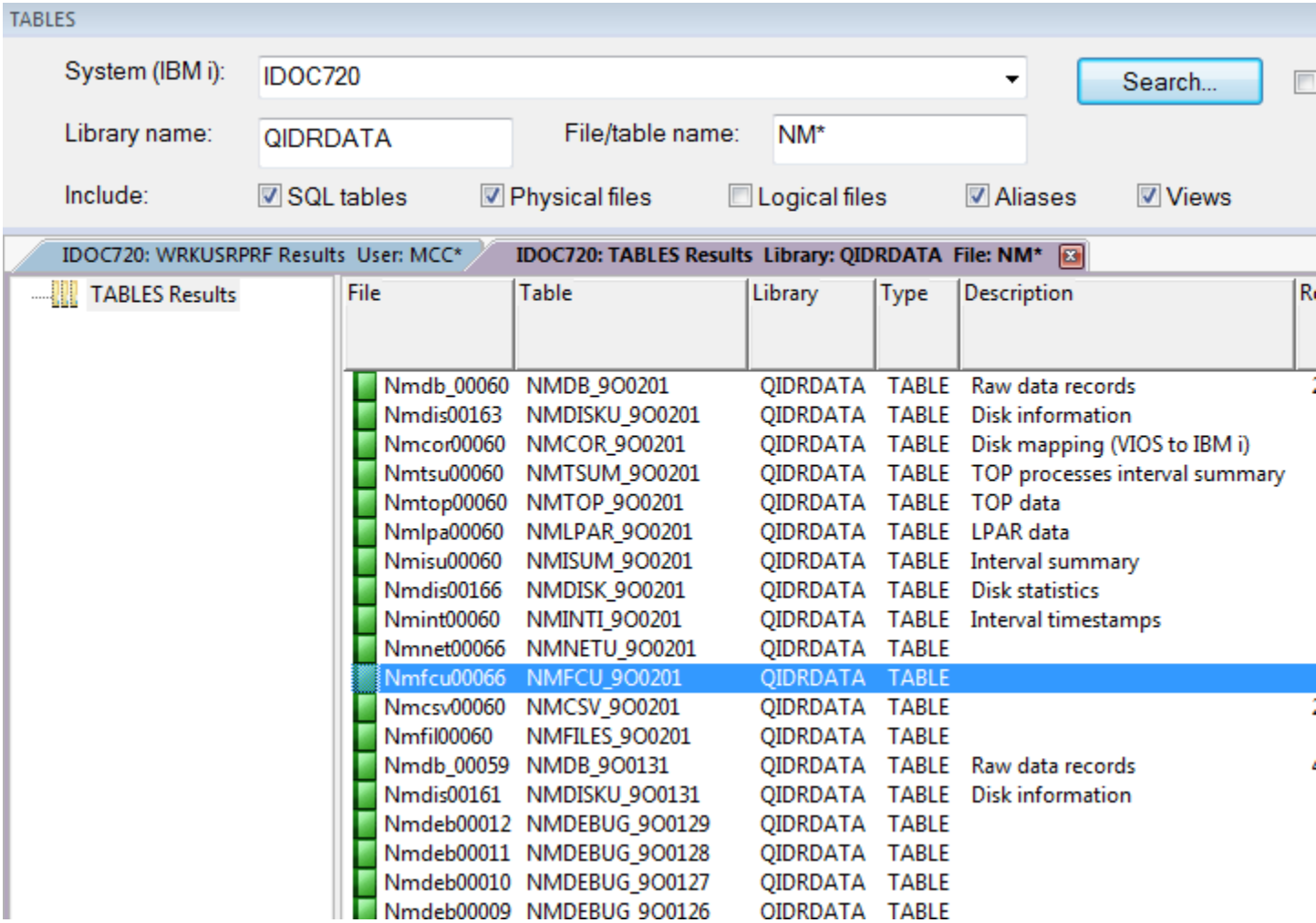
This pane allows you to browse the IBM i physical files, logical files and/or SQL created tables that exist on the system. Click the  icon in the toolbar of the Main Window to display the pane near the top of the Main Window. Pressing the toolbar button again will dismiss the pane.



TABLES Pane Example

A user may specify the system, library and generic table name to search. The type of objects returned may also be specified (SQL tables, physical files, logical files, aliases or views.)

Clicking search will display the results either in a new window or an existing one depending on if the “open a new results window” checkbox is checked.



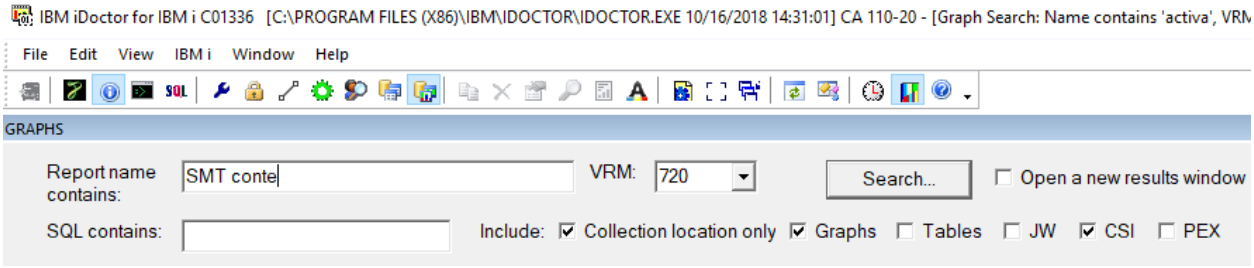
TABLES Results View

Double-click an SQL table to open it in the Data Viewer. Additional options are available by right-clicking an item displayed in the list (Delete, Properties, etc.)

4.24 Graph Search Pane

This pane allows you to browse or search the iDoctor report databases for graphs or reports of interest. This provides information such as the folder in which the graph or report is located, the VRMs of IBM i where the report will exist as well as any required files or PTFs needed for the report to appear.

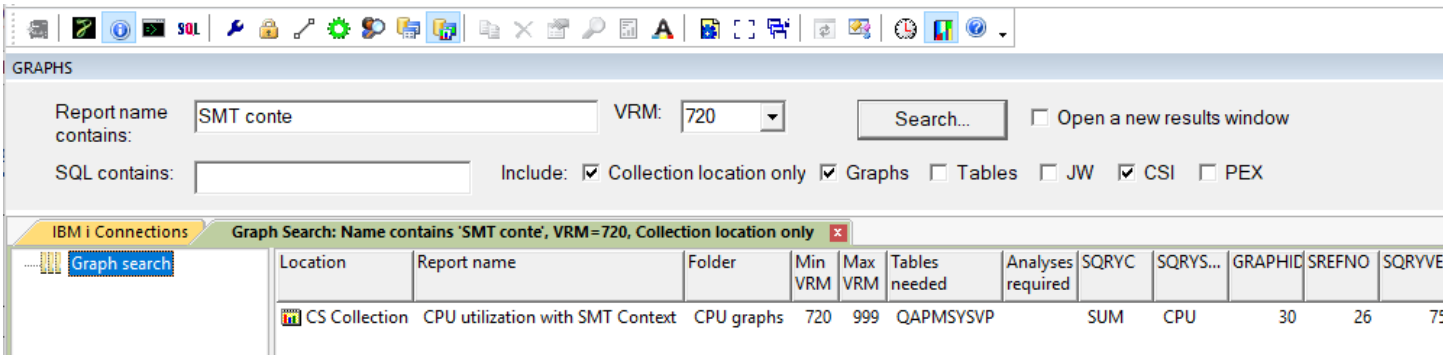
Click the  icon in the toolbar of the Main Window to display the pane near the top of the Main Window. Pressing the toolbar button again will dismiss the pane.



Graph Search Pane Example

A user may specify the report name filter, the VRM or specify something to look for in an SQL statement like a filename. You can also specify options such as which components to search. Only Job Watcher (JW), Collection Services Investigator (CSI) and PEX Analyzer (PEX) are currently supported by this interface.

Clicking search will display the results either in a new window or an existing one depending on if the "open a new results window" checkbox is checked.



Graph Search Results View

This list includes the following columns described in the table below:

Column	Description
Location	Indicates where in iDoctor you can find this graph or report. The component identifiers are either (CS, JW or PEX). For example: CS collection = Collection Services Investigator in the folder name shown in the Folder column directly under the collection. In some cases, graphs or reports are only found by drilling down from another graph and this will be noted here if this is required.
Report name	The graph or report name
Folder	The folder name and sometimes sub folder as well to find this graph or report in.
Min VRM	The minimum IBM i release in nnn format such as 610, 710, 720, 730, etc where this report will appear. The minimum release currently functions at is V6R1 (i.e. 610)
Max VRM	The maximum IBM i release in nnn format this report should appear at. Use a value of 0 if no max.
Tables needed	A list of “key” tables needed for the graph or report to appear. The core files of the component are also needed but not included here for simplicity.
Analyses required	This column indicates if any iDoctor analyses must be ran before this graph or report will appear.
SQRYCAT	Identifies the folder within the component that this graph or report appears in (within the .mdb iDoctor databases such as iDocCS.mdb.)
SQRYCATSUB	Identifies the sub folder within the component that this graph or report appears in (within the .mdb iDoctor databases such as iDocCS.mdb)
GRAPHID	Identifies the unique identifier for this graph within table QAIDRGPH in the iDoctor reports database.
SREFNO	Identifies the unique identifier for the SQL statement behind this report in table QAIDRSQL in the iDoctor reports database.
SQRYVER	This is used internally within iDoctor’s code to identify different types of reports based on which files exist.

Double-click an item in this list will display the iDoctor graph definition properties behind it. Because these are all iDoctor-defined reports, these properties are all read-only!

4.25 Collections database

The iDoctor collections database is an optional feature that scans a system and identifies the following types of collections that exist:

- PEX
- Job Watcher
- Collection Services (CSI)
- Disk Watcher

The database resides in the following tables in QUSRSYS:

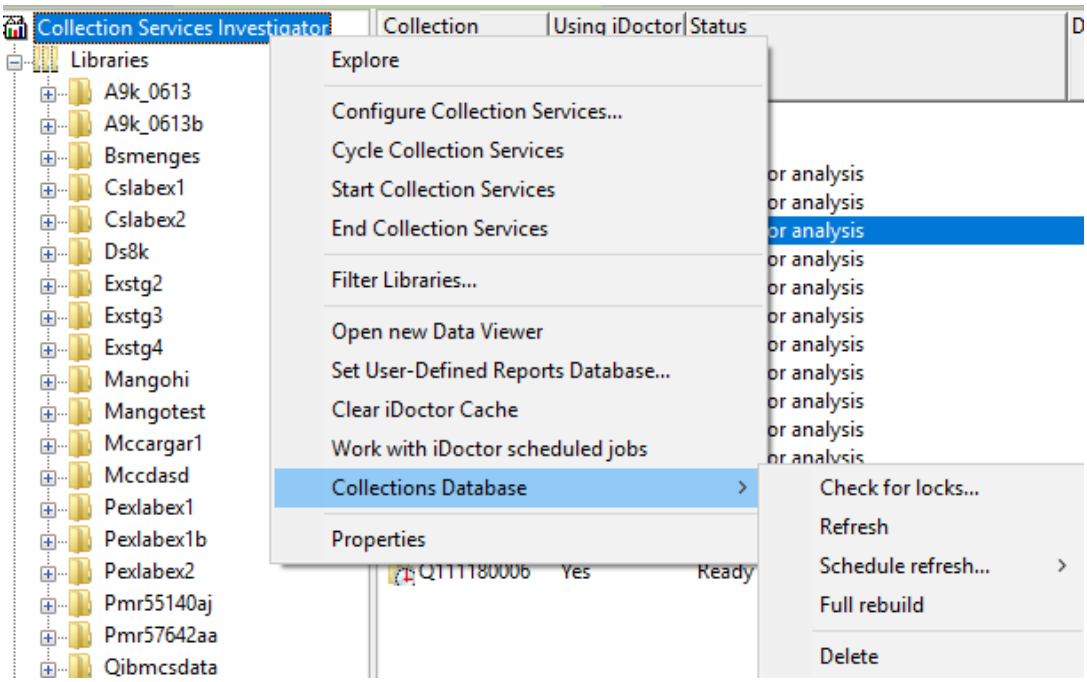
Table	Description
QAIDRCNL1	This file contains 1 record per library per collection type.
QAIDRCNC1	This file contains 1 record per collection found.

The database provides the ability to:

Drill down in the overview graphs in CSI or Job Watcher to drill down into another component’s data for the same system in the same time period. This is typically used to drill down from CSI to Job Watcher and/or PEX.

More quickly browse the collections on the system in the Libraries folder if the **Preference -> Misc -> Use the collections database to see which collections exist** is checked. This will be faster but has the disadvantage that the collection information may not be up to date.

The database is refreshed either manually or on a schedule that you can specify. The collections database is controlled via the Collections database menu found from any component that supports it (i.e. Job Watcher icon in Job Watcher).



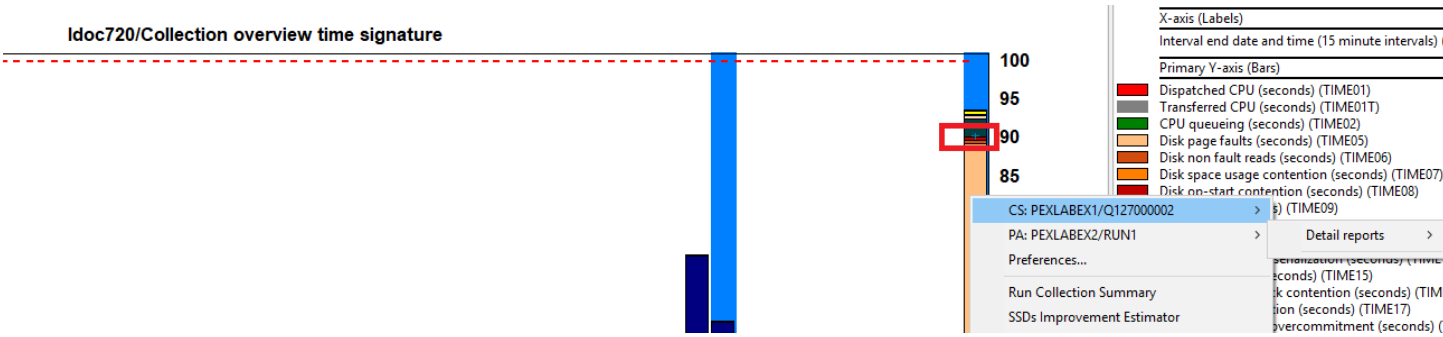
Collections Database menu options

Menu	Description
Check for locks	This opens the WRKOBJLCK pane to check for any locks on the database. Typically, if any user has a CSI/JW Collection overview time signature graph open a lock will exist and these graphs and/or jobs will need to be closed before rebuilding the database.
Refresh	This will check for new data and refresh the database. This is faster than the full rebuild option but will not recreate the database files. In some cases, if the database file formats have changed or errors involving the database persist, then the Full rebuild option is required.
Schedule refresh	This option allows you to schedule a daily refresh of the database. For this to work correctly, then no locks should exist on the database files at the indicated time.
Full rebuild	This option will delete and fully recreate the database. Use the Check for locks option before using this.
Delete	This will delete the Collections database files. If you use this option, you should do a full rebuild afterwards or some functions may no longer work correctly.

4.25.1 Intra-component drill down support

Some overview graphs in Job Watcher and Collection Services Investigator use the Collections database to identify and drill into other types of data. If a drill down into another type of component is available a "widget" is shown on the graph which is special shape. Then the menu option will contain options for the current component and an option for each available additional component you can drill into for this system and time period.

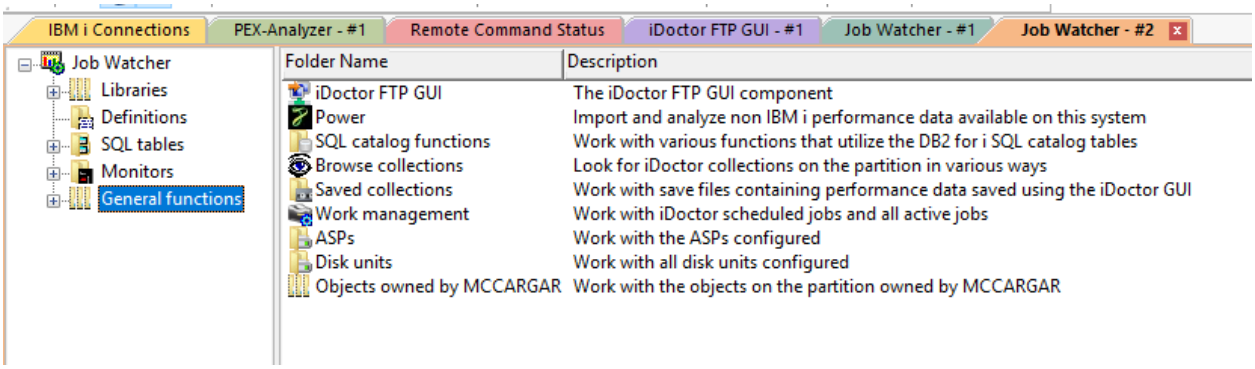
In the following example CSI graph the plus sign (within the red box) indicates a drill down into PEX Analyzer is available and the normal drill downs into CSI are available under the CS menu. The collection name for each is also listed in the menu.



CSI Collection overview time signature showing intra-component drill down into PEX Analyzer.

5 IBM i General Functions

This section documents the functions available within the General Functions folder in the IBM i components: Job Watcher, Collection Services Investigator, Disk Watcher or PEX Analyzer. Many different types of functions that are not commonly needed are grouped together here in this folder.



General functions folder in Job Watcher

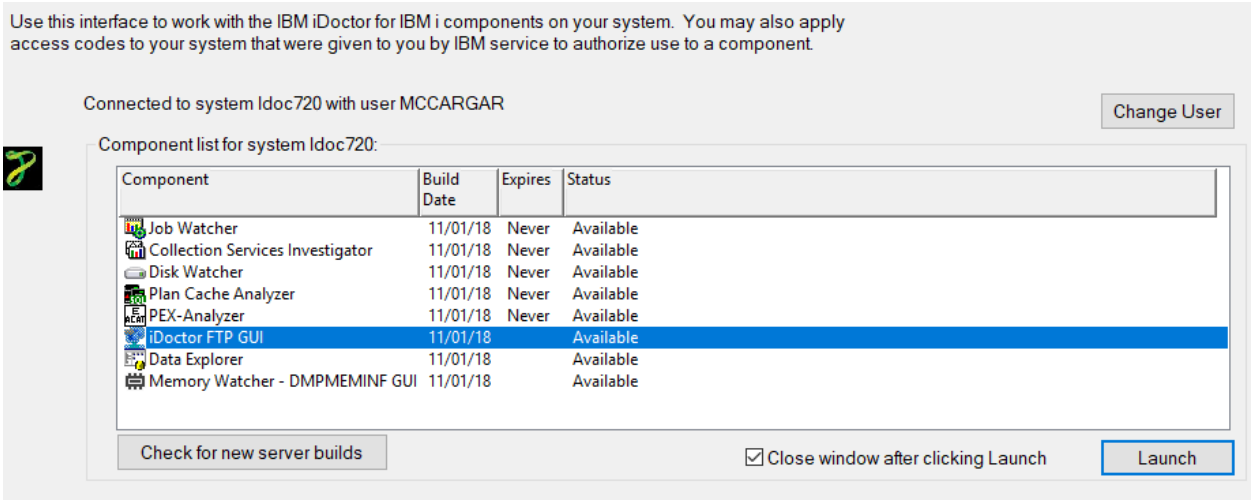
5.1 iDoctor FTP GUI

This section covers the iDoctor FTP GUI sub component. This provides functions to access both the IFS and objects within libraries on the IBM i you are connected to.

This section covers the interfaces specific to IBM i connections. For Power (non-IBM i connections), the interface to work with file systems/directories is discussed in Chapter 7.5 Power -> General Functions.

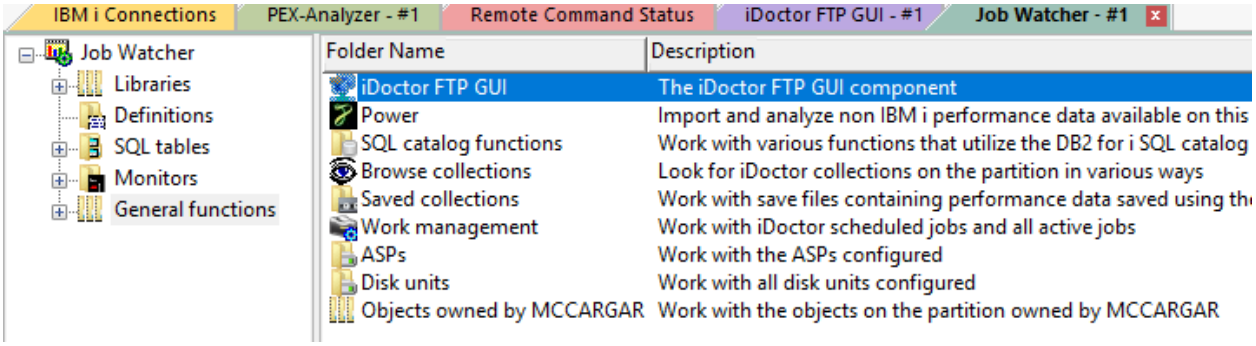
This interface can be launched in two ways:

1) Double-click the IBM i connection and it will exist in the iDoctor components window:



iDoctor Components Window

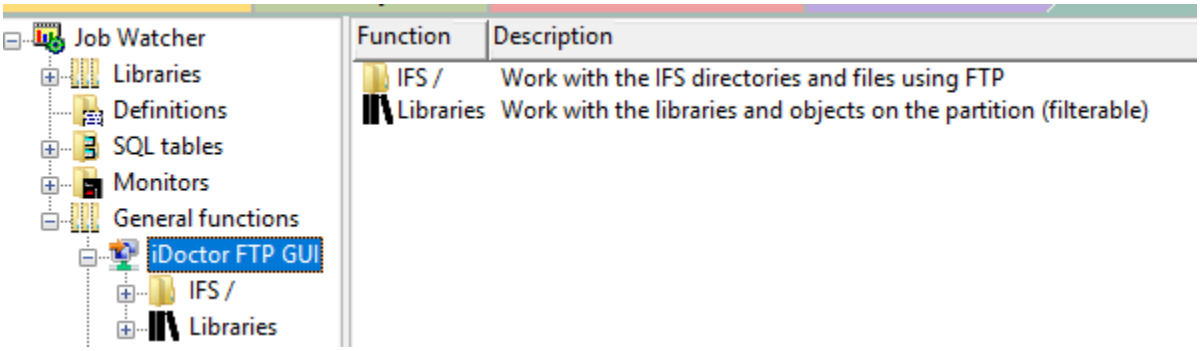
2) From the General functions -> iDoctor FTP GUI within the components: Collection Services Investigator, Job Watcher, Disk Watcher or PEX Analyzer.



Job Watcher -> General functions -> iDoctor FTP GUI folder

Note: The type of FTP connections used within this folder is based on the IBM i connection settings within the IBM i Connections View.

The iDoctor FTP GUI folder contains an option to view the IFS or work with the libraries on the system.

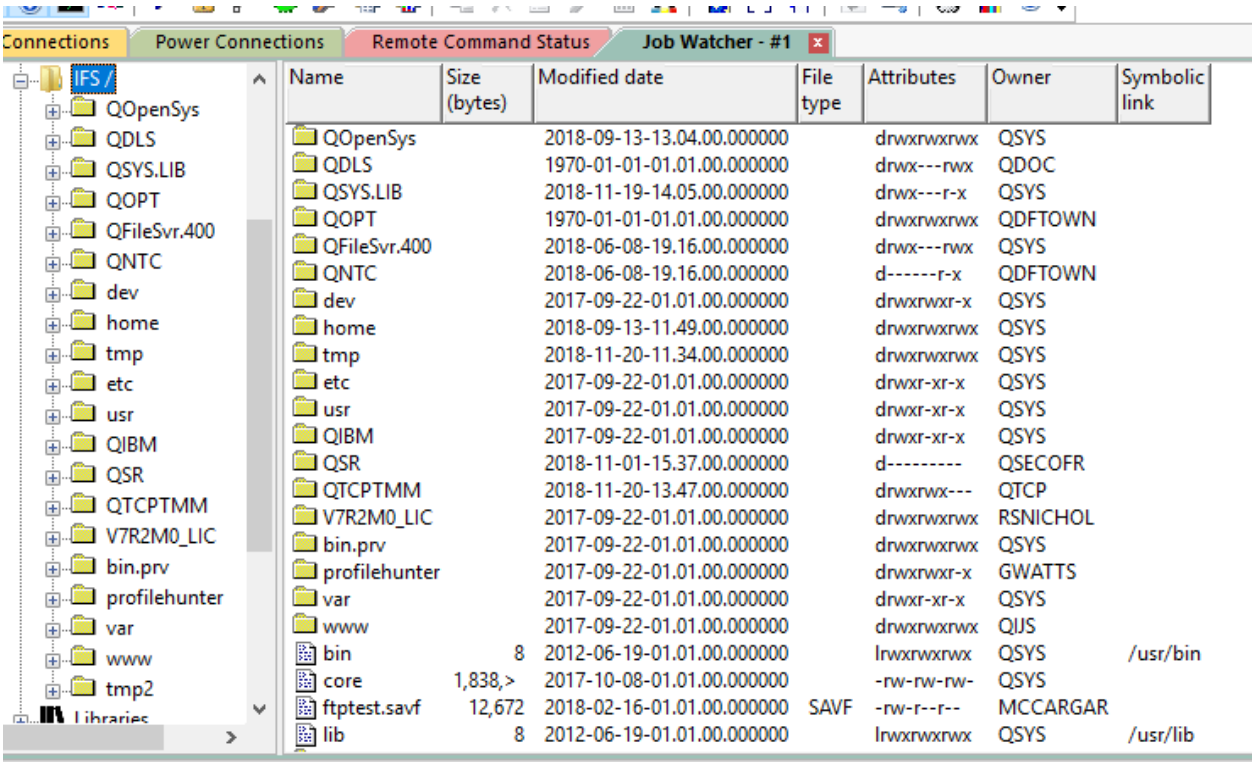


iDoctor FTP GUI folder on an IBM i system

Folder	Description
IFS /	This allow the user to browse the IFS starting at the directory listed. (Note: / is the root folder.) To change the starting point directory, right-click this folder and use the Set FTP Preferences menu option.
Libraries	This folder is used to browse the libraries on the system and work with the objects within them. **Tip:** If you have many libraries you may wish to right-click this folder and use the Filter libraries… menu option first!

5.1.1 IFS

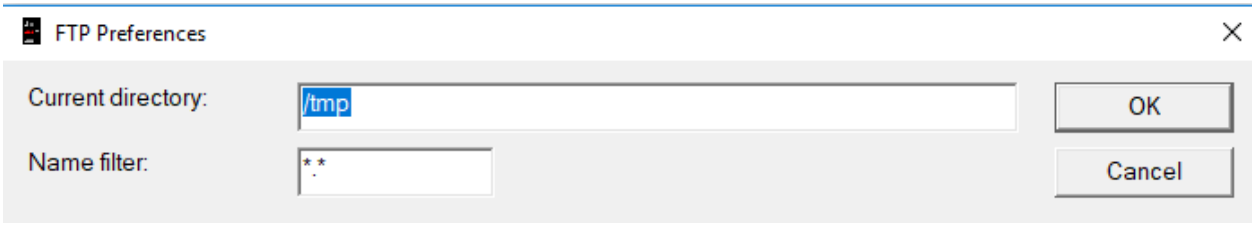
This folder lets users work with directories and files on the current system they are using in the Integrated File System. By default, the starting point is the root folder /.



IFS / folder's contents

5.1.1.1 Set FTP Preferences

The FTP Preferences window allows a user to modify the current (default) root directory shown in the iDoctor FTP GUI's IFS folder. There is also a name filter available will lets you filter the data returned in all directories shown using a wildcard file/folder name value.

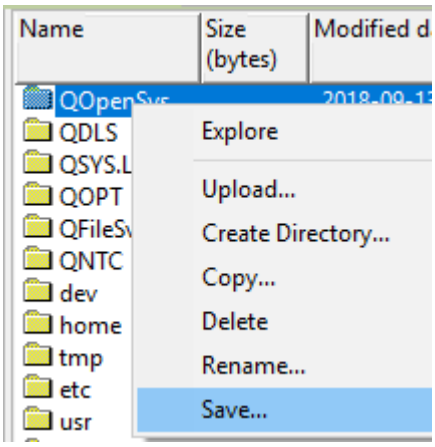


FTP Preferences Window

Option	Description
Current directory	Use this option to change the current IFS root directory (and the default for next time.) This allows you to specify a subdirectory further down than normal if that makes it faster/easier for you to access the needed files without needing to navigate the tree next time.
Name filter	The name filter is a wild card string to reduce the results shown in the directories displayed in the IFS folder (and subfolders.) By default the value is *.* (all files/folders with any extension). For example, use *.nmon to show only NMON files.

5.1.1.2 Directory menu options

The default right-click menu options available for any folder / subfolder in this view are:

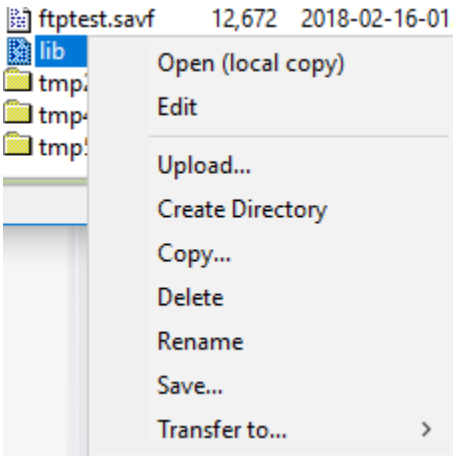


/QOpenSys directory menu options

Menu	Description
Explore	Expands the current folder and displays the results in the list.
Upload	This will present an interface to transfer one or more files from the PC to the remote system.
Create Directory	This option will create a new directory on the IFS in the desired location.
Copy…	This will present an interface that allows the user to copy the directory and all its contents to another location on the system.
Delete	This will delete the directory from the system.
Rename	This option will rename the directory.
Save	Use this option to save the directory and its contents to a save file.

5.1.1.3 File menu options

The default menu options for a file within the IFS are:



File menu options

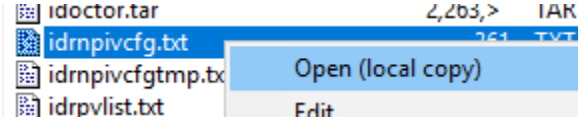
Tip: An additional menu option called Analyze Data will appear for.nmon, .npiv and .sea files for Analysis purposes.

Menu	Description
Open (local copy)	This option will first download the file to the PC and then attempt to open it using the default program for the current file type.
Edit	This option will download the file to the PC and then present an interface that lets you edit the results. When finished the file is sent back to the remote system. **Tip:** Only use this option for text files
Analyze Data (nmon, npiv, sea)	This option will display the Analyze Data window which allows .nmon, .npiv and/or .sea files to be analyzed and sent to DB files in the desired collection.
Upload	This will present an interface to transfer one or more files from the PC to the remote system.
Create Directory	This option allows the user to create a directory on the system.
Copy…	This will present an interface that allows the user to copy the selection to another location on the system.
Delete	This will delete the selection from the system.
Rename	This option will rename the currently selected file.
Save	Use this option to save the selection to a save file.
Transfer to	This option presents an interface that allows the user to transfer the current file(s) to another system, to IBM or to the PC.

5.1.1.4 Open (local copy)

This option is used to download and then execute the default Windows program for the file type being opened. For example, using this option on a .txt file would probably open the file in Notepad.

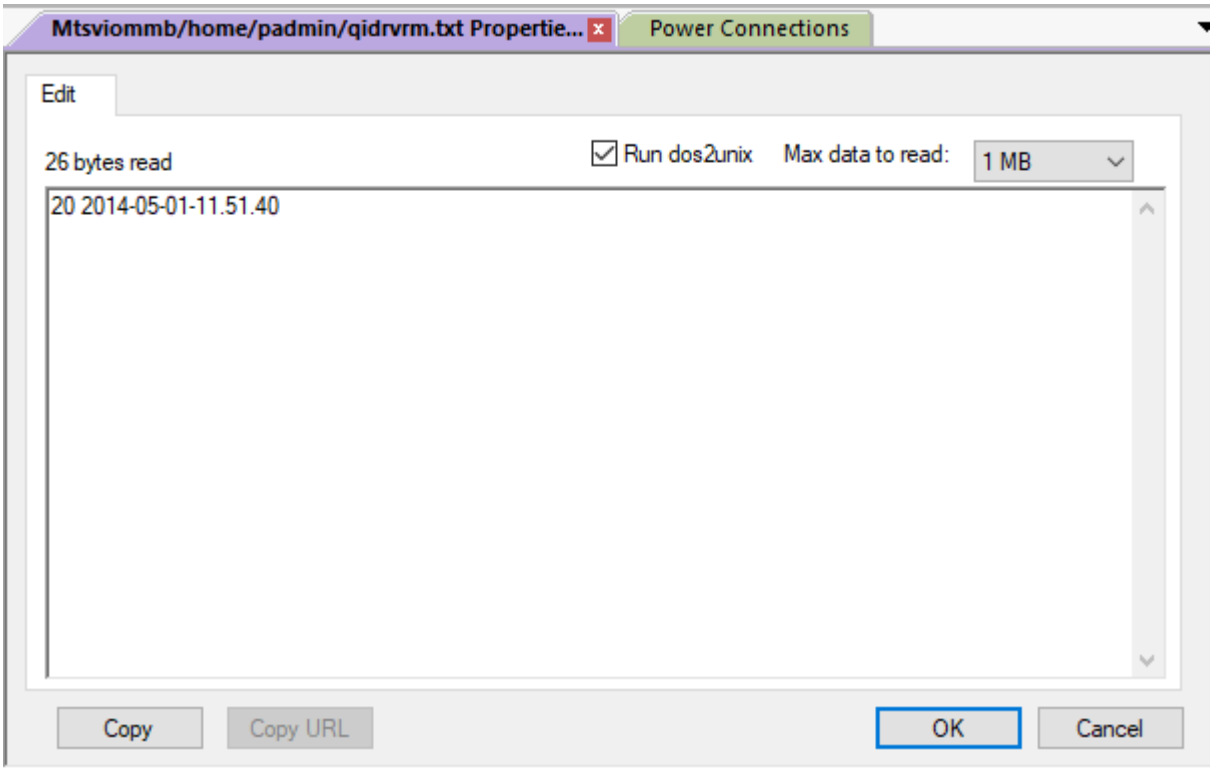
Tip: Files are downloaded by default into the iDoctor temp directory.



Open (local copy) menu

5.1.1.5 Edit

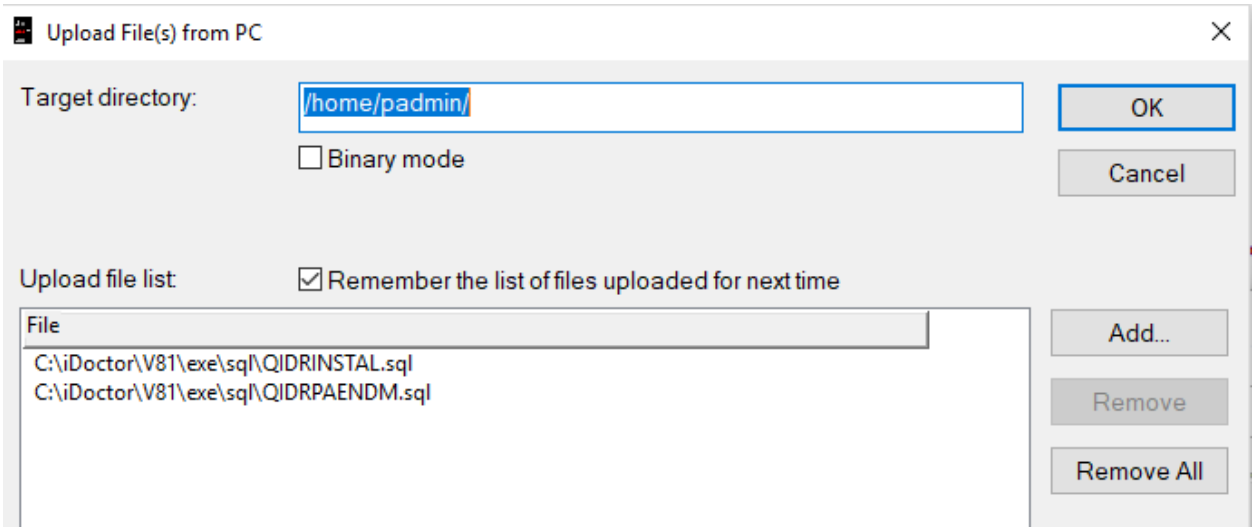
This option can be used to download the file to the PC and then open it within an editor inside of iDoctor.



Option	Description
Max data to read	This indicates how much data is downloaded to the PC from the file. The possible values are 1 MB, 10 MB and 100 MB. **Tip:** You should not edit anything other than text files with this interface and only if the entire data has been read.
OK	If changes have been made this will send the file back to the remote system and replace it.

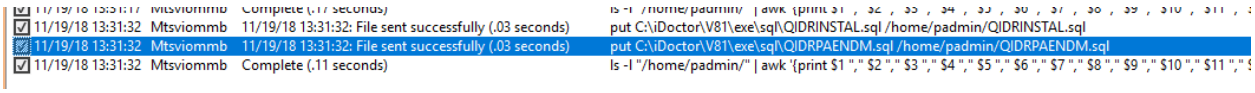
5.1.1.6 Upload File(s) from PC

The Upload file(s) window allows the user to transfer 1 or more files from the PC to the desired directory on the remote system. These transfers occur via the Remote command status view and the results of which can be viewed there.



Upload File(s) from PC Window

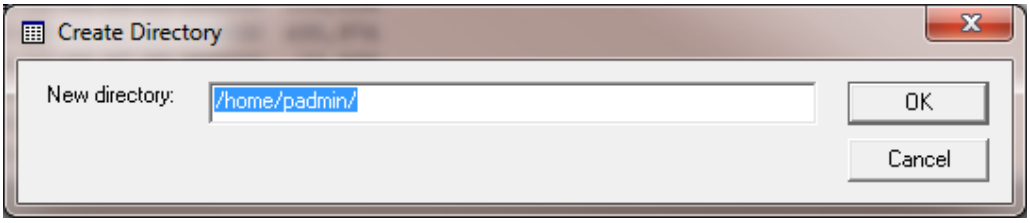
Option	Description
Target directory	This value contains the desired location to send the file(s) on the PC to.
Binary mode	Check this box if you wish to use binary mode to send the files.
Remember the list…	Use this option is you want the GUI to remember the list of files and repopulate the list again with the same set of files. This can be handy if sending the same files to multiple LPARs.
Upload file list	This is the list of files on the PC that will be transferred to the remote server.
Add button	Use this button to add files from the PC to the list.
Remove button	This button removes the selected files from the list.
Remove all button	Removes everything from the list.



Results in Remote Command Status View

5.1.1.7 Create Directory

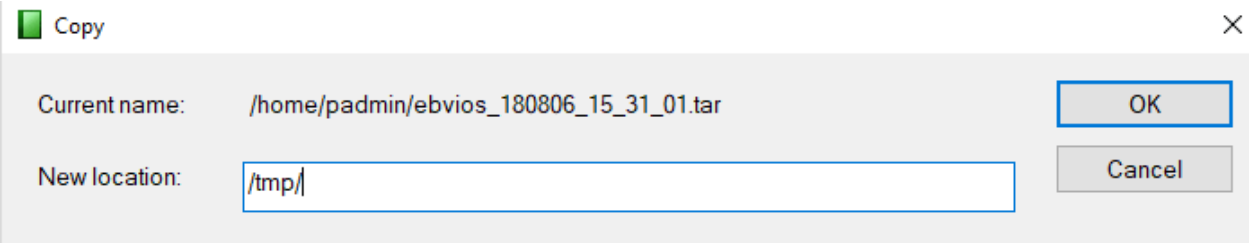
This option is used to create a directory on the remote server. By default, the window will show the path for the current directory and you will need to modify this path appropriately.



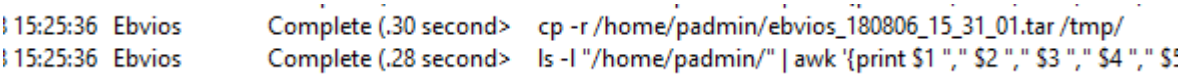
Create Directory Window

5.1.1.8 Copy

This option is to copy one or more files/directories to the desired new location. Only if specifying a single file may be the value entered a file name. Otherwise the value provided should be the directory to copy the selections into.



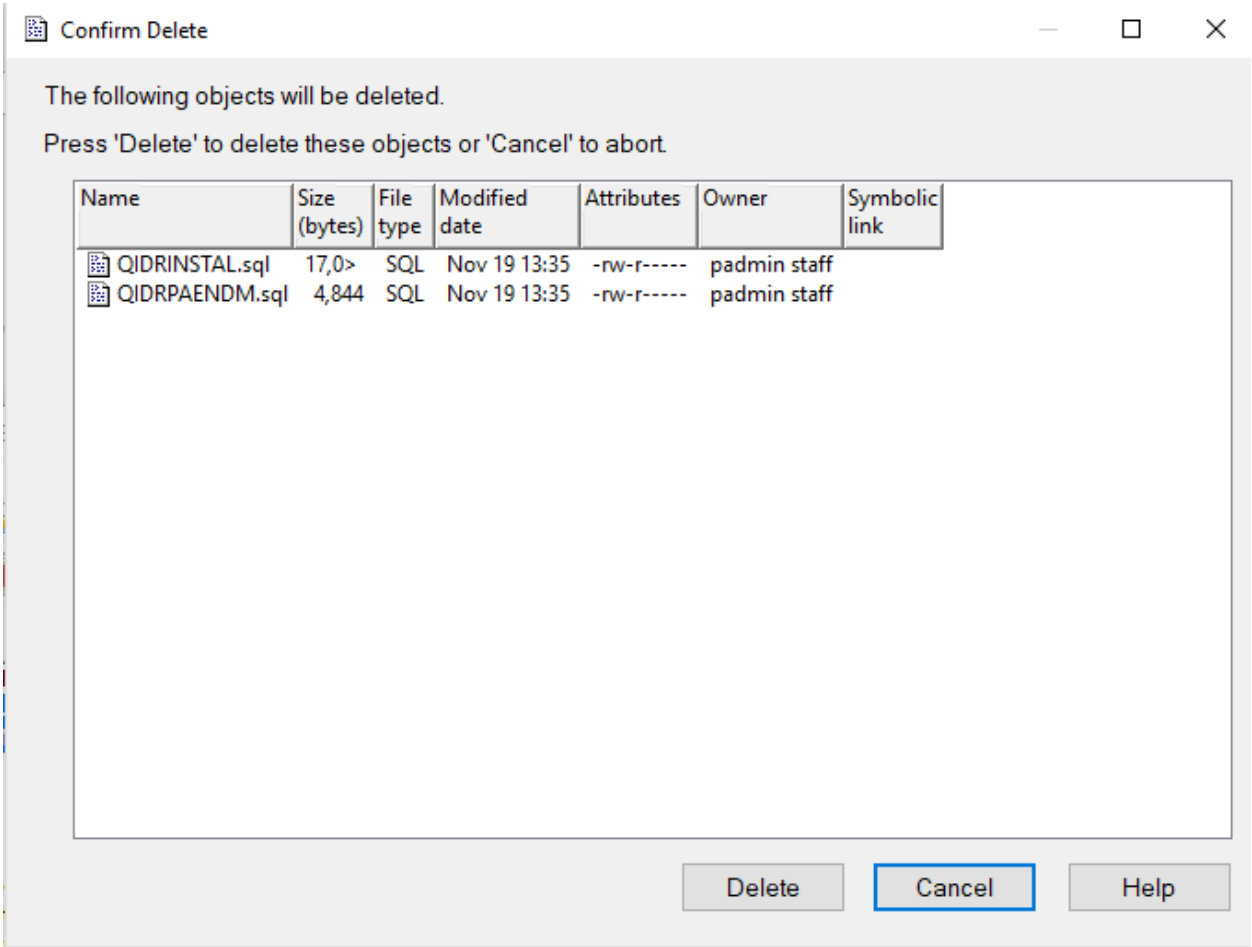
Copy Window



Remote Command Status View Copy

5.1.1.9 Delete

The Delete option will remove the file(s) and/or directories and their contents from the remote server.

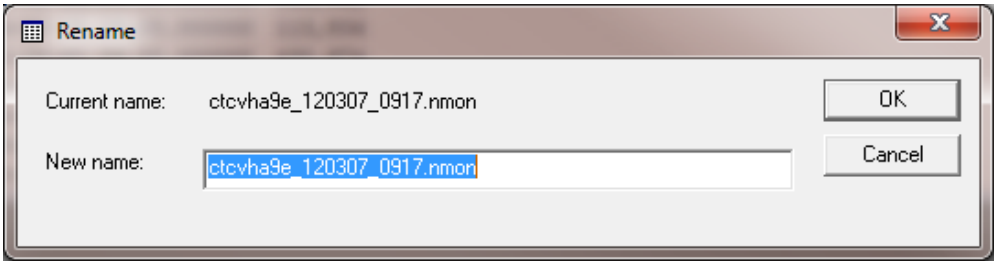


Confirm Delete Window

5.1.1.10 Rename

The Rename menu option allows you to change a specific file or directory and give it a new name.

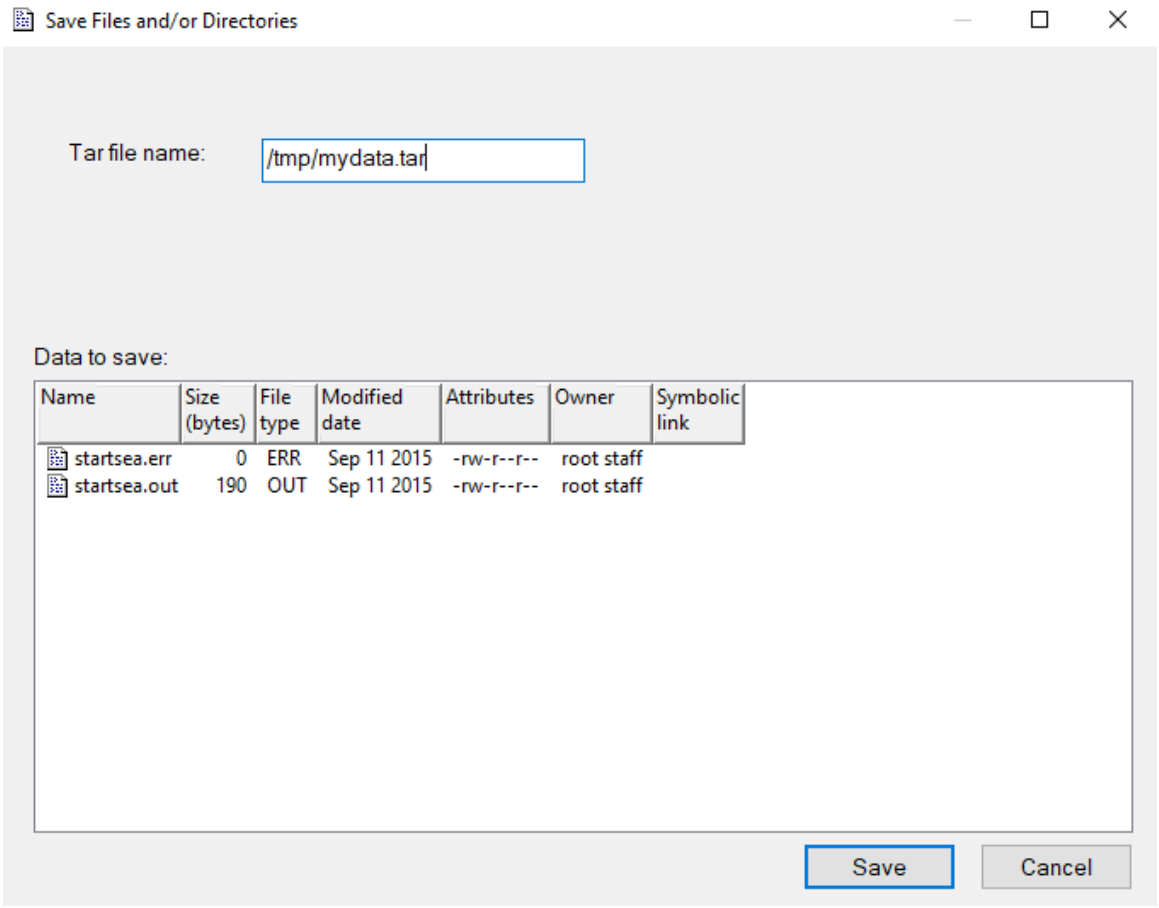
The interface looks like this:



Rename Window

5.1.1.11 Save

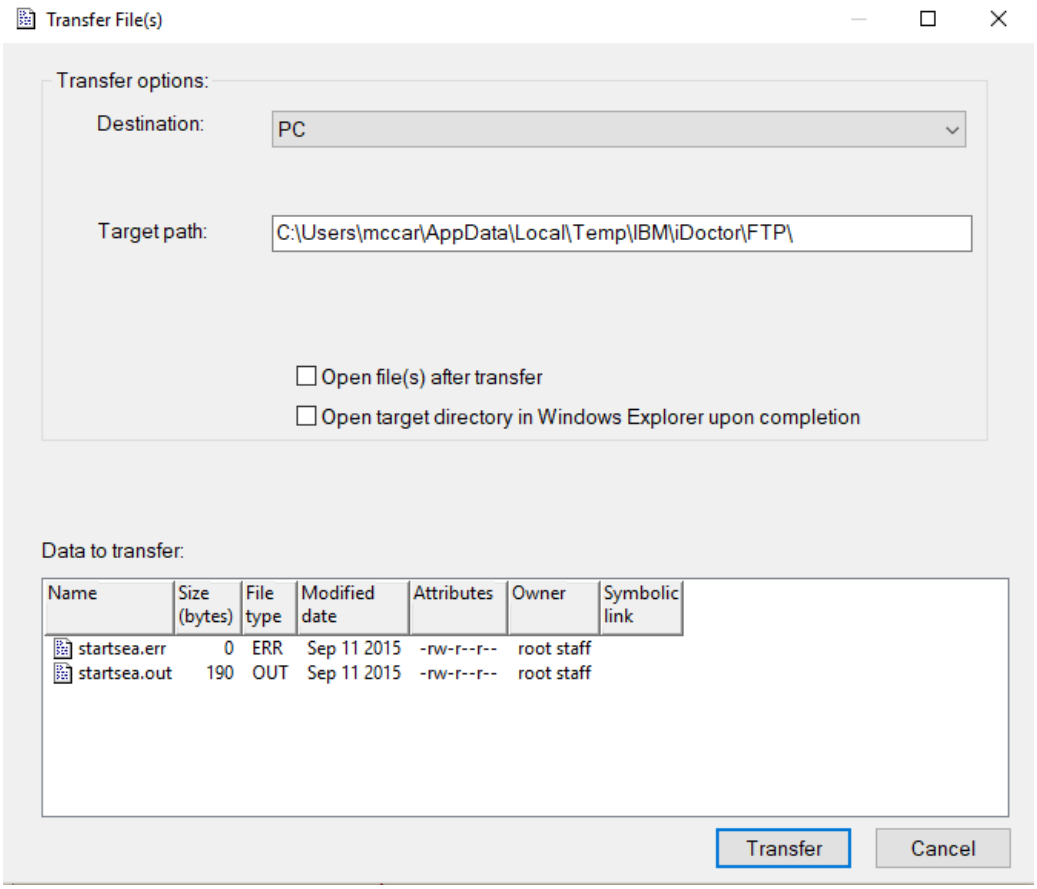
This option will save all selections to a tar file that you specify.



Save Files and/or Directories Window

5.1.1.12 Transfer to -> PC

This option can be used to download the selected file(s) to the PC. The files will be placed in the iDoctor temp directory's FTP subdirectory by default.

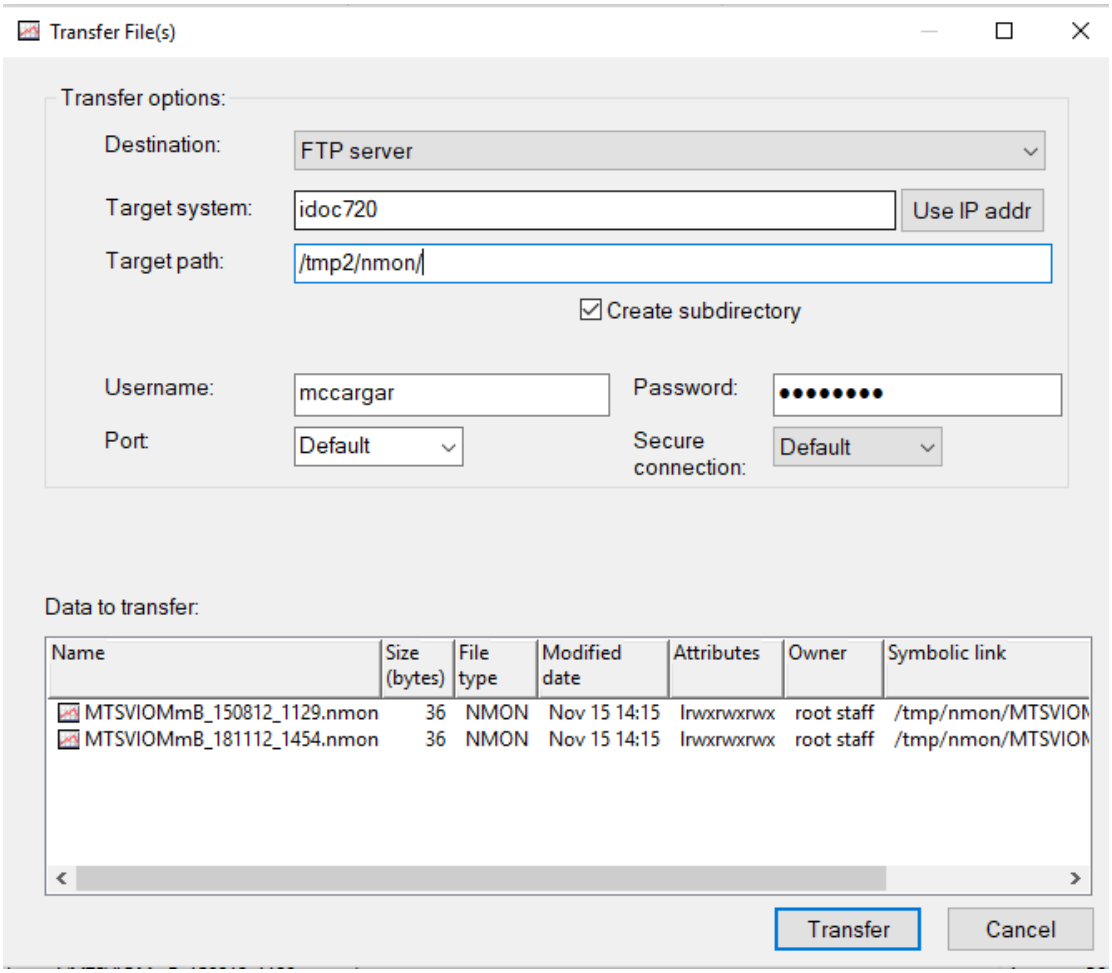


Transfer File(s) Window

Option	Description
Destination	This indicates where the files will be sent.
Target path	The directory on the PC to send the files to. Only if transferring a single file would this be a filename instead of a directory.
Open file(s) after transfer	This option will open each file (in Notepad) after they are transferred to the PC. **Tip:** Avoid using this option if you are transferring many files!
Open target directory in Windows Explorer	If checked, then after the download completes, Windows Explorer will be automatically opened to show the files downloaded on the PC.

5.1.1.13 Transfer to -> FTP Server

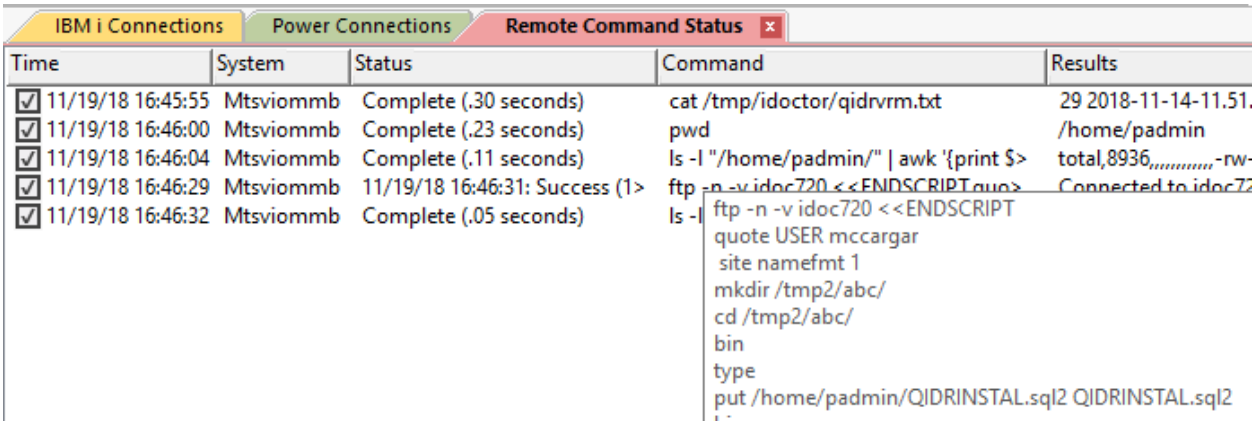
This option will send the selected files to the specified target system in the path indicated. This occurs by running the ftp command on the current remote server to the target system. The files will be sent directly from the system you are working with to the target system.



Transfer File(s) -> FTP server

Option	Description
Destination	This indicates where the files will be sent.
Target system	The system to send the selected files to.
Target path	The directory on the target system to send the selected file(s) to.
Create subdirectory	This will create the directory specified if it does not exist already. **Note:** This only works for a single directory, it will NOT create multiple directories in one shot.
Username	The user profile to use when making the connection to the remote server.
Password	The password to use when making the remote connection. Specify a password such as your email address if connecting with the “anonymous” user.
Port	This the port to use when making the FTP connection and should be either Default or Secure. If Secure is used, then the ftp -s parameter will be used to enable a secure TLS/SSL.
Secure connection	Select SSL for this value if you wish to make a secure connection (or use Secure for the Port parameter value.)
Data to transfer	This is the list of files to send to the target remote system.

After pressing the Transfer button, the required command will be sent to the Remote Command Status View and executed there. A scripted ftp command is used within the SSH connection that the Remote Command Status View provides.

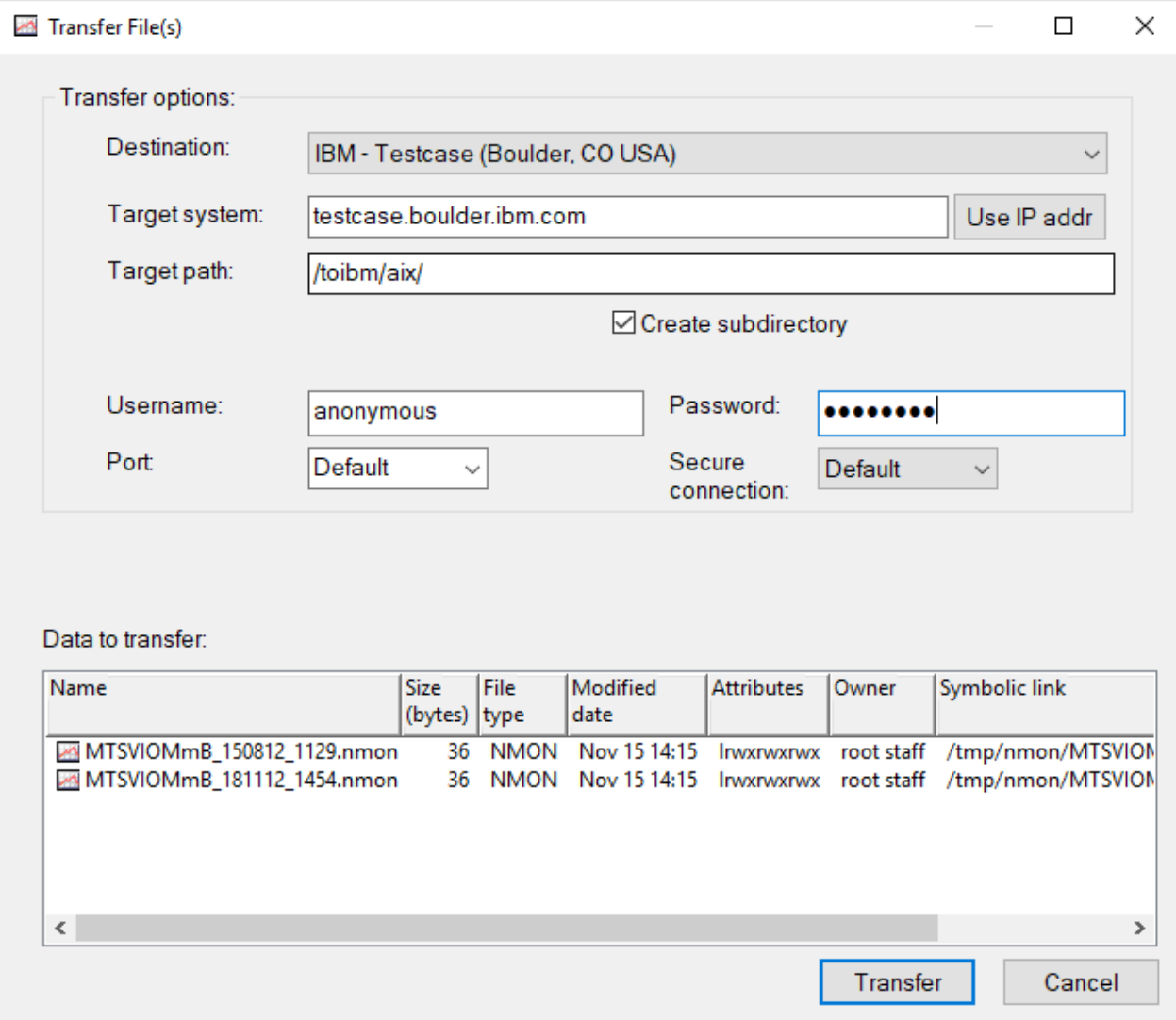


FTP script used to execute a file transfer from a VIOS to an IBM I within the Remote Command Status View

Tip: The results of running the script can be seen in the Results column.

5.1.1.14 Transfer to -> IBM - Testcase

This option will send the selected files to IBM's testcase system in Boulder, CO. This occurs by running the ftp command on the current remote server to the target system. The files will be sent directly from the system you are working with to the target system. Typically, you can use the username anonymous and password of your email address when making this connection.



Transfer File(s) -> IBM - Testcase

Option	**Description**
Destination	This indicates where the files will be sent.
Target system	The system to send the selected files to.
Target path	The directory on the target system to send the selected file(s) to.
Create subdirectory	This will create the directory specified if it does not exist already. **Note:** This only works for a single directory, it will NOT create multiple directories in one shot.
Username	The user profile to use when making the connection to the remote server.
Password	The password to use when making the remote connection. Specify a password such as your email address if connecting with the “anonymous” user.
Port	This the port to use when making the FTP connection and should be either Default or Secure. If Secure is used, then the ftp -s parameter will be used to enable a secure TLS/SSL.
Secure connection	Select SSL for this value if you wish to make a secure connection (or use Secure for the Port parameter value.)
Data to transfer	This is the list of files to send to the target remote system.

5.1.1.15 Transfer to -> IBM – ECUREP

This option will send the selected files to IBM's ECUREP system in Germany. See the previous section for more information on this interface.

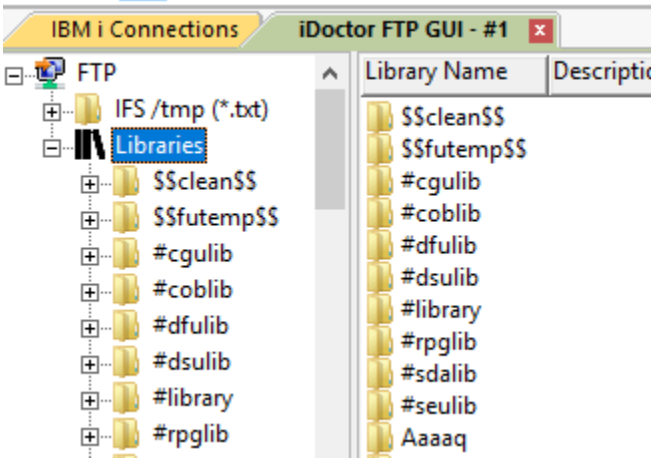
5.1.1.16 Transfer to -> IBM – Blue Diamond Lab

These options will send the selected files to IBM's Blue Diamond Lab. A secure FTP connection must be used when making the connection. See the previous section for more information on this interface.

5.1.2 Libraries

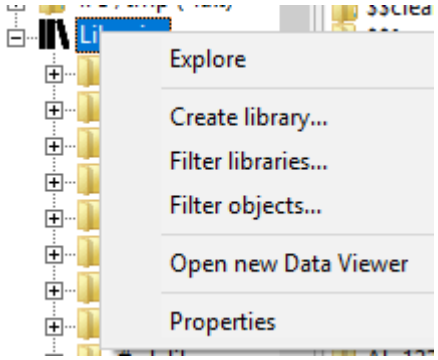
The iDoctor FTP GUI's libraries folder is an interface over the IBM i objects and libraries that exist on the system. The list of libraries and objects shown can be filtered down and reduced if necessary.

This interface can be used to work with files on the system or to work with various types of common IBM i objects such as save files and output queues.



iDoctor FTP GUI -> Libraries folder

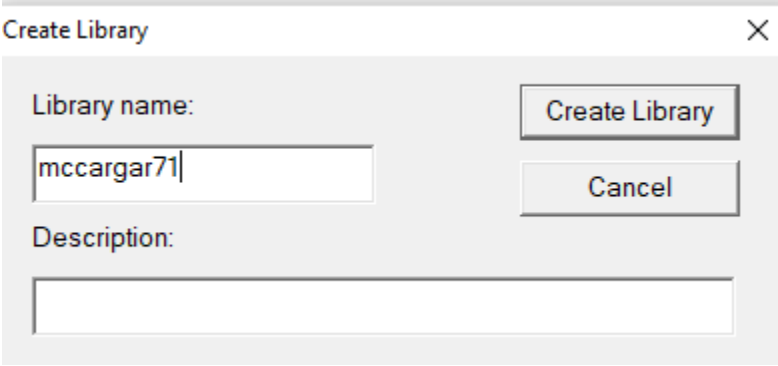
Right-click the Libraries folder to display the available options:



Option	Description
Explore	Displays the libraries within the Libraries folder (based on any filtering in effect.)
Create library	Displays an interface to create a new library on the IBM i.
Filter libraries…	This option displays the Filter libraries window which is used to reduce the libraries returned in the Libraries folder.
Filter objects…	This option displays the Filter objects window which provides several options to filter the list of objects returned within libraries on the system. The options used will affect the list of libraries returned as well to only return libraries that contain objects matching these filters.
Properties	This will display the iDoctor properties for the client and server.

5.1.2.1 Create Library

This window is used to create a new library on the IBM i.

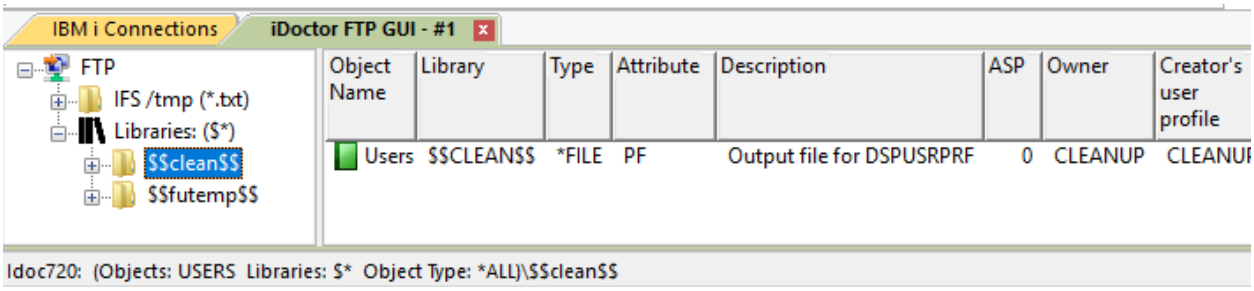


Create Library Window

5.1.2.2 Filter libraries

Use this option to filter the number of libraries returned in the Libraries folder. When a filter is applied the Libraries folder name will change to indicate this. This interface is covered in more detail in Chapter 4 Filter libraries

Tip: Your current library filters will be listed next to the Libraries folder and in the status bar at the bottom of the Main Window.

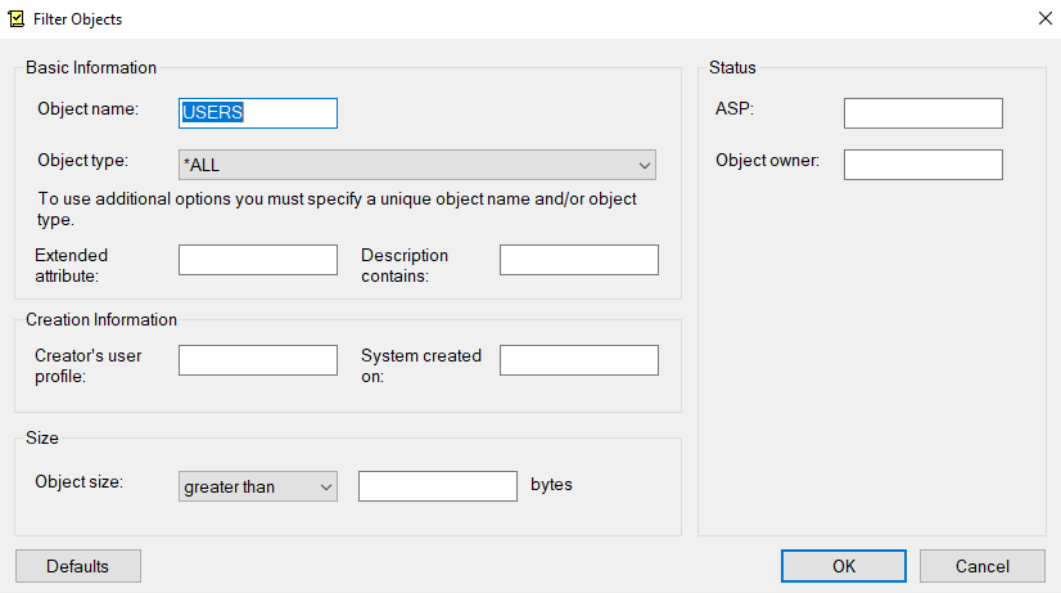


*iDoctor FTP GUI with object name filter of USERS and library name filter of $**

5.1.2.3 Filter objects

This interface is used to filter the objects returned in each library displayed within the Libraries folder in the iDoctor FTP GUI. These filters also apply to the initial list of libraries shown within the Libraries folder. i.e. Only libraries that contain objects matching these filters will be included.

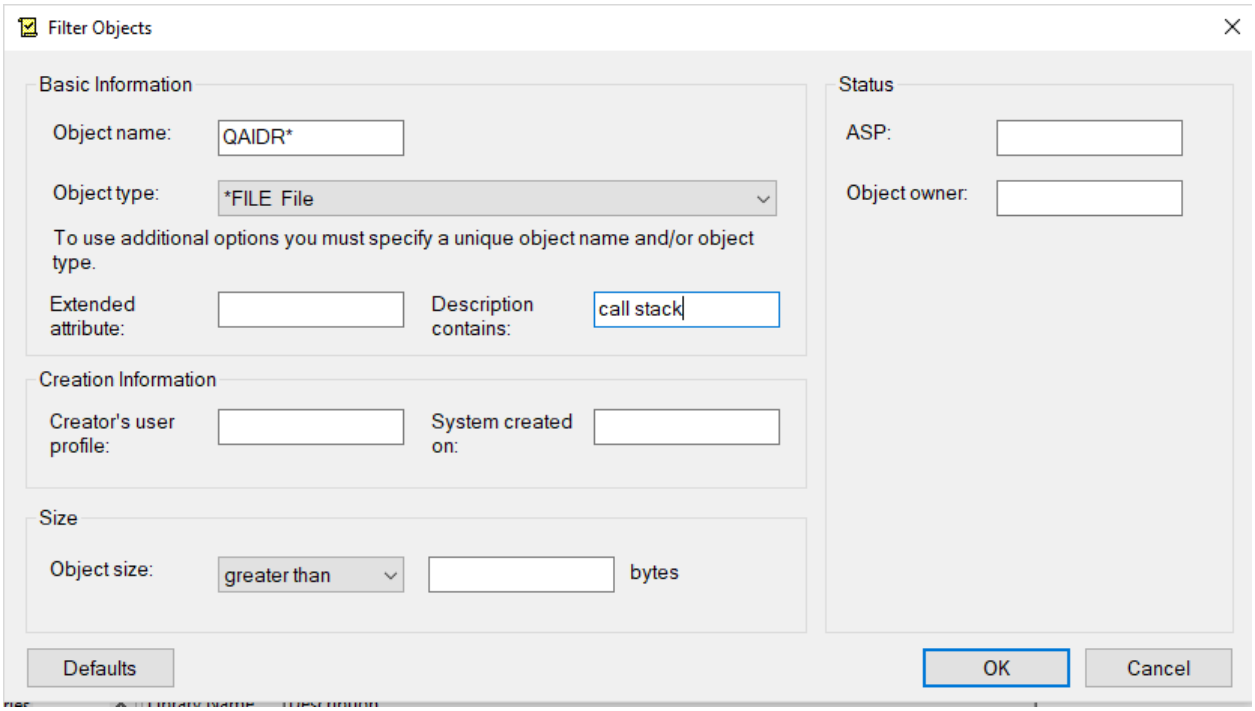
Note: On this interface you must specify a specific object name or object type in order to use all the filtering options on this screen after Object type. If you specify *ALL objects or objects QAIDR* with *ALL object type, then none of the additional filters will be available.



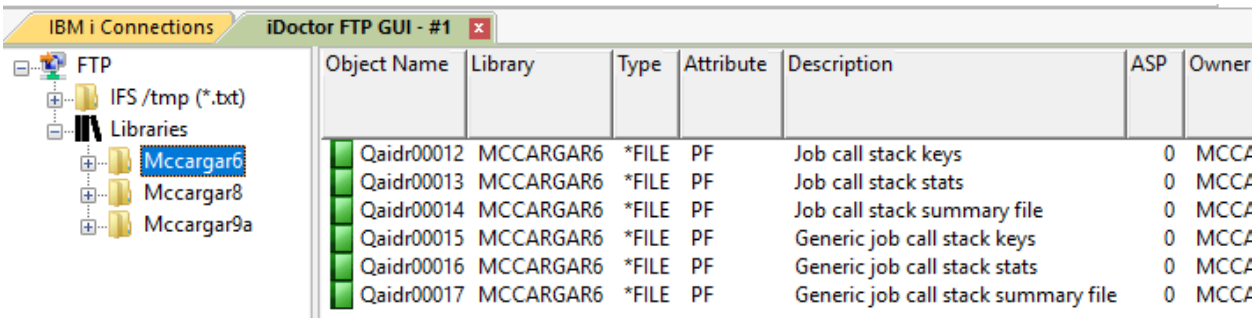
Filter Objects Window

Option	Description
Object name	This must be *ALL/blank to return all objects, or a single object name or generic object name such as QAIDR*. **Note:** On this interface you must specify a specific object name or object type in order to use all the filtering options on this screen after Object type.
Object type	This is a list of all IBM i object types as well as *ALL for all object types.
Extended attribute	You can use this filter to limit the objects by extended attribute such as PF, LF or SAVF.
Description contains	This option allows you to limit the objects returned to only those containing the indicated value in the object's text description.
Creator's user profile	This is the user profile that created the object initially.
System created on	This value is the IBM i system that the object was originally created on. **Note:** This filter is only functional when connected to IBM i 7.3 and higher systems.
Object size	This option allows you to filter the objects returned by size. The size value is given in bytes and can be indicated as a greater than or less than operation.
(Independent) ASP	This is the independent ASP number to filter the objects returned by. (i.e. 33)
Object owner	This value will filter the object list based on the user profile that owns the object.

An example of using this interface to find all *FILE objects named QAIDR* that contain 'call stack' in the object description is the following:



Filter Objects window looking for descriptions that contain 'call stack'

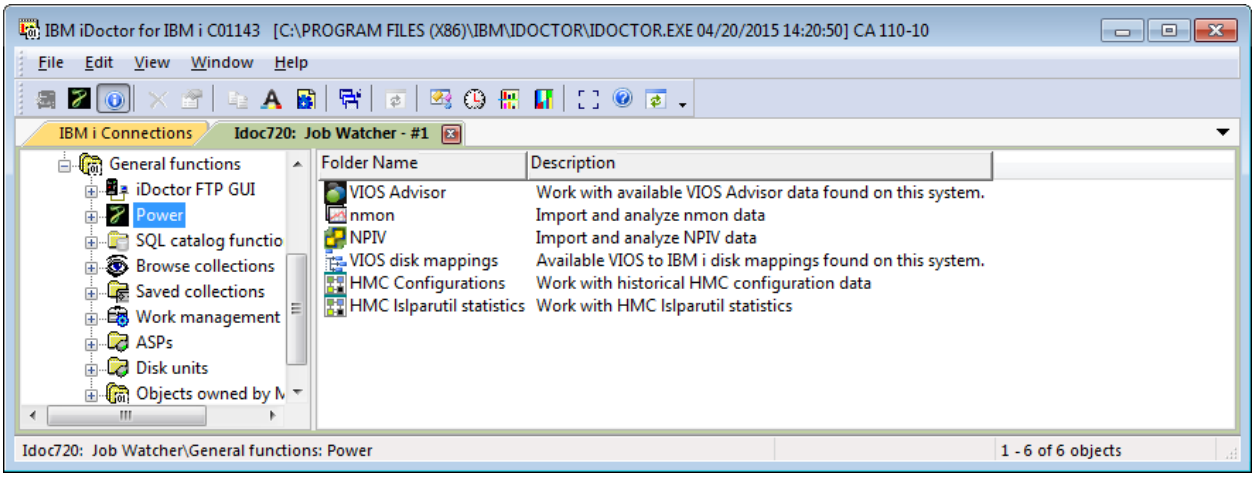


Libraries -> mccargar6 objects matching the filter

5.2 Power

The Power folder provides access to analysis functions for non-IBM i data such as nmon, npiv, VIOS advisor and more. It is provided here as a way for users to analyze already collected data when they may not have authority to connect to the VIOS, HMC, etc. directly.

A different subfolder is provided for each type of data supported.

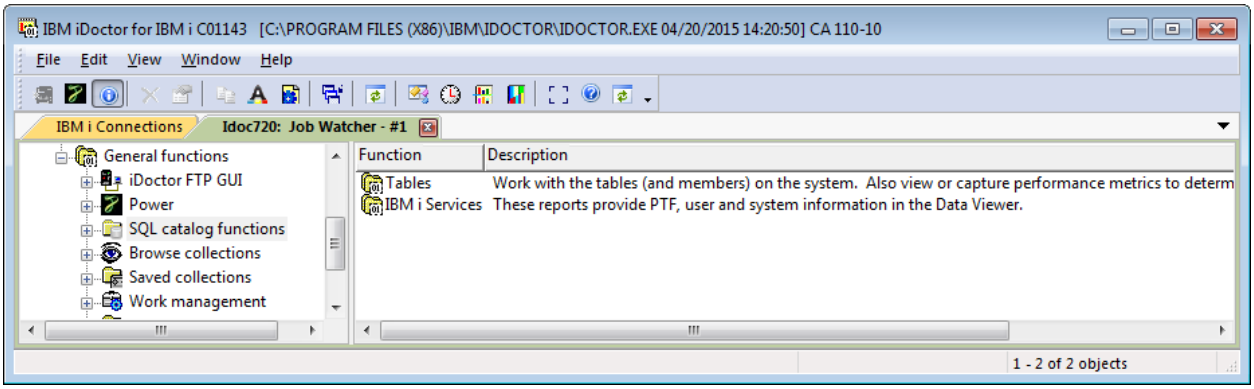


General functions -> Power folder in Job Watcher

For more information on the Power-based (non-IBM i) functions, please visit the chapter on Power.

5.3 SQL catalog functions

This folder provides functions related to the SQL catalog tables and SQL services found on DB2 for IBM i.



SQL catalog functions folder

At 7.1 and higher a collection utility is available that can help you determine which tables and/or members would be good candidates to move to SSDs. This utility can also be used to capture general performance statistics for tables and members over time.

A subfolder called IBM i Services is also available that provides reports made available with recent technology refreshes made to IBM i and DB2 for i in the SQL catalog tables. These reports are opened in the Data Viewer.

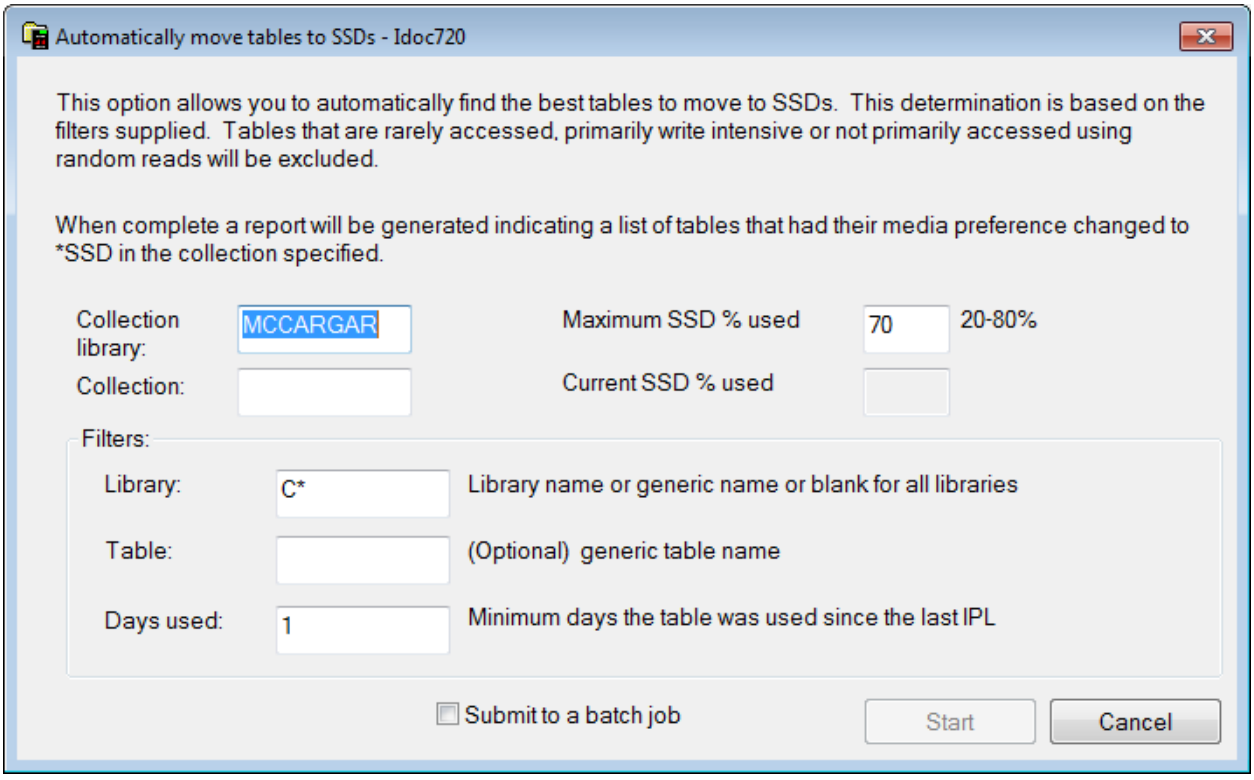
Right-clicking the SQL catalog functions folder provides the following options:

Menu	Description
Automatically fill SSDs	**Note:** This function is available at 7.1+ only. This provides the ability to capture performance metrics over time for the desired libraries and tables. This information is used to determine which tables/members would be the best candidates to move to SSDs. Once complete a report will be generated indicating which tables had their media preference changed to *SSD. **Note:** This function will NOT be available if no SSDs were found on the current system.
Start Collection…	**Note:** This function is available at 7.1+ only. This option is used to capture statistics about the tables or members on the system over time. **WARNING:** Capturing these statistics can be extremely time consuming and the appropriate filters, number of intervals and interval durations should be used based on the number of tables on your system that you are analyzing.
Filter Libraries…	This option can be used to filter the libraries shown under Tables -> Libraries subfolder.

5.3.1 Automatically move tables (or members) to SSDs (7.1+)

This window is used to kick off a process that will determine the best candidates to move to SSDs and automatically change the media preference to *SSD on the tables found based on the filters provided.

An example of this interface is:



Automatically move tables to SSDs

The options available on this screen are:

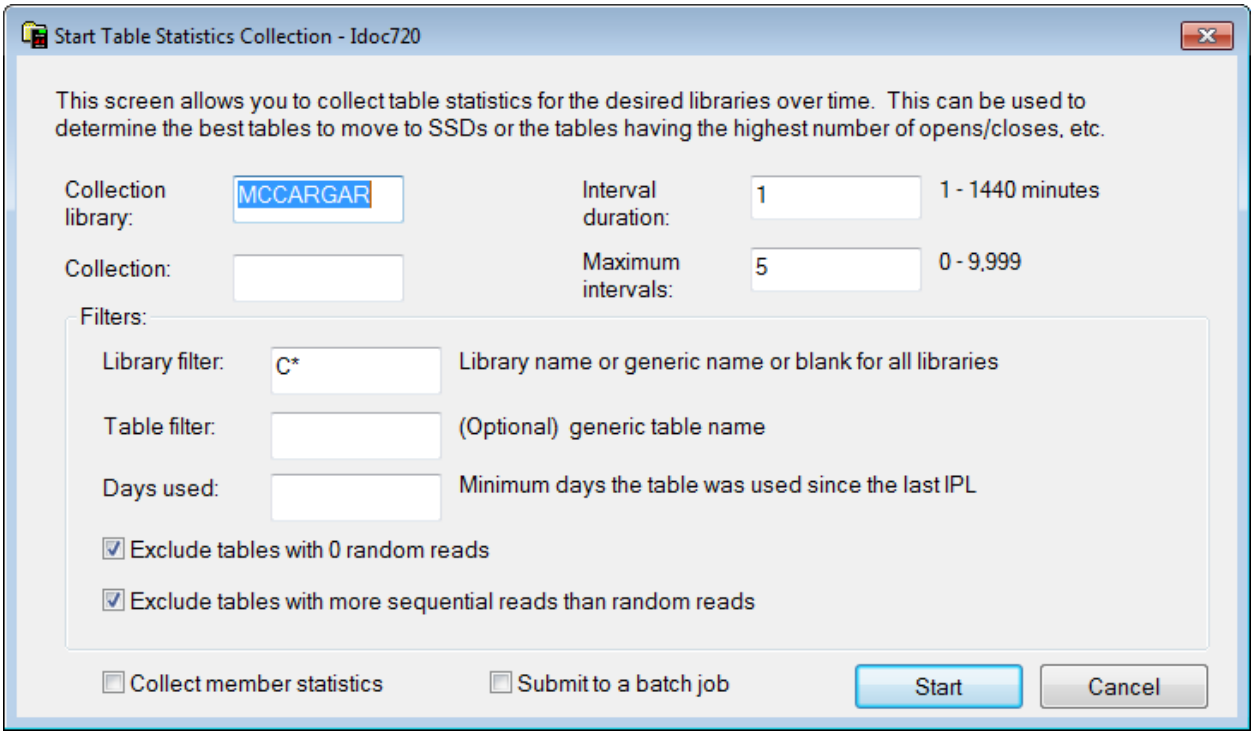
Option	Description
Collection library	The library where the report will be generated.
Collection	The collection or name of the report to generate.
Maximum SSD % used	This setting is used to avoid overfilling the SSDs by keeping the percentage of SSD % space used less than the value provided. If the SSD % used exceeds this value then the media preference setting will not be changed by this process.
Current SSD % used	This will list the current SSD % of disk space used for this system.
Submit to a batch job	This option will submit this process to a batch job rather than running it interactively in the Remote SQL Statement Status View within the iDoctor GUI.

Filters	Description
Library filter	This can be either the library name or generic library name of the tables to search or blank to look at tables in all libraries. **Note:** If you have a system with thousands/millions of tables, then using appropriate filters is highly recommended or this process could take hours or even days.
Table filter	This is an optional generic table name filter.
Days used	This filter allows you to exclude tables that have not been used at least N days since the last IPL.

5.3.2 Start Table (or member) Statistics Collection (7.1+)

Note: This function is available at 7.1+ only. This option is used to capture statistics about the tables or members on the system over time.

WARNING: Capturing these statistics can be extremely time consuming and the appropriate filters, number of intervals and interval durations must be used based on the number of tables on your system that you are analyzing.



Start Table Statistics Collection

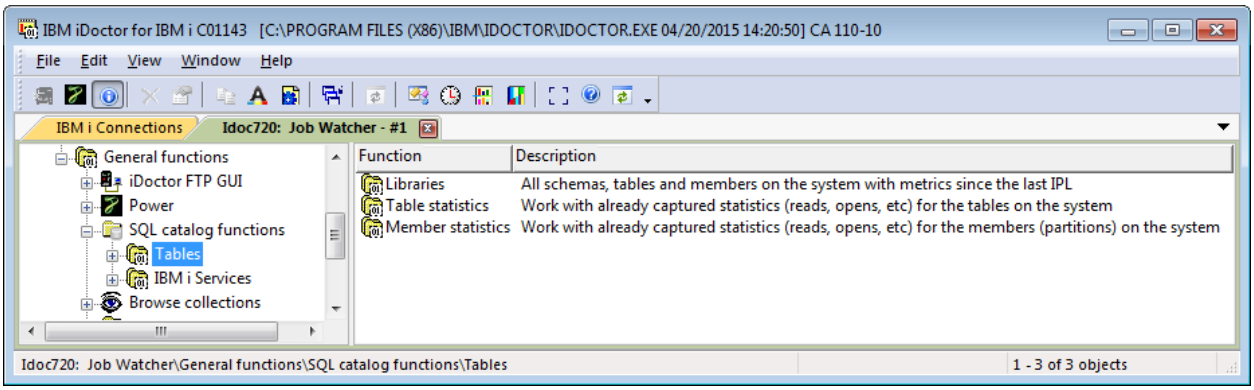
The options available on this screen are:

Option	Description
Collection library	The library where the collection will be generated.
Collection	The collection name to generate.
Interval duration	This value indicates how often a fresh snapshot of the statistics will be captured. Keep in mind that if you are analyzing several thousands of tables, the time it takes to capture this data could easily exceed the interval duration. It is typically best to keep this value fairly large such as 60 minutes or greater.
Maximum intervals	This value is the number of intervals to collect.
Collect member (or table) statistics	Check this box if you also want to collect statistics at the member (or table) level depending on if you selected the Start Table Statistics Collection or the Start Member Statistics Collection option.
Submit to a batch job	This option will submit this process to a batch job rather than running it interactively in the Remote SQL Statement Status View within the iDoctor GUI.

Filters	Description
Library filter	This can be either the library name or generic library name of the tables to search or blank to look at tables in all libraries. **Note:** If you have a system with thousands/millions of tables, then using appropriate filters is highly recommended or this process could take hours or even days.
Table filter	This is an optional generic table name filter.
Days used	This filter allows you to exclude tables that have not been used at least N days since the last IPL.
Exclude tables with 0 random reads	When checked, statistics for tables that had 0 random reads are not captured.
Exclude tables with more sequential reads than random reads	When checked, statistics for tables that had more sequential reads than random reads are excluded.

5.3.3 Tables (7.1+)

The SQL catalog functions -> Tables subfolder provides users the ability to view statistics for all tables on the system in real time or view collected performance metrics on the tables and members on the system.



General functions -> SQL catalog functions -> Tables folder

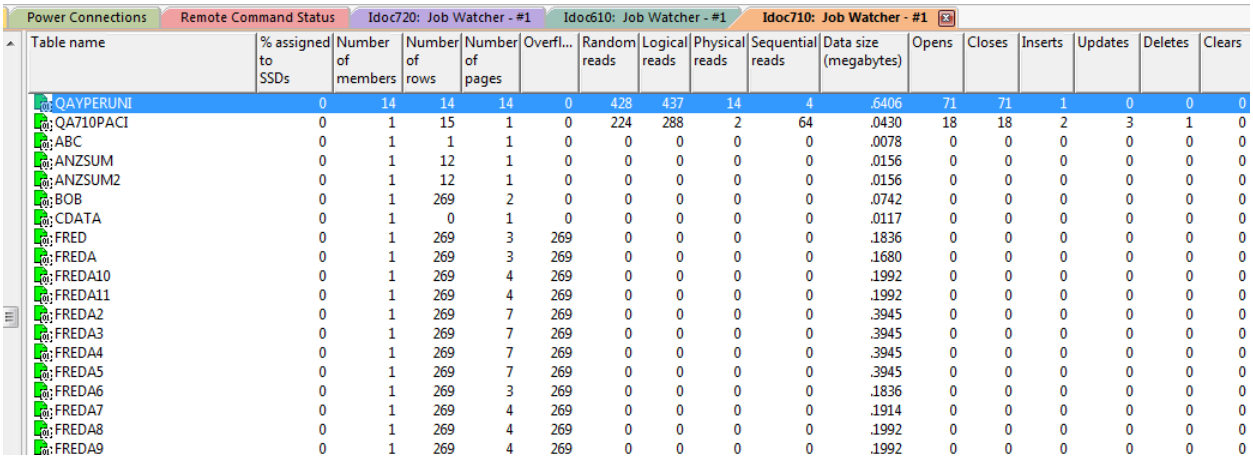
5.3.3.1 Libraries (metrics since the last IPL)

This folder contains each library on the system or a filtered list of libraries if the Filter Libraries menu option has been used. Within each of these libraries you will the tables they contain and the performance metrics available since the last IPL.

5.3.3.1.1 Tables

By selecting a library you will be presented with a list of all tables found in the library and their statistics. The data is sorted by random reads in descending sequence by default.

If desired you may expand a table to view the members (partitions) within each.



Power Connections | Remote Command Status | Idoc720: Job Watcher - #1 | Idoc610: Job Watcher - #1 | **Idoc710: Job Watcher - #1**

Table name	% assigned to SSDs	Number of members	Number of rows	Number of pages	Overfl...	Random reads	Logical reads	Physical reads	Sequential reads	Data size (megabytes)	Opens	Closes	Inserts	Updates	Deletes	Clears
QAYPERUNI	0	14	14	14	0	428	437	14	4	.6406	71	71	1	0	0	0
QA710PACI	0	1	15	1	0	224	288	2	64	.0430	18	18	2	3	1	0
ABC	0	1	1	1	0	0	0	0	0	.0078	0	0	0	0	0	0
ANZSUM	0	1	12	1	0	0	0	0	0	.0156	0	0	0	0	0	0
ANZSUM2	0	1	12	1	0	0	0	0	0	.0156	0	0	0	0	0	0
BOB	0	1	269	2	0	0	0	0	0	.0742	0	0	0	0	0	0
CDATA	0	1	0	1	0	0	0	0	0	.0117	0	0	0	0	0	0
FRED	0	1	269	3	269	0	0	0	0	.1836	0	0	0	0	0	0
FREDA	0	1	269	3	269	0	0	0	0	.1680	0	0	0	0	0	0
FREDA10	0	1	269	4	269	0	0	0	0	.1992	0	0	0	0	0	0
FREDA11	0	1	269	4	269	0	0	0	0	.1992	0	0	0	0	0	0
FREDA2	0	1	269	7	269	0	0	0	0	.3945	0	0	0	0	0	0
FREDA3	0	1	269	7	269	0	0	0	0	.3945	0	0	0	0	0	0
FREDA4	0	1	269	7	269	0	0	0	0	.3945	0	0	0	0	0	0
FREDA5	0	1	269	7	269	0	0	0	0	.3945	0	0	0	0	0	0
FREDA6	0	1	269	3	269	0	0	0	0	.1836	0	0	0	0	0	0
FREDA7	0	1	269	4	269	0	0	0	0	.1914	0	0	0	0	0	0
FREDA8	0	1	269	4	269	0	0	0	0	.1992	0	0	0	0	0	0
FREDA9	0	1	269	4	269	0	0	0	0	.1992	0	0	0	0	0	0

Real-time table metrics for the tables within library mccargar

The data provided in this window includes the following:

Column	Description
% assigned to SSDs	This value represents the current data allocation percentage to SSDs for the table.
Number of members	Number of partitions or members of the table.
Number of rows	Number of valid rows in all partitions or members of the table.
Number of pages	Number of 64K pages in all partitions or members of the table.
Overflow	The estimated number of rows that have overflowed to variable length segments. If the table does not contain variable length or LOB columns, contains 0.
Random reads	Number of random read operations of all partitions or members of the table since the last IPL.
Logical reads	Number of logical read operations of all partitions or members of the table since the last IPL.
Physical reads	Number of physical read operations of all partitions or members of the table since the last IPL.
Sequential reads	Number of sequential read operations of all partitions or members of the table since the last IPL.
Data size (megabytes)	Total size of the data spaces in all partitions or members of the table.
Opens	Number of full opens of all partitions or members of the table since the last IPL.
Closes	Number of full closes of all partitions or members of the table since the last IPL.
Inserts	Number of insert operations of all partitions or members of the table since the last IPL.
Updates	Number of update operations of all partitions or members of the table since the last IPL.
Deletes	Number of delete operations of all partitions or members of the table since the last IPL.
Clears	Number of clear operations (CLRPFM operations) of all partitions or members of the table since the last IPL.
Copies	Number of data space copy operations (certain CPYxxx operations) of all partitions or members of the table since the last IPL.
Reorganize	Number of data space reorganize operations (non-interruptible RGZPFM operations) of all partitions or members of the table since the last IPL.
Index build	Number of creates or rebuilds of indexes that reference any partition or member of the table since the last IPL. This does not include maintained temporary indexes.
Change time	Maximum timestamp of the last change that occurred to any partition or member of the table.
Last used time	Maximum timestamp of the last time any partition or member was used directly by an application for native record I/O or SQL operations. If no partition or member has ever been used, contains null.
Non-SSD space used (bytes)	Space used by all partitions or members for this file allocated to disks other than SSDs (in bytes.)
SSD space used (bytes)	Space used by all partitions or members for this file allocated to SSDs (in bytes.)

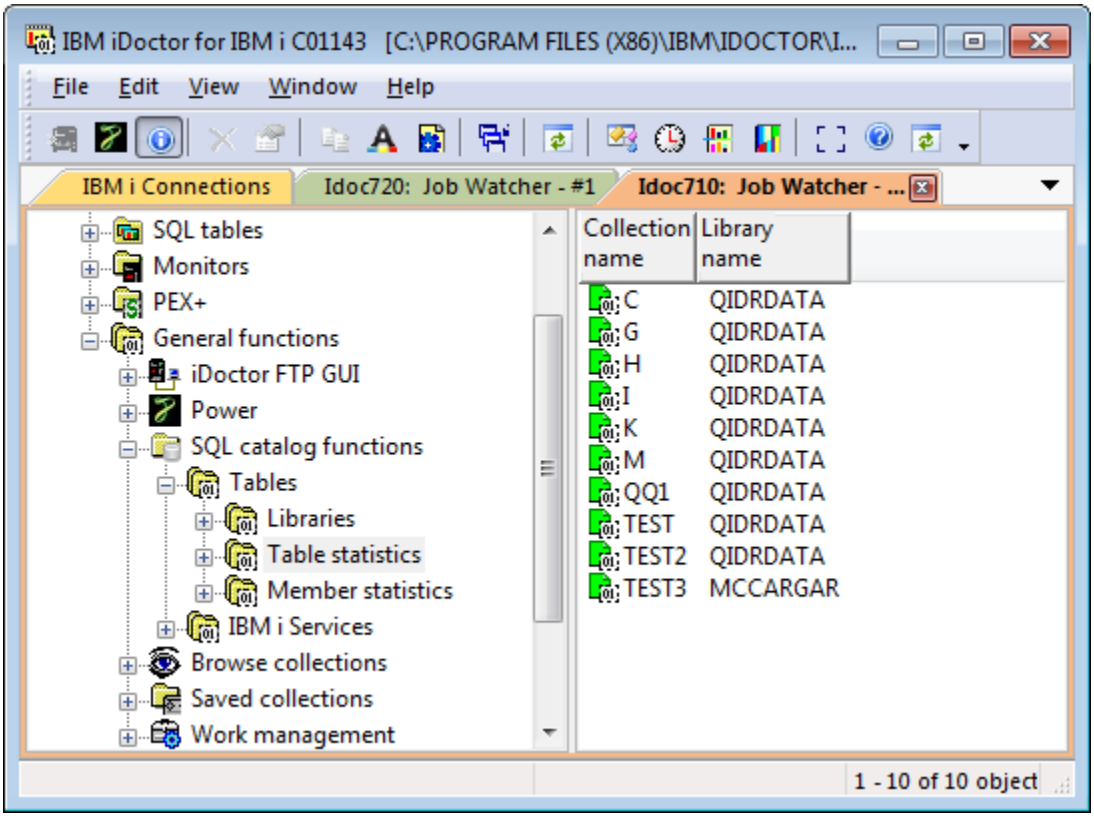
5.3.3.1.2 Members

Expanding a table in the tree/list interface will display the list of members and the real-time metrics for each member since the last IPL. The statistics shown are the same as covered for tables in the previous section.

You can also double-click a table to open it in the Data Viewer.

5.3.3.2 Table Statistics

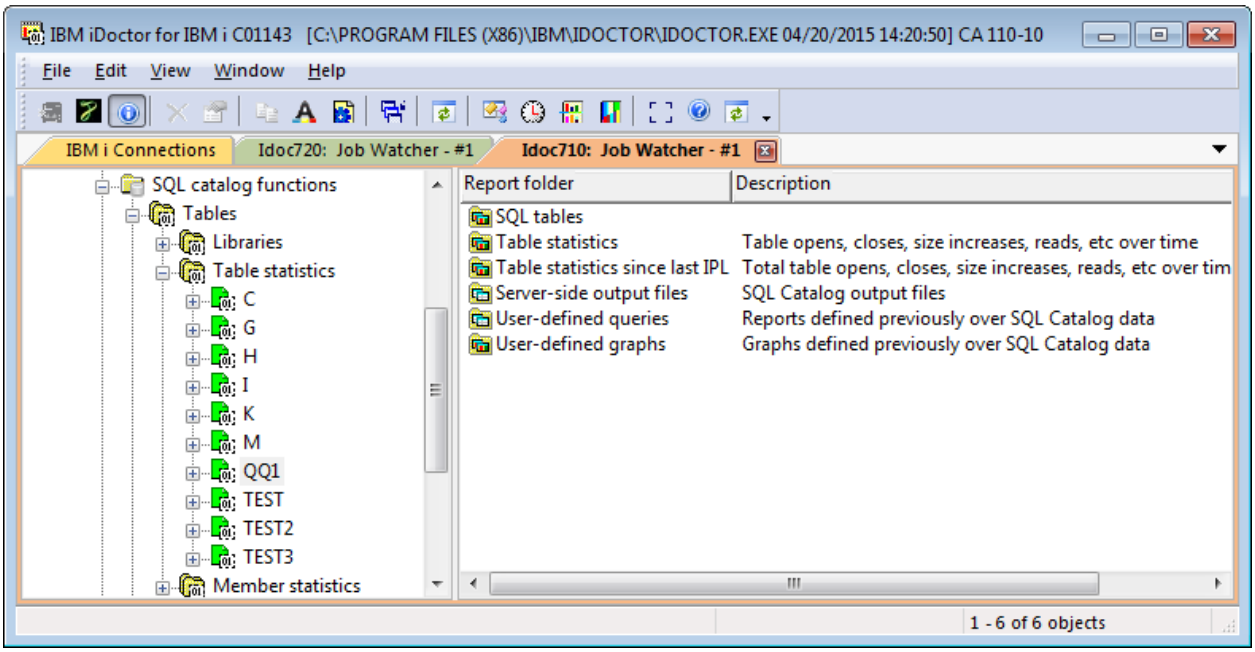
This folder provides access to graphs and reports over already captured table statistics data produced by iDoctor. All of the collections found on the system of this type will be displayed.



List of table statistics collections found on a system within Job Watcher

5.3.3.2.1 Collections

Expanding a Table Statistics collection provides the following options:



Contents of a Table Statistics collection

Graphs are available under the table statistics and table statistics since last IPL folder.

Note: The table statistics folder will only be shown if more than 1 snapshot has been collected.

5.3.3.2.2 Table Statistics Reports

The graphs available under the Table statistics folder are:

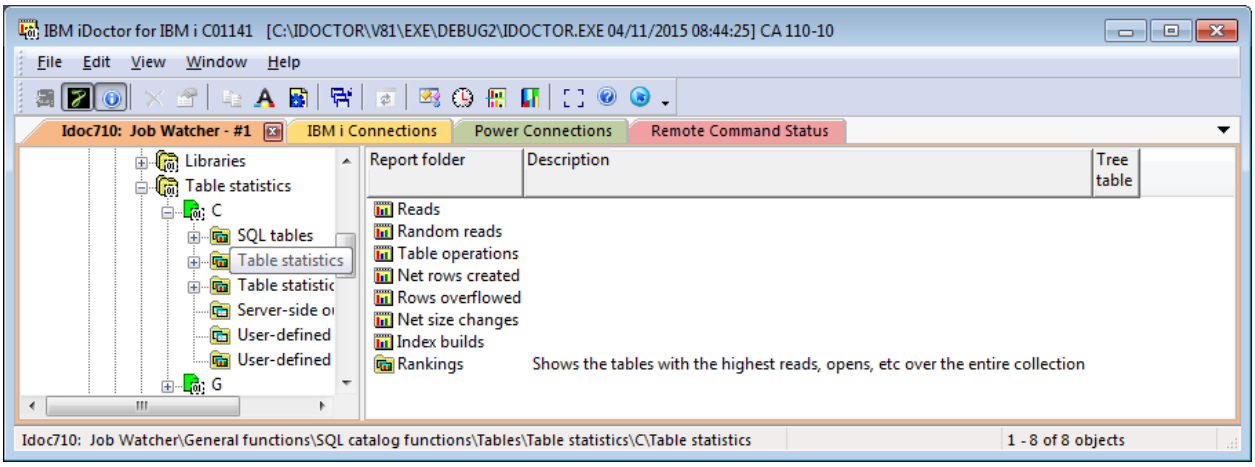


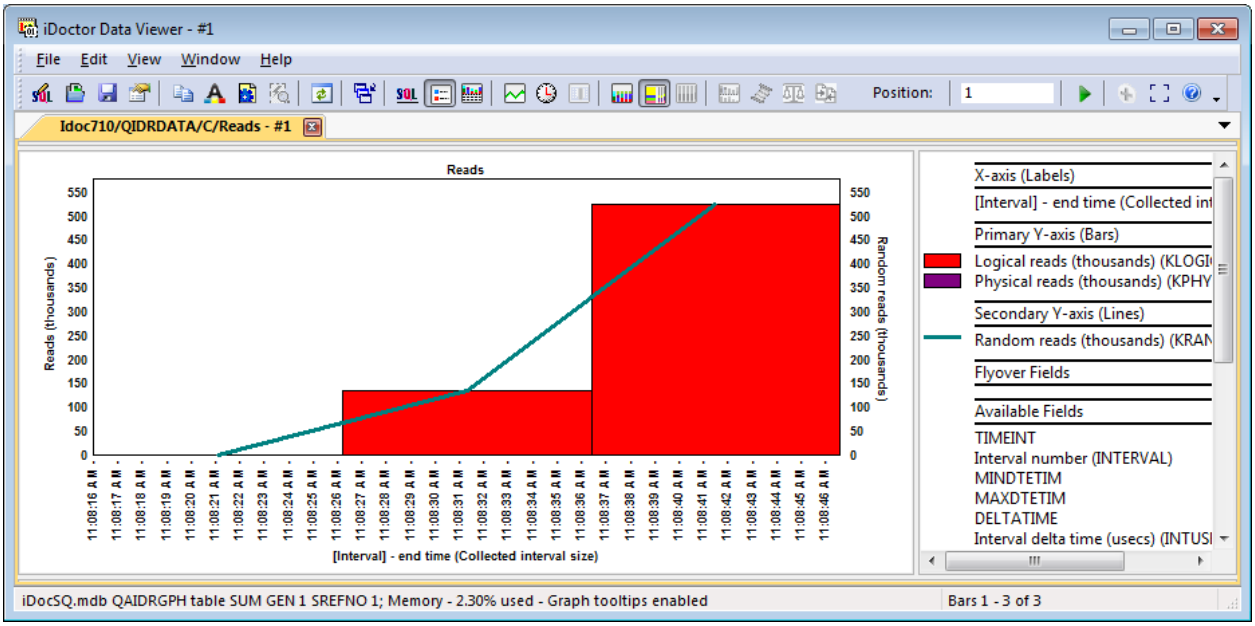
Table statistics folder graphs

These graphs are available for either all tables collected over time, or ranked by the desired metric(s). You may drill down from the over time graphs to the rankings graphs for the selected time period.

Note: All graphs are based on the tables specified to be included in the data collection over the number of intervals and interval durations provided.

5.3.3.2.2.1 Reads

This graph shows the logical and physical reads over time with random reads on the second Y-axis.



Reads

5.3.3.2.2.2 Random reads

This graph displays the random reads by itself over time.

5.3.3.2.2.3 Table operations

This graph displays several types of miscellaneous table functions: Opens, closes, inserts, updates, deletes, clear, copy and reorganize.

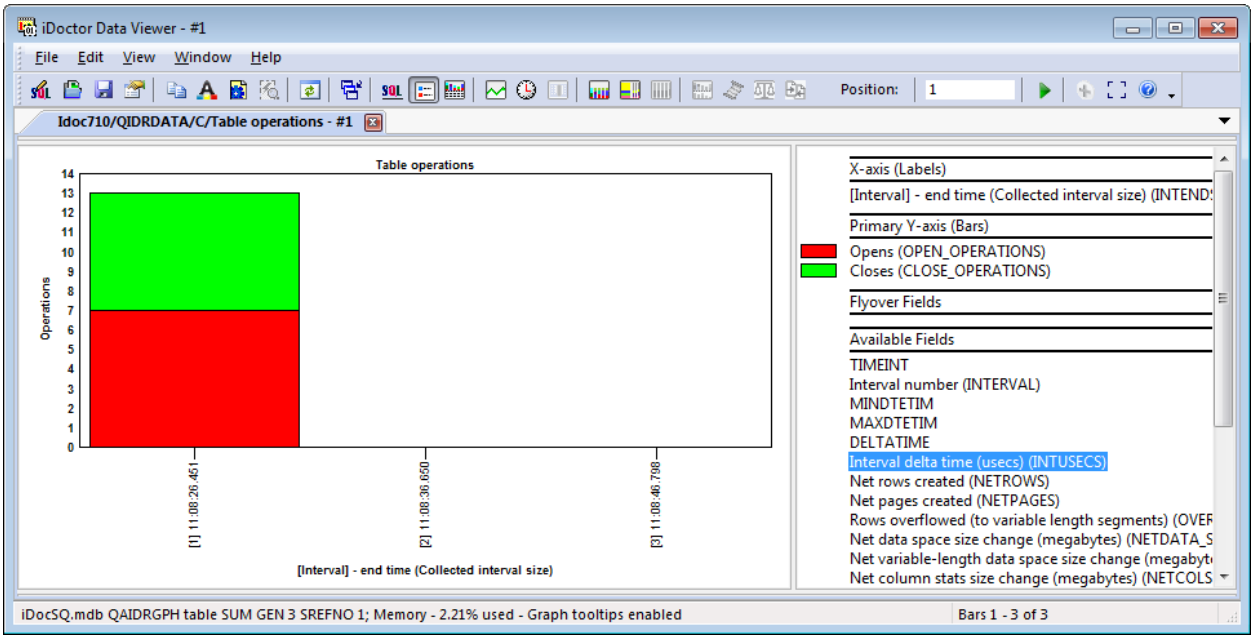


Table operations

5.3.3.2.2.4 Net rows created

This graph displays the net rows (rows created – rows destroyed) created over time.

5.3.3.2.2.5 Rows overflowed

This graph shows the estimated number of rows overflowed to variable length segments over time.

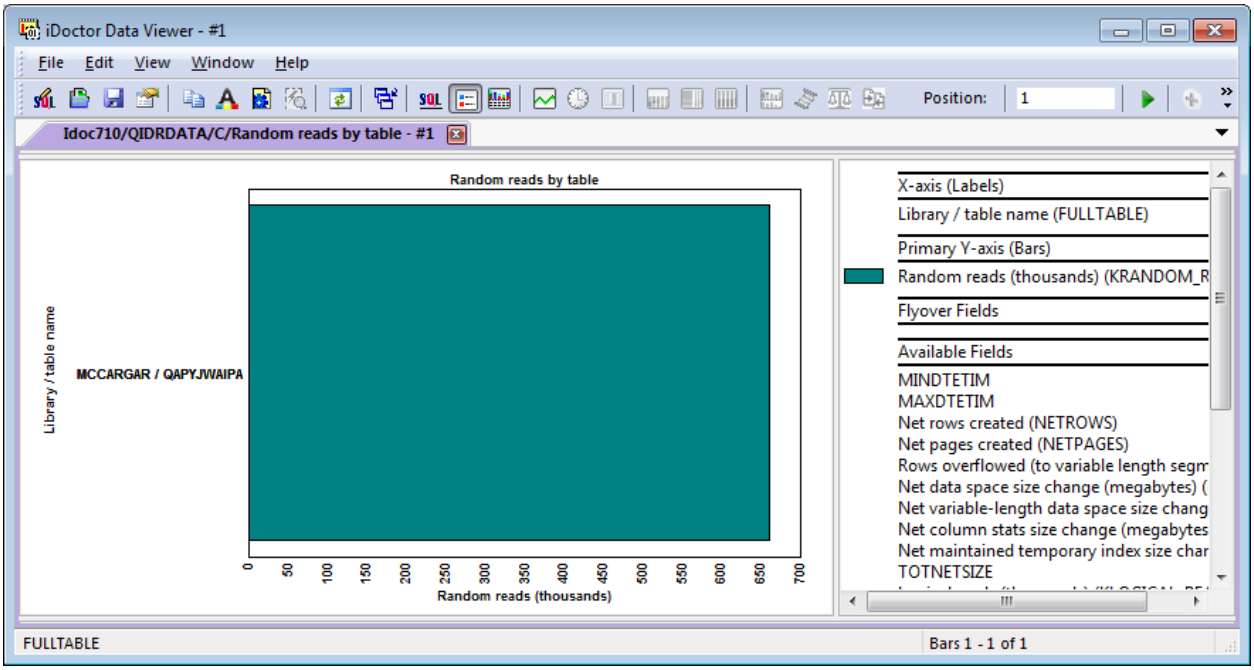
5.3.3.2.2.6 Net size changes

This graph shows the net data space size changes (in megabytes) over time.

5.3.3.2.2.7 Rankings

The same set of graphs covered previously are also provided as rankings charts by table name. These graphs can either be produced for the entire collection duration or based on the selected time period in one of the over-time charts.

An example is provided below:



Random reads by table graph

5.3.3.2.3 Table Statistics since last IPL Reports

The graphs available under the Table statistics since last IPL folder are:

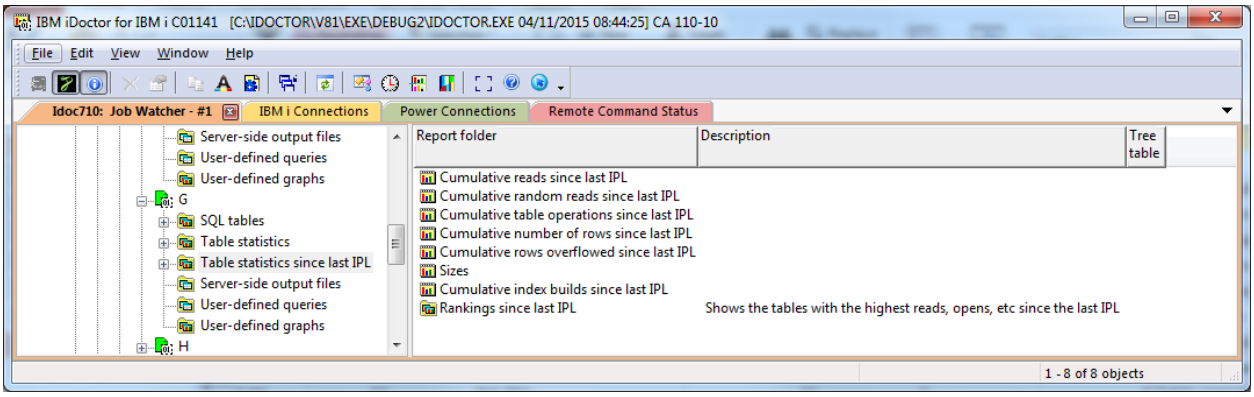


Table statistics since last IPL folder

These graphs are available for either all tables collected over time, or ranked by the desired metric(s). You may drill down from the over time graphs to the rankings graphs for the selected time period.

Graphs named “Cumulative” imply that delta statistics are not provided; the values on these graphs will always stay the same or increase over time.

Note: All graphs are based on the tables specified to be included in the data collection over the number of intervals and interval durations provided.

5.3.3.2.3.1 Cumulative reads since last IPL

This graph shows the cumulative logical, physical ans sequential reads over time with random reads on the second Y-axis.

Cumulative reads since last IPL

5.3.3.2.3.2 Cumulative random reads since last IPL

This graph displays the cumulative random reads by itself over time.

5.3.3.2.3.3 Cumulative table operations since last IPL

This graph displays cumulative values for several types of miscellaneous table functions: Opens, closes, inserts, updates, deletes, clear, copy and reorganize.

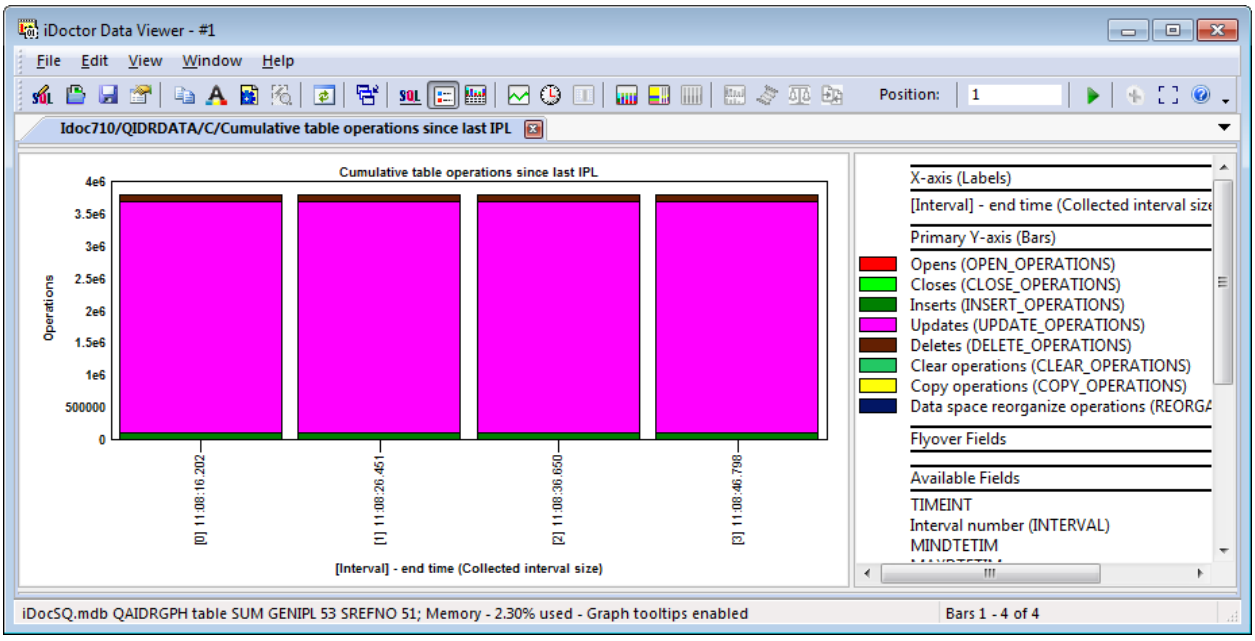


Table operations

5.3.3.2.3.4 Cumulative number of rows since last IPL

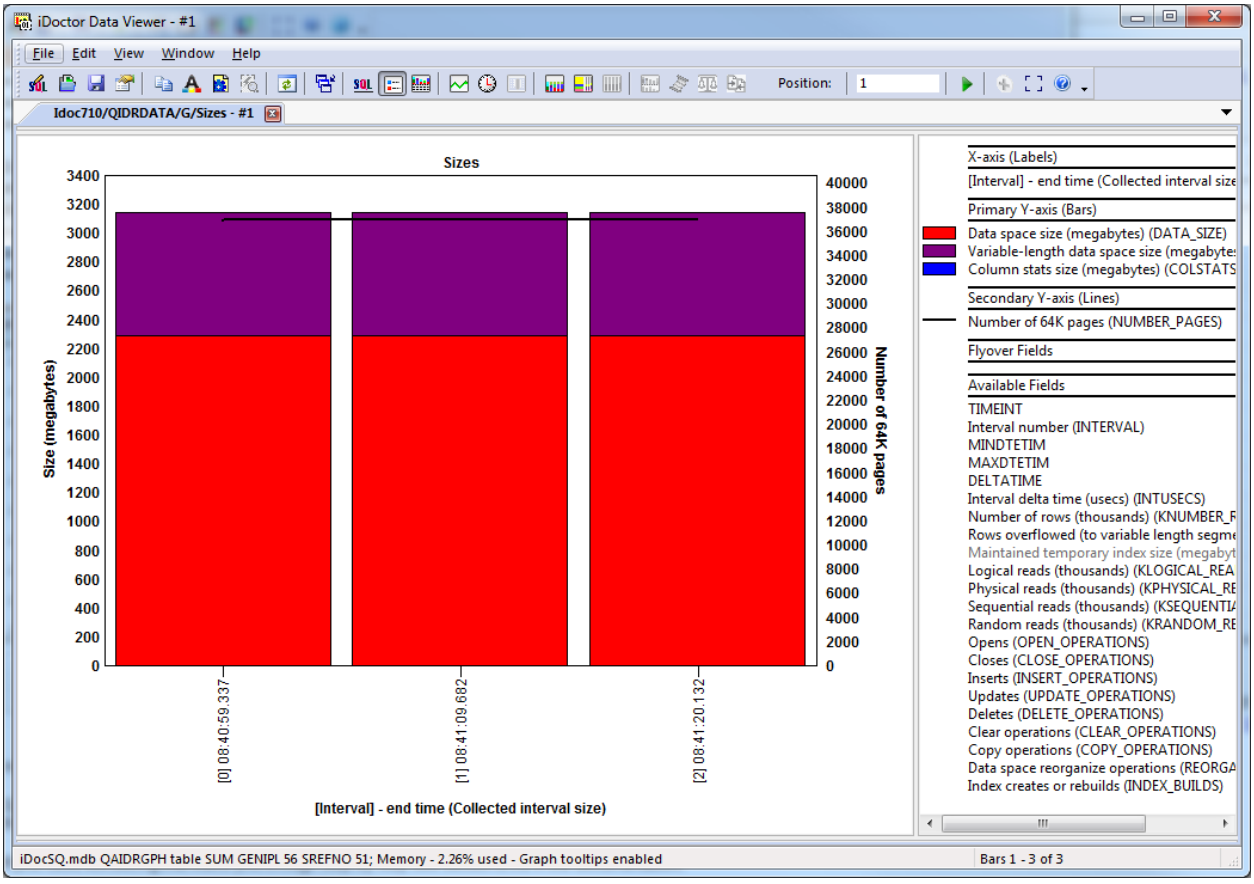
This graph displays the cumulative total number of rows for all tables collected and the total number of 64K pages over time.

5.3.3.2.3.5 Cumulative rows overflowed since last IPL

This graph shows the cumulative estimated number of rows overflowed to variable length segments over time.

5.3.3.2.3.6 Sizes

This graph shows the various types of table sizes (in megabytes) over time. Unlike the other graphs these values are not cumulative.

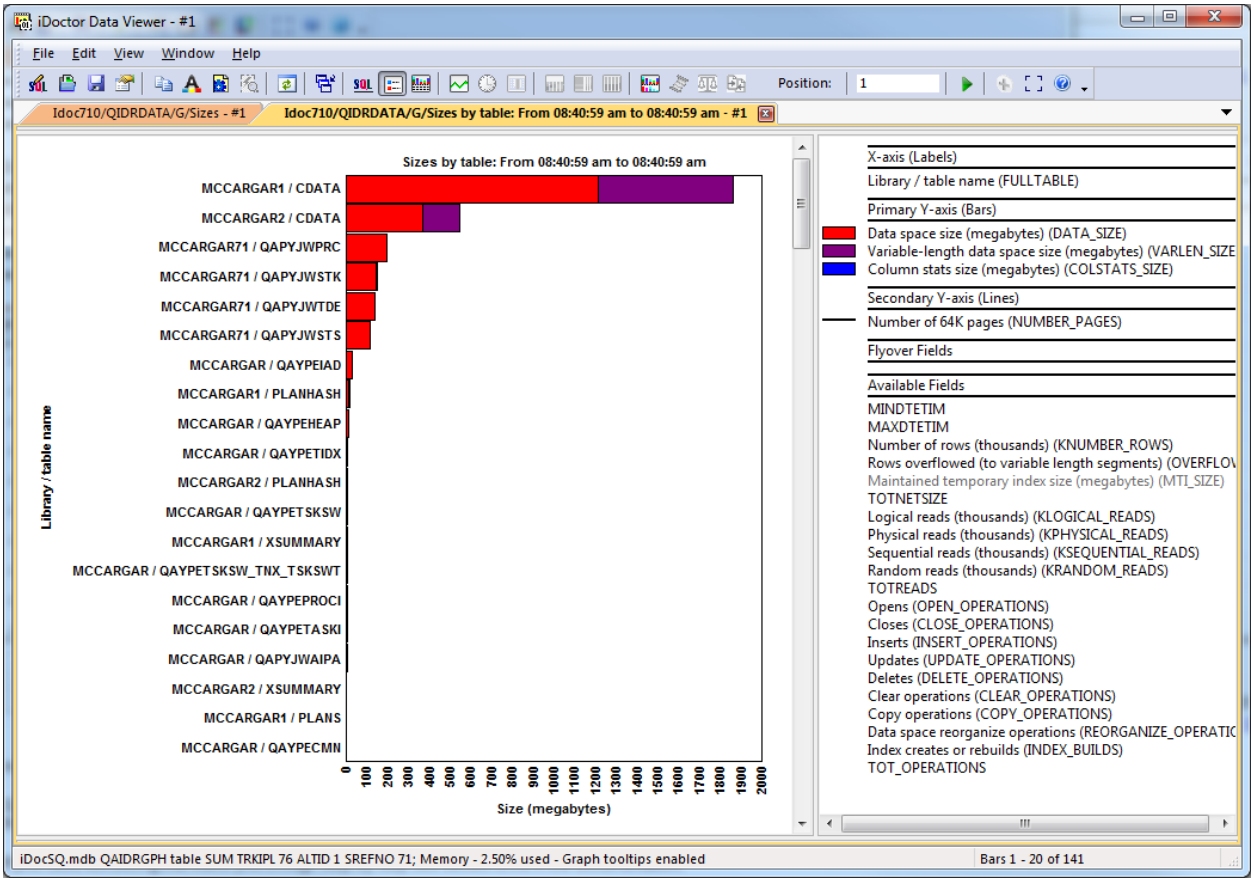


Sizes graph

5.3.3.2.3.7 Rankings

The same set of graphs covered previously are also provided as rankings charts by table name. These graphs can either be produced for the entire collection duration or based on the selected time period in one of the over-time charts.

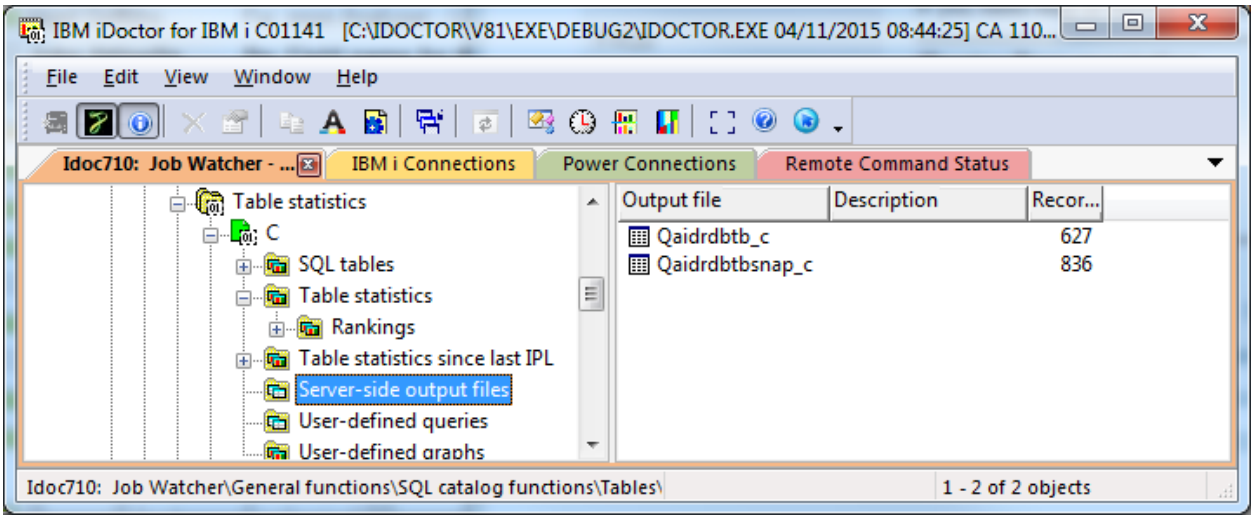
An example is provided below:



Sizes by table for a time period

5.3.3.2.4 Server-side output files

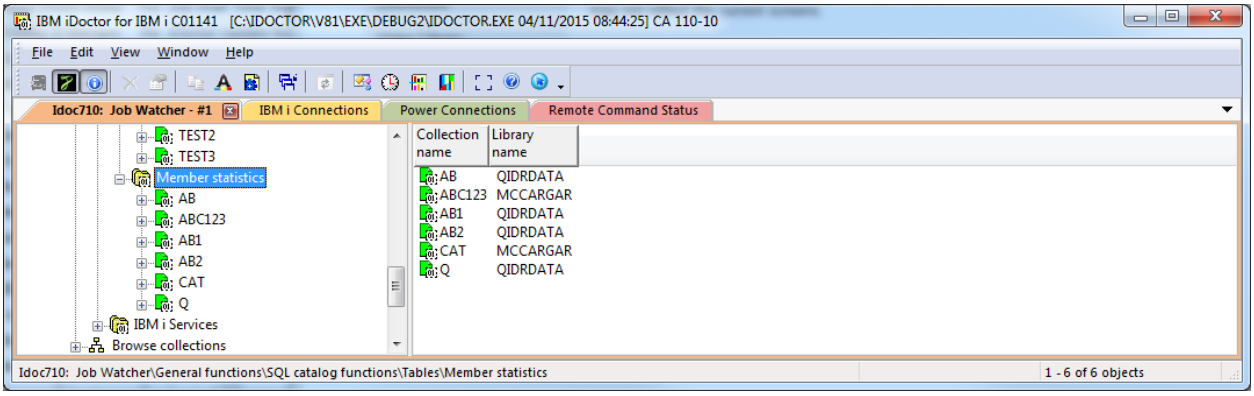
This folder under a Table Statistics collection provides a list of tables that were created as part of the collection. If only 1 snapshot was collected then only table QAIDRDBTBSNAP_<<COLNAME>> will exist in this folder. If more than 1 snapshot was captured then a table of delta statistics for each interval will be calculated and provided in QAIDRDBTB_<<COLNAME>>.



Server-side output files for a Table Statistics collection

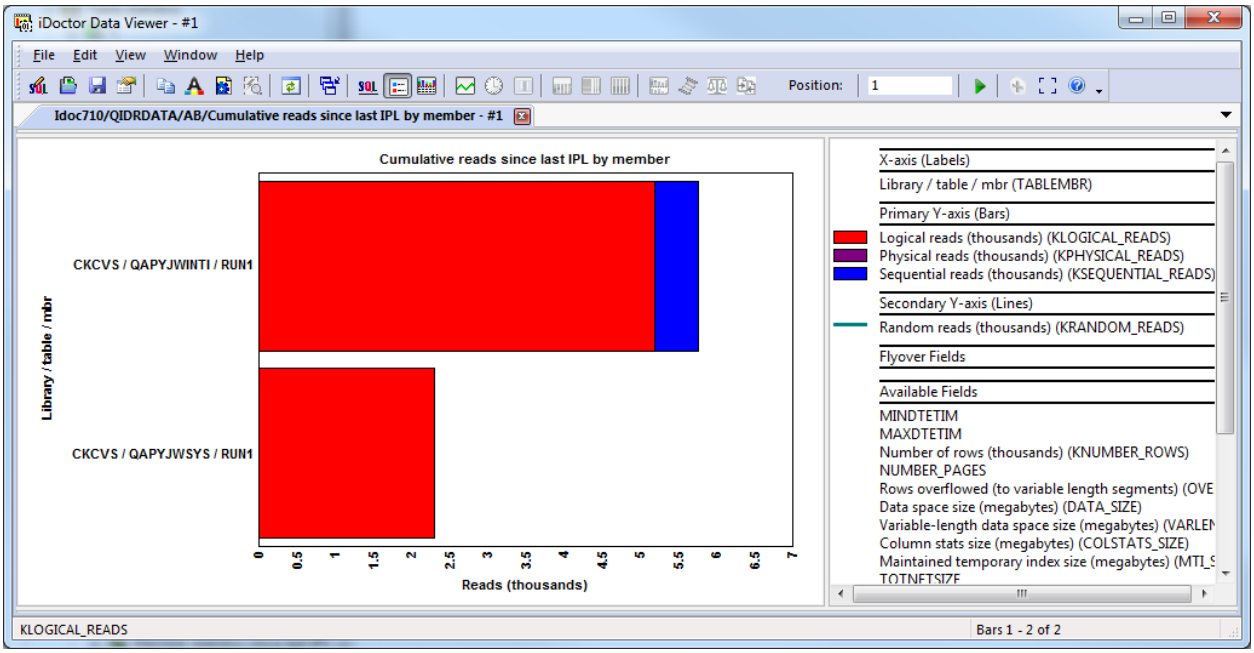
5.3.3.3 Member Statistics

This folder provides access to graphs and reports over already captured member statistics data produced by iDoctor. All of the collections found on the system of this type will be displayed.



List of member statistics collections found on a system within Job Watcher

An example of a Member statistics since last IPL graph is:

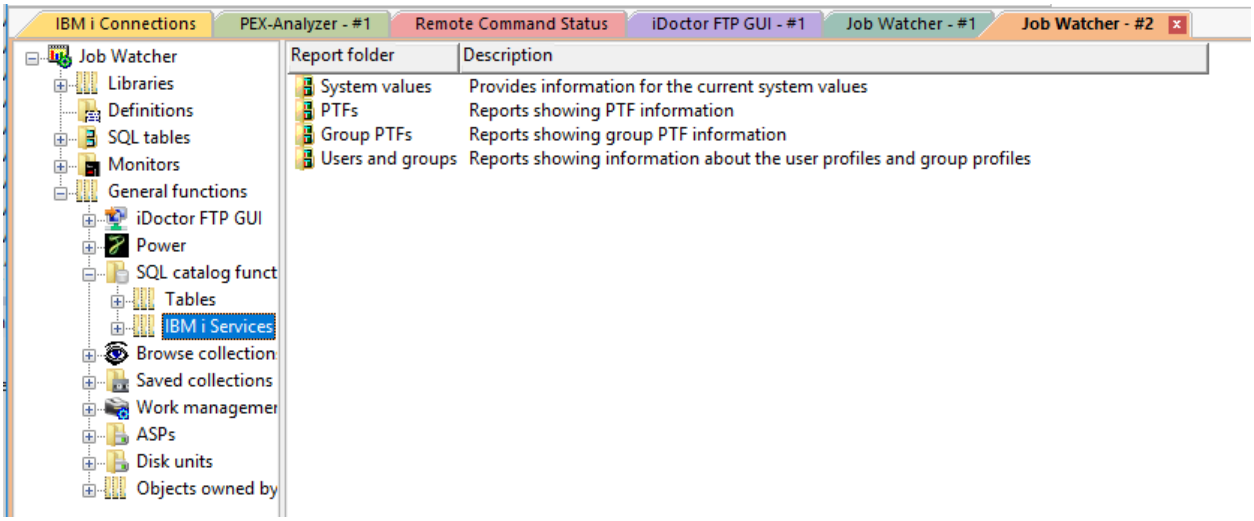


Cumulative reads since last IPL by member

Note: All the the member statistics collections and reports are identical to the ones for Table statistics except instead of providing table rankings graphs they are member ranking graphs. For this reason they are not documented, refer to the previous sections for information about the graphing options for member statistics if desired.

5.3.4 IBM i Services

The General functions -> SQL catalog functions -> IBM i Services folder provides the following reporting options from the data provided with DB2's SQL catalog tables:



General functions -> SQL catalog functions -> IBM i Services folder in Job Watcher

For more information on the data behind these reports visit this web page:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/DB2%20for%20i%20-%20Technology%20Updates

5.3.4.1 System values

These reports show information about the current IBM i system values. The reports available are:

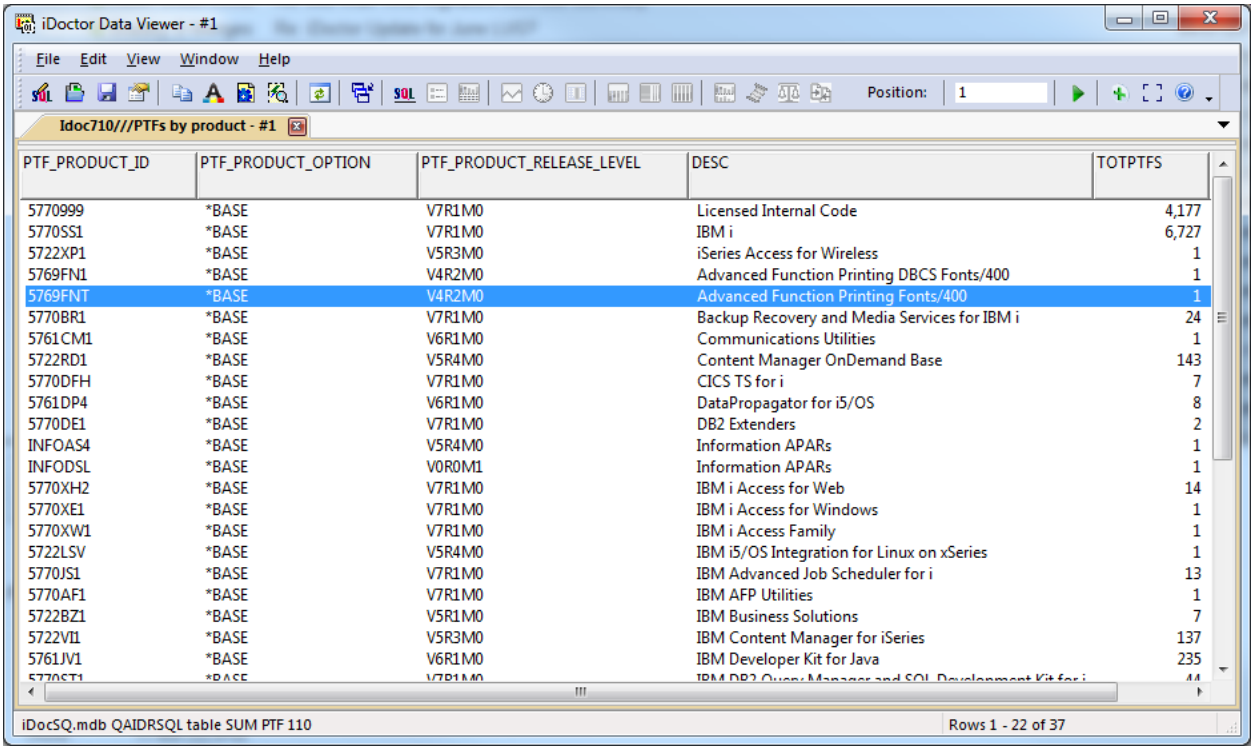
1) All system values
2) System values related by maximums

5.3.4.2 PTFs

These reports show PTF information for the current system.

5.3.4.2.1 PTFs by product

This report provides a count of the total PTFs for each product ID, option and release level.



PTFs by product

5.3.4.2.2 List all PTFs

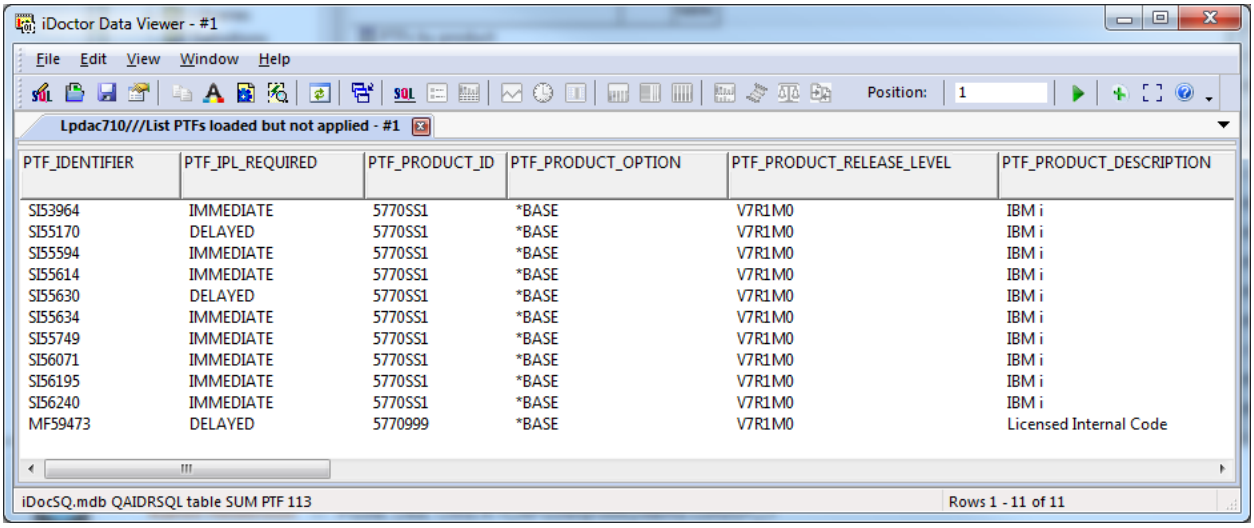
This report lists all the data available for all PTFs found on the system.

5.3.4.2.3 List PTFs impacted by the next IPL

This report informs you of any PTFs that will be effected by the next IPL.

5.3.4.2.4 List PTFs loaded but not applied

This report provides a list of PTFs loaded but not yet applied.



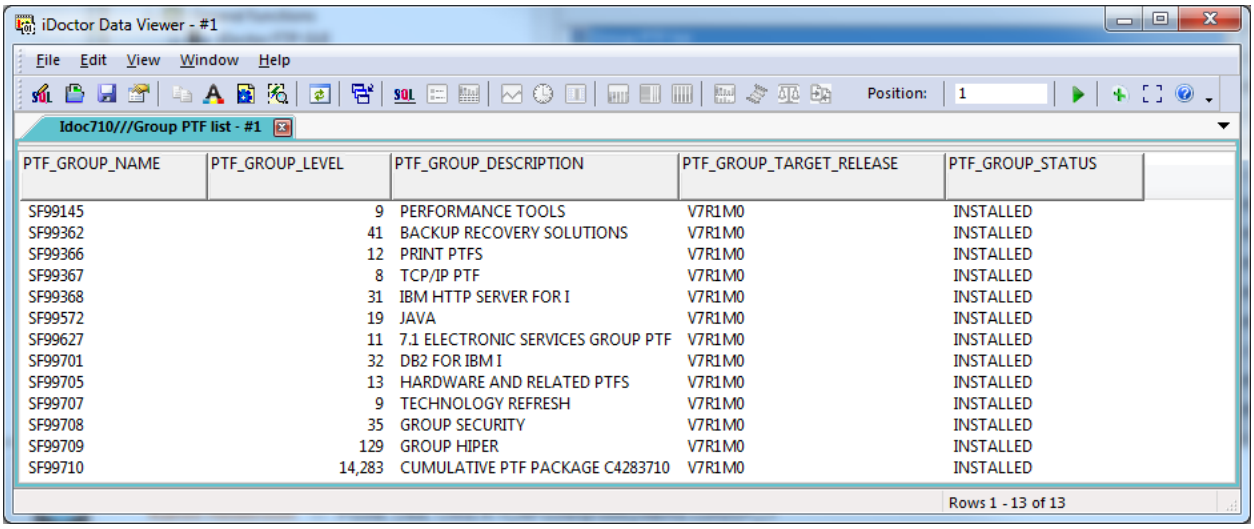
PTFs loaded but not applied

5.3.4.3 Group PTFs

These reports are provide information about the group PTFs found on the system.

5.3.4.3.1 Group PTF list

This report lists all group PTFs found on the system and the PTF group level installed.



PTF_GROUP_NAME	PTF_GROUP_LEVEL	PTF_GROUP_DESCRIPTION	PTF_GROUP_TARGET_RELEASE	PTF_GROUP_STATUS
SF99145	9	PERFORMANCE TOOLS	V7R1M0	INSTALLED
SF99362	41	BACKUP RECOVERY SOLUTIONS	V7R1M0	INSTALLED
SF99366	12	PRINT PTFS	V7R1M0	INSTALLED
SF99367	8	TCP/IP PTF	V7R1M0	INSTALLED
SF99368	31	IBM HTTP SERVER FOR I	V7R1M0	INSTALLED
SF99572	19	JAVA	V7R1M0	INSTALLED
SF99627	11	7.1 ELECTRONIC SERVICES GROUP PTF	V7R1M0	INSTALLED
SF99701	32	DB2 FOR IBM I	V7R1M0	INSTALLED
SF99705	13	HARDWARE AND RELATED PTFS	V7R1M0	INSTALLED
SF99707	9	TECHNOLOGY REFRESH	V7R1M0	INSTALLED
SF99708	35	GROUP SECURITY	V7R1M0	INSTALLED
SF99709	129	GROUP HIPER	V7R1M0	INSTALLED
SF99710	14,283	CUMULATIVE PTF PACKAGE C4283710	V7R1M0	INSTALLED

Group PTF List

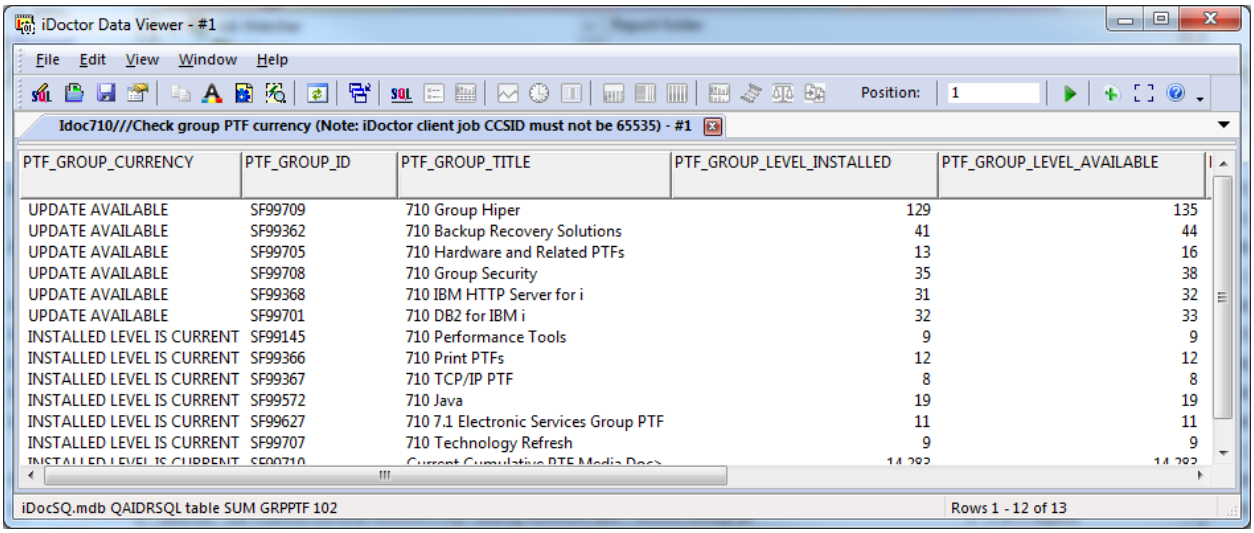
5.3.4.3.2 DB PTF Group Level

This report simply lists the DB PTF Group level.

5.3.4.3.3 Check group PTF currency

This option will check each group PTF and look for any updates available. It will inform you of the level installed and the level that is available.

An example of this report is:

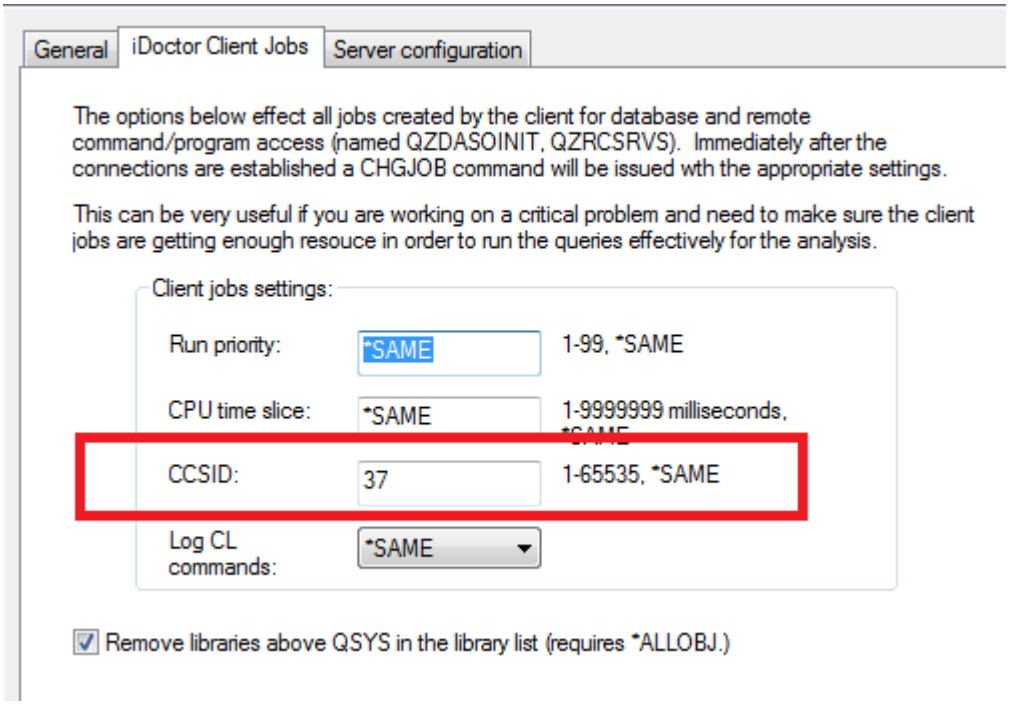


PTF_GROUP_CURRENCY	PTF_GROUP_ID	PTF_GROUP_TITLE	PTF_GROUP_LEVEL_INSTALLED	PTF_GROUP_LEVEL_AVAILABLE
UPDATE AVAILABLE	SF99709	710 Group Hiper	129	135
UPDATE AVAILABLE	SF99362	710 Backup Recovery Solutions	41	44
UPDATE AVAILABLE	SF99705	710 Hardware and Related PTFs	13	16
UPDATE AVAILABLE	SF99708	710 Group Security	35	38
UPDATE AVAILABLE	SF99368	710 IBM HTTP Server for i	31	32
UPDATE AVAILABLE	SF99701	710 DB2 for IBM i	32	33
INSTALLED LEVEL IS CURRENT	SF99145	710 Performance Tools	9	9
INSTALLED LEVEL IS CURRENT	SF99366	710 Print PTFs	12	12
INSTALLED LEVEL IS CURRENT	SF99367	710 TCP/IP PTF	8	8
INSTALLED LEVEL IS CURRENT	SF99572	710 Java	19	19
INSTALLED LEVEL IS CURRENT	SF99627	710 7.1 Electronic Services Group PTF	11	11
INSTALLED LEVEL IS CURRENT	SF99707	710 Technology Refresh	9	9

Check group PTF currentcy

Note: In order for this report to work, the iDoctor client QZDASOINIT jobs' CCSID must not be 65535. You can check or change this by right-clicking the component icon (i.e. Job Watcher, PEX Analyzer, etc.)

in the component view and using Properties -> iDoctor Client Jobs then change the CCSID to 37 if necessary.



Component properties -> iDoctor Client Jobs -> CCSID setting

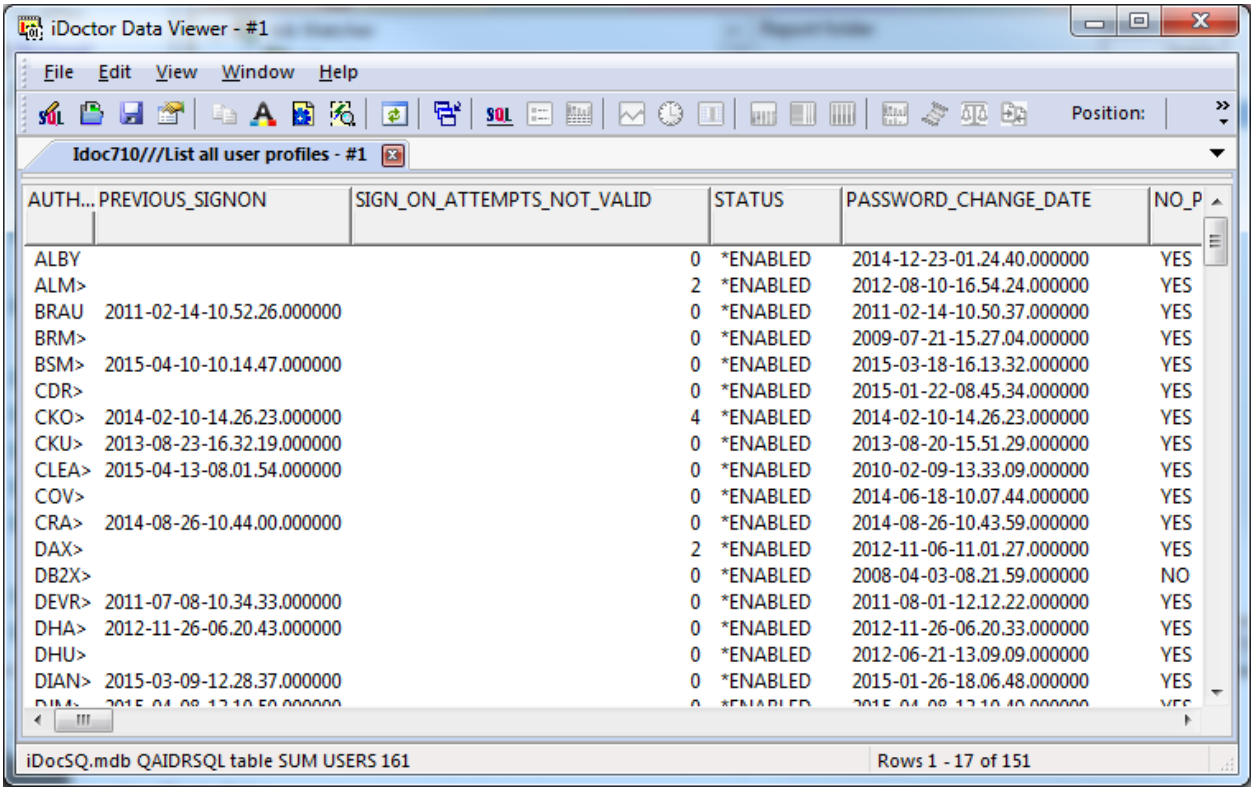
5.3.4.4 Users and groups

5.3.4.4.1 List all group profiles

This report lists all the group profile names and the user profiles that belong to each.

5.3.4.4.2 List all user profiles

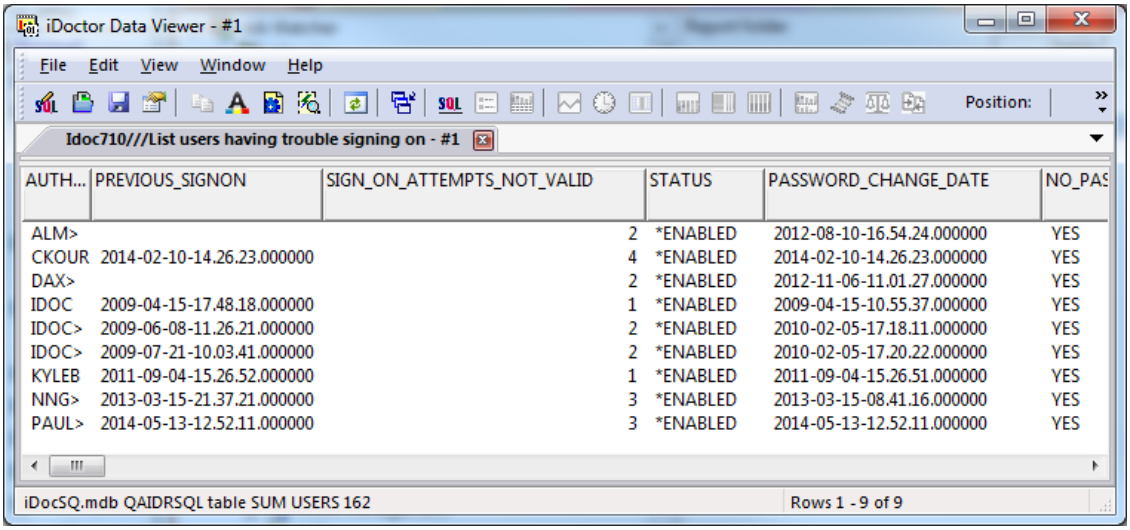
This report provides detailed information about all user profiles installed on a system.



List all user profiles

5.3.4.4.3 List users having trouble signing on

This report will list all user profiles that have invalid signon attempts to the system.



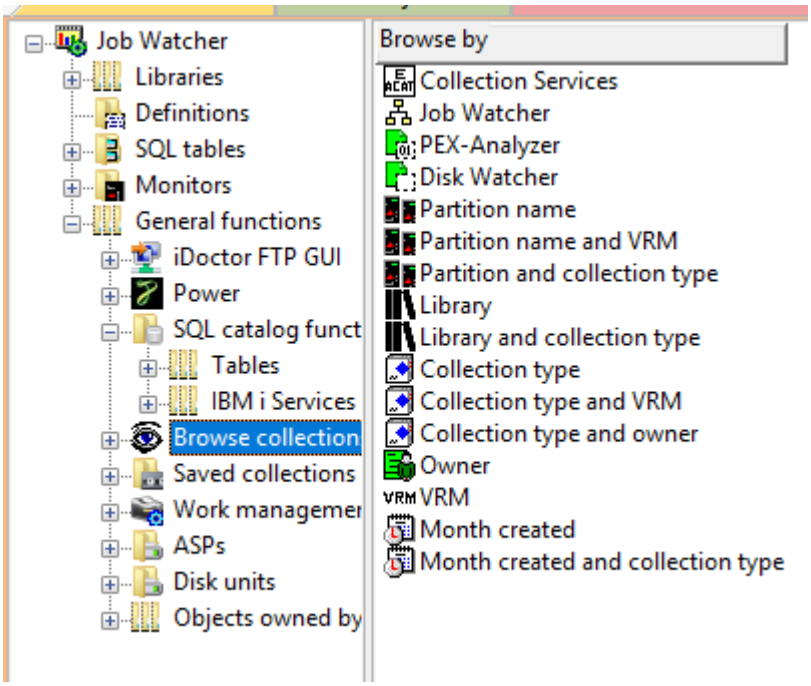
List users having trouble signing on

5.4 Browse Collections

The Browse Collections function displays lists of collections on your system in various ways. All current IBM i collection types used by iDoctor are included in the collections shown.

Note: This interface is only accessible if the system has a license to both PEX Analyzer and Job Watcher.

This option works by first building a database of the available collections and then grouping and displaying the results in various folders. An example of the ways to browse collections are:



Browse Collections options

Within any of these folders are additional folders that appropriately group the data (partition name, library name, etc.) and within those are the lists of collections. In most cases all of the normal graphing/reporting options found in the applicable iDoctor component are also available when right-clicking collections.

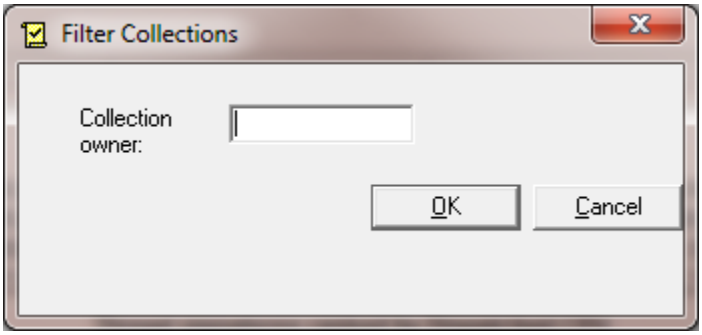
5.4.1 Menu Options

When right-clicking the Browse Collections folder the following options are available:

Menu	Description
Filter collections	This menu brings up a window that lets you filter the collections shown by collection owner.
Refresh collections database	This option will refresh the collections database. The database tables are not recreated, but any changes to collections on the system will be checked for and updated in the database.
Full rebuild collections database	This option will completely replace the collections database which may take several minutes. This option may be required if changes have been made to iDoctor (such as additional fields added to the database.)

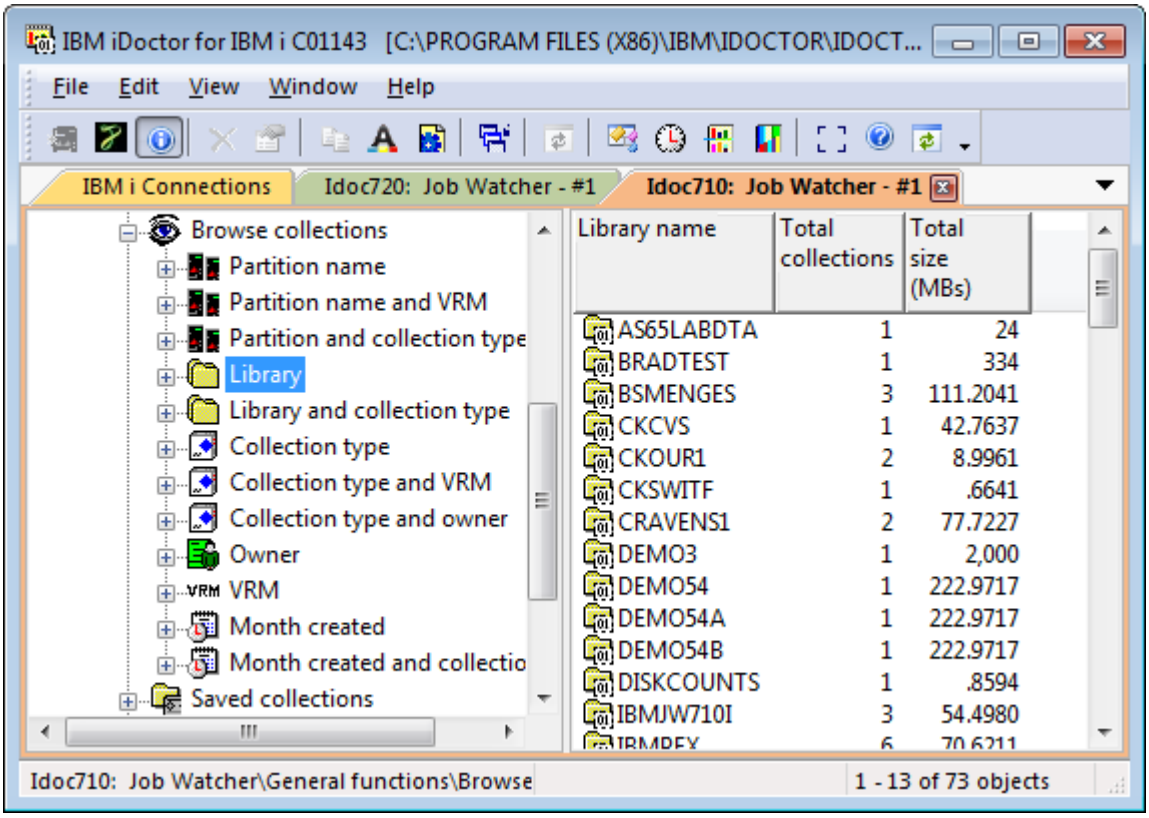
5.4.2 Filter collections

Use this interface to filter the list of collections displayed in the Browse Collections interface by collection owner.



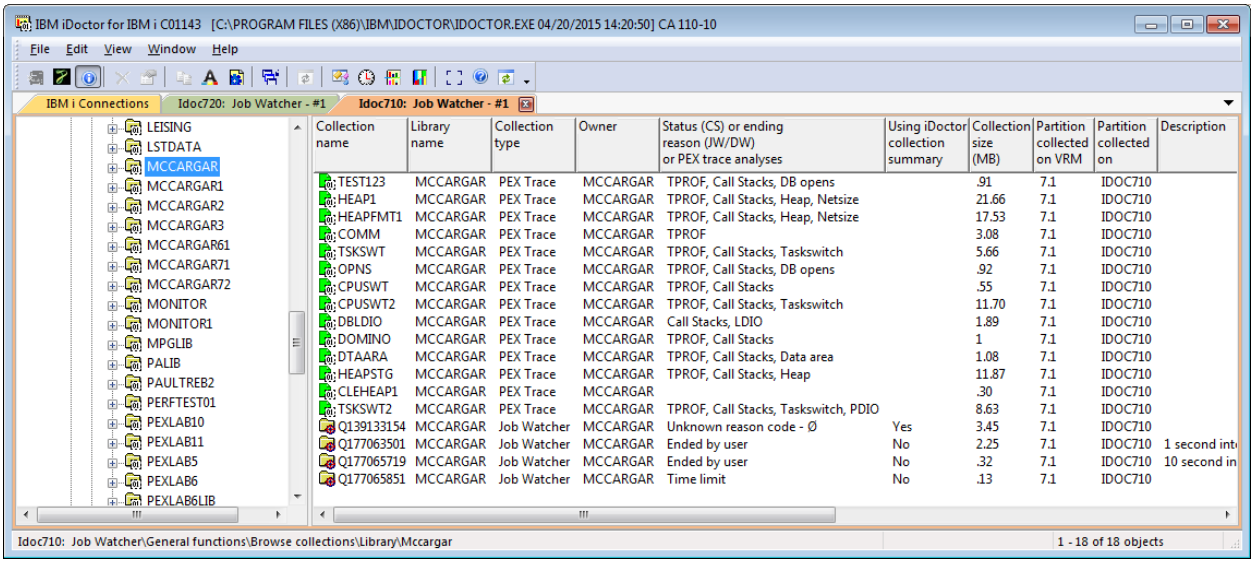
5.4.3 Examples

This folder displays each library with the total number of collections and the estimated total size (in megabytes)

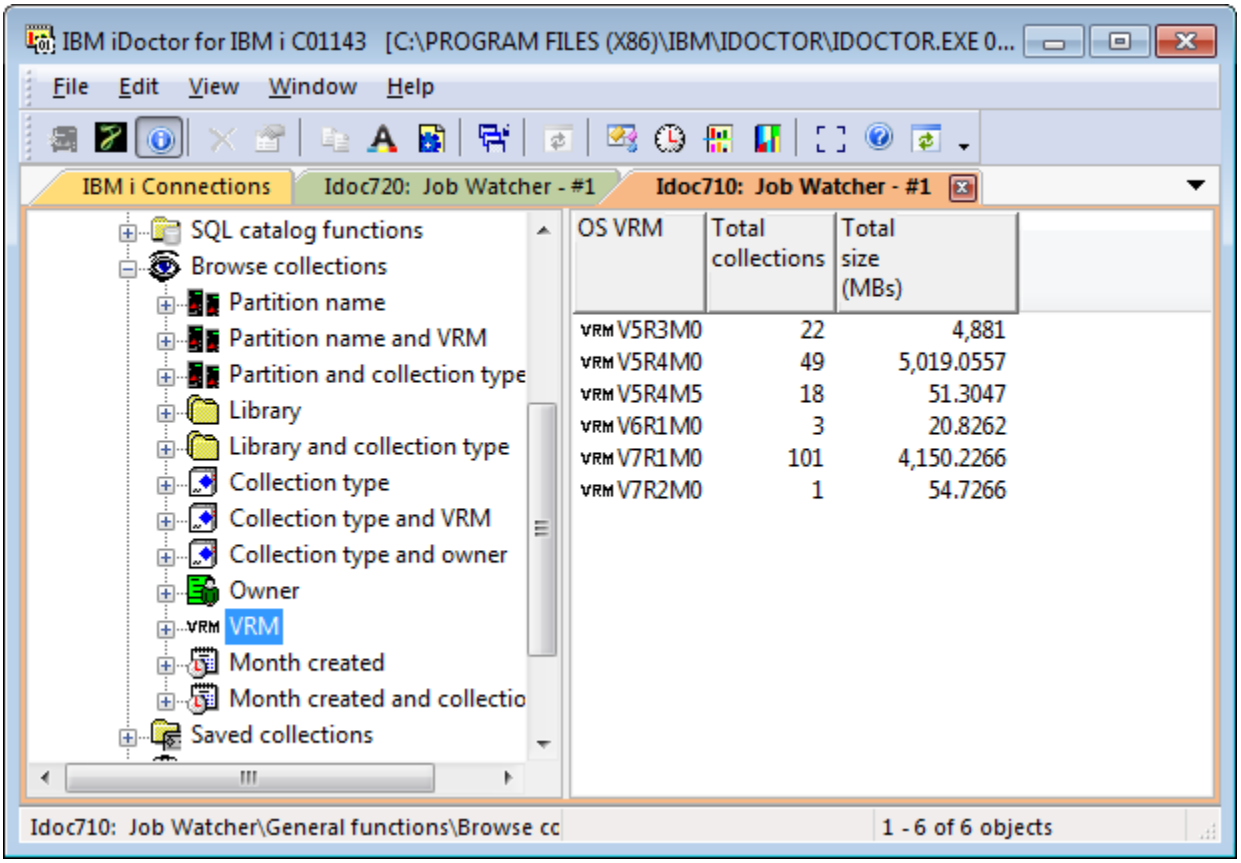


Browse Collections -> Library folder

The next example shows the list of Job Watcher collections in a library:



Browse Collections -> Library MCCARGAR



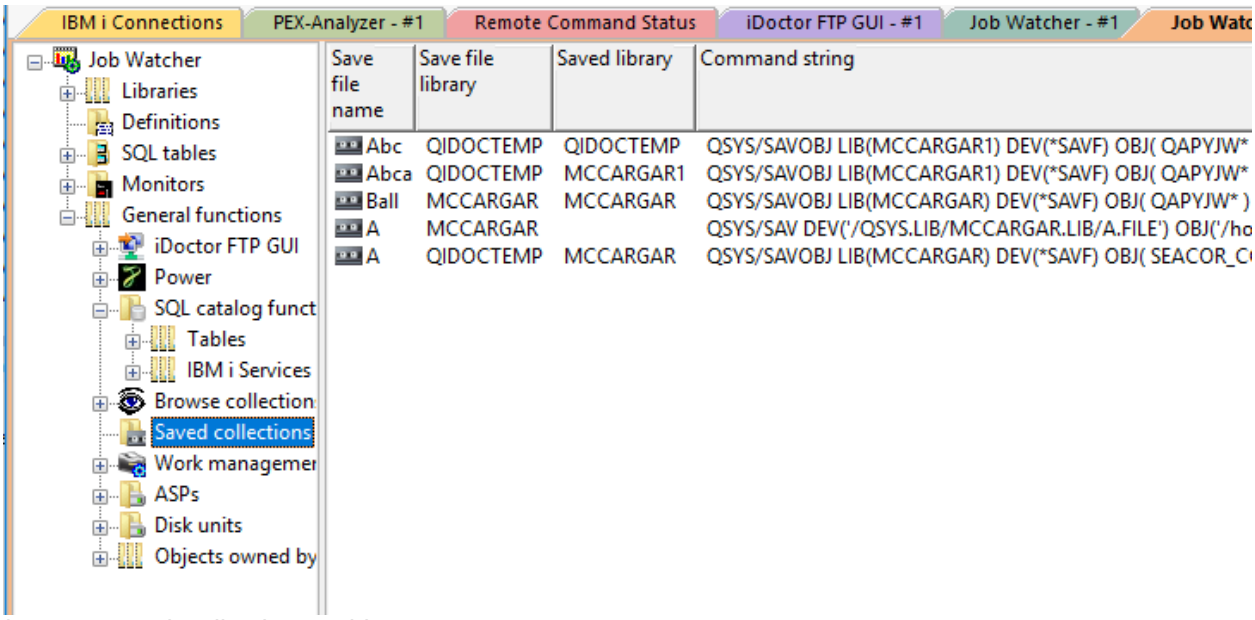
Browse Collections -> VRM

5.5 Saved collections

Most components allow you to save iDoctor collections to a save file. The Saved Collections folder within a component allows you to work with any saved collections found on the system.

Note: This interface will only show collections that were saved using the iDoctor GUI.

The interface looks like the following:



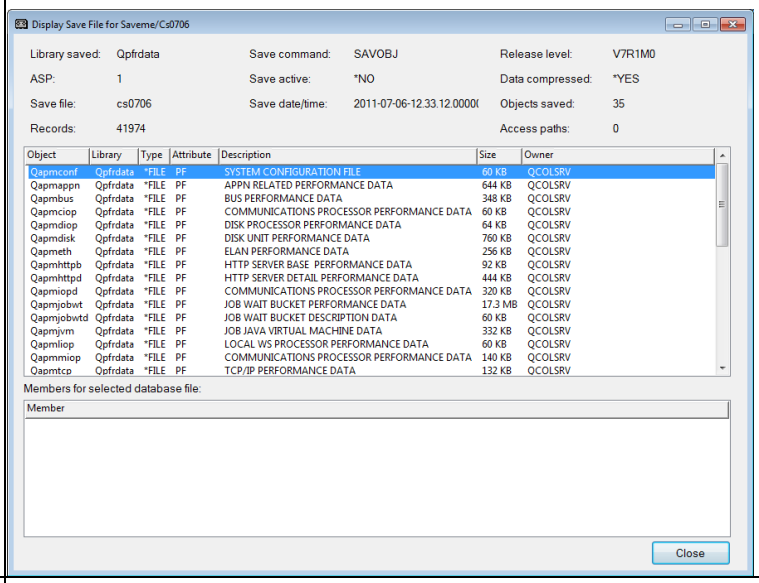
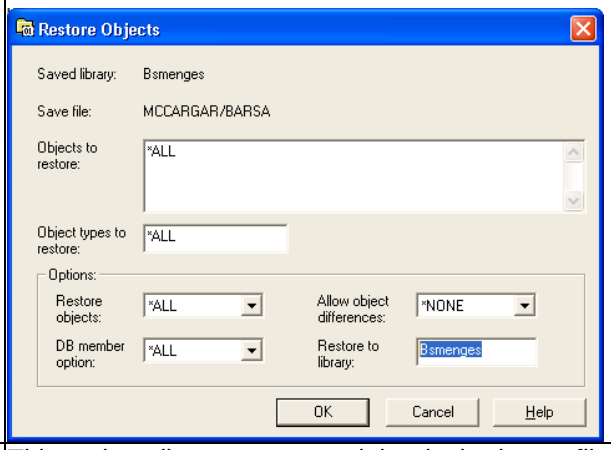
iDoctor Saved Collections Folder

The folder contains a row for every save file found that contains an iDoctor collection saved using the iDoctor GUI. The following table describes the data shown in this folder:

Field	Description
Save file name	The name of the save file.
Save file library	Library name of the save file.
Saved library	The library that was saved.
Command string	The command string that was used to save the collection.

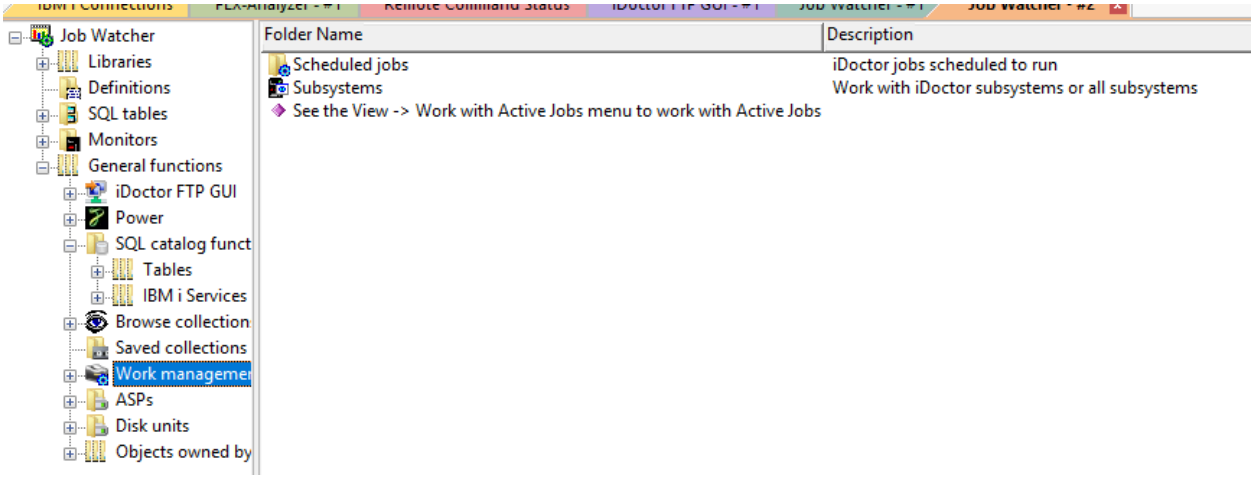
5.5.1 Menu Options

By right clicking on a save file there are a number of options available:

Popup Menu	Description
Display	Displays the contents of the save file into a new window. 
Delete	Removes the selected save files from the system.
Restore	This option allows you to restore the saved collection to the desired library on the current system. Change the **Restore to library** parameter if you desire to the restore the collection to a different library than it was originally created in. 
Transfer to	This option allows you to send the desired save files to another system or to IBM.
Properties	Displays object information for the save file.

5.6 Work Management

This folder contains functions for working with the scheduled jobs, active jobs and subsystems on the current system.



Work Management folder

5.6.1 Scheduled Jobs

Most components allow you to schedule collections to run at a later time. The Scheduled Jobs folder allows you to work with the iDoctor created scheduled jobs on the system.

The folder contains a row for every iDoctor scheduled job defined on the system.
The following table describes the data shown in this folder:

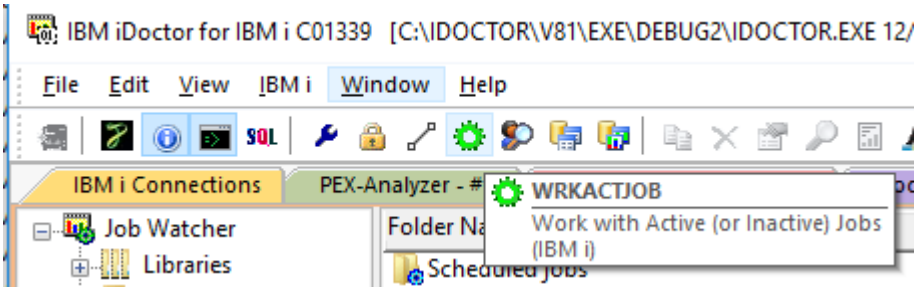
Field	Description
Job	The name of the job that will be created when the scheduled job is submitted.
Type	The type of iDoctor job that will run.
Submitted by	The user profile on the system that created the iDoctor scheduled job.
Status	The status of the scheduled job. This will either be Scheduled or Held. If the status is Held the scheduled job must be released before it will run.
Scheduled date/time	The date and time when the scheduled job will run. This may list either a specific date and time, a specific day and time to run on a weekly basis, or Daily if this job should run every day.
Next submit date	The date when this job is next scheduled to run.
Job entry number	The entry number of the scheduled job as defined on the system.
Description	A description that has been assigned to the scheduled job entry.
Command	The command string that will be executed when the scheduled job runs.

By right clicking on a scheduled job entry there are a number of options available:

Popup Menu	Description
Submit immediately	Submits the scheduled job immediately. The scheduled job will still run again at the next scheduled date and time.
Hold/Release	Hold or release the selected scheduled job entry. If held the scheduled job will not run.
Delete	Removes the selected scheduled job entries from the system.

5.6.2 Active jobs

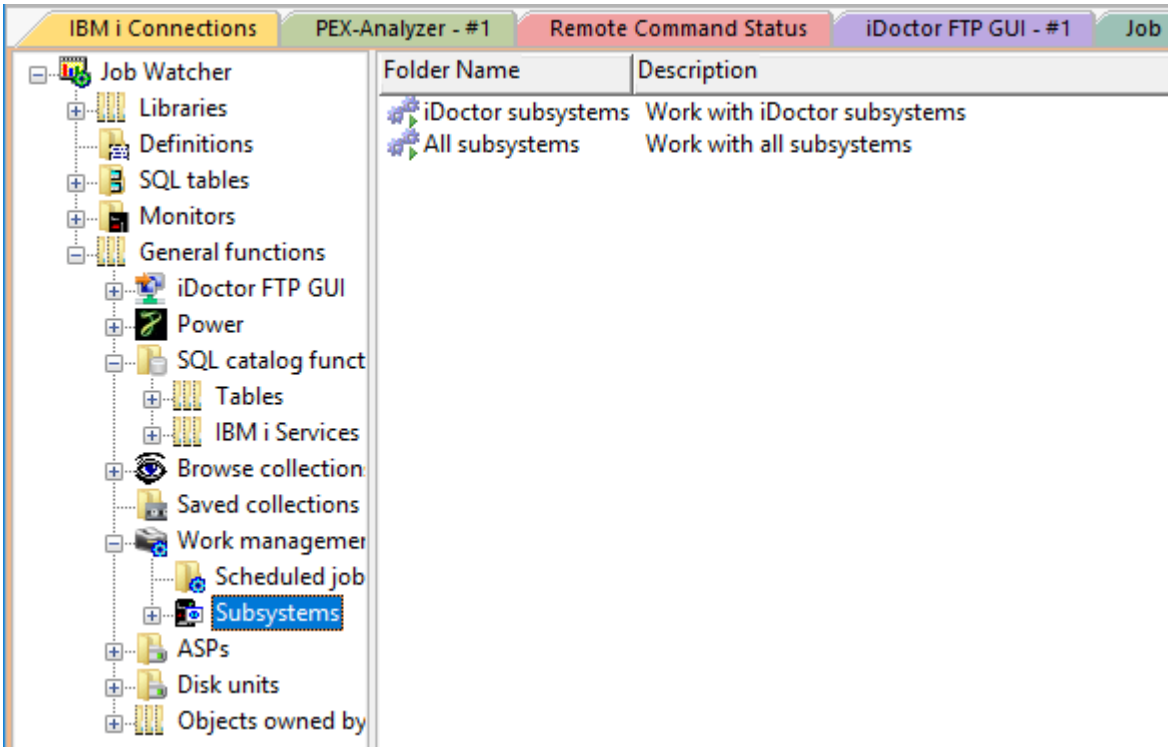
To work with Active jobs on the system use the WRKACTJOB pane/toolbar button.



WRKACTJOB toolbar button

5.6.3 Subsystems

The subsystems folder provides the following 2 options for working with subsystems on the current system:



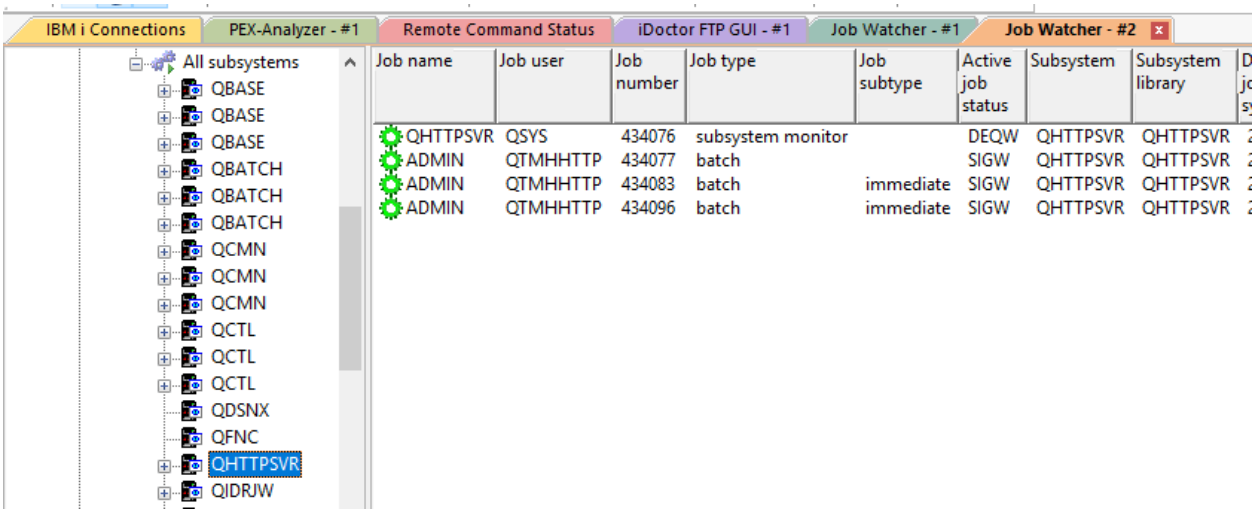
Work Management -> Subsystems folder

5.6.3.1 iDoctor subsystems

The iDoctor subsystems shown are based on those specified at install time. With the latest builds this will just show a single subsystem (default QSYS/QIDRJW.)

5.6.3.2 All subsystems

This option displays a list of all subsystems found on the system and allows you to work with the jobs running within each subsystem.

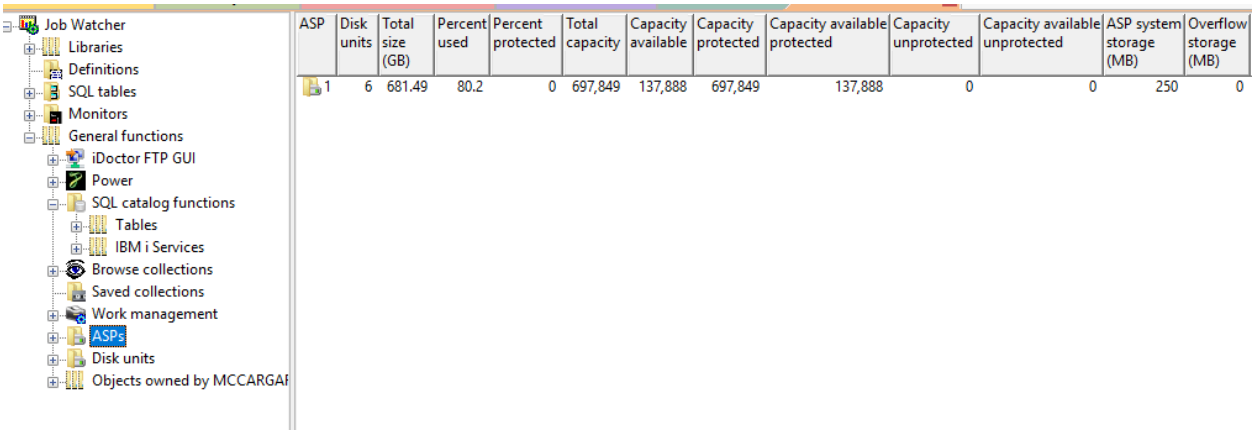


Displaying the active jobs within a subsystem

5.7 ASPs

The ASPs folder displays disk information for the current system on a per ASP basis. You can also expand the ASPs to see information about the disk units within the selected ASP. This information is similar to WRKDSKSTS but includes extra fields not found there.

This interface looks like the following:



iDoctor ASPs Folder

The folder contains a row for every ASP found on the system. The following table describes the data shown in this folder:

Tip: Right-click the ASPs folder and pick the Select fields menu to rearrange or add/remove fields from this list.

Field	Description
ASP	The ASP identifier.
Disk units	Total number of disk units in the ASP.
Total size (GB)	The total size of all disk units in the ASP (in gigabytes).
% used	The percent of available disk space used.
% warning threshold	The SLIC warning threshold of available disk space used for the ASP.
% protected	The percent of ASP disk space that is protected by mirroring or device parity
Type	The type of ASP. The possible values are: System ASP User ASP (no libraries) User ASP (with libraries)
ASP system storage (MB)	The amount of storage in megabytes currently allocated in the ASP for system use.
Overflow storage (MB)	The number of megabytes of storage that has overflowed from the user ASP into the system ASP.
Error log size (MB)	The number of megabytes of auxiliary storage allocated to the error log.
Machine log size (MB)	The number of megabytes of auxiliary storage allocated to the machine log.
Machine trace size (MB)	The number of megabytes of auxiliary storage allocated to the machine trace.
Main storage dump size (MB)	The number of megabytes of auxiliary storage allocated to the main storage dump space.
Microcode size (MB)	The number of megabytes of auxiliary storage allocated to the microcode and space used by the microcode.
End immediately if system ASP full	This field has meaning in the system ASP (ASP 1) only. *No* If a request for space in the system ASP cannot be satisfied because there is not enough storage, the system will be allowed to continue running. *Yes* If a request for space in the system ASP cannot be satisfied because there is not enough storage, the system will be ended immediately.
Compressed units	Whether there are compressed disk units in the ASP. The following are the possible values: *None* No compressed disk units in this ASP. *Some* Compressed and uncompressed disk units in this ASP. *All* All disk units in this ASP are compressed.
Balance status	The current status of the balance function for this ASP. The following special values are returned: *No activity* No balance activity has occurred for this ASP. *Running* The ASP balance function is currently running for this ASP. *Ending* The ASP balance function is currently in the process of ending. Either the time limit has run out or the End ASP Balance (ENDASPBAL) command

		was issued for this ASP.
	Ended before balance complete	The ASP balance function has run, but was ended before the ASP was completely balanced. The Start ASP Balance (STRASPBAL) command can be used to restart the balance function.
	Completed	The ASP balance function has completed running. The ASP is completely balanced.
Balance type	The type of balance activity that is currently running or was done last.	
Mirroring role	The current role of a physical IASP.	
	Not configured	Remote IASP mirroring is not configured.
	No physical IASP copy	System does not own a physical IASP copy.
	Unknown	Remote mirror role is unknown.
	Has detached mirror copy	System owns a detached mirror copy.
	Has mirror copy	System owns the mirror copy.
	Has production copy	System owns the production copy

By right clicking on an ASP within the list the following options are available:

Popup Menu	**Description**
Reset Statistics	This option resets the disk unit level statistics for the selected ASP.
Add Disk Watcher Definition	Defines a new Disk Watcher definition on the system. The interface will preselect the definition to only include the ASPs selected.
Add PEX Filter	Defines a new PEX filter using the selected ASP.

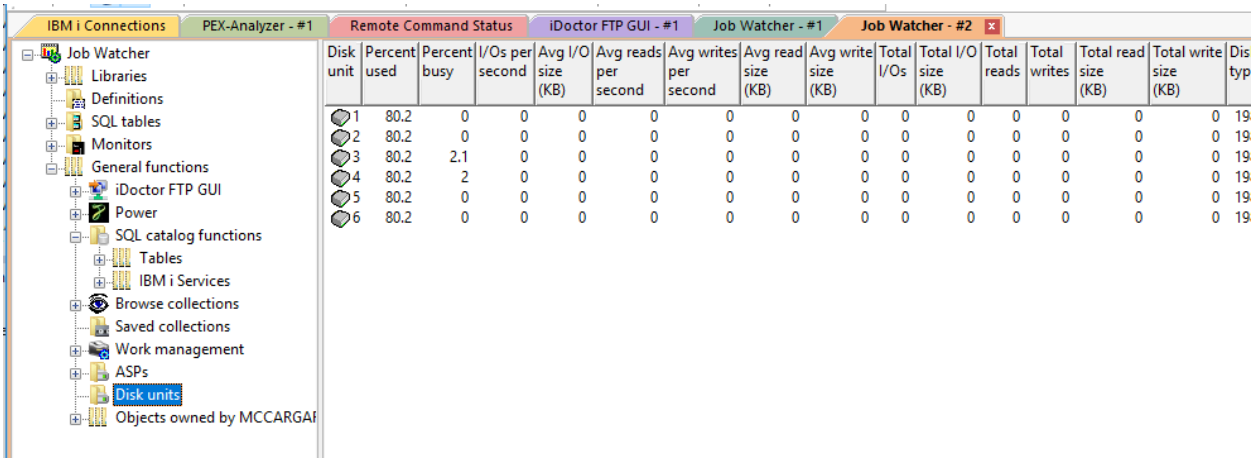
Expanding the ASP will show the disk units within. See the next section for more information. The functionality available for units within an ASP is identical to that available in the Disk Units folder.

5.8 Disk units

The Disk Units folder displays disk information for the current system. This information is similar to WRKDSKSTS but includes extra fields not found there.

This interface looks like the following:

Tip: Right-click the Disk units folder and pick the Select fields menu to rearrange or add/remove fields from this list.



iDoctor Disk units Folder

The folder contains a row for every disk unit found on the system. The following table describes the data shown in this folder:

Field	Description
Disk unit	The disk unit identifier
Disk type	The disk unit type
ASP	The ASP the disk unit belongs to.
Size (GB)	The total capacity of the disk unit (in gigabytes).
% used	The percent of available disk space used.
% busy	The percent of time the disk unit was busy during the sample period.
I/Os per second	The rate of I/Os occurring per second during the sample period.
Avg I/O size (KB)	The average size of each I/O occurring during the sample period.
Avg reads per second	The average number of reads per second during the sample period.
Avg writes per second	The average number of writes per second during the sample period.
Avg read size (KB)	The average size of each read during the sample period.
Avg write size (KB)	The average size of each write during the sample period.
Total I/Os	The total number of I/Os during the sample period.
Total I/O size (KB)	The total size of all I/Os added together during the sample period.
Total reads	The total number of reads during the sample period.
Total writes	The total number of writes during the sample period.
Total read size (KB)	The total size of all reads added together during the sample period.
Total write size (KB)	The total size of all writes added together during the sample period.
Compression status	The current status of compression for this disk unit. The possible values are: Active, Not active
Disk protection type	The type of protection that has been assigned to this disk unit. The following values are returned: *None* — No storage protection has been set up for this disk unit. *Mirrored* — This disk unit has been set up with mirrored protection. *Parity* — This disk unit is part of a parity protection array.
Mirrored status	The mirrored status of the unit. *Blank* — Not mirrored *1* — This mirrored unit of a mirrored pair is active (that is, online with current data). *2* — This mirrored unit is being synchronized. *3* — This mirrored unit is suspended.
Mirrored reported	This mirrored unit reported present. Information concerning this unit may or may not be current. *0* The mirrored unit is missing. Information concerning the unit may not be current. *1* The mirrored unit reported at the time this information was gathered. The information current to that point in time.

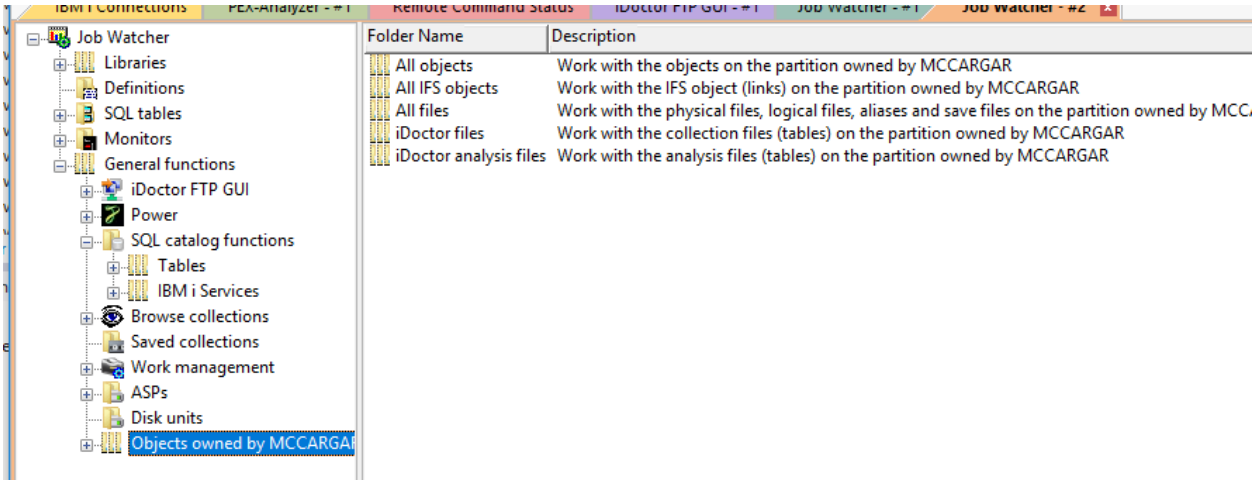
Mirrored protected	The mirrored status of the mirrored pair of which this unit is a member. *0* One mirrored unit of a mirrored pair is not active. *1* Both mirrored units of a mirrored pair are active.
Disk model	The model of the disk unit.
Disk serial number	The serial number of the disk unit.
Resource name	The unique system-assigned name of the disk unit.

By right clicking on a disk unit within the list the following options are available:

Popup Menu	**Description**
Add Disk Watcher Definition	Defines a new Disk Watcher definition on the system. The interface will preselect the definition to only include the units selected.
Add PEX Filter	Defines a new PEX filter using the selected disk units.

5.9 Objects owned by user

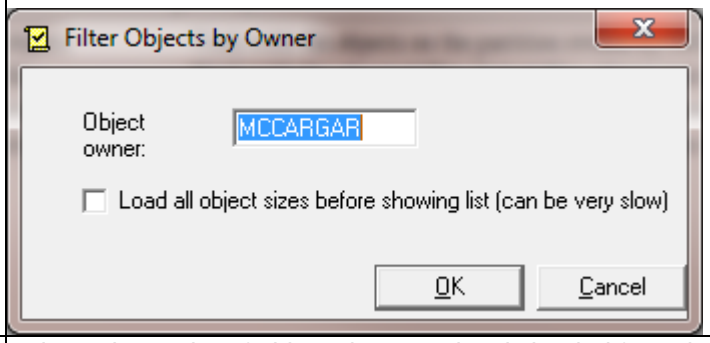
This folder provides options for viewing the disk space consumption of objects or files on the system by the current user. Options are available to view the disk space usage for all objects or only objects created by iDoctor.



Objects owned by <User> folder options

5.9.1 Menu options

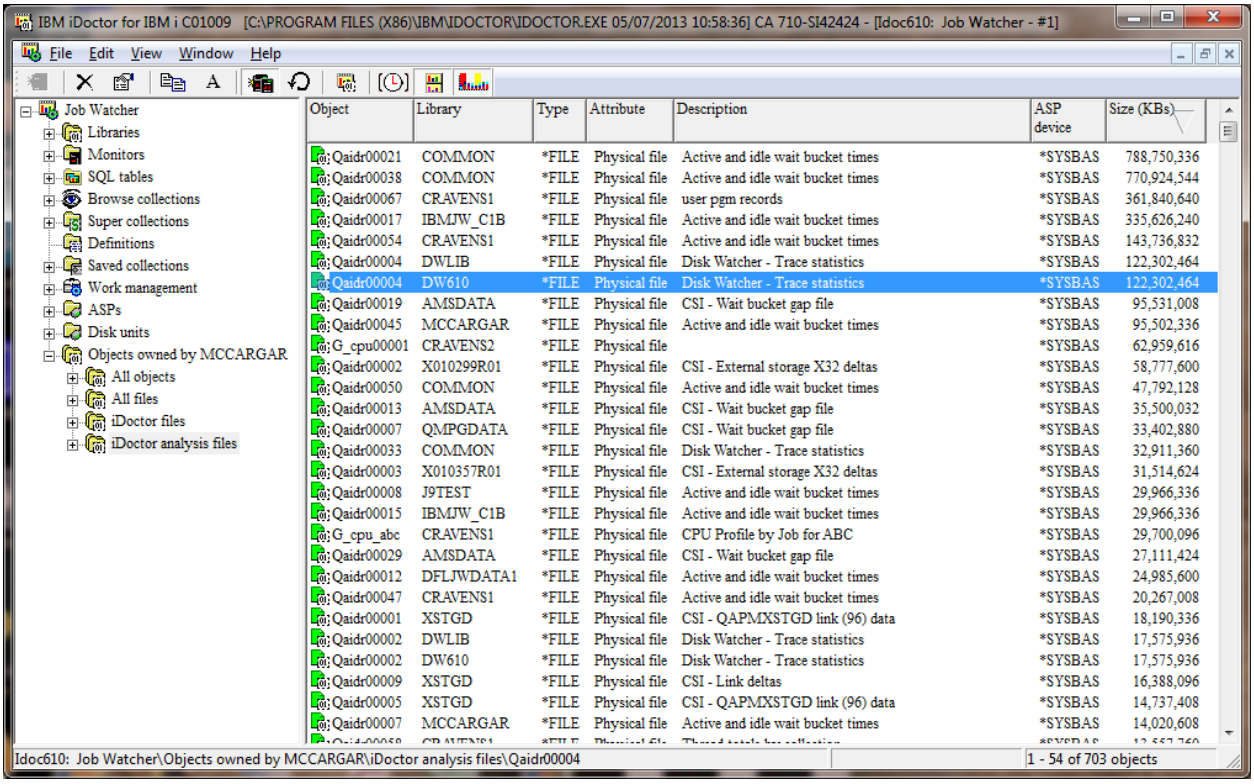
This folder provides the following menu options:

Popup Menu	Description
Filter objects by owner	Displays the Filter Objects by Owner window which lets you configure the user profile to view disk consumption for and whether all the object sizes should be loaded before loading the list. Depending on the number of objects involved, the checkbox to load all object sizes could be extremely slow. 
Clear object sizes cache	Deletes the cache of object sizes previously loaded from the current GUI session. This will cause them to be retrieved again and is only useful if the sizes have changed recently.

5.9.2 Object listings

Each of the folders provided in this interface display lists of objects on the system. In most cases they will take a minute or two (or more depending on the # of objects) to open, especially the 1st time these options are used.

They can be sorted by size and then deleted via the GUI if desired. Use the “load all object sizes” checkbox described above for best results.



6 The Data Viewer

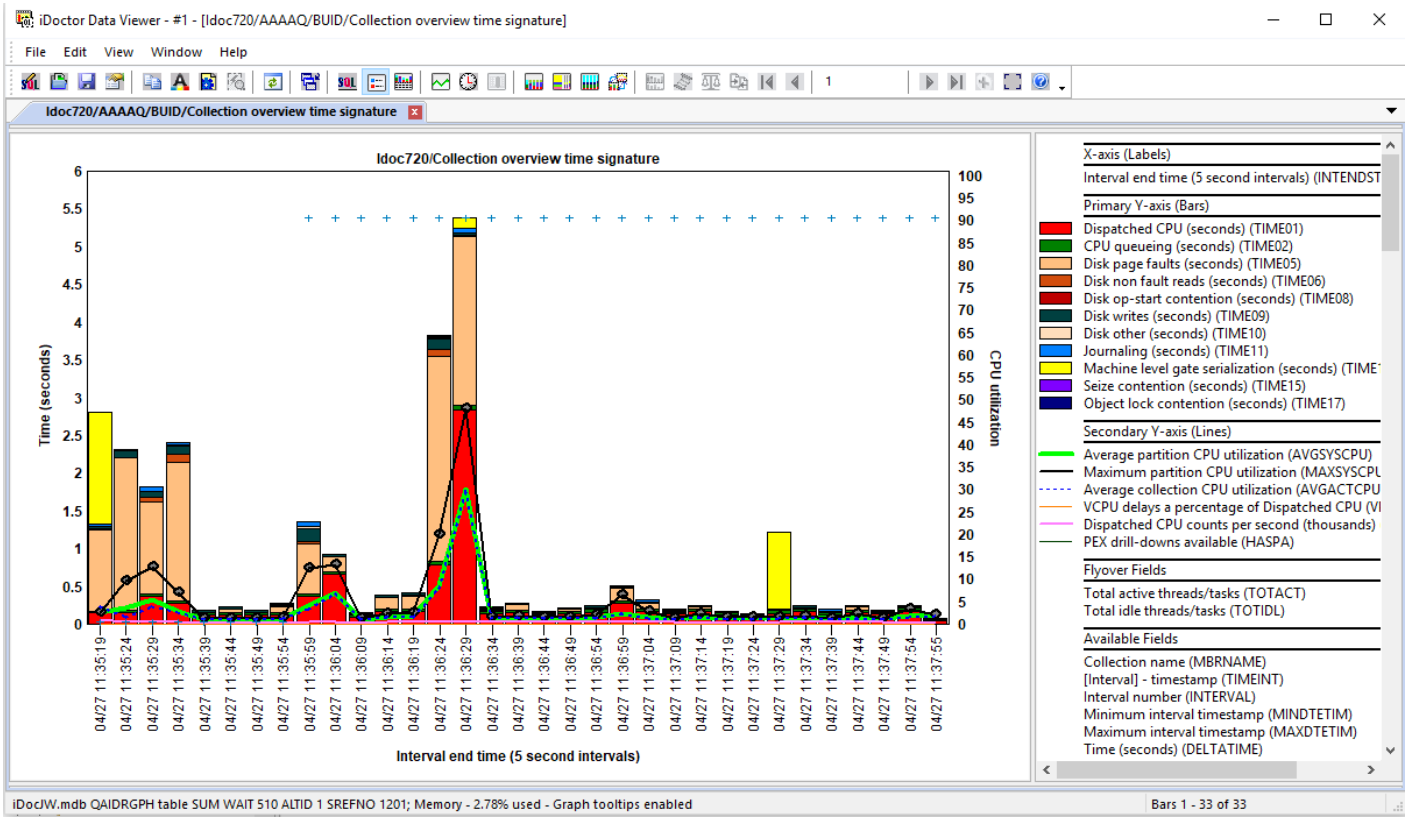
The Data Viewer is a frame window within IBM iDoctor for IBM i used for displaying tables and graphs over data on the system. You can have as many Data Viewer windows open at one time as you want. The data behind the views within a Data Viewer may come from any number of systems desired which allows for easier comparisons.

These views are manipulated in the same way as within the Main Window (they can be tiled, cascaded, etc). Tables and graphs are typically opened from the component views in the Main Window. There are also drill down options to open tables and graphs directly from the Data Viewer window instead of going back to the Main Window.

The data behind a table or graph in the Data Viewer is produced using an SQL query. These SQL statements usually are IBM-defined but the capability to define and open user-defined graphs and queries in the Data Viewer is also supported. The SQL statement behind tables and graphs can either be manipulated in the query definition interface or using the SQL editor found above every table or graph by clicking the SQL button on the toolbar of the Data Viewer.

In addition to running SQL statements to produce table and graph views the Data Viewer also has support for viewing spool files containing job logs.

An example of a Data Viewer, showing a graph from Job Watcher is shown below:



Data Viewer Showing a Job Watcher Graph

6.1 Toolbar

The Data Viewer Toolbar

The following describes the different toolbar icons available in the Data Viewer and their purpose:

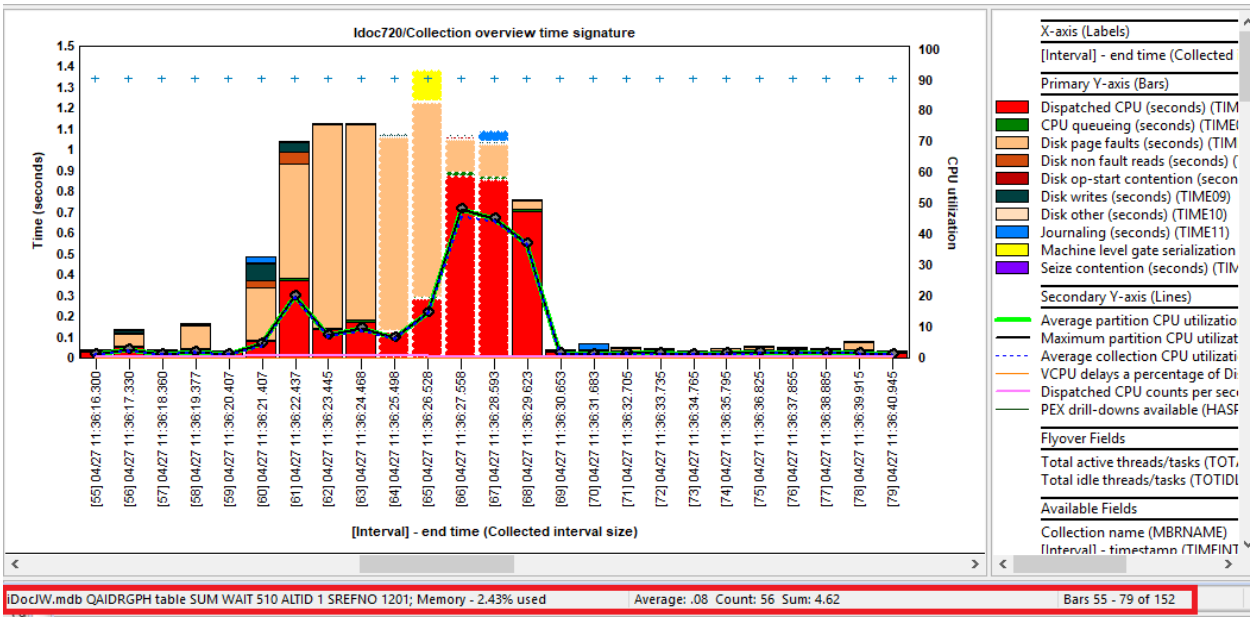
Icon	Description
	Opens a new instance of an SQL Query View. The SQL Query View is used to create a query using Structured Query Language (SQL). The top portion of the view is an area where you can enter an SQL statement (also known as the SQL Editor) and the bottom portion is the result or output from the statement above.
	This option allows you to open any library/file/member on the system using the Open File/SQL Table Window. This window lets you browse for the file or SQL table you wish to open.
	This option allows you to either save the contents of a table view to a file or if viewing a graph to a .jpg image. When using this option on a table the entire contents of the table are saved. You can choose between rich text, comma separated and tab separated text formats. If you wish to include/exclude the header in the saved file, see the Preferences -> File tab.
	Use this option to view the available properties for the active graph or table view.
	Copies the current selection from the current table, graph, or selection made within the SQL Editor to the clipboard. If a table view or SQL editor has current focus, this is only enabled when something has been selected. For graph views this will copy an image of the current graph (without the legend) to the clipboard. If you wish to include the legend in your image, then use Alt-Print Screen to copy the Data Viewer and everything in it to the clipboard. You may want to maximize the current graph view within the Data Viewer before doing this.
	This icon displays the Set Font window allowing you to set the font used for the table views and the legend used in graph views in the application.
	Displays the iDoctor Preferences interface.
	This icon displays the Find Window, allowing you to locate and position to 1 or more records in a table view, matching the desired criteria.
	Refresh the currently active table or graph view.
	This button will display the Window Manager which lets you work with a list of all tabs/views that are opened. This lets you find and activate the desired view/window or close one or more views quickly.
	Shows or hides the SQL Editor containing the SQL statement behind the current graph or table view.
	Shows or hides the graph's legend. This button will display the legend when it is pressed in and not show it otherwise.
	Shows or hides the Synchronized Table View beneath most graphs. This provides the raw data behind the SQL statement and graph currently shown. This table view is synchronized with the graph in all scrolling and selections made.
	Displays a list of available alternate graph views available for the current graph. alternate views (in most cases) allow you to quickly redisplay the graph in another way using the data already retrieved. In some cases, the query may need to be reran to produce the view.
	Modifies the time interval size used for the current time range graph. The SQL statement behind the graph will be reran using a different time range grouping, based on the selection made.
	This option lets the user filter the graph by day of week and/or the time of day. This can be useful if you want to exclude weekends or certain hours of the day. It is only enabled on some types of graphs (CSI Historical Summaries).
	Normalizes the bars in a time range graph. This option can be useful if there are wide variations in the time taken to produce intervals shown on the graph. For example, if Job Watcher was slow to initialize and the 1st 2 intervals took 10 times longer than the rest of the intervals, the rest of the intervals may be barely visible unless this option is turned on. When normalizing a graph, each value on the primary Y axis, is divided by the interval's duration applicable to each Y axis value. This can create a "flattening" effect to bar heights by drawing time values based on relative contributions.

	Toggles the use of variable width bar mode. Intervals that took longer to collect are drawn with wider bars than intervals taking less time. This allows you to see if the collection did not collect intervals at a consistent rate. Typically, this can happen if the system is overburdened and the collection itself cannot be performed optimally.
	Shows (or hides) the idle waits in wait bucket graphs in CSI and JW. This toggle will show all wait buckets including ones that are typically not of interest or revert to the original set. **Note:** You can configure which wait buckets are considered "idle" and which are not in the Wait Bucket Preferences.
	Toggles the graph format for the current graph. Depending on the type of graph you are using this option may be enabled and when pressed will modify the graph to show the same data but in a different way
	This is the button to enable/disable Side-by-Side Comparison Mode. This option is only enabled if 2 or more graphs or tables exist in the current Data Viewer.
	Use this option to synchronize the scrolling of data while in Side-by-Side Comparison Mode. This means when both graphs will scroll together when one of them is scrolled.
	Use this option to synchronize the Primary Y-axis scaling while in Side-by-Side Comparison Mode. This means when both graphs will use the same min/max values on the primary Y-axis.
	This button will take the user to the 1st row set of data in the current report. **Note:** iDoctor defines a block of X rows as a "row set.". By default, this is 1000 rows but is configurable in **Preferences -> SQL -> Data Viewer number of rows per row set**
	This button will move the current report to the previous row set of data. Typically, this means moves the current record position 1000 rows backward.
1	This text field on the toolbar indicates the current scroll position (or row position) within the report being viewed. You can type a new value into this field and the graph or table will move to that position. **Note:** This value is updated automatically after pressing any of the fast forward/reverse buttons around this field
	This button will move the current report to the next row set of data. Typically, this means moves the current record position 1000 rows forward.
	This button will take the user to the last row set of data in the current report.
	Use this option to perform math functions over the selected rows in the current table or table beneath a graph. Clicking this button provides a menu where you can pick which math function to perform (or none). After selecting an option placing the mouse over cells in the selected rows will show the results. The possible math functions are None – No math function is performed, this is the default. Sum – Adds up all values for the current column's selected rows. Average – Average of all values for the current column's selected rows. Min and Max – Displays the min and maximum values from the current column's selected rows. Percent of – Displays the percentage of the current cell's value of the total from the current column's selected rows. Delta – Displays the difference between the rows indicated. The following is an example of using the math function to add up the values in a column for the selected rows. The total is shown in the flyover.

	18 CPX0A04 Other waits 55 14 CPX0A4E MACHINE LEVEL GATE SERIALIZATION 2 14 CPX0A4E MACHINE LEVEL GATE SERIALIZATION 3 18 CPX0A04 Other waits 4 18 CPX0A04 Other waits 5 18 CPX0A04 Other waits 6 18 CPX0A04 Other waits 7 32 CPX0A4A ABNORMAL CONTENTION 8 18 CPX0A04 Other waits 9 18 CPX0A04 Other waits 10
	This option enables full screen mode. This will temporarily remove all iDoctor menus and toolbar options and maximize the Main Window. To return back to your previous state, simply click the Close Full Screen button or press Esc on the keyboard.
	This option displays the properties for iDoctor. This button performs the same action as the Help -> About menu.

6.2 Status Bar

The Data Viewer status bar provides additional details about the current view you are working with in the grey area at the very bottom of the window.



Data Viewer Status Bar (red box around it) and Graph

The status bar has 3 parts; some of which will only be filled based on the current view and status:

1) General Status Pane
2) Math Pane
3) Position Indicator Pane

6.2.1 General Status Pane

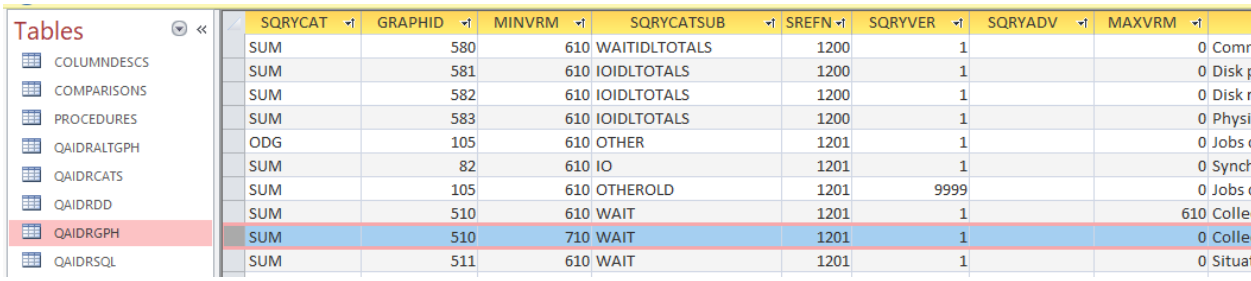
This portion of the status bar identifies different things depending on the current selection.

1) If an iDoctor graph is shown and the mouse is not over any data (over white space), then this information identifies the .mdb database (MS Access) and where in the database the report is located. The available memory (based on GDI objects) is also available here. For example:

iDocJW.mdb QAIDRGPH table SUM WAIT 510 ALTID 1 SREFNO 1201; Memory - 4.72% used

   a. iDocJW.mdb identifies the database as iDocJW.mdb in the iDoctor install directory (typically C:\program files (x86)\ibm\idoctor)

   b. QAIDRGPH table SUM WAIT 510 ALTID 1 SREFNO 1201 means table QAIDRGPH where SQRYCAT = SUM and SQRYCATSUB = WAIT, GRAPHID is 510 and SREFNO = 1201 in QAIDRGPH and in table QAIDRSQL.

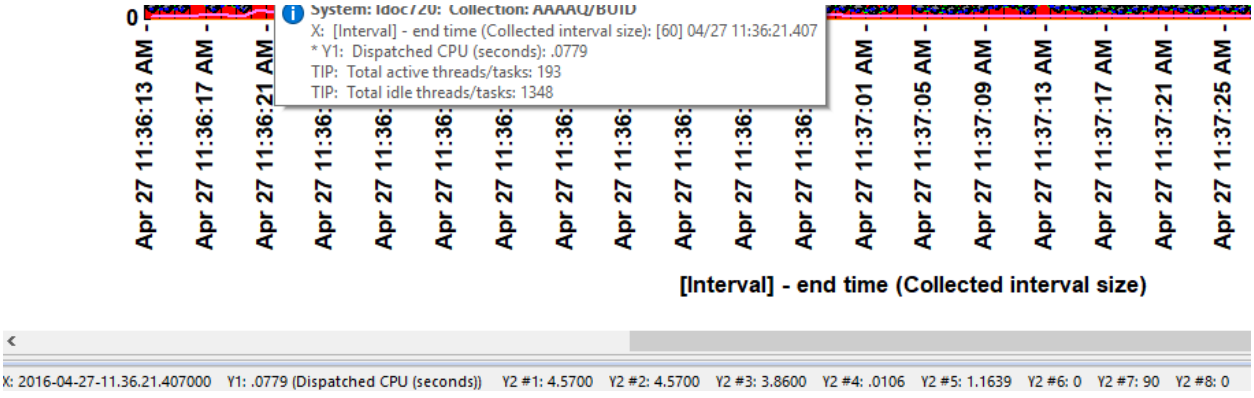


SQRYCAT	GRAPHID	MINVRM	SQRYCATSUB	SREFN	SQRYVER	SQRYADV	MAXVRM	
SUM	580	610	WAITIDLTOTALS	1200	1		0	Comm
SUM	581	610	IOIDLTOTALS	1200	1		0	Disk p
SUM	582	610	IOIDLTOTALS	1200	1		0	Disk r
SUM	583	610	IOIDLTOTALS	1200	1		0	Physi
ODG	105	610	OTHER	1201	1		0	Jobs
SUM	82	610	IO	1201	1		0	Synch
SUM	105	610	OTHEROLD	1201	9999		0	Jobs
SUM	510	610	WAIT	1201	1		610	Colle
SUM	510	710	WAIT	1201	1		0	Colle
SUM	511	610	WAIT	1201	1		0	Situa

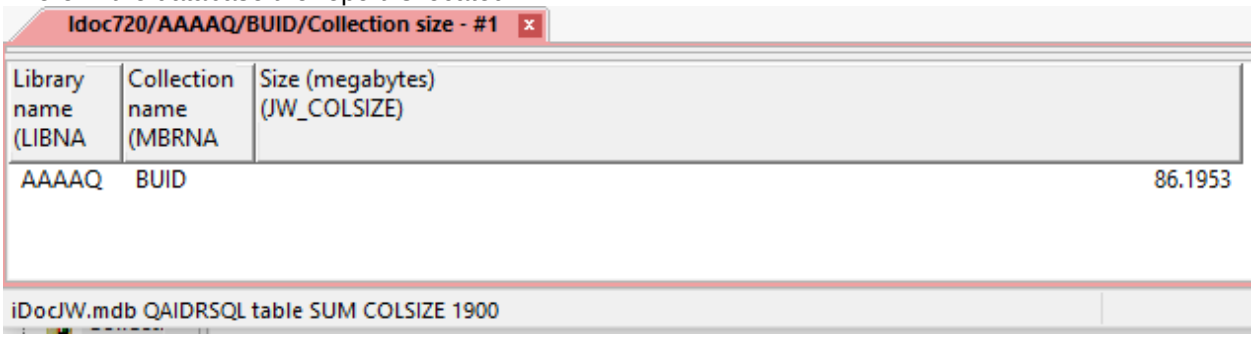
iDocJW.mdb QAIDRGPH table GRAPHID = 510, SQRYCAT = 'SUM' and SQRYCATSUB = 'WAIT'

   c. The memory used is 4.72%. This is based on GDI objects and not based on normal RAM. You can increase this if you have access to change your Windows registry settings, by using the menu Edit -> Increase Windows GDI limit menu from the Main Window. This is a one-time step and increases the maximum GDI limit in all windows application from 10000 to 40000 objects. Because the graphing mechanism consumes large numbers of these, this may be desirable depending on how iDoctor is used.

2) If an iDoctor graph is shown and the mouse is over a selection, the general status bar pane contains the data relative to that point.

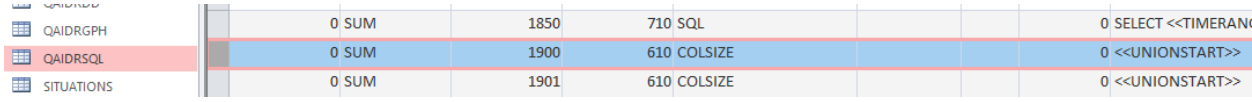


3) If an iDoctor report is shown, then this information identifies the .mdb database (MS Access) and where in the database the report is located.



a. iDocJW.mdb identifies the database as iDocJW.mdb in the iDoctor install directory (typically C:\program files (x86)\ibm\idoctor)

b. QAIDRSQL table SUM COLSIZE 1900 means table QAIDRSQL where SQRYCAT = SUM and SQRYCATSUB = COLSIZE and SREFNO = 1900.

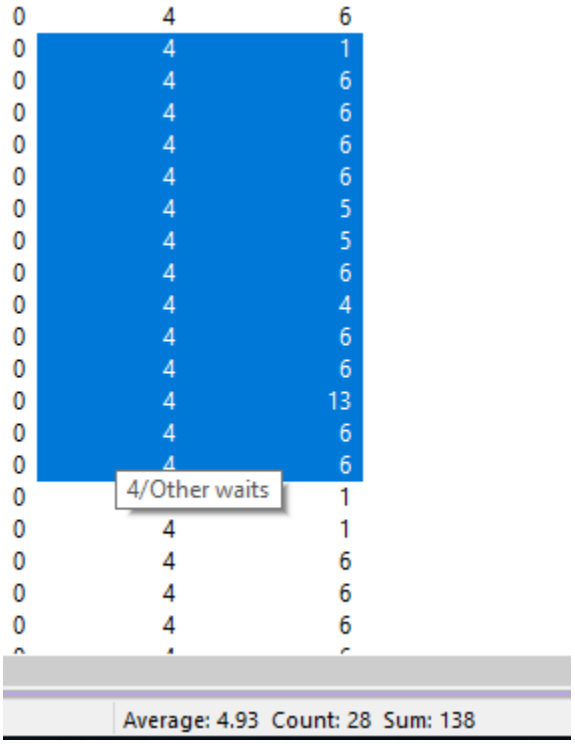


iDocJW.mdb QAIDRSQL table SREFNO = 1900, SQRYCAT = 'SUM' and SQRYCATSUB = 'COLSZE'

6.2.2 Math Pane

The Math Pane will only be filled in some situations where a selection has been made in a table or graph.

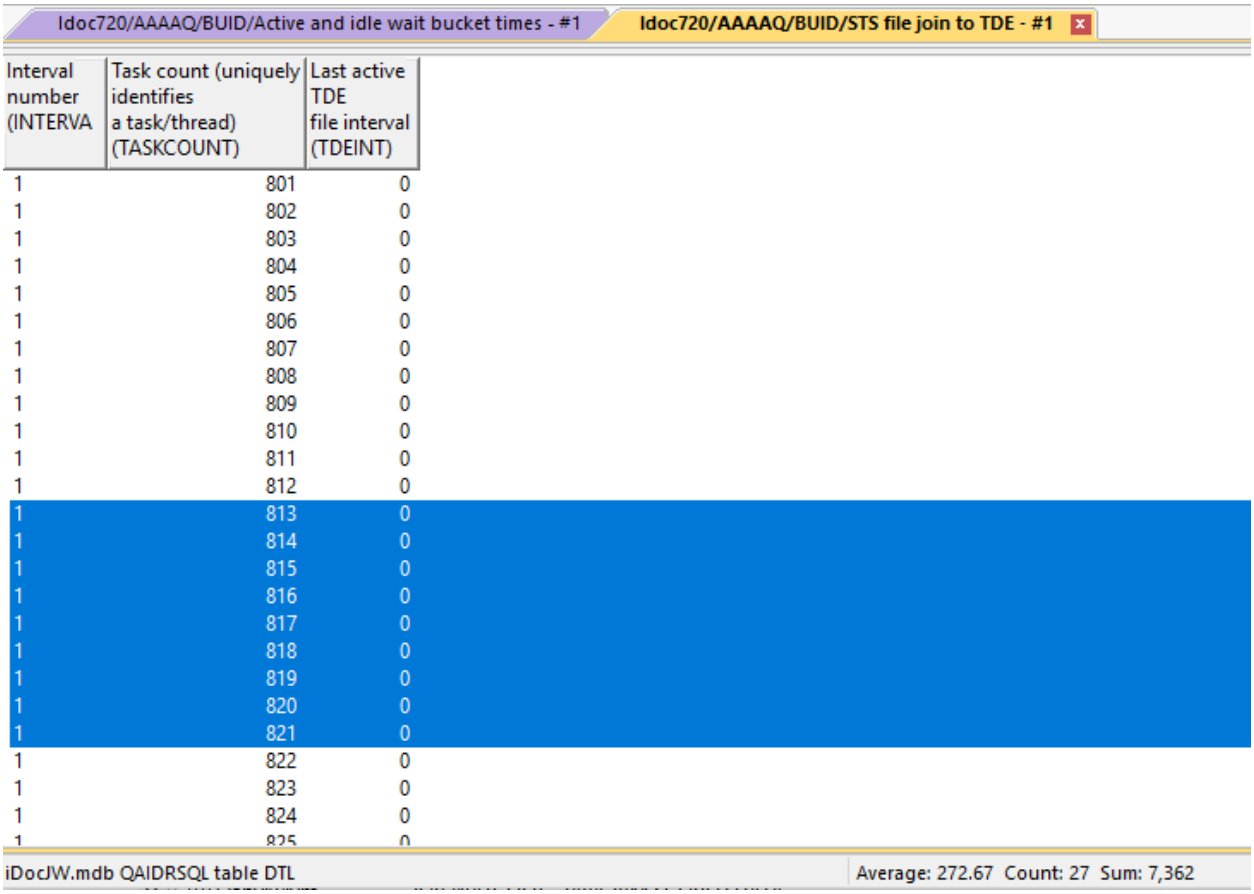
A typical usage scenario would be if the user needs to know the sum of several values in a table. By selecting the desired cells, the average, count and sum of the values selected will be given.



Math Pane Example

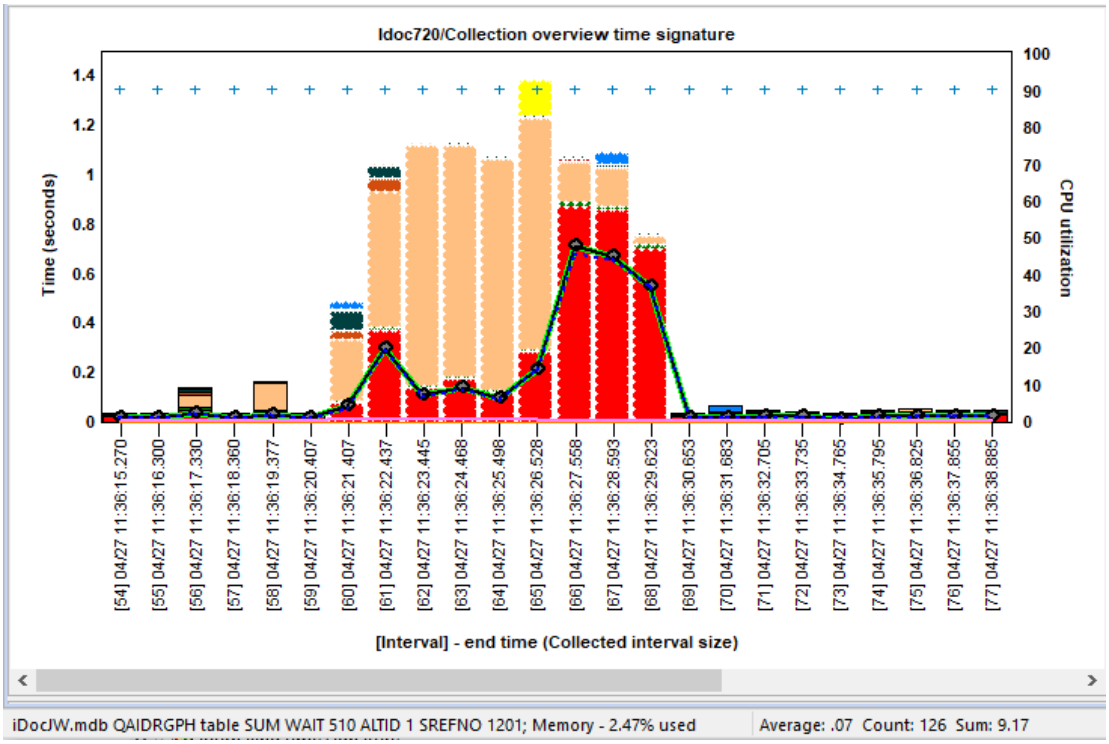
Selecting the entire rows of a table can also be used by the Math Pane and all non-numeric fields will be discarded from the calculations.

Note: A maximum limit of 150 rows and 50 columns applies. Exceeding this will cause no data to be shown in this pane.



Math Pane Row Selection Example

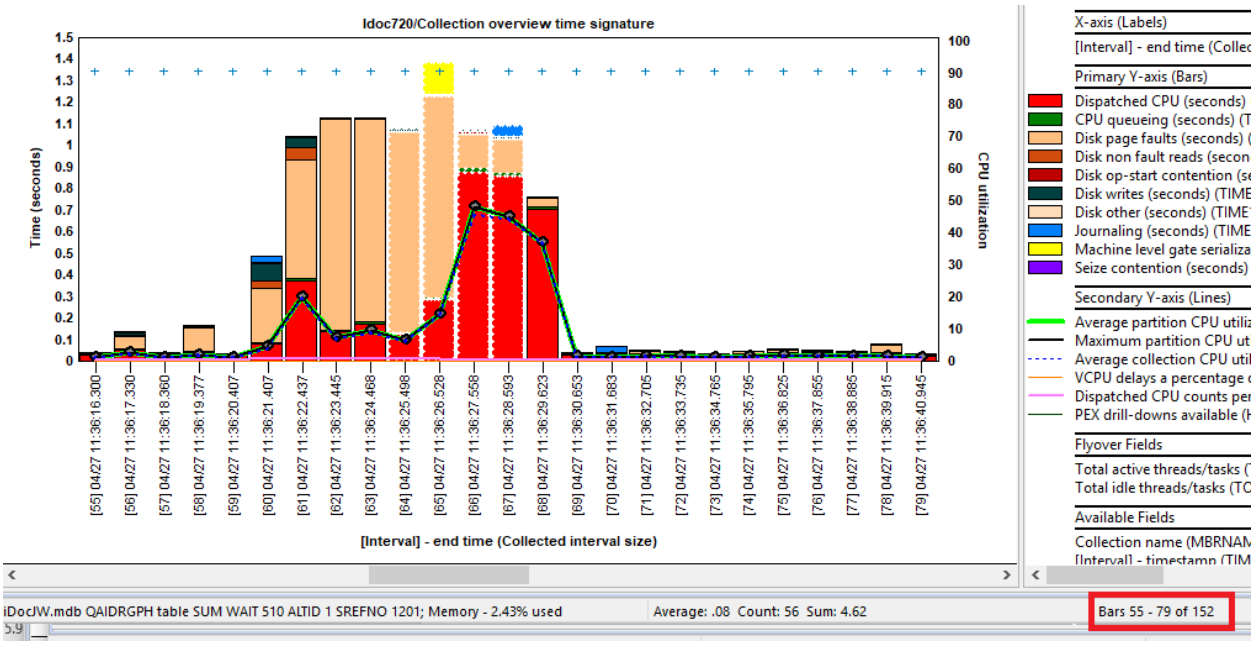
This feature also applies to graphs when making selections. The numbers on the Y1-axis will be added up and shown in the Math Pane.



Graph Math Pane Example

6.2.3 Position Indicator Pane

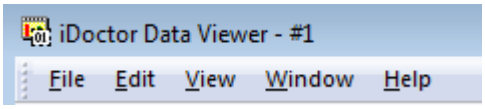
This pane displays for the current view, the total records/bars and the scrolled location within that total.



Position Indicator Pane Example

6.3 Menu Options

This section discusses the menu options available within the iDoctor Data Viewer. This only covers the menus available at the top of the Data Viewer window and does not cover the popup menus available within views displayed inside the Data Viewer.



The Data Viewer Menus

The table below outlines the different types of menu operations that may be performed within the Data Viewer.

File Menu	Description
Open iDoctor Session…	This option allows a user to open a previously saved iDoctor session file (*.idr.) When choosing this option, you will be prompted for the file name to open and if continued all views in the Main Window and Data Viewers will be closed and then the previous session's state will be restored.
Save iDoctor Session As	Use this option to save the current state of all open views, tables and graphs to a session file.
Restore Previous iDoctor Session	This option can be used to open the last previously used iDoctor session.
New SQL Query	Opens a new instance of an SQL Query View. The SQL Query View is used to create a query using Structured Query Language (SQL). The top portion of the view is an area where you can enter an SQL statement and the bottom portion is the result or output from the statement above. You can either edit the statement directly or use the query definition interface to change it. Any changes you make via query definition will be immediately visible in the top portion of this view.
Open File/Member	This option allows you to open any library/file/member on the system using the Open File/SQL Table Window. This window lets you browse for the physical or logical file you wish to open. If you do not specify the member, you will be prompted to select the member from a list if the file is a multiple member file.
Save | View As...	This option allows you to save the contents of a table view to a file. When using this option, the entire contents of the table are saved. You can choose between rich text, comma separated and tab separated text formats.
Save | Selection As...	The option allows you to save the **selected** contents of a table to a file. When using this option only the selected records or block of cells are written to the file. When using this option, you can choose between rich text, comma separated and tab separated text formats. This option is not available for graph views.
Save | Query Definition...	This option allows you to save the current table's query definition to the local user-defined reports database on the PC. Query definitions are saved into the user-defined queries folder under collections and can be reused. A window will be displayed asking for a description of the query and which component it applies to (if this is not already known).
Save | Graph Definition...	Allows the graph definition behind the current graph view to be saved to the local user-defined reports database on the PC. Graph definitions are saved into the user-defined graphs folder under collections and can be reused. A window will be displayed asking for a description of the graph and which component it applies to (if this is not already known).
Close	This menu will close the active view in the Data Viewer.
Print	This menu allows you to print the active graph view or contents of the SQL Editor. Before using this option, set focus into the desired SQL Editor or graph view by clicking inside the view. **Note:** This option is not currently available for table views.
Close Data Viewer	Use the menu to immediately close the current data viewer and all open views and child windows within it. If queries are currently executing when performing this action, you may have to request this option a 2nd time since the queries will be canceled on the 1st attempt to close the Data Viewer.

Edit Menu	Description
Undo	Undo changes made in the SQL Editor with current focus.
Zoom Out	If working in a graph view and you've zoomed in already, this option will allow you to zoom back out 1 level.
Zoom Out All	If working in a graph view and you've zoomed in 1 or more times, this option will restore the graph to its original state.
Cut	Cut the current selection from the SQL Editor to the clipboard.
Copy	Copies the current selection from the current table, or SQL Editor to the clipboard. If a table view or SQL Editor has current focus, this is only enabled when something has been selected. For graph views this will copy an image of the current graph (without the legend) to the clipboard. If you wish to include the legend in your image, then use Alt-Print Screen to copy the Data Viewer and everything in it to the clipboard. You may want to maximize the current graph view within the Data Viewer before doing this.
Paste	Paste the current text selection on the clipboard into the SQL Editor.
Find...	This option opens the find window for the SQL Editor or table view **with current focus**. Click on a table or SQL editor first to give the desired view focus if necessary. When used in an SQL editor it allows you to quickly search for the next occurrence of a value in the SQL statement. This option allows you to reposition the current record position in a table view, based on some input you supply. The Find Dialog will be displayed, and you can use it to search for a specific value.
Find Next	Find the next occurrence of a value within the SQL Editor or table view. This option doesn't apply to graph views.
Find Previous	Find the previous occurrence of a value within the SQL Editor or table view.
Replace	Displays a window allowing you to perform text replacement in the SQL editor.
Set Font	This menu displays a window allowing you to set the font used for the table views in the IBM iDoctor for IBM i application. This option does not apply to the graph views. The font sizes used in the graph views are controlled in the Preferences window.
Preferences…	This menu displays a window allowing you to set user Preferences for the application.
Select All	Selects the entire contents of the SQL Editor or all rows in a table.

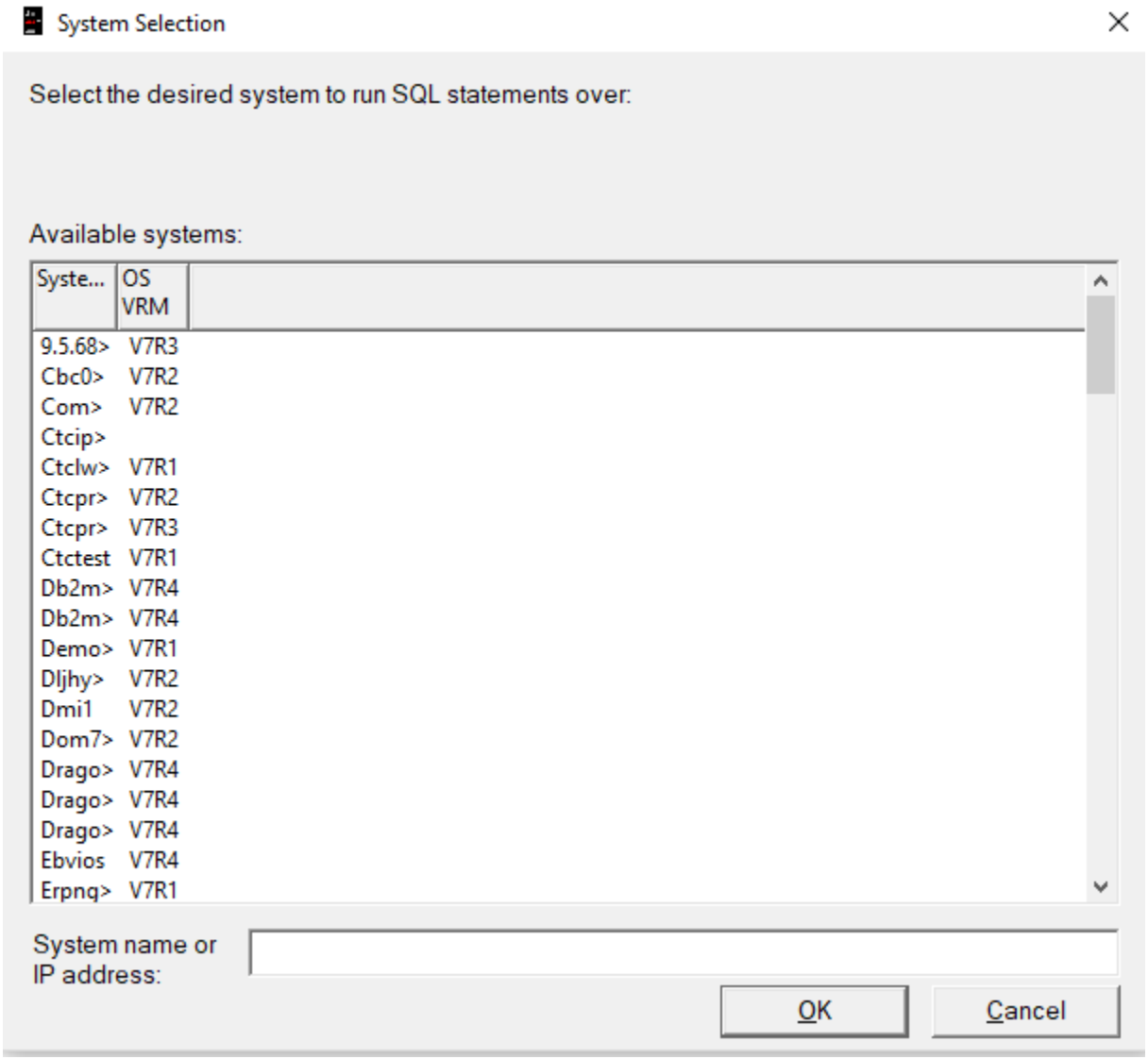
View Menu	Description
Customize	This option is used to customize the menu and toolbar options shown. You can control how they appear, remove/add options or reset them to their iDoctor-shipped defaults as desired.
Full Screen	This menu will display the Data Viewer maximized without a menu bar or toolbar in order to use all available space. You can press ESC or use the Close Window button to get out of this mode. **Note:** This option will be disabled if the MDI tabbed style is None.
MDI Tabbed Style	Use this option to change the current MDI tabbed style being used. There are 3 styles of MDI tabs available in iDoctor: 1) None – this is a classic Windows MDI without tabs 2) Standard – allows users to tile and cascade but you cannot create groups of MDI tabs to compare with other tabs. 3) Grouped – Tabs cannot be tiled or cascaded but you can create groups of MDI tabs in order to make comparisons.
Application Look	Use this option to change the current application look setting used by iDoctor. The available options are various types of color schemes and Windows OS releases.
Record Quick View	This menu will vertically display the currently selected record(s) in the active table view if one is available. This can be very useful to see all the details for a specific record of data without having to scroll as much or if you wish to more easily compare to records in a table.
Status Bar	This menu will either show or hide the status bar. If the status bar is already visible then there will be a checkmark next to the menu.
SQL Message Log	Displays a view containing all SQL statements executed by the iDoctor client and the result (time to execute, or error message). The log is cleared every time the GUI is restarted. The file name for this log specified by the Preferences -> SQL -> Logfile value.
Field Names	This option will toggle the display of column headings so only the SQL given field name is shown. This applies only to table views. **Tip:** A checkbox indicates if this option is enabled or not. MBRNAME | TIMEINT | INTERVAL | INTENDSTR | MINDTETIM BUID | [1] 04/27 11:35:19.745 | 1 | 2016-04-27-11.35.19.745000 | 2016-04-27-11.35.17.779720 BUID | [2] 04/27 11:35:21.805 | 2 | 2016-04-27-11.35.21.805000 | 2016-04-27-11.35.19.745001
Field Descriptions	This option will toggle the display of column headings to long field descriptions (if known). This applies only to table views. Collection name | [Interval] - timestamp | Interval number | Interval end timestamp | Minimum interval timestamp BUID | [1] 04/27 11:35:19.745 | 1 | 2016-04-27-11.35.19.745000 | 2016-04-27-11.35.17.779720 BUID | [2] 04/27 11:35:21.805 | 2 | 2016-04-27-11.35.21.805000 | 2016-04-27-11.35.19.745001
Field Names and Descriptions	This option will toggle the display of column headings to both short field names and descriptions. This applies only to table views. Collection name (MBRNAME) | [Interval] - timestamp (TIMEINT) | Interval number (INTERVAL) | Interval end timestamp (INTENDSTR) | Minimum interval timestamp (MINDTETIM) BUID | [1] 04/27 11:35:19.745 | 1 | 2016-04-27-11.35.19.745000 | 2016-04-27-11.35.17.779720 BUID | [2] 04/27 11:35:21.805 | 2 | 2016-04-27-11.35.21.805000 | 2016-04-27-11.35.19.745001
Refresh	Refresh the current view.

Window Menu	Description
Cascade	Use this menu to rearrange all views in the Data Viewer in an overlapping sequence starting in the upper left corner of the window. **Note:** This option is not shown when the MDI tabbed style is set to Grouped.
Tile Horizontally	Use this menu to rearrange all views in the Data Viewer such that each view will have an equal distribution of the available height in the Data Viewer. The views will not overlap each other. **Note:** This option is not shown when the MDI tabbed style is set to Grouped.
Tile Vertically	Use this menu to rearrange all views in the Data Viewer such that each view will have an equal distribution of the available width in the Data Viewer. The views will not overlap each other. **Note:** This option is not shown when the MDI tabbed style is set to Grouped.
Close All	This option can be used to close all open tabs/views.

Help Menu	Description
IBM iDoctor for IBM i website	Launches your web browser and takes you to the iDoctor website.
IBM iDoctor for IBM i downloads	Launches your web browser and takes you to the iDoctor downloads page.
IBM iDoctor for IBM i documentation	Launches your web browser and takes you to the iDoctor documentation page.
About	This will display version information for the IBM iDoctor for IBM i client.

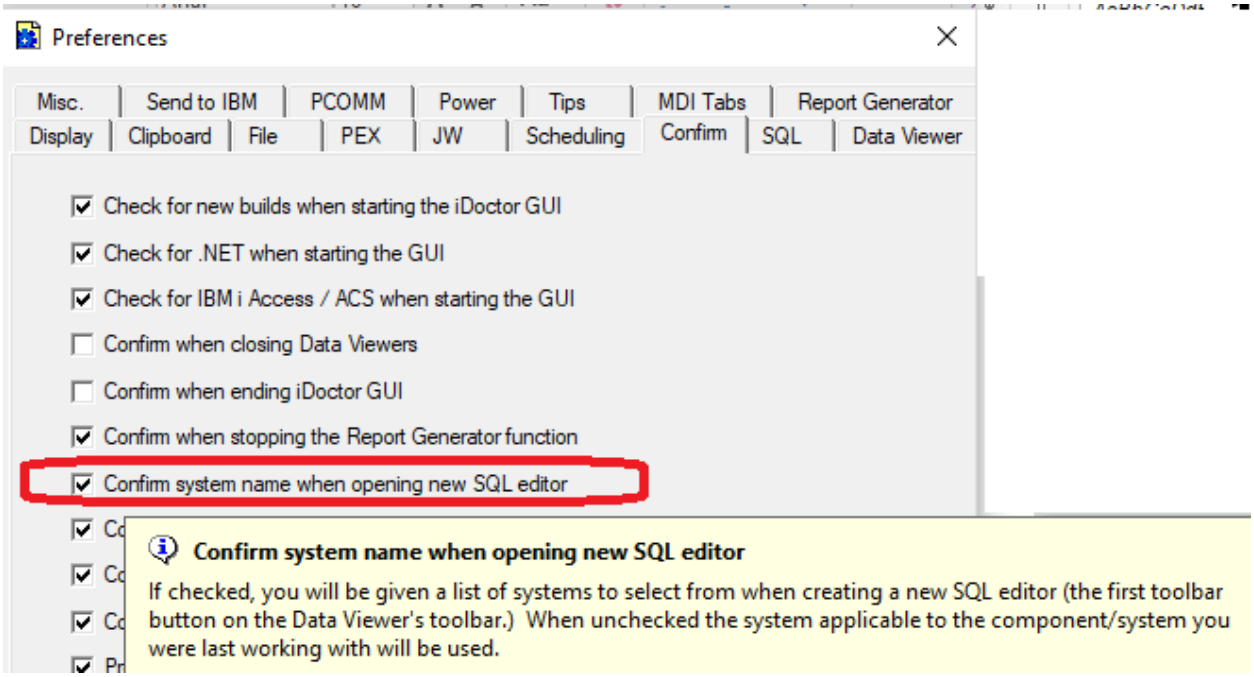
6.4 SQL Query View

The SQL Query View refers to the view you get when you press the SQL Query View button  on the Data Viewer toolbar. Pressing this button will initially provide you with an option to select the system name you wish to write an SQL statement against like the following:



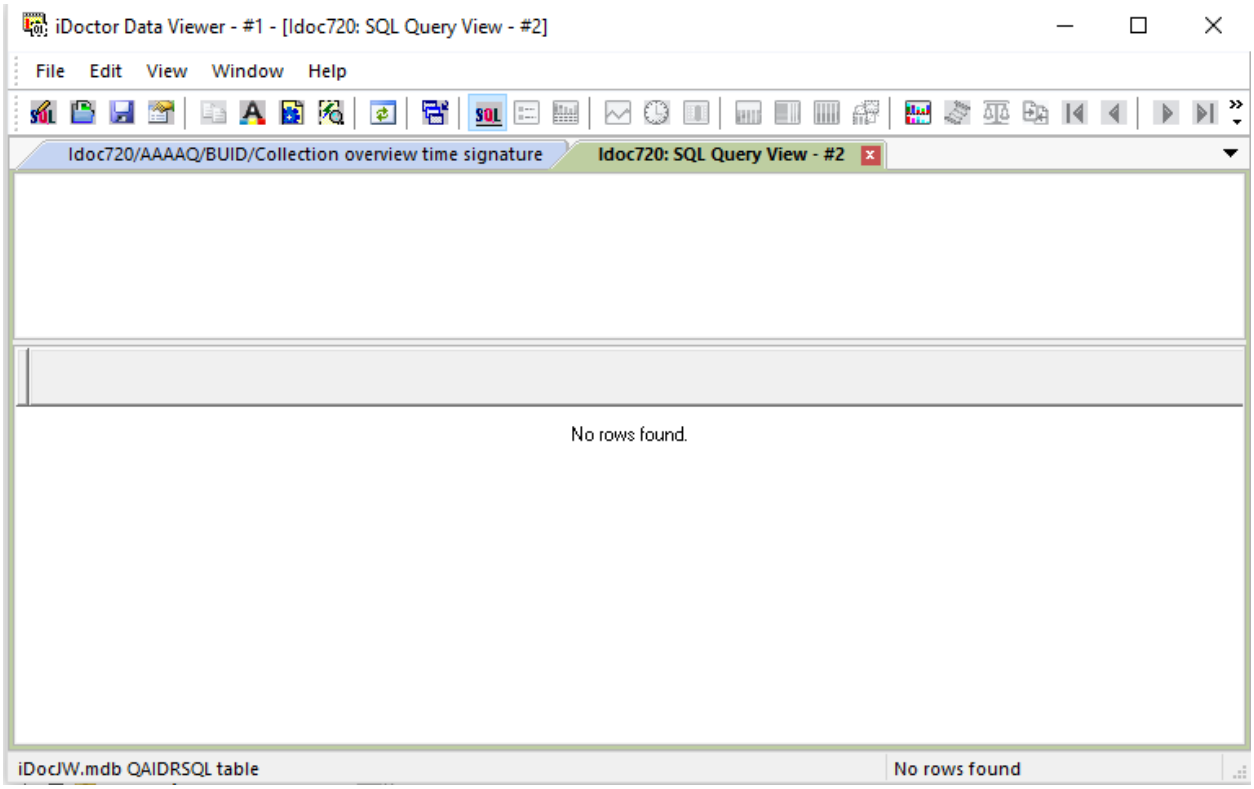
System Selection Window

Tip: If you do not want to see the System Selection Window and just have iDoctor use the current system you are working with, the uncheck the Confirm system name when opening new SQL editor option in Preferences-> Confirm.



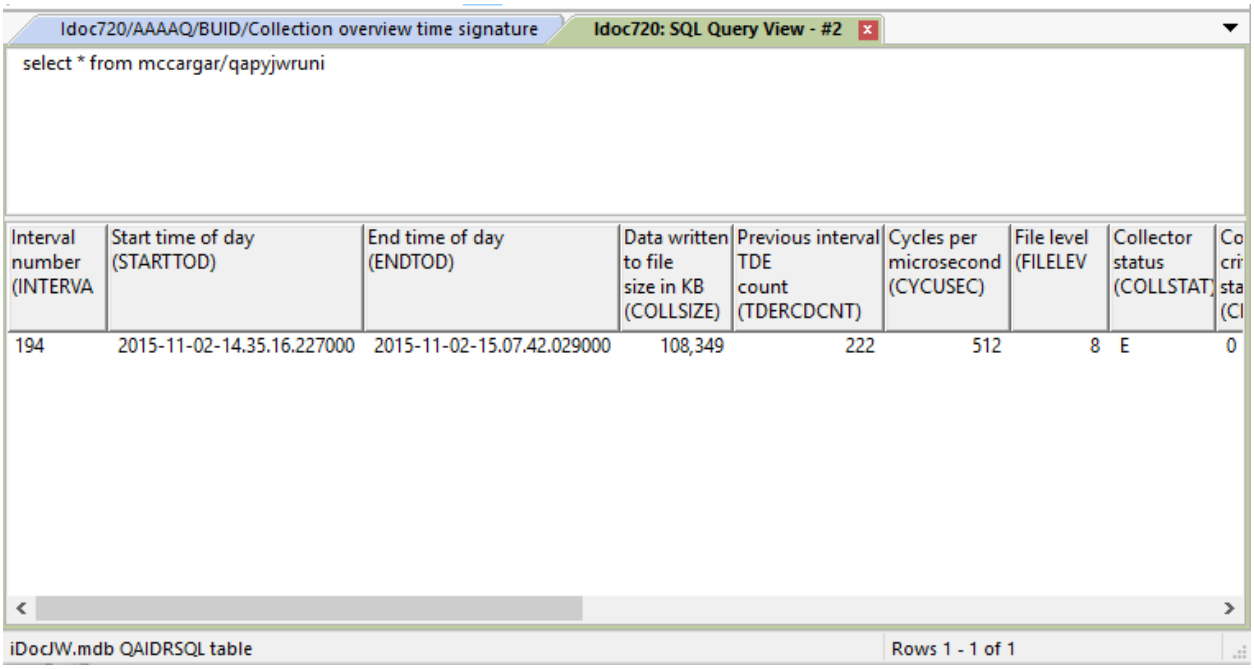
Preferences -> Confirm

After selecting a system, you will be presented with an empty SQL query view where you can write any SQL statements you wish against the system name provided.



SQL Query View

At the top of this view is an SQL Editor and the bottom the results of the execution of that statement will be displayed. If a SQL select statement or WITH statement are used the results are shown as a table. See the next section for more information about the SQL Editor in iDoctor



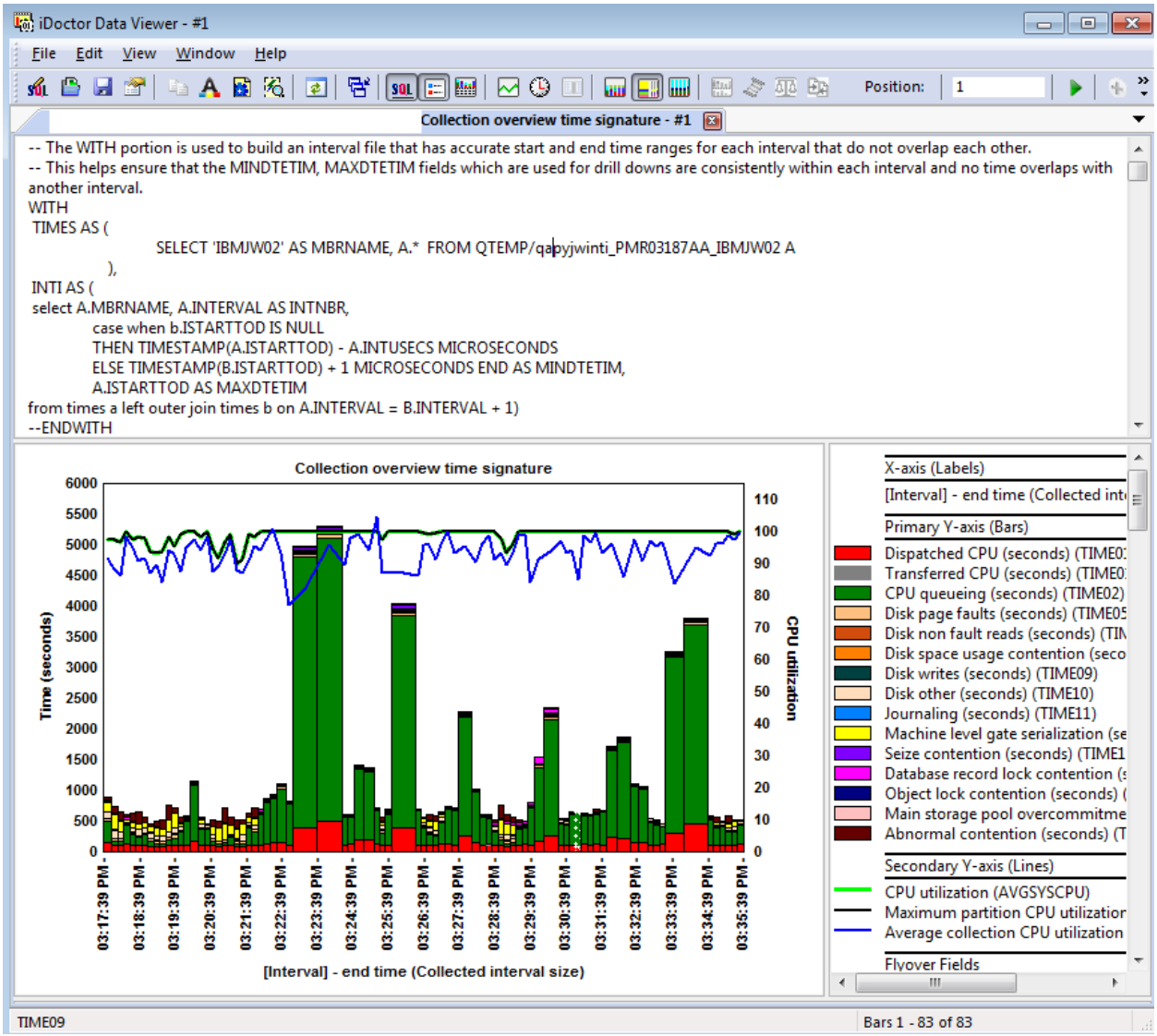
SQL Query View with data

6.5 SQL Editor

The SQL editor lets you dynamically execute and display the results of one or more SQL statement. If multiple SQL statements are to be ran, then each should end with a semicolon.

The top portion of the view is an area where you can enter an SQL statement and the bottom portion is the result from the SELECT or WITH statement which should be the last statement in the SQL editor. This bottom view is either a table view or graph view.

Note: All iDoctor SQL statements (and RUNSQL on green screen) by default use System naming convention and not SQL naming convention where “library/file” syntax is used and not “library.file”. Other tools such as Run SQL Scripts uses SQL naming convention. Use the convert menu options if necessary to copy and paste SQL statements between iDoctor and Run SQL Scripts.



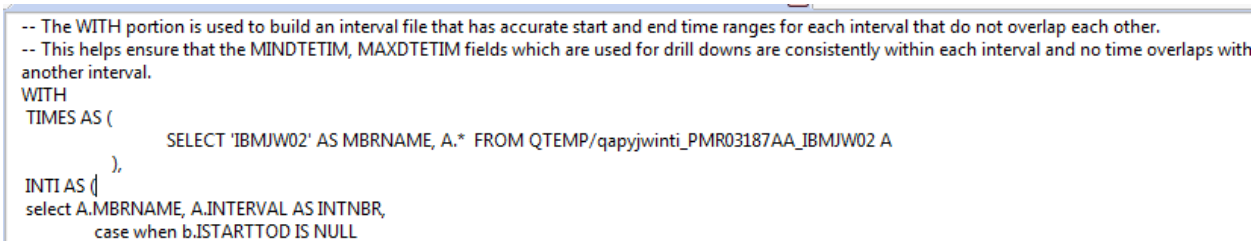
SQL Editor with attached graph

The queries you create with this view may be saved and restored for later use and their definitions can be viewed and manipulated using the query definition interface.
Note: The query definition interface may not work properly if the SQL Editor contains complex SQL statements or embedded comments.

To execute your SQL statement, right-click on the SQL Editor and choose the Execute menu or press the F4 key. This will cause the SQL statement(s) to be executed and the table or graph will be redisplayed.

The SQL Editor is used to execute one or more SQL statements in the order that they appear. Each SQL statement must end with a semicolon. If you want the results to be viewable under the editor, the last statement must be an SQL WITH or SQL SELECT statement. Typically statements before the SELECT or WITH could be calls to SQL stored procedures or commands to create aliases or drop tables, etc.

You can also add any comments to the SQL Editor using two dashes like in this example:

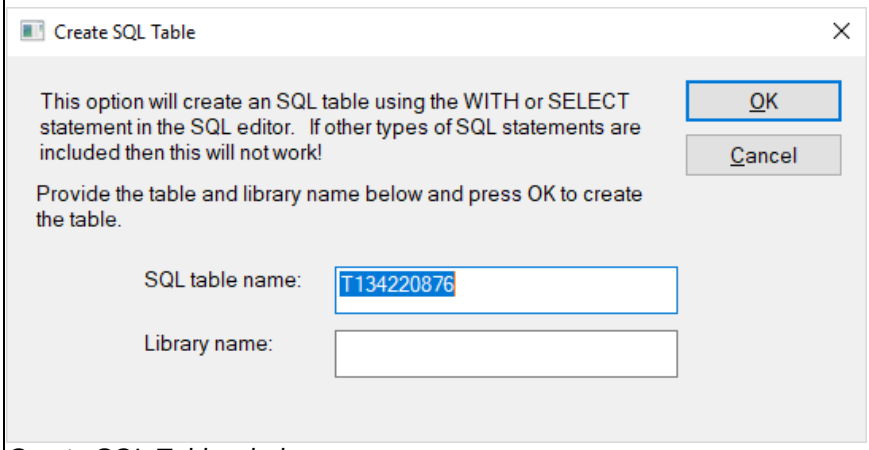


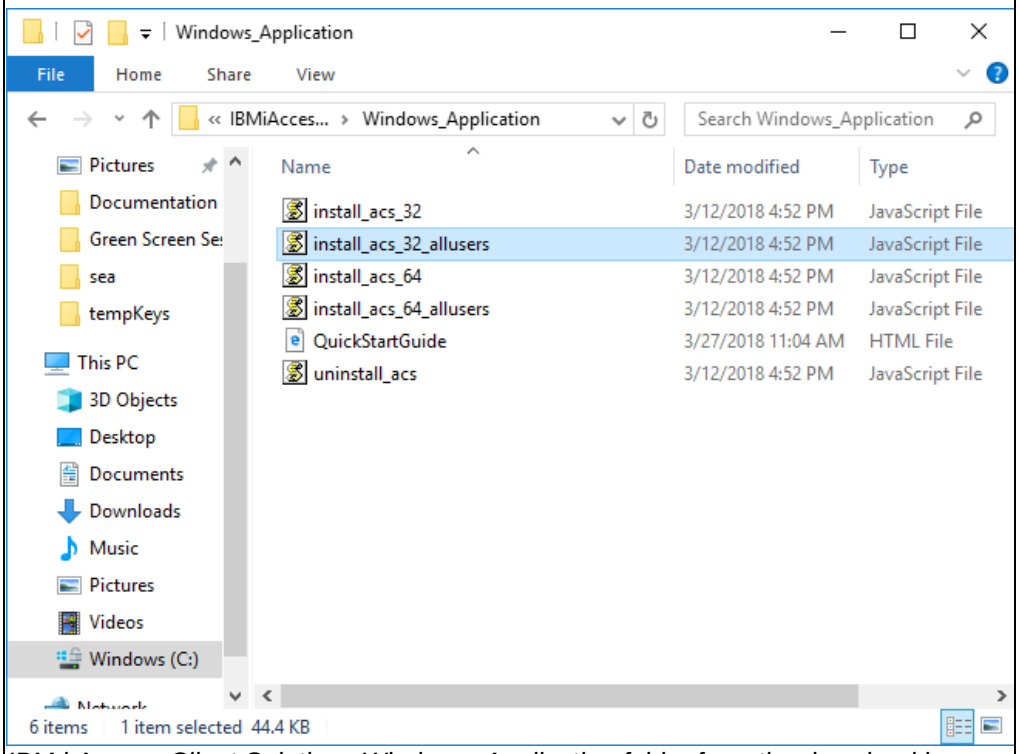
SQL Editor showing comments in 1st line

Most table and graph views in iDoctor's Data Viewer provide an SQL Editor in a hidden view above them. Some comparison modes do not provide this. You can show or hide the SQL Editor using the SQL button on the toolbar. You can also open a new SQL Editor with an attached table view called the SQL Query View, using the 1st icon on the toolbar of the Data Viewer.

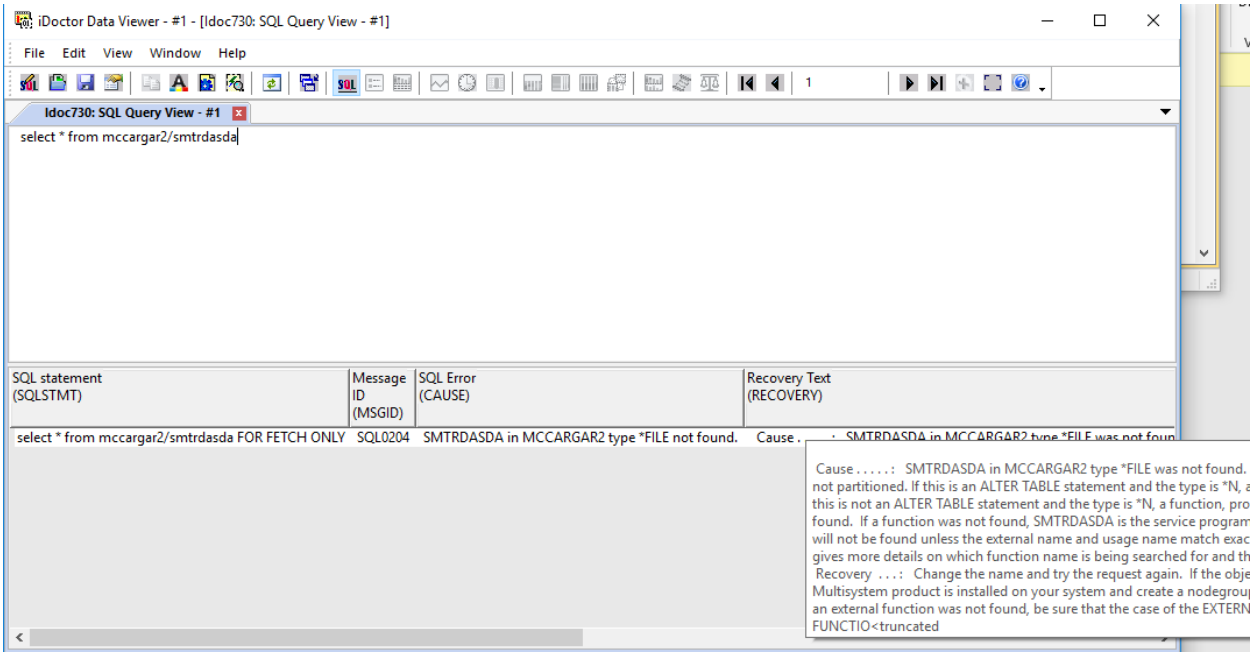
Note: The SQL Editor can be used to run any SQL statements desired within the current QZDASOINIT job. This allows advanced users to define their own stored procedures, drop tables, and create tables or indexes using the SQL editor in iDoctor. For a history of the SQL statements issued and results, use the SQL message log (View -> SQL Message Log menu.) But to enable this logging you will need to use Preferences -> SQL tab and configure this there.

The following options are available in the SQL Editor's popup menu:

Popup Menu	Description
Execute	Execute the SQL statement(s) within the SQL Editor. If multiple statements are used, then all statements are executed from top to bottom and each must end with a semicolon. If necessary, comment out any statements you do not wish to execute.
Create SQL table	This option will create an SQL table on the IBM i based on the SELECT or WITH statement in the SQL editor. This can be useful if you wish to save the data provided by iDoctor to your own file for later viewing. By using this option, you will be prompted for the library and table name where you wish the data to go:  *Create SQL Table window*
Change collection or DB members	This option allows the user to pick a different collection or DB member to work with based on the type of data you are currently using. iDoctor will check the tables found in your SQL statement and attempt to determine which type of performance data is being used. If the files are for Job Watcher, PEX Analyzer, Disk Watcher or Collection Services Investigator then you will be presented with the Choose Collection window and be asked which collection in which library you wish to run the SQL statement over. This allows you to change the collection for the current graph/report you are working with. If unable to determine the type of data, then the Choose Database Members window is shown instead that lists each table in the SQL statement containing multiple members and you will need to pick the members to use for each one (or all of them.)
Display job log	Displays the job log for the QZDASOINIT job that is responsible for running the SQL statements shown in the current table or graph.
Convert SQL naming to System naming	This option can be used to modify the SQL statement so that all tables using SQL naming convention (library.table) will change to System naming convention instead (library/table).
Convert System naming to SQL naming	This option can be used to modify the SQL statement so that all tables using System naming convention (library/table) will change to SQL naming convention instead (library.table).
Launch SQL in Run SQL Scripts	This option will copy the contents of the SQL Editor into a temporary file and open it in the Run SQL Scripts tool provided in either: 1) IBM i Access Client Solution - Run SQL Scripts window. 2) System i Access for Windows – IBM i Navigator – Run SQL Scripts **Tip:** If this option does not work and you have IBM i Access Client Solutions installed, then you have not yet installed 32-bit ACS on your Windows machine using the javascript provided in the install download. After unzipping the files, you will need

	to expand the Windows_Application folder and run **install_acs_32_allusers.js**. **Note:** The 32-bit option is needed because iDoctor is a 32-bit application and it will be launching ACS Run SQL Scripts.  *IBM i Access Client Solutions Windows_Application folder from the download image* From the Run SQL Scripts interface you can use the Visual Explain interface for performance tuning purposes. **Note: Run SQL Scripts uses SQL naming convention by default and iDoctor uses System naming convention. Use the Convert System naming to SQL naming convention option before using this menu option.**
Cut	Cut the current selection from the SQL Editor to the clipboard.
Copy	Copies the current selection from the selection made within the SQL Editor to the clipboard.
Paste	Paste the current text selection on the clipboard into the SQL Editor at the current position.
Find	This option opens the find window for the SQL Editor allowing you to quickly search for the next occurrence of a value in the SQL statement.
Find Next	Find the next occurrence of a value within the SQL Editor.
Replace	Displays a window allowing you to perform text replacement in the SQL Editor.
Select All	Selects the entire contents of the SQL Editor.

If executing the SQL statement within the SQL Editor results in an error, the SQL Editor will show the error instead of the expected graph or table view results. Correct the error and rerun the statement to display the graph or table attached to the SQL Editor.



Example of an error shown in the SQL Editor

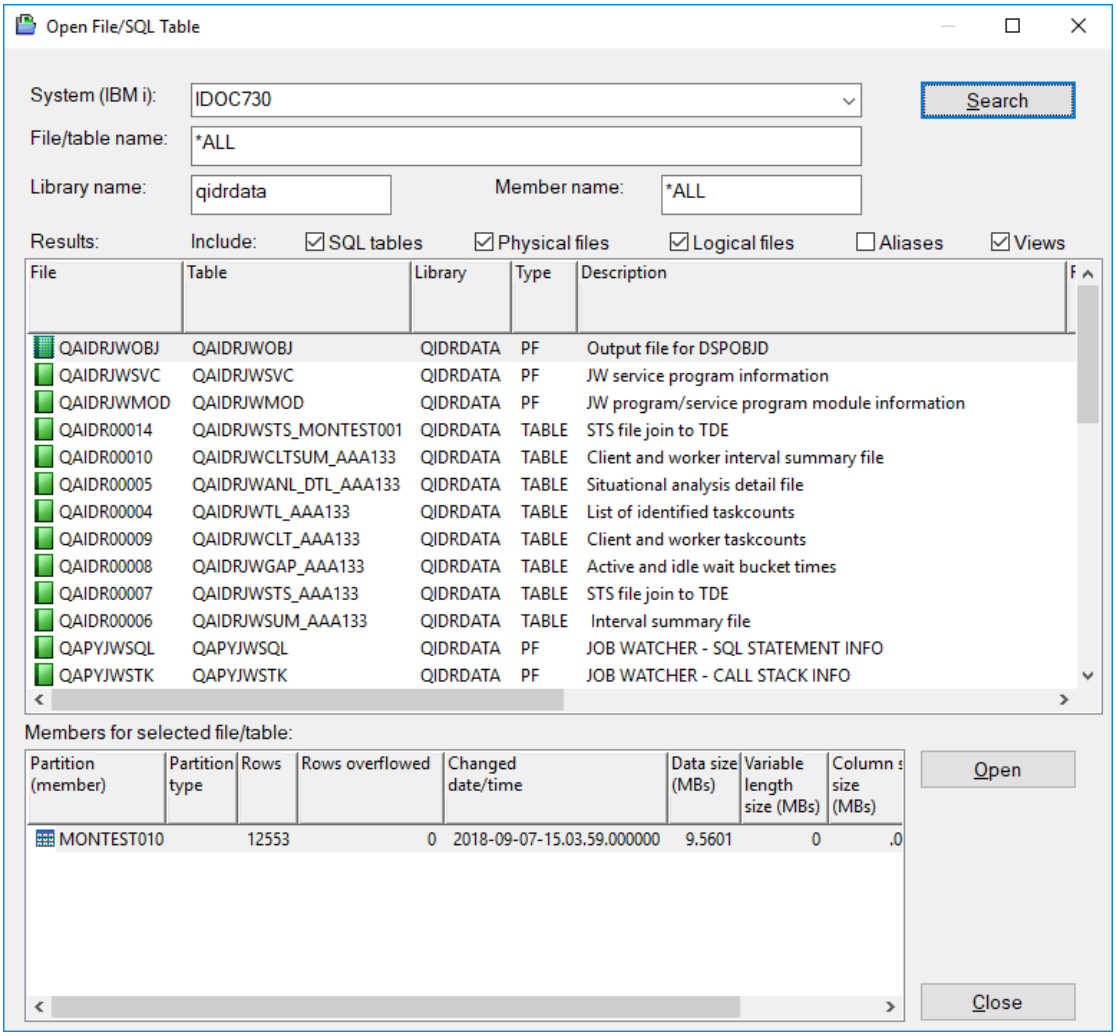
6.6 Open File/SQL Table Window

This option allows the user to open any library/file/member on the system. A window is displayed where the user can browse for the desired type of file and member to open.

The following 5 types of objects can be browsed and opened with this window:

1) SQL tables
2) Physical files
3) Logical files
4) Aliases
5) Views

Of these 5 types only physical files and logical files will potentially fill the list with members to choose from. SQL tables, aliases and views can be opened by double-clicking them without needing to select a member.



Open File/SQL Table Window

The following table describes the interface elements within this window.

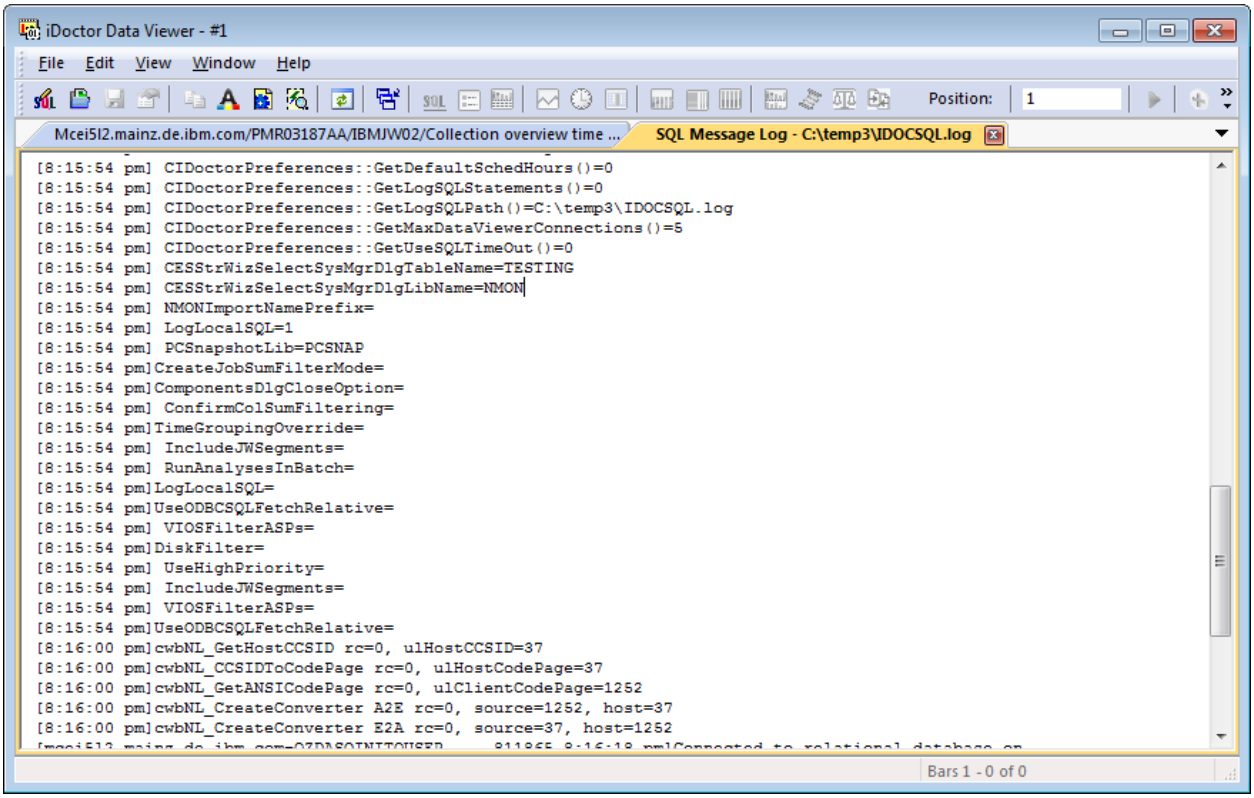
Option	Description
System (IBM i)	The IBM i system to connect to and look for files to open. **Note:** This drop-down list of system names is the same as those found in the IBM i Connections View. Add a system there for it to appear automatically in this list.
File/table name	The name of the physical file, SQL table, alias, view or logical file to search for. This can be *ALL or blank to include all file names or it can be a generic file name like QAYPE*.
Library name	The name of the library name to search for tables. This can be *ALL or blank to include all libraries or it can be a generic library name like MC*.
Member name	The member name to search for within the Members for selected file/table list.
Include options	This list of checkboxes indicates which types of objects should be included in the list of results.
Results	This is the list of objects (files, aliases, views, etc) that match the filtering options above after pressing the Search button.
Search button	The search button will query the IBM i for the objects of interest based on the filters provided.
Members for selected file/table list	This is the list of members for the selected file in the results list (file list) and that also match the member name filter. **Note:** This list is not applicable if the currently selected type of object in the results list is VIEW or ALIAS. This list will always be empty for those types.
Open button	Opens the selected library/file/member or SQL table, alias or view in the Data Viewer.

6.7 SQL Message Log View

This view displays a history of all SQL statements executed by iDoctor in the current session. After each statement listed in the log is either a confirmation that it was successful including the time it took for the system to execute the query or the SQL error resulting from the attempt.

This logging must be turned on via the Preferences -> SQL interface. From Preferences you can configure the name of the file that will contain the log. If SQL logging is turned on, the log file is cleared every time the GUI is restarted. If multiple instances of iDoctor are running on the PC at the same time, this option won't work properly.

You can open the SQL Message Log by using the View -> SQL Message Log menu in the Data Viewer.



Example of the SQL Message Log View in the Data Viewer

Note: As mentioned in the Preferences -> SQL interface, turning on SQL message logging is primarily for IBM service use only and it should be not be left on in most cases. By doing so there is a higher likelihood you will crash the GUI especially if multiple queries are running simultaneously, so avoid using this option unless you need it to help debug a problem.

6.8 Table Views

A table view shows data from database files via SQL statements executed on the system. The user can display many records in a table view and use the scroll bar to quickly move to the records desired via relative positioning.

The SQL behind the table view can be modified at any time using the SQL Editor. The Query Definition interface is also available for SQL statements that are not overly complex. Filters may also be added by right-clicking the column headers and using the Add Filter menu.

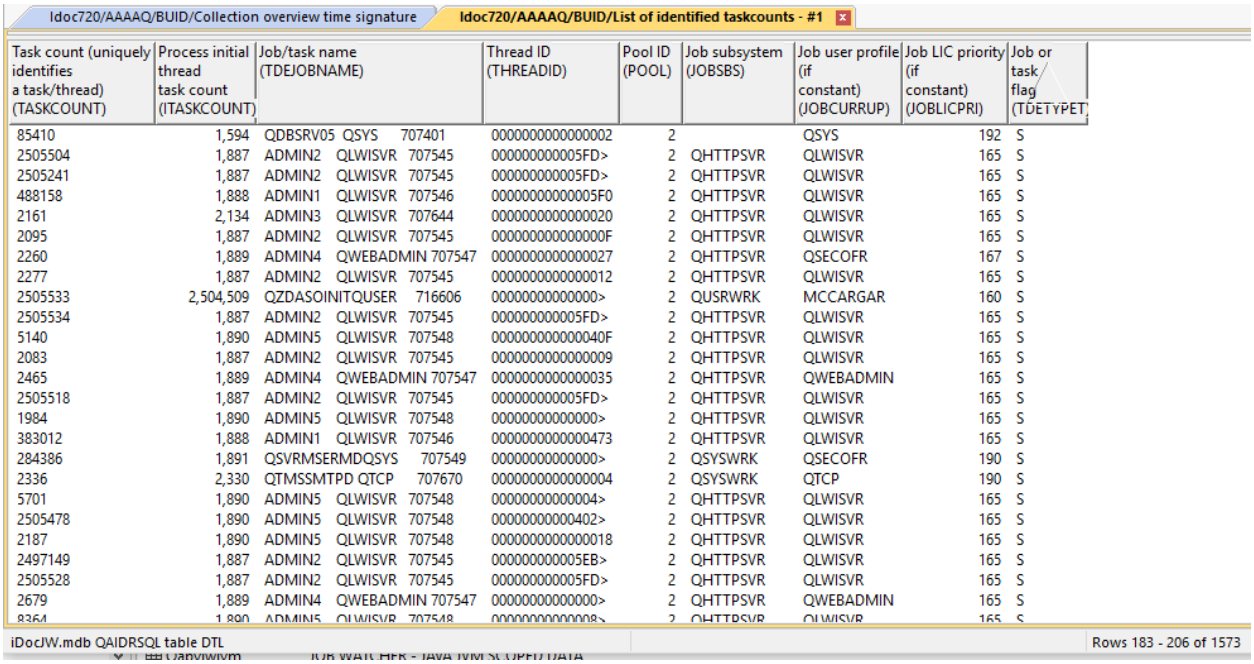
The data may also be sorted by clicking the desired column to sort by. Left click will sort a column in ascending sequence. Right click will sort a column in descending sequence. If you hold the down the shift key the next column clicked will be added to the existing sort sequence. You can also change the sort order by modifying the ORDER BY clause of the SELECT statement in the SQL Editor. Another option to modify the sort is using the Sort By tab of the Query Definition interface.

Data in a table view may be selected for copy and paste to a file or to the clipboard. A set of records -or- a block of cells may be selected at any one time. Click the left mouse button and drag across the cells desired to make a block selection. Once a selection is made, use the Edit | Copy to copy the current selection to the clipboard. Use the File | Save Selection As... menu to write the selection to a file.

Tip: Making cell/row selections will also fill the Math Pane in the status bar with the sum of all numeric values in the selection.

The record Position Indicator Pane in the status bar will show which records are currently being viewed out of the total possible in the active view.

An example of a table view is the following:



Task count (uniquely identifies a task/thread) (TASKCOUNT)	Process initial thread task count (ITASKCOUNT)	Job/task name (TDEJOBNAME)	Thread ID (THREADID)	Pool ID (POOL)	Job subsystem (JOBSBS)	Job user profile (if constant) (JOBCURRUP)	Job LIC priority (if constant) (JOBLICPRI)	Job or task flag (TDETYPET
85410	1,594	QDBSRV05 QSYS 707401	0000000000000002	2		QSYS	192	S
2505504	1,887	ADMIN2 QLWISVR 707545	000000000005FD>	2	QHTTPSVR	QLWISVR	165	S
2505241	1,887	ADMIN2 QLWISVR 707545	000000000005FD>	2	QHTTPSVR	QLWISVR	165	S
488158	1,888	ADMIN1 QLWISVR 707546	00000000000005F0	2	QHTTPSVR	QLWISVR	165	S
2161	2,134	ADMIN3 QLWISVR 707644	0000000000000020	2	QHTTPSVR	QLWISVR	165	S
2095	1,887	ADMIN2 QLWISVR 707545	000000000000000F	2	QHTTPSVR	QLWISVR	165	S
2260	1,889	ADMIN4 QWEBADMIN 707547	0000000000000027	2	QHTTPSVR	QSECOFR	167	S
2277	1,887	ADMIN2 QLWISVR 707545	0000000000000012	2	QHTTPSVR	QLWISVR	165	S
2505533	2,504,509	QZDASOINITQUSER 716606	0000000000000>	2	QUSRWRK	MCCARGAR	160	S
2505534	1,887	ADMIN2 QLWISVR 707545	000000000005FD>	2	QHTTPSVR	QLWISVR	165	S
5140	1,890	ADMIN5 QLWISVR 707548	000000000000040F	2	QHTTPSVR	QLWISVR	165	S
2083	1,887	ADMIN2 QLWISVR 707545	0000000000000009	2	QHTTPSVR	QLWISVR	165	S
2465	1,889	ADMIN4 QWEBADMIN 707547	0000000000000035	2	QHTTPSVR	QWEBADMIN	165	S
2505518	1,887	ADMIN2 QLWISVR 707545	000000000005FD>	2	QHTTPSVR	QLWISVR	165	S
1984	1,890	ADMIN5 QLWISVR 707548	0000000000000>	2	QHTTPSVR	QLWISVR	165	S
383012	1,888	ADMIN1 QLWISVR 707546	0000000000000473	2	QHTTPSVR	QLWISVR	165	S
284386	1,891	QSVRMSERMDQSYS 707549	0000000000000>	2	QSYSWRK	QSECOFR	190	S
2336	2,330	QTMSSMTPD QTCP 707670	0000000000000004	2	QSYSWRK	QTCP	190	S
5701	1,890	ADMIN5 QLWISVR 707548	00000000000004>	2	QHTTPSVR	QLWISVR	165	S
2505478	1,890	ADMIN5 QLWISVR 707548	00000000000402>	2	QHTTPSVR	QLWISVR	165	S
2187	1,890	ADMIN5 QLWISVR 707548	0000000000000018	2	QHTTPSVR	QLWISVR	165	S
2497149	1,887	ADMIN2 QLWISVR 707545	000000000005EB>	2	QHTTPSVR	QLWISVR	165	S
2505528	1,887	ADMIN2 QLWISVR 707545	000000000005FD>	2	QHTTPSVR	QLWISVR	165	S
2679	1,889	ADMIN4 QWEBADMIN 707547	00000000000000>	2	QHTTPSVR	QWEBADMIN	165	S
8364	1,890	ADMIN5 QLWISVR 707548	0000000000000>	2	QHTTPSVR	QLWISVR	165	S

Table View Example

6.8.1 Row Menu Options

A popup menu is available by right-clicking on any row within the table. The following options are available:

Menu	Description
Record Quick View	This option will display a vertical view of the current row(s) selected. If multiples are selected this option can be used to show a comparison between two rows in a side-by-side view.
Copy	Copies the current text selection to the clipboard. This may consist of rows or block of cells. Make a block selection by holding down the left mouse button and draw a box.
Find...	This menu allows the user to reposition the current record in a table view, based on input supplied if matching information is found.
Save | View As...	This option allows you to save the contents of a table view to a file. When using this option, the entire contents of the table are saved. You can choose between rich text, comma separated and tab separated text formats.
Save | Selection As...	The option allows you to save the **selected** contents of a table to a file. When using this option only the selected records or block of cells are written to the file. When using this option, you can choose between rich text, comma separated and tab separated text formats. This option is not available for graph views.
Save | Query Definition...	This option allows you to save the current table's query definition to the local user-defined reports database on the PC. Query definitions are saved into the user-defined queries folder under collections and can be reused. A window will be displayed asking for a description of the query and which component it applies to (if this is not already known).

Set Font	This menu displays a window allowing customization of the font used for all table views.
Preferences...	This menu displays a window allowing the user to set customized settings for the IBM iDoctor for IBM i application.
Graph Definition	This menu contains an option to create a new user-defined graph from the current report.
Query Definition	This menu contains a set of options each letting the user work with a portion of the query behind the active table view. **Tip:** This interface doesn't work with many types of SQL statements. This interface is considered obsolete. You should use the column filtering options instead or learn SQL and use the SQL editor.
Duplicate as table view	This option creates a copy of the current report as a new table view in the Data Viewer.
Properties	Displays the properties for the current report. The information shown in the property pages varies based on the type of report being viewed.
Search Google for 'X'	This option will open the default web browser and do a search on the contents of the table cell or column header you right-clicked on.

Depending on the type of report shown in a table view, other menus applicable to that report type will be shown. These are mentioned in the documentation for each of the components.

6.8.2 Column Menu Options

Right-clicking columns provide additional options for filtering, sorting or hiding the column from view. Using these options to filter/sort tables are recommend instead of using the Query Definition Interface.

The following options are available:

Menu	Description
Sort descending	Changes the sort order of the SQL statement, removing existing fields in the sort and adding the current field in descending order to the sort sequence.
Sort ascending	Changes the sort order of the SQL statement, removing existing fields in the sort and adding the current field in ascending order to the sort sequence.
Edit…	This option lets you change the column description shown for the selected column.
Add filter...	Displays the Filter interface for tables and provides options for defining filtering based on the current selection (column, row) in the table.
Remove selected filter	This option removes the filter defined for the current column. When filters are defined, the column header text is shown in a red color.
Remove all filters	This option removes all filters defined in the SQL statement that were added by the Table Filtering Interface.
Hide	Removes the selected column from view.
Unhide all columns	Redisplays all columns that were previously removed using the Hide menu option.

6.8.3 Making Row Selections

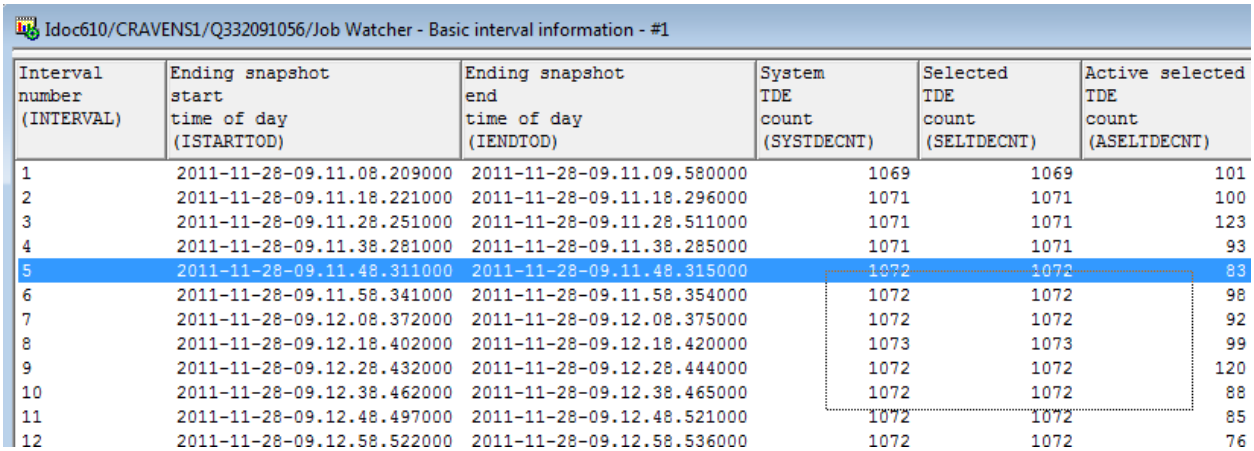
Row selections are made by selecting (clicking on) the desired row.
If you desire to select a continuous range of rows, then hold down the shift key and click on the 1st row and then click again on another row. All rows between the 1st and last selection will be selected.

After the selection is made you can copy it to the clipboard by pressing the Ctrl+C keys or using the Copy  button on the toolbar. You can also export the desired rows to an Excel (CSV format) file using the File -> Save Selection As menu. In addition, your numeric fields will be added up in some situations and shown in the Math Pane.

6.8.4 Making Cell Selections

Cells in table views are selected in iDoctor by performing the following action:

1) Left-click and hold the button down on the desired (cell). The point clicked on should be the upper left position of the set of cell(s) desired to be selected.
2) With the button held down , move the mouse down and to the right. During this process a box is drawn.



Interval number (INTERVAL)	Ending snapshot start time of day (ISTARTTOD)	Ending snapshot end time of day (IENDTOD)	System TDE count (SYSTDECNT)	Selected TDE count (SELTDECNT)	Active selected TDE count (ASELTDECNT)
1	2011-11-28-09.11.08.209000	2011-11-28-09.11.09.580000	1069	1069	101
2	2011-11-28-09.11.18.221000	2011-11-28-09.11.18.296000	1071	1071	100
3	2011-11-28-09.11.28.251000	2011-11-28-09.11.28.511000	1071	1071	123
4	2011-11-28-09.11.38.281000	2011-11-28-09.11.38.285000	1071	1071	93
5	2011-11-28-09.11.48.311000	2011-11-28-09.11.48.315000	1072	1072	83
6	2011-11-28-09.11.58.341000	2011-11-28-09.11.58.354000	1072	1072	98
7	2011-11-28-09.12.08.372000	2011-11-28-09.12.08.375000	1072	1072	92
8	2011-11-28-09.12.18.402000	2011-11-28-09.12.18.420000	1073	1073	99
9	2011-11-28-09.12.28.432000	2011-11-28-09.12.28.444000	1072	1072	120
10	2011-11-28-09.12.38.462000	2011-11-28-09.12.38.465000	1072	1072	88
11	2011-11-28-09.12.48.497000	2011-11-28-09.12.48.521000	1072	1072	85
12	2011-11-28-09.12.58.522000	2011-11-28-09.12.58.536000	1072	1072	76

3) Release the mouse button at the desired point, all cells within the box drawn will be selected.

Idoc610/CRAVENS1/Q332091056/Job Watcher - Basic interval information - #1

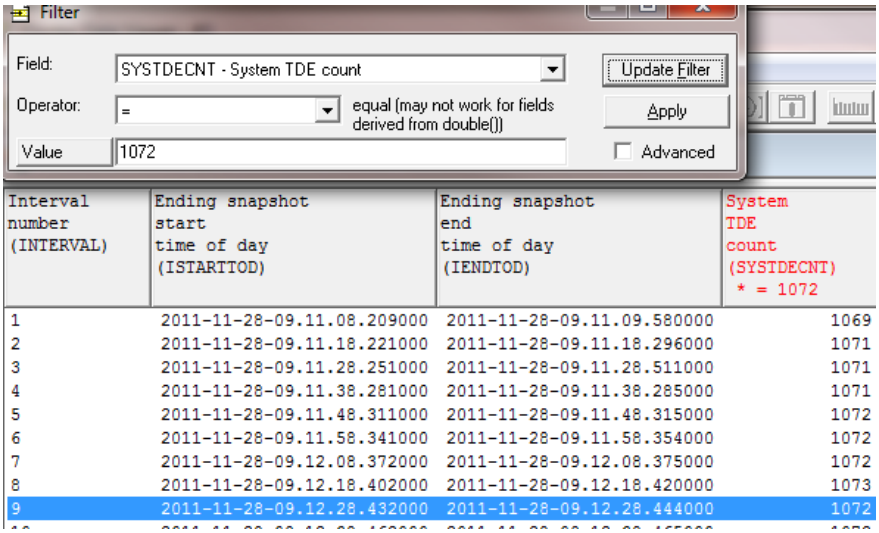
Interval number (INTERVAL)	Ending snapshot start time of day (ISTARTTOD)	Ending snapshot end time of day (IENDTOD)	System TDE count (SYSTDECNT)	Selected TDE count (SELTDECNT)	Active selected TDE count (ASELTDECNT)	Ma ta co (E
1	2011-11-28-09.11.08.209000	2011-11-28-09.11.09.580000	1069	1069	101	
2	2011-11-28-09.11.18.221000	2011-11-28-09.11.18.296000	1071	1071	100	
3	2011-11-28-09.11.28.251000	2011-11-28-09.11.28.511000	1071	1071	123	
4	2011-11-28-09.11.38.281000	2011-11-28-09.11.38.285000	1071	1071	93	
5	2011-11-28-09.11.48.311000	2011-11-28-09.11.48.315000	1072	1072	83	
6	2011-11-28-09.11.58.341000	2011-11-28-09.11.58.354000	1072	1072	98	
7	2011-11-28-09.12.08.372000	2011-11-28-09.12.08.375000	1072	1072	92	
8	2011-11-28-09.12.18.402000	2011-11-28-09.12.18.420000	1073	1073	99	
9	2011-11-28-09.12.28.432000	2011-11-28-09.12.28.444000	1072	1072	120	
10	2011-11-28-09.12.38.462000	2011-11-28-09.12.38.465000	1072	1072	88	
11	2011-11-28-09.12.48.497000	2011-11-28-09.12.48.521000	1072	1072	85	
12	2011-11-28-09.12.58.522000	2011-11-28-09.12.58.536000	1072	1072	76	
13	2011-11-28-09.13.08.552000	2011-11-28-09.13.08.567000	1072	1072	107	

After the selection is made you can copy it to the clipboard by pressing the Ctrl+C keys or using the Copy  button on the toolbar. In addition, your numeric fields will be added up, counted and averaged in the Math Pane in the status bar.

6.8.5 Filter

The Filter interface is accessed by right-clicking a column and using the Add Filter menu.
Filters can be defined on one or more columns and each filter will modify the SQL statement (within the where clause) to perform the desired filtering. The column header text of columns that have filters applied are drawn with a red color.

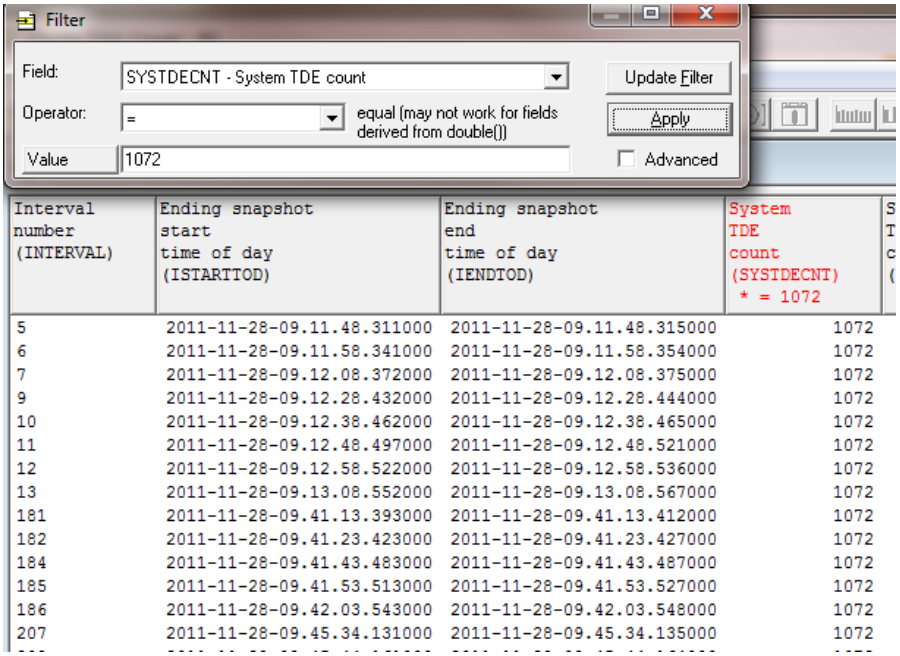
The following shows an example of a table with the Filter window open (but the filter is not yet applied):



Interval number (INTERVAL)	Ending snapshot start time of day (ISTARTTOD)	Ending snapshot end time of day (IENDTOD)	System TDE count (SYSTDECNT) * = 1072
1	2011-11-28-09.11.08.209000	2011-11-28-09.11.09.580000	1069
2	2011-11-28-09.11.18.221000	2011-11-28-09.11.18.296000	1071
3	2011-11-28-09.11.28.251000	2011-11-28-09.11.28.511000	1071
4	2011-11-28-09.11.38.281000	2011-11-28-09.11.38.285000	1071
5	2011-11-28-09.11.48.311000	2011-11-28-09.11.48.315000	1072
6	2011-11-28-09.11.58.341000	2011-11-28-09.11.58.354000	1072
7	2011-11-28-09.12.08.372000	2011-11-28-09.12.08.375000	1072
8	2011-11-28-09.12.18.402000	2011-11-28-09.12.18.420000	1073
9	2011-11-28-09.12.28.432000	2011-11-28-09.12.28.444000	1072

Table with Filter window, Apply button not yet pressed

Pressing the Apply button reruns the query using the desired filter and updates the results.

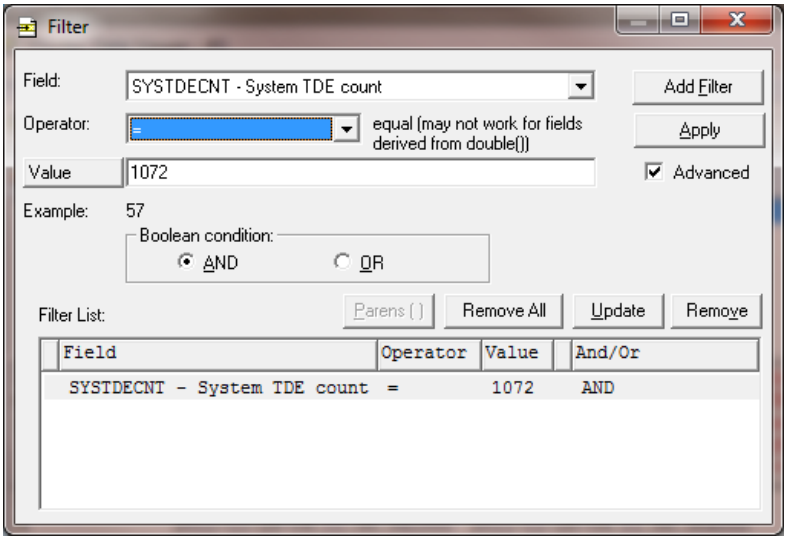


Interval number (INTERVAL)	Ending snapshot start time of day (ISTARTTOD)	Ending snapshot end time of day (IENDTOD)	System TDE count (SYSTDECNT) * = 1072	S T c (
5	2011-11-28-09.11.48.311000	2011-11-28-09.11.48.315000	1072	
6	2011-11-28-09.11.58.341000	2011-11-28-09.11.58.354000	1072	
7	2011-11-28-09.12.08.372000	2011-11-28-09.12.08.375000	1072	
9	2011-11-28-09.12.28.432000	2011-11-28-09.12.28.444000	1072	
10	2011-11-28-09.12.38.462000	2011-11-28-09.12.38.465000	1072	
11	2011-11-28-09.12.48.497000	2011-11-28-09.12.48.521000	1072	
12	2011-11-28-09.12.58.522000	2011-11-28-09.12.58.536000	1072	
13	2011-11-28-09.13.08.552000	2011-11-28-09.13.08.567000	1072	
181	2011-11-28-09.41.13.393000	2011-11-28-09.41.13.412000	1072	
182	2011-11-28-09.41.23.423000	2011-11-28-09.41.23.427000	1072	
184	2011-11-28-09.41.43.483000	2011-11-28-09.41.43.487000	1072	
185	2011-11-28-09.41.53.513000	2011-11-28-09.41.53.527000	1072	
186	2011-11-28-09.42.03.543000	2011-11-28-09.42.03.548000	1072	
207	2011-11-28-09.45.34.131000	2011-11-28-09.45.34.135000	1072	

Table with Filter window, Apply button has been pressed

The following options are available:

Element	Description
Field	This is the desired field to filter on. By default, this is the same field that was right-clicked.
Operator	The type of operation to use for this filter. The possible values are: = < <= > >= <> Is null Is not null Range List Not list **Note:** the = (equal) operator may not work for fields derived from the double function.
Value	This is the value to apply to the filter. Generally, the filter is something like FIELD OP VALUE where OP is the operator, FIELD is the field to filter on and VALUE is a constant numeric or text string.
Add/Update Filter	This button will add or update the desired filter in the table. The change does not take effect until the Apply button is pressed or the table is refreshed.
Apply	This button will rerun the query behind the table and apply any changes made to filters defined.
Advanced checkbox	This checkbox will hide/display the additional options on this window. They are typically not needed unless you wish to define more advanced options. See the section of the Record Selection tab of the Query Definition interface for more information (these options behave the same as in that interface)



Filter Window with Advanced options displayed

6.8.5.1 SQL Statement Changes

When filters are used, the SQL statement is modified so that the existing SQL statement becomes a subselect of a new statement. For example the SQL behind the table shown in the previous section is:

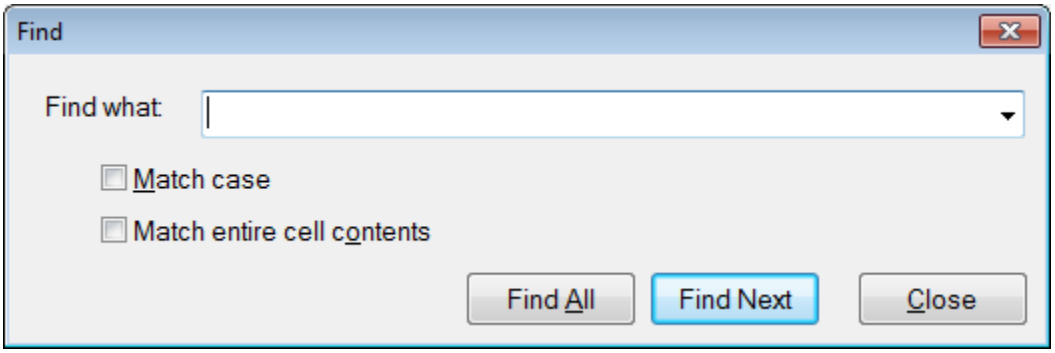
```
SELECT * FROM (SELECT * FROM CRAVENS1/QAPYJWINTI ORDER BY INTERVAL)    GUI_FILTER
WHERE SYSTDECNT = 1072  ORDER BY INTERVAL
```

The identifier "GUI_FILTER" is used by the iDoctor GUI to indicate that this special type of filtering has been defined. Removing or changing this value in the SQL Editor will cause the filters to be no longer usable via the GUI.

6.8.6 Find Window

The Find Window allows a user to perform a search over a Table View. Use the Edit -> Find... menu or right-click on a Table View and choose the Find... menu to use the Find Window. Find allows the user to search for a text string within a specific column.

An example of the find window is shown below:



Find Window for a Table View

The following describes the elements on the Find window.

GUI Element	Field Description
Find what	Enter the string you would like to search for.
Match case	Check this to perform a case-sensitive search.
Match entire cell contents	Check this to indicate that records should only match if the value in the find what text box matches an entire column within the row exactly.
Find All	Click this button to find and select all matching occurrences of the Find what value in the table.
Find Next	Clicking this button will perform a search over the active Table View for the next occurrence (depending on direction up or down) of the string in the Find what textbox in the specified column. **Note:** You can also close this window and press F3 to do this action. Shift+F3 will find the previous occurrence.

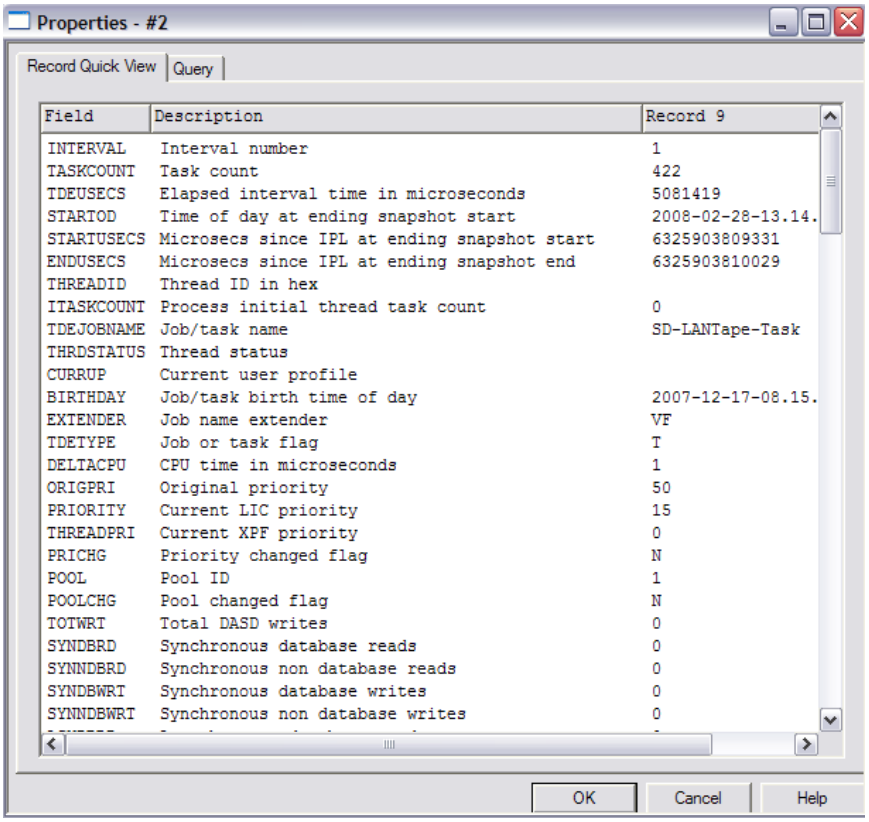
6.8.7 Properties

This section describes the properties available for all table views in iDoctor. Additional tabs are available in some components for certain types of reports.

6.8.7.1 Record Quick View

This window is part of the property pages for a table view. The Record Quick View page shows all data for the selected record from the table in a vertical list. This can make it easier to see all the data for a single record if many fields exist in the table. Access this window by double-clicking on any record in a table view or by using the Properties menu.

An example of this window for a Job Watcher table view is the following:



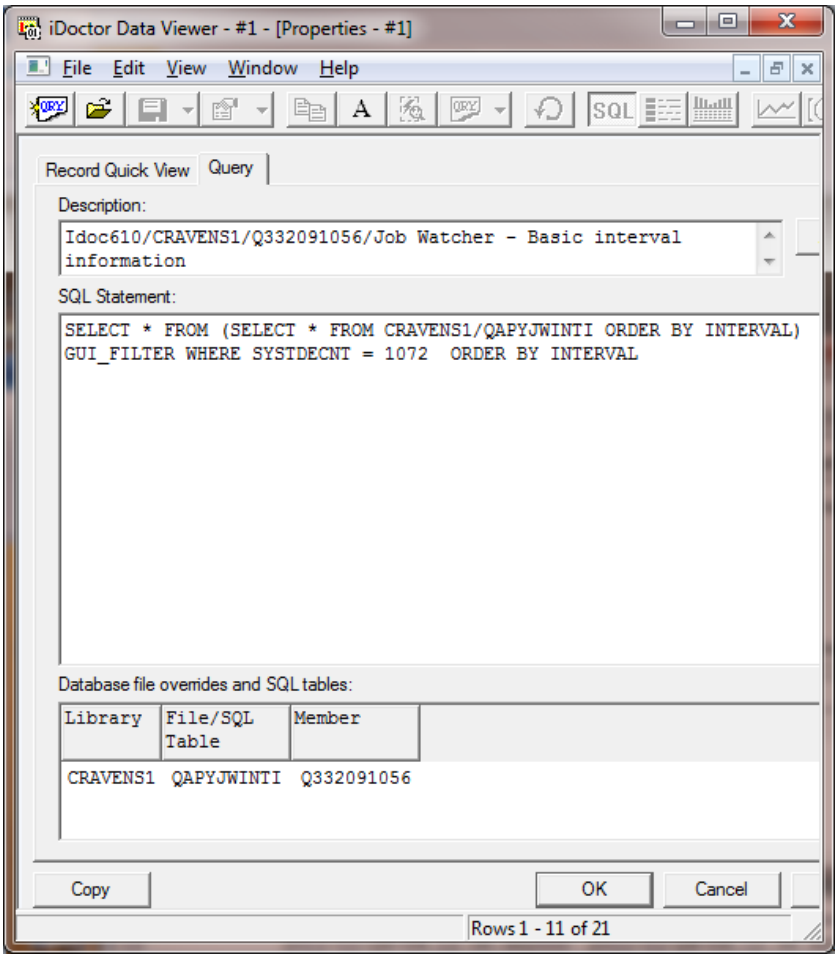
Record Quick View

6.8.7.2 Query

The Query page of the Properties window displays the SQL statement used to produce the current table view. This window also displays the title of the table view and the overrides (or SQL tables) used to produce the current table view. Because SQL does not support multiple member tables, overrides (see the OVRDBF command) are issued before the query is executed to select which file(s)/member(s) should be used when running the SQL statements.

Note: In some cases, aliases are used instead of database file overrides, such as when querying multiple collections simultaneously in Job Watcher.

An example of this page is the following:



Properties - Query Tab

The interface elements within this window are described in more detail in the table below:

Interface Element	Description
Description	The text description identifying the report. This is shown on the title bar of the view and can be changed. Update the view's title using the Apply button.
SQL Statement	The complete SQL statement for the query definition.
Database file overrides and SQL tables	This list identifies all physical file/members or SQL tables or aliases in the SQL statement. When a physical file and member is listed an override is used to point to a specific member when executing the query. Aliases used by iDoctor are typically created in QTEMP. **Note:** If the SQL statement contains aliases, then the library and alias name will be listed, and the member field will be blank.

6.8.8 Query Definitions

NOTE: <u>The interface options in this section are provided "as-is" and will not work properly for several types of SQL Statements. Using the Filter interface, or the SQL Editor is highly recommended instead.</u>

Tables and graphs are created via an underlying query definition or SQL Statement. The query definition defines exactly how data is to be retrieved and from what files(s), SQL tables, views or aliases. The Query Definition Interface is an interface over (the outermost portion of) an SQL statement. Most table and graph views in iDoctor provide a query definition menu that lets the user work with the SQL Statement behind the report.

The Query Definition Interface allows a user to customize the query for the active table or graph within the Data Viewer. Right-click on the view and use the Query Definition menu to open the Query Definition Interface for the desired tab.

The tabs within the interface are:

Field selection - indicates the order of the fields and the ones to display or hide
Record selection - used to filter out or only include records that meet certain characteristics
Sort by - indicates which field(s) the data should be sorted by
Group by - allows for the definition of the group by and having clause of the SQL statement

To use the more advanced features of the Query Definition Interface like the "group by" page, the user needs to understand basic SQL statement syntax. However, most features like field selection, record selection and sort by have been designed to be understandable by anyone.

6.8.8.1 Limitations

The Query Definition interface is built by parsing the contents of an SQL statement. This parsing works well for many queries but it does not acknowledge all types of SQL syntax. It will parse most SQL select statements containing "joins" but there are some very complex statements that are not parseable. Although a query can be parsed that contains joins the types of joins, and the files being joined are not changeable through the interface.

The query definition can be used to adjust the where clause, order by and group by clauses of the outermost part of the SQL statement. Any order by clauses, where clauses, or group by clauses for subqueries within the SQL statement are not configurable through this interface.

This interface does not support any type of statements except SQL SELECT statements.

6.8.8.2 Field Selection

The field selection panel allows you to hide or reorder the fields in the associated table view. You may also use this panel to create your own fields by using the buttons at the top of the window to add or edit fields.

The following types of operations can be performed on the field selection page:

- Changing field visibility
- Reordering fields
- Creating/editing fields

Instructions for performing each of these types of operations follows:

Working with field visibility

Visible fields are indicated by a checkmark in the Show? column within the Field List. If a field is not checked, then it will not be shown.

You may use the Toggle Selected button to check/uncheck the checkbox for the selected fields. This can be very handy when you want to hide or show a large number of fields at once.

Reordering fields

The order that the fields are displayed in the Field List, directly affects the order that the fields are displayed in the table view.

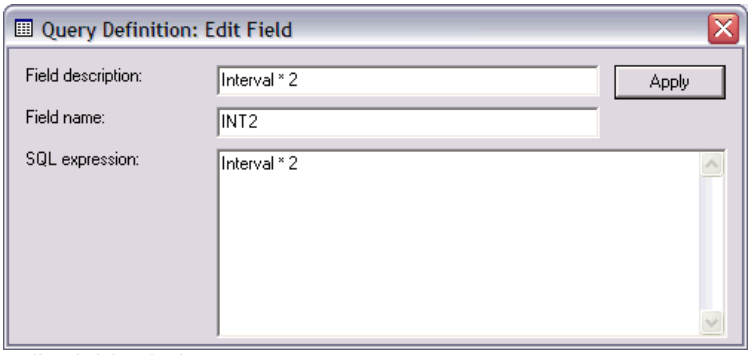
To reorder fields:

1. Select the fields you wish to reorder using the mouse and ctrl/shift keys.
2. Press the left mouse button over one of the selected fields and hold it down.
3. Drag the selection to your desired position in the list. You can scroll through to the bottom of the list if desired.
4. Release the left mouse button.

Creating new fields

To create a new field:

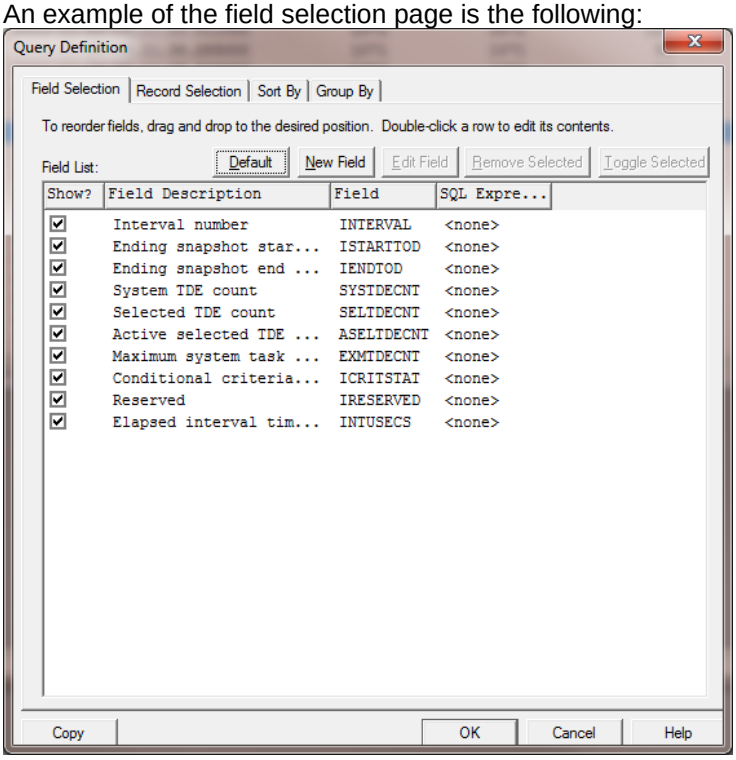
1. Click the New Field button. After doing this a new field will be added to the list below the current selection.
2. Double click on the new field added to the list or select it and press the Edit Field button.
3. Modify the field description, field name and SQL expression through the Edit Field window. The field description will be shown as the column header. The field name is the short name identifier and the expression must be a valid SQL expression for the field list for this query.



Edit Field Window

If desired the Edit field window can remain open to change multiple fields at once by clicking other fields from the field list. The values for the selected field will be shown in the Edit Field window as the selection changes.

An example of the field selection page is the following:

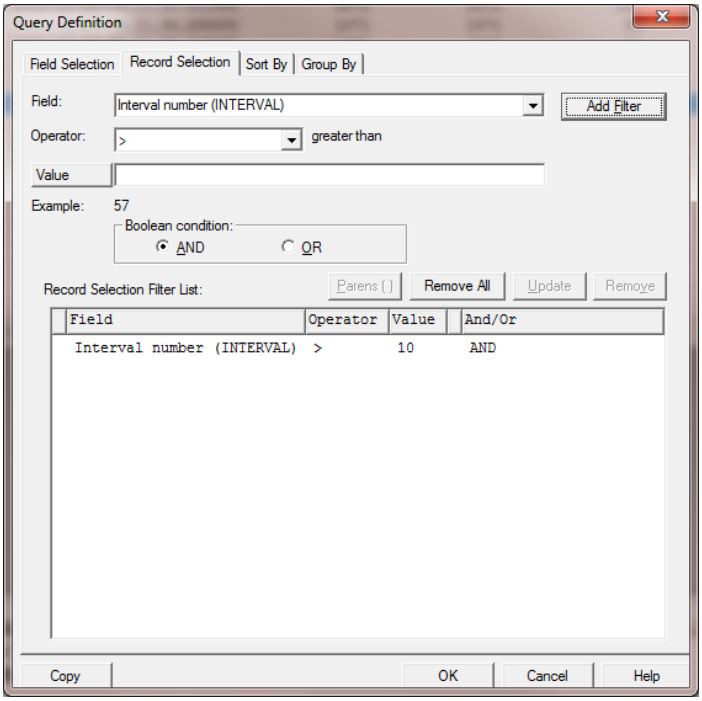


Field Selection Tab

6.8.8.3 Record Selection

The Record Selection Tab allows a user to limit the number of records returned in the active table or graph view.

An example of the Record Selection Page is shown below:



Record Selection Tab

To Add a Filter:

1. First select the field or type in its short name into the field drop down list.
2. Depending on the type of the field selected, various operators available will be displayed in the operator list.
3. Select the desired operator from the operator list.
4. Type in the value that the operator should test for. For example, to specify only records where CPUTIME field is greater than 10 the operator selected would be > (greater than) and the value would be 10.
5. Press the Add Filter button to add the filter to the list.
6. Press the OK button to close this interface and run the query using the new filter.

By selecting more than one concurrent records in the list and pressing the 'Parens ( )' button the user can add or remove a set of parentheses. To remove parentheses around multiple filters, select the range of records that contain the starting and ending parentheses and click the 'Parens ( )' button. Parentheses are necessary in order to make complex evaluations in the where clause of an SQL statement such as: CPUTIME >10 OR (IO > 1000 AND CPUTIME >= 1)

As the selection changes in the list, the interface objects above the list will change based on the current selection. This allows the user to quickly change values in the filter list by selecting any item in the list, changing any values from the fields above the list, and clicking the 'Update' button. The 'Update' button will update the selected row in the filter list.

A description of all the GUI elements on this panel follows:

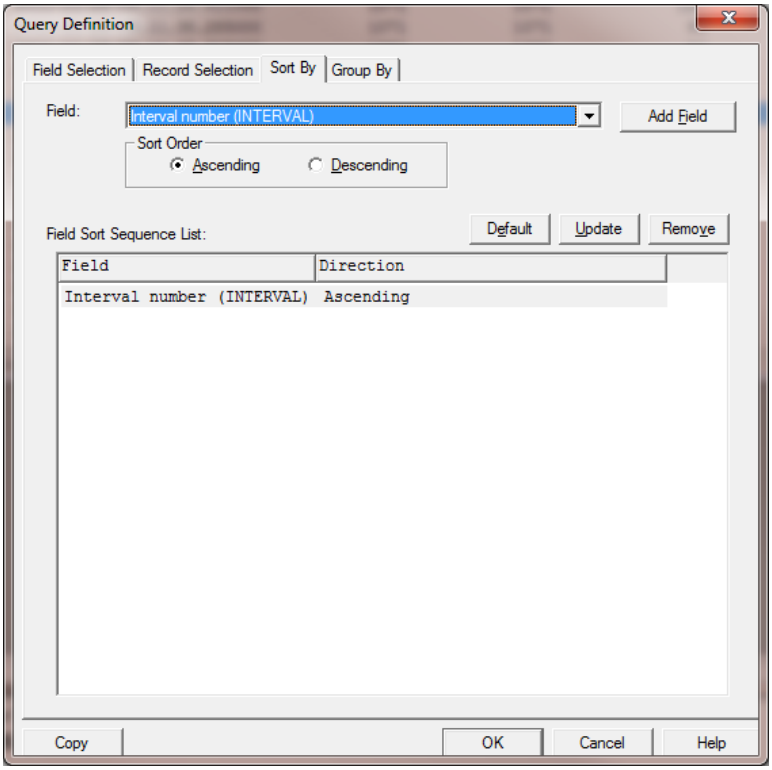
GUI element name	Description
Field drop-down list	This is a list of every field in the current report. Select a field to filter by before clicking the 'Add Filter' or 'Update' buttons. The short name of a field may also be entered.
Operator list	This is a list of every operator available for the currently selected field. A text field has a different set of available operators than does a numeric field. The set of operators is also different for a timestamp field. The operators 'Field contains', 'Field starts with', 'Field ends with', 'Field xxx', etc are not valid for numeric and timestamp fields. The following operators are supported on this page: Equal Less than Less than or equal to Greater than Greater than or equal to Not equal Is null Is Not null Range List Not List Field contains Field starts with Field ends with Field does not contain Field does not start with Field does not end with
Value text box	Use this textbox to enter the value to apply to the current field using the selected operator. The value should match the format presented by the 'Example' label directly beneath the text box. Text fields should have their values enclosed in 'single quotes' and if the operator is 'Range', 'List' or 'Not list' then more than one values each separated by a space is expected. Whenever entering a value, follow the example provided.
Add Filter button	This button creates a new filter and adds the filter to the Record Selection Filter List.
Value/Expression button	This button allows the user to enter a valid SQL expression instead of a single value. This provides greater flexibility but requires that you know SQL syntax. Any errors in the SQL statement will prevent the query from running and will cause an SQL error message.
AND/OR options	Use this to indicate whether two filters should be ANDed together or OR'd together.
Parens ( ) button	The 'Parens ( )' button allows grouping of multiple filters in the Record Selection Filter List into a single logical expression by placing parentheses around the set of filters. If parentheses already exist for the starting and ending record in the selected range, the parenthese will be removed by pressing this button.
Remove All	This button will clear the list of filters.
Update button	This provides the ability to change the selected filter from the Record Selection Filter List.
Remove button	This button allows the user to remove one or more records from the Record Selection Filter List.
Record Selection Filter List	This is a list of all of the active filters to be applied to the report. Use the 'Add Filter' button to add a filter to the list. Press the OK button on the bottom of the Query Definition dialog to close the dialog and display the report using the filters from the list.

6.8.8.4 Sort by

The Sort By Page allows a user to change the order in which records are sorted in a table or graph view. This screen displays a list of fields to sort by and the sort direction for each field. The field at the top of the list has highest precedence in the sort sequence.

To add a field to the sort sequence list:

1. Select the field to add to the list using the Field drop-down list.
2. Select the sort order: ascending or descending.
3. Press the Add Field button. The new field will be added to the Sort Sequence list.



Sort By Tab

The GUI elements on this page are described in the table below:

GUI element name	Description
Field drop-down list	This is a list of every field available to sort by in the active view. Choose the field to add to the 'Field Sort Sequence' list before clicking the 'Add Field' or 'Update'
Add Field button	This button adds a field to the sort sequence list, using the sort order currently specified.
Sort order options	Each field may be sorted in ascending (A-Z) or descending (Z-A) order. Choose the desired sort order before clicking 'Add Field' or 'Update'.
Default button	The default button changes the active sort order to whatever the default sort sequence is for the active report. In most cases, this will clear the sort sequence to nothing. In this case the sort order will be based on an ascending sort by relative record number of the raw data in the file.
Update button	The update button will change the sort sequence definition for the currently selected item in the list.
Remove button	This button allows the removal of one or more sort definitions from the list.
Field sort sequence list	This list represents the current sort order to apply to the active table or graph. The field at the top of the list has highest precedence in the sort sequence. Selecting any item in the list allows the option to change the value for the selected item using the 'Update' button.

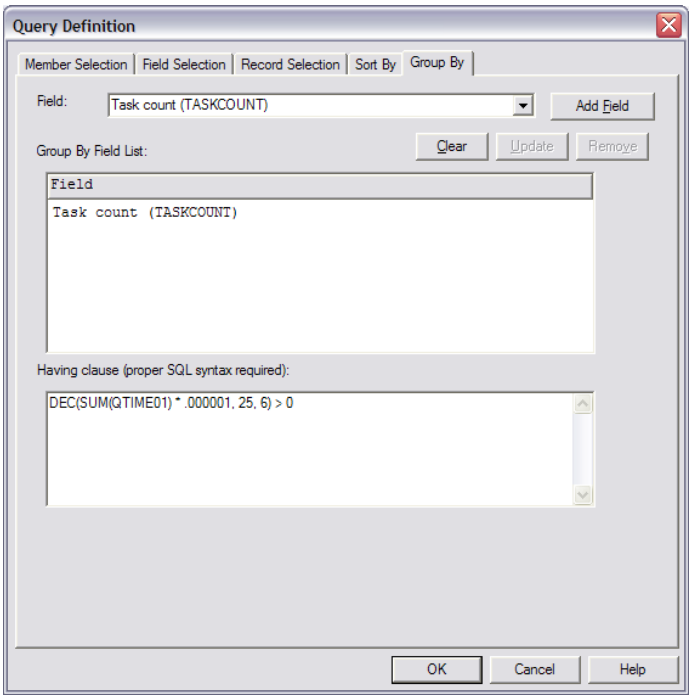
6.8.8.5 Group by

The Group By Tab allows a user to define the fields that should be used as part of a Group by query. The field at the top of the list has highest precedence in the SQL GROUP BY clause.

Group by queries are only valid when the fields on the field selection page comply with the rules SQL has with running group by SQL statements. Any fields that are not part of the group by clause must be summarized in order to exist in the field selection or the query will not run.

To add a field to the group by list:

1. Select the field to add to the list using the Field drop-down list.
2. Press the Add Field button. The new field will be added to the Group By Field List.



Group By Tab

The GUI elements on this page are described in the table below:

GUI element name	Description
Field drop-down list	This is a list of every field in the active view. Choose the field to add to the 'Group By Field' list before clicking the 'Add Field' or 'Update' button
Add Field button	This button adds a field to the group by list.
Clear button	Removes all fields from the list.
Update button	Use the update button to change the selected field in the list to match the selected field in the drop-down list.
Remove button	The remove button will delete all selected fields from the Group By Field List.
Group By Field List	This list represents the GROUP BY clause in the Group By query. Selecting any item in the list allows the option to change the value for the selected item using the 'Update' button.
Having clause	This is the exact syntax to use for the Having clause for the group by query. Specifying a Having clause is not required.

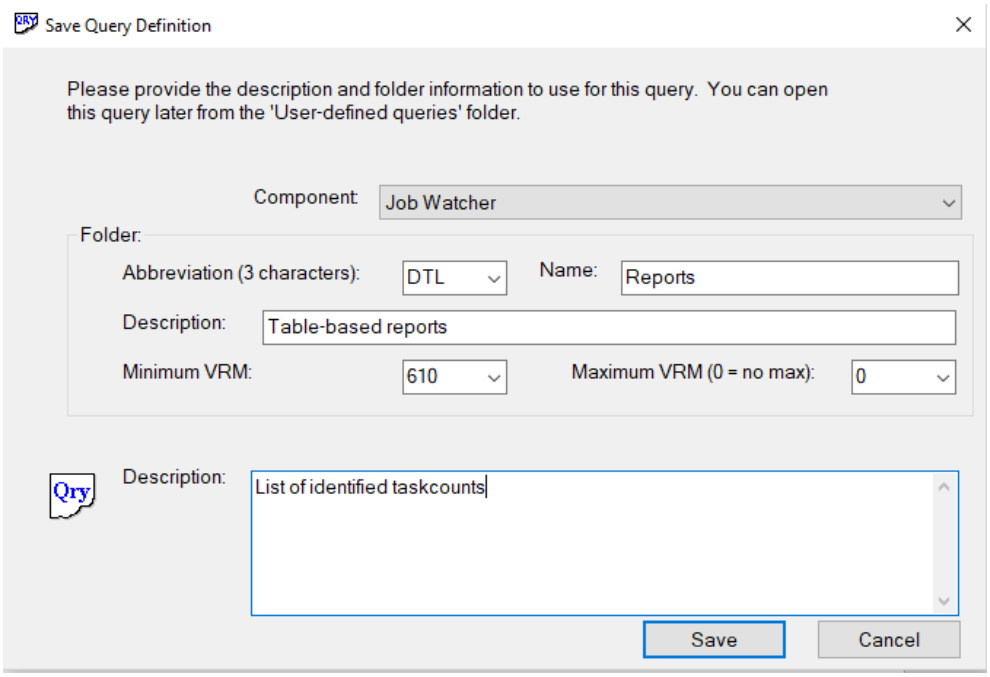
6.8.8.6 Reset

The reset submenu under the query definition popup menu may be used to reset a query back to its original state. Whenever a table or graph view is loaded the initial SQL statement is saved. If at some point it is desired to discard the changes made to the report, use the Reset menu.

6.8.8.7 Save Query Definition (Save As...)

Query Definitions are saved using the Query Definition -> Save As... menu for an active table view. The query definition behind a graph view is saved using the Graph Definition -> Save As... menu for a graph view. All Query Definitions are saved into the user-defined reports database. You can specify the folder name to save the query definition into within this database. These are accessed later from the User-defined reports folder for the collection type you are working with. Typically, these queries can be reused on any collection data of the same type (i.e. Job Watcher.)

An example of the Save Query Definition interface is shown below:



The interface elements within this window are described in more detail below:

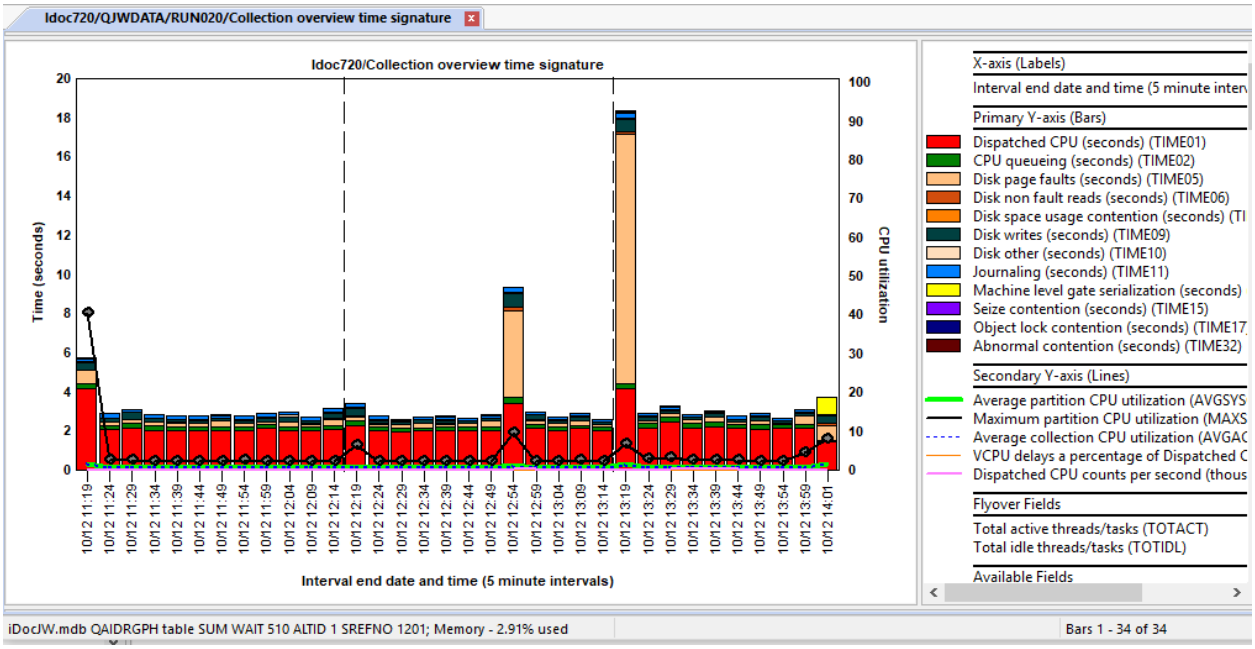
Option	Description
Component	The name of the component this query should be visible to.
Folder abbreviation	3-character identifier for the folder the query should be saved into. If you wish to define new folder give it a name and unique abbreviation like 'DX1'
Folder name	The name of the folder to store the query into
Folder description	A long description to give the user-defined folder. (optional)
Minimum VRM	The minimum IBM i VRM that this query should be visible too in nnn format (i.e. 610, 710, 730, etc.) If the collection was created on a system older than this value, then this query will not appear.
Maximum VRM	The maximum IBM i VRM that this query should be visible too. Use the value of 0 for no maximum.
Description	The user-defined description for the query. This description can be up to 250 characters long.

6.9 Graph Views

The graph views in iDoctor display line, bar, area and pie charts built using SQL statements executed against data on the system. There are several different types of graphs supported: vertical stacked bar, vertical bar (side-by-side), horizontal stacked bar, horizontal bar (side-by-side), pie chart, step, area, vertical overlapping bar and horizontal overlapping bar.

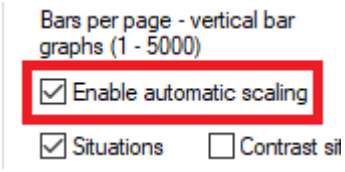
In most cases, each color in the graph represents a different field from the query and each stacked bar represents a single row in the query results returned. In a few cases in iDoctor, a new 'flattening' technique is used where a single stacked bar may be built from many rows and different values of a single field make up the various colors. (See 12x loop advanced graphs in CSI)

The graphs provide an optional attached legend identifying the fields in the SQL statement by color and where used within the graph. This button  on the toolbar may be used to hide or show the legend.



Job Watcher graph built from multiple collections

If necessary, use the scroll bars to navigate through the data shown in the graphs. Due to the potential to view vast amounts of data at one time, the graph data is shown a page at a time. The number of bars shown per page is configurable through the Preferences interface. When scrolling through the data the scale of the axes can be set to adjust automatically. This is another option on the Preferences interface.

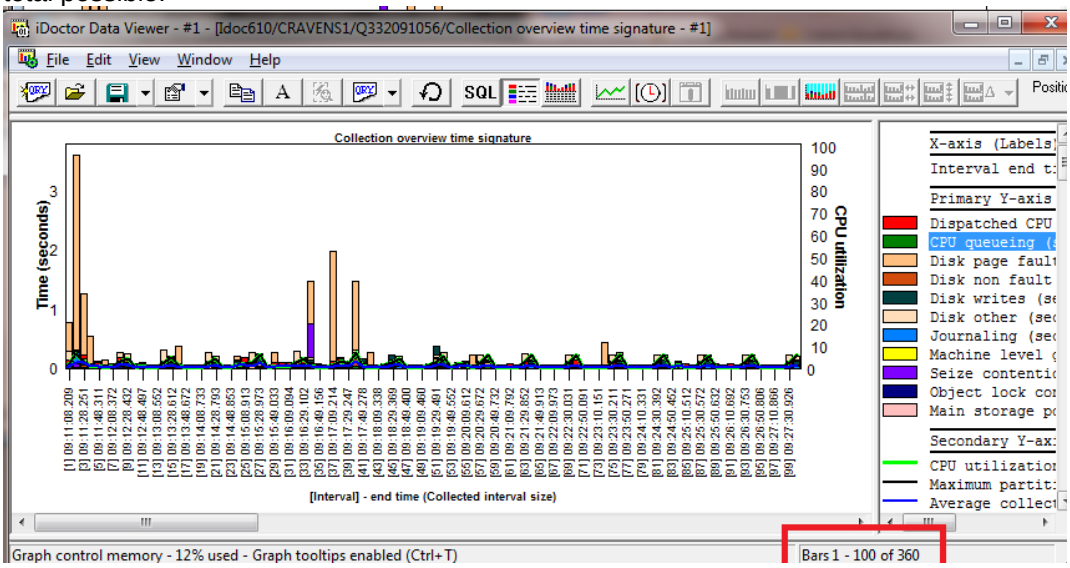


Preferences -> Display -> Enable automatic scaling

If automatic scaling is disabled, then the graph scale will be set to the maximum/minimum values of the first page shown in the graph.

Additional information about each piece of data in the graph is available by moving the mouse over the bar of interest. A flyover help window will appear in yellow providing this information. Some of this information is also displayed in the status bar as the mouse moves from bar to bar. The user can also click on any bar to get a complete look at all the information for that particular piece of the graph and any other applicable data that goes with it (interval, job, etc)..

The Position Indicator Pane in the status bar indicates exactly which bars are being viewed out of the total possible.



Graph position indicator pane in status bar highlighted in red

6.9.1 iDoctor-supplied graphs

iDoctor-supplied graphs are graphs shipped by IBM within iDoctor. Typically, iDoctor-supplied graphs will have additional drill-down options that are not accessible from the user-defined graphs.

6.9.2 User-defined graphs

User-defined graphs are created by the user and saved into a graph definition within the iDoctor User Defined Reports Database.

A user-defined graph can be initially created either from a table view or by modifying and saving an iDoctor-supplied graph. Creating a graph from a table view is done using the **Graph Definition | Define New...** popup menu of a table view.

6.9.3 Axis Types in iDoctor

All iDoctor graphs fall into two categories: vertical and horizontal.

6.9.3.1 Vertical graphs

Vertical graphs have an X-axis (labels) at the bottom, the primary Y-axis is on the left and the optional secondary Y-axis is on the right (always lines.) The primary Y-axis is what defines the graph type (see the next section.)

The list of graph types that are vertical are:

1. Step
2. Lines
3. Pie
4. Vertical bar

6.9.3.2 Horizontal graphs

Horizontal bar graphs have an X-axis (labels) on the left, the primary Y-axis is on the bottom and no secondary Y-axis.

The list of graph types that are horizontal are:

1. Gantt
2. Horizontal bar

6.9.4 Graph Types

iDoctor supports many different types of graphs. Most iDoctor graphs are stacked bar graphs with either horizontal or vertical bars. The vertical bar graphs in iDoctor will often have lines on the Y2-axis.

Users can customize the graphs how they'd like and save them as user-defined graphs to the User Defined Reports Database. Use the graph popup menu Graph Definition -> Set Graph type to change the graph type.

Note: The graph type refers **only to the primary Y-axis**. If supported a secondary Y-axis showing lines may also exist.

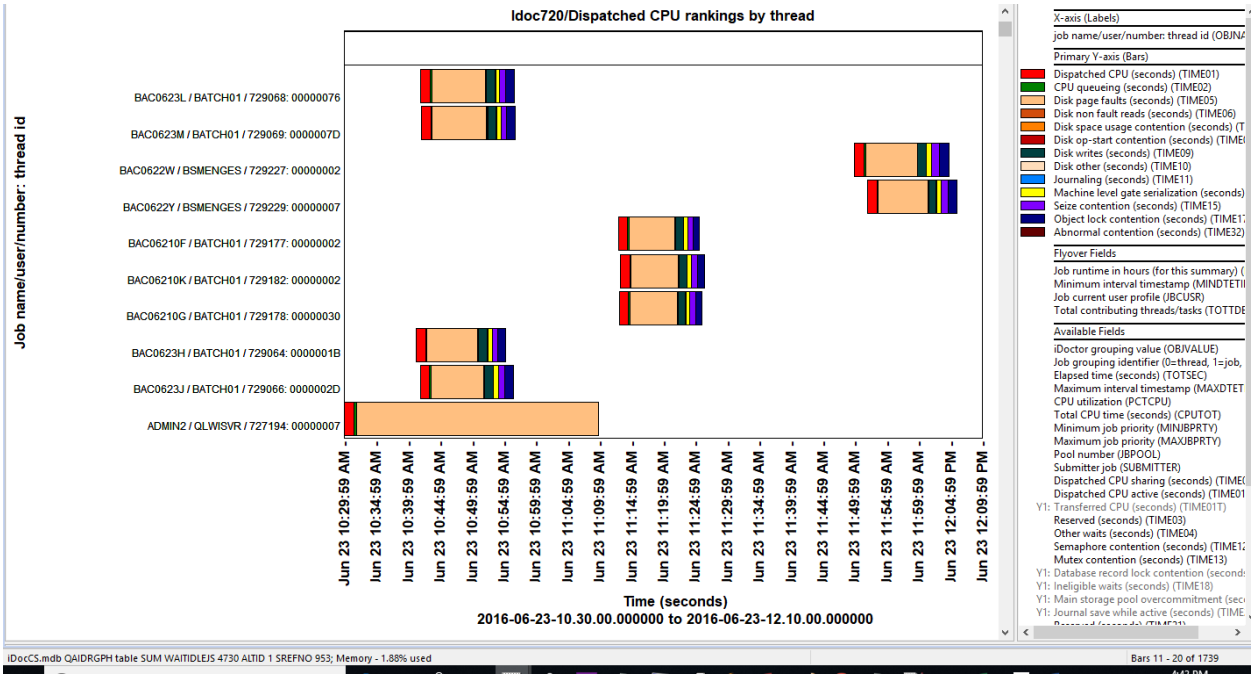
Examples for each type of graph available are listed below:

6.9.4.1 Gantt

Gantt charts are used in iDoctor to display where a job executed over time in comparison to other jobs.

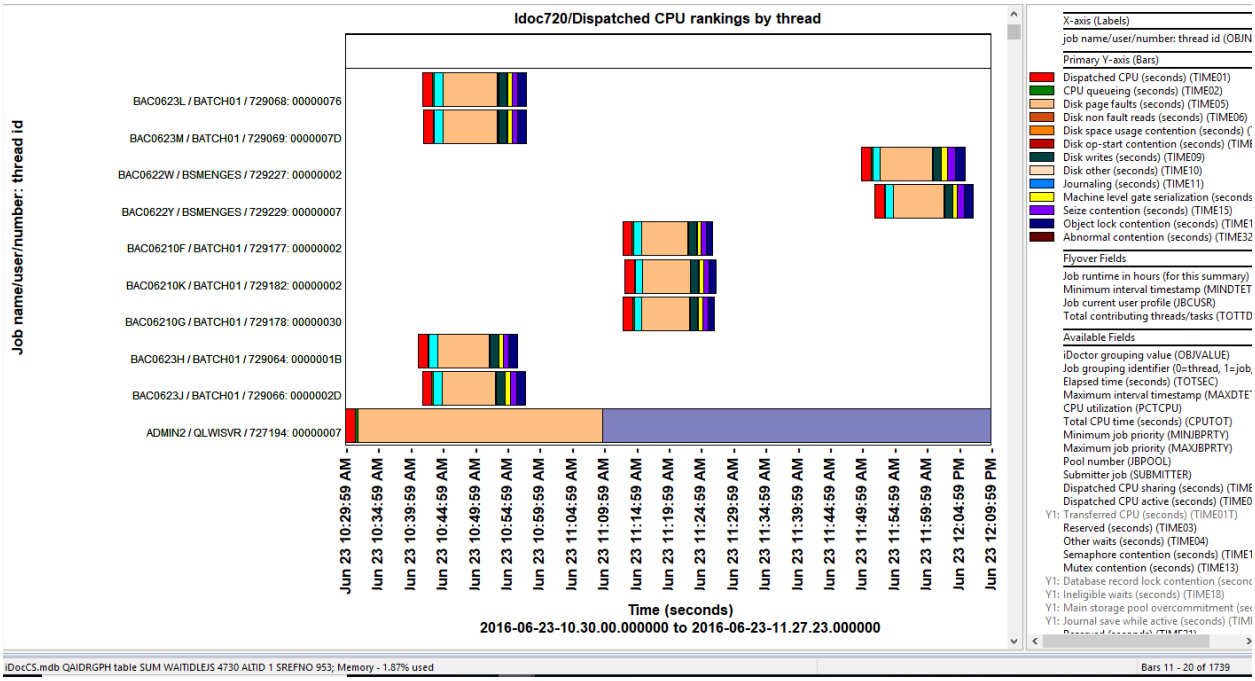
The time of day is listed on the bottom and each job starts and ends throughout that period as shown in the graph.

Note: Keep in mind the duration of the job could be longer than what is visible here depending on which wait buckets (or other metrics) are graphed. Only if ALL wait buckets are included would the duration of the jobs be accurate in this style of graph.



Gantt chart in Collection Services Investigator (excluding idle waits)

To include all wait buckets on this chart, press this button  on the Data Viewer toolbar. Pressing it again would toggle these idle wait off.

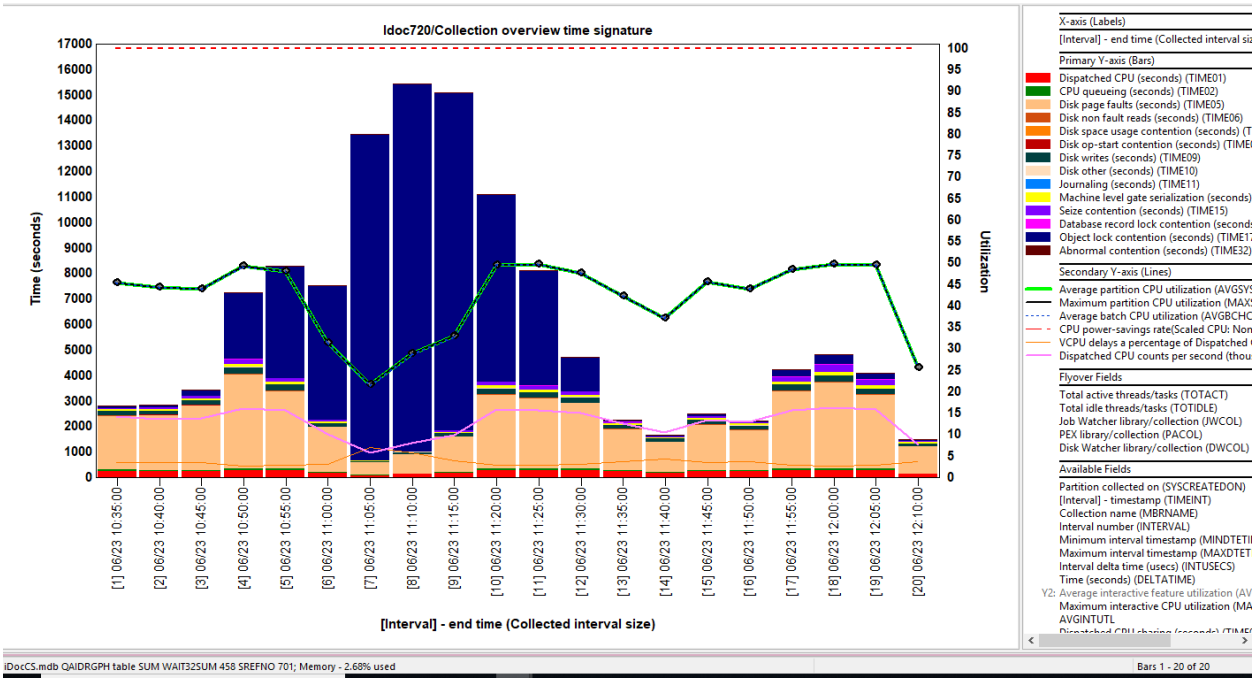


Gantt chart including idle waits

6.9.4.2 Step

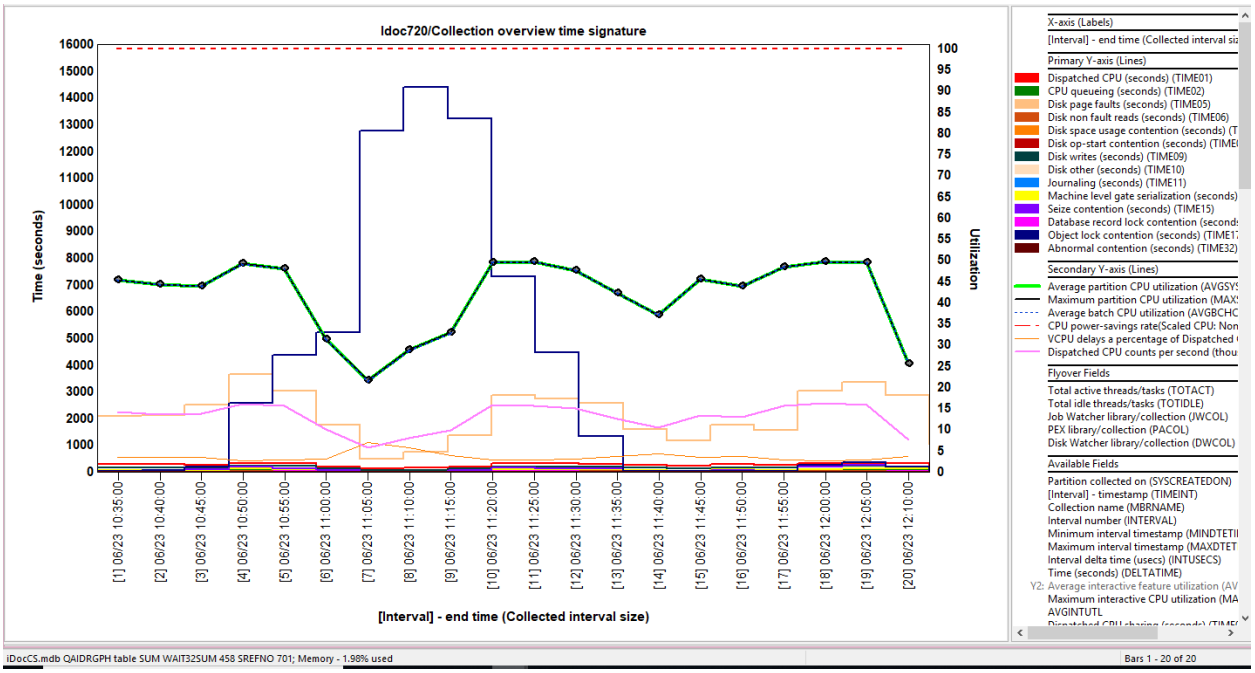
Step charts are line graphs, but the lines are drawn like stairs in a staircase instead of straight lines.

For example, a vertical stacked bar graph in CSI looks like this:



Vertical stacked bar graph

But when converted to a Step graph looks like this:

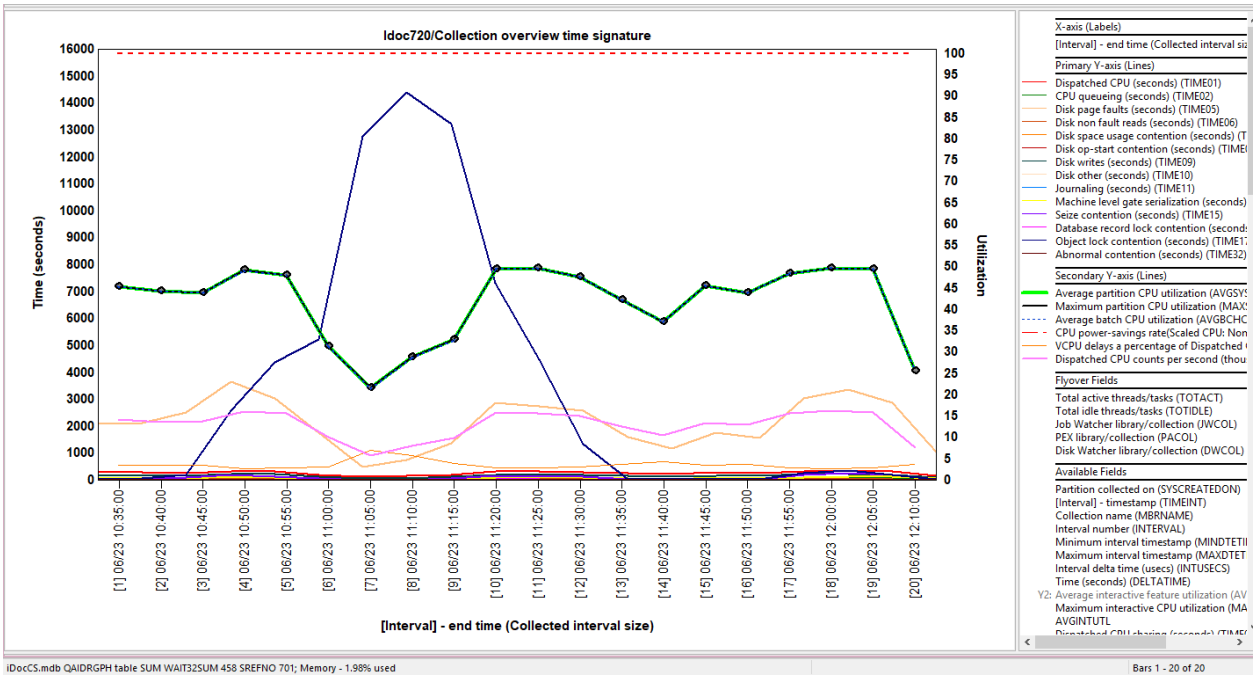


Step graph example

6.9.4.3 Lines

A lines graph in iDoctor can be used in iDoctor if desired by the user. It changes the metrics on the primary Y-axis to each be a line.

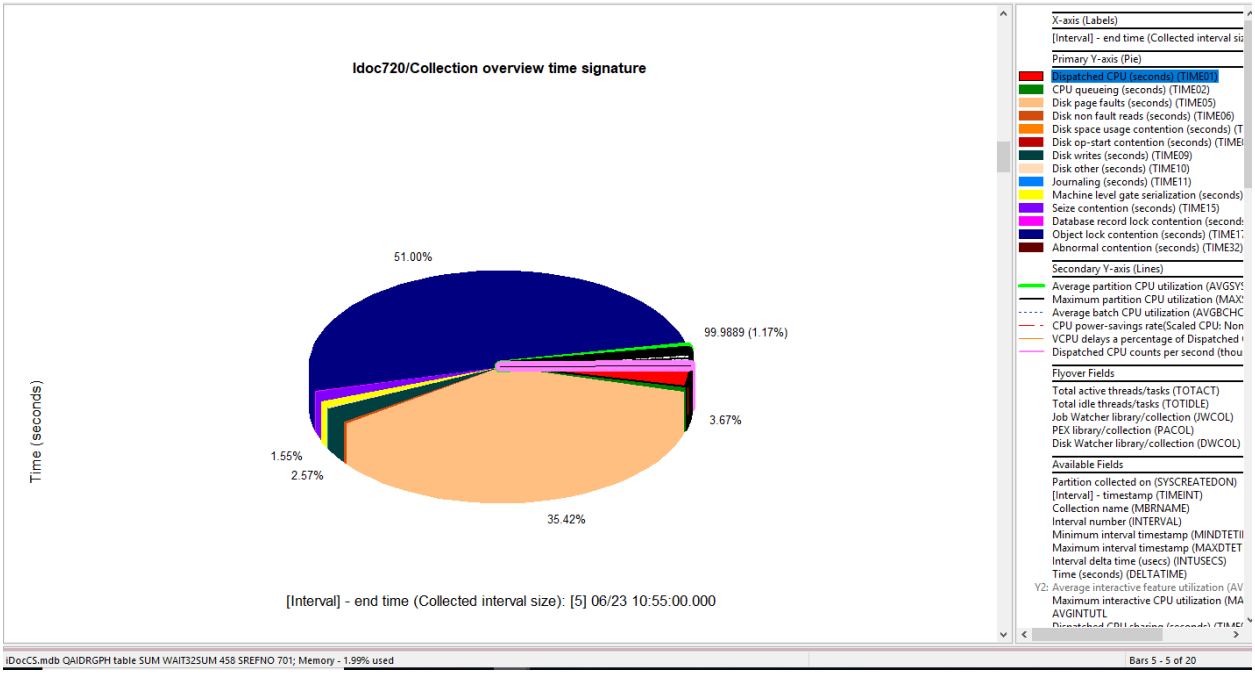
Note: The secondary Y-axis line options for widgets are not currently available on the primary Y-axis for this type of graph.



Line graph example

6.9.4.4 Pie

Pie charts display the metrics on the Y1-axis as percentages of the total of all values. This type of chart works best if the bars shown per page is set to 1. You can do this using the **Graph Definition -> Set bars per page -> 1** menu option or simply use the **Graph Definition -> Set graph type -> Pie (with 1 record per page)** option to do this in 1 step.



Pie chart example

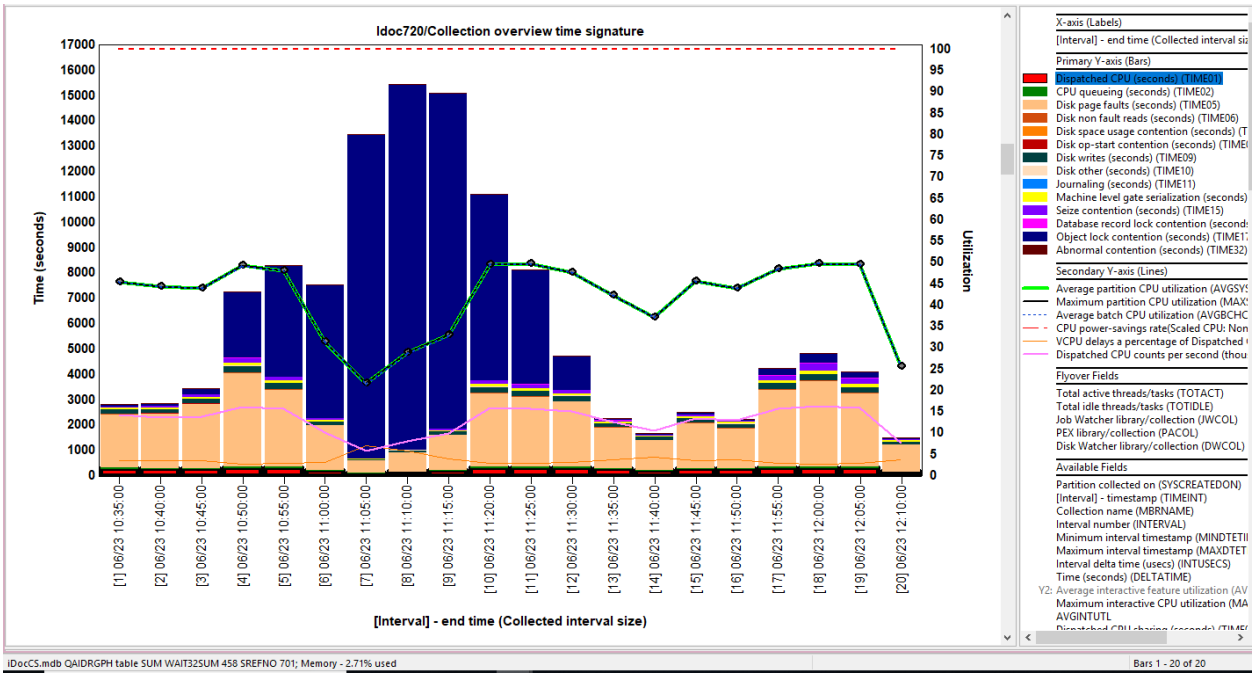
6.9.4.5 Vertical bars stacked

This type of graph is the most common type used in iDoctor. It allows both a Y1 and Y2 axis, giving multiple types of data on the graph simultaneously. The Y1-axis contains stacked bar colors. If you need to see the total value of all colors for a record or time interval then simply click on it to select it and the Math Pane in the status bar will give you the total SUM of those values.

The Y2-axis in all iDoctor graphs are always lines. These can be configured from the legend by right-clicking on them to add widgets (shapes) to the line points if desired. Various styles of lines can also be used (dashed, dots, etc) and the colors and widths may also be adjusted.

Tip: Graphs of this type allow the use of the Toggle Graph Format  button on the toolbar.

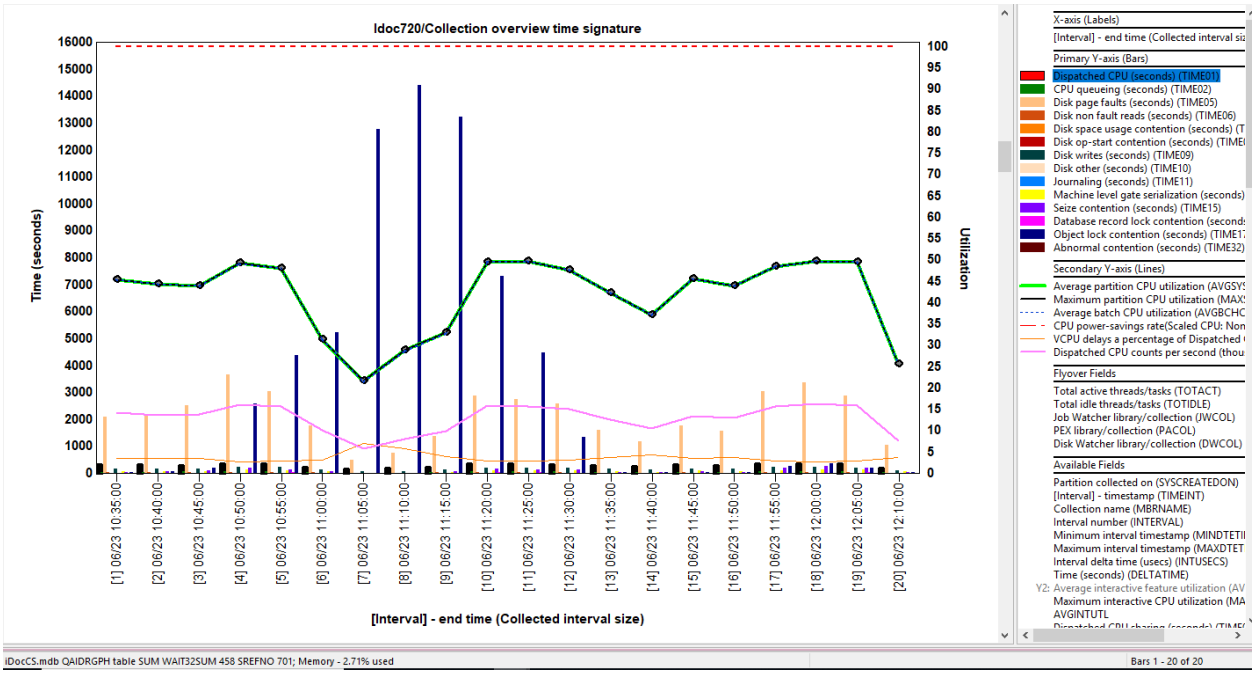
1. Pressing this button for a time-based graph will convert the graph to Lines on the Y1 axis instead.
2. Pressing this button for a rankings graph will convert the graph to Horizontal stacked bars and the Y2 axis will not be available.



Vertical stacked bar graph

6.9.4.6 Vertical bars side-by-side

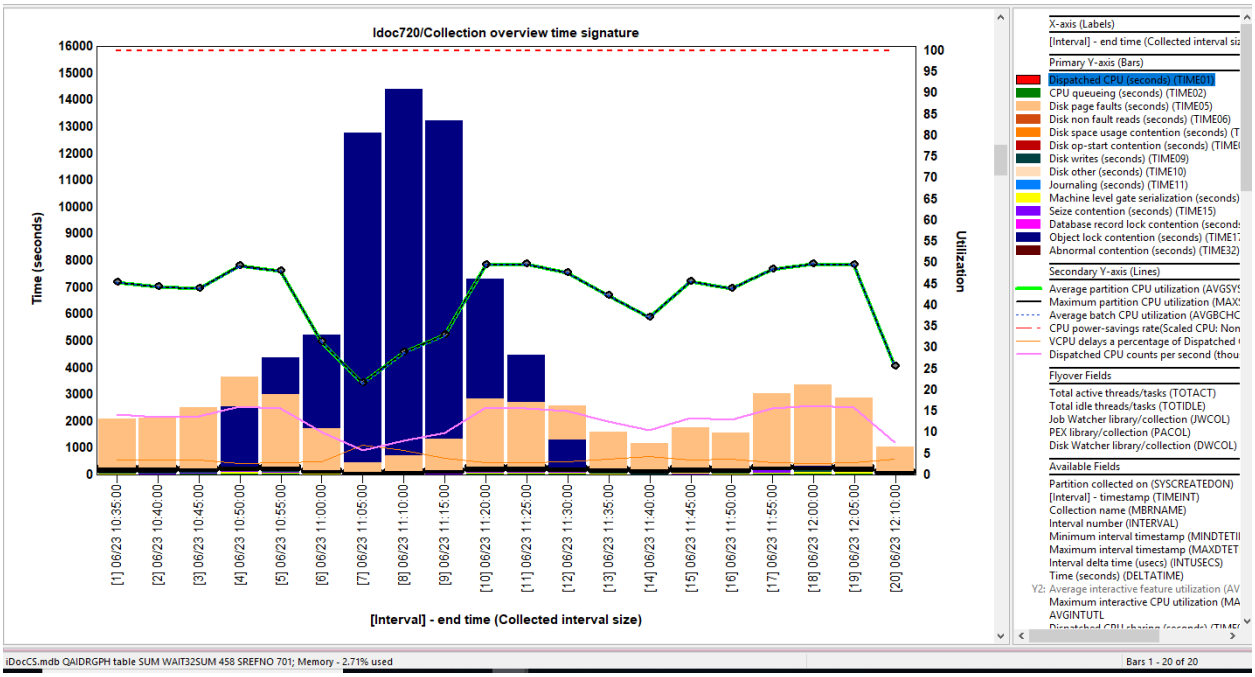
Vertical side-by-side graphs will show vertical bars, but each metric per X-axis point is shown side-by-side instead of stacked. This type of graph is best used for only 2-4 metrics on the Y1-axis.



Vertical side-by-side graph

6.9.4.7 Vertical bars overlapping

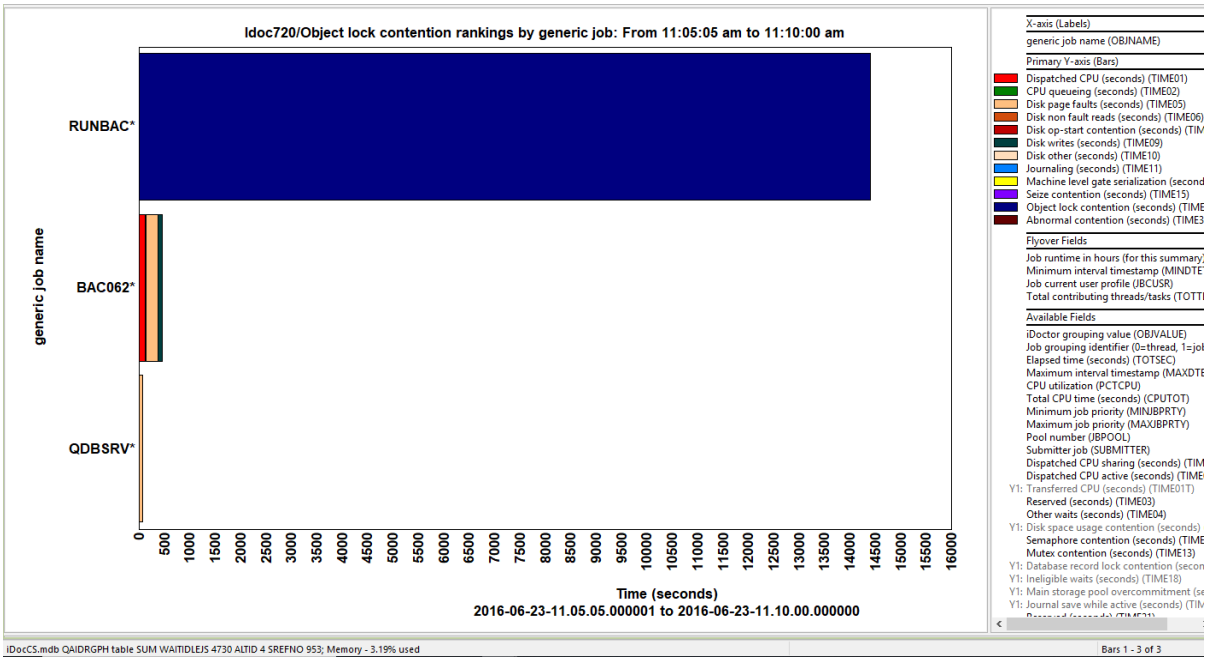
Vertical overlapping bar graphs will show you the maximum metric per interval at the top of each bar. The metrics overlap each other and only the highest values will appear at the top of each bar/interval.



Vertical overlapping graph

6.9.4.8 Horizontal bars stacked

This type of graph is typically used in iDoctor for Ranking graphs. This graph type does not include a Y2-axis but since the labels are drawn horizontally are usually easier to read when listing job names or object names.

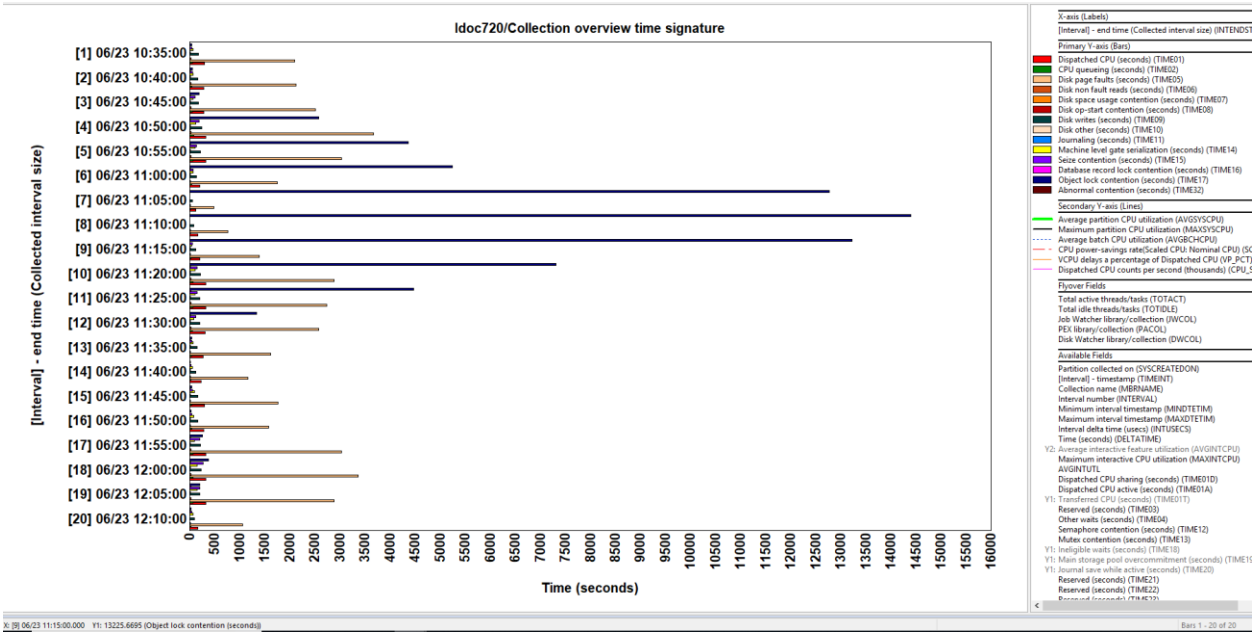


Horizontal bars stacked graphs

Tip: Graphs of this type allow the use of the Toggle Graph Format  button on the toolbar. Pressing this button will convert the graph to Vertical stacked bars instead.

6.9.4.9 Horizontal bars side-by-side

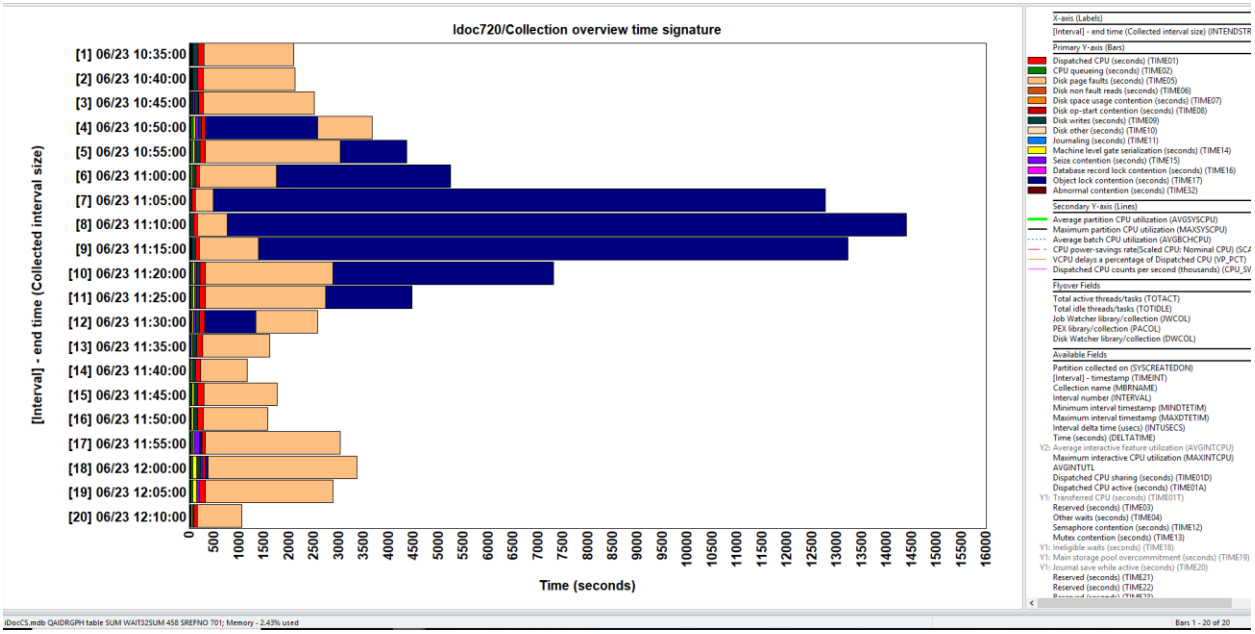
Horizontal side-by-side graphs will show horizontal bars, but each metric per point is shown side-by-side instead of stacked. This type of graph is best used for only 2-4 metrics on the Y1-axis.



Horizontal bars side-by-side

6.9.4.10 Horizontal bars overlapping

Horizontal overlapping bar graphs will show you the maximum metric per label at the end of each bar. The metrics overlap each other and only the highest values will appear at the end of each bar.



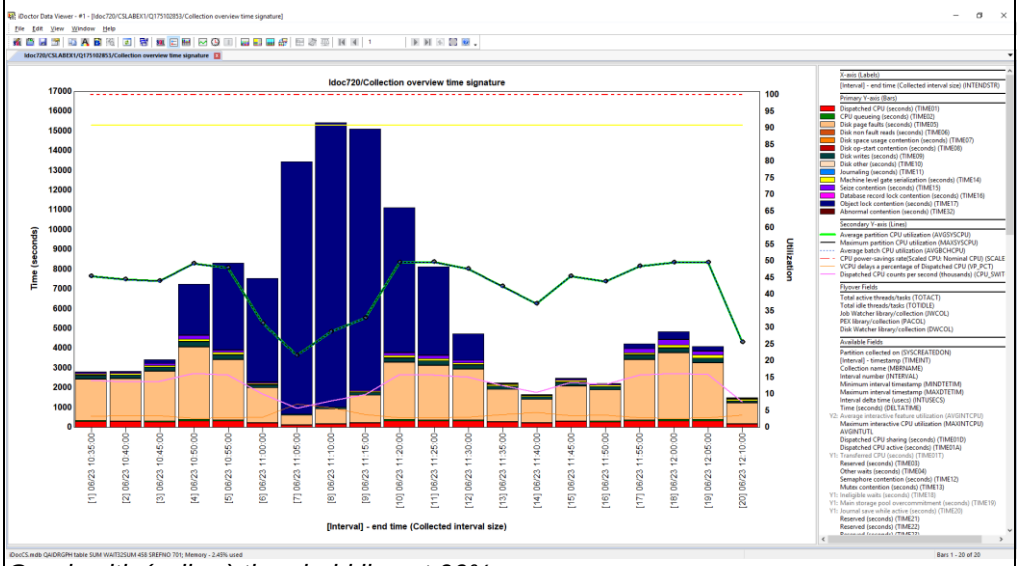
Horizontal bars overlapping

6.9.5 Graph Menu

iDoctor graphs typically offer at the top the most likely drill downs you may wish to use next. Depending on the component and the graph being used, you will see options like Rankings, Detail reports and more.

From an Overview (time-based) chart typically you will select an area of interest and drill down into a rankings chart next. This will rank jobs, or disk units, or whatever type of data you are graphing into logical grouping and sorts them.

Under these options you will find the following general features via the right-mouse click popup menu:

Popup Menu	Description
Preferences	Displays the Preferences window. With this interface the user can change the number of bars shown per page and customize font and other graph settings.
Choose collection or DB members	This will display an interface allowing you to change the current collection you are working with to another one. If you are not working with recognized iDoctor data, then you will be prompted to change DB physical file members instead if applicable.
Change SQL Parameters	Displays the Change SQL Parameters interface allowing you to modify settings in the SQL statement behind the graph. In several graphs in iDoctor parameters are provided to allow users to filter the data more easily.
Graph Definition -> Show legend	Displays the legend if it is not already visible.
Graph Definition -> Set graph type	Allows you to modify the graph type of the primary Y-axis. See the previous section for more information.
Graph Definition -> Set bars per page	Allows the user to quickly set the bars shown per page, or to reset this value back to the default.
Graph Definition -> Adjust primary scale	Allows the user to quickly adjust the primary axis maximum value based on a percentage of the current maximum scale value shown. Use the Reset option to set the scale based on the largest value in the graph.
Graph Definition -> Adjust secondary scale	Allows the user to quickly adjust the secondary axis maximum value based on a percentage of the current maximum scale value shown. Use the Reset option to set the scale based on the largest value in the graph
Graph Definition -> Set threshold	This option will draw a threshold line straight across a vertical bar graph at the desired percentage of the Y2-axis. For example, this graph shows a threshold line added at 90%.  *Graph with (yellow) threshold line at 90%*
Graph Definition -> General	Displays the graph definition interface's General tab.
Graph Definition -> X-axis	Displays the graph definition interface's X-axis tab.
Graph Definition -> Primary Y-axis	Displays the graph definition interface's Primary Y-axis tab.
Graph Definition -> Secondary Y-axis	Displays the graph definition interface's Secondary Y-axis tab.
Graph Definition -> Flyover	Displays the graph definition interface's Flyover tab.
Graph Definition -> Save As	Saves the graph definition.
Duplicate -> using	This option creates an SQL view in QTEMP over the current SQL statement and

a temporary SQL view	reopens a new graph instance using this new view. The view will vastly simplify the SQL statement behind the graph so that it becomes something like "SELECT * FROM QTEMP/VIEWNAME" **Note: <u>This option will only work if the SQL Statement contains SQL tables and not mult-member physical files.</u>**
Duplicate as Table	Produces a new table view based on the SQL Statement used to produce the graph.
Duplicate as Graph	Produces a duplicate copy of the graph into the desired Data Viewer. This lets you move a graph from one Data Viewer to another
Properties	Displays the properties for the currently selected point on the graph as well as other information such as the SQL statement behind the graph view.

Other popup menu items are shown depending on the type of data/analysis being viewed. These additional options are covered under the documentation for the appropriate collection type.

6.9.6 Legend

All graphs provide an optional legend. The legend view shows all the fields displayed on the graph and the fields defined in the SQL Statement that are available to be shown on the graph.

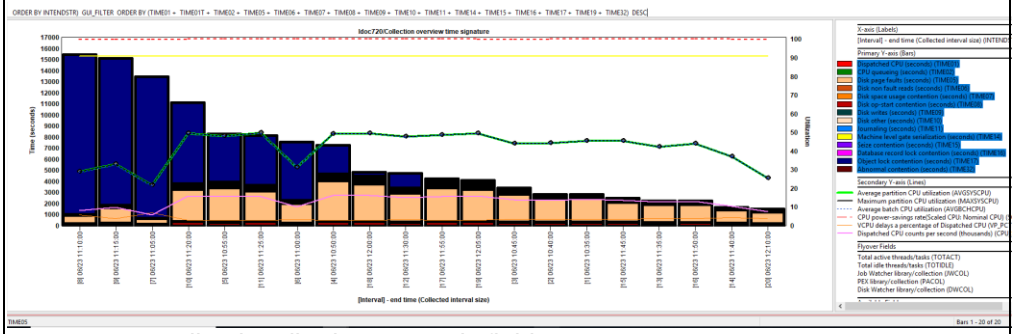
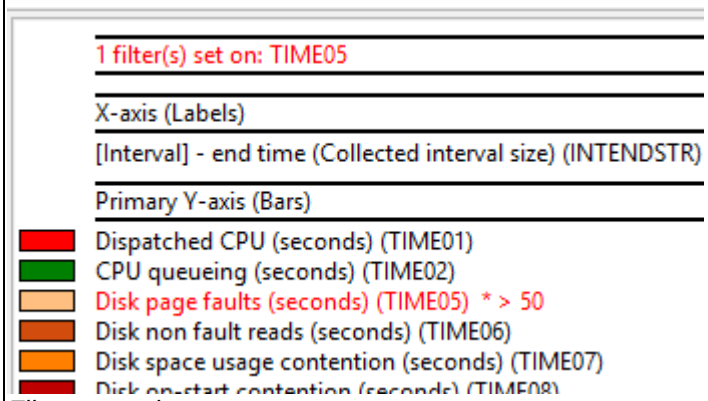
Tip: You can drag and drop fields in the legend to:

1. Move things to the X-axis label
2. Remove things from the graph
3. Move something to the flyover.
4. Rearrange the order of data shown on the graph.
5. Add fields from the available fields section to the graph.

The Legend View is broken up into the following sections, each representing a portion of the graph:

Interface Element	**Description**
X-Axis (Labels)	Contains the fields shown that make up the labels along the X-axis. Up to 3 fields may be used to construct the X-axis labels.
Primary Y-Axis	Identifies the list of fields and colors (and patterns) that make up the colors on the graph's Primary Y-axis. You can drag and drop fields in this section to rearrange their order shown on the graph.
Secondary Y-Axis (Lines)	Displays the list of fields and colors used for the secondary Y axis lines shown on the graph. These fields are only displayable for vertical graphs.
Flyover Fields	Displays the list of fields to be displayed as optional flyovers when placing the mouse over an area on the graph.
Available Fields	This section lists all fields that are not defined in any of the previous sections in the legend.

You can also manipulate the legend and graph using the Legend View's popup menu:

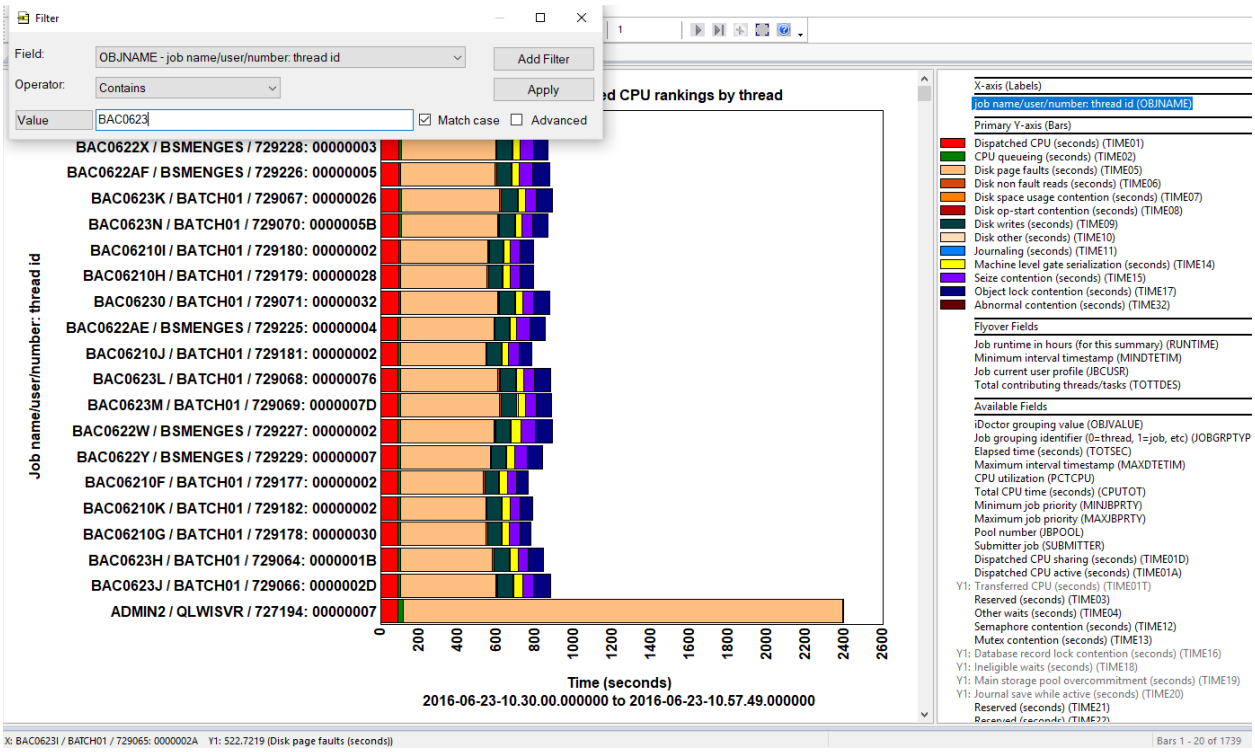
Popup Menu	Description
Alternate Views	If available, allows the user to quickly switch the graph to another IBM-supplied graph built from the current SQL statement.
Sort	Select 1 or more fields and use this option to rerun the SQL statement, changing the order of the data shown on the graph. **Tip:** If you select multiple fields then the values are added together before sorting. For example, this graph was sorted by ALL Primary Y-axis fields so the highest total for all buckets is shown first:  *Sort ->Descending by all primary Y-axis fields*
Edit…	This option shows the Edit Column interface which lets you modify a column's settings such as description and color. In iDoctor, a field's short name identifies it uniquely across all components.
Add Filter…	This lets you define a filter to add to the graph to reduce the data shown. **Tip:** If a filter is applied the graph will be updated and the filter value will be shown in the graph legend with a red font.  *Filter example*
Remove selected filter	If the currently selected field in the legend has a filter this option will remove it.
Remove ALL filters	This will remove all filters from the graph.
Set color	Allows the user to change the selected field's color using the Color Window.
Set pattern	Allows the user to change the pattern for the selected bar. These patterns are only visible if the Display Patterns option in the Preferences interface is enabled.
Set graph type	Allows the user to modify the graph type of the primary Y-axis.
Hide/show borders	This option allows you to remove or redisplay the thin border around every bar in the graph. Removing the border around a field in the bar graph can cause it to become more noticeable.
Add to X-axis label	Adds the selected field to the X-axis
Add to primary Y-axis	Adds the selected field to the Primary Y-axis

Add to secondary Y-axis	Adds the selected field to the Secondary Y-axis
Add to flyover	Adds the selected field to the Flyover.
Remove all except selected	Removes all fields from the current section of the graph you are working with except the selected one.
Remove from graph	Removes the selected field from the graph and adds it to the list of Available fields.
Hide legend	Hides the legend.
Search google for XYZ	This will open the default web browser and perform a Google search for the selected field description/name.

6.9.7 Filter

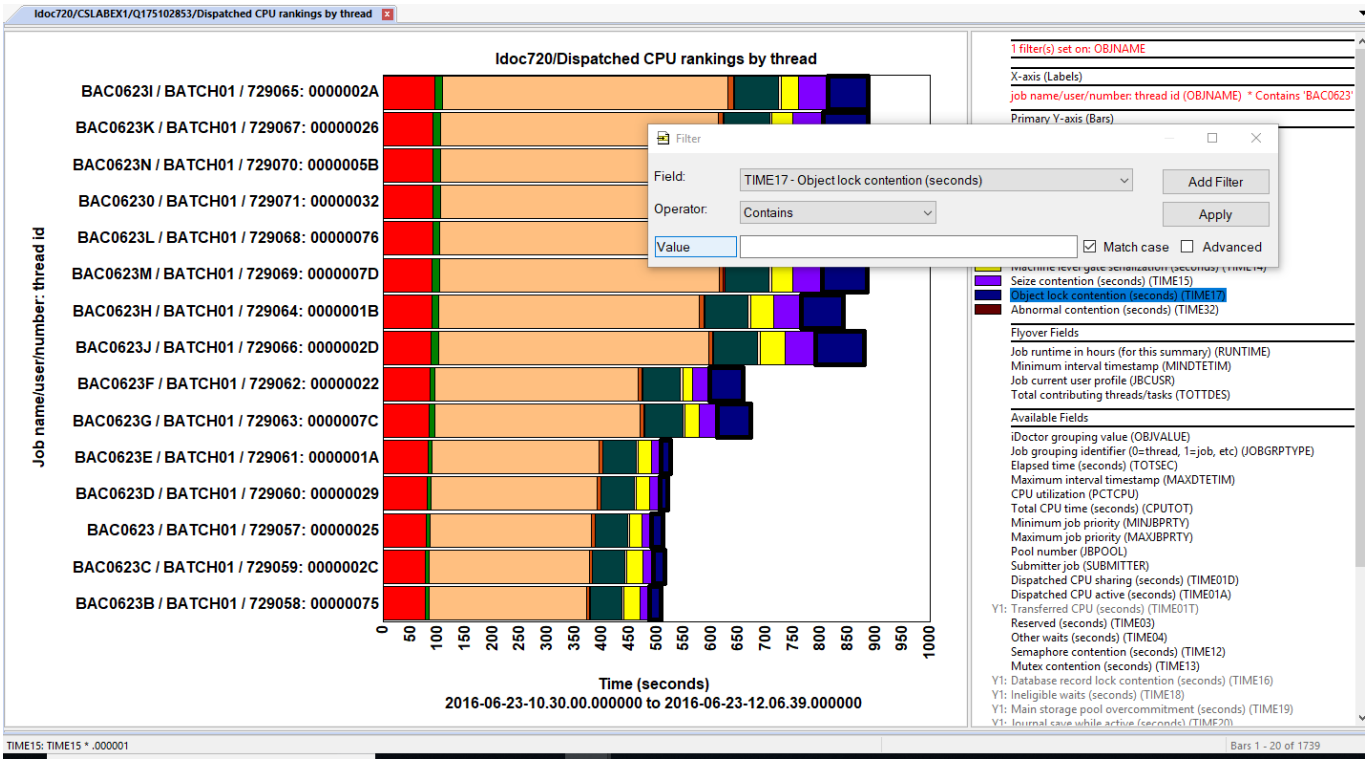
The Filter interface is accessed by right-clicking a field in the legend and using the Add Filter menu. Filters can be defined on one or more columns and each filter will modify the SQL statement (within the where clause) to perform the desired filtering. The text for fields that have filters applied are drawn with a red color.

The following shows an example of a graph with the Filter window open (but the filter is not yet applied):



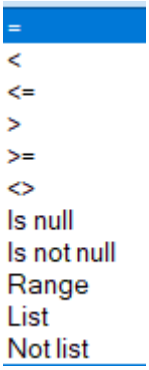
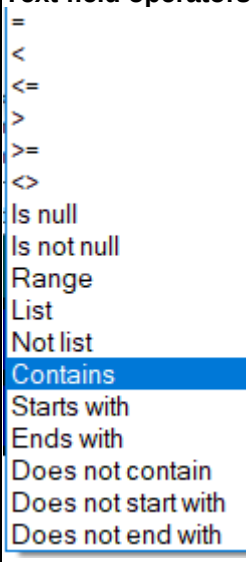
Graph with Filter window, Apply button not yet pressed

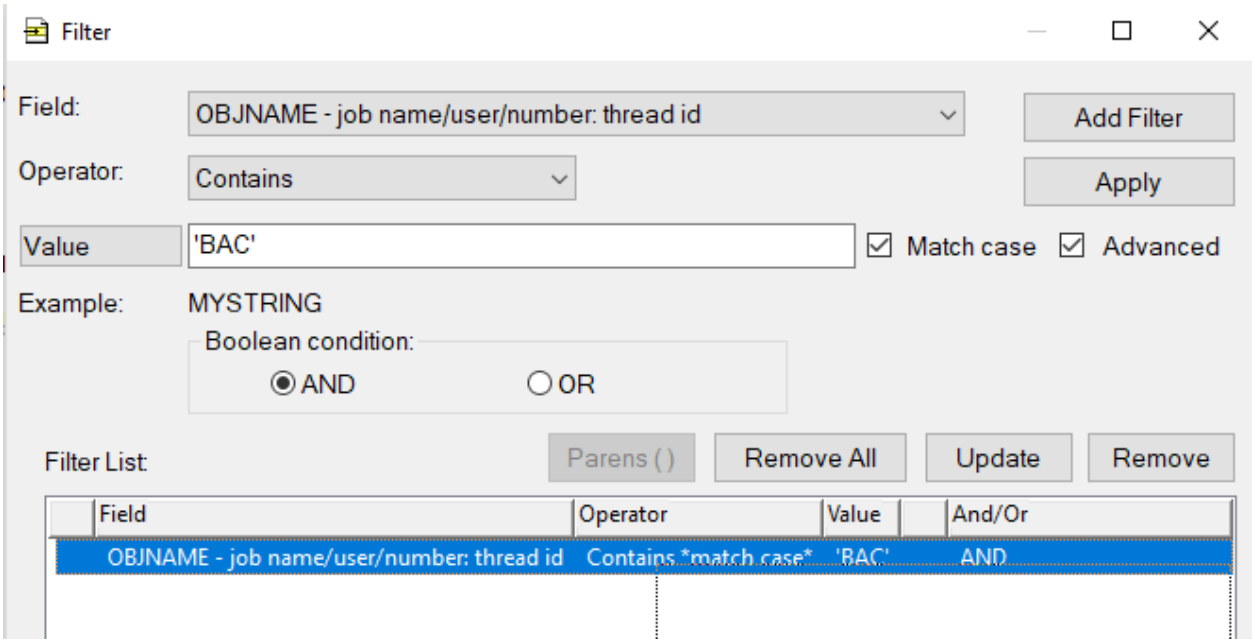
Pressing the Apply button reruns the query using the desired filter and updates the results.



Graph with Filter window, Apply button has been pressed

The following options are available:

Option	Description
Field	The field to filter the graph data on. By default, this is the same field that was right-clicked (or last clicked on from the legend.)
Operator	The type of operation to use for this filter. The possible operators available depends on the type of the field selected. **Numeric field operators:**  **Text field operators:**  Note: the = (equal) operator may not work for fields derived from the double function.
Value	This is the value to apply to the filter. Generally, the filter is something like FIELD OP VALUE where OP is the operator, FIELD is the field to filter on and VALUE is a constant numeric or text string.
Match case	This checkbox indicates whether a case-sensitive search will be performed.
Add/Update Filter	This button will add or update the desired filter in the graph. The change does not take effect until the Apply button is pressed or the graph is refreshed.
Apply	This button will rerun the query behind the graph and apply any changes made to filters defined.
Advanced checkbox	This checkbox will hide/display the additional options on this window. Generally, they are not needed unless you wish to define more advanced options.



Filter Window with Advanced options displayed

Option	Description
Boolean condition	When using multiple filters this allows you to define if you want ALL conditions true (use AND) or EITHER one (use OR). By default, all filters must be true and use the AND Boolean condition.
Parens ()	This let you place parentheses around the multiple filters selected. This allows you to control how the logic is performed and the order of operations to have special filtering like: (X = 1 AND Y = 2) OR Z = 3 This button will only be enabled if 2 or more fields in the list are selected.
Remove All	Removes all filters from the list. Hit the Apply button to rerun the SQL statement and update the graph
Update	This will update the selected filter in the list based on any changes in the fields above.
Remove	Removes the selected filter(s) from the list.
Filter list	The list of filters to apply to the current SQL statement.

6.9.7.1 SQL Statement Changes

When filters are used, the SQL statement is modified so that the existing SQL statement becomes a subselect of a new statement. For example the SQL behind the table shown in the previous section is:

```
SELECT * FROM (SELECT * FROM CRAVENS1/QAPYJWINTI ORDER BY INTERVAL)    GUI_FILTER
WHERE SYSTDECNT = 1072  ORDER BY INTERVAL
```

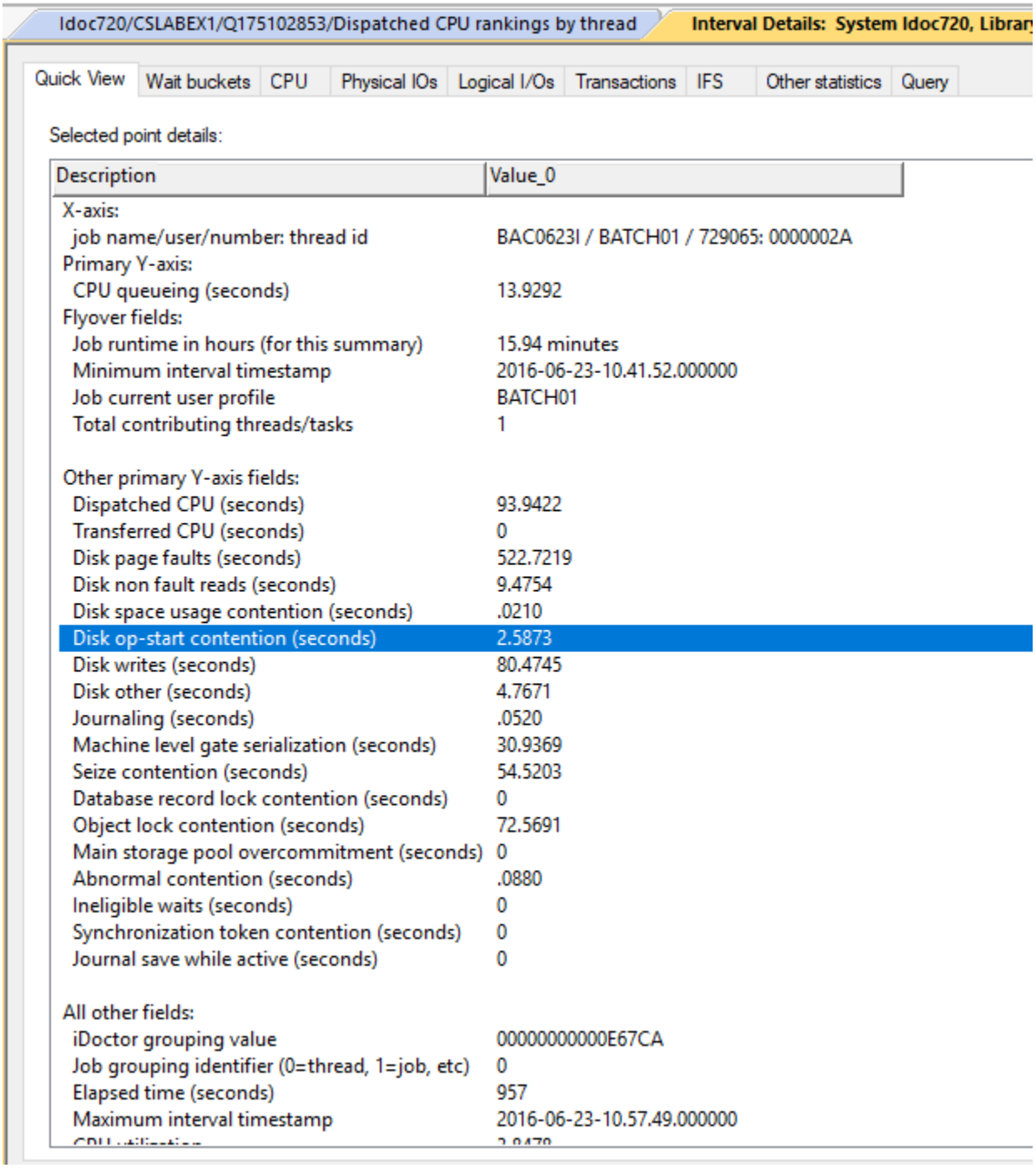
The identifier "GUI_FILTER" is used by the iDoctor GUI to indicate that this special type of filtering has been defined. Removing or changing this value will cause the filters to be no longer usable via the GUI.

6.9.8 Properties

This section describes the Properties interface for iDoctor graphs. Additional tabs are available in some components and are covered in the documentation for those components.

6.9.8.1 Quick View

The Quick View page contains all the information about a bar in the graph from the data retrieved by the SQL Statement. Access this screen by double-clicking on any bar in a graph view.

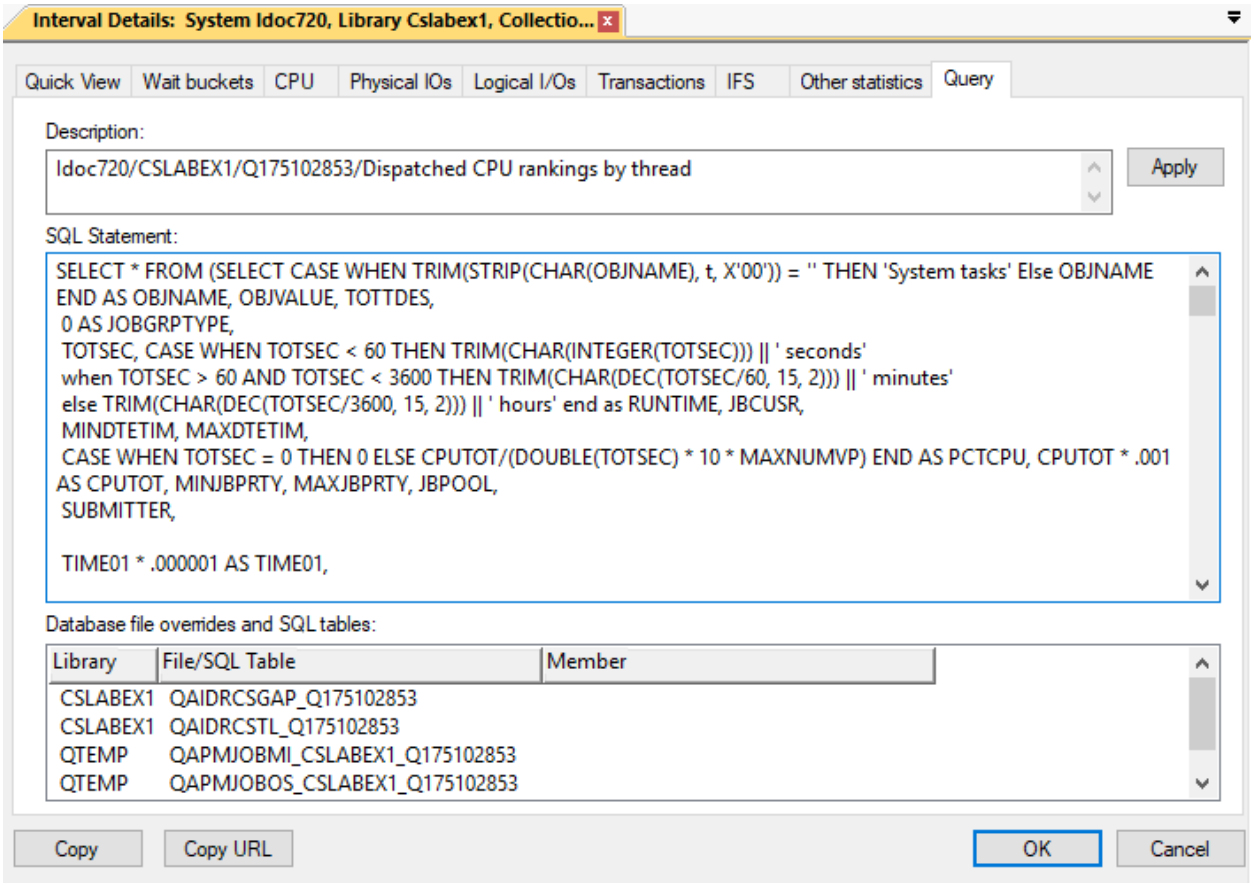


Graph Properties – Quick View Example

6.9.8.2 Query

The Query page of the Graph View -> Properties window displays the SQL statement used to produce the current graph view. This window also displays the title of the graph view and the overrides (or SQL tables/aliases) found in the SQL statement.

An example of this page is the following:



Graph View - Properties - Query Tab

Option	Description
Description	The text description identifying the graph. This is shown on the title bar of the view and can be changed. Update the view's title using the Apply button.
SQL Statement	The complete SQL statement for the graph.
Database file overrides and SQL tables	This list identifies all physical file/members or SQL tables or aliases in the SQL statement. When a physical file and member is listed an override is used to point to a specific member when executing the query. Aliases used by iDoctor are typically created in QTEMP. **Note:** If the SQL statement contains aliases or SQL tables, then the library and alias name will be listed, and the member field will be blank.

6.9.9 Graph Definitions

In iDoctor, users can define graphs over data generated by any SQL statement desired. Graphs are defined using a graph definition in iDoctor. Graph definitions supply the unique information that builds a

user-defined or iDoctor-supplied graph. Like query definitions, graph definitions are stored in a report database.

A graph definition defines everything needed to display the graph including the query definition (SQL statement.) Whenever a graph definition is saved, the query definition is also saved.

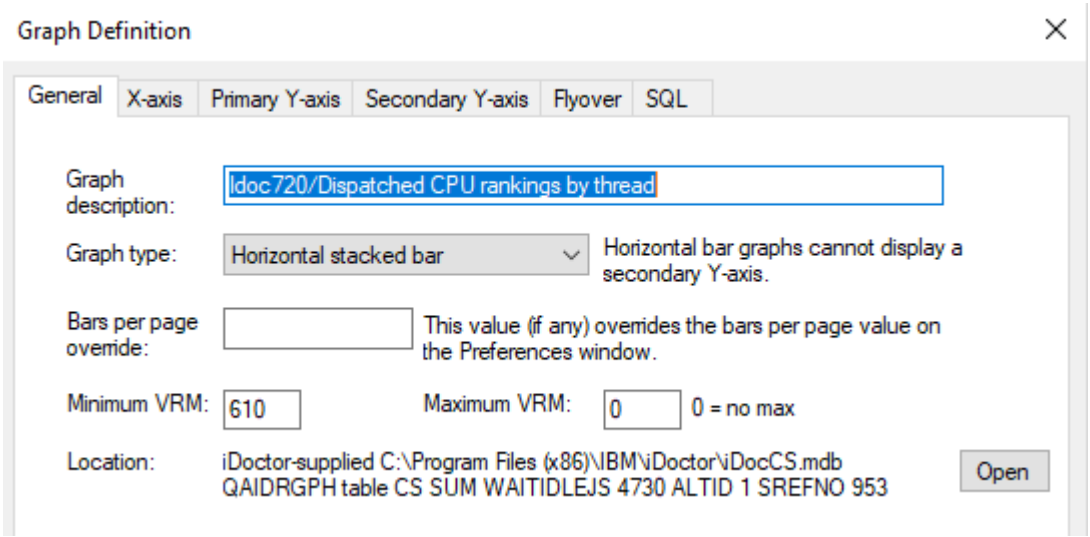
The menu to create a new graph is the **Graph Definition -> Define New...** option within a table view. This action will display the graph definition interface. You can also open the graph definition interface for a graph using the graph definition menu or by double-clicking the legend in a graph.

There are several panels in the interface that make up the graph definition. These pages are discussed in greater detail in the next sections. A summary of the pages that make up a graph definition is shown below:

Page Name	Description
General	Defines the general features of the graph, like the type of graph and the graph's title.
X-axis	Defines the field(s) to show on the X-axis as well as the text to separate them if desired.
Primary Y-axis	This page defines the fields, colors, patterns and descriptions to use for the bars in the graph. Up to 32 different fields/colors may be defined in the graph definition.
Secondary Y-axis	This page identifies the secondary Y-axis. This axis consists of multiple lines of the desired color and width. This axis can only be shown on vertical bar graphs.
Flyover	Lists the additional fields to show when the mouse is placed over a bar in the graph.
SQL	This tab contains the parameterized SQL statement for the graph. iDoctor uses many parameters which are replaced at run-time and these parameter markers are revealed with this view.

6.9.9.1 General

The general page lets the user define the graph description, the type of graph to display as well as the number of bars to show on the graph if this should differ from the value shown on the preferences interface.

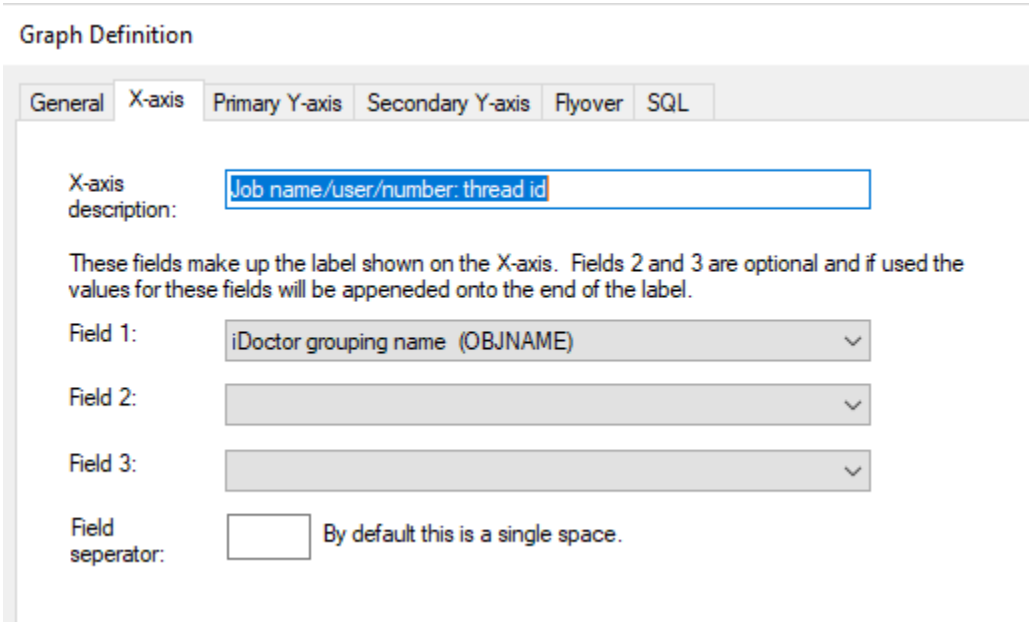


Graph Definition General Tab

Option	Description
Graph Description	A description of the graph that is displayed as the graph's title. (50 characters max)
Graph type	This identifies the type of graph and refers to the look of the Primary Y-axis specifically.
Bars per page override	This value can be used to optionally specify the number of bars to show per page on this graph. If a value is not specified on this page, then the applicable bars per page value on the Preferences window will be used instead.
Minimum VRM	This is the minimum VRM of the collection data that this graph will appear for. This is checked against the IBM i OS VRM for the system that created the data collection. **Note:** For iDoctor-supplied graphs, this value is read-only.
Maximum VRM	This is the maximum VRM of the collection data that this graph will appear for. This is checked against the IBM i OS VRM for the system that created the data collection. **Note:** For iDoctor-supplied graphs, this value is read-only.
Location	This identifies where the graph definition is located within the indicated report database.
Open button	This will open the report database where this graph definition is located. If the DB is located on the PC then this will only work if MS Access is installed.

6.9.9.2 X-axis

The X-axis page lets you define the fields to display as the X-axis label. Up to 3 fields may be used to make up the label and the text to separate each field may be specified here as well.



Graph Definition X-axis

Options	Description
X-Axis Description	The description to display under the X-Axis on the graph.
Field 1	The field to use as the 1st X-axis field on the graph.
Field 2	The optional field to use as the 2nd X-axis field on the graph.
Field 3	The optional field to use as the 3rd X-axis field on the graph.
Field separator	The text to separate the fields in the X-axis label if multiple fields are used. By default, this is a single space.

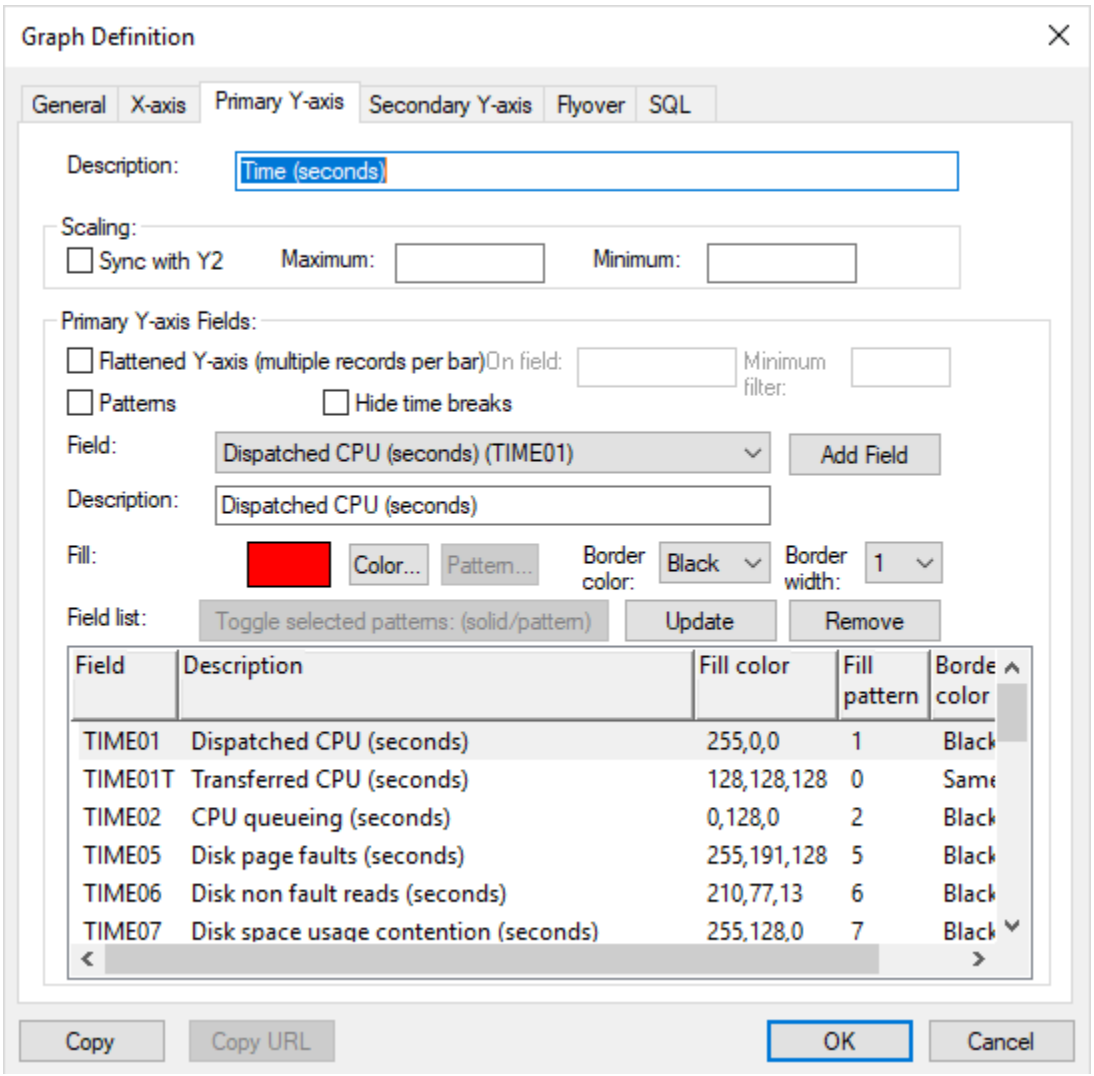
6.9.9.3 Primary Y-axis

Use the Primary Y-axis page to define the fields that should be displayed on Y1-axis for the graph. Each field has several options to customize how it appears on the graph.

Note: Keep in mind that in iDoctor if the graph type is a horizontal (bar) graph then the Y1-axis appears at the bottom of the graph and not the left.

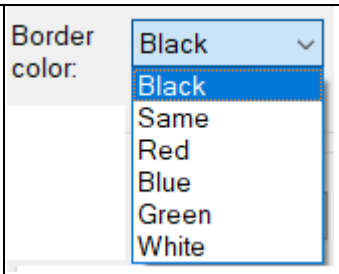
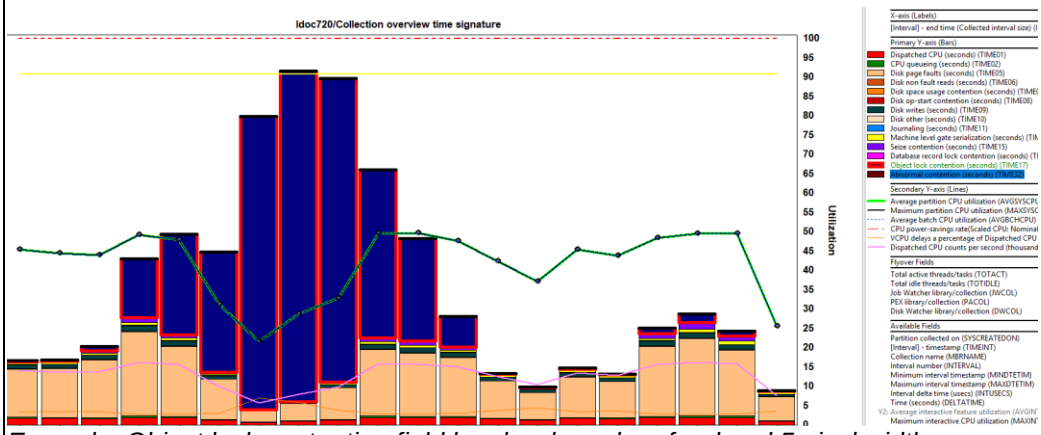
To Add a Field.

1. Select the field you wish to use for the new field from the Field drop-down list.
2. If desired, modify the description of the field from the field description.
3. If desired, define a color and pattern for this Y-Axis field. If this is not done, a color will be automatically assigned. The pattern only applies if the **Display -> Preferences -> Patterns** checkbox is checked.
4. Click the Add Field button to add the field to the list of fields.



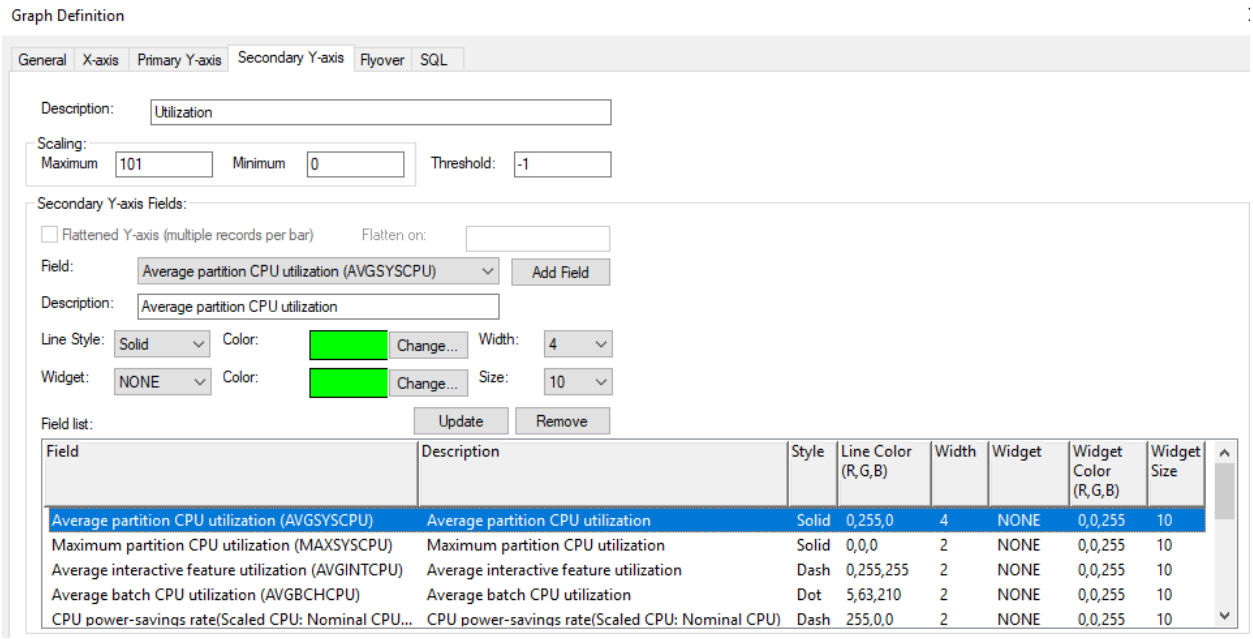
Graph Definition Primary Y-axis

Option	Description
Primary Y-axis Description	A description of the primary Y-axis. (50 characters max)
Scaling - Sync with Y2	This value indicates if the scaling (min/max values) of the primary Y-axis will be in sync with the secondary Y-axis or not.
Scaling - Maximum	This value (if not blank) will be the maximum value shown on the Y1 axis. **Note:** If values occur in the data beyond this value, then the graph data will be truncated.
Scaling - Minimum	This value (if not blank) will be the minimum value shown on the Y1 axis. **Note:** If values occur in the data beyond this value, then the graph data will be truncated.
Flattened Y-axis	If checked, the graph becomes a "flattened" graph. Flattened graphs are built such that each stacked bar is defined from multiple rows in the data. **Note:** Special SQL syntax is required for this to work properly. For an example, see the Memory pool graphs -> Flattened style in CSI. The SQL statement must contain a ROW_NUM field.
(Flattened-)On field	This value indicates the field the flattened graph is using to define the colors/values shown. The flattened-on field is used to determine the colors on the graph. For example, if this is a memory pool graph and the flattened-on field is POOL (the pool number), then pool 1 might be red, pool 2 green, etc.
(Flattened) Minimum Filter	This is used typically to filter the data in the graph by a maximum number of occurrences/colors. For example, if each color is a disk unit and the SQL is setup properly to rank the disk units, this could define the max disk units to include in the results. The SQL parameter used is <<STACKEDFILTER>>. If this parameter does not exist in the SQL statement, then modifying this value will have no effect. Example graphs in iDoctor that use this setting include: CSI – System graphs - LPAR CPU time.
Patterns	This indicates if patterns (hatchings) will be shown on the graph. It overrides the same setting in **Preferences –> Display -> Patterns**.
Hide time breaks	This setting allows the user to turn off the vertical dashed lines used to indicate breaks between different collections when graphing multiple collections.
Field	Allows selection of a field to add to (or modify in) in the field list. Changing the field will update the description to match the field description for the selected field.
Description	The 50-character description that identifies the data in the graph for the current field. This description will be displayed in the graph's legend.
Fill Color Button	Displays the Color window so the color to use for the field selected (in the list) can be modified. If no color is selected a color will be randomly assigned.
Fill Color Pattern Button	Shows the Fill Pattern Selection window that allows a user to modify the hatching (pattern) to use for the current field. These patterns will only be shown if the Display Patterns checkbox is checked or the Display Patterns option within the Preferences interface is checked.
Border color	If the graph is a bar graph, then this changes the color for the border. The default border width is 1 pixel. The possible values are:

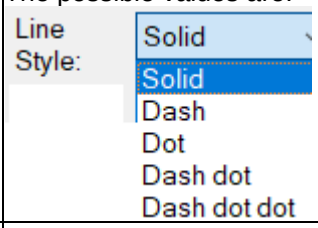
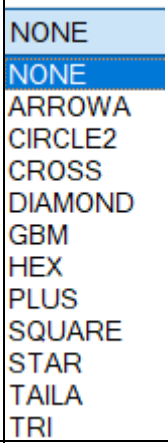
	 A value of “Same” means the color will match the fill color for the field (which in effect means no border will be visible. **Tip:** If the number of bars per page on the graph exceeds 250 then the border is automatically removed from the graph to avoid the graph appearance changing too much and looking more and more black with many bars.  *Example: Object lock contention field has border color of red and 5 pixel width*
Border width	If the graph is a bar graph, then this changes the width for the border. The default border width is 1 pixel.
Toggle selected patterns	This option will toggle the selected fields' pattern setting to solid fill or back to a pattern. Use this button to more easily construct a graph where only 1 or 2 fields use a pattern and the rest show a solid fill pattern.
Update	The update button is used to modify the selected field in the field list.
Remove	This option will remove the selected fields from the list.
Field list	Displays the field names, descriptions and colors and more to use for the fields on the Y1-axis of the graph.

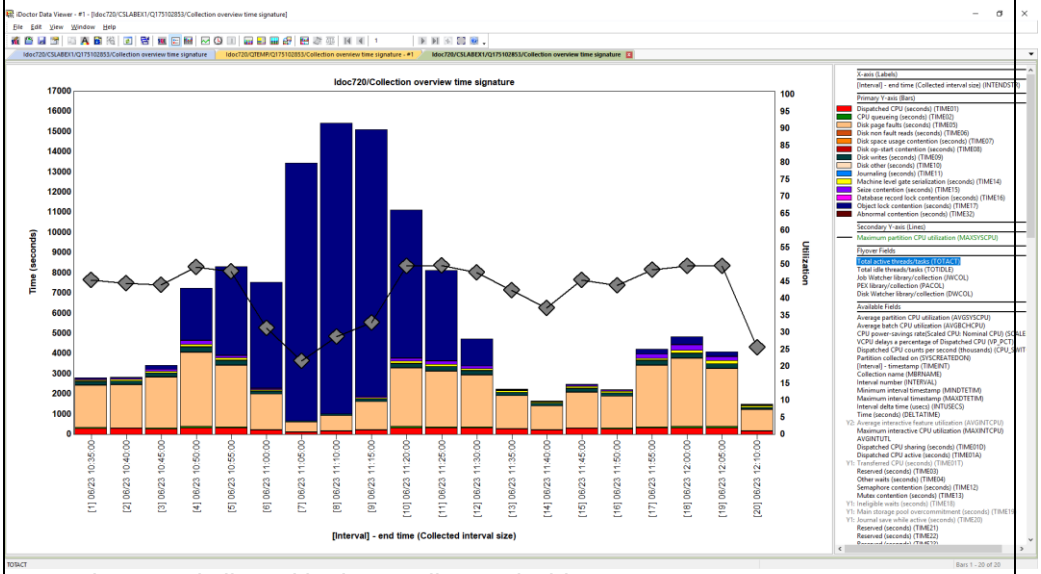
6.9.9.4 Secondary Y-axis

This page allows the user to define a secondary Y-axis consisting of 1 or more lines on a graph. This axis is NOT visible for horizontal graphs and only will appear for vertical graphs. It will always appear on the right-side of the graph.



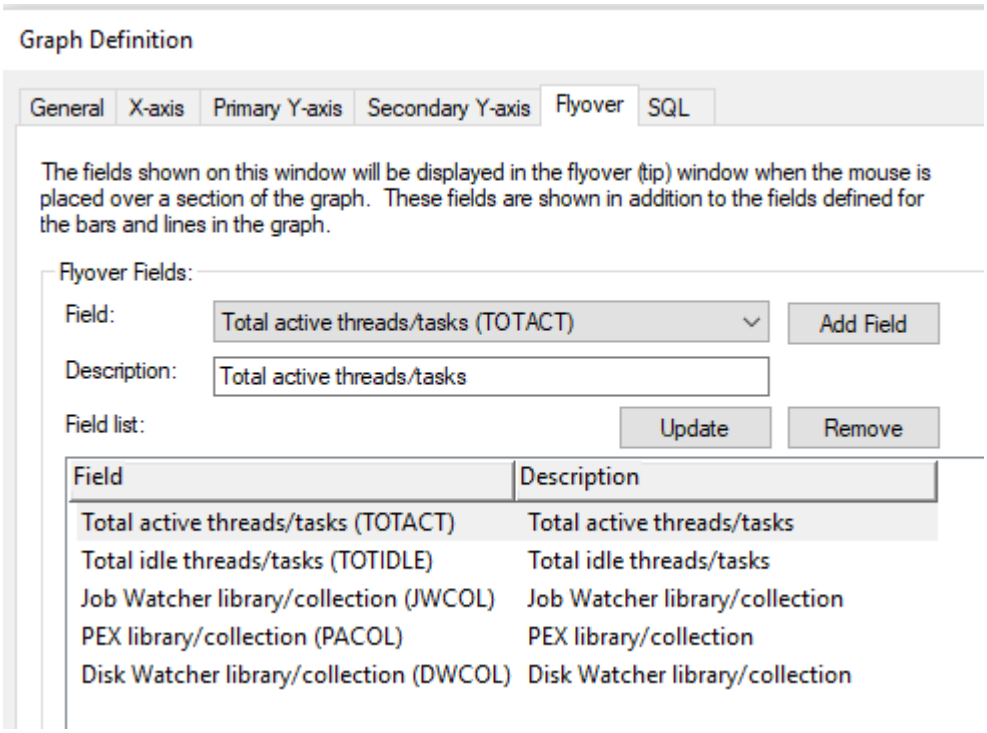
Graph Definition Secondary Y-axis

Option	Description
Description	The title to give the secondary Y-axis.
Scaling - Maximum	This value (if not blank) will be the maximum value shown on the axis.
Scaling - Minimum	This value (if not blank) will be the minimum value shown on the axis.
Scaling – Threshold	Specifying a value here (other than -1) will draw a yellow threshold line at the value given.
Flattened Y-axis	This value is always read on but included here as it does affect the Y2-axis. To change it you must do so from the Primary Y-axis tab. If checked, the graph becomes a "flattened" graph. Flattened graphs are built such that each stacked bar is defined from multiple rows in the data. **Note:** Special SQL syntax is required for this to work properly. For an example, see the Memory pool graphs -> Flattened style in CSI. The SQL statement must contain a ROW_NUM field.
Flatten on	This value is always read on but included here as it does affect the Y2-axis. To change it you must do so from the Primary Y-axis tab. This value indicates the field the flattened graph is using to define the colors/values shown. The flattened-on field is used to determine the colors on the graph. For example, if this is a memory pool graph and the flattened-on field is POOL (the pool number), then pool 1 might be red, pool 2 green, etc.
Field	The list of fields available to add to the secondary Y-axis.
Description	The description for the field to show on the legend to add to the secondary Y-axis. (50 characters max)
Line Style	This changes the style for the line. The possible values are: 
Line Color	The color to use for the current line.
Line width	The number of pixels wide to draw the current line.
Widget (name)	This allows each point of the line to contain an optional shape called a widget. These shapes have different names and the choices are: 
Widget color	This allows each point of the line to contain an optional shape called a widget. This is the color to draw the widget with.

	Press the Change button to change the color.
Widget size	This allows each point of the line to contain an optional shape called a widget. This is the size to draw the widget. The bigger the number, the bigger the widget will appear.  *Secondary Y-axis line with size 30 diamond widget*
Update	The update button is used to modify the selected field in the field list.
Remove	This option will remove the selected fields from the list.
Field list	Displays the field names, descriptions, colors and more to use for the fields on the Y2-axis of the graph.

6.9.9.5 Flyover

This page allows the user to define up to additional fields to show on the flyover window for the current graph. Flyovers are shown when the mouse is placed over a point of interest on the graph (like a bar or line).



Graph Definition - Flyover

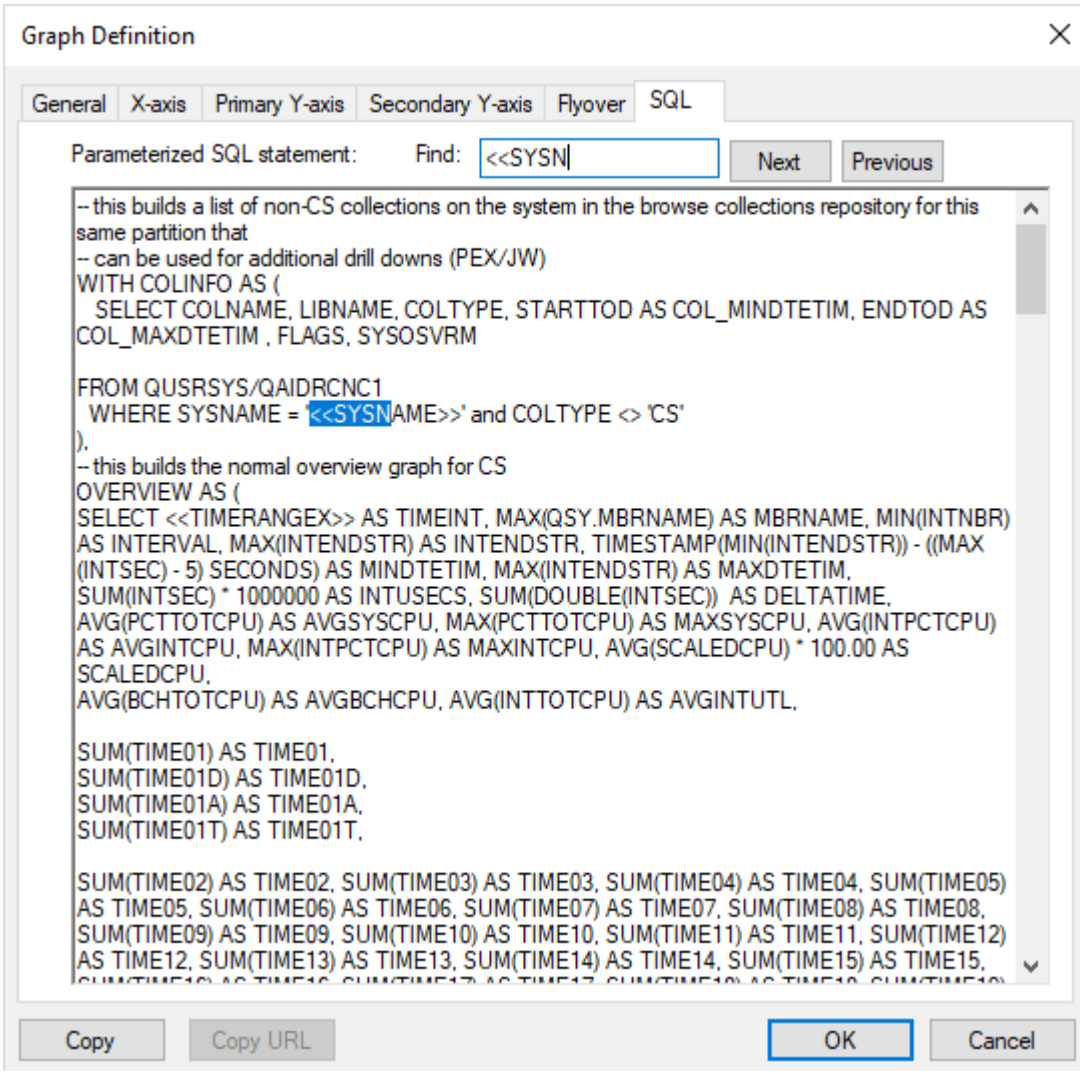
Option	Description
Field	The list of fields available to add to the flyover.
Description	The description for the flyover field to include. (50 characters max)
Update	The update button is used to modify the selected field in the field list.
Remove	This option will remove the selected fields from the list.
Field list	Displays the field names and descriptions for the flyovers on the graph.

6.9.9.6 SQL

This tab displays the parameterized version of the SQL statement behind the graph. iDoctor uses many parameters for iDoctor-supplied SQL statements and this interface will reveal them.

The window also features a text field where the desired input can be searched for within the statement.

Note: The SQL statement cannot be edited via this interface unless you are viewing the graph definition via the User-Defined Reports Database. It cannot be edited for IBM defined graphs.



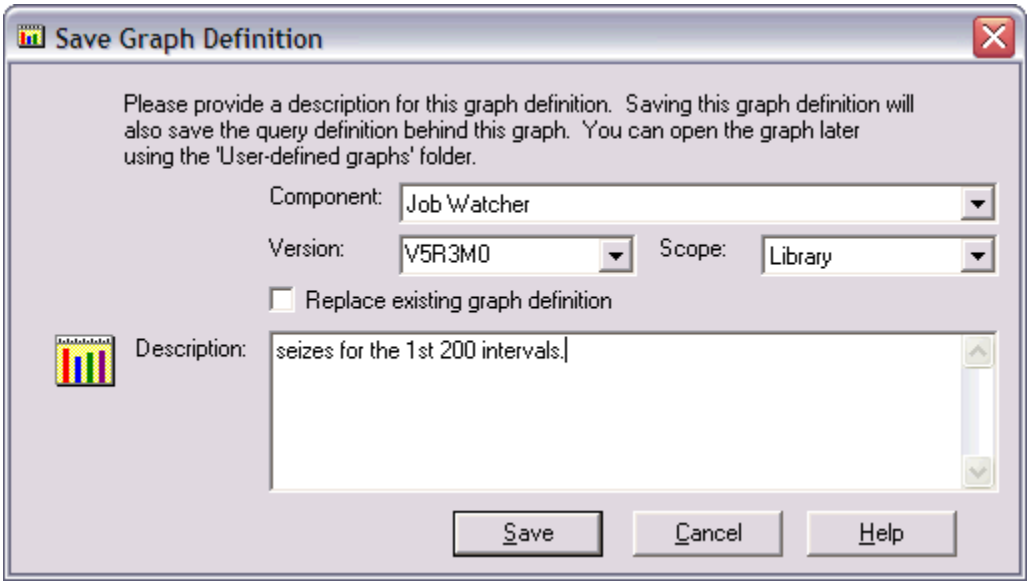
Graph Definition -> SQL

Option	Description
Find	Enter the desired text to find in this field and press the Next or Previous buttons to locate it within the SQL statement.
Next	Locates the next occurrence of the indicated text in the SQL statement. This is based on the current cursor position within the Parameterized SQL statement field.
Previous	Finds the previous occurrence of the indicated text in the SQL statement. This is based on the current cursor position within the Parameterized SQL statement field.
Parameterized SQL Statements	This is the version of the SQL statement stored in the iDoctor report databases or the User-Defined Report Databases. These SQL statements typically contain parameters.

6.9.9.7 Save Graph Definition (Save As...)

Graph Definitions are saved using the Graph Definition -> Save As... menu for the active graph view. All Graph Definitions are saved into the current local database on the client. This is configurable using the User-defined reports menu found by right clicking on the component icon in the tree/list views.

An example of the Save Graph Definition interface is shown below:



The interface elements within this window are described in more detail below:

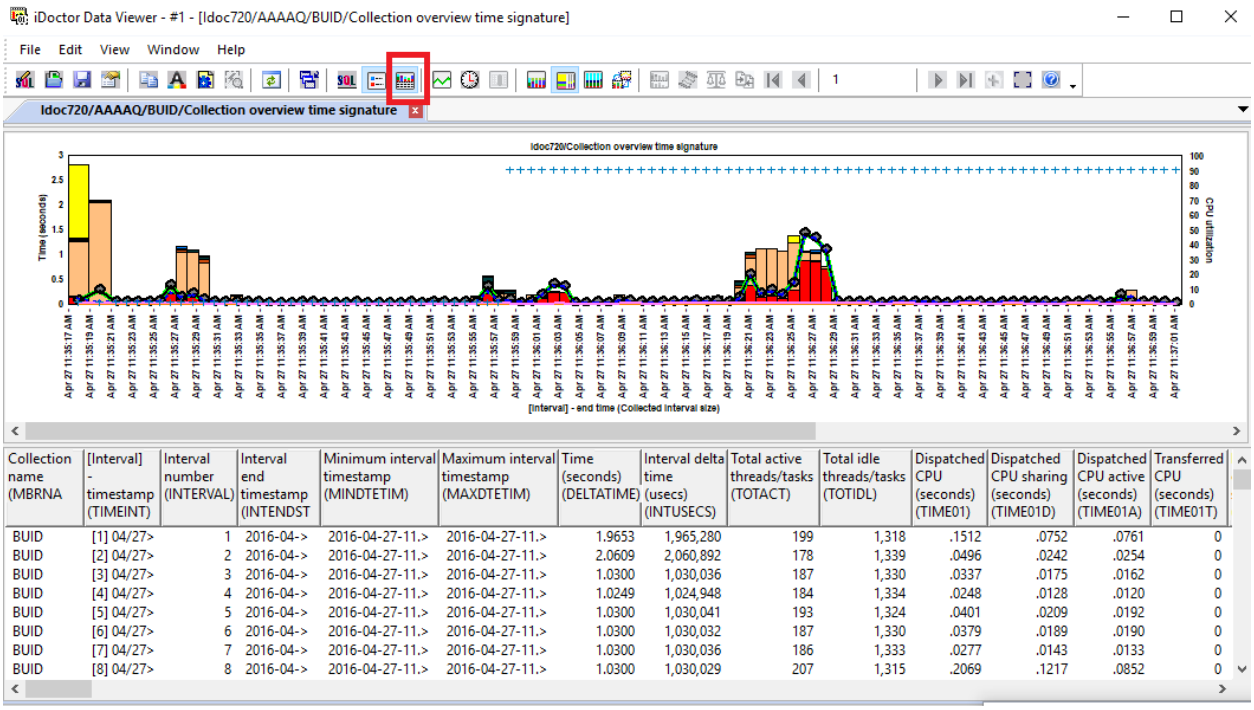
GUI element	Description
Component	The name of the component this graph should be visible in.
Version	The collection OS version the graph definition should be visible to. If this is set as V5R3, then this user-defined graph will not be visible under collections of a different OS release. You must save the graph multiple times, once for each desired release to accomplish this.
Scope	Use this option to set the scope of the query. This determines at which level (all systems, current system, current library, or current collection) the query should be visible.
Replace existing graph definition option	Check this box to replace the saved graph definition with the one currently being used. This checkbox is only visible if the graph view was created from a user-defined graph definition.
Description	The description for the Graph Definition. This description can be up to 250 characters long.

6.9.10 Synchronized Table View

Most graphs in iDoctor will have available beneath it a table view of data. This is the raw data behind the SQL statement used to build the graph. By default, this data is hidden from view unless you need it.

Press the  button on the Data Viewer toolbar while focus is on a graph view to open the table behind the graph. **Note:** This view is not available when using the ASP comparison functions.

This view works just like a regular table view except it is synchronized with the graph. Any scrolling and selections made are also done in the table and vice versa.



Collection name (MBRNA	[Interval] - timestamp (TIMEINT)	Interval number (INTERVAL)	Interval end timestamp (INTENDST	Minimum interval timestamp (MINDTETIM)	Maximum interval timestamp (MAXDTETIM)	Time (seconds) (DELTATIME)	Interval delta time (usecs) (INTUSECS)	Total active threads/tasks (TOTACT)	Total idle threads/tasks (TOTIDL)	Dispatched CPU (seconds) (TIME01)	Dispatched CPU sharing (seconds) (TIME01D)	Dispatched CPU active (seconds) (TIME01A)	Transferred CPU (seconds) (TIME01T)
BUID	[1] 04/27>	1	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.9653	1,965,280	199	1,318	.1512	.0752	.0761	0
BUID	[2] 04/27>	2	2016-04->	2016-04-27-11.>	2016-04-27-11.>	2.0609	2,060,892	178	1,339	.0496	.0242	.0254	0
BUID	[3] 04/27>	3	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.0300	1,030,036	187	1,330	.0337	.0175	.0162	0
BUID	[4] 04/27>	4	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.0249	1,024,948	184	1,334	.0248	.0128	.0120	0
BUID	[5] 04/27>	5	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.0300	1,030,041	193	1,324	.0401	.0209	.0192	0
BUID	[6] 04/27>	6	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.0300	1,030,032	187	1,330	.0379	.0189	.0190	0
BUID	[7] 04/27>	7	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.0300	1,030,036	186	1,333	.0277	.0143	.0133	0
BUID	[8] 04/27>	8	2016-04->	2016-04-27-11.>	2016-04-27-11.>	1.0300	1,030,029	207	1,315	.2069	.1217	.0852	0

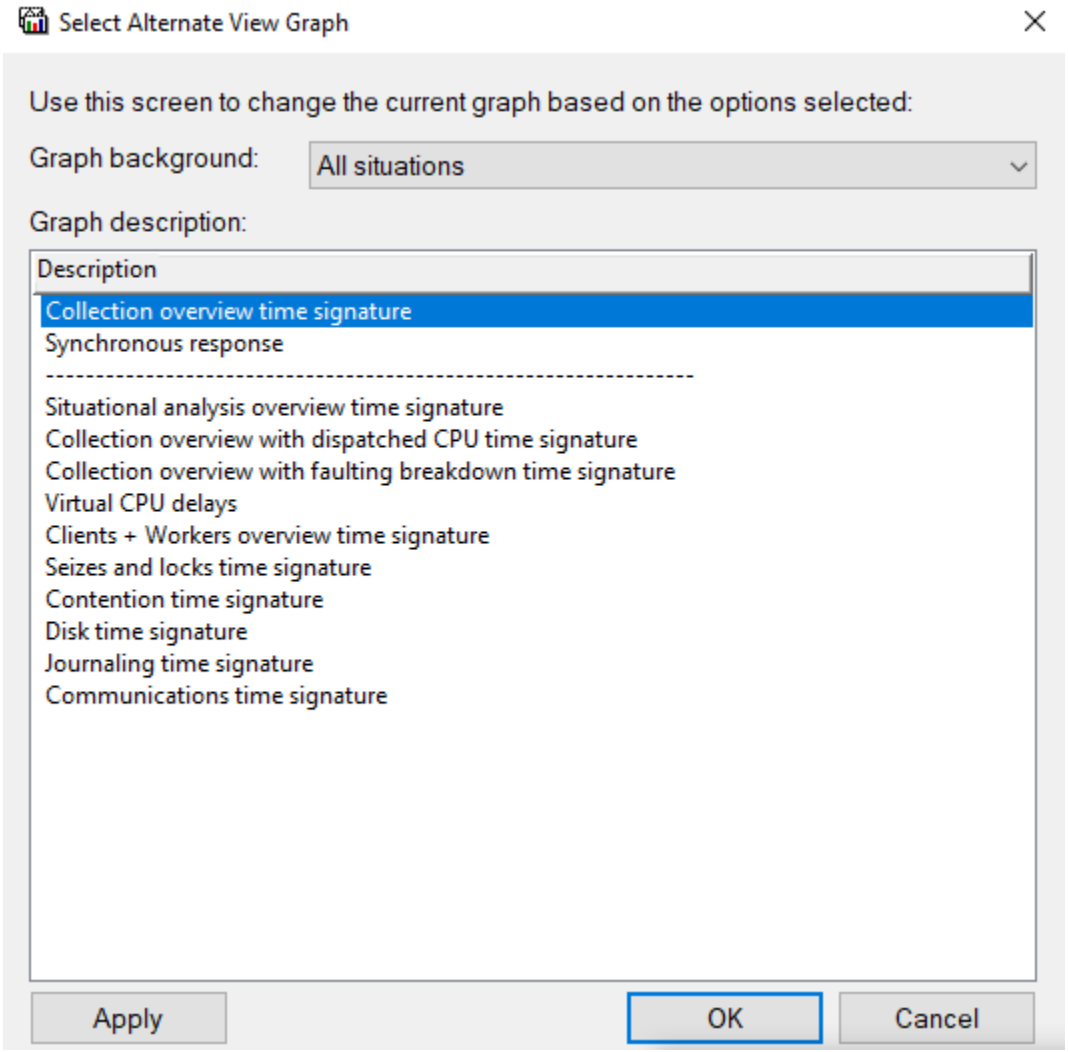
Synchronized Table View and Graph

6.9.11 Alternate Views

This feature allows a user to quickly change a graph and compare it with another one from the same data (SQL statement) being used. Press the  button to activate this option when looking at a graph.

Note: Many iDoctor graphs provide this capability. Only multiple graphs using the same SQL statement will provide this option.

Be aware that some alternate view graphs will require the SQL statement to be reran and some do not. If the SQL statement contains parameters that are different for the alternate view then the SQL query must be reran. “Flattened-style” graphs also must be ran again.



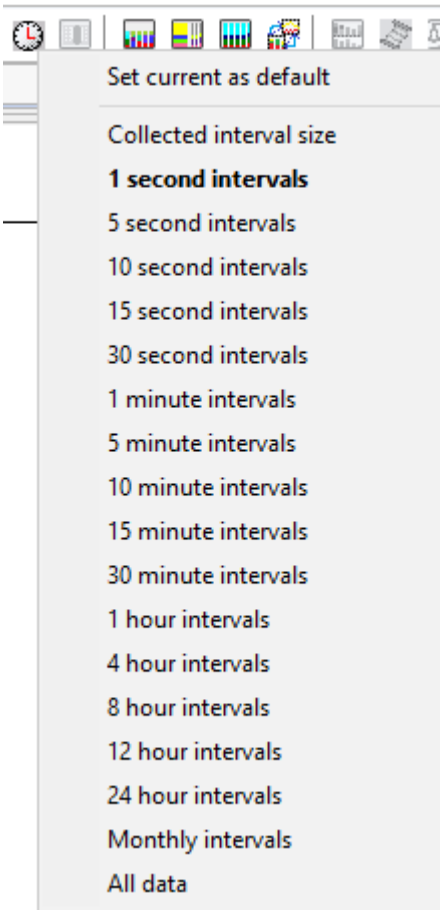
Select Alternate View Graph window

GUI element	Description
Graph background	This lets you customize the graph background to either be one of the following with regards to Situational Analysis (if it applies to the current graph) 1. All situations 2. No situations 3. 1 specific type of situation. All situation types are included in the list as well as the number of times each situation occurred in the collection within parentheses.
Graph description list	This is the available list of alternate view graphs. Typically, a user would pick one and then press the Apply button (Alt-A)
Apply	This will change the current graph based on the graph selected in the list. This window will remain open.
OK	This will change the current graph based on the graph selected in the list and then close this window.
Cancel	Closes the window without doing anything.

6.9.12 Time interval size (clock icon)

The clock icon on the Data Viewer toolbar allows a user to modify how a time-based graph is grouped. Many groupings are available and make it easier to get the big picture view of thousands of records of data more quickly rather than scrolling through it or trying to graph thousands of points on a single screen (often not feasible!)

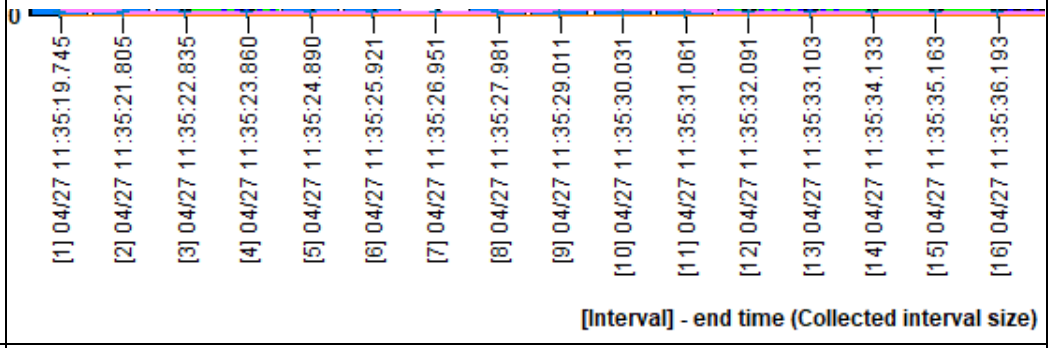
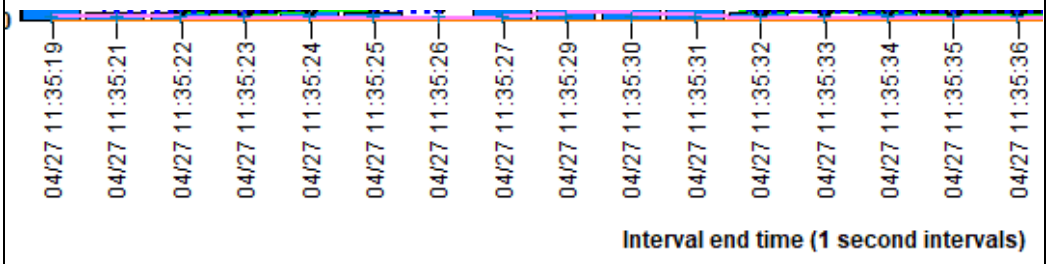
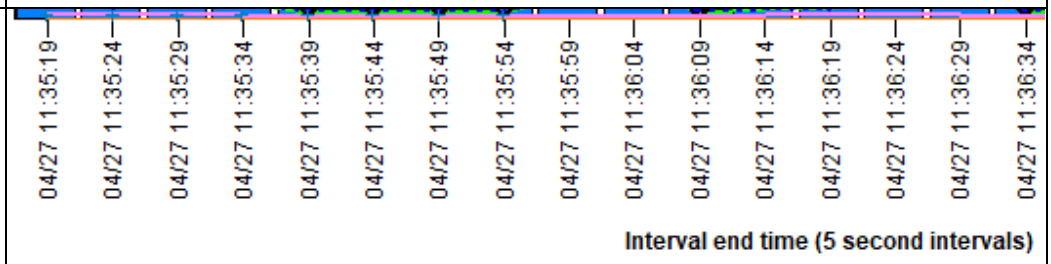
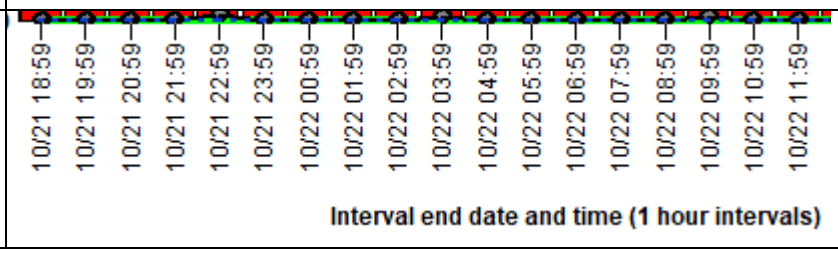
To enable this option, press this  button and a list of options will be shown. Picking the desired grouping will rerun the SQL statement using the desired time grouping.



Clock icon menu

Note: The number of options here will vary a bit based on the time grouping that was used to create the current collection. For example, if using Collection Services and the collection was created with 15 min time intervals, then no options between 1 second and 15 minutes would be shown.

Tip: Be aware that a date will only appear on the X-axis label in all cases if the **Preference -> Data Viewer -> Always show dates on graphs** is checked.

Option	Description
Set current as default	This will use the currently shown grouping on the graph and use this is a default preference for all iDoctor graphs
Collected interval size	This will graph the data at whatever grouping that was used to create the original data. This option will include the interval number in the X-axis label (unless variable width bar mode is enabled.) 
1 second intervals	Groups the data at 1 second intervals 
5 second intervals	
1-hour intervals	

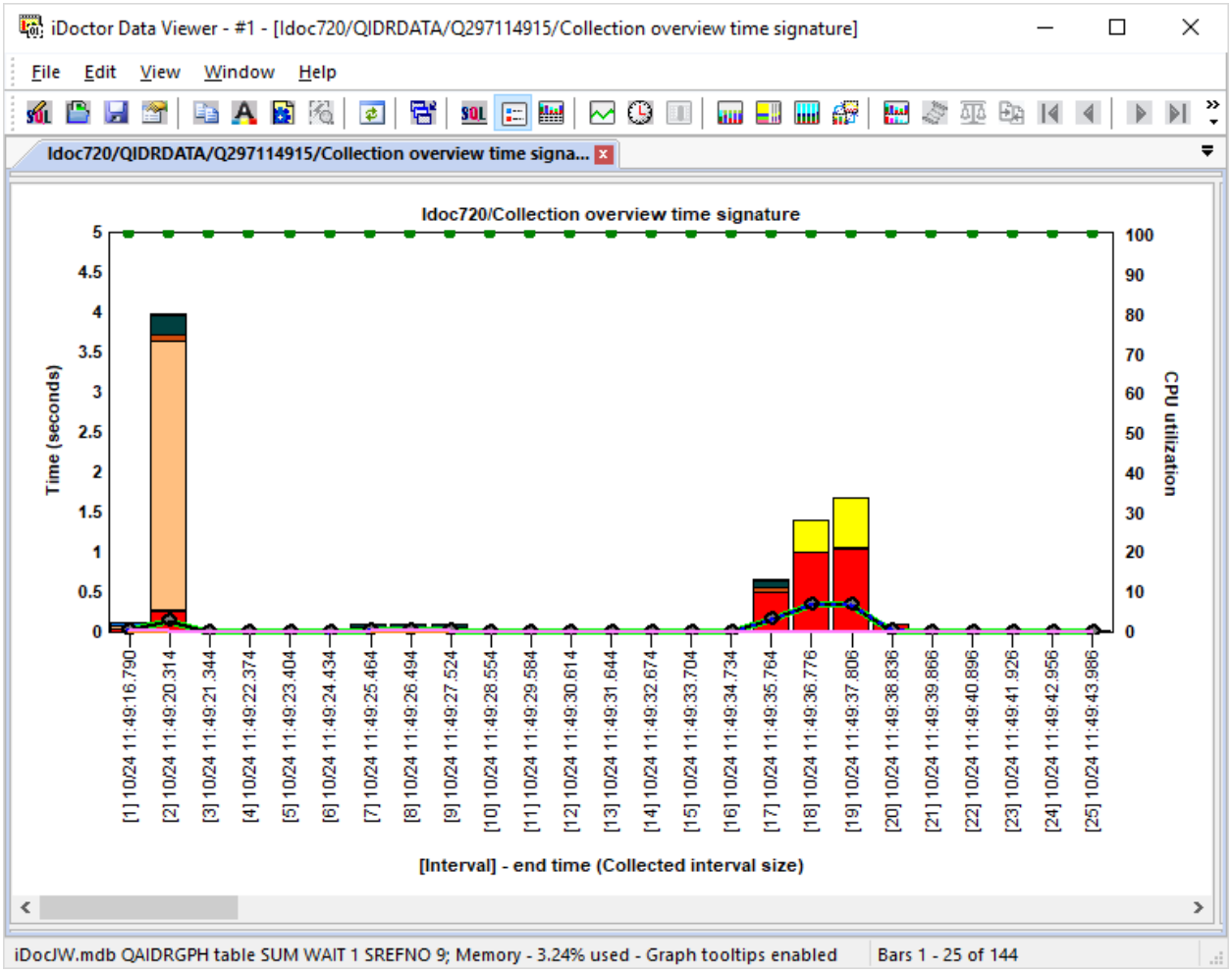
6.9.13 Normalize option

This option  can be useful if there are variations in the time taken to produce intervals shown on the graph. For example, if Job Watcher was slow to initialize and the 1st 2 intervals took 10 times longer than the rest of the intervals, the rest of the intervals may be barely visible unless this option is turned on.

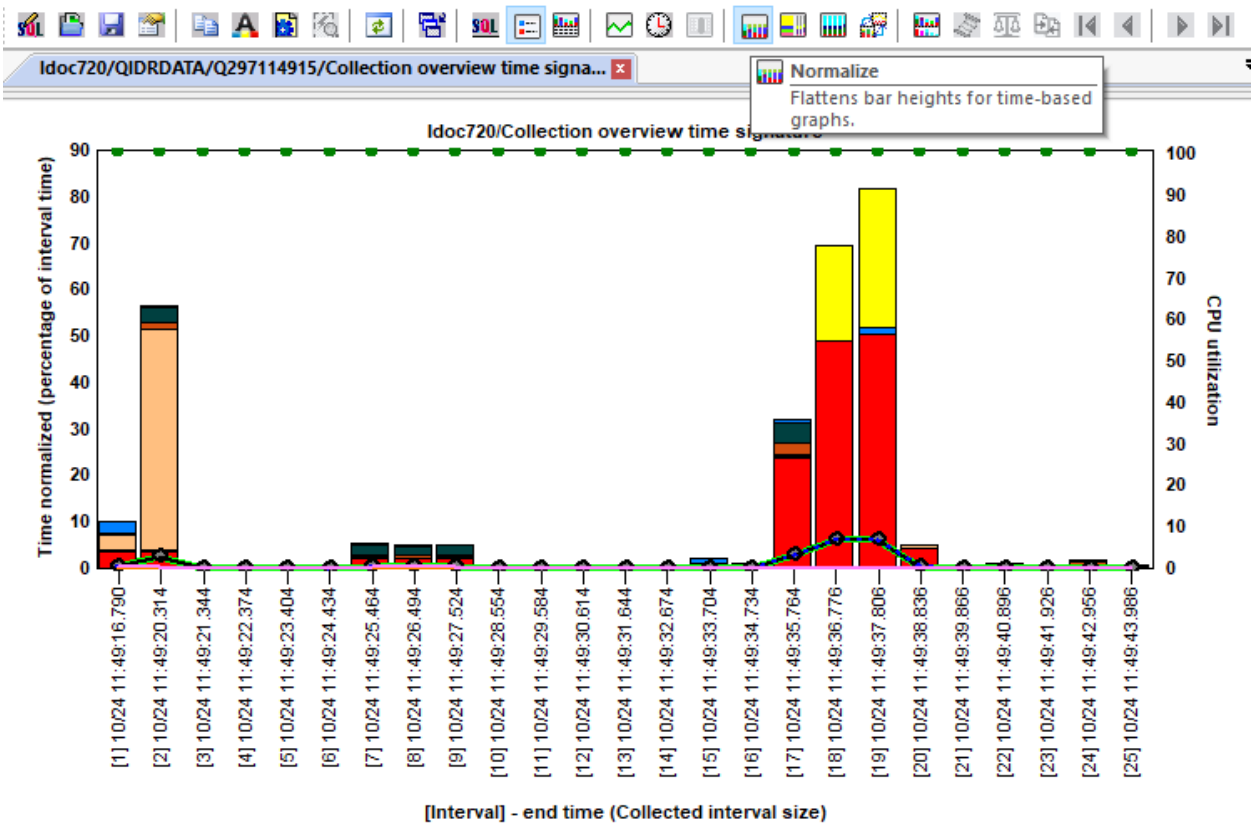
When normalizing a graph, each value on the primary Y axis, is divided by the interval's duration applicable to each Y axis value. This can create a "flattening" effect to bar heights by drawing time values

based on relative contributions. Keep in mind that the graph was changed to be percentage based on the amount of time each interval took and what you are seeing are not the true values collected.

The following example shows the difference between a graph with the normalize option NOT used and then with it applied. In this case the Job Watcher collection had a longer than normal 2nd interval because the option to collect all threads active or idle on interval 1 was enabled.



Collection overtime graph



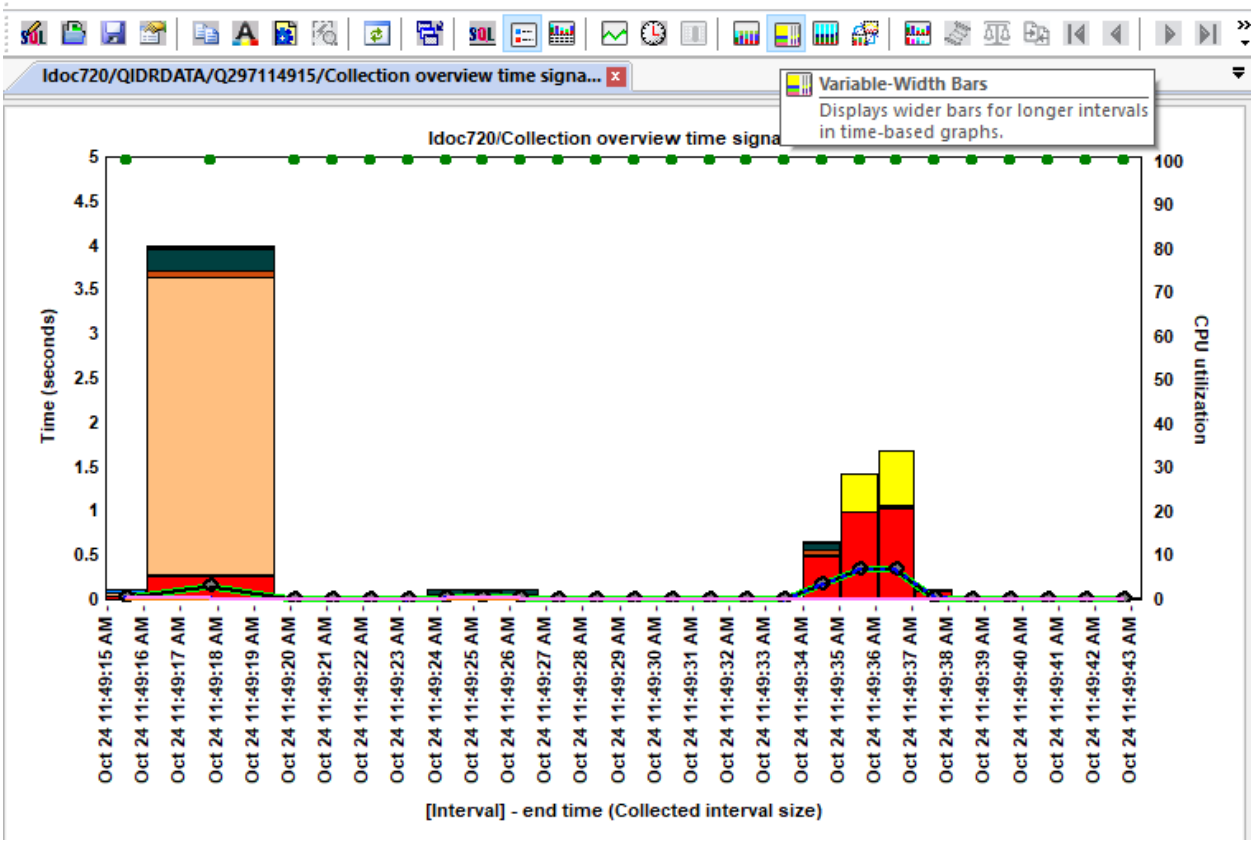
Collection overview graph with Normalize option

6.9.14 Variable-width bar mode option

This option  can also be useful if there are variations in the time taken to produce intervals shown on the graph. For example, if Job Watcher was slow to initialize and the 1st 2 intervals took 10 times longer than the rest of the intervals, then this fact may not be obvious unless this option is turned on.

This graphing technique will change the graph so the amount of time each interval took is visually graphed when normally they are not. Longer intervals will have wider bars and short intervals will have thinner bars. This mode also changes the X-axis labeling so that each bar is not labeled. Instead the true time periods are graphed where they occurred.

In the following example the 2nd interval was 4 times longer than the others and can be easily seen with variable-width bar mode enabled.



Collection overview graph with variable-width bar mode enabled

6.9.15 Toggle graph format

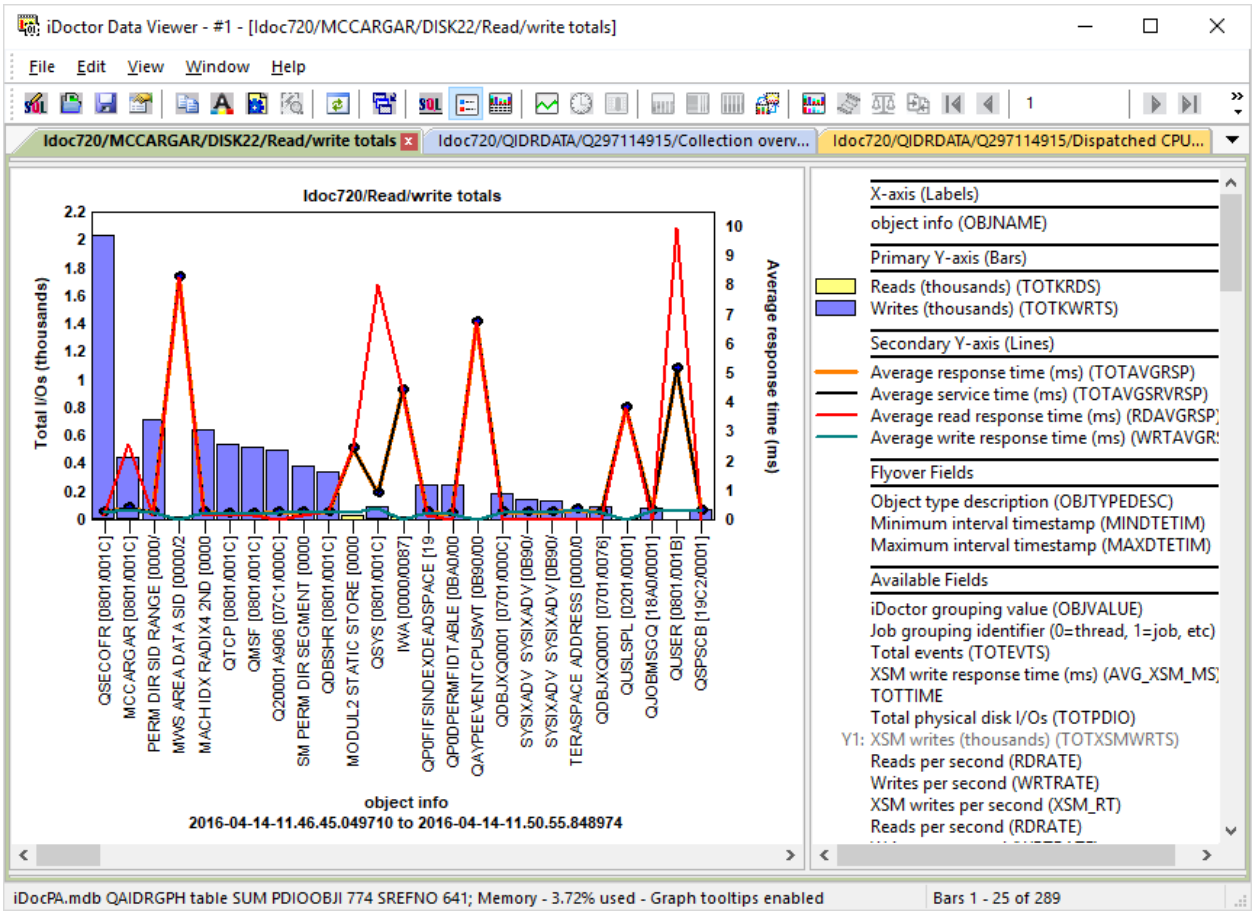
This button on the toolbar allows a graph to be quickly modified and presented in a different way. There are two types of graphs that have this ability available:

1) Rankings graphs – toggles from horizontal to vertical bars.
2) Time-based overview graphs – toggles from stacked vertical bars to lines

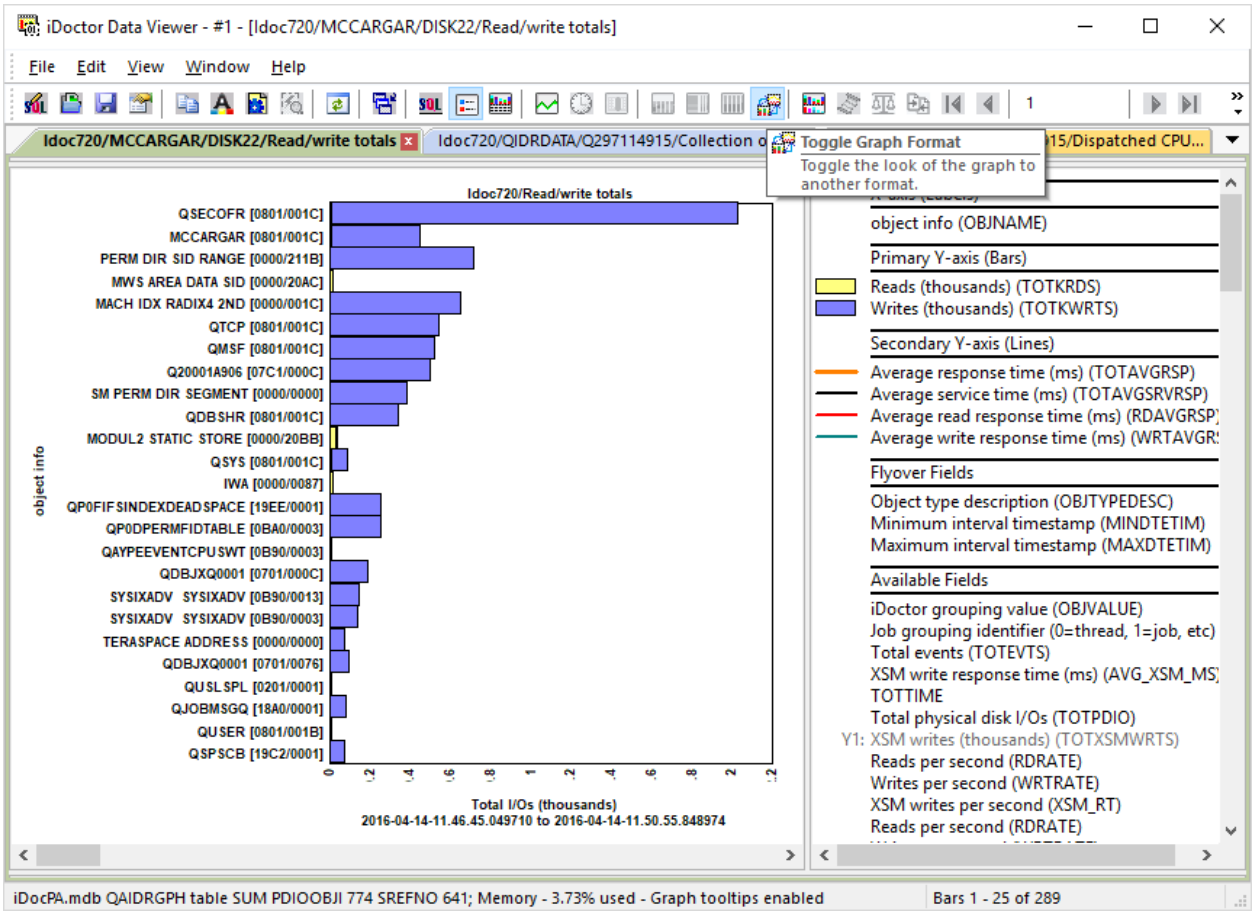
Please note that using this option is like setting a preference. iDoctor will remember this the next time this graph or any other graph in the same folder is opened.

In this example from PEX Analyzer, a PDIO rankings graph is toggled from vertical to horizontal bars.

Note: This removes the Y2-axis and the avg response time metric but does make the label easier to read.



PEX – PDIO -> by object info -> Read/write totals

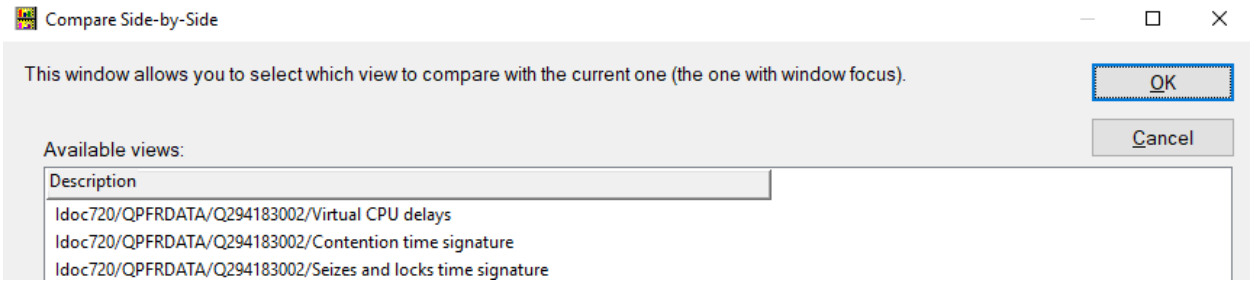


PEX – PDIO -> by object info -> Read/write totals with Toggle graph format enabled

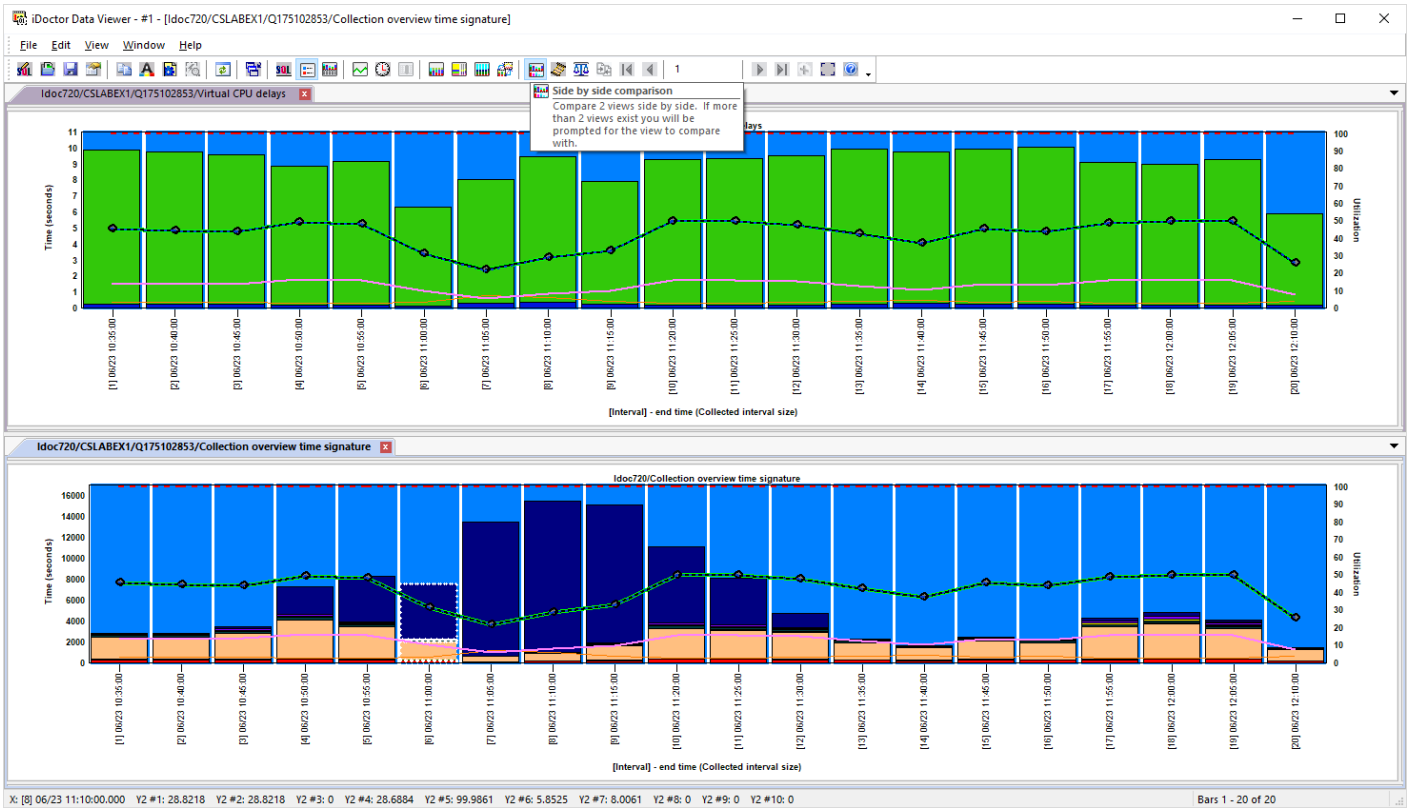
6.9.16 Side-by-side comparison mode

This option is enabled by pressing this button  when 2 or more graphs or reports exist in a Data Viewer. If only 2 graphs exist in the Data Viewer, then the view will be split so 1 graph is on the top and 1 is on the bottom so they can be compared.

If more than 2 graphs exist in the Data Viewer when pressing this button, then you will be prompted for the other graph you wish to compare the current one with.



Compare Side-by-Side window



Side-by-side comparison mode

After enabling this graph mode additional toolbar buttons will be enabled on the Data Viewer toolbar providing additional options for comparing the graphs:

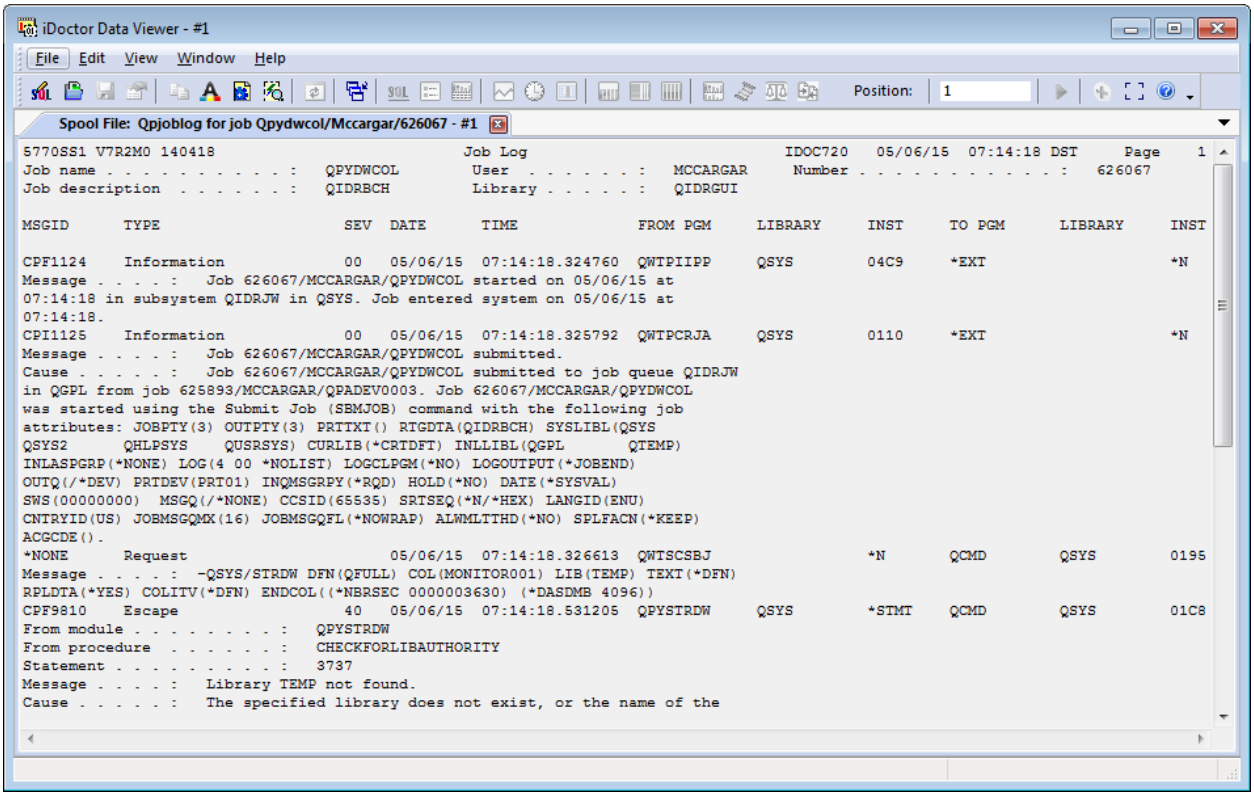
Button	Description
	Use this option to synchronize the scrolling of data while in Side-by-Side Comparison Mode. If pressed, then both graphs will scroll together when one of them is scrolled.
	Use this option to synchronize the Primary Y-axis scaling while in Side-by-Side Comparison Mode. If pressed, then both graphs will use the same min/max values on the primary Y-axis.

6.10 Spool File Views

The Data Viewer can be used to display the contents of spool files on the server. Whenever a job log for a collection is viewed that has already ended the job log is displayed in this view.

The spool file viewer will read in the entire contents of the spool file into the viewer. Although this will cause delays when reading large files this allows the user to more quickly perform a text search using the Find feature on the toolbar after the data is loaded into the client.

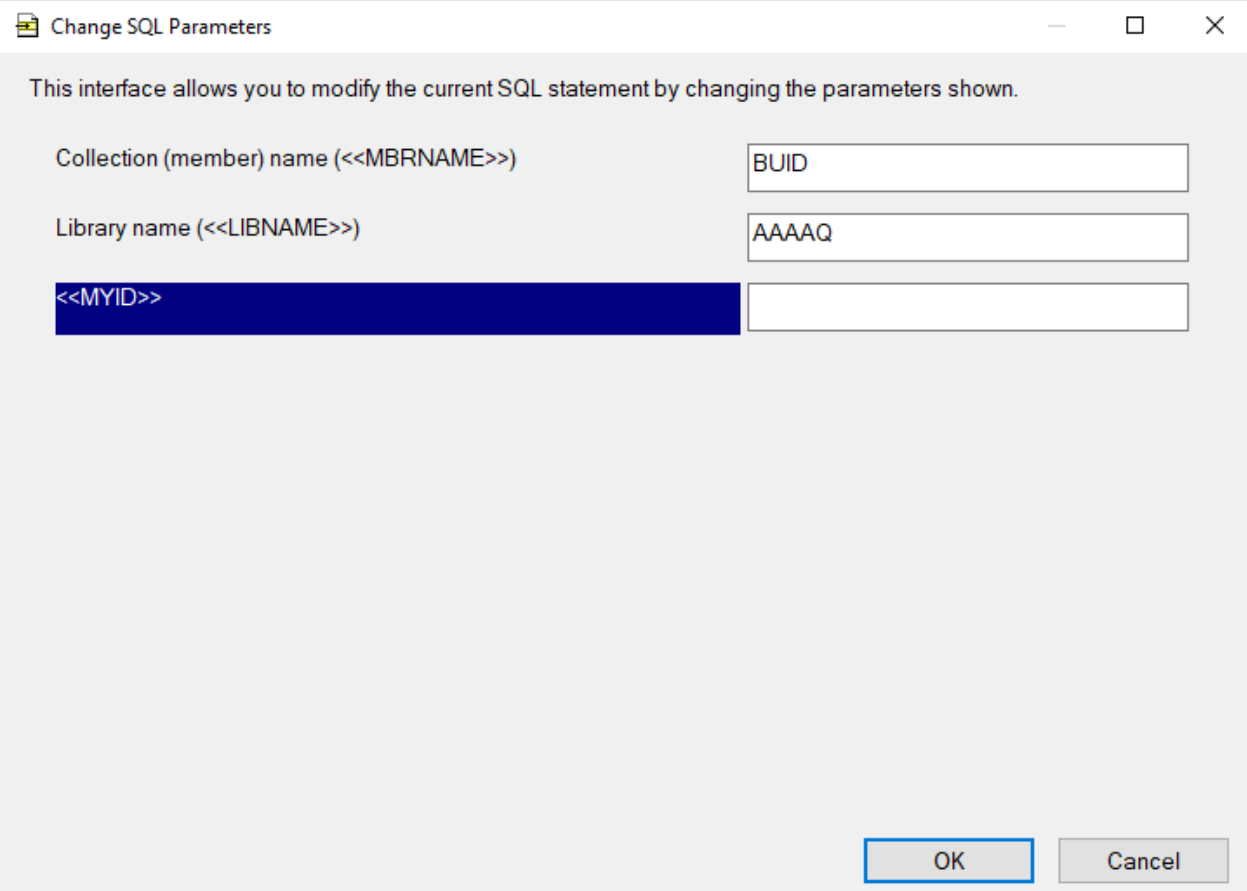
Other types of spool files besides job logs are displayable. However, they can only be opened using the iDoctor FTP GUI component. An example of a Spool File View is shown below:



Spool File View showing a Job Log.

6.11 Change SQL Parameters

This interface allows the user to modify the iDoctor-defined or user-defined parameters included in an SQL statement. In some cases, the interface will appear when first opening a report if a parameter value is unknown and needs to be filled in by the user.



Change SQL Parameters Window

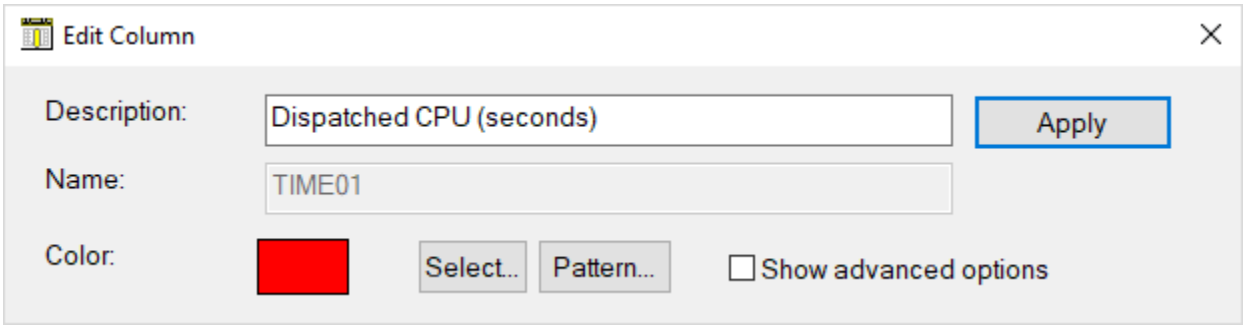
Each parameter in the SQL statement will be listed on this screen. Known values are filled and unknown values requiring user's input will be highlighted in blue.

6.12 Edit Column

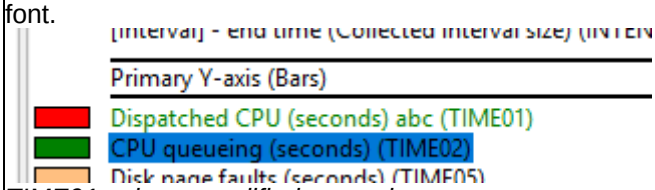
The Edit Column window is used to modify the attributes for a field/column in an iDoctor table or graph.

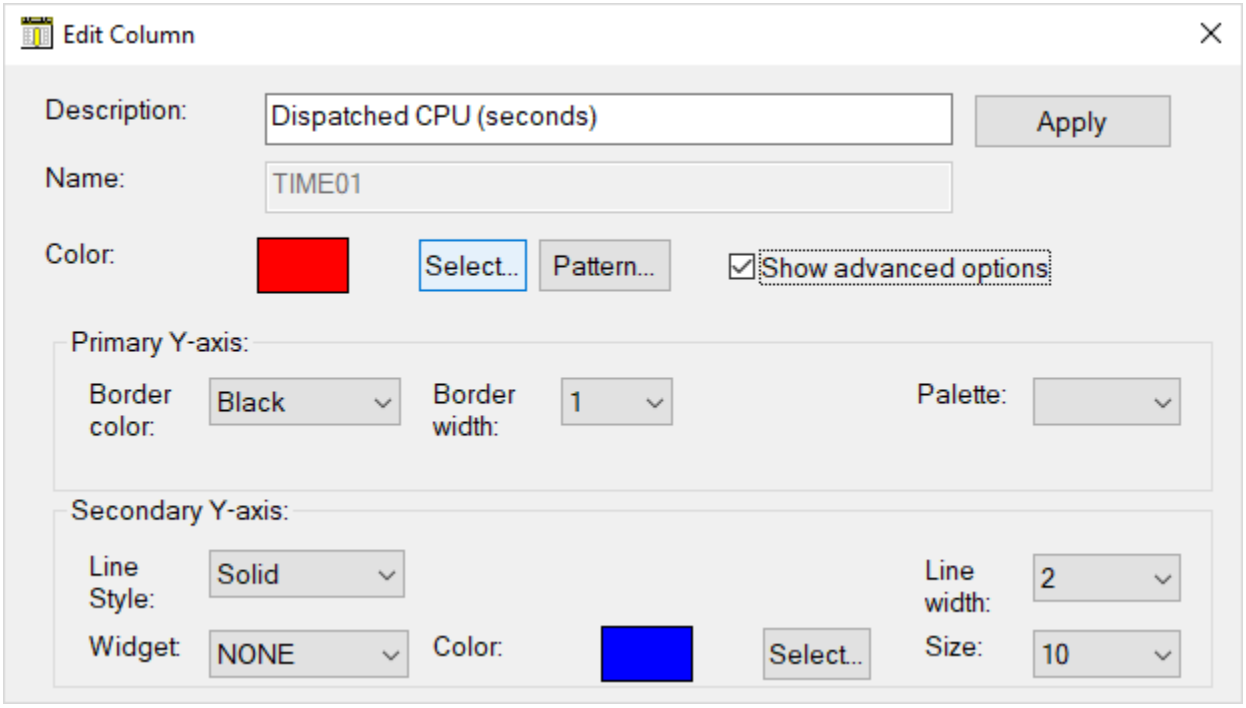
From tables this option is accessed by right-clicking on a column heading and using the Edit… menu. For graphs this option is accessed by right-clicking on a column in the graph legend and using the Edit… menu. Once the edit column window is opened from the graph legend you can click on other columns in the legend and it will load their settings.

Note: Changing a columns attributes will make those changes permanent and globally throughout iDoctor. The column's changes are saved in the User-Defined Reports Database. You can remove those changes later by going into database's Column settings overrides folder and deleting the column.

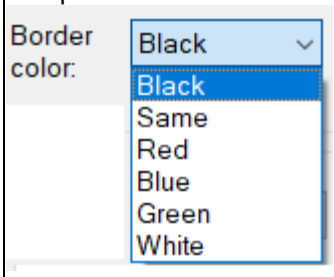
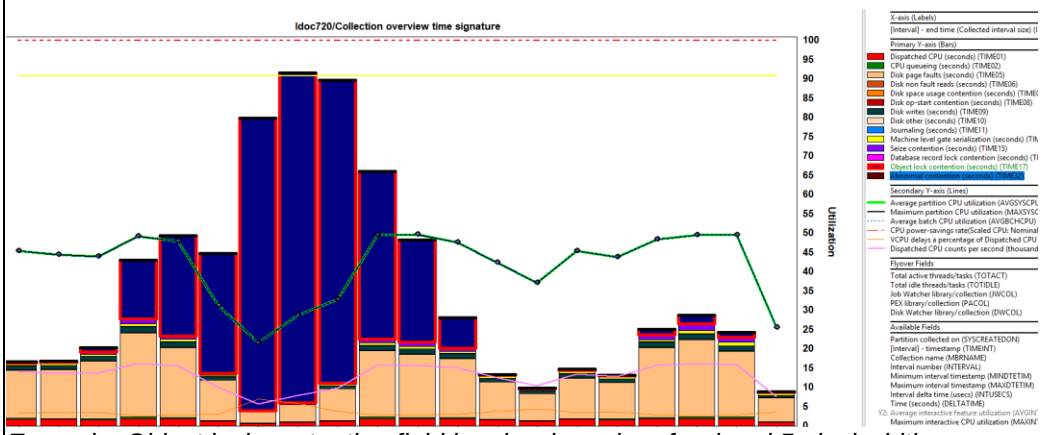
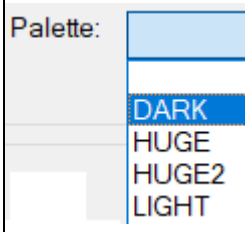


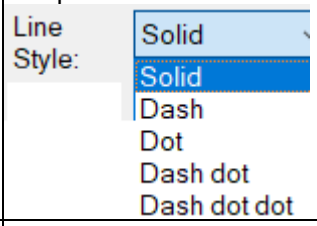
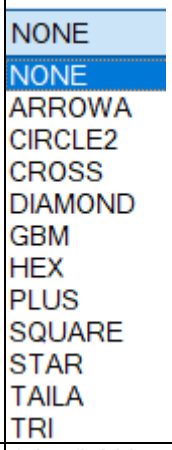
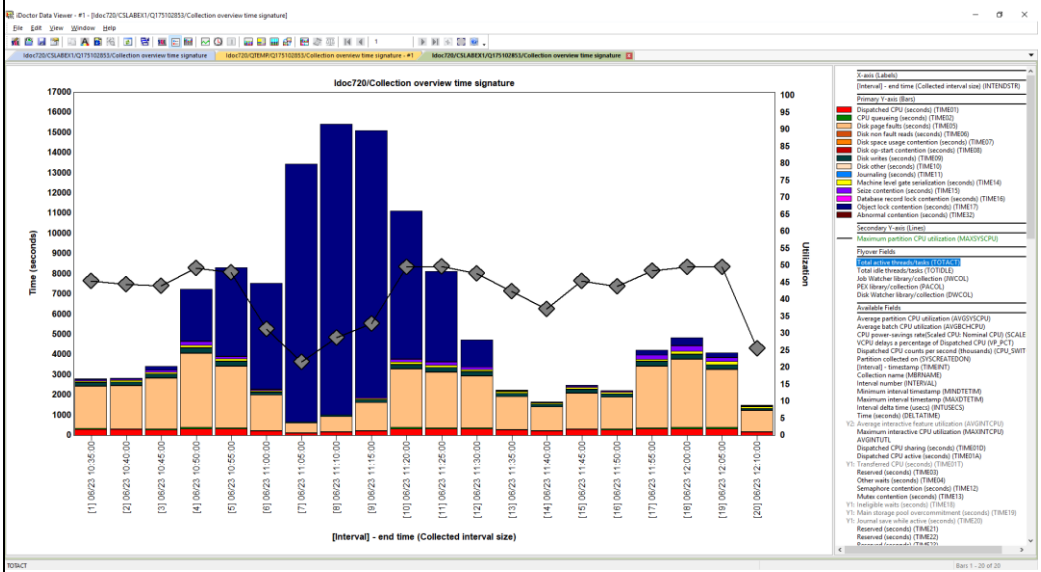
Edit Column

Option	Description
Description	The description to give the field name.
Apply button	This will apply the changes to the graph / report you are working with. If this is an iDoctor-defined field then these changes are saved and reused until removed from the User-Defined Reports Database -> Column settings overrides folder. **Tip:** Columns that have been changed from the iDoctor's defaults will have a green font.  *TIME01 column modified example*
Name	This is the name of the field or column in the graph/report.
Color	The color to give the field when on a bar graph or line. Applies only when the field is shown on a graph.
Select button	Allows the user to modify the color used by showing the Color Window.
Pattern	Allows the modification of the hatching/pattern used on graphs for this field, but only if the **Preferences -> Display -> Patterns** checkbox is checked.
Show advanced options	Displays additional settings for graphing.



Edit Column -> Show advanced options checked

Option	Description
Primary Y-axis border color	If the field is used on a graph's primary Y-axis and the graph is a bar graph, then this changes the color for the border. The default border width is 1 pixel. The possible values are:  A value of "Same" means the color will match the fill color for the field (which in effect means no border will be visible. **Tip:** If the number of bars per page on the graph exceeds 250 then the border is automatically removed from the graph to avoid the graph appearance changing too much and looking more and more black with many bars.  *Example: Object lock contention field has border color of red and 5 pixel width*
Primary Y-axis border width	If the field is used on a graph's primary Y-axis and the graph is a bar graph, then this changes the width for the border. The default border width is 1 pixel.
Primary Y-axis Palette	For flattened style graphs only, the colors used are dynamic because there are a variable number of values shown on the graph. This value lets you define a palette (a series of many colors predefined and shipped with iDoctor) for the field. **Note:** When drawing flattened style graphs, each color will be used from the Palette one at a time and when all colors are used the colors will wrap and be used again. The possible Palettes are:  Using the 1st option (blank) will cause a random color to be assigned each time. This is the default setting. DARK is a series of darker colors and LIGHT are lighter colors. The HUGE palettes cover all colors are predefined, so the same colors will be used each

	time the same graph is reopened for the same data. Otherwise if no Palette is defined the colors will be randomly defined every time and they may not look good depending on chance.
Secondary Y-axis line style	If the field is used on a graph's secondary Y-axis, this changes the style for the line. The possible values are: 
Secondary Y-axis line width	If the field is used on a graph's secondary Y-axis, this changes the width for the line. The possible values are 1-5 pixels.
Secondary Y-axis Widget (name)	If the field is used on a graph's secondary Y-axis, this allows each point of the line to contain an optional shape called a widget. These shapes have different names and the choices are: 
Secondary Y-axis Widget color	If the field is used on a graph's secondary Y-axis, this allows each point of the line to contain an optional shape called a widget. This is the color to draw the widget with. Press the select button to change the color.
Secondary Y-axis Widget size	If the field is used on a graph's secondary Y-axis, this allows each point of the line to contain an optional shape called a widget. This is the size to draw the widget. The bigger the number, the bigger the widget will appear.  *Secondary Y-axis line with size 30 diamond widget*

6.13 Color Window

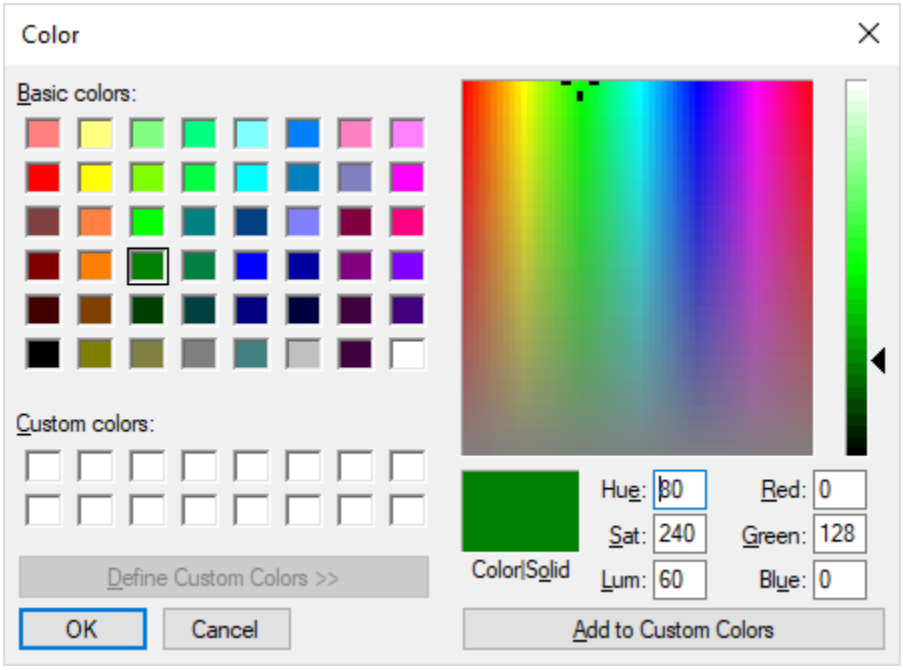
When modifying the color of a field in iDoctor, this window is used. Colors for iDoctor fields are defined using RGB values (Red/Green/Blue) where each color is a number between 0 and 255.

Examples:

Red = 255,0,0

Green = 0,128,0 or 0,255,0

Blue = 0,0,255



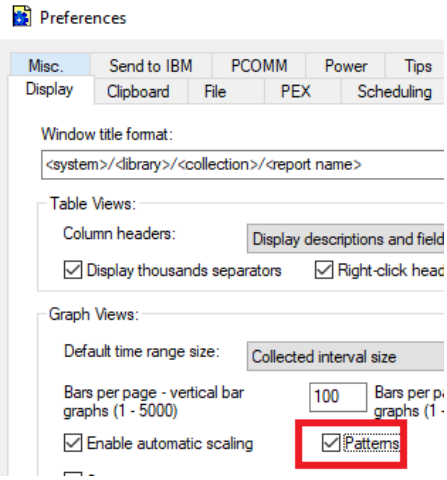
Color Window

The RGB values are shown in the boxes in the bottom right-corner of this interface. You can create your own colors here and save / reuse here if desired.

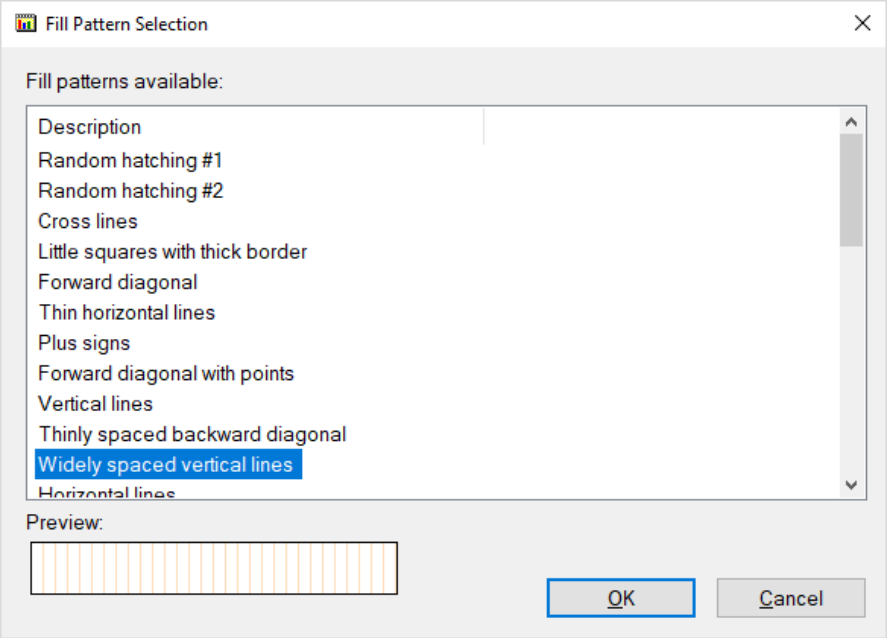
6.14 Fill Pattern Selection

This window allows a user to select the type of pattern they would like for the current field you are working with. This option is available from the Edit Column window or from the Graph Definition Primary Y-axis page.

Note: Making changes here has no effect on the graphs unless the **Preference -> Display -> Patterns** is checked.



Preferences -> Display -> Patterns



Fill Pattern Selection

6.15 SQL Parameters used by iDoctor

iDoctor uses many parameters in the SQL statements it ships. If modifying an SQL statement or creating a user-defined report in some cases it is useful to understand these parameters.

All parameters use a convention of <<PARM_NAME>> where PARM_NAME is the parameter name.

When using a graph view the Change SQL Parameters menu can be used to change many of the parameters.

Parameters used by iDoctor:

Parameter	Description	Components
<<MBRNAME>>	Collection name or member name	All
<<LIBNAME>>	Collection library name	All
<<STACKEDFILTER>>	"Flattened" graphing filter	All
<<POOLNUM>>	ASP number (1-255)	DW
<<QDDASP>>	ASP number (1-255)	PEX
<<DSASP>>	ASP number (1-255) (comma separated list)	CSI
<<BSDARSIZE>>	BSDAR grouping length (7 recommended)	PEX
<<DEVRNAM>>	Device resource name	DW
<<DEVRNAME>>	Device resource name	DW
<<DMFLAG>>	Disk mirror flag (A, B, %)	CSI
<<DSMDLN>>	Disk model number (comma separated list)	CSI
<<DISKNAMEFILTER>>	Disk name filter	AIX/VIOS
<<DSKSELECTION>>	Disk selection criteria	CSI
<<DSTYPE>>	Disk type (comma separated list)	CSI
<<STUNITNUM>>	Disk unit number	DW
<<UNITNUM>>	Disk unit number	DW
<<MAXDTETIM>>	End of time range selection	All
<<QTISTY>>	Event subtype	PEX
<<QTITY>>	Event type	PEX
<<TYPEDESC>>	Event type description	PEX
<<QTITY_SELECTION>>	Event type selection	PEX
<<DEVICEFILTER>>	Filter string for devices (NPIV)	AIX/VIOS
<<GENJOBLEN>>	Generic job grouping length	All
<<GENJOBSTART>>	Generic job grouping start	All
<<GENJOBNAME>>	Generic job name	All
<<SHOW_ALL_OBJECTS>>	Include All objects (1), DB files (0)	PEX - PDIO

<<PERM_OR_TEMP>>	Include perm (P), temp (T) I/Os	PEX – PDIO
<<IOTYPEFILTER>>	IO type filter: faults ('FT'), sync reads ('RS'), sync writes ('WS'), page outs ('PO'), async reads ('RA'), async writes ('WA')	PEX – PDIO
<<OBJGRPBYFIELD>>	Job group by	All
<<OBJHAVINGFIELD>>	Job group by having clause	All
<<JOBGRPTYPE>>	Job group type	All
<<OBJSELECTION>>	Job selection criteria	All
<<QRECN_CSV_LIST>>	List of QRECNs (CSV format)	PEX
<<JBPOOL>>	Memory pool number	CSI
<<STACKEDNAME>>	Name contains for flattened graphs (filter)	All
<<QSGOCX_SELECTION>>	Object location criteria	PEX - PDIO
<<OBJNAME>>	Object name	All
<<QSGONM_SELECTION>>	Object name criteria	PEX - PDIO
<<OBJVALUE>>	Object value	All
<<PDIOSORT>>	PDIO sort field (See Preferences -> PEX option)	PEX – PDIO
<<PROCNAME>>	Procedure name	JW, PEX
<<PGMNAME>>	Program name	JW, PEX
<<SQLTABLEFILTERS>>	SQL table filters (not used currently)	CSI, JW
<<MINDTETIM>>	Start of time range selection	All
<<SYSNAME>>	System name (data collected on)	All
<<TPDRN>>	Tape device resource name	CSI
<<TASKCOUNT>>	Taskcount identifier	JW
<<TIMERANGEGRPBY>>	Time group by	All
<<BUCKETNUM>>	Wait bucket (01-32)	All
<<BUCKETDESC>>	Wait bucket description	All

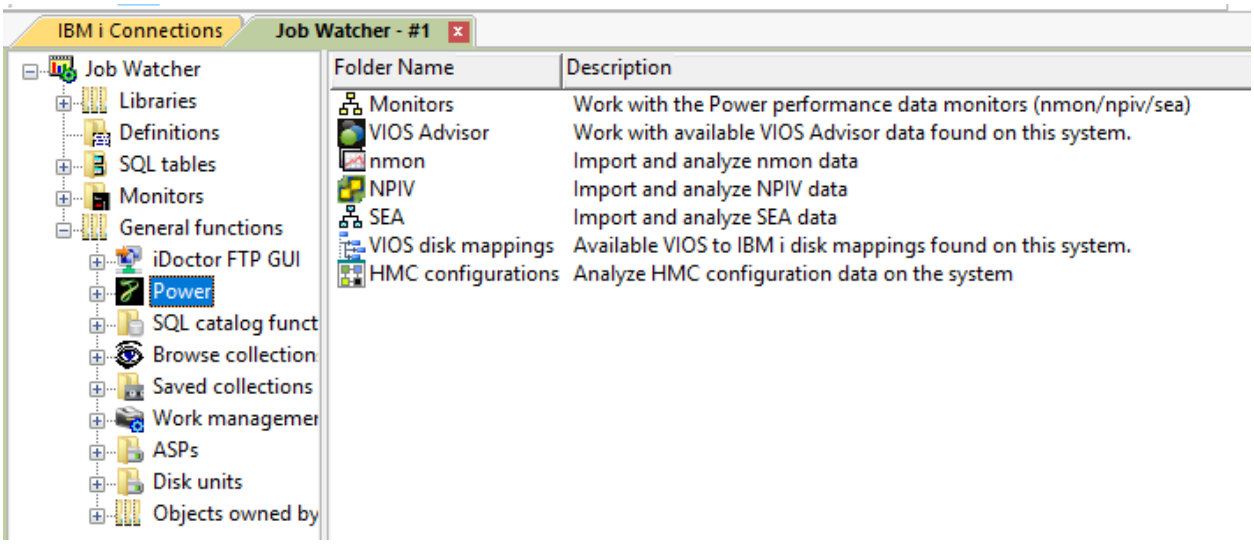
<<SWGROUP>>	Workload capping group ID	CSI
<<SWGNAME>>	Workload capping group name from file QAPMSYSWLC	CSI
<<TIMERANGEX>>	X-axis time label	All

7 Power

This chapter covers the functions available that are not specific to IBM i. This includes several options available to analyze performance on HMC, VIOS, AIX, Linux and more.

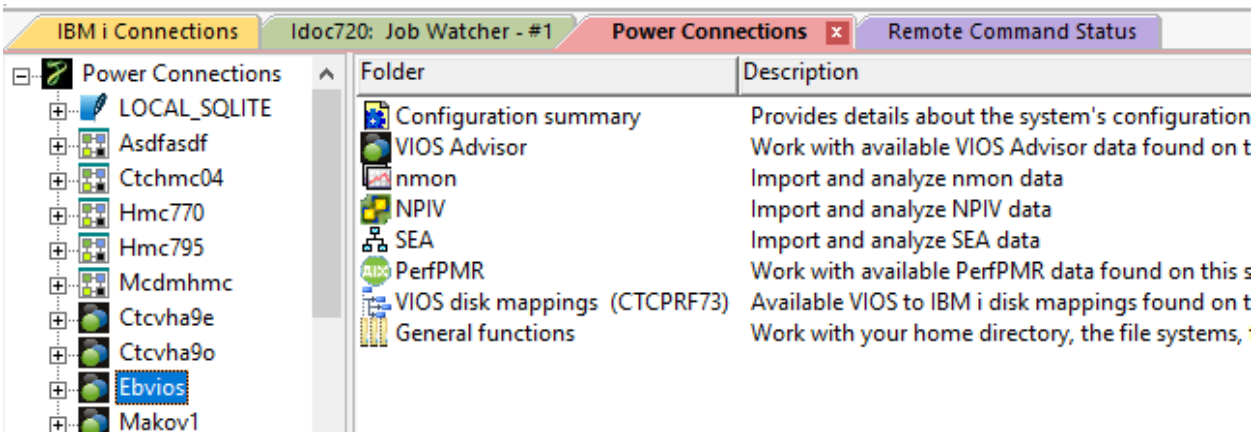
You can access these functions in multiple ways:

1) From an IBM i component, visit the **General functions -> Power** folder. The types of data you can work with are listed under this folder and data will need to be sent to the IFS (from other systems like VIOS) to use these options.



Job Watcher -> General functions -> Power

2) From the Power Connections View, under a VIOS, AIX or Linux type connection each type of data will be listed in its own subfolder. In this environment, the raw performance data will reside on the remote server



Power Connections -> VIOS

Note: These functions rely heavily on the use of Java and .NET (on the PC) as well as SSH/FTP connections being available to the desired servers. The required level of Java or .NET must be installed on the PC (check the PC Requirements section for specific levels if necessary.) Also check the connection settings to configure the type of FTP connection to use for each server.

7.1 Support issues

All features in this section **(Power Connections/Power)** are free but also offered **"as-is" with no warranty of any kind**.

Note: Development testing has occurred only minimally for HMC and VIOS types of connections and some of these functions may not work properly on Linux or AIX systems as they are not regularly tested.

If you have issues, feel free to contact us at idoctor@us.ibm.com but support for problems in this area of iDoctor is not guaranteed and only on a best effort basis.

7.2 Installation

When you make a connection in the Power Connections View to a non IBM-i system such as (VIOS/AIX), the 1st time iDoctor will attempt to FTP (transfer) several scripts to the system that are used for data collection and analysis purposes. If you refuse to allow this then most of the functions in this section will not work. This also requires a working FTP connection between the PC and the remote system.

For VIOS this requires the use of the padmin signon and most commands are executed via use of oem_setup_env. This is required in order to use these functions with the GUI.

These iDoctor scripts are installed to the /tmp/idoctor directory by default but this is configurable using the **Preferences -> Power -> Remote script directory** setting.

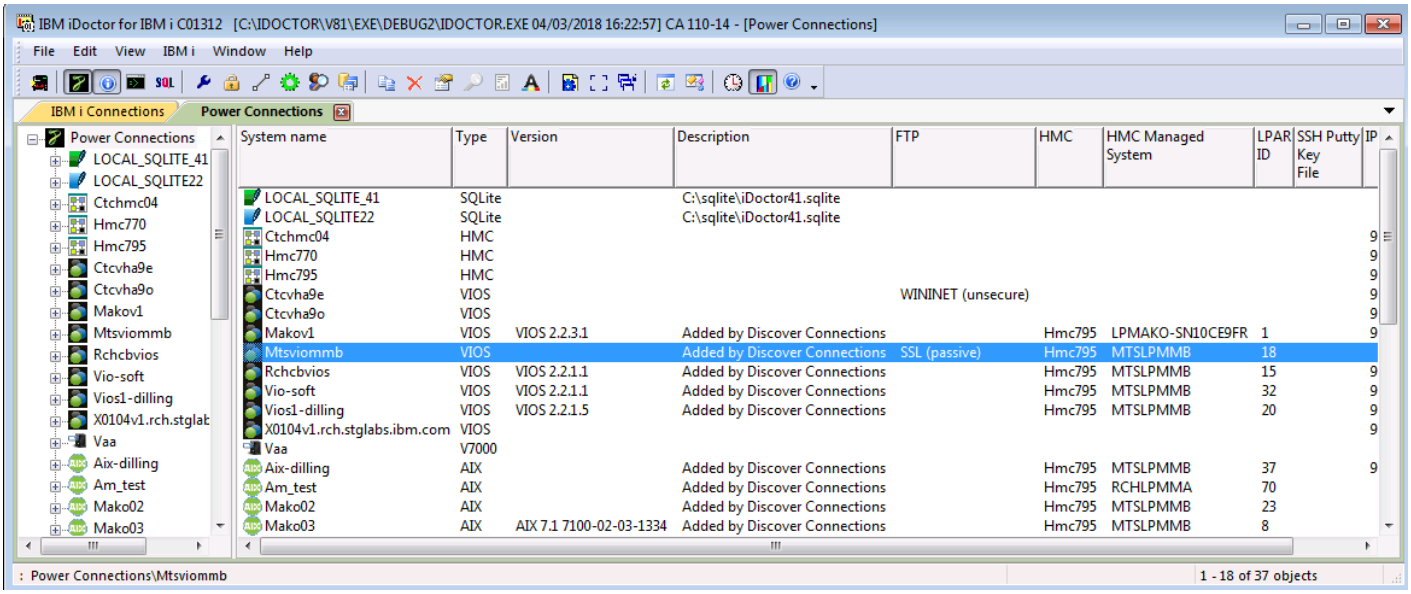
7.3 Data types supported

This section covers the various types of data supported by iDoctor within the Power functions:

Data type	Description
HMC configuration	This option will scan the HMC and capture a snapshot of the current configuration that can be saved/viewed in iDoctor. **NOTE:** This only applies to HMC type connections.
VIOS Advisor	**Only for VIOS**, this data type runs a collection using the **part** command (that includes a nmon collection within it) for 10 to 60 minutes. This produces a .tar file on the system that can be analyzed using the GUI or manually by sending the results to a client system (PC/Mac.) This results in a web-based report (within the .tar file) that advises you on what steps to take to improve performance on the VIOS.
Nmon	For AIX, VIOS and Linux this is a higher-level monitor function that typically is ran 24x7 and provides reporting a bit like Collection Services Investigator (CPU and other stats) over time. It is collected through the topas_nmon (or nmon) command and iDoctor provides multiple ways to kick this off or you could collect it on your own.
Npiv	This type of data is collected using one of the iDoctor scripts. The data is captured via the fcstat command and provides virtual/physical fiber channels statistics for the system based on the NPIV configuration. **NOTE: These functions have only been tested on VIOS and may not work on AIX or Linux.**
Sea	This type of data is collected using one of the iDoctor scripts to capture Shared Ethernet Adapter statistics. **Note:** This same data for the most part can also be captured via nmon, except that the nmon -sea data collection options do not include some of the error counters provided by the iDoctor SEA data collection script. **NOTE: These functions have only been tested on VIOS and may not work on AIX or Linux.**
PerfPMR	PerfPMR is used by AIX/VIOS support to help analyze performance problems in more detail that what can be done with a nmon collection. Options exist in iDoctor to install and collect this type of data, but analysis is not done currently with iDoctor.

7.4 Power Connections View

The Power Connections view allows you to work with all the connections defined to non-IBM i systems (HMC, VIOS, AIX, Linux, etc) created in iDoctor. You can easily add or remove connections to these types of systems through this view. The primary purpose of this view is to provide a quick and easy way to launch iDoctor functionality for any system desired.



Main Window displaying the Power Connections View

The top level of the tree/list contains several columns that represent settings for each connection. All of these values supplied are based on the last known connection made to the system and may not reflect current system settings.

Note: When iDoctor connects to these systems it will primarily use SSH to make the connection but in some cases, FTP will also be used.

Column	**Description**
System	System/partition name or IP address.
Type	The type of system the connection applies to. The possible values are: 0) SQlite: (**Note**: This refers to an SQLite database/ODBC data source connection on the PC.) 1) HMC 2) AIX 3) VIOS 4) V7000 5) Linux 6) Other
Version	The version of the operating system for each connection (if known.)
Description	An optional description given to the connection within iDoctor.
FTP	This indicates the preferred FTP method to use for each connection.
HMC	This value identifies the managing HMC (if applicable) for the partition. This value is filled by iDoctor after connecting to the HMC and using the Discover Connections interface to find new connections.
HMC Managed System	This value identifies the HMC managed system for the partition.
LPAR ID	This column lists the LPAR ID as defined in the HMC for the partition.
SSH Putty Key File	An SSH key file can be used on a connection to help ensure a secure connection to this system. Typically, these have a passphrase that must be entered when making the connection. You must use the Putty tool to generate this: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
IP Address	The last known IP address of the system.

Since this interface is a tree and a list, there are different options available at each level of the tree. The top level of the tree contains a list of all your Power connections.

When right-clicking the Power Connections (root) folder, the following options are shown:

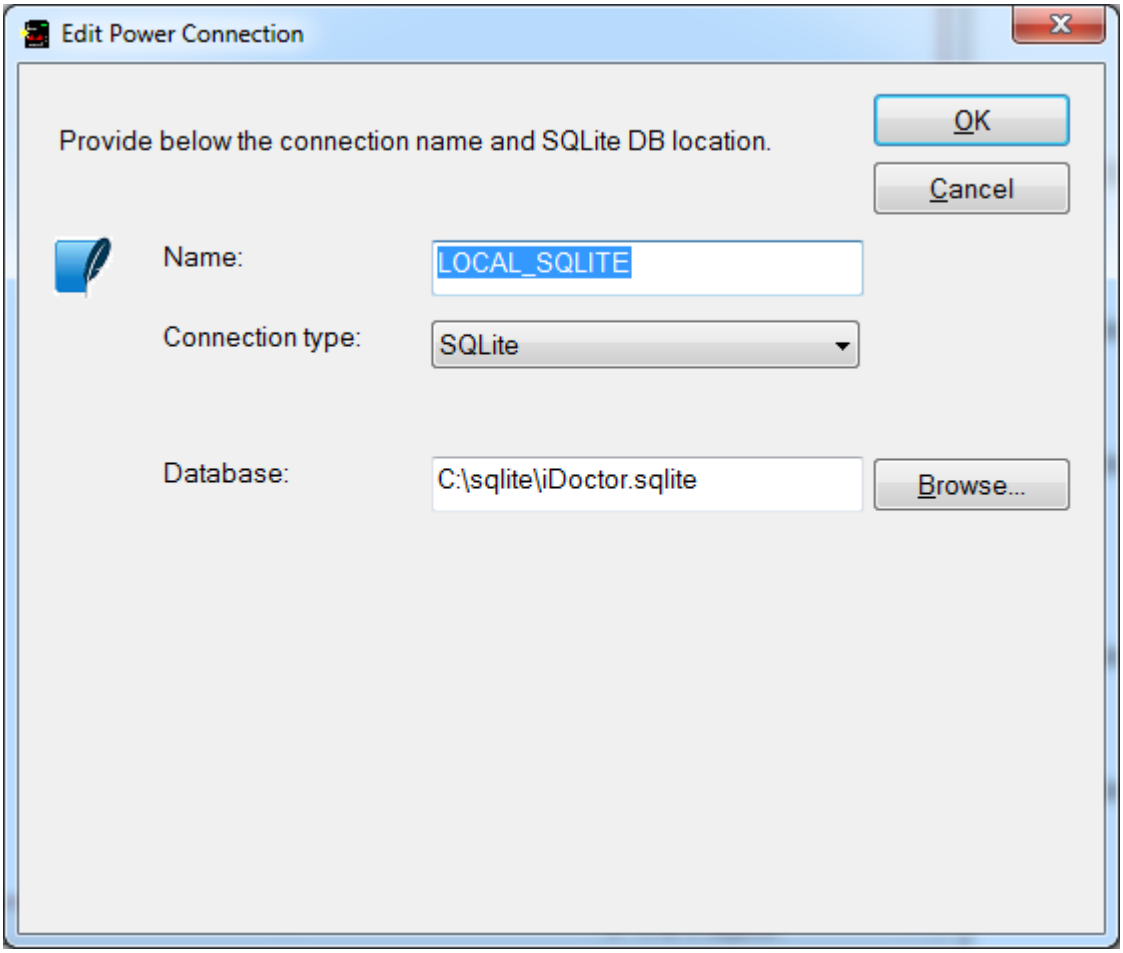
Power Connections Menu	**Description**
Add Connection	Use this menu to add a non-IBM i connection to the Power Connections View.
Set Analysis Database	This option allows you to configure which analysis database iDoctor should be using when analyzing non-IBM i (Power) performance data. This allows you to either specify an IBM i to use to store the performance data on, or a local SQLite instance may be used instead (which are viewable within the Power Connections tree.)
Export connections	Use this option to create a Windows registry file that contains a list of all your non-IBM i connections. This file can be used to restore all your connections at a later time or to another PC.

Each type of connection and the options they provide will be discussed in the next sections.

7.4.1 SQLite

SQLite connections are databases on the PC that store performance data for non-IBM i systems. The types of data supported includes nmon, npiv and sea (shared Ethernet adapter stats.)

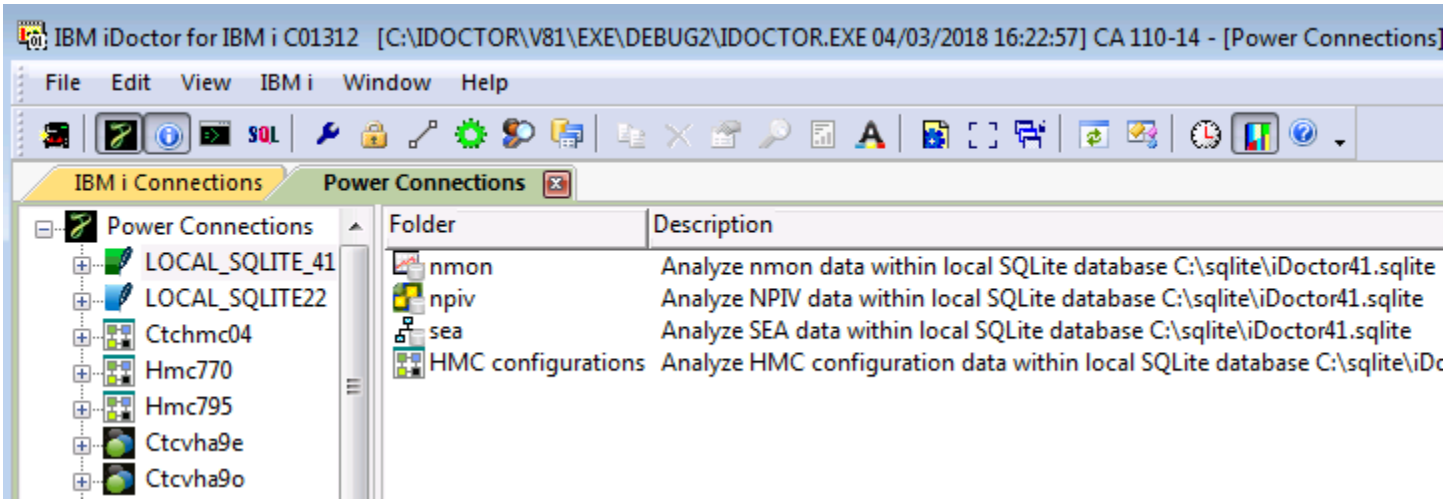
Each SQLite connection is equivalent to the ODBC connection name on the PC for a specific SQLite DB file on the PC.



SQLite Connection Example

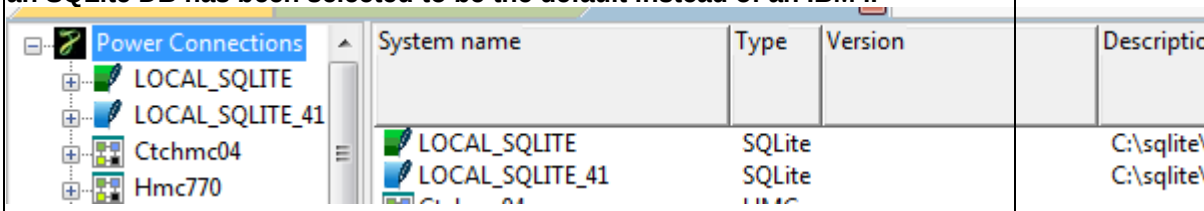
You may add new SQLite database connections by using the Add Connection window within the Power Connections View.

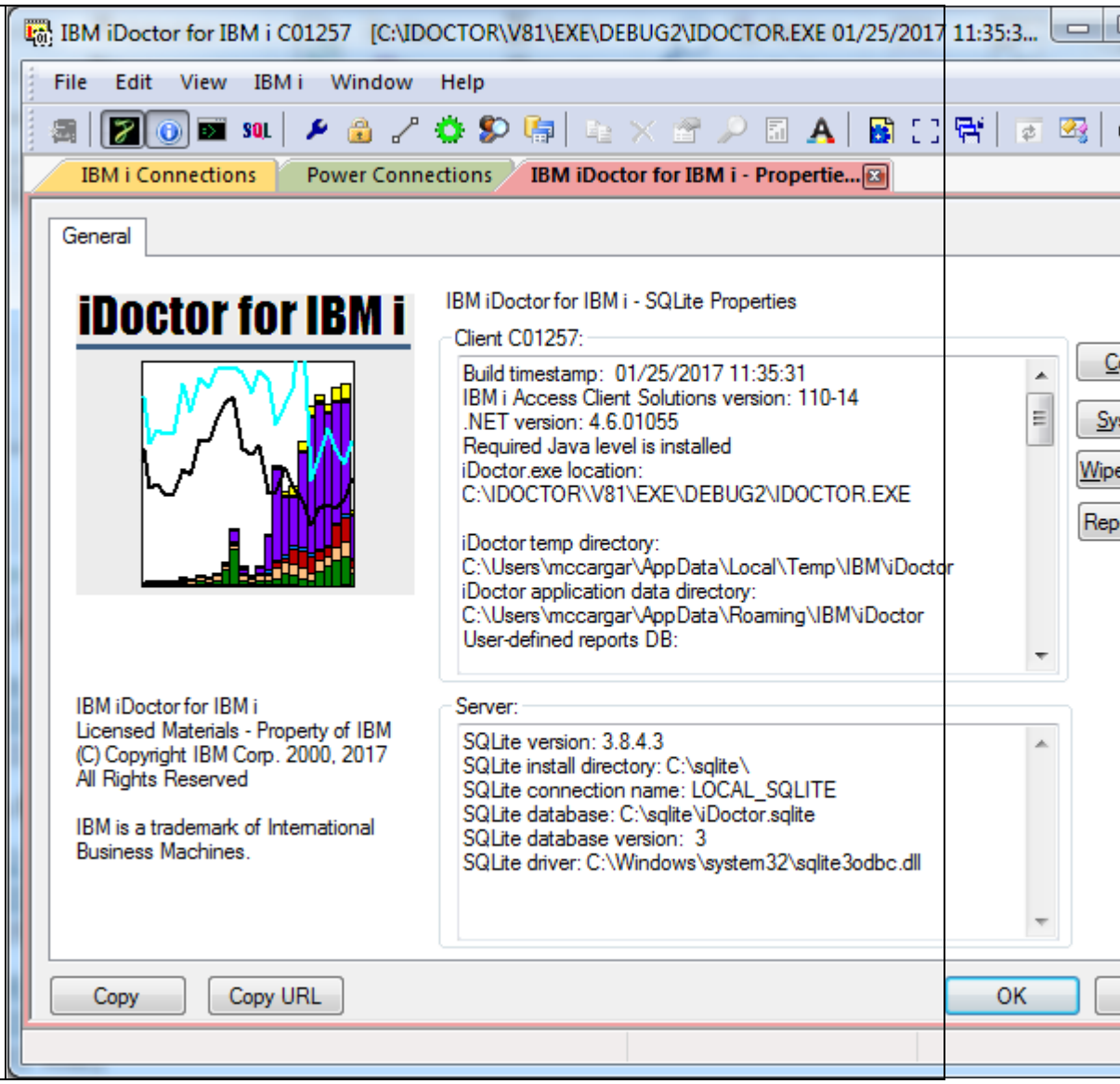
Each SQLite connection contains a set folders that allow the user to view data available for each type.



SQLite connection folder

When right-clicking on an SQLite connection the following options will be shown:

Menu	Description
Analyze Data (nmon, npiv, sea)	This option allows you to import nmon, or npiv/sea data files into the selected SQLite database for analysis purposes.
Add Connection	Use this menu to add another connection to the Power Connections View.
Delete	This will delete the selected connection(s) from the Power Connections View.
Edit	This option allows you to modify the selected connection's settings.
Set As Default Analysis Database	This option lets you set the current SQLite DB selection as iDoctor's default analysis database for looking at non-IBM i performance data. **Note: The default analysis database will have a green color instead of blue if an SQLite DB has been selected to be the default instead of an IBM i.** 
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Select all	Selects all connections in the list.
Properties	This option will display the location and some of the settings for the SQLite database.

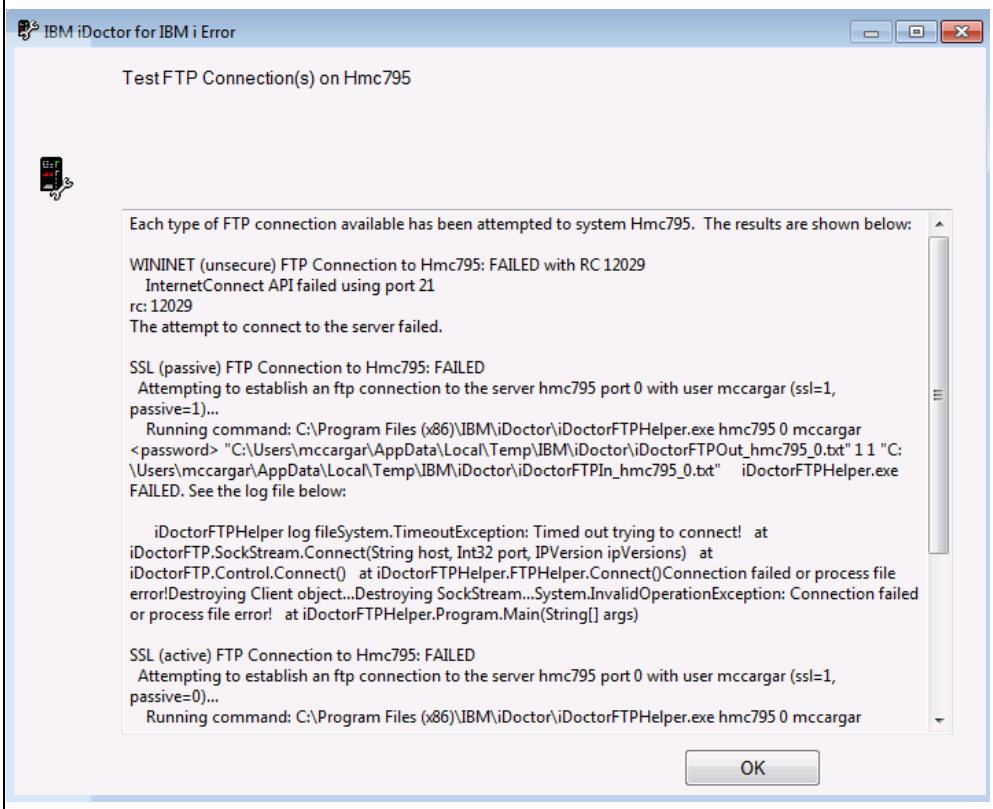


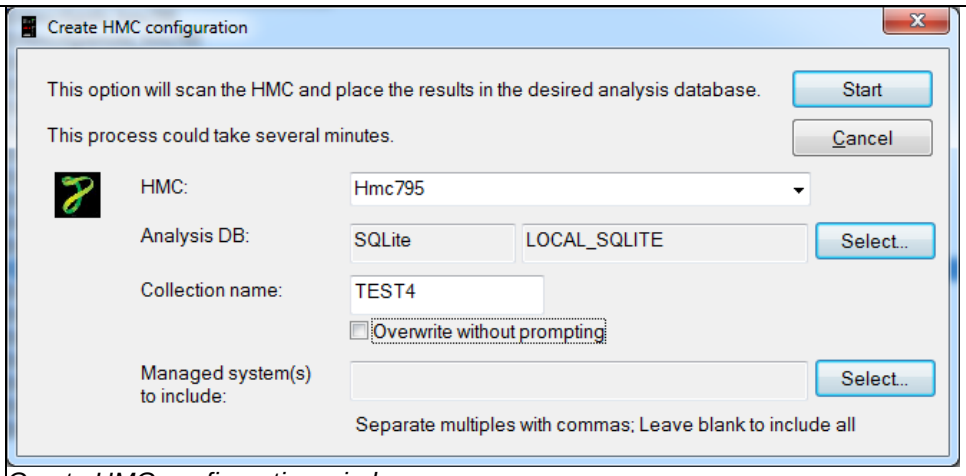
Note: See the Chapter on Power for additional information on how to import and analyze data in an SQLite database.

7.4.2 HMC Connections

HMC type connections in the Power Connections View, provide access to the HMC, including the ability to drill down and view the managed systems and LPARs within each managed system. You can also utilize the Discover Connections function to look for and add the desired types of connections found on the HMC to iDoctor more easily.

When right-clicking on an HMC the following options will be shown:

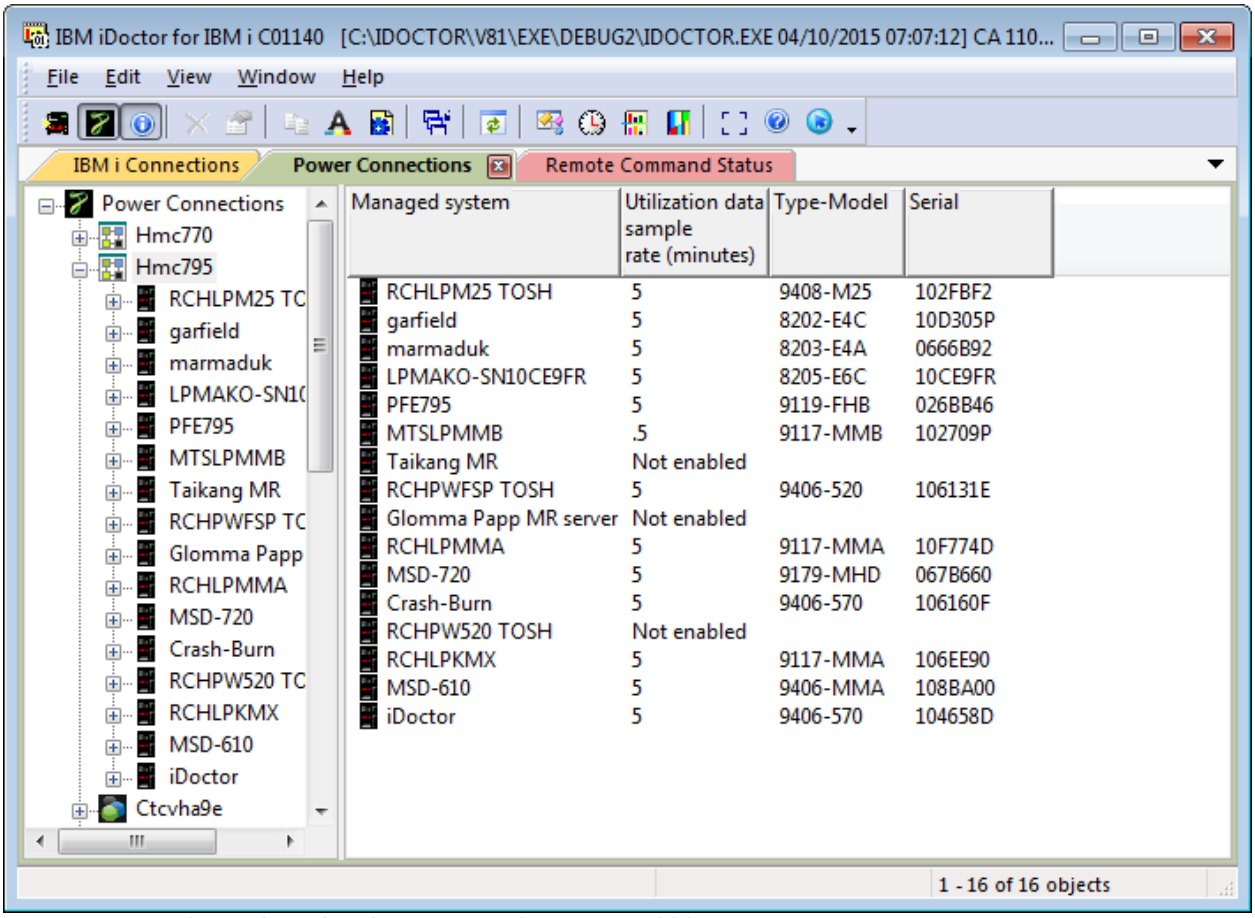
HMC Menu	Description
Terminal Sessions – Launch Putty (SSH)	If Putty is installed and preferences on the Power page are configured correctly to allow iDoctor to know its location, then this option will attempt to launch Putty using an SSH connection to the selected system.
Terminal Sessions – Launch Putty (Telnet)	If Putty is installed and preferences on the Power page are configured correctly to allow iDoctor to know its location, then this option will attempt to launch Putty using a telnet connection to the selected system.
Terminal Sessions – End all Putty sessions	This option will kill all open instances of Putty regardless of whether they were started by iDoctor or not.
Check – FTP Connections	This option will check each of the types of FTP Connections available with iDoctor to the selected system and indicate which methods work and which do not.  **Note:** The GUI may appear to hang during this step, but this is currently normal.
Check – FTP Connections (verbose)	This is the same as the previous option but also provides additional debug information.
Check – Connection (show ping results)	This option will ping the selected system and displays the results.
Check – IP Address	This option will ping the selected system in order to determine the current IP address.
Start HMC Collection	This option will kick off the HMC collection of the desired type (configuration ~~or lslparutil~~) and store the results in the currently specified analysis database. **Note:** The lslparutil support has been removed.

	 *Create HMC configuration window*
Discover Connections	This option displays the Discover Connections function to look for and add the desired types of connections found on the HMC to iDoctor.
Add Connection	Use this menu to add a Power connection to the Power Connections View.
Delete	This will delete the selected connection(s) from the Power Connections View.
Edit	This option allows you to modify the selected connection's settings.
Set Analysis Database	This option allows you to configure which analysis database iDoctor should be using when analyzing non IBM i (Power) performance data.
Export connections	Use this option to create a Windows registry file that contains a list of all your Power connections. This file can be used to restore all your connections at a later time or to another system.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.
Set User-Defined Reports Database	This option allows a user to load/use another user's iDoctor user-defined reports/graphs that they have previously created. When saving user-defined reports these are saved into the specified database. This can either be an IBM i library or a local database on the PC (MDB file). To find the current user-defined reports DB settings, either use this menu option or see the application properties (Help -> About menu) and then look for the "User-defined reports DB" location.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Select all	Selects all connections in the list.

7.4.2.1 Managed systems

Expanding an HMC will reveal the managed systems it contains and also provides the ability to change the data collection settings for any of the managed systems.

For the purposes of analyzing data it is recommended to keep the sample rate the same for all managed systems that you wish to analyze within iDoctor.



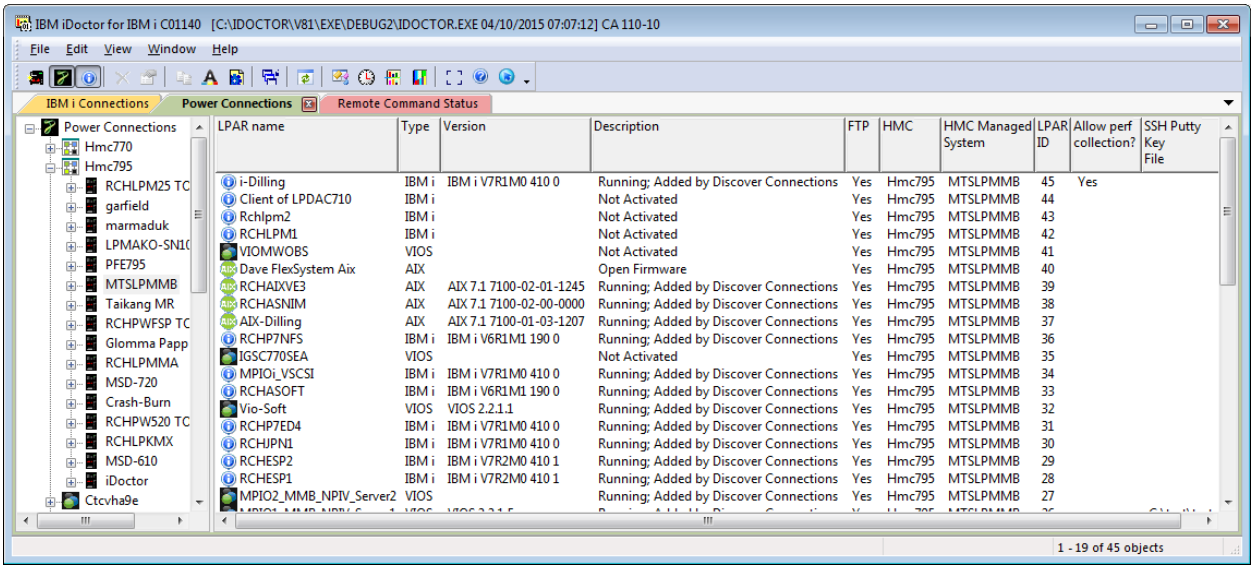
Power Connections View showing managed systems within an HMC

Right-clicking on a managed system provides the following options:

Managed System Menu	**Description**
Set lslparutil sample rate	This option provides the ability to set the lslparutil data collection sample rate for the selected managed systems. The possible values are: 1 minute, 5 minutes, 30 minutes, 1 hour or disable.

7.4.2.2 LPARs

Expanding a managed system will display basic information about the LPARs defined within it. If desired, you can also drill down further into a desired LPAR in order to view iDoctor functions available within for each type (IBM i, Linux, AIX, VIOS.)



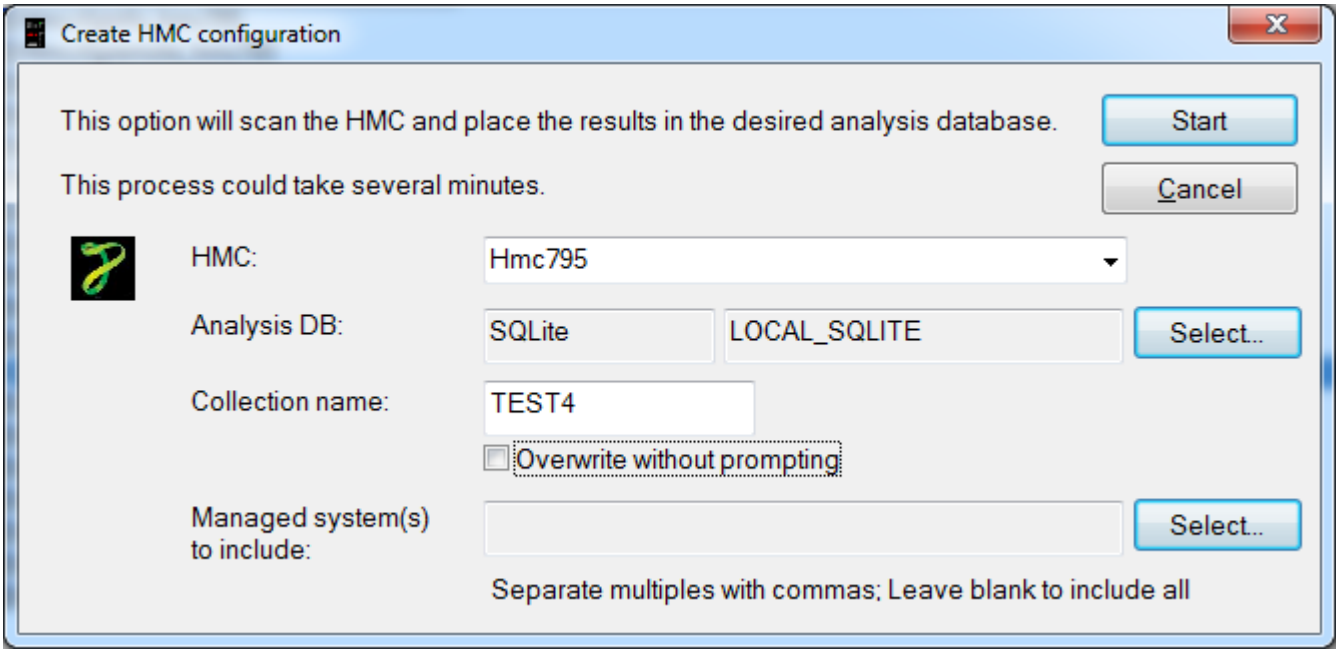
Power Connections View displaying LPARs within an HMC's managed system

Right-clicking on an LPAR provides the following options:

LPAR Menu	Description
Launch iDoctor components…	This action will either display the iDoctor components window if the LPAR is an IBM i or it will launch the applicable component for the type of LPAR selected into a new window.
Enable performance collection	This option will set the allow performance collection flag to true for the selected LPARs.
Disable performance collection	This option will set the allow performance collection flag to false for the selected LPARs.

7.4.2.3 Create HMC configuration

This option will kick off the HMC collection to capture configuration data and stores the results in the currently specified analysis database.



Create HMC configuration window

7.4.3 VIOS Connections

After the HMC's in the Power Connections View you will find the VIOS connections (if any exist.)
A VIOS connection can be used to drill down into the VIOS Investigator component directly within the Power Connections View or you can right-click and use the Launch option to open VIOS Investigator for the selected system into a new window.

When right-clicking on a VIOS the following options will be shown:

VIOS Menu	Description
Launch VIOS Investigator	This option will launch the VIOS Investigator component for the selected VIOS. This provides options for collecting or analyzing performance data on the VIOS.
Terminal Sessions – Launch Putty (SSH)	If Putty is installed and preferences on the Power page are configured correctly to allow iDoctor to know its location, then this option will attempt to launch Putty using an SSH connection to the selected system.
Terminal Sessions – Launch Putty (Telnet)	If Putty is installed and preferences on the Power page are configured correctly to allow iDoctor to know its location, then this option will attempt to launch Putty using a telnet connection to the selected system.
Terminal Sessions – End all Putty sessions	This option will kill all open instances of Putty regardless of whether they were started by iDoctor or not.
Check – FTP Connections	This option will check each of the types of FTP Connections available with iDoctor to the selected system and indicate which methods work and which do not. **Note:** The GUI may appear to hang during this step, but this is currently normal.
Check – FTP Connections (verbose)	This is the same as the previous option but also provides additional debug information.
Check – Connection (show ping results)	This option will ping the selected system and displays the results.
Check – IP Address	This option will ping the selected system in order to determine the current IP address.
Start Power Collection - Wizard	This will display the Power Collection Wizard where the user can determine which types of performance data to collect on the system.
Start Power Collection - VIOS Advisor	Use these options to collect a VIOS Advisor collection (which is really the part command), on the desired VIOS.
Start Power Collection – nmon	These options are used to kick off nmon collections on the desired VIOS.
Start Power Collection – NPIV	This option is used to kick off an NPIV collection on the desired system.
Start Power Collection – SEA	This option is used to kick off an SEA (Shared Ethernet Adapter) collection on the desired system. **Note:** an option exists in nmon to collect most of these same statistics
Start Power Collection – PerfPMR	This option will kick off a PerfPMR collection which is typically need by AIX support to analyze problems.
Analyze Data (nmon, npiv, sea)	This is used to analyze .nmon, .npiv or .sea files stored on the PC. The files will be sent to and analyzed on the currently specified analysis database.
Install PerfPMR	This option can be used to install the PerfPMR tool on this system.
Add Connection	Use this menu to add a Power connection to the Power Connections View.
Delete	This will delete the selected connection(s) from the Power Connections View.
Edit	This option allows you to modify the selected connection's settings.
Set Analysis Database	This option allows you to configure which analysis database iDoctor should be using when analyzing non IBM i (Power) performance data.
Export connections	Use this option to create a Windows registry file that contains a list of all your Power connections. This file can be used to restore all your connections at a later time or to another system.
Create Disk Mapping (VIOS to IBM i)	This option displays the Create Disk Mapping window which allows a user to add an IBM i to VIOS disk mapping to their system. **Note:** The IBM i used for this function is the same as the analysis database (which must be set to an IBM i, if using this option.)

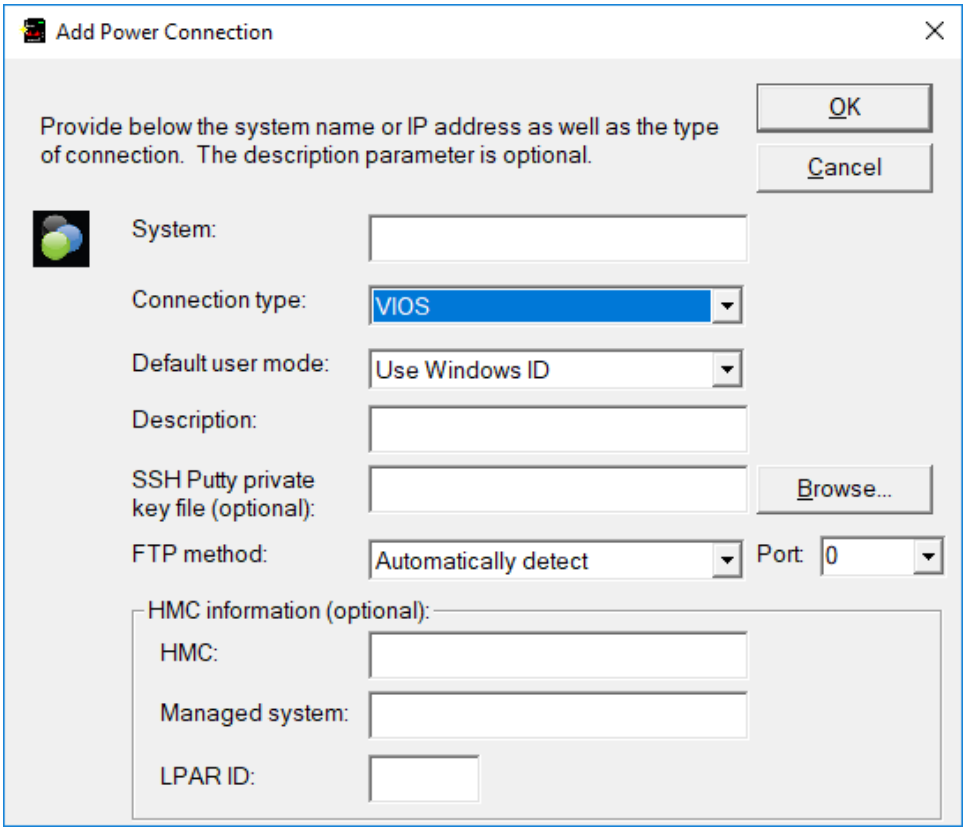
Reset Disk Mapping Signon	This option removes the SysMgrs file from the IFS under the current user's home directory (if it exists.) This will reset the Create Disk Mapping process so the user can pick a different HMC or HMC user to create the next disk mapping with.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.
Set User-Defined Reports Database	This option allows a user to load/use another user's iDoctor user-defined reports/graphs that they have previously created. When saving user-defined reports these are saved into the specified database. This can either be an IBM i library or a local database on the PC (MDB file). To find the current user-defined reports DB settings, either use this menu option or see the application properties (Help -> About menu) and then look for the "User-defined reports DB" location.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Select all	Selects all connections in the list.
Properties	This option displays the properties for the system/component.

7.4.4 Add/Edit Power Connection

This window allows a user to add (or edit) a connection to the list.

Simply provide the system name or IP address, the type of connection, default user mode and an optional description and click OK to register the system on your PC and add it to the list.

An example of this interface is:



Add Power Connection Window

The options available on this screen are described in the following table:

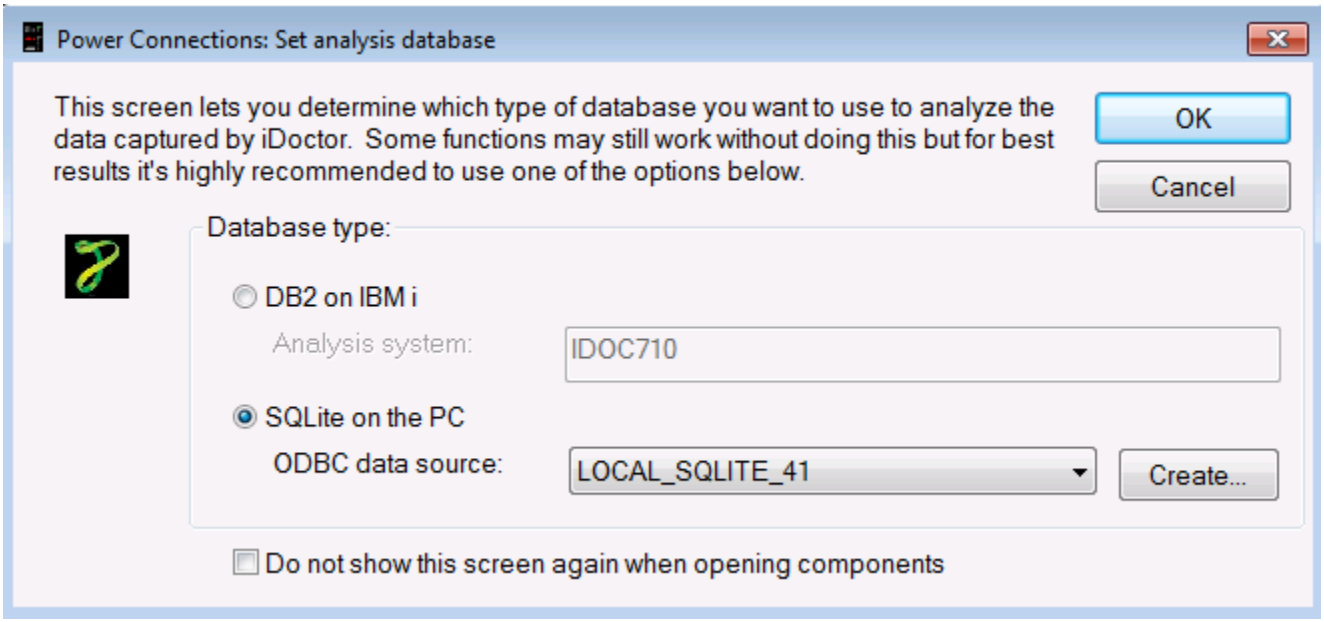
Option	Description
System	System/partition name or IP address.
Connection type	This drop down list provides the possible types of connections you can create. The possible values are: AIX VIOS V7000 HMC Linux Other
Default user mode	Indicates how the value for the user name to use when making connections to this system will be determined.
Description	An optional description given to the connection.
SSH Putty Private key file	An SSH key file can be used on a connection to help ensure a secure connection to this system. Typically, these have a passphrase that must be entered when making the connection. You can use the Putty tool to generate this: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
FTP method	This option lets you pick which FTP method is preferred when connecting to this system. “Automatically detect” is the default option but several options are available.
HMC	The name of the HMC this system is associated with. **Note:** This is an optional parameter and is prefilled by the Discover Connections interface.
Managed system	This name of the HMC managed system that this connection is associated with. **Note:** This is an optional parameter and is prefilled by the Discover Connections interface.
LPAR ID	This value is the LPAR ID as specified on the HMC this connection is associated with. **Note:** This is an optional parameter and is prefilled by the Discover Connections interface.

7.4.5 Set Analysis Database

This window allows a user to configure which type of database to use when working with Power (non IBM i) performance data such as nmon, npiv data from an VIOS, etc.

Note: You could also analyze this type of data on an IBM i system under Job Watcher, PEX Analyzer or the other IBM i components by using the General functions -> Power folder.

An example of this interface is:

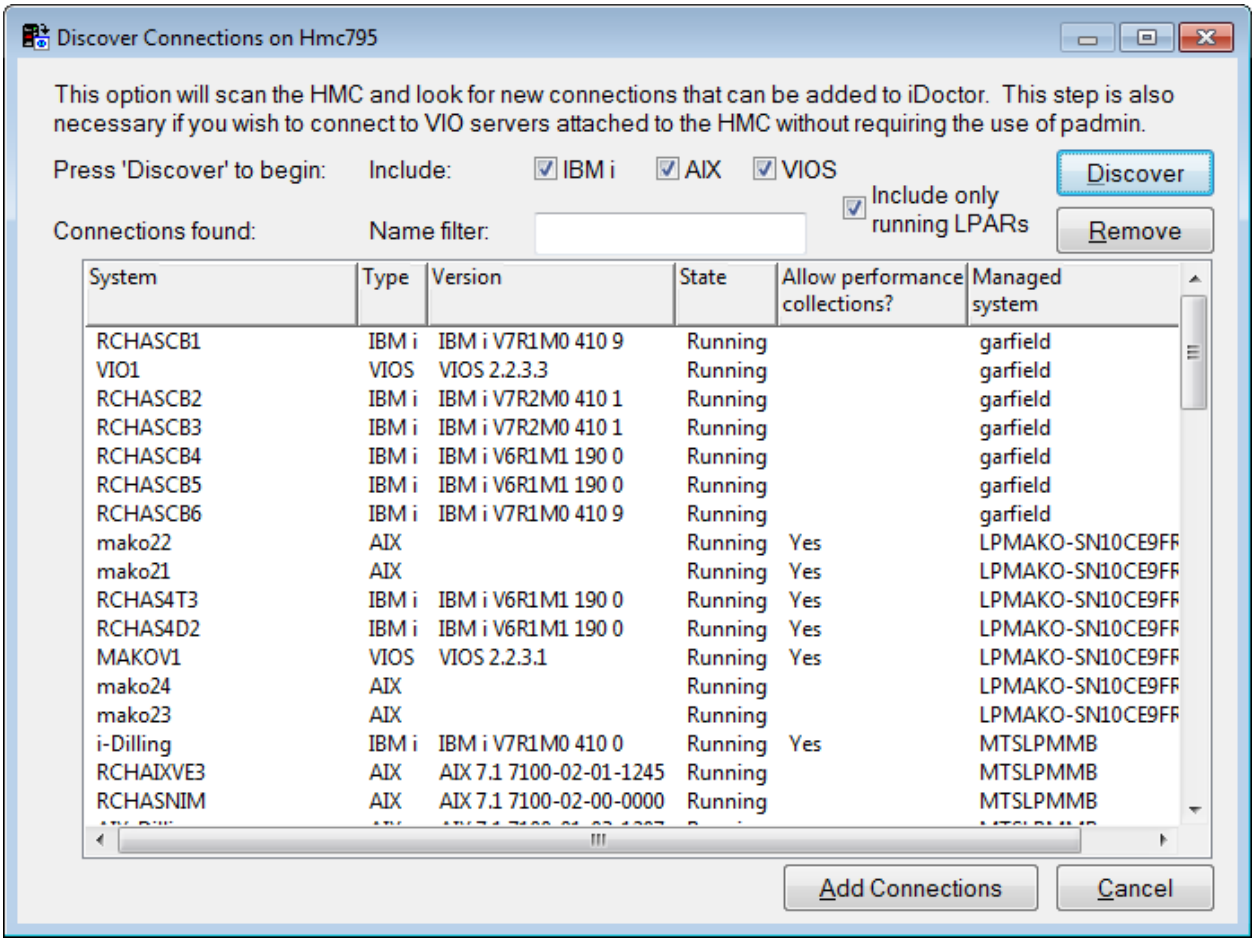


Set analysis database Window

Currently the only supported options are DB2 on IBM i and SQLite on the PC.

7.4.6 Discover Connections

The Discover Connections interface allows HMC users the ability to browse or search for LPARs by name or type. The user will be presented with a list of LPARs matching the search criteria and if desired these connections can be added to the Power Connections View by pressing the Add Connections button.



Discover Connections

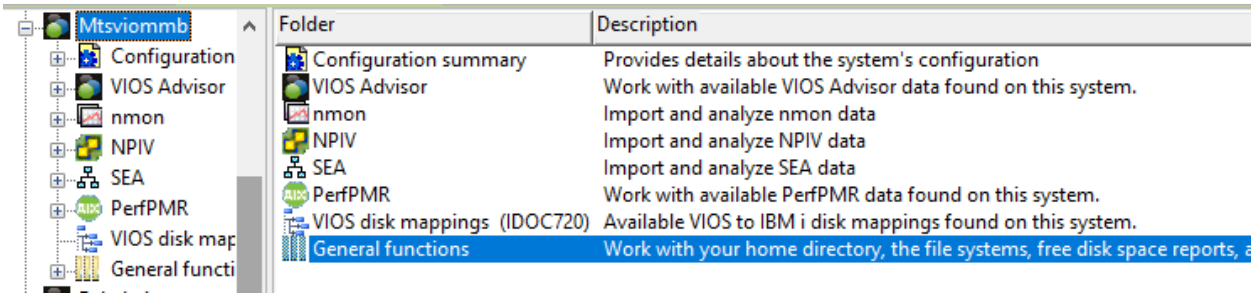
The options available on this screen are described in the following table:

Option	Description
Include	These all you to include/exclude LPARs based on the type (IBM i, AIX or VIOS.)
Name filter	This option provides the ability to search for an LPAR based on name.
Include only running LPARs	This option gives the ability to only return LPARs that report a Running state from the HMC.
Discover button	Press this button to perform a search based on the criteria given.
Remove button	This button will remove all selected LPARs from the search results window.
Add Connections	This button will add the selected LPARs to the Power Connections View.

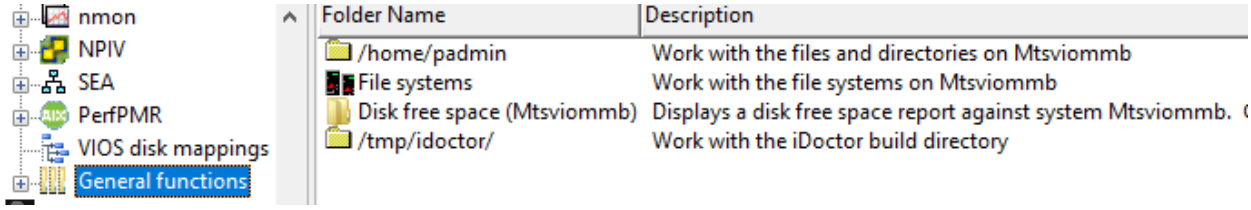
7.5 General Functions

This section covers the options related to viewing directories and files on non-IBM i systems within the Power Connections interface. These options do NOT apply to HMC type connections. They apply only to AIX, Linux and VIOS types of connections.

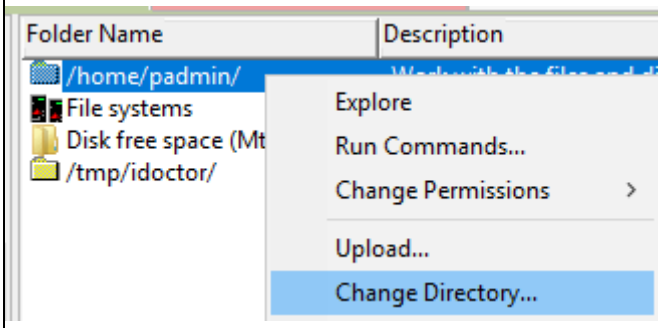
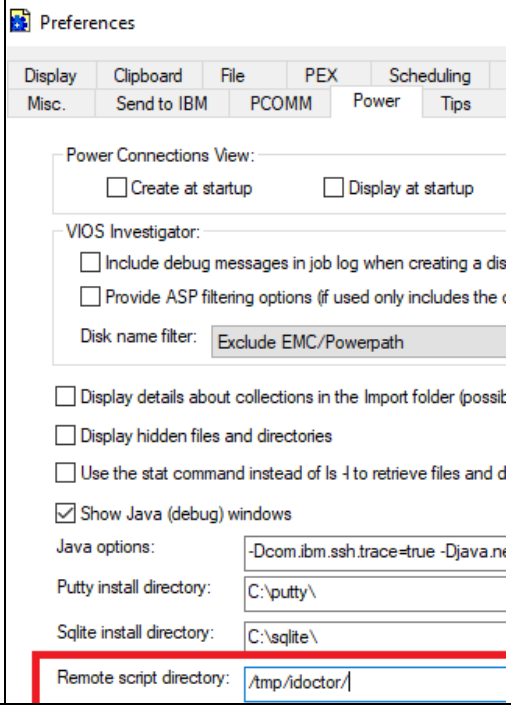
For example, under a VIOS connection a General functions folder exists that provides additional options.



Power Connections -> (VIOS) mtsviommb -> General functions folder



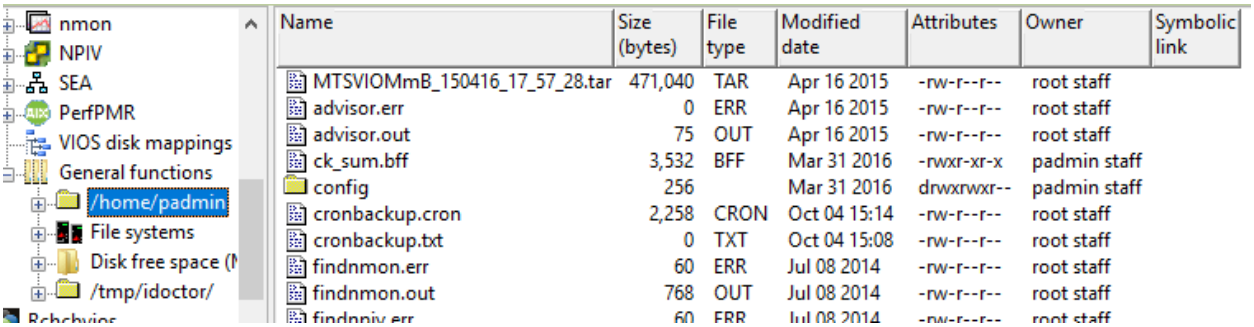
Expanding the General functions folder

Folder	Description
/home/padmin	Allows the user to browse the file system and defaults to the user's home directory. Since VIOS connections require the use of padmin this will be /home/padmin. **Tip:** The value show for this folder's name is a preference and can be changed by right-clicking the folder and using the Change Directory menu option.  *Change Directory menu*
File systems	Displays the file systems for the current AIX, VIOS, Linux system and allows the user to work with them.
Disk free space	Displays disk free space report for the current system.
/tmp/idoctor	This lets the user work with the iDoctor build directory. This is where all the scripts that iDoctor loads when first connecting to the system are stored. Tip: The value for this folder's name is a preference that can be changed in the **Preferences -> Power -> Remote script directory** setting. 

7.5.1 Files and directories

Many folders exist in the Power Connections interface that let users work with directories and files on the current system they are using. For example, expanding the /home/padmin/ folder will show a view like

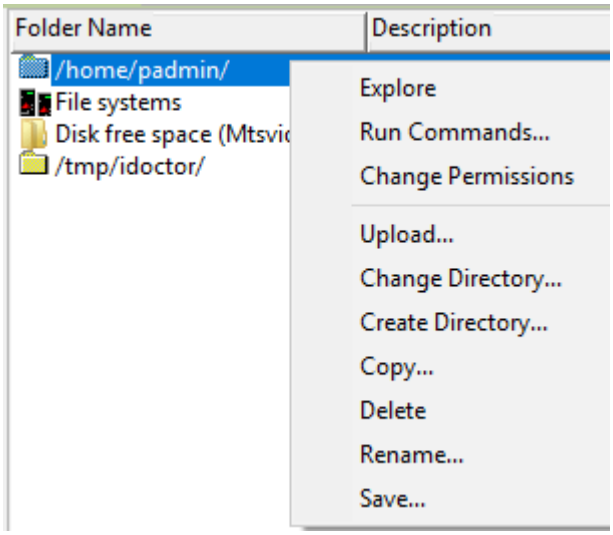
this with files and subdirectories that you can work with. This is designed to be very similar to typical “ls -l” command output on AIX or VIOS systems.



/home/padmin folder’s contents

7.5.1.1 Directory menu options

The default right-click menu options available for the /home/padmin folder or any folder / subfolder in the Power Connections view are:

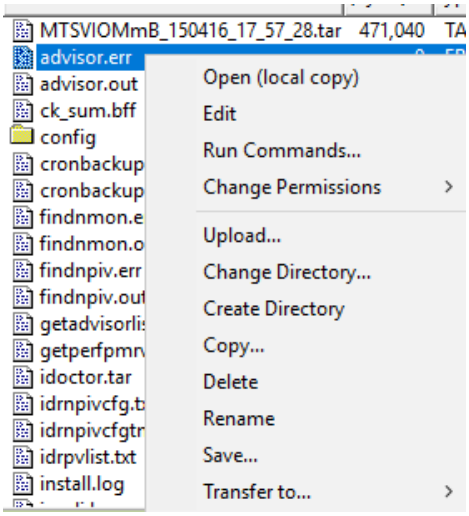


/home/padmin directory menu options

Menu	Description
Explore	Expands the current folder and displays the results in the list.
Run commands…	This lets you run one or more commands on the system you are working with. The results of running these commands will be shown in notepad on the PC.
Change Permissions	Several sub menu options exist to either grant or revoke authorities on the current folder. **Note:** This will only work if the currently connected user can do this.
Upload	This will present an interface to transfer one or more files from the PC to the remote system.
Change Directory	This option allows the user to change the directory they are working with by default. **Note:** This option only applies to the /home/padmin folder under the General functions folder.
Create Directory	This command will issue a mkdir command on the remote system to create the desired directory.
Copy…	This will present an interface that allows the user to copy the directory and all of its contents to another location on the system.
Delete	This will delete the directory from the system.
Rename	This option will rename the directory.
Save	Use this option to save the directory and its contents to a tar file.

7.5.1.2 File menu options

The default menu options for a file in the Power Connections view are:



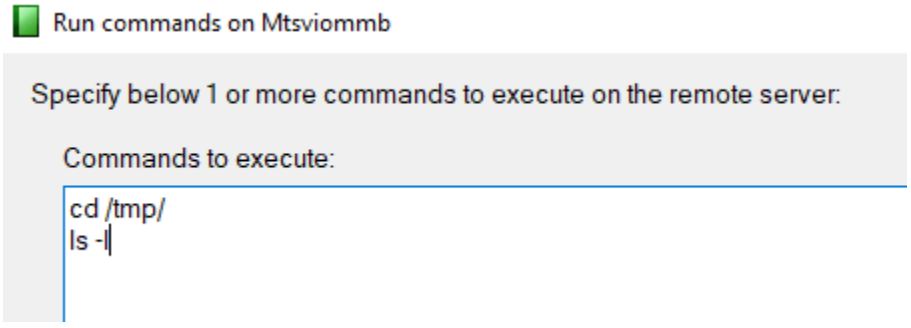
File menu options

Tip: Additional menu options apply to .nmon, .npiv and .sea files for Analysis purposes.

Menu	Description
Open (local copy)	This option will first download the file to the PC and then attempt to open it using the default program for the current file type.
Edit	This option will download the file to the PC and then present an interface that lets you edit the results. When finished the file is sent back to the remote system. **Tip:** Only use this option for text files
Analyze Data	This option will display the Analyze Data window which allows .nmon, .npiv and/or .sea files to be analyzed and sent to DB files in the desired collection.
Run commands…	This lets you run one or more commands on the system you are working with. The results of running these commands will be shown in notepad on the PC.
Change Permissions	Several sub menu options exist to either grant or revoke authorities on the current selection. **Note:** This will only work if the currently connected user can do this.
Upload	This will present an interface to transfer one or more files from the PC to the remote system.
Change Directory	This option allows the user to change the directory they are working with by default. **Note:** This option only applies to the /home/padmin folder under the General functions folder.
Create Directory	This option allows the user to create a directory on the system.
Copy…	This will present an interface that allows the user to copy the selection to another location on the system.
Delete	This will delete the selection from the system.
Rename	This option will rename the currently selected file.
Save	Use this option to save the selection to a tar file.
Transfer to	This option presents an interface that allows the user to transfer the current file(s) to another system, to IBM or to the PC.

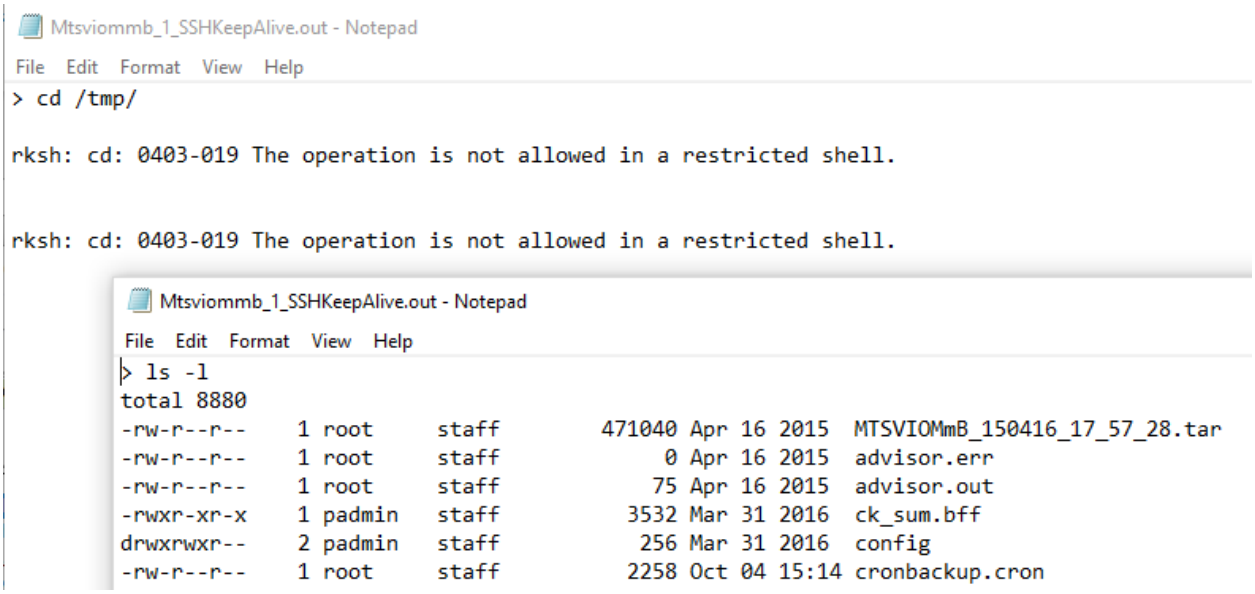
7.5.1.3 Run Commands

This interface simply allows a user to run 1 or more commands on the remote system using the current SSH connection. Keep in mind that these commands will only work if the syntax is correct and applicable for the type of system currently being used. On VIOS the commands run in a restricted shell state, so some commands cannot be used here.



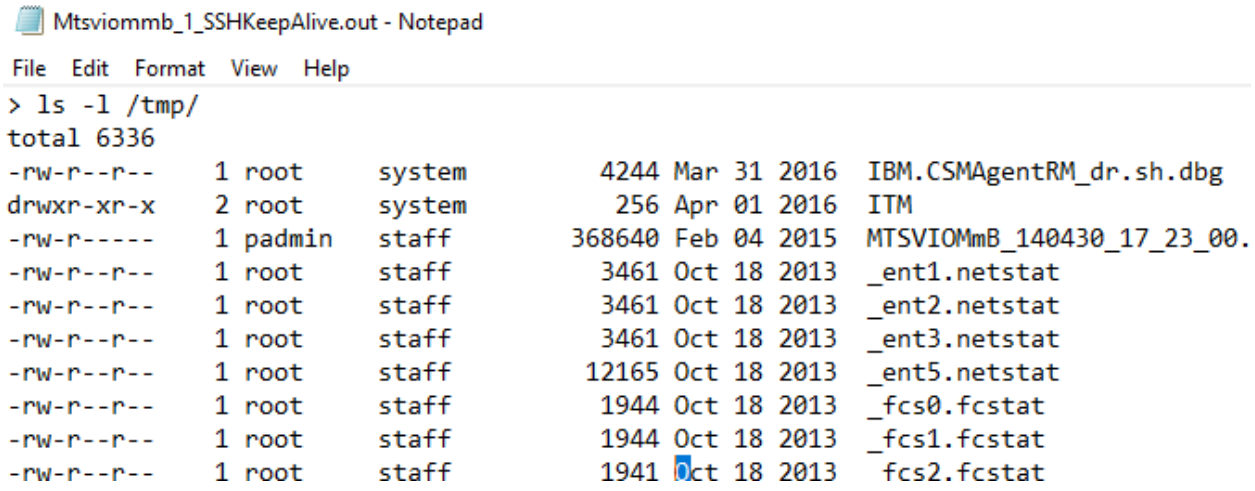
Run commands on <system> window

Each command will run and the results of each will be displayed in a notepad window. For example, the above commands will result in the following:



Result of running 2 commands

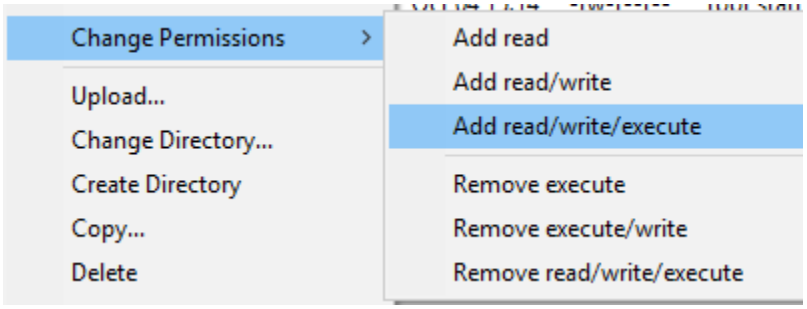
Note: On VIOS many commands such as “cd” are not allowed in the restricted shell. In this situation the command would need to be written as ls -l /tmp/



Result of running ls -l /tmp/

7.5.1.4 Change Permissions

These options allow the user to add or more permissions for the current file(s) or directory(ies) selected.



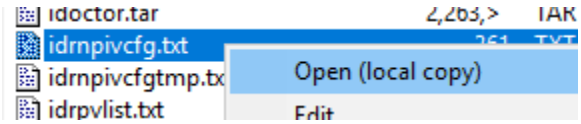
Change Permissions menu

This will issue the appropriate chmod command in the Remote Command Status view.

7.5.1.5 Open (local copy)

This option is used to download and then execute the default Windows program for the file type being opened. For example, using this option on a .txt file would probably open the file in Notepad.

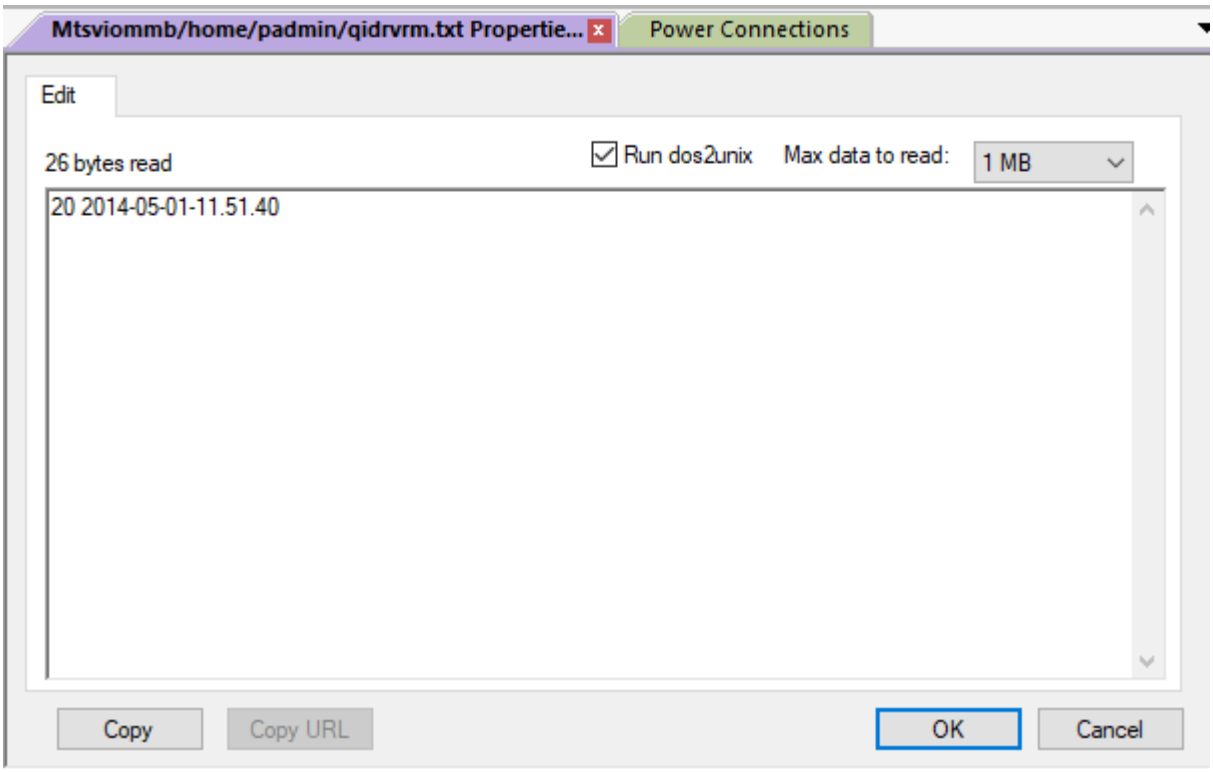
Tip: Files are downloaded by default into the iDoctor temp directory.



Open (local copy) menu

7.5.1.6 Edit

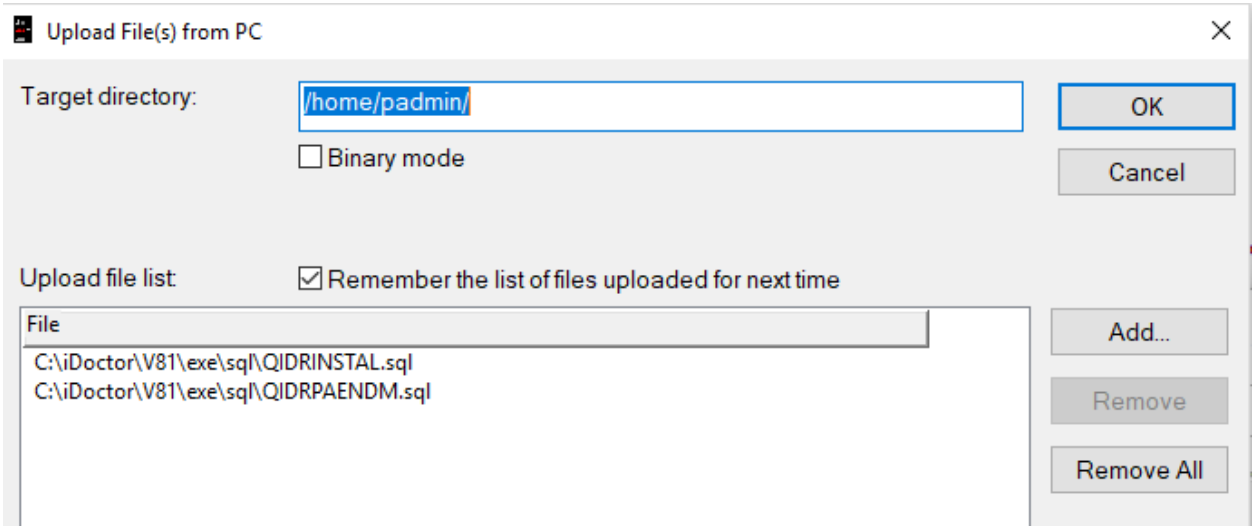
This option can be used to download the file to the PC and then open it within an editor inside of iDoctor.



Option	Description
Run dos2unix	This option will run the dos2unix command on Linux systems after the file has been edited and then transferred back to the system. **Note:** It does NOT appear unless connected to a Linux system.
Max data to read	This indicates how much data is downloaded to the PC from the file. The possible values are 1 MB, 10 MB and 100 MB. **Tip:** You should not edit anything other than text files with this interface and only if the entire data has been read.
OK	If changes have been made this will send the file back to the remote system and replace it.

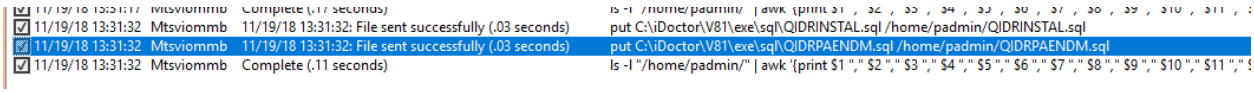
7.5.1.7 Upload File(s) from PC

The Upload file(s) window allows the user to transfer 1 or more files from the PC to the desired directory on the remote system. These transfers occur via the Remote command status view and the results of which can be viewed there.



Upload File(s) from PC Window

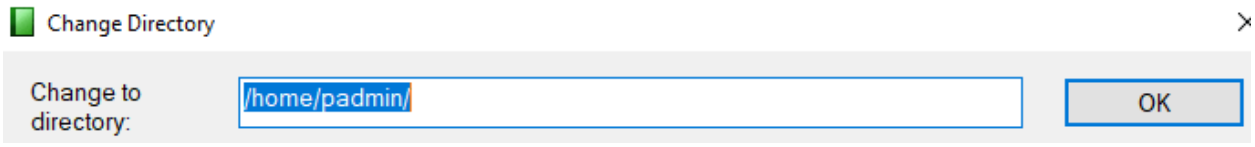
Option	Description
Target directory	This value contains the desired location to send the file(s) on the PC to.
Binary mode	Check this box if you wish to use binary mode to send the files.
Remember the list…	Use this option is you want the GUI to remember the list of files and repopulate the list again with the same set of files. This can be handy if sending the same files to multiple LPARs.
Upload file list	This is the list of files on the PC that will be transferred to the remote server.
Add button	Use this button to add files from the PC to the list.
Remove button	This button removes the selected files from the list.
Remove all button	Removes everything from the list.



Results in Remote Command Status View

7.5.1.8 Change Directory

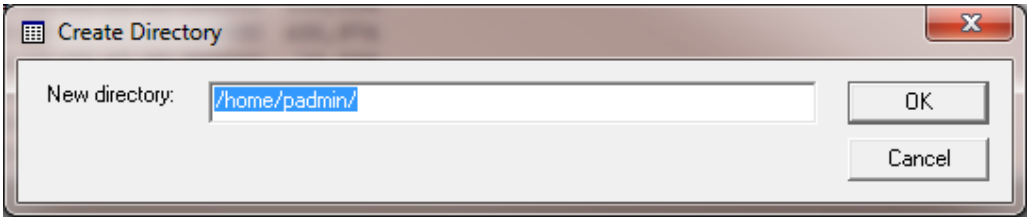
This option is used to change the current directory used for the first folder under the General functions folder. If selecting a directory and using this menu the default value listed will be directory selected.



Change Directory Window

7.5.1.9 Create Directory

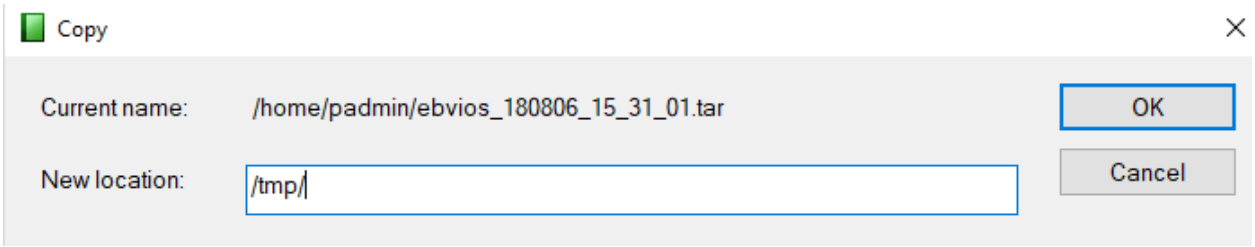
This option is used to create a directory on the remote server. By default, the window will show the path for the current directory and you will need to modify this path appropriately.



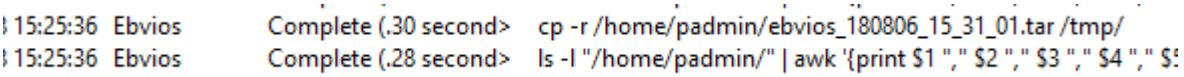
Create Directory Window

7.5.1.10 Copy

This option is to copy one or more files/directories to the desired new location. Only if specifying a single file may be the value entered a file name. Otherwise the value provided should be the directory to copy the selections into.



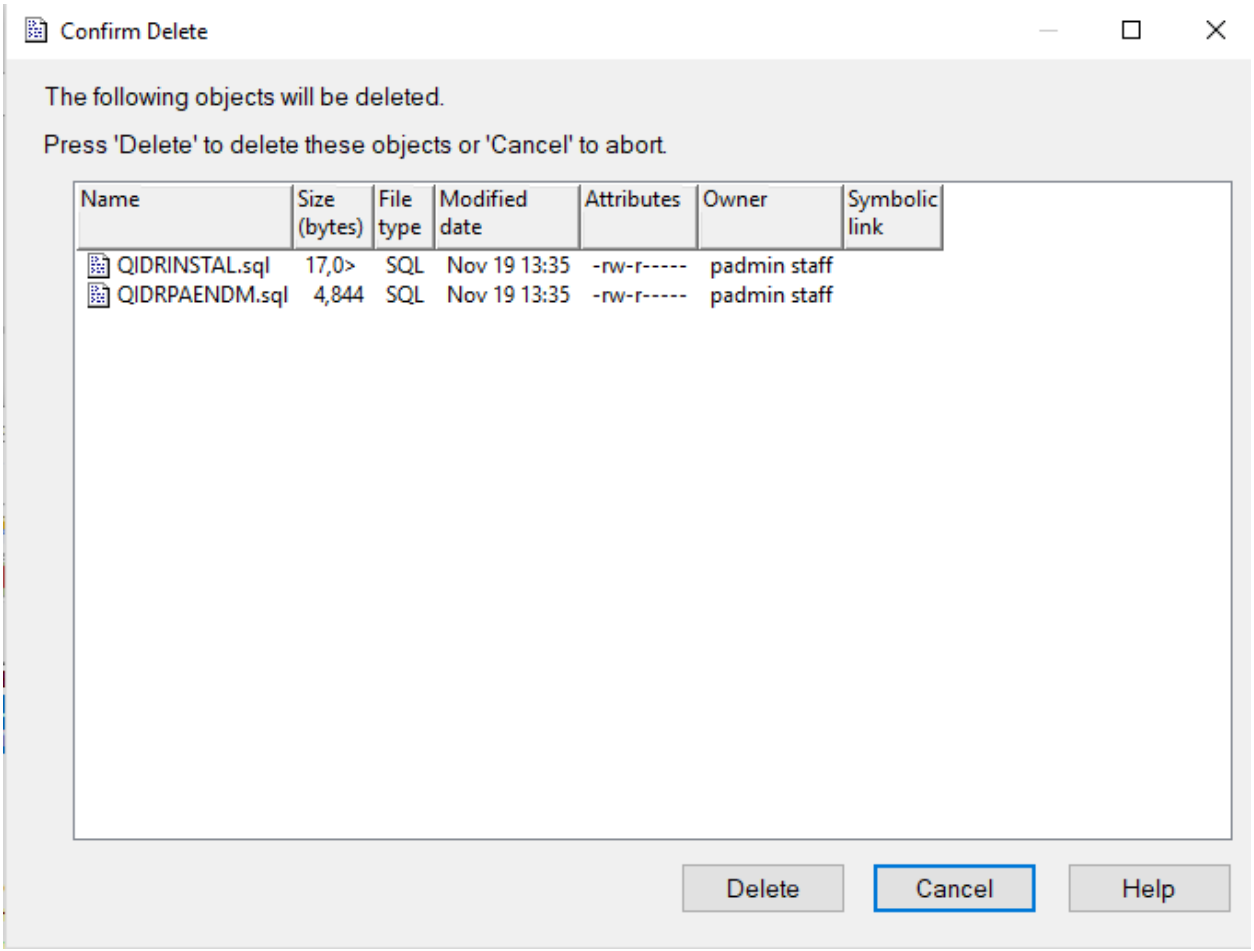
Copy Window



Remote Command Status View Copy

7.5.1.11 Delete

The Delete option will remove the file(s) and/or directories and their contents from the remote server.

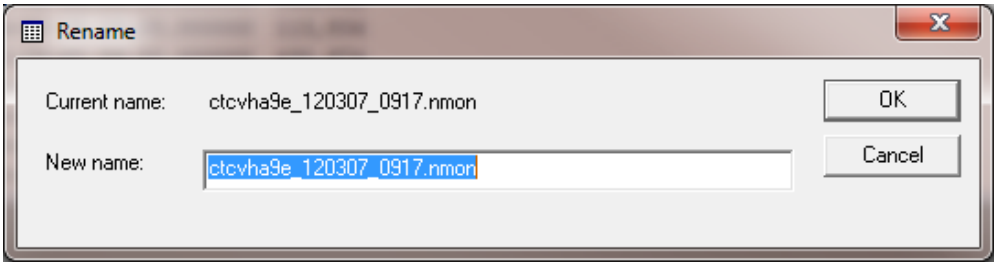


Confirm Delete Window

7.5.1.12 Rename

The Rename menu option allows you to change a specific file or directory and give it a new name.

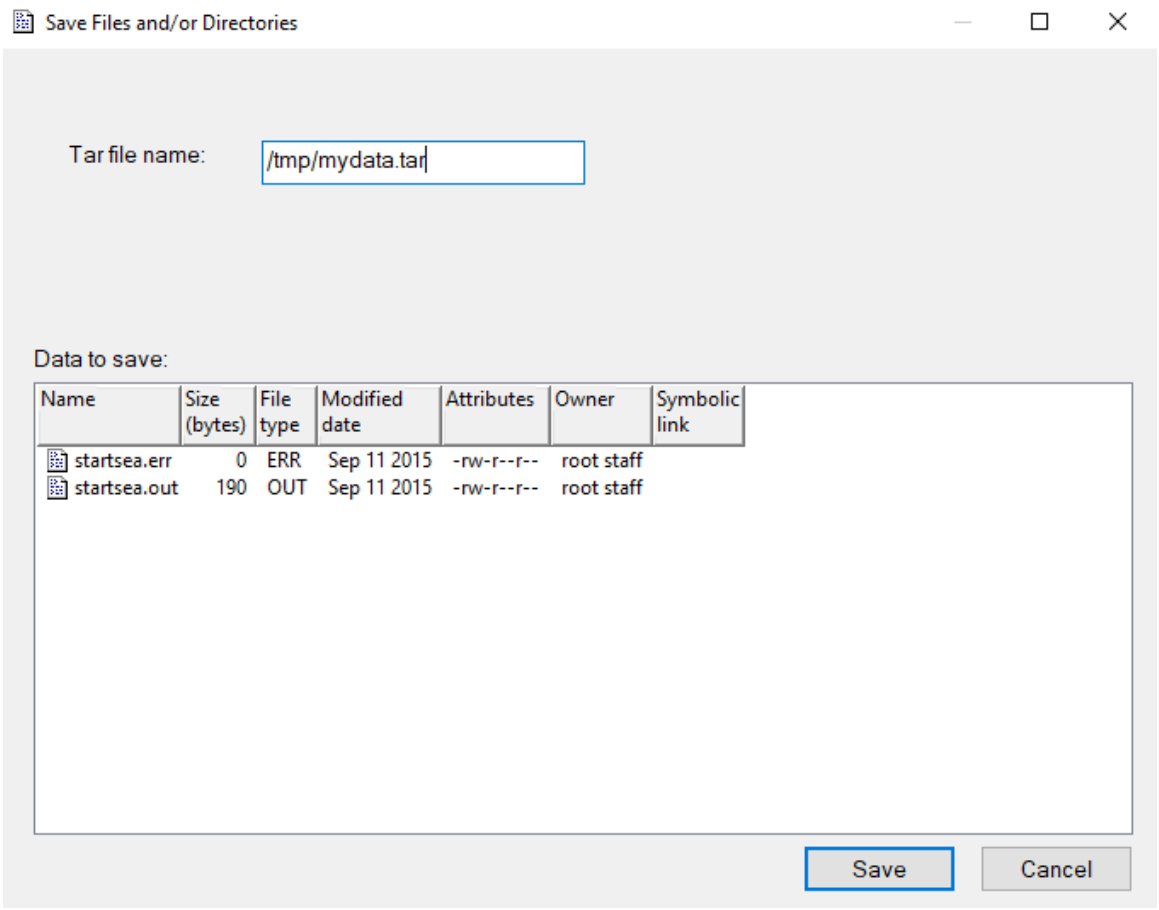
The interface looks like this:



Rename Window

7.5.1.13 Save

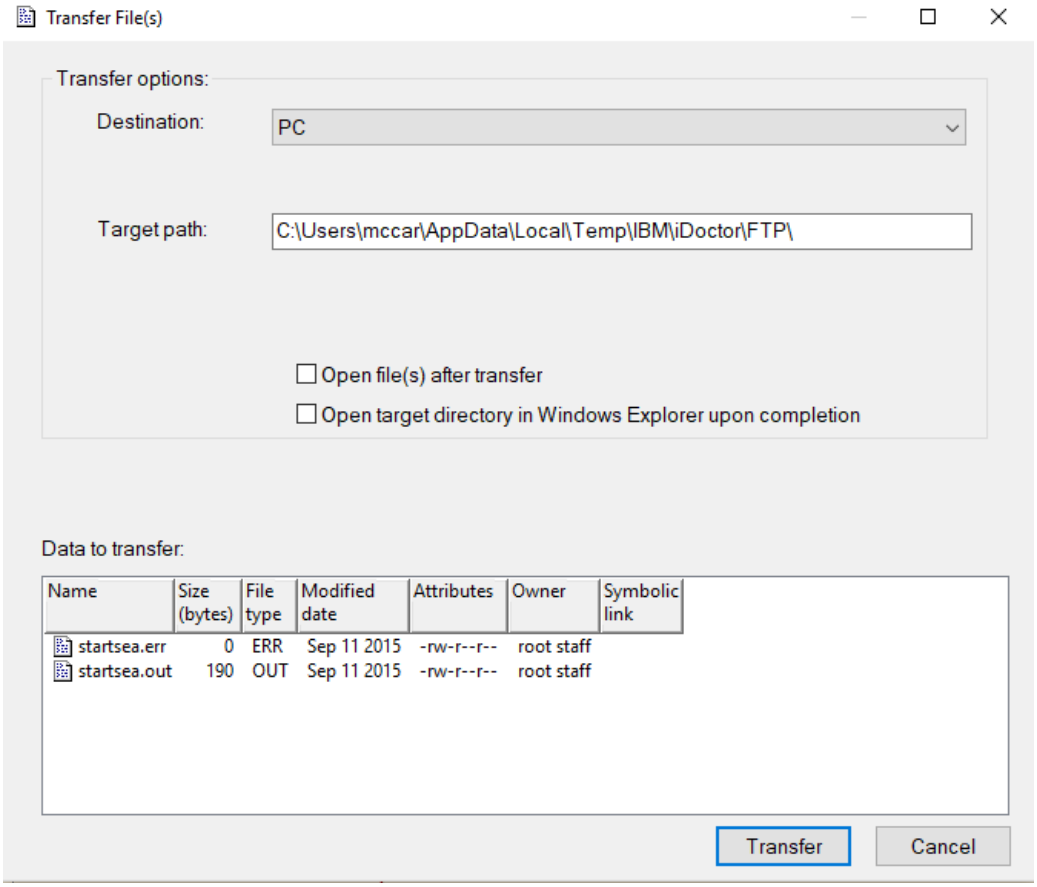
This option will save all selections to a tar file that you specify.



Save Files and/or Directories Window

7.5.1.14 Transfer to -> PC

This option can be used to download the selected file(s) to the PC. The files will be placed in the iDoctor temp directory's FTP subdirectory by default.

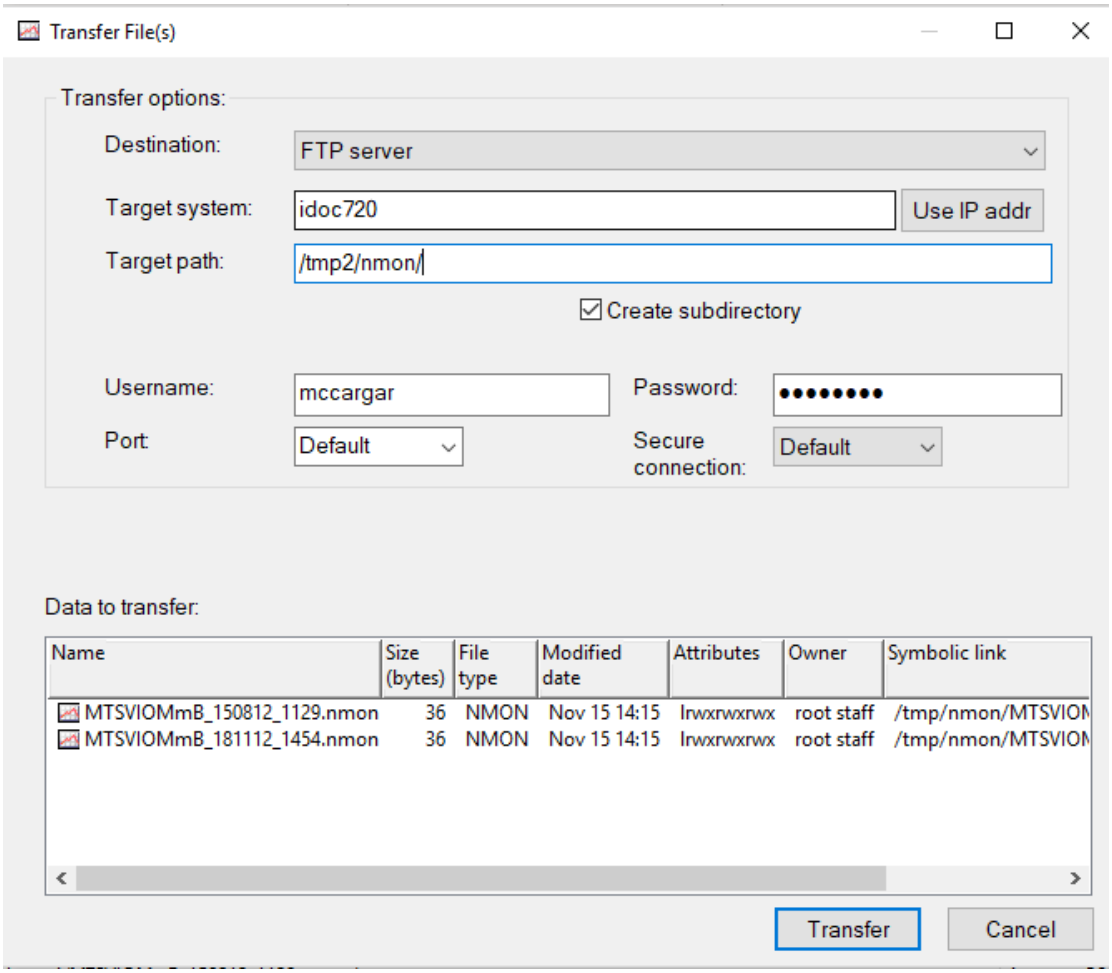


Transfer File(s) Window

Option	Description
Destination	This indicates where the files will be sent.
Target path	The directory on the PC to send the files to. Only if transferring a single file would this be a filename instead of a directory.
Open file(s) after transfer	This option will open each file (in Notepad) after they are transferred to the PC. **Tip:** Avoid using this option if you are transferring many files!
Open target directory in Windows Explorer	If checked, then after the download completes, Windows Explorer will be automatically opened to show the files downloaded on the PC.

7.5.1.15 Transfer to -> FTP Server

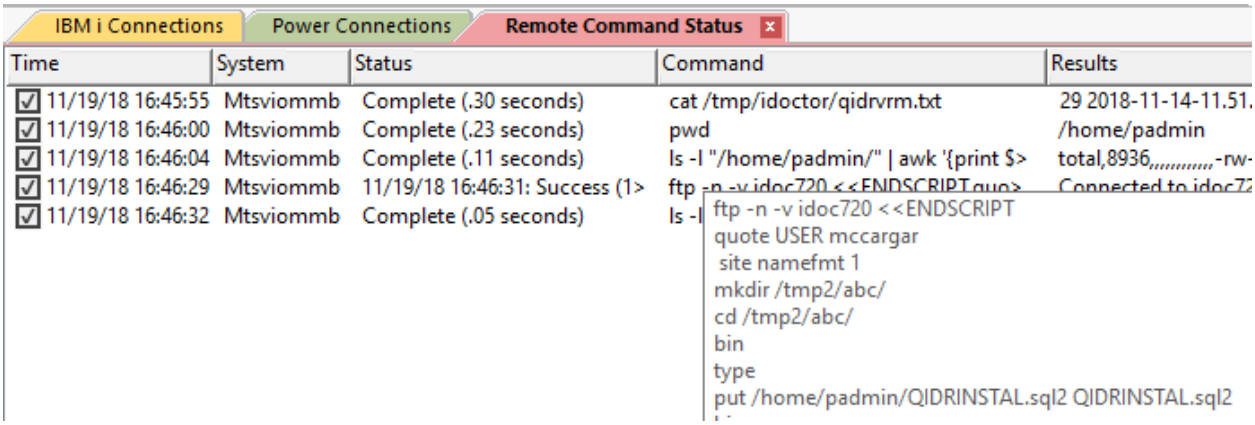
This option will send the selected files to the specified target system in the path indicated. This occurs by running the ftp command on the current remote server to the target system. The files will be sent directly from the system you are working with to the target system.



Transfer File(s) -> FTP server

Option	Description
Destination	This indicates where the files will be sent.
Target system	The system to send the selected files to.
Target path	The directory on the target system to send the selected file(s) to.
Create subdirectory	This will create the directory specified if it does not exist already. **Note:** This only works for a single directory, it will NOT create multiple directories in one shot.
Username	The user profile to use when making the connection to the remote server.
Password	The password to use when making the remote connection. Specify a password such as your email address if connecting with the “anonymous” user.
Port	This the port to use when making the FTP connection and should be either Default or Secure. If Secure is used, then the ftp -s parameter will be used to enable a secure TLS/SSL.
Secure connection	Select SSL for this value if you wish to make a secure connection (or use Secure for the Port parameter value.)
Data to transfer	This is the list of files to send to the target remote system.

After pressing the Transfer button, the required command will be sent to the Remote Command Status View and executed there. A scripted ftp command is used within the SSH connection that the Remote Command Status View provides.

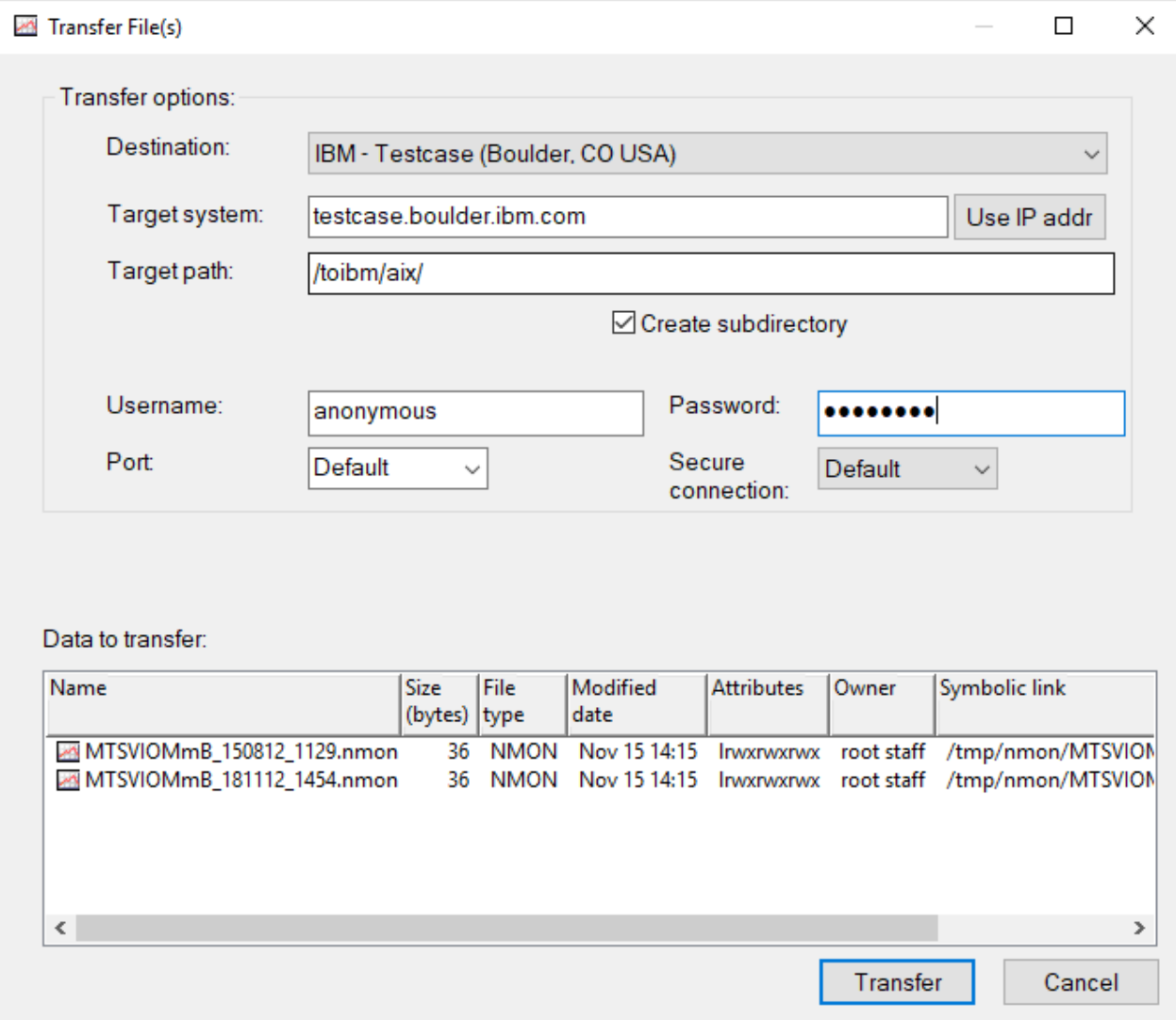


FTP script used to execute a file transfer from a VIOS to an IBM I within the Remote Command Status View

Tip: The results of running the script can be seen in the Results column.

7.5.1.16 Transfer to -> IBM - Testcase

This option will send the selected files to IBM's testcase system in Boulder, CO. This occurs by running the ftp command on the current remote server to the target system. The files will be sent directly from the system you are working with to the target system. Typically, you can use the username anonymous and password of your email address when making this connection.



Transfer File(s) -> IBM - Testcase

Option	**Description**
Destination	This indicates where the files will be sent.
Target system	The system to send the selected files to.
Target path	The directory on the target system to send the selected file(s) to.
Create subdirectory	This will create the directory specified if it does not exist already. **Note:** This only works for a single directory, it will NOT create multiple directories in one shot.
Username	The user profile to use when making the connection to the remote server.
Password	The password to use when making the remote connection. Specify a password such as your email address if connecting with the “anonymous” user.
Port	This the port to use when making the FTP connection and should be either Default or Secure. If Secure is used, then the ftp -s parameter will be used to enable a secure TLS/SSL.
Secure connection	Select SSL for this value if you wish to make a secure connection (or use Secure for the Port parameter value.)
Data to transfer	This is the list of files to send to the target remote system.

7.5.1.17 Transfer to -> IBM – ECUREP

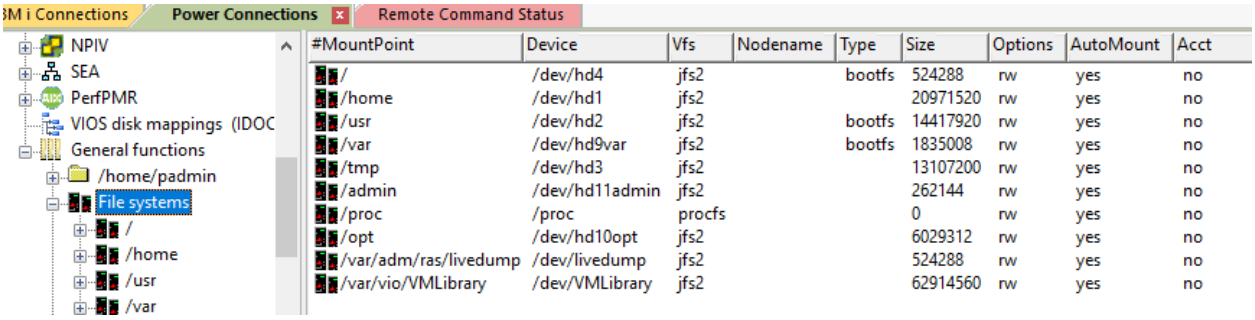
This option will send the selected files to IBM's ECUREP system in Germany. See the previous section for more information on this interface.

7.5.1.18 Transfer to -> IBM – Blue Diamond Lab

These options will send the selected files to IBM's Blue Diamond Lab. A secure FTP connection must be used when making the connection. See the previous section for more information on this interface.

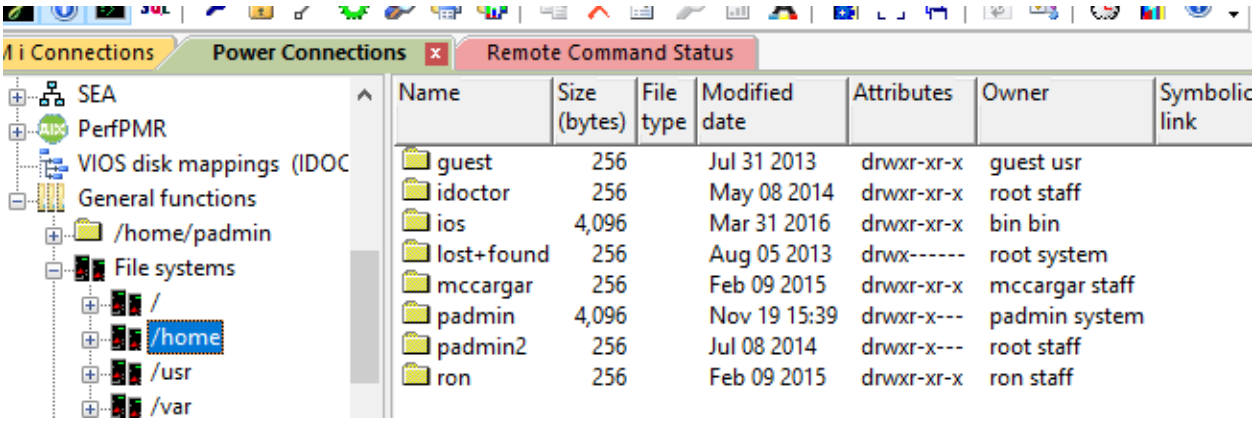
7.5.2 File Systems

The General functions -> File systems folder provides a graphical interface for the lsfs command on the VIOS or AIX system. This will display all file systems defined on the current system and allows you to work with each one.



Power Connections -> VIOS -> General functions -> File systems folder

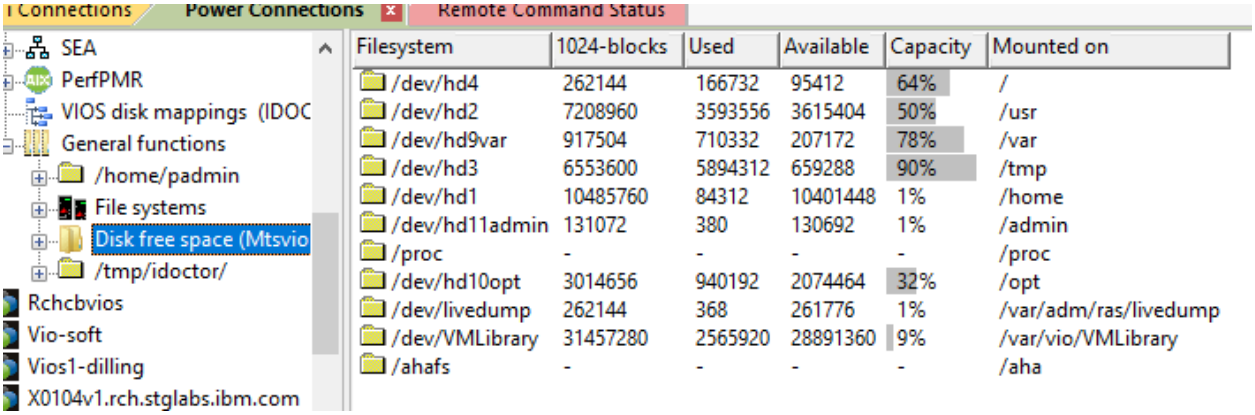
Expanding a file system will show its contents and the data within can be worked with using the features discussed in the previous section.



File system /home

7.5.3 Disk free space (system)

This option will display the disk free space for each file system on the server. The view is produced by running df -kP command. Expanding any of these file systems will display its contents.



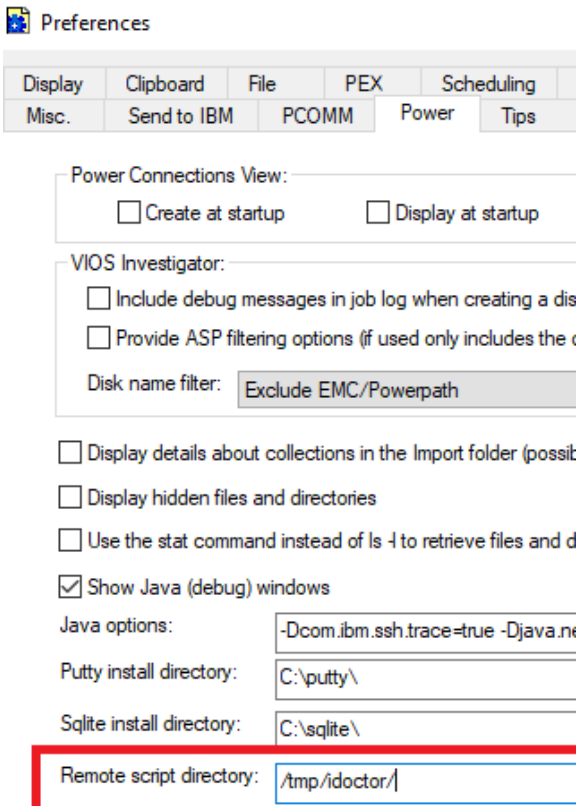
Disk free space folder

7.5.4 /tmp/idoctor

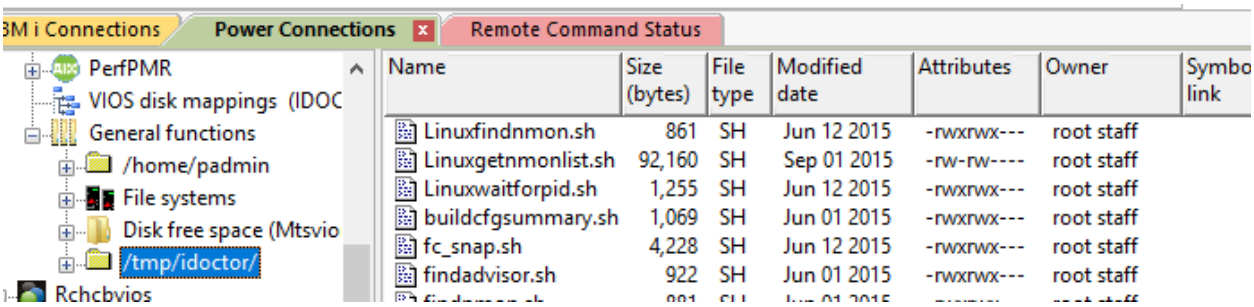
This folder is created by the iDoctor GUI the first time you make a connection to the AIX, Linux or VIOS system from the Power Connections interface. This folder contains the scripts that are needed to find, view and collect nmon, npiv or SEA statistics. Typically, this folder is for IBM use only.

The version information for the scripts in this folder are stored in file qidrvrm.txt.

Tip: The value for this folder's name is a preference that can be changed in the **Preferences -> Power -> Remote script directory** setting.



Preferences -> Power -> Remote script directory setting



/tmp/idoctor folder example

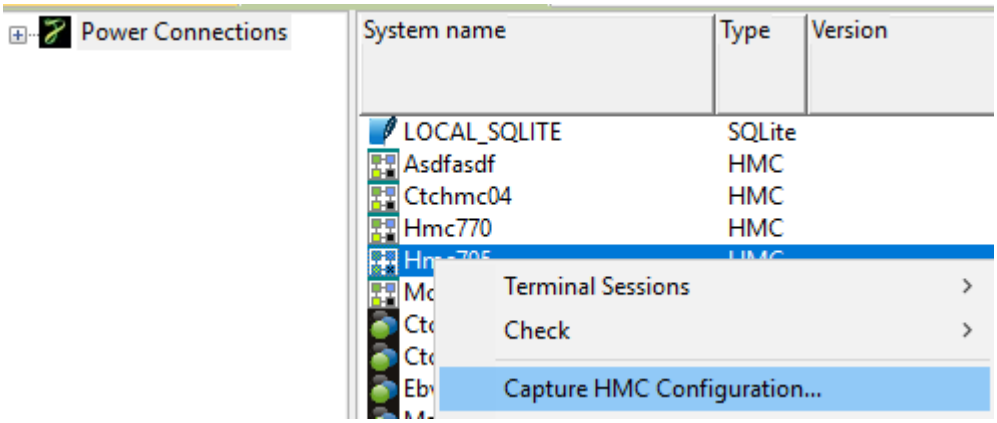
7.6 Collecting Data

When dealing with the non-IBM i systems there are several types of data that can be collected, and these can be captured in various ways.

7.6.1 Capture HMC Configuration

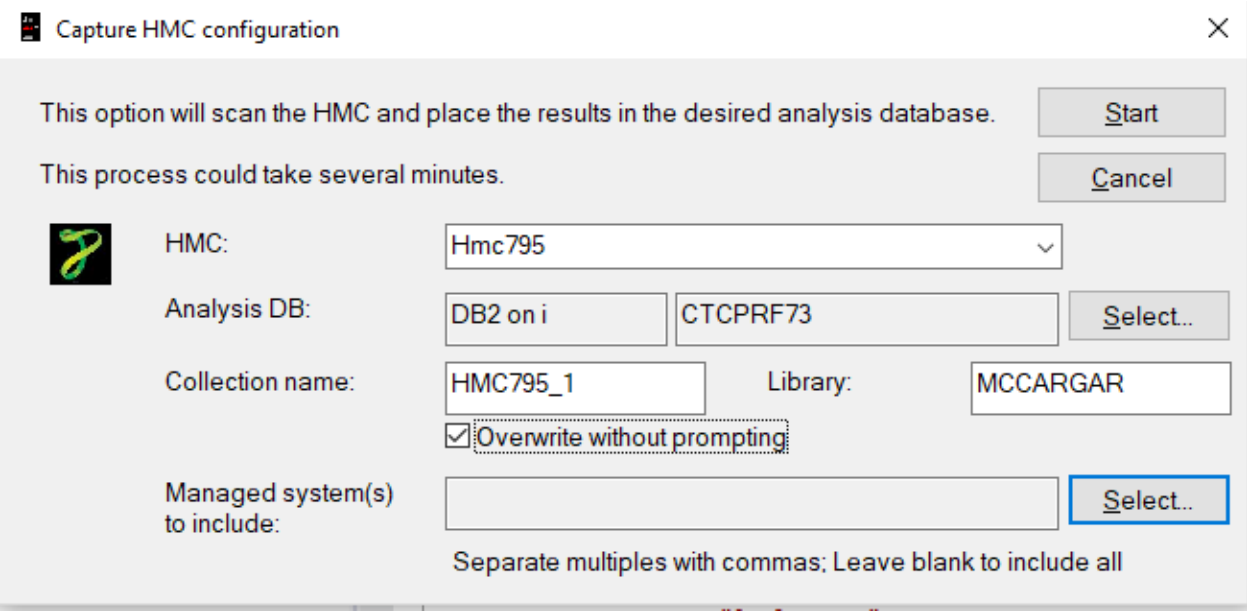
When working with HMC connections you can kick off a collection of the HMC's configuration to be saved and reviewed later with iDoctor.

To access this feature right-click an HMC in the Power Connections interface and use the Capture HMC Configuration... menu.



Power Connections -> HMC -> Capture HMC Configuration menu

The Capture HMC Configuration window looks like this:

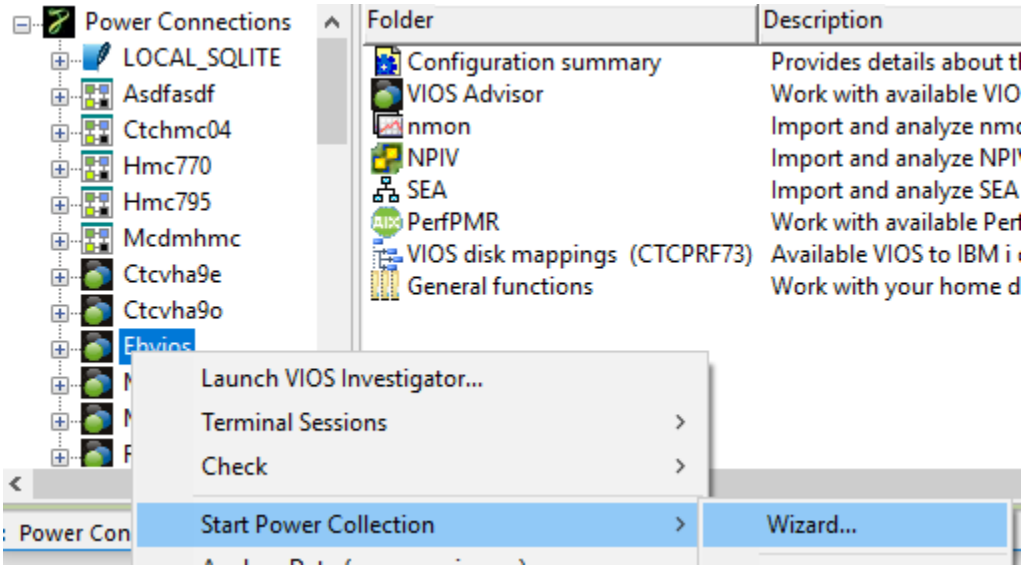


Capture HMC configuration

Option	Description
HMC	The name of the HMC to capture data from. Only HMCs in the Power Connections view will be listed here. Add more HMCs to the Power Connections view if necessary.
Analysis DB	Indicates where the data will be sent to for analysis. The two options are either an IBM i or an SQLite DB (ODBC connection) on the PC. Use the select… button to change this setting.
Collection name	The name of the collection within the analysis DB to store the configuration results.
Library	This is the library name to store the collection in. **Note:** This only applies when the Analysis DB is an IBM i.
Overwrite without prompting	Replaces the existing collection without warning if checked.
Managed system(s) to include	This field is a comma separated list of managed systems on the HMC to capture data for. You must use the Select… button to configure which to include. If the field is blank then all managed system(s) will be included in the results.

7.6.2 Power Collection Wizard

You can collect data once using the Power Collection Wizard. This lets you have more control to where the data will be stored. To access this feature right-click a system in the Power Connections interface and use the Start Power Collection -> Wizard… menu option.

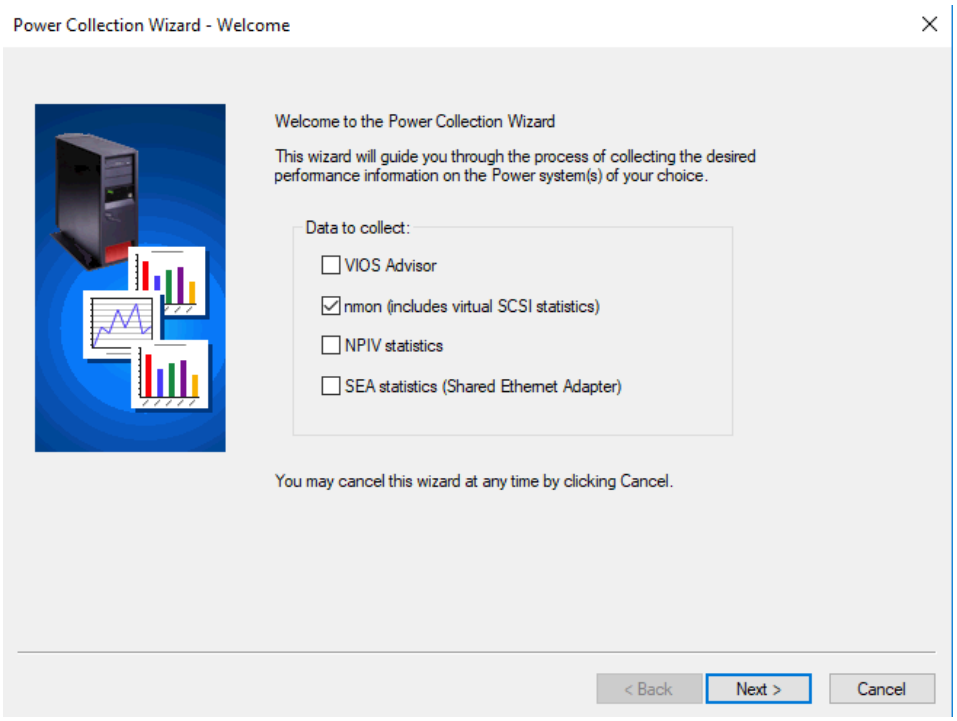


Start Power Collection -> Wizard menu

The Power Collection Wizard allows a user to remotely start a collection of the desired type(s) against one or more non-IBM i systems (AIX/VIOS/Linux.) This will only work if SSH connection(s) to each system are available from the current PC.

The next section covers the interface:

7.6.2.1 Welcome

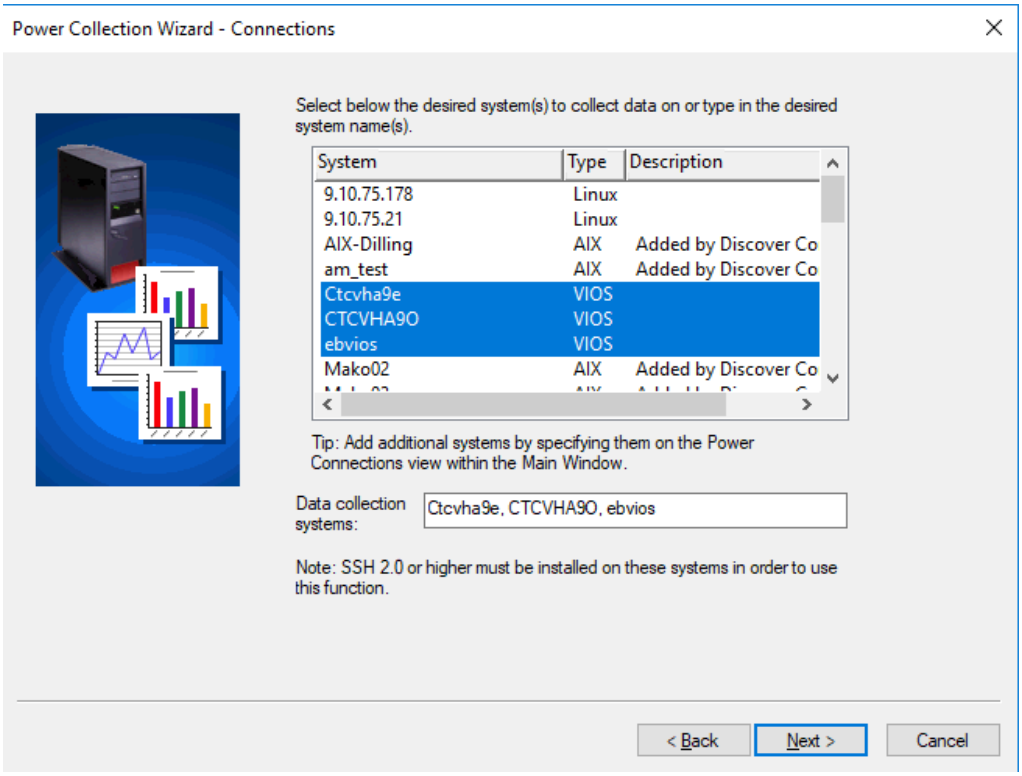


Welcome panel

This screen gives the user the option to 1-4 different types of data to collect. You can pick all of them or only 1. Keep in mind that some of these options only work and apply on VIOS and all of them were out recently tested on VIOS.

Note: The NMON option by default will include virtual SCSI statistics.

7.6.2.2 Connections



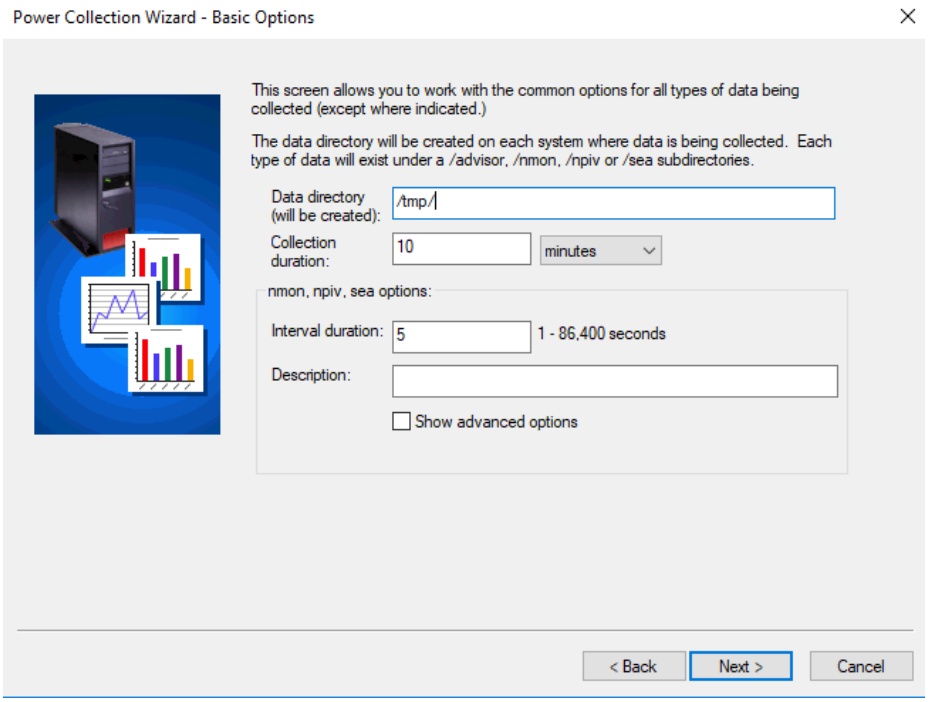
Connections panel

The Connections panel asks the user to specify the non IBM-i system(s) to collect the data on. You can select more than 1 system here by using the Ctrl key while clicking the mouse.

Note: SSH 2.0 or higher must be installed on these systems.

Option	Description
System list	This list by default contains a list of all types of connections from the Power Connections View. It is recommended to add your systems to the Power Connections View first before using this interface.

7.6.2.3 Basic Options



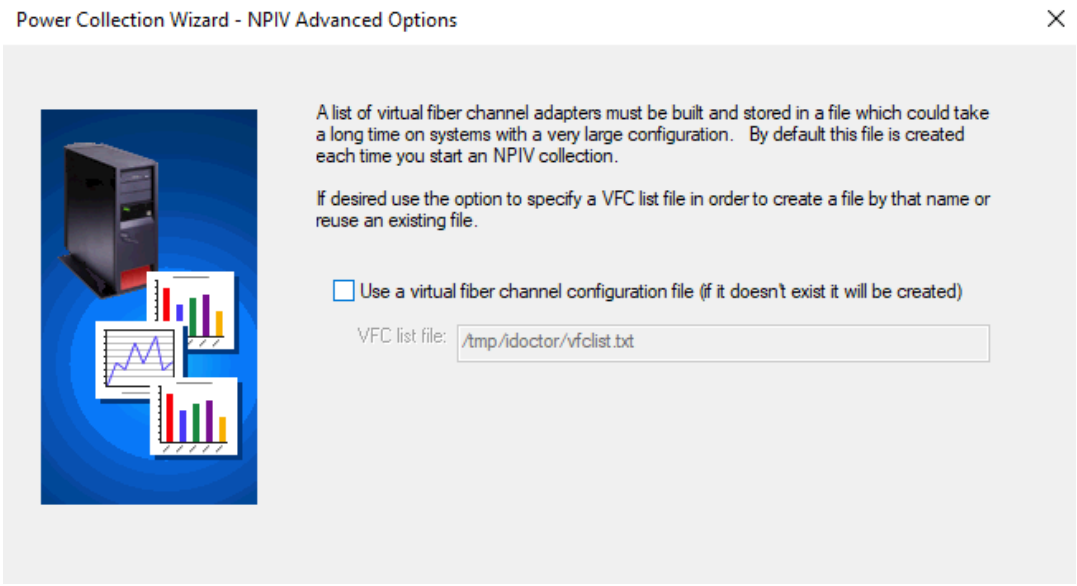
Basic Options panel

The Basic Options panel asks the user for key information about the data being collected such as the directory to create data in, interval duration and the number of intervals to collect.

You may also check the box to show additional advanced options if desired.

Option	Description
Data directory	This is the root directory on each type of data created will be stored. Each type of data is stored in a subdirectory under this directory based on the type (/nmon, /npiv, /sea, etc) created.
Collection duration	Indicates how long the collection(s) should run for. This setting can be entered in hours, minutes or seconds using the drop-down list. **Note:** Keep in mind if desired it is possible to analyze the data in these types of collections before the collection duration has been reached.
Interval duration	This value specifies how much of a delay there is between intervals (in seconds).
Description	A description to associate with the collection(s) created.
Show advanced options	Check this box to see additional data collection options that are specific to nmon or npiv. **Note:** Not checking this option will take the user directly to the Finish panel.

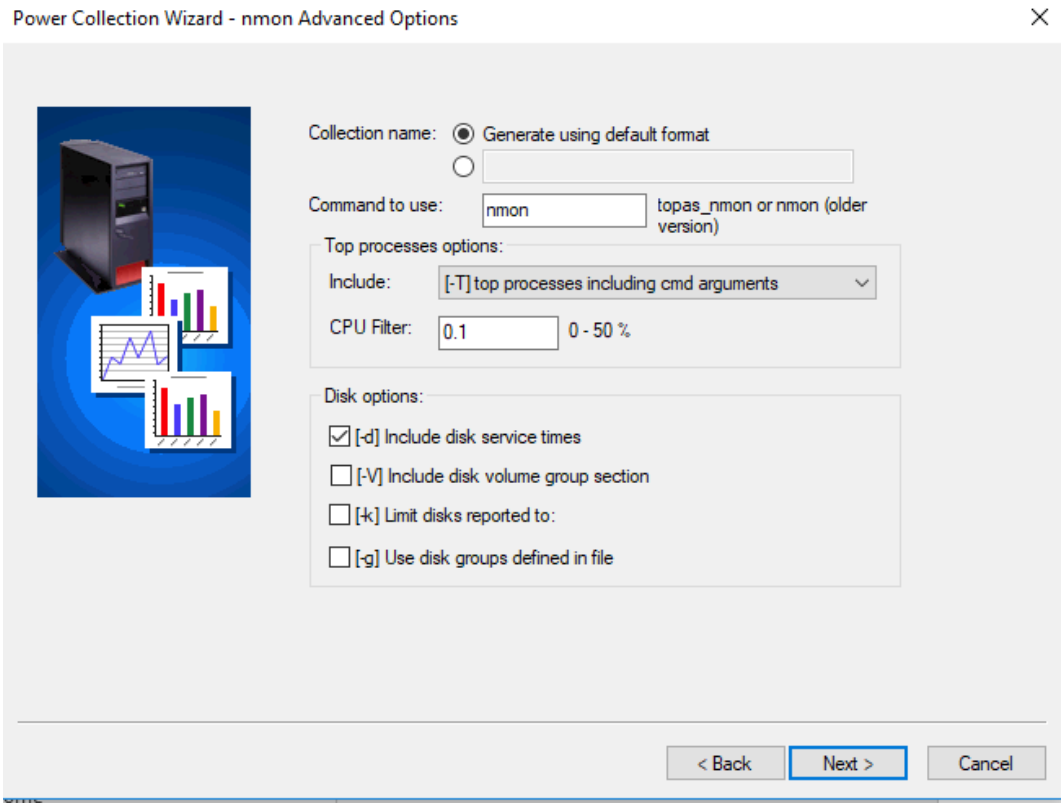
7.6.2.4 NPIV Advanced Options



NPIV Advanced Options panel

This screen allows the user to specify if a list of virtual fiber channel adapters will be generated by the NPIV data collection script. If this is not checked (the default), then the list of virtual fiber channel adapters will be determined by the script. It will be collected once at the start of collection and not checked again during collection.

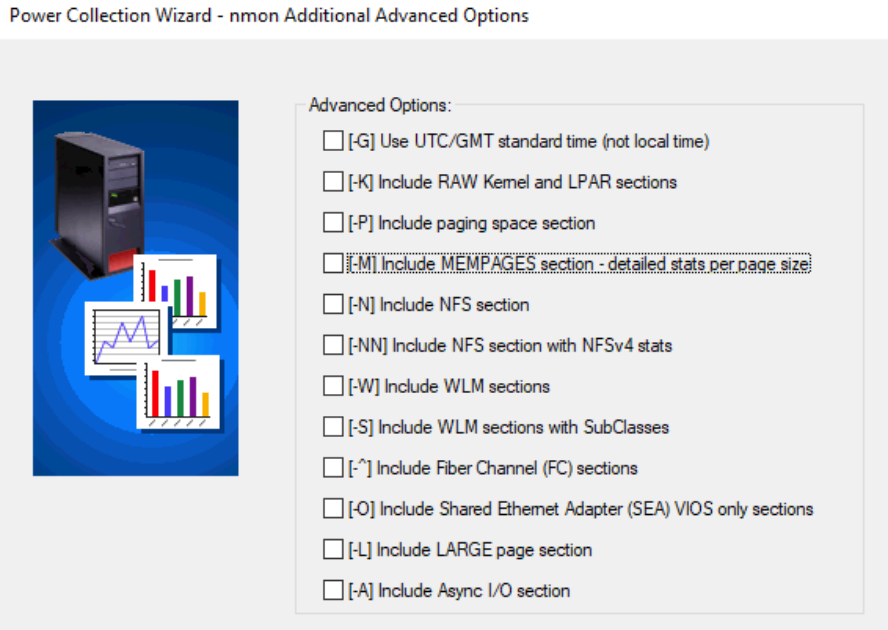
7.6.2.5 NMON Advanced Options



NMON Advanced Options

GUI Element	Description
Collection name	This if the file name to use for the NMON collection. Using the "generate using default format" is recommended. That format will contain the system name and timestamp automatically within the name. If specifying your own collection name, the name must end with.nmon.
Command to use	By default 'topas_nmon' should be used. But if you are on an older version of AIX where NMON is not automatically installed (via the topas_nmon command) and you'll installed NMON yourself on the AIX IBM it is possible to change this value to 'nmon' and it probably will work.
Top processes options: Include	These options indicate whether TOP processes will be included by NMON and which option to use. The options available are: None (-t) Includes top processes (-T) Includes top processes and saves command line arguments in the UARG section. (-Y) Includes top processes and adds together all commands with the same name.
Top processes options: CPU filter	Specifies the percentage of process threshold at which the command ignores the TOP processes statistics. The default percentage is .1%. The command does not save the TOP processes statistics if the process is using less processor than the given percentage.
Disk options: (-d) Include disk service times	If checked includes disk service times. Depending on the version of NMON used you may get read service time (DISKREADSERV) and write service time (DISKWRITESERV) separately or you may get them added together into 1 value (DISKSERV). iDoctor will attempt to utilize either appropriately. If you do NOT include this option in your collection then the graphs will not show response times and service times.
Disk options: (-V) Include disk volume group section	If checked includes the disk volume group section.
Disk options: (-k) Limit disk reported to	This value if used will filter the disk data to only include those disk names that match the comma separated list provided. For example you could specify: hdisk1, hdisk2 in the text box next to this value to only include hdisk1 and hdisk2 in the disk graphs provided.
Disk options: (-g) Use disk groups defined in (remote) file	Specifies the file (on the remote system) that contains the user-defined disk groups, using the *filename* parameter. Each line in the file begins with a group name. The list of disks follows the group name and is separated with spaces. The file can contain a maximum of 64 disk groups. A disk can belong to various disk groups.

7.6.2.6 NMON Additional Advanced Options



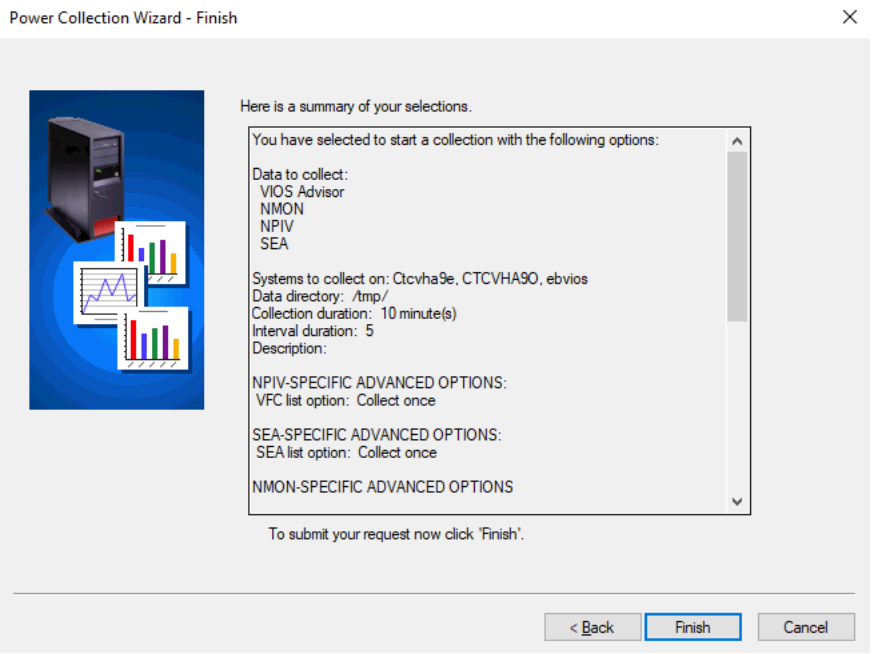
Advanced Options panel

The Advanced Options panel lets the user specify additional parameters when creating the NMON collection.

For more information on all the parameters available (in recording mode) visit this page:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.cmds%2Fdoc%2Faixcmds4%2Fnmon.htm

7.6.2.7 Finish



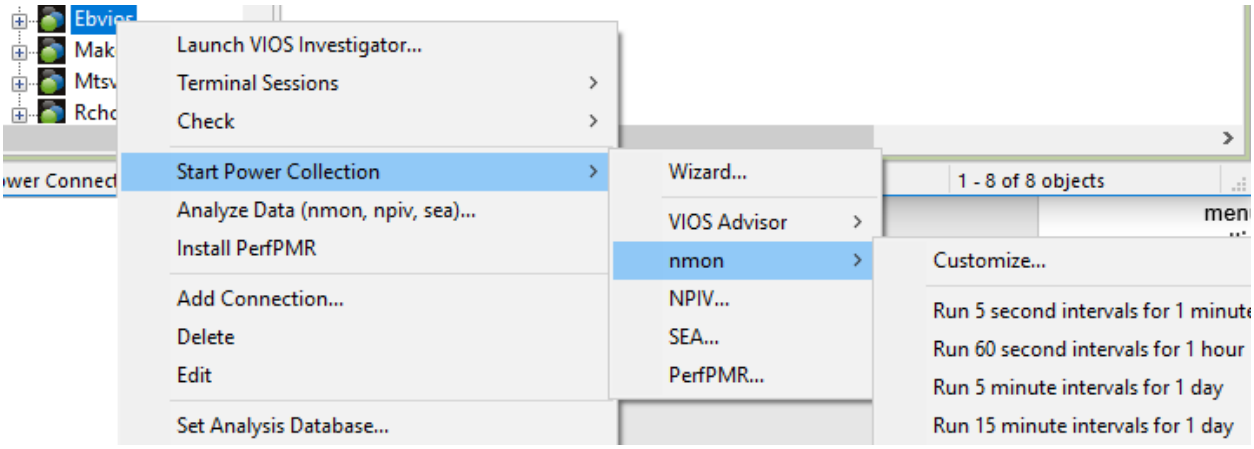
Finish panel

This panel summarizes your selections and includes the remote command string(s) that will be issued on the remote system(s).

After hitting finish this should start the collections on the server(s) indicated and you will need to wait typically at least a minute or two for data to begin to appear in the specified locations.

7.6.3 Power Fast-path options

You can collect data by selecting a non-IBM i connection type and using one the Start Power Collection sub menu options like nmon. These options are available in order to more quickly capture data using default settings and will skip the Power Collection Wizard interface completely.



Power Fast-path options

When using these options data will be stored in the default locations which can be configured in the **Preferences -> Power -> Remote data directory** setting.

7.6.4 24x7 VIOS data collection monitors

If the VIOS and companion IBM i system are setup properly it is possible to enable 24x7 data collection of the data types nmon, npiv and sea. This support only works for VIOS systems and currently requires several manual install steps to enable.

Data is automatically collected on the VIOS, sent to the IBM in the IFS and then processed and analyzed using SQL and moved in DB files that iDoctor can graph. Old data is automatically removed based on the desired retention settings.

For more information on enabling this please see the following file on our website:

http://public.dhe.ibm.com/services/us/igsc/idoctor/VIOSMonitorSetup.pdf

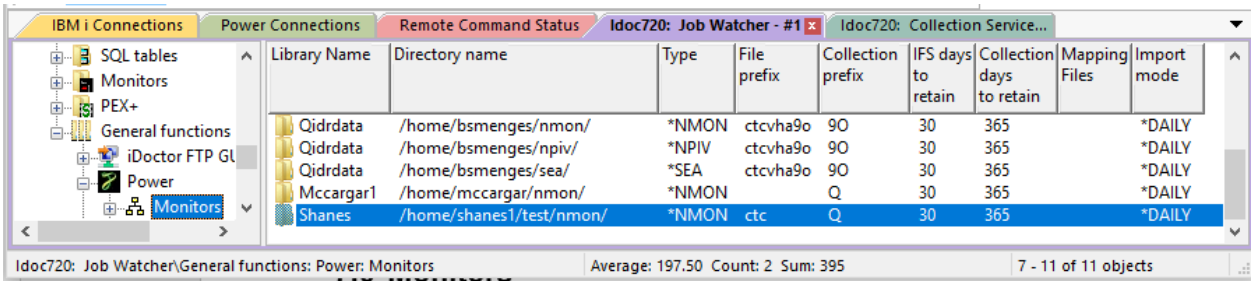
7.7 Monitors

Note: This option only applies when connected to an IBM i system and will not appear if the analysis DB is set to SQLite.

The Monitors folder allows the configuration of 24x7 data collection and retention settings for the IBM i system currently being used. This support works best after manual setup on the VIOS and IBM i (see previous section) is done to allow the automatic transfer of data from the VIOS to the IBM i's IFS.

Note: Data placed into IFS directories manually would also allow these functions to work.

This folder allows a user to control the settings in terms of which IFS files/directories to check for and what iDoctor DB tables (nmon/npiv and sea types) will be created as a result when new data comes into the system on the IFS.



Job Watcher -> General functions -> Power -> Monitors folder

Each entry in this folder displays the library and type of data that can be found within each. It also lists the directory name on the IFS as well as additional retention settings that apply for both the IFS files and the DB collection tables that iDoctor creates.

The columns in the Monitors folder are described below:

Column	Description
Library name	The library where the data is located for this monitor entry.
Directory name	The IFS directory where the incoming data from the VIOS is located.
Type	Indicates *ALL, *NMON, *NPIV or *SEA based on the type(s) of data this entry is for. This setting also means that when looking for IFS files in the directory, the file extensions must be either .nmon, .npiv or .sea depending on the value used.
(IFS) File prefix	Specifies the prefix of files that will be included in the monitor. This allows for files from different sources to be distinguished when they reside in the same directory. For example: One entry could have a file prefix value of **vios1** and the 2nd entry could use **vios2**. The same IFS data could end up different libraries on the IBM i if desired after being analyzed.
Collection prefix	Specifies the prefix that should be used for the collections when importing them. After this prefix a numbering scheme is used (0000-9999.)
IFS days to retain	Specifies the number of days matching files in the IFS directory should be retained before the monitor deletes them. Note that this determination is based on date within the file name, and not the change/create timestamps of the filenames.
Collection days to retain	Specifies the number of days to retain the database files generated by the monitor. This determination is made based on the timestamp of the most recent interval in the collection.
Mapping files	An optional listing of up to 5 VIOS to IBM i disk mapping files.
Import mode	Determines the behavior of the monitor when importing files. *NORMAL Every file imported from the IFS has an associated performance collection in the database. *DAILY Matching files of the same type from the same day are merged into a single performance collection.

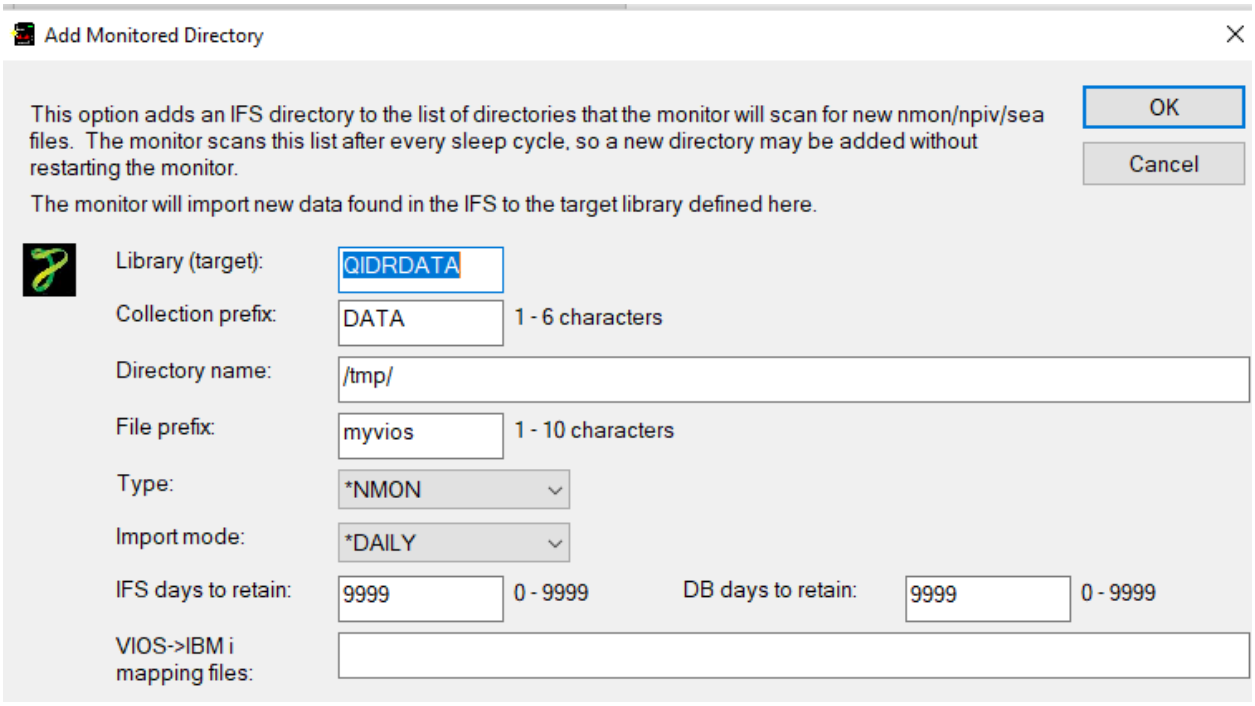
7.7.1 Menus

The Monitors folder provides the following menu options:

Option	Description
Select fields…	This lets you control which fields to show in the Monitors folder (see the previous section for the list of columns.)
Add Monitored directory	This will create a new monitor entry based on the settings you specify on the Add Monitored Directory window.
Start Monitor	This will kick off a new monitor job on the IBM i. Only 1 can be active at a time. This will check each monitor entry for new data and import needed files in the libraries indicated.
End Monitor	Ends the monitor job either delayed or immediate. *DELAY A signal is sent to the monitor job, which will end cleanly as soon as the signal is recognized. This should happen within 15 seconds in most cases. *IMMED An ENDJOB *IMMED command is issued to immediately end the directory monitor job. Not recommended for normal use.
Tracing -> Enable	Turns on debug tracing for the monitor job. Typically, you only need to use this if iDoctor support asks for it.
Tracing -> Disable	Turns off debug tracing.
Tracing -> View log	Downloads the current monitor log to the PC and displays it in Notepad,

7.7.2 Add Monitored Directory

This option adds an IFS directory to the list of directories that the iDoctor monitor will scan for new NMON/NPIV/SEA files. The monitor scans this list after every sleep cycle, so a new directory may be added without restarting the monitor. As new files are uploaded to the IFS directory, the monitor will import them according to the rules defined on this window.

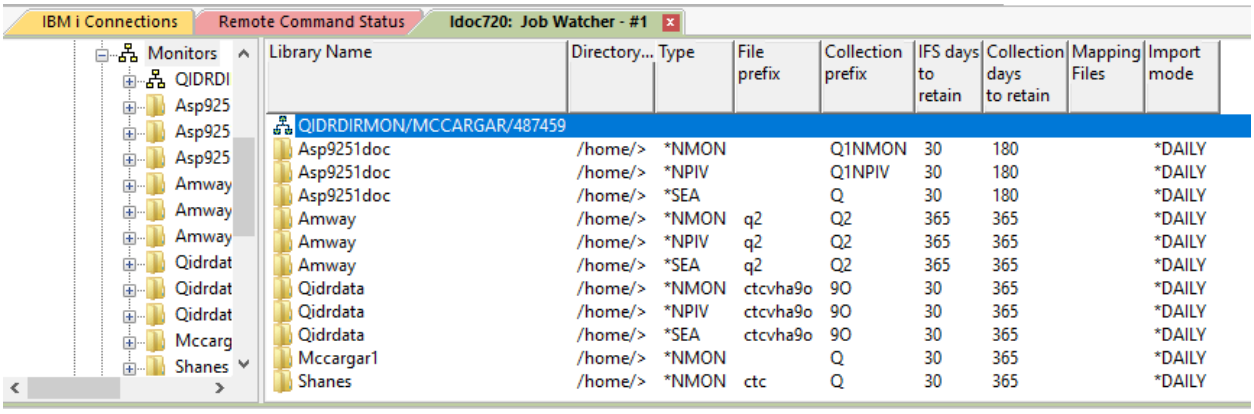


Add Monitored Directory window

Option	Description
Library name	The library where the data is located for this monitor entry.
Collection prefix	Specifies the prefix that should be used for the collections when importing them. After this prefix a numbering scheme is used (0000-9999.)
Directory name	The IFS directory where the incoming data from the VIOS is located (on the IBM i.)
(IFS) File prefix	Specifies the prefix of files that will be included in the monitor. This allows for files from different sources to be distinguished when they reside in the same directory. For example: One entry could have a file prefix value of vios1 and the 2nd entry could use vios2. The same IFS data could end up different libraries on the IBM i if desired after being analyzed.
Type	Indicates *ALL, *NMON, *NPIV or *SEA based on the type(s) of data this entry is for. This setting also means that when looking for IFS files in the directory, the file extensions must be either .nmon, .npiv or .sea depending on the value used.
Import mode	Determines the behavior of the monitor when importing files. *NORMAL Every file imported from the IFS has an associated performance collection in the database. *DAILY Matching files of the same type from the same day are merged into a single performance collection.
IFS days to retain	Specifies the number of days matching files in the IFS directory should be retained before the monitor deletes them. Note that this determination is based on date within the file name, and not the change/create timestamps of the filenames.
DB (collection) days to retain	Specifies the number of days to retain the database files generated by the monitor. This determination is made based on the timestamp of the most recent interval in the collection.
Mapping files	A list of up to 5 VIOS to IBM i disk mapping files, where every 20 characters specify first a library, and then a file name. These must be appropriately spaced.

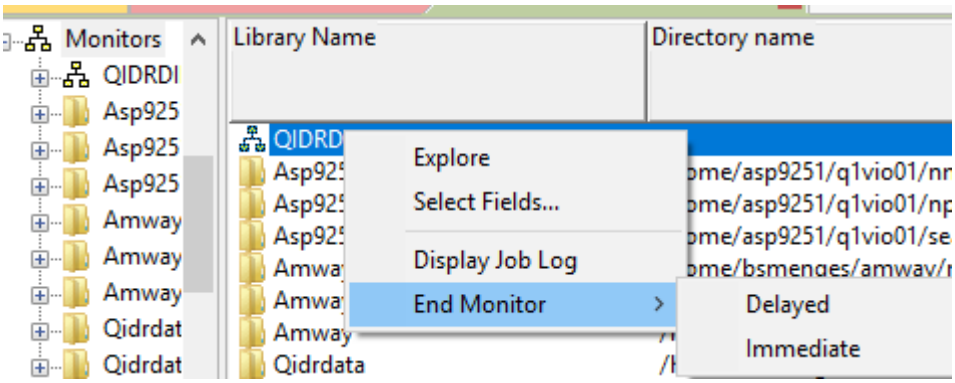
7.7.3 Active or Not

In order to tell if the monitor of this type is running on an IBM i just look and see if a job name is listed at the top of the entries within the Monitors folder. If not, then the Monitor is not currently running on the IBM i and no data will be processed. Only 1 monitor can be active at a time.



Monitors folder showing the active monitor job

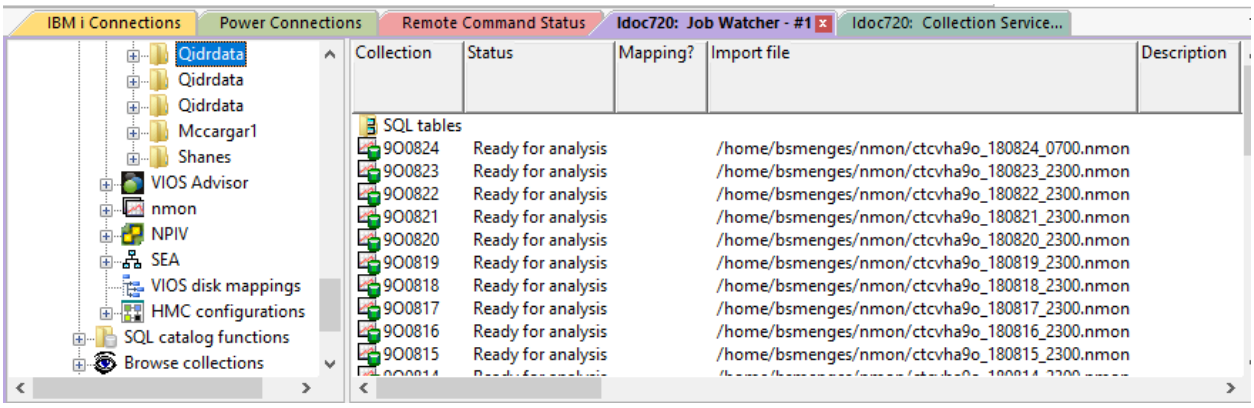
If you expand the job you can view the call stack. If you right-click the job, there are options to end the monitor or display the job log.



Monitor job options

7.7.4 Collections (within a monitor)

Each monitor entry contains a list of one or more collections. These collections are documented in the appropriate sections for nmon, npiv and sea types of data.



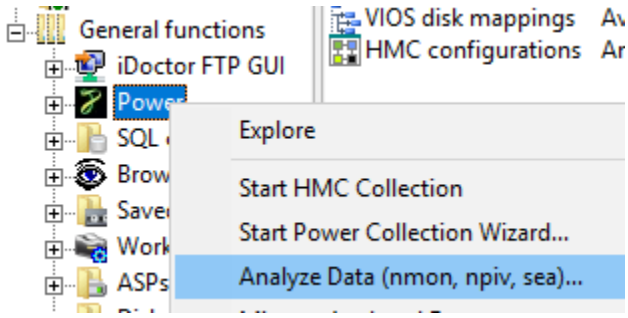
Job Watcher -> General functions -> Power -> Monitors -> Qidrdata (nmon) folder

7.8 Analyze Data Window

The Analyze Data window is used to import already collected nmon, npiv or sea stats data into the current analysis database. The analysis database could be an SQLite DB on the PC or a library on an IBM i.

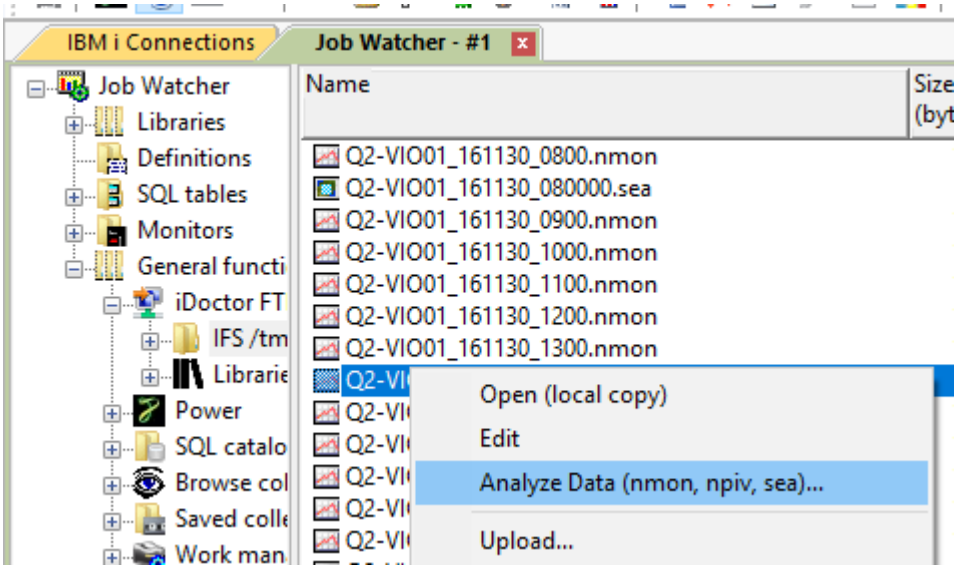
This window is accessible using one of these methods:

1) From an IBM i component, right-click the General functions -> Power folder.



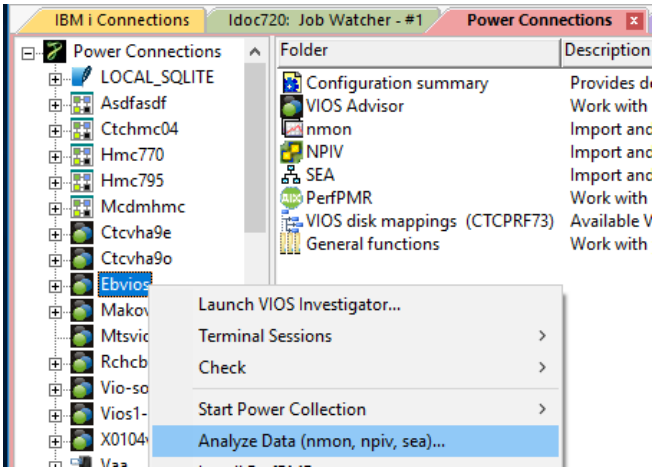
General functions -> Power (menu) -> Analyze Data option

2) From an IBM i component, when viewing the IFS, and right-clicking a file with the extension of .nmon, .npiv or .sea.



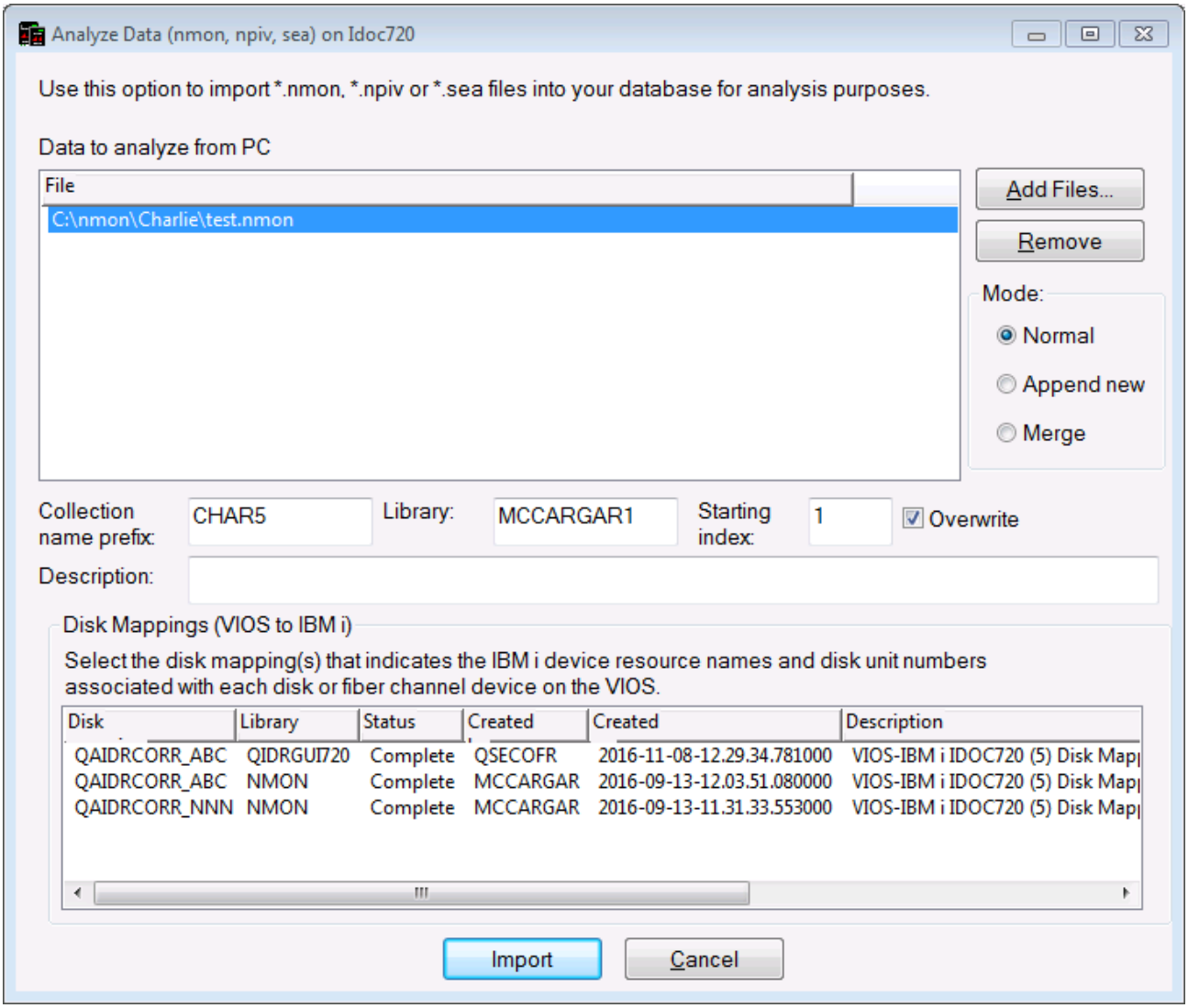
General functions -> IFS -> .nmon file (menu) -> Analyze Data option

3) From the Power Connections View, right click one of the connection folders.

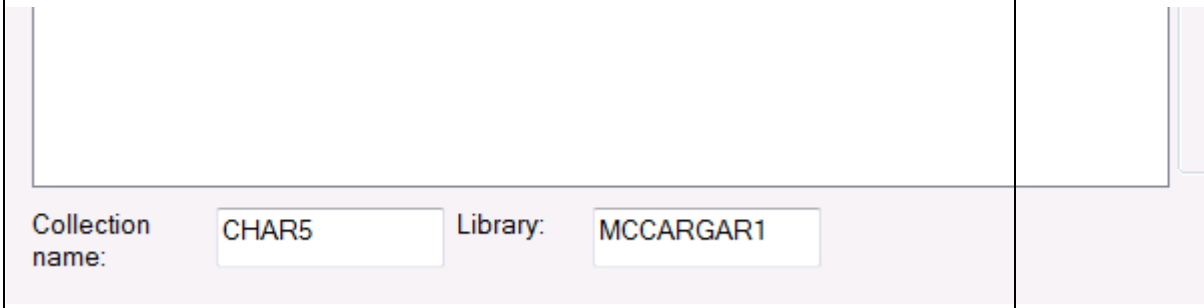


Power Connections -> VIOS (menu) -> Analyze Data option

An example of this window is:



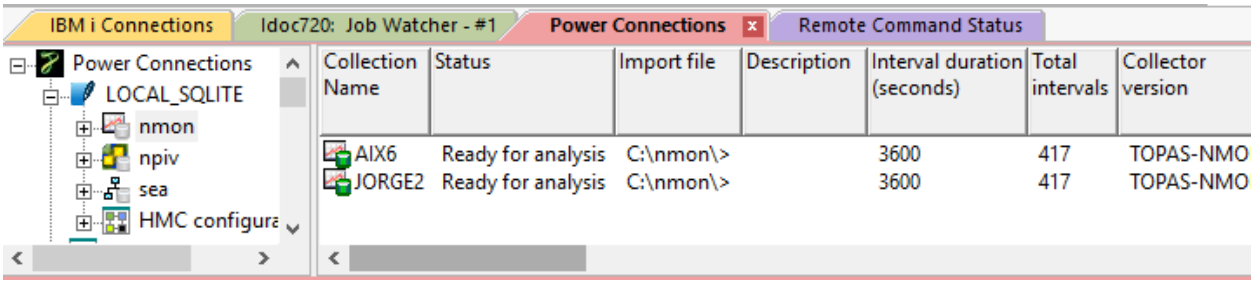
Analyze Data window when connected to an IBM i

Option	Description
Data to analyze from PC list	This is the list of files that will be analyzed and imported into the analysis DB (specified in the title bar of the window.) **Note:** When this window is initiated from the IFS on an IBM i, then the contents of this list is preloaded based on the selections made prior to selecting the Analyze Data menu.
Add Files button	This button will allow the user to select 1 or more nmon, sea or npiv files to add to the list. **Note:** This button will not appear when this window is initiated by selecting one or more files from the IFS
Remove button	This button will remove any selections made from the data to analyze from PC list.
Mode	The mode controls the type of import to perform. These options are available: **Normal:** In this mode each file specified will cause a new collection to be created. The collection name prefix is used to determine the beginning of each collection name, and a number is added to the end of the name (from 001 to 999 and based on the starting index and overwrite values.) The number used is based on the names that are available in the database is overwrite is unchecked. **Append new:** This mode should only be used if you wish to reimport the same file you have imported previously, and it now contains additional data. With this option the collection name and library are used as is.  **Merge:** This mode is used to combine data from multiple files into a single collection. With this option the collection name and library are used as is. This method is recommended to merge many files that represent different sets of data from the same system/day together, so it can be analyzed more easily.
Collection name prefix: (IBM i only)	This option only appears when Mode is “normal.” This is the 1st 7 characters of the collection name and the last 3 character will be a number from 001 to 999 based on the starting index value.
Collection name	This option only appears when the Mode is “append new” or “merge. It specifies the collection name to use when creating new performance data.
Library (IBM i only)	It specifies the IBM i library where the performance data will be stored.
Starting index	This index applies only to the normal mode and indicates the desired starting point when building the collection names (1 = 001, 50 = 050, etc.) If the 'Overwrite' box is checked the first collection will be based on this value given. Otherwise the first collection created will be the first name available without deleting existing data.
Overwrite	This option only appears when Mode is “normal.” This option indicates if any existing collections will be overwritten or not.
Description	This value is a description to give the collection(s) that are being created.
Disk mappings (IBM i only)	This is a list of VIOS to IBM i disk mappings that have been previously create with iDoctor that you may associated with the data being imported. By specifying a disk mapping appropriate for the data being used it will provide additional analysis options

	in some of the nmon and npiv graphs and reports.

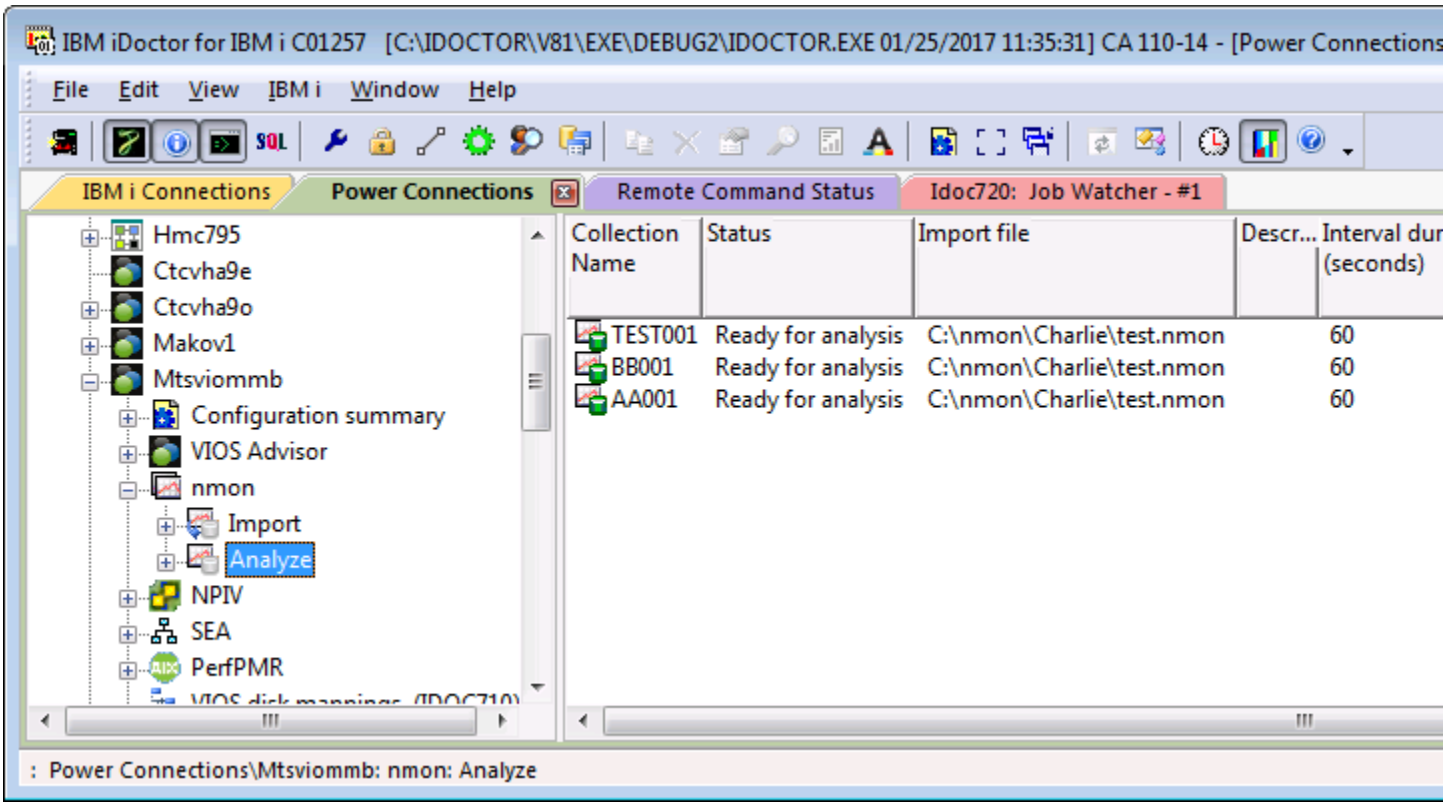
After using this option, the data will be available within the analysis DB (either an IBM i or SQLite DB on PC).

For example, the interface to view SQLite nmon collections looks like this:



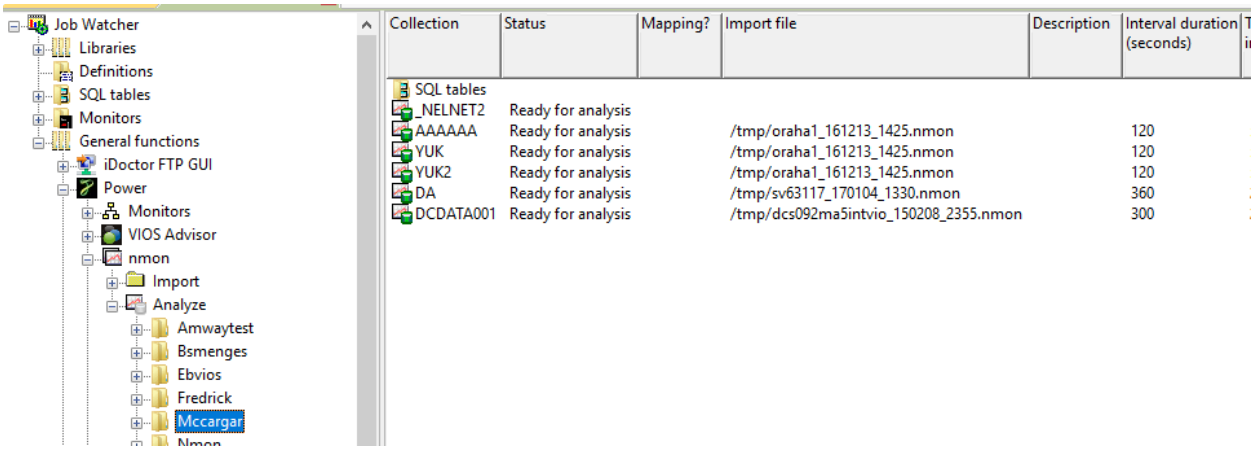
SQLite nmon collections list

When connected to a VIOS within Power Connections and using either SQLite or an IBM I as the analysis DB then use the VIOS name -> nmon -> Analyze folder:



SQLite nmon collection list under VIOS mtsviommb

When connected to an IBM i then use the General functions -> Power -> nmon -> <specifc library> folder to view the nmon data found on the current IBM I (which may or may not be the same as the analysis DB.)



IBM i nmon collection list

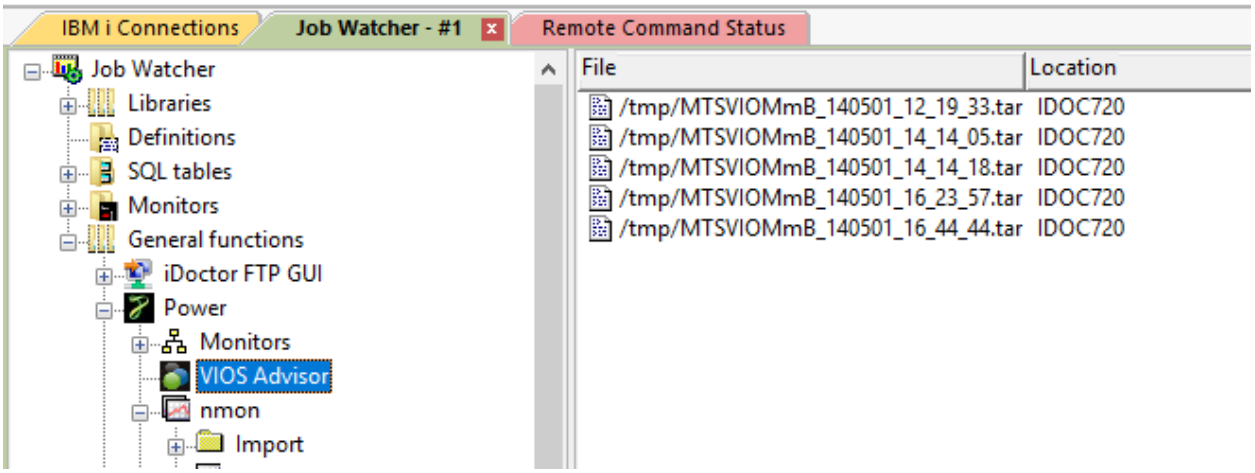
7.9 VIOS Advisor

The VIOS Advisor folder provides a listing of VIOS Advisor data that has been found on the remote server (typically either a VIOS or when connected to an IBM, in the IFS.) From this view a user can download the data to the PC to unzip and analyze it.

Visit this website for more details on VIOS Advisor:

http://www-01.ibm.com/support/knowledgecenter/POWER7/p7hb1/iphb1_vios_perf_adv.htm

An example of the contents of this folder is:



General functions -> Power -> VIOS Advisor folder in Job Watcher

The following menu options are available when right-clicking the VIOS Advisor folder:

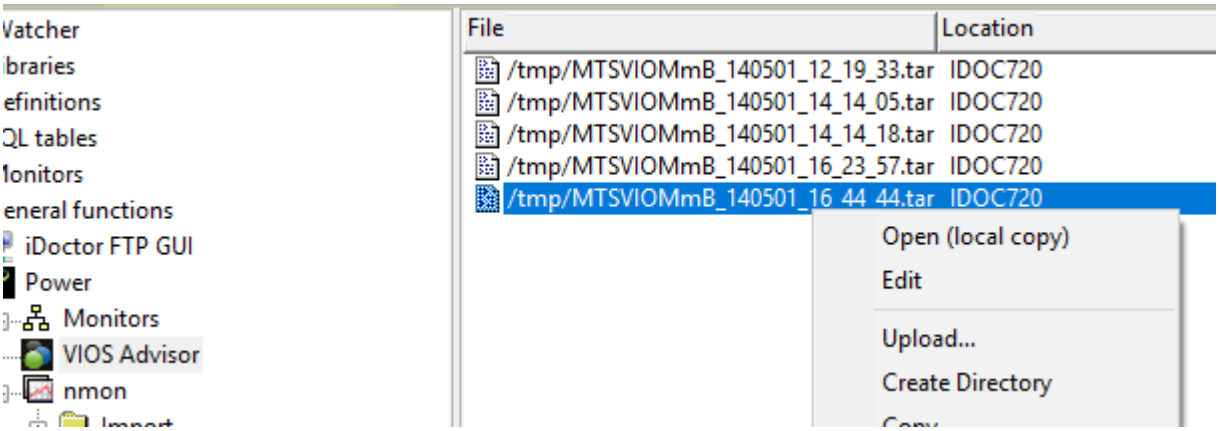
Menu	Description
Explore	Displays the contents of the folder in the list.
Upload…	This option displays the Upload files from PC window which is used to transfer any previously collected VIOS Advisor .tar files from the PC to the server in the desired directory.
Find VIOS Advisor data	This option will scan the specified directories on the server looking for VIOS Advisor data. Afterwards, refreshes of the VIOS Advisor folder will reveal the data found.
Start VIOS Advisor	This option provides several choices for how VIOS Advisor data shall be collected.

7.9.1 Analyzing

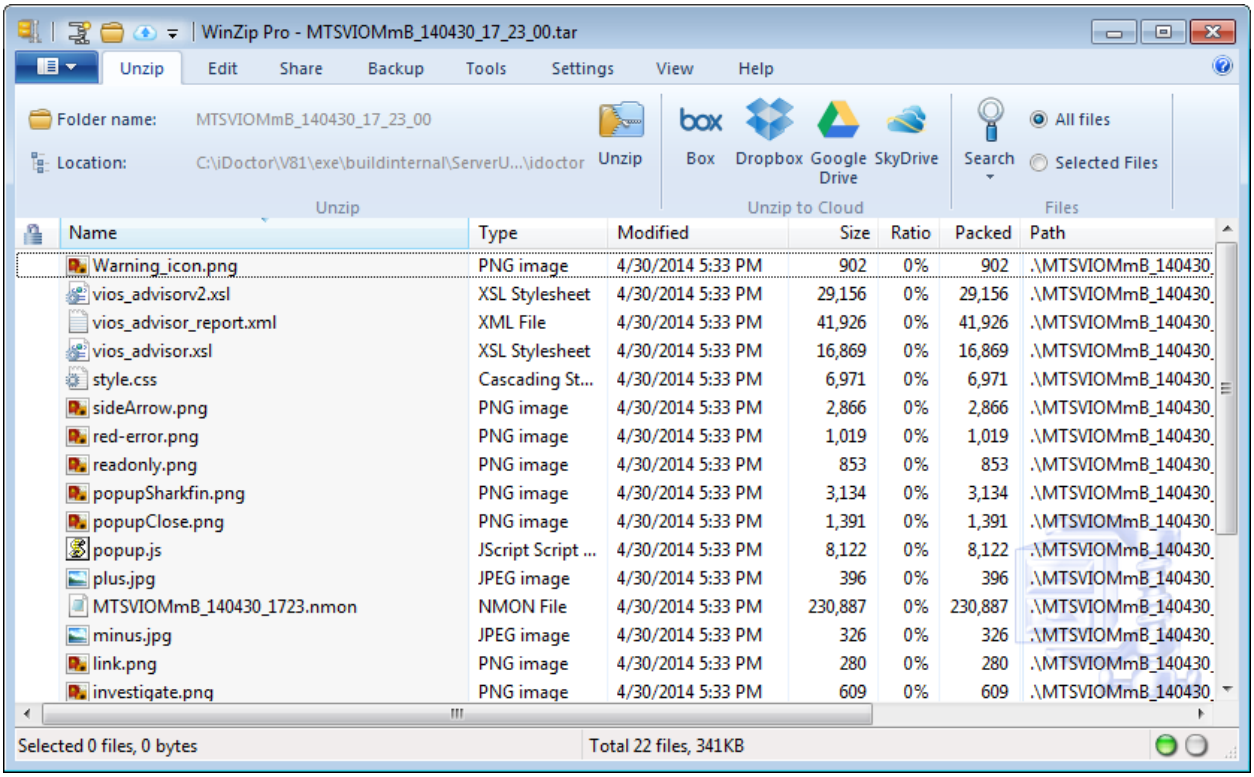
VIOS Advisor data is stored in a .tar file which will need to be transferred to a system that has a web browser and a .tar file extractor installed. After extracting the data, you will need to open the **vios_advisor_report.xml** file in a web browser in order to view the report.

To analyze this data on your PC with the iDoctor GUI you will need to have something like 7Zip or WinZip installed and configured to be the default program used when opening .tar files. Here is an example of what the process looks like to analyze this data.

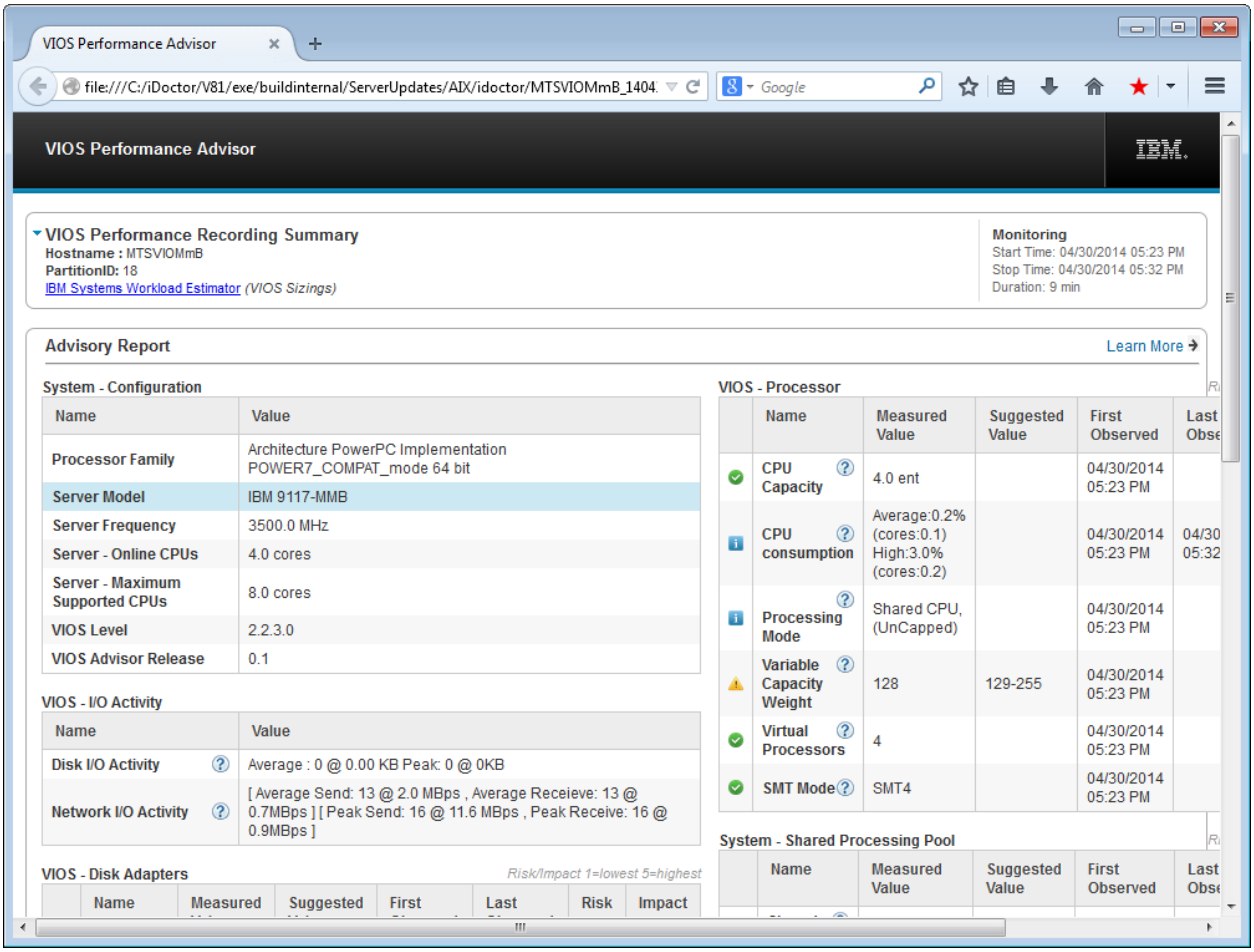
1) Right-click the desired .tar file and use the Open menu.



2) Unzip the data to the directory of your choice by pressing the Unzip button.



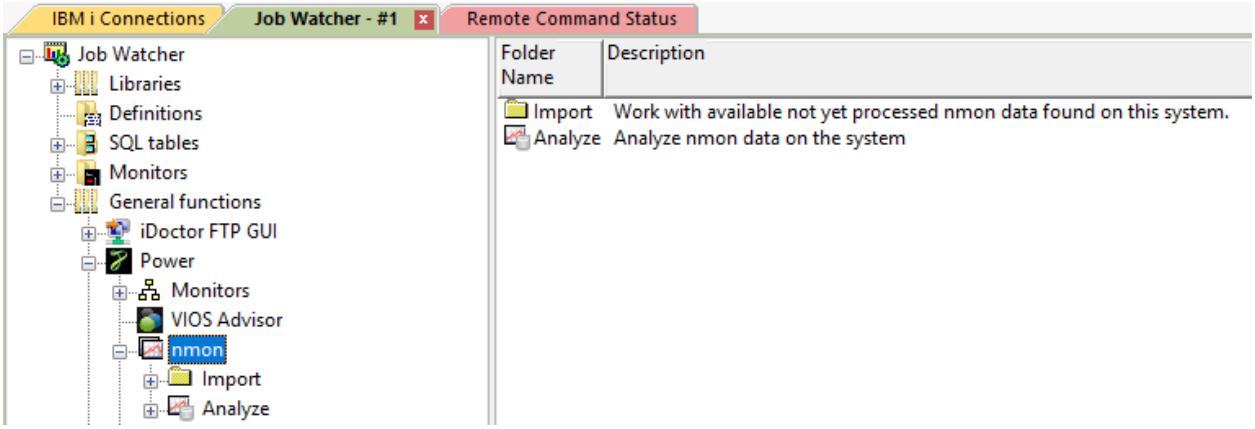
3) Open the vios_advisor_report.xml into either FireFox or Internet Explorer. <u>This report does not work correctly with Google Chrome.</u>



VIOS Performance Advisor Report

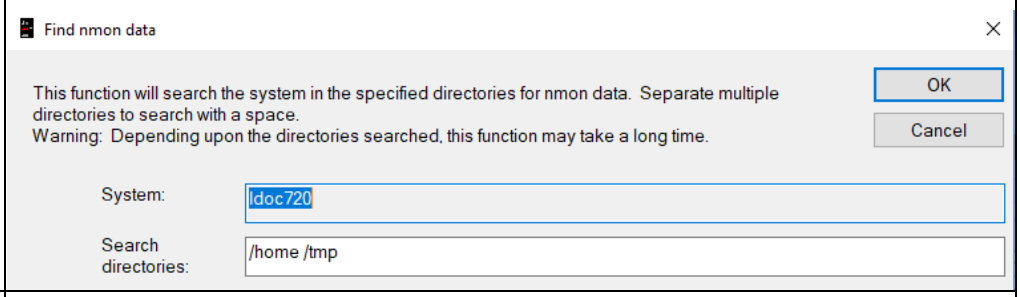
7.10 nmon

The next folder under Power is nmon which provides the ability to either import or analyze nmon data found on the current system.



General functions -> Power -> nmon folder

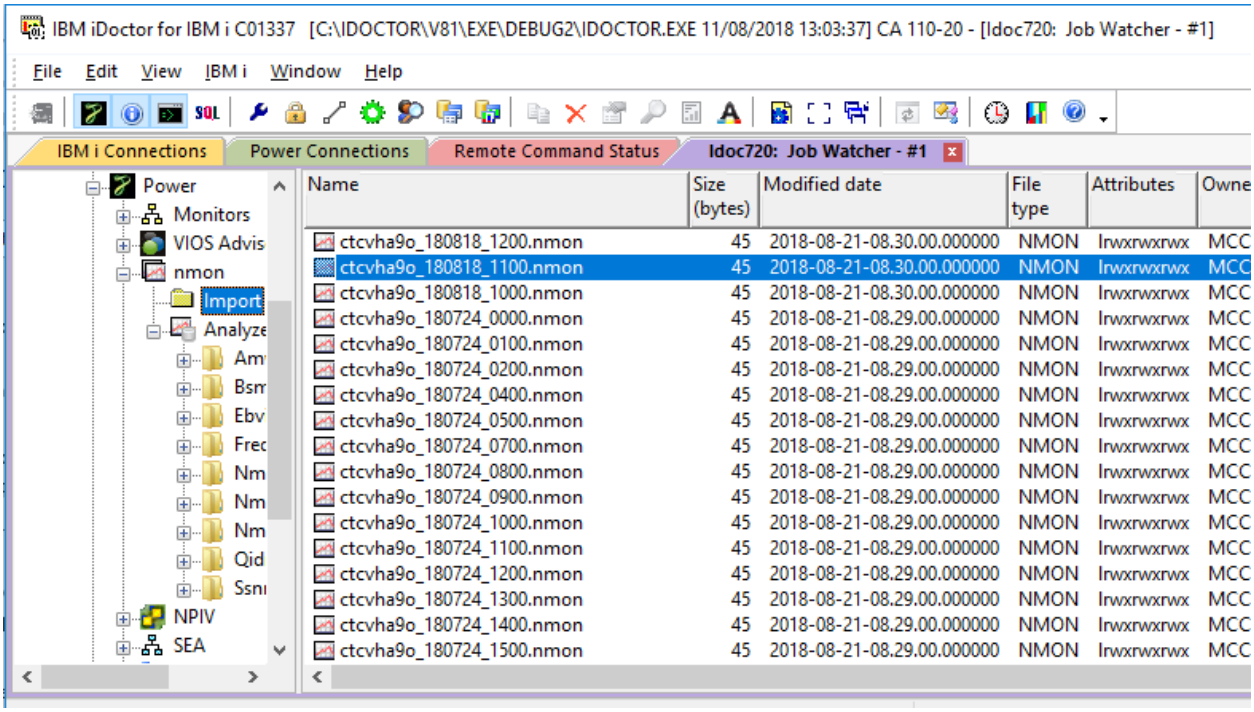
The following menu options are available when right-clicking the nmon folder:

Menu	Description
Explore	Displays the contents of the folder in the list.
Upload…	This option displays the Upload files from PC window which is used to transfer any previously collected nmon files from the PC to the IFS in the desired directory. **Note:** This option only applies when connected to an IBM i (using General functions -> Power and not via the Power Connections interface.)
Find nmon data	This option will scan the specified directories on the current system looking for nmon data. This operation could take seconds to several minutes. Afterwards a refresh of the nmon -> import folder will reveal the data found. 
Start nmon	This option provides several choices for collecting nmon data.
Analyze Data (nmon, npiv, sea)	This option allows you to upload nmon data from your PC to the current analysis system for graphing purposes. After this process completes the data will be available under the Analyze folder. **Note:** If not currently connected to an IBM i, then the data will be sent to the IBM i you have set as the analysis database.

7.10.1 Import

This folder displays the available raw nmon data found on the system that has not yet been processed into the analysis database.

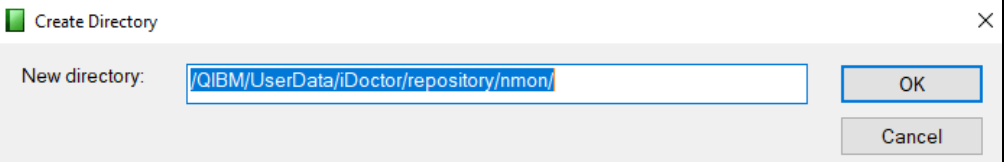
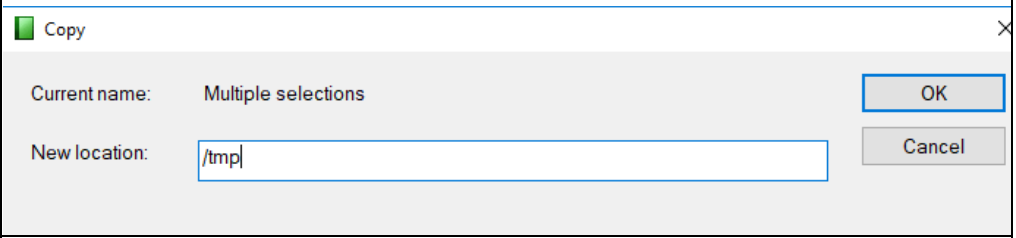
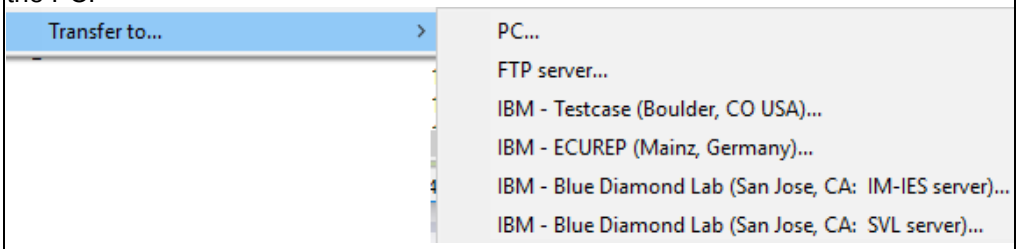
Note: In order to find new data that has been sent to the system, you must use the Find nmon data menu option.



General functions -> Power -> nmon -> Import folder

Note: The contents of the Import folder are cached and only rebuilt each time the Find NMON data option is used. If these files are removed (outside of this interface) the change will NOT be immediately reflected here and only rerunning the Find option will correct this.

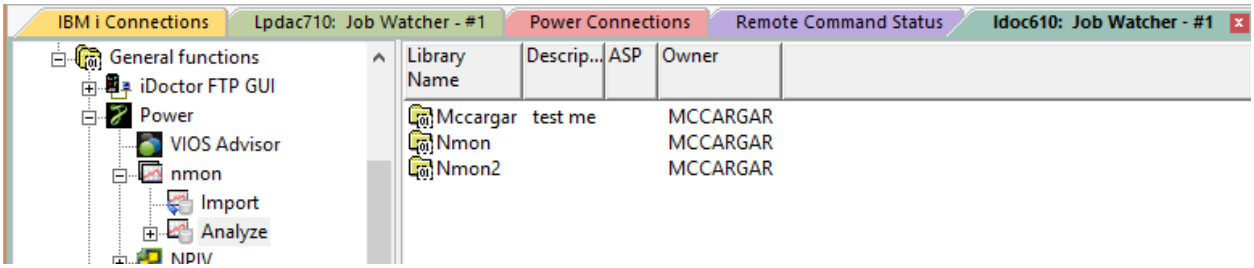
When right-clicking on an .nmon file in this folder, the following menu options are available:

Menu	Description
Open (local copy)	This option will cause the .nmon file to be downloaded to the PC and then opened using the default program specified by Windows for .nmon files.
Edit	This option will open an iDoctor editor which allows you to make changes to the file.
Analyze Data	This menu will cause the selected .nmon files to be processed by the iDoctor stored procedure that analyzes nmon data. Once complete the graphable nmon data can be found under the nmon -> Analyze folder.
Upload…	This option displays the Upload files from PC window which is used to transfer any previously collected nmon files from the PC to the remote system in the desired directory.
Create Directory	Creates a directory on the remote system you are working with (IFS, VIOS or AIX.) The default path listed is the current directory for this folder. 
Copy…	This option allows you to Copy all selected files to a new directory of your choice. 
Delete	Removes the selected files from the system.
Rename	Renames the selected file.
Save	This option will Save the desired files to either a SAVF (IBM i) or tar file (AIX, VIOS) depending on the type of system being used.
Transfer to	This menu provides several choices for sending the .nmon files to another server or the PC. 

7.10.2 Analyze

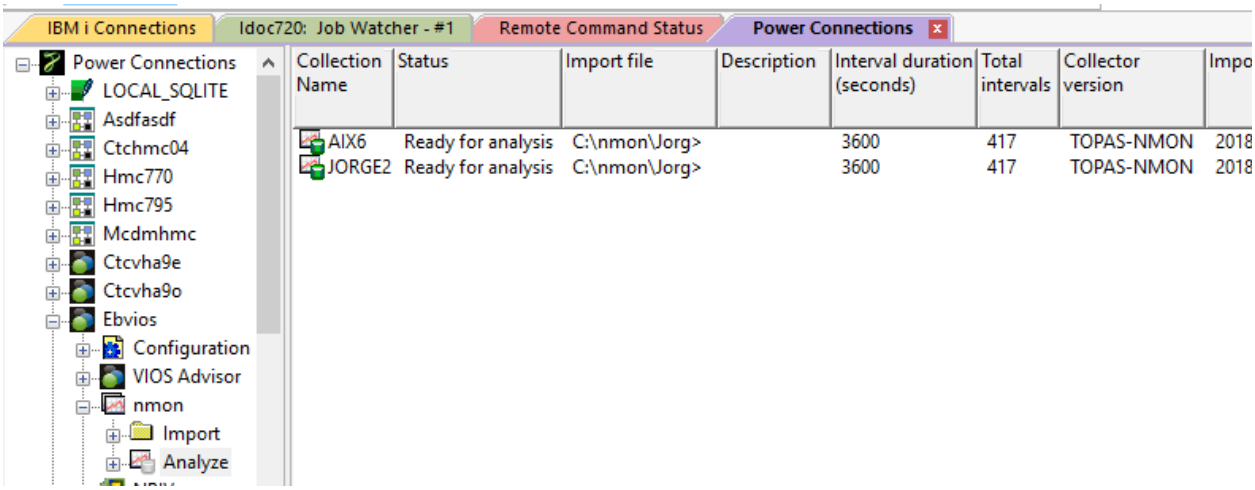
If using an IBM i as the analysis DB or connected currently to an IBM i then this folder provides a list of all libraries on the current system that contains nmon data.

An example is:



General functions -> Power -> nmon -> Analyze folder

Note: If using SQLite as the analysis DB then this folder contains the collections (example below.)

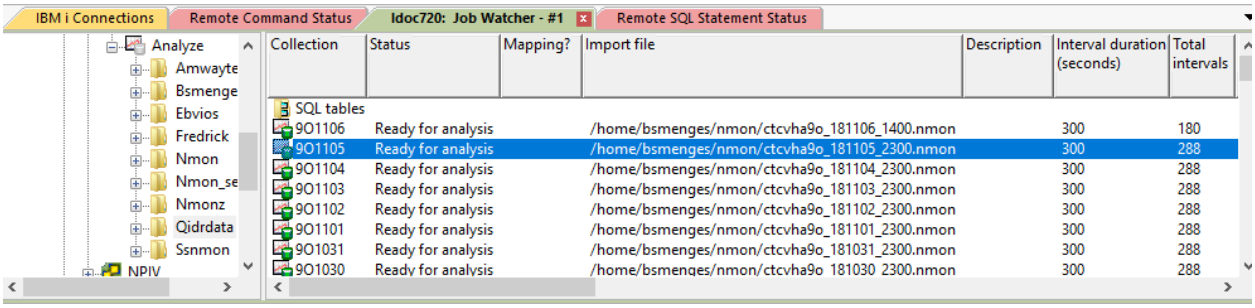


Power Connections -> VIOS -> nmon -> Analyze folder

7.10.3 Collections

Each library (or SQLite DB or monitor entry) contains a list of nmon collections that have been imported into it and are ready to graph. Each collection consists of a set of iDoctor created SQL Tables that were derived from the nmon data and in some cases a VIOS Investigator disk mapping (optional).

Each collection contains a detailed set of reporting options (graphs or tables) within it.



List of nmon collections in library Qidrdata

7.10.3.1 Collection Fields

The list of collections contains several columns which are described below:

Field	Description
Collection	Name of the collection. This is a short unique name (10 characters or less) given to the collection at import time based on the collection name prefix value.
Status	Indicates if all necessary VIOS Investigator tables have been created successfully.
Mapping?	Indicates if a disk mapping was used when creating this collection. Not having a disk mapping will mean fewer analysis options will be available.
Import File	This column displays the original file name that was used when the import occurred.
Description	A description given to the collection when the import occurred.
Interval duration	The duration of each interval within the nmon data.
Total intervals	The total number of intervals (snapshots) taken.
Collector version	Depending on the release of nmon, this is either a version number or TOPAS-NMON.
Import time	The date and time the import occurred.
Start time	The date and time when nmon started.
End time	The date and time when the nmon collection ended.
Host name	The name of the host on the system the nmon data was collected on.
Node name	The node name for the system the nmon data was collected on.
User name	The user name that started the nmon collection.
Build	The nmon build number used.
Command	The command used to create the nmon data.
AIX	The version of AIX installed when the collection was created.
System serial number	The system serial number of the system where the nmon data was created.
Mode	This indicates the import mode used on the Analyze data window when importing the data into the database. 0 = Normal 1 = Append 2 = Merge

7.10.3.2 Menus

The following menu options are available by right-clicking on a nmon collection.

Menu	Description
Analyses	This contains the list of analysis options available to run against the selected collections. In iDoctor, these analyses are SQL stored procedures. Currently these are the options available: **Analyze Collection** – This is the interface that lets you work with all the analyses. **Run Change nmon sensitive user data** – This will modify the data in the nmon collection to hide/replace potential any customer sensitive data. **Note:** This option only applies when connected to an IBM i or using an IBM i as the analysis DB.
nmon Analyzer	These graphs and tables are designed to look like the nmon Analyzer Excel spreadsheets as much as possible. The rest of the nmon graphs provided below are designed to look more like the IBM i Collection Services Investigator component so some of the terminology used on those graphs is different. **Note:** Not all graphs are provided, this is a work in progress.
System	These are high-level graphs and tables covering a variety of areas on the system (CPU, kernel, paging, memory, configuration, etc) The system configuration sub folder shows various reports regarding the hardware and logical configuration of the system.
CPU	The CPU graphs show CPU utilizations with or without CPU idle percentages included.
Disk	The disk graphs display the various disk statistics provided by nmon. Within the first folder are additional ranking graphs that allow the user to rank by disk name. If a valid disk mapping has been provided, then the following additional ranking graphs are available: By disk unit, by disk path, by ASP and by disk type **Note:** It is highly recommended to include disk response times in your nmon data. See the section on the Power Collection Wizard for more information on how to include these.
Reports	These are additional reports (table views) over the data.
TOP processes	These graphs show statistics for the Top processes collected by nmon. The top processes are optionally collected using a CPU filtering parameter within the Start NMON Collection Wizard.
Generate Reports	Launches the Report Generator function that lets you create multiple reports at once.
Open import file	This option allows the user to the transfer the original .nmon file to the PC and open it. **Note:** This will only work if the original import file still exists on the server. It will be opened in either the default Windows application for .nmon files or it can be opened in nmon Analyzer (which requires Excel to be installed.) **Note:** If you wish to use the option to open nmon Analyzer then make sure a copy of nmon analyzer exists in the iDoctor install directory (which is typically C:\program files(x86\IBM\iDoctor) called **nmon analyser.xlsm** **This file should exist with all iDoctor builds generated in 2016 onward, but you can download and replace this file as desired.**
Copy	This function will copy the desired collection to a different name in the same library or copy one or more selected collections to another library.
Delete	Removes the selected collections from the system.

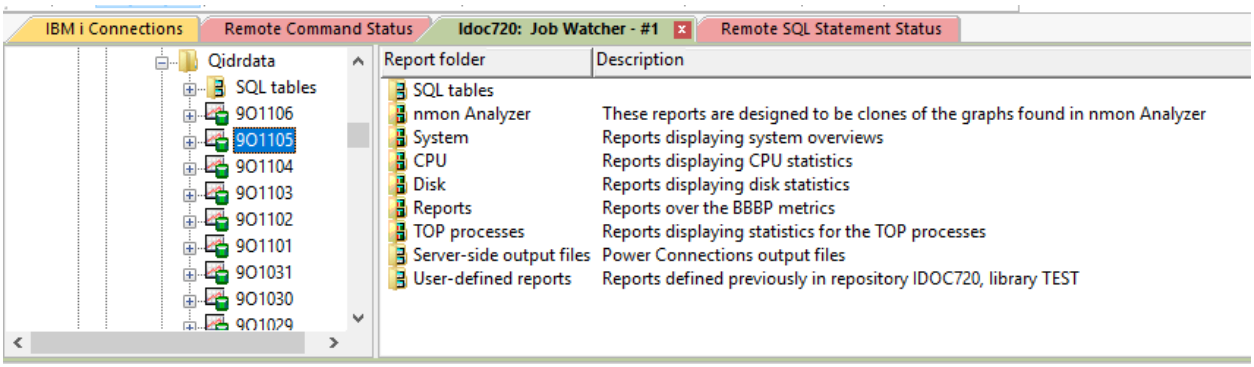
	Note: This action does NOT remove the original .nmon import file from the IFS.
Rename	Renames the specified nmon collection
Save…	Saves the specified nmon collection(s) to a save file on the IBM i.
Transfer to	This menu provides several choices for sending the nmon collections to another server or the PC.

Additional menu options that are common to all library folders in iDoctor are discussed here.

7.10.4 Reports

This section describes the reporting options available for nmon collections.

Many graphs are found within several folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the appropriate menu.



Job Watcher -> General Functions -> Power -> nmon -> Analyze -> Qidrdata -> (Collection) 9O1105

Each folder contains a series of graphs or reports. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Some graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one.

Tip: Some graph types (such as the disk graphs) support graphing multiple collections at once.
To do this, select all desired collections from the list of collections, then right-click and pick the desired graph from the menu.

7.10.4.1 Menus

Right-clicking a graph gives a menu with the following options:

Menu	Field Description
Open graph(s)	Opens the selected graphs into a new Data Viewer or an existing one depending on the submenu available that shows the list of Data Viewers (if any are open).
Edit	This option will open the graph without running the SQL statement. The SQL Editor will be opened allowing the user to modify the query before running the SQL.

7.10.4.2 nmon Analyzer

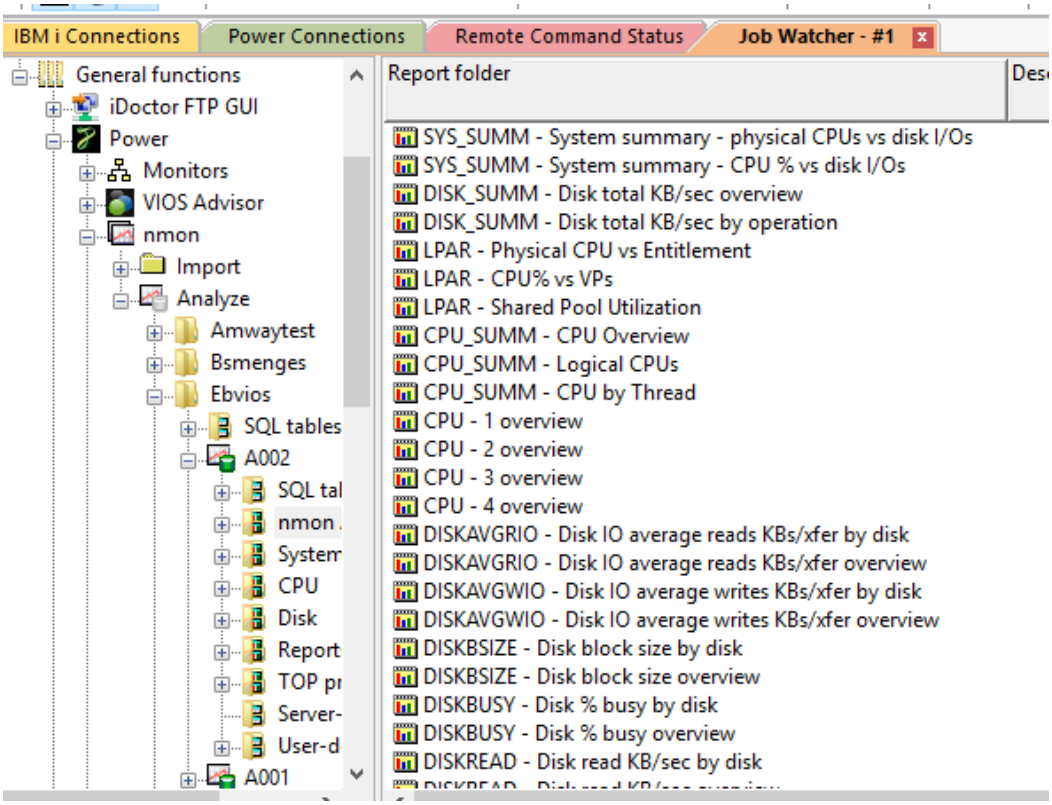
These graphs are designed to look like the graphs/sheets found in the nmon Analyzer Excel spreadsheet.

This section covers all nmon Analyzer sheets from A-L as well as the SYS_SUMM sheets. The other sheets are also covered but stored in additional folders.

Note: Some graphs will NOT appear depending on what data exists in the original nmon file. If the data tag does not exist, then the graphs will not appear here.

For more information on nmon Analyzer, visit this website:

https://www.ibm.com/developerworks/community/wikis/home/wiki/Power%20Systems/page/nmon_analyser

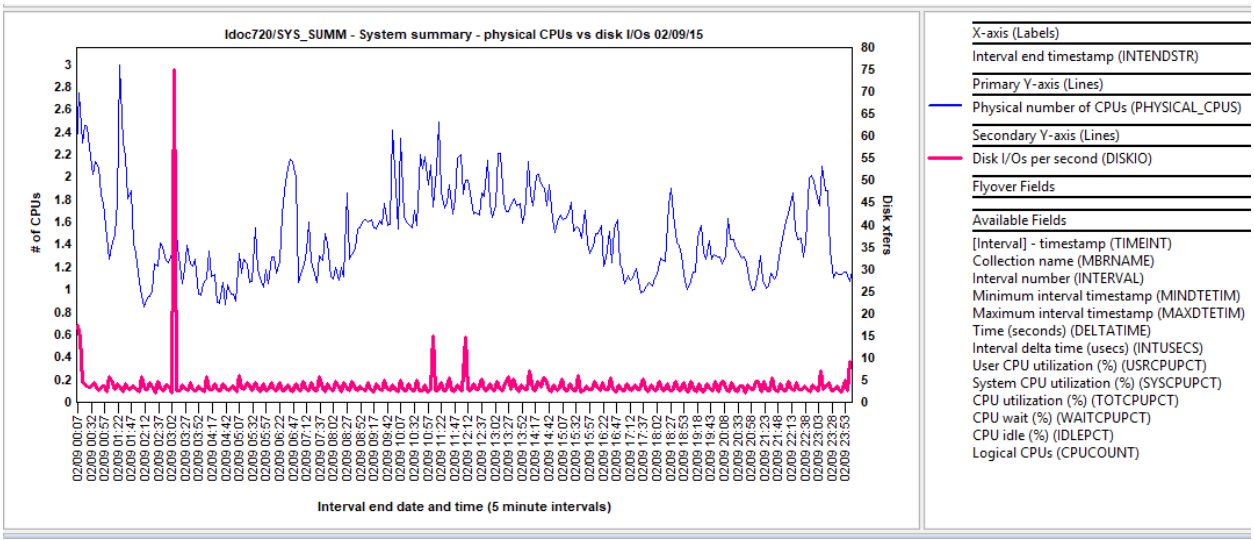


nmon Analyzer folder

7.10.4.2.1 SYS_SUMM – System summary – physical CPUs vs disk I/Os

This line graph provides a comparison of the # of physical CPUs used vs total disk I/Os over time.

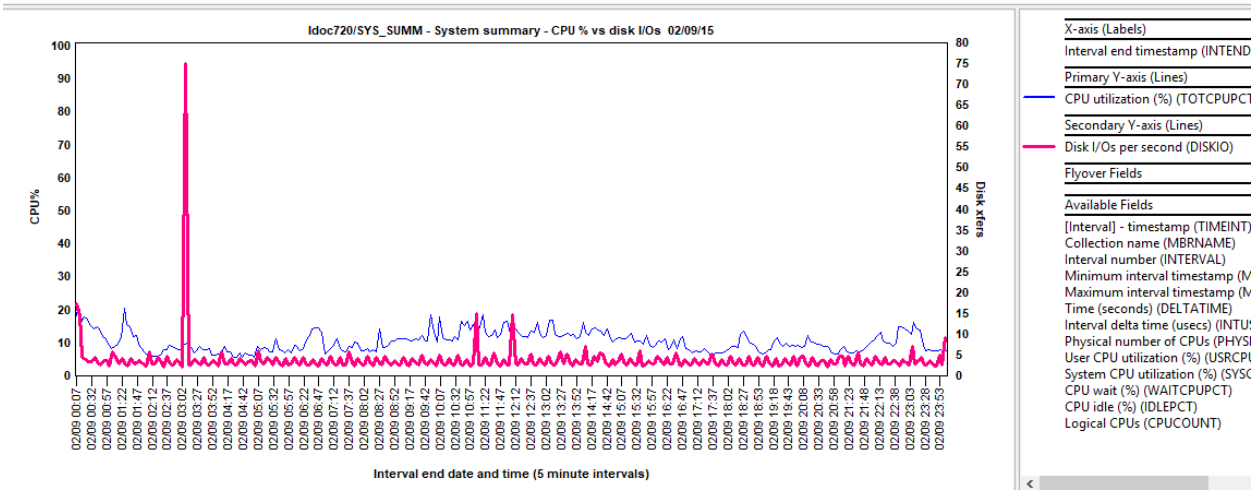
Note: This graph requires “LPAR” data records to be returned in the nmon output for the number of physical CPUs to be listed correctly at a value > 0.



SYS_SUMM – System summary – physical CPUs vs disk I/Os

7.10.4.2.2 SYS_SUMM - System summary - CPU % vs disk I/Os

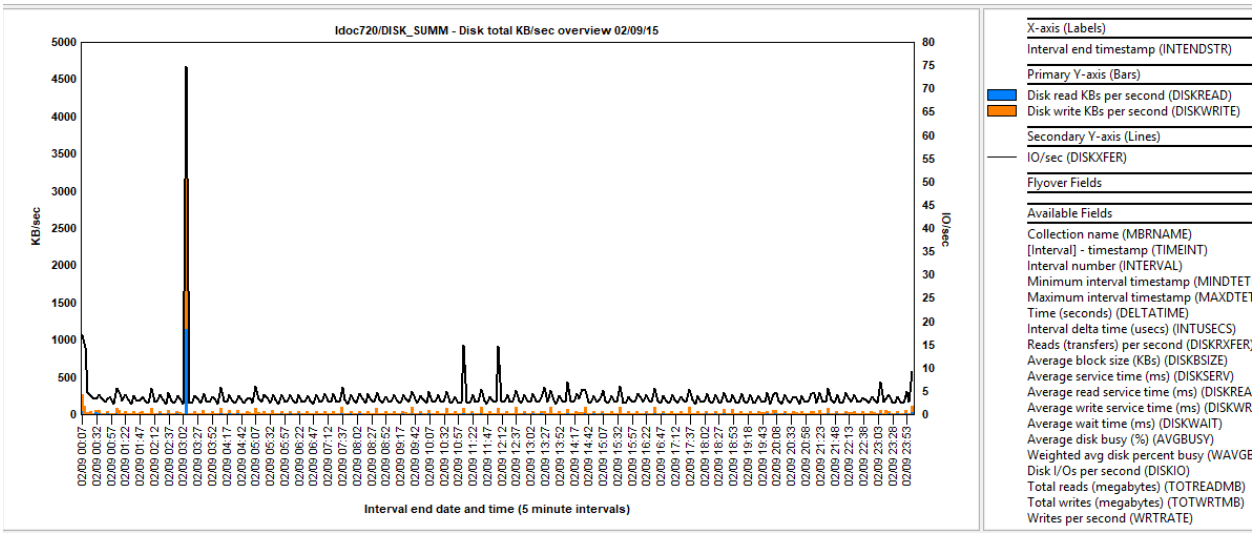
This line graph compares CPU utilization with total disk I/Os over time.



SYS_SUMM - System summary - CPU % vs disk I/Os

7.10.4.2.3 DISK_SUMM - Disk total KB/sec overview

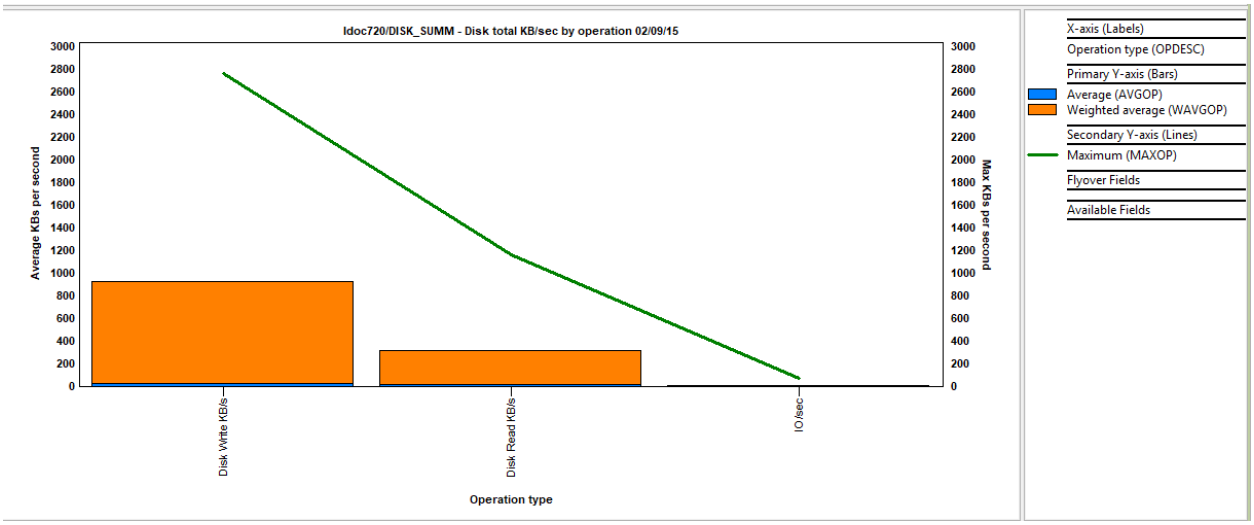
This graph compares disk read and write size rates over time with I/Os per second on the Y2.



DISK_SUMM - Disk total KB/sec overview

7.10.4.2.4 DISK_SUMM - Disk total KB/sec by operation

This graph displays the avg, weighted average and maximum values for each type of metric in the previous graph.



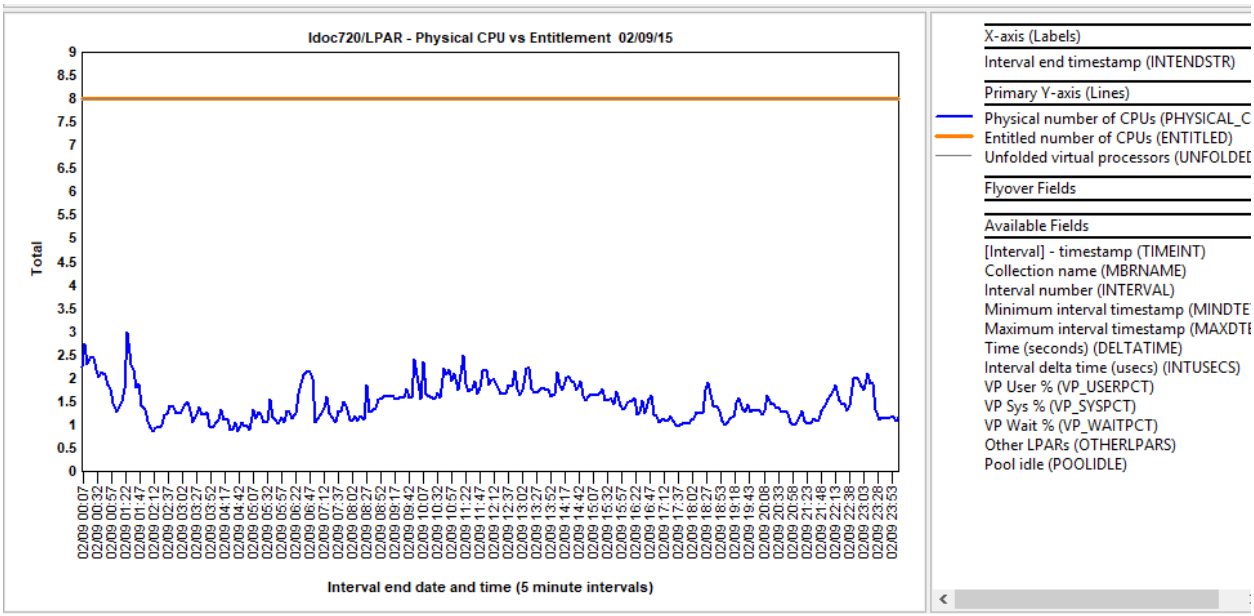
DISK_SUMM - Disk total KB/sec by operation

7.10.4.2.5 LPAR – Physical CPU vs Entitlement

This graph compares physical CPUs vs entitled CPUs along with the unfolded virtual processors.

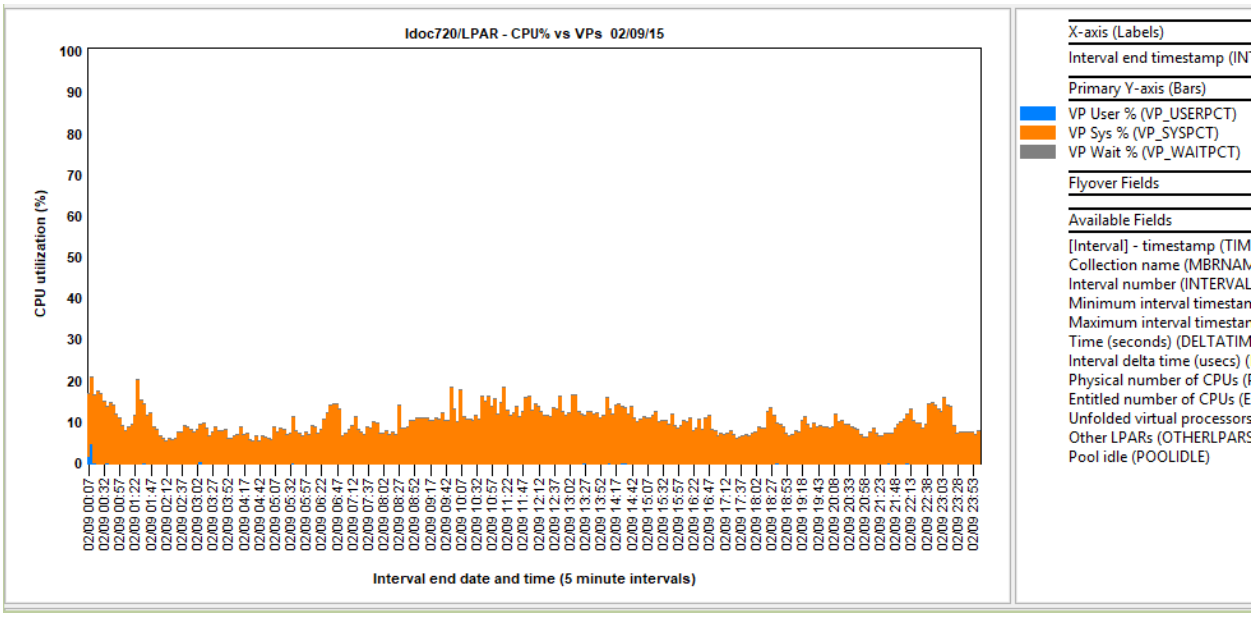
For an uncapped partition the number of physical processors may exceed the entitlement but can never exceed the number of virtual processors allocated to the partition. For AIX the graph also shows the number of unfolded virtual processors (AIX will “fold” - stop dispatching work to - excess processors in order to minimize scheduling costs).

Note that the ratio of physical processor to entitlement (shown as %entc in the output of the `lparstat` command) will generally be higher than CPU% on the CPU_ALL sheet. The reason for this is that a partition that is within its entitlement may wait for a short period of time before ceding a processor that enters an I/O wait or becomes idle. This can eliminate unnecessary context switches.



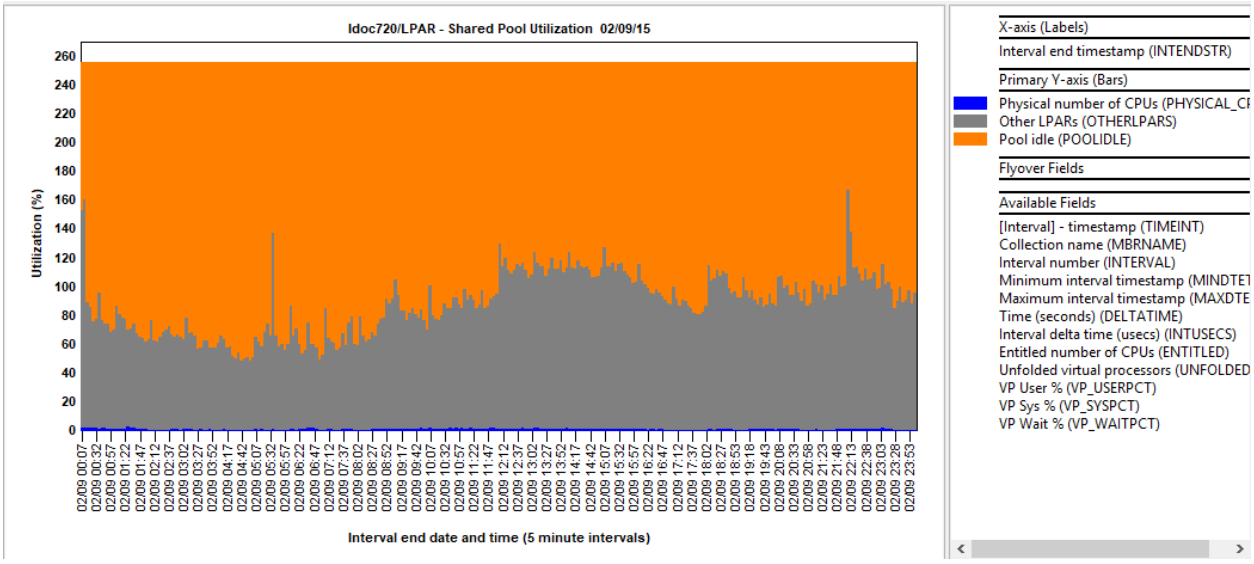
7.10.4.2.6 LPAR – CPU% vs VPs

This graph shows CPU utilization as a percentage of virtual processors – for AIX this is broken down in to VP User%, VP Sys% and VP Wait%. This level of detail is not available for Linux or releases of NMON prior to version 12.



7.10.4.2.7 LPAR – Shared Pool Utilization

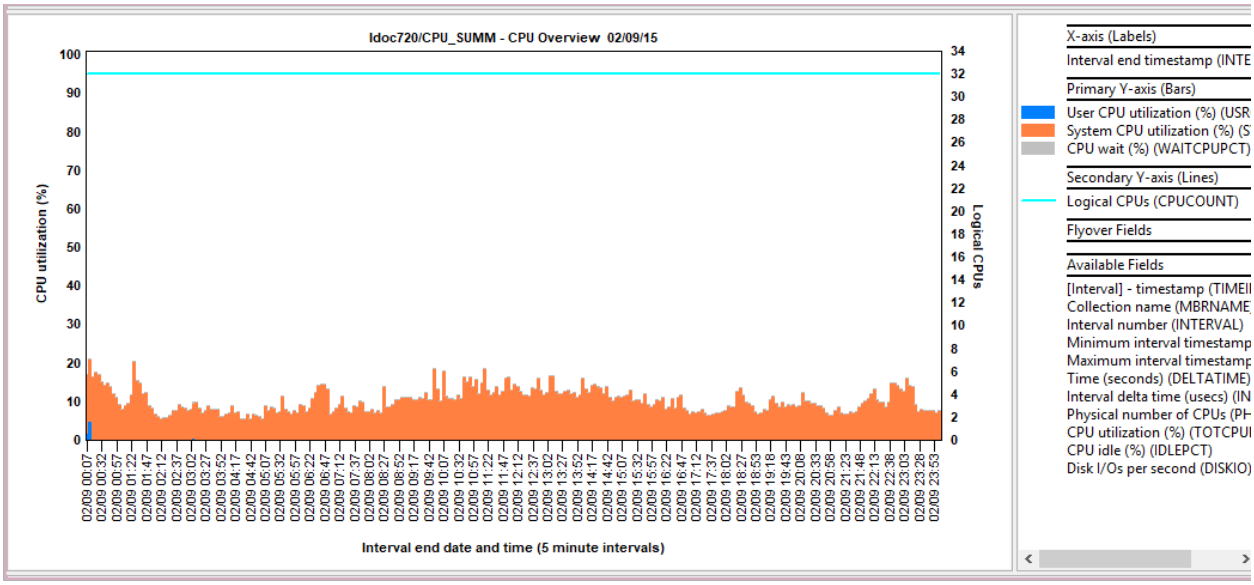
This graph is only present for AIX systems and shows CPU utilization of the shared pool by this and other partitions. Note that if the partition is not authorized to see utilization of the shared pool then the pool will appear to be 100% utilized.



7.10.4.2.8 CPU_SUMM - CPU Overview

This graph provides a breakdown of CPU Utilization by thread (logical processor) and by core over the collection period. The chart can be very useful in identifying situations in which the system is thread-starved (i.e. too few threads to fully utilize the logical processors) or where the workload is dominated by a small number of single-threaded processes.

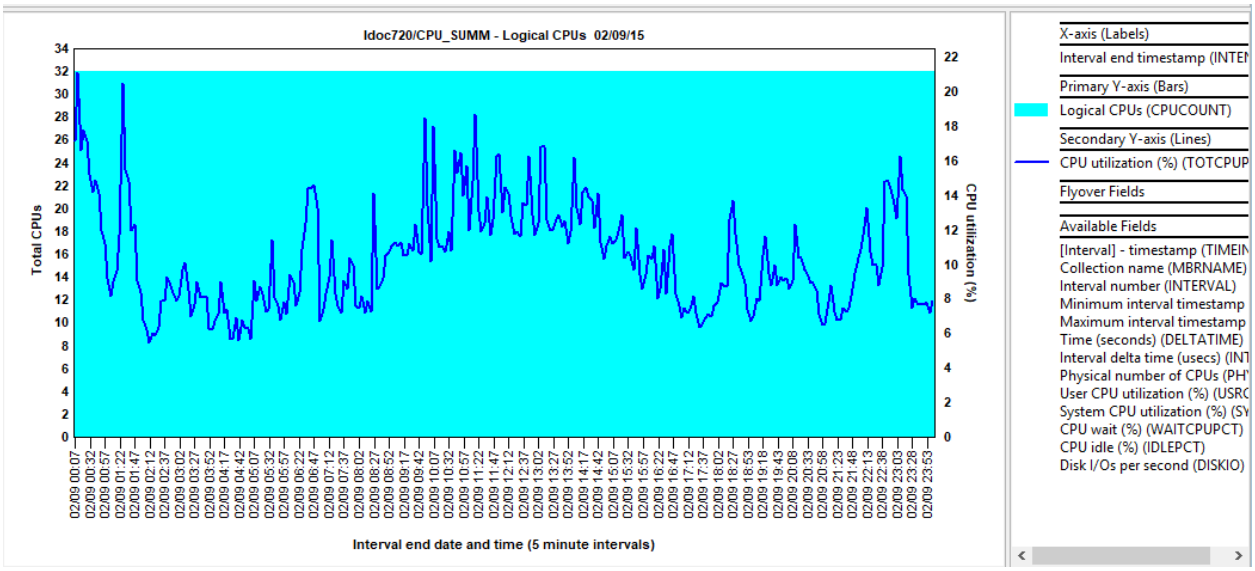
Note that if CPUs have been dynamically reconfigured during the collection period, these figures reflect only those intervals when the CPU was varied on.



CPU_SUMM - CPU Overview

7.10.4.2.9 CPU_SUMM - Logical CPUs

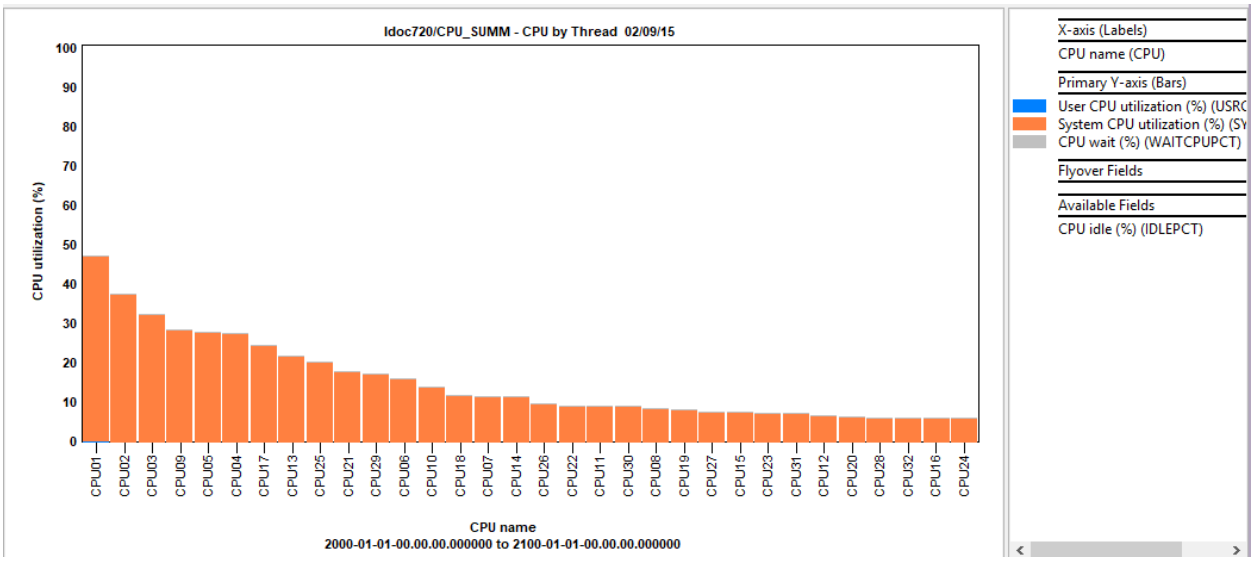
This graphs logical CPUs vs CPU utilization on the Y2.



CPU_SUMM - Logical CPUs

7.10.4.2.10 CPU_SUMM - CPU by Thread

This graph ranks the CPU threads / name.

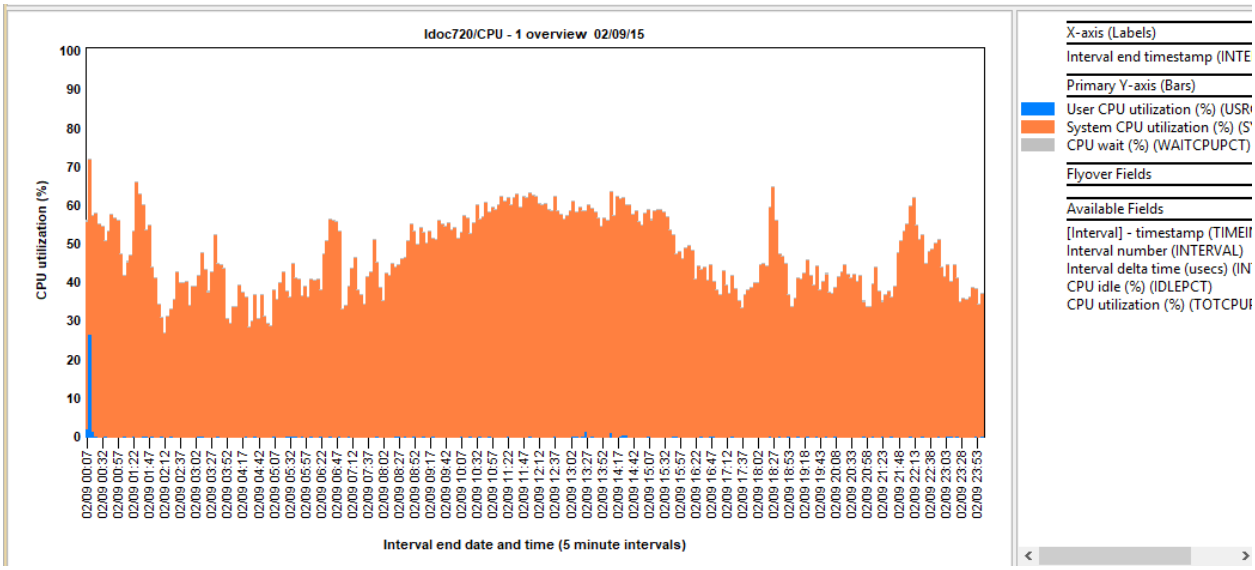


CPU_SUMM - CPU by Thread

7.10.4.2.11 CPU – N Overview

These sheets show User CPU utilization, System %, CPU wait% and CPU idle % by time of day for each logical processor. Note that for micro partitions, the CPU Idle % and CPU Wait % figures will include times when the physical processor was ceded to the shared pool.

Typically only graphs for CPU numbers 1-4 are provided but the SQL statement can be easily modified to change this graph.



7.10.4.2.12 DISKAVGRIO - Disk IO Average reads KBs/xfer by disk

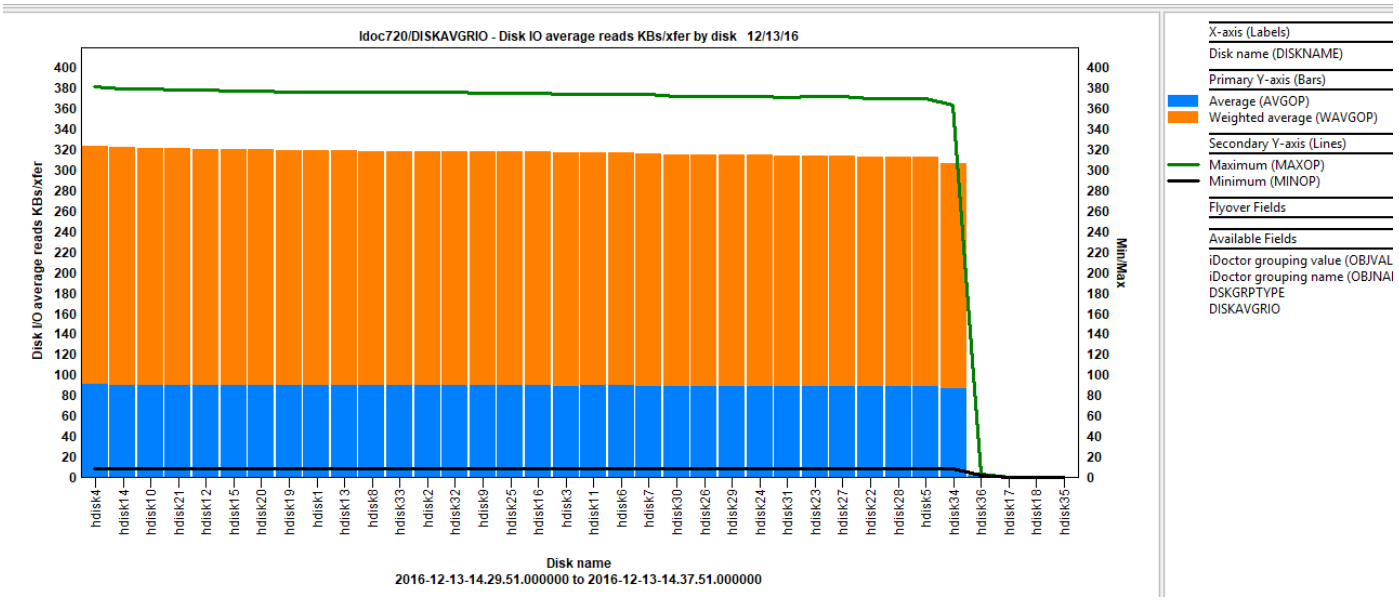
NOTE: AIX documentation here states the following which means the current nmon produced description "Disk IO Average Reads per second" for this field is **wrong**:

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.prftools/io_stat.htm

DISKAVGRIO, Disk IO Average Reads KBs/xfer

Average number of KBs that are read from the disk per read I/O operation.

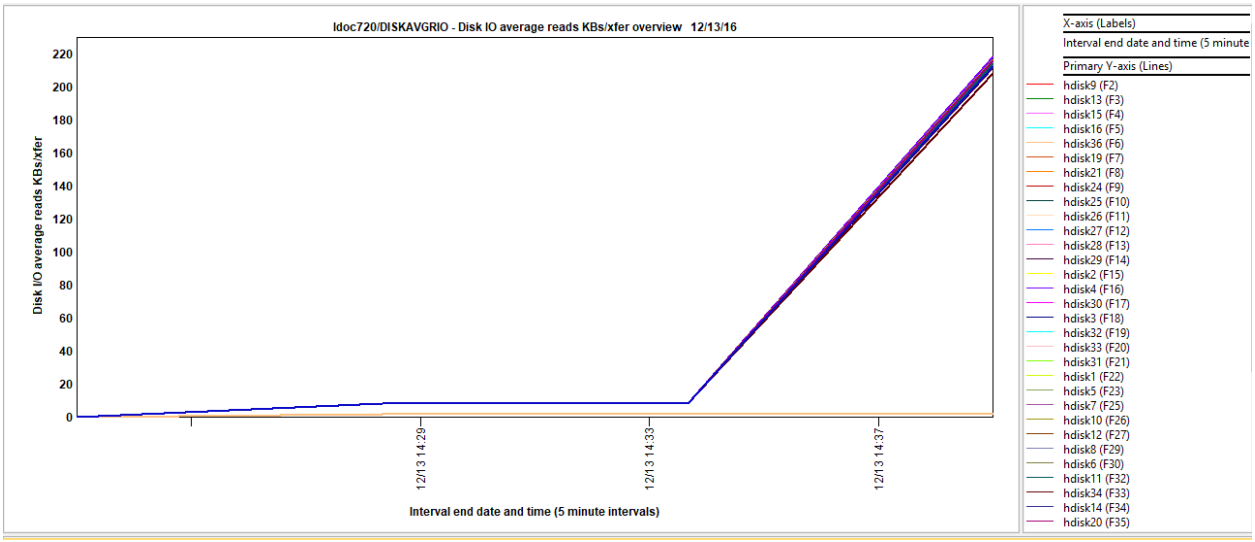
This graph shows the average disk read KBs per IO per disk and compares that with the min and max values on the Y2.



DISKAVGRIO - Disk IO Average reads KBs/xfer by disk

7.10.4.2.13 DISKAVGRIO – Disk IO Average reads KBs/xfer overview

This graph displays the average disk read KBs per IO over time. **Note:** It will only work well for a limited number of disks. If you have more than 48 disks only the 1st 48 disks in the data will be shown.



7.10.4.2.14 DISKAVGWIO - Disk IO Average writes per second by disk

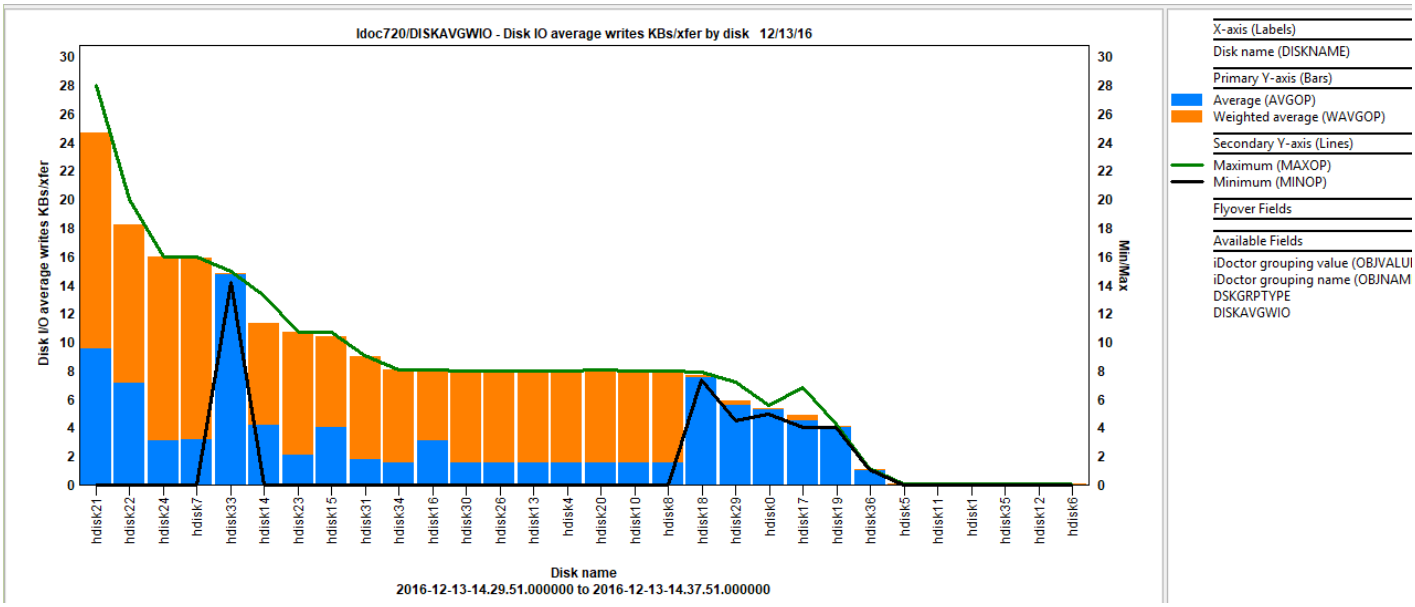
NOTE: AIX documentation here states the following which means the current nmon produced description "Disk IO Average Writes per second" for this field is **wrong**:

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.prftools/io_stat.htm

DISKAVGWIO, Disk IO Average Writes KBs/xfer

Average number of KBs that are written to the disk per write I/O operation.

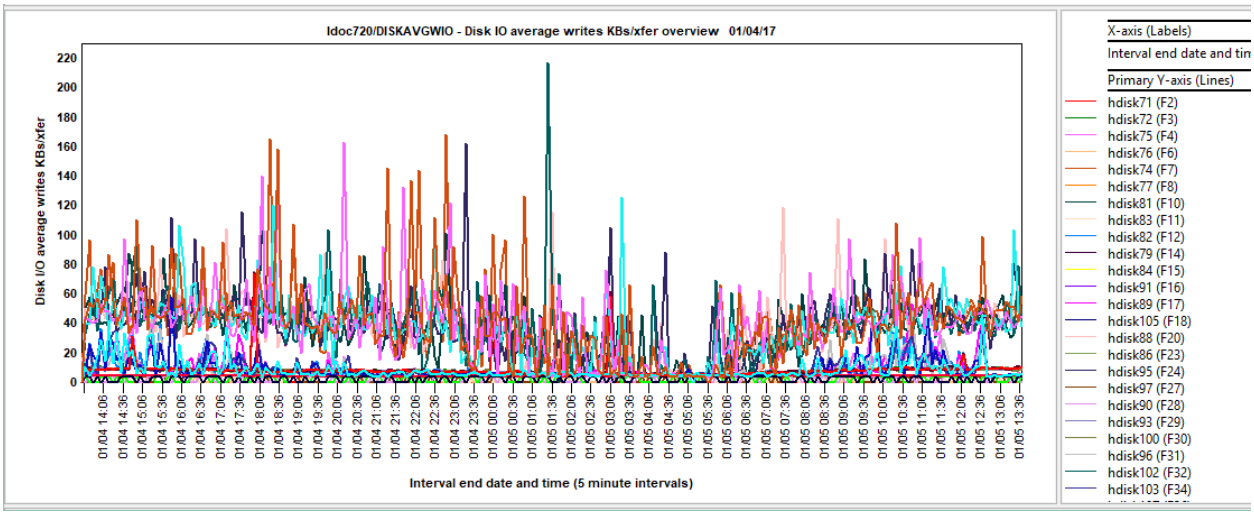
This graph shows the average disk writes KBs per IO per disk and compares that with the min and max values on the Y2.



DISKAVGRIO - Disk IO Average writes KBs/xfer by disk

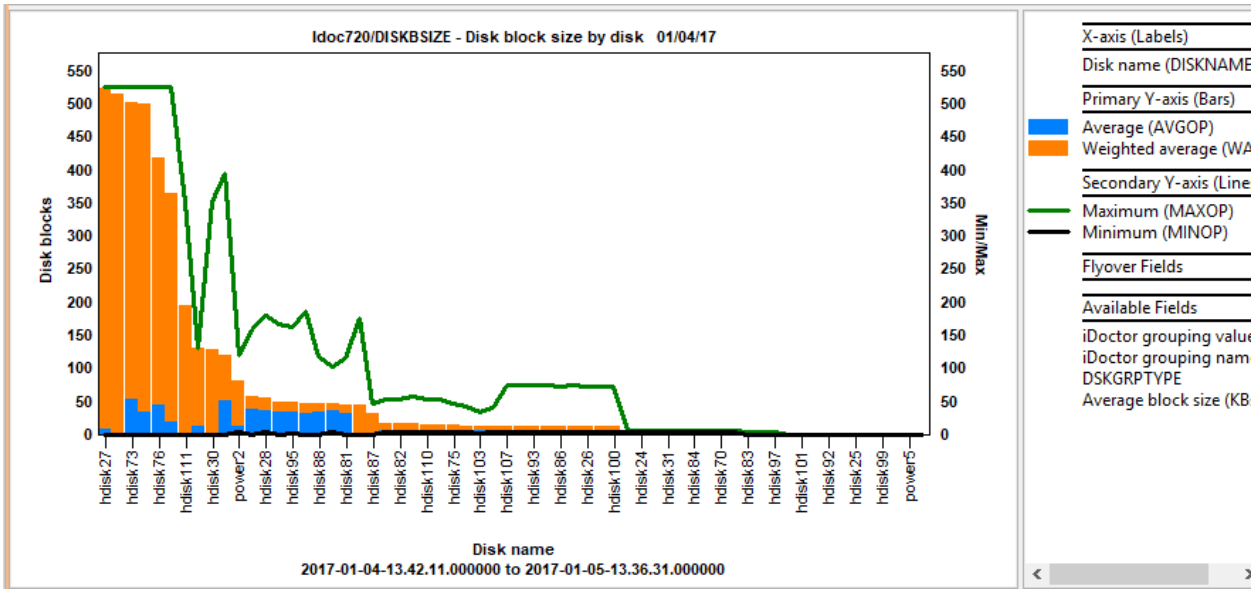
7.10.4.2.15 DISKAVGWIO – Disk IO Average writes KBs/xfer overview

This graph displays the average disk read KBs per IO over time. **Note:** It will only work well for a limited number of disks. If you have more than 48 disks only the 1st 48 disks in the data will be shown.



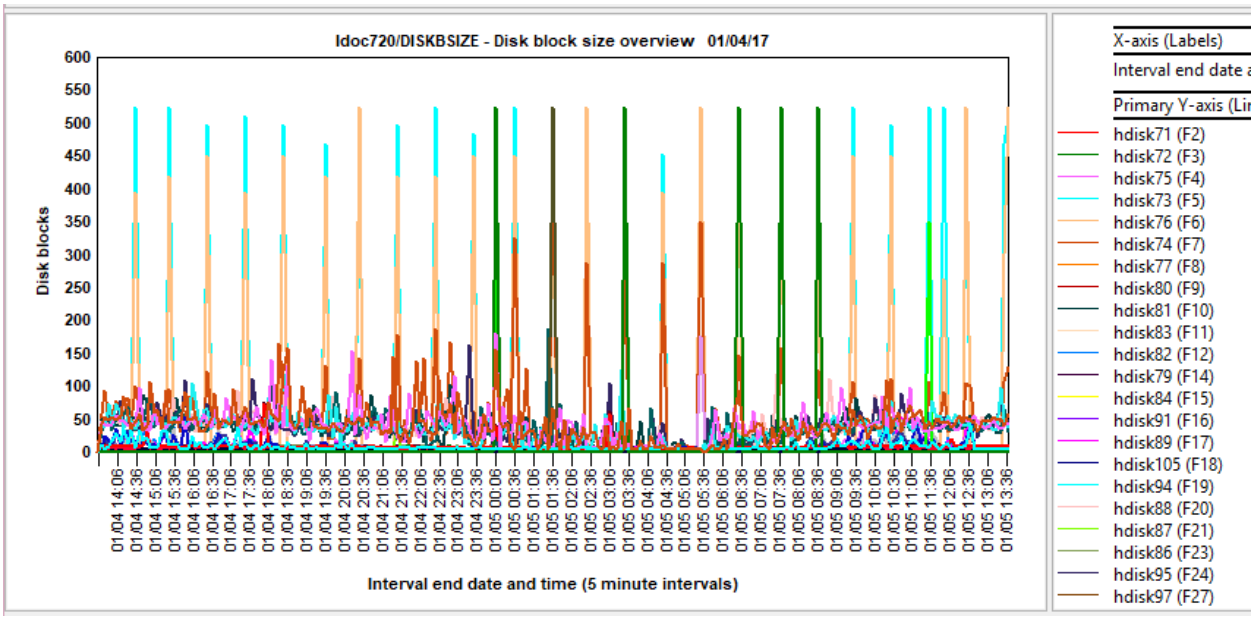
7.10.4.2.16 DISKBSIZE – Disk block size by disk

This graph shows the average disk blocks read and written per disk and compares that with the min and max values on the Y2. If this number is not very close to the stripe size for the device there may be a problem that could be solved by increasing the value of numclust.



7.10.4.2.17 DISKBSIZE – Disk block size overview

This graph displays each disk's total disk blocks read/written over time. **Note:** It will only work well for a limited number of disks. If you have more than 48 disks only the 1st 48 disks in the data will be shown.



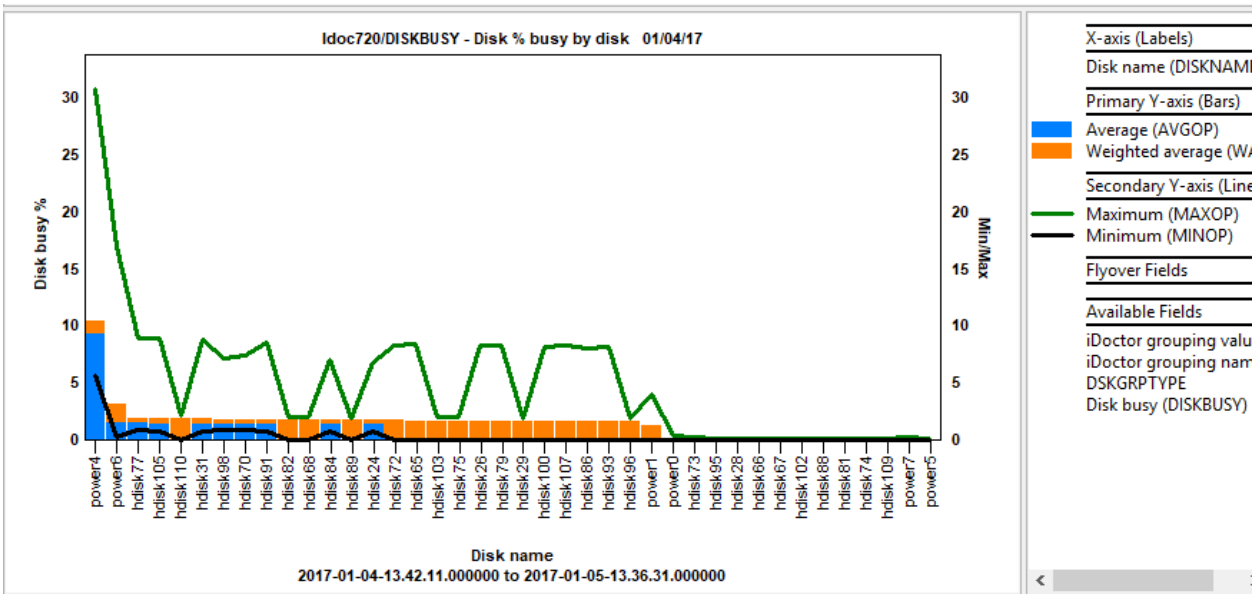
7.10.4.2.18 DISKBUSY – Disk % busy by disk

This graph shows the average disk percent busy and compares that with the min and max values on the Y2.

This is the same as the %tm_act value recorded by iostat.

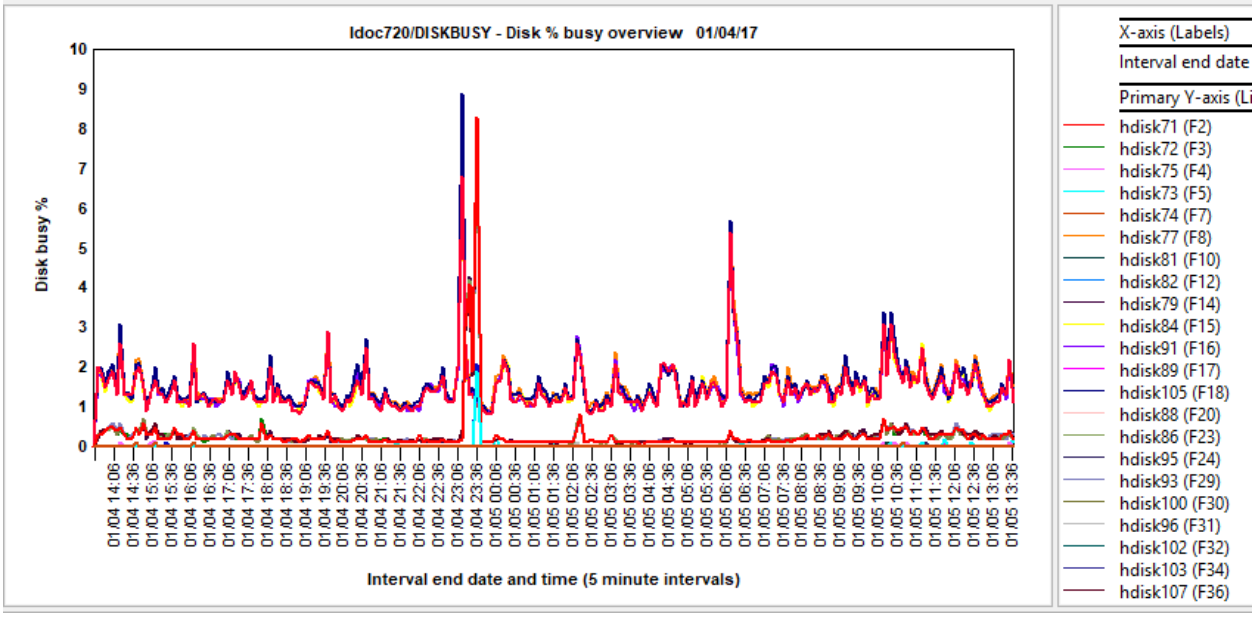
Note: if this graph contains all zero values then it means you forgot to enable iostat collection before starting nmon:

```
chdev -l sys0 -a iostat=true
```



7.10.4.2.19 DISKBUSY – Disk % busy overview

This graph displays each disk's percent busy over time as a line graph. **Note:** It will only work well for a limited number of disks. If you have more than 48 disks only the 1st 48 disks in the data will be shown.

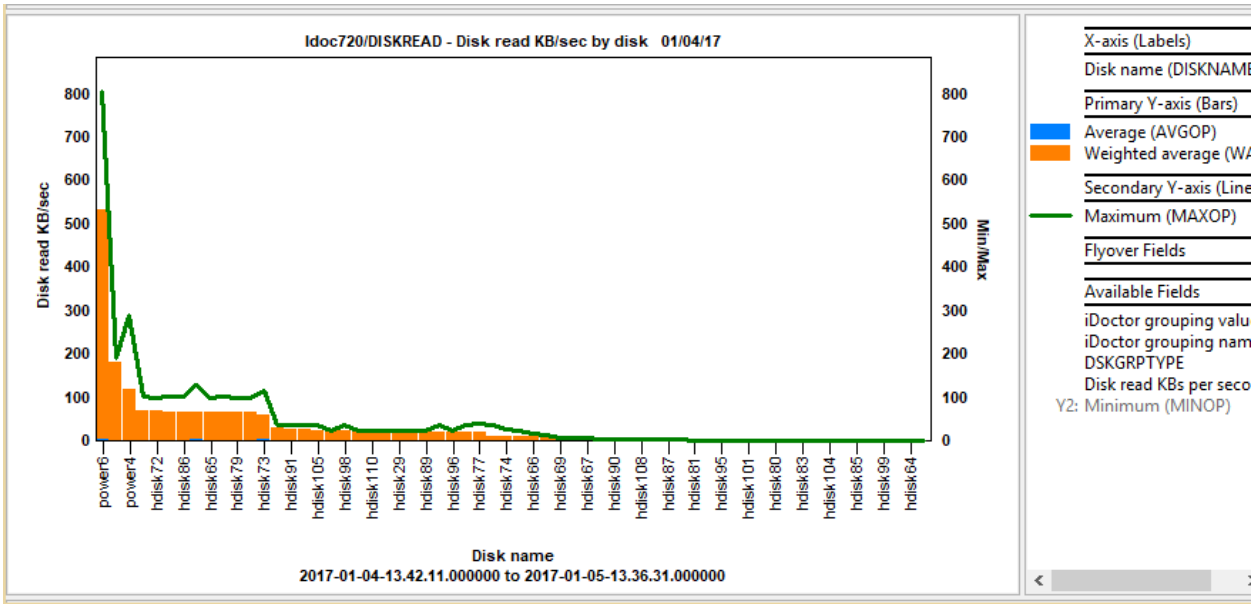


7.10.4.2.20 DISKREAD – Disk read KB/sec by disk

`DISKREAD, Disk Read KB/s`

Total read operations from the disk in KBs per second.

This graph shows the average disk read KB/sec and compares that with the min and max values on the Y2.

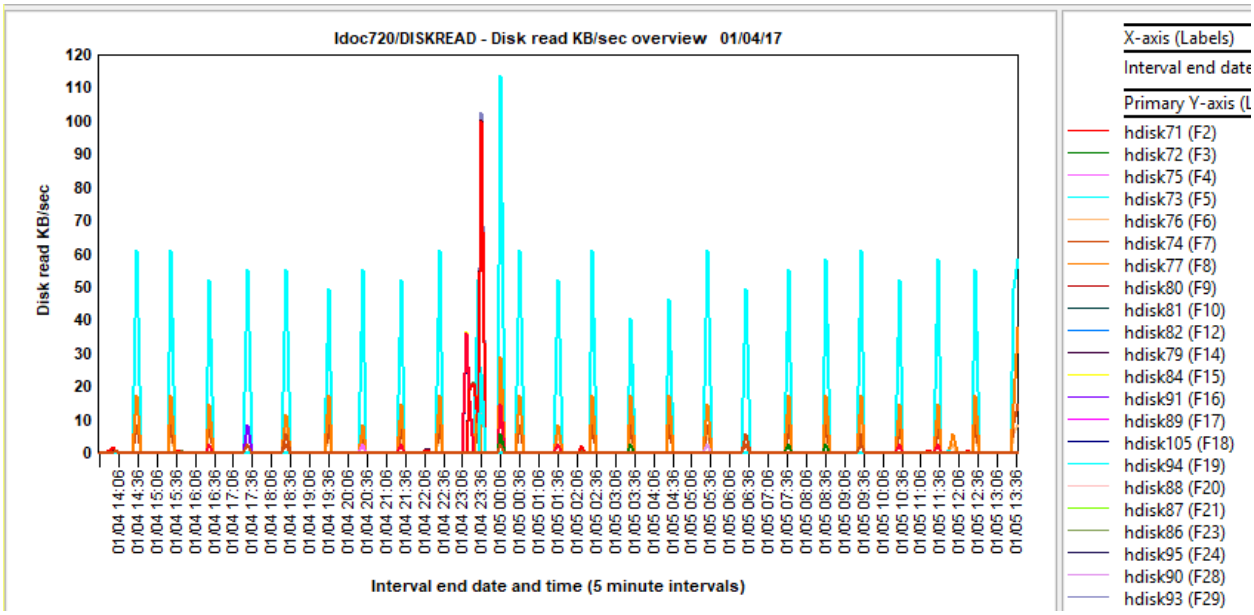


7.10.4.2.21 DISKREAD – Disk read KB/sec overview

`DISKREAD, Disk Read KB/s`

Total read operations from the disk in KBs per second.

This graph shows the average disk read KBs/sec over time as a line graph. **Note:** It will only show a maximum of 48 disks.

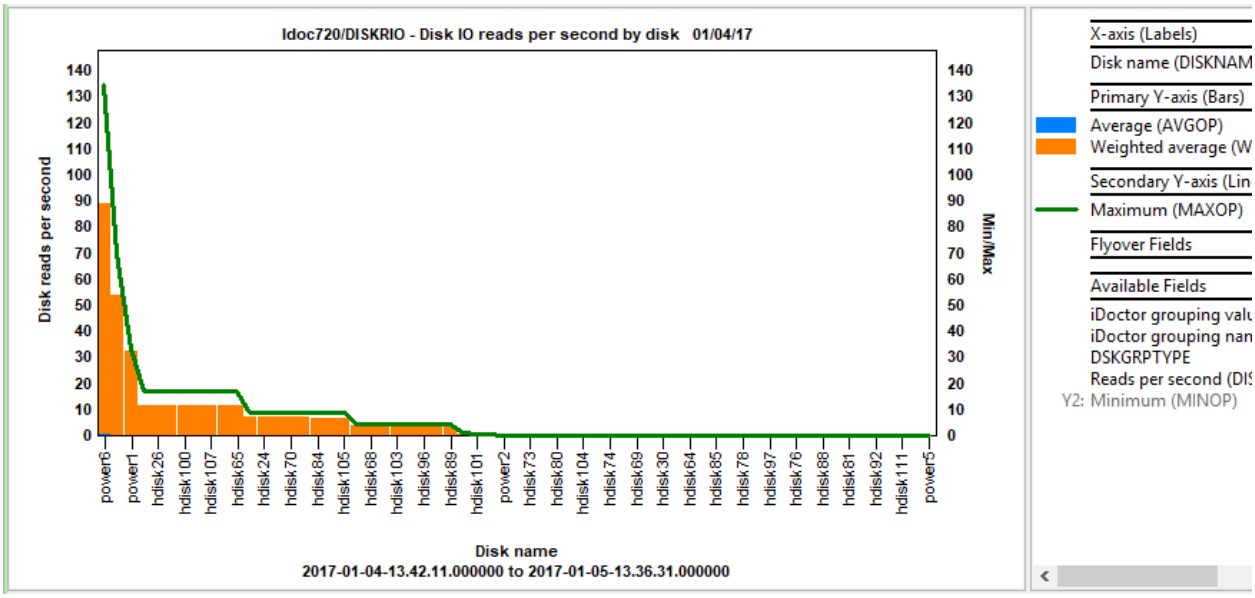


7.10.4.2.22 DISKRIO – Disk IO reads per second by disk

`DISKRIO, Disk IO Reads per second`

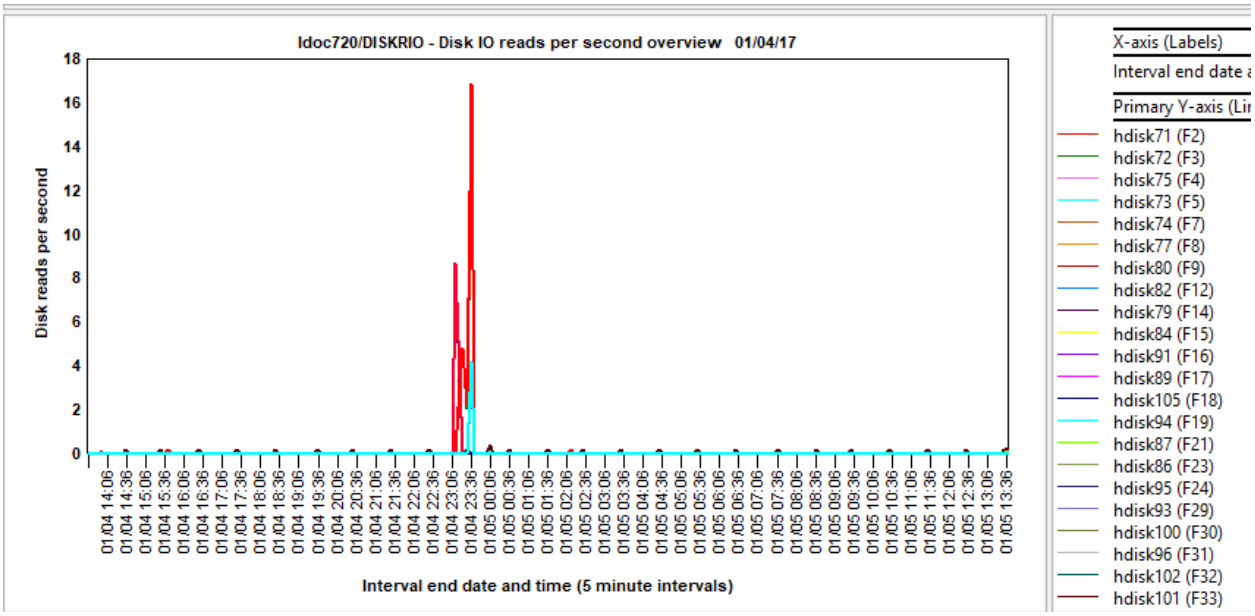
Number of disk read I/O transfers per second.

This graph shows the average disk reads per second and compares that with the min and max values on the Y2.



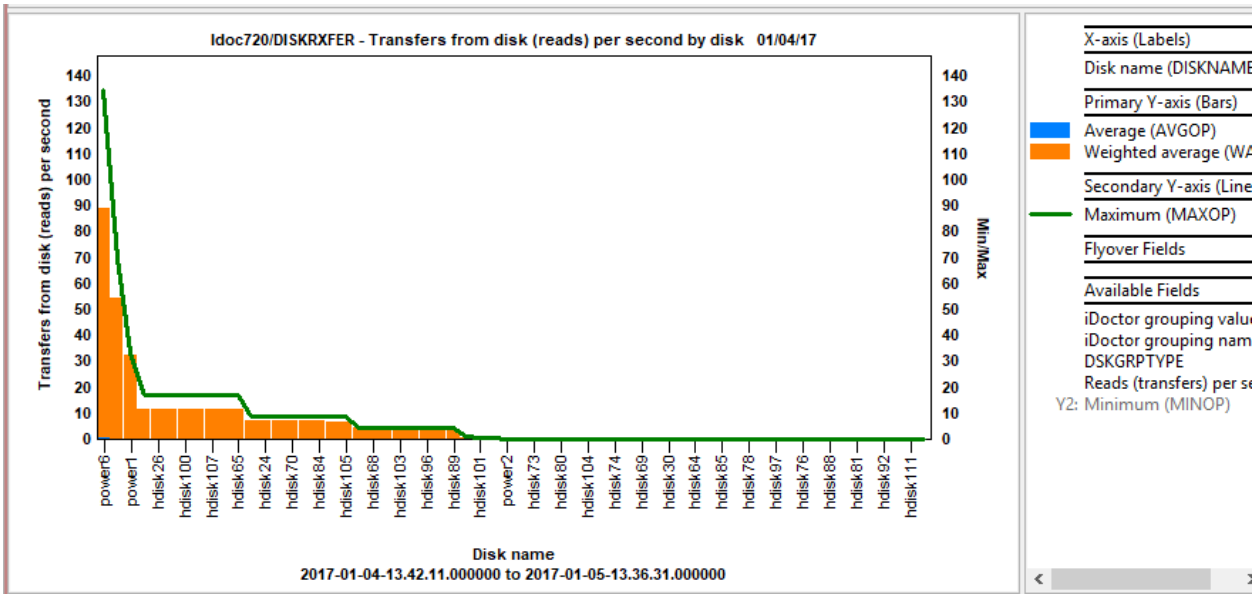
7.10.4.2.23 DISKRIO – Disk IO reads per second overview

This graph shows the average disk read per second overtime as a line graph. **Note:** It will only show a maximum of 48 disks.



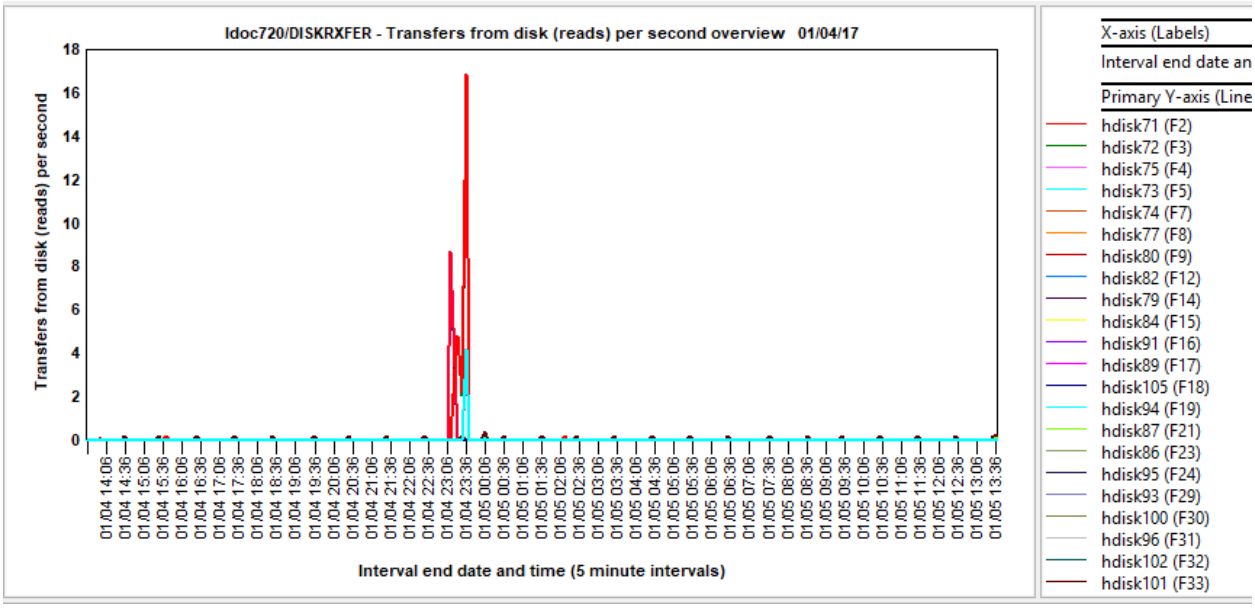
7.10.4.2.24 DISKRXFER – Transfer from disk (reads) per second by disk

This graph shows the average transfer from disk reads per second and compares that with the min and max values on the Y2.



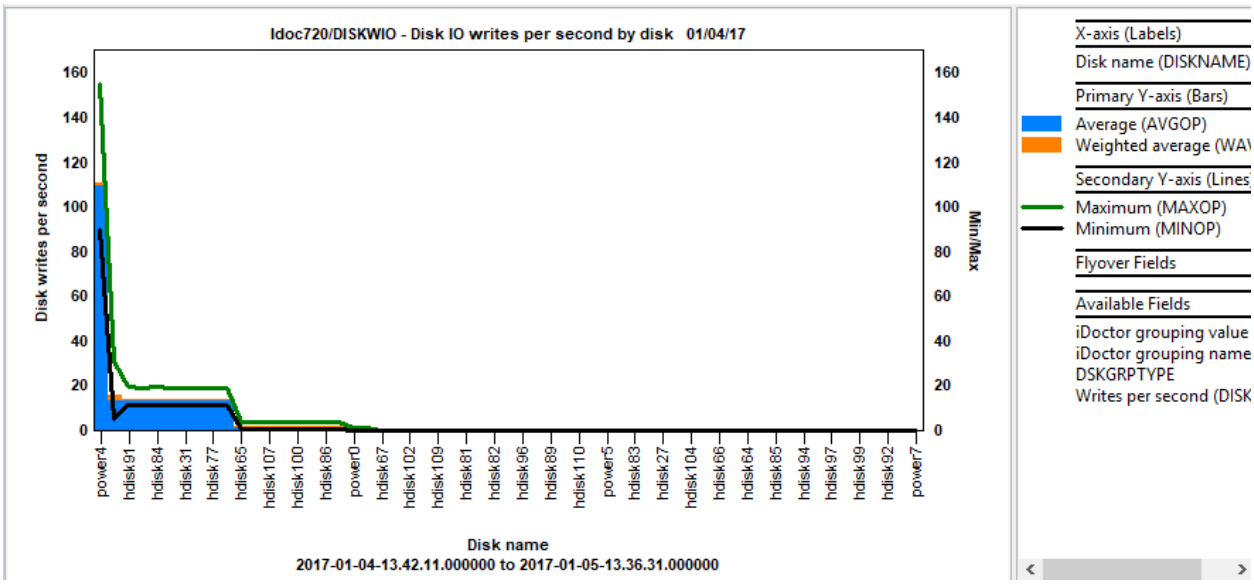
7.10.4.2.25 DISKRXFER – Transfer from disk (reads) per second overview

Note: It will only show a maximum of 48 disks.



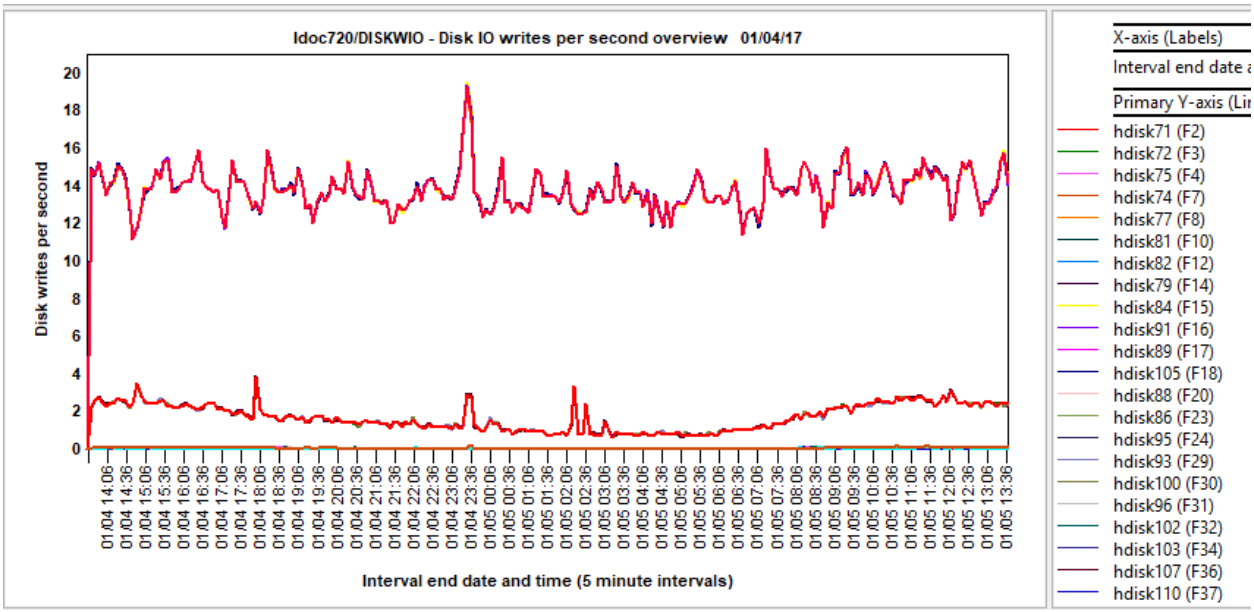
7.10.4.2.26 DISKWIO – Disk IO writes per second by disk

This graph shows the average disk writes per second and compares that with the min and max values on the Y2.



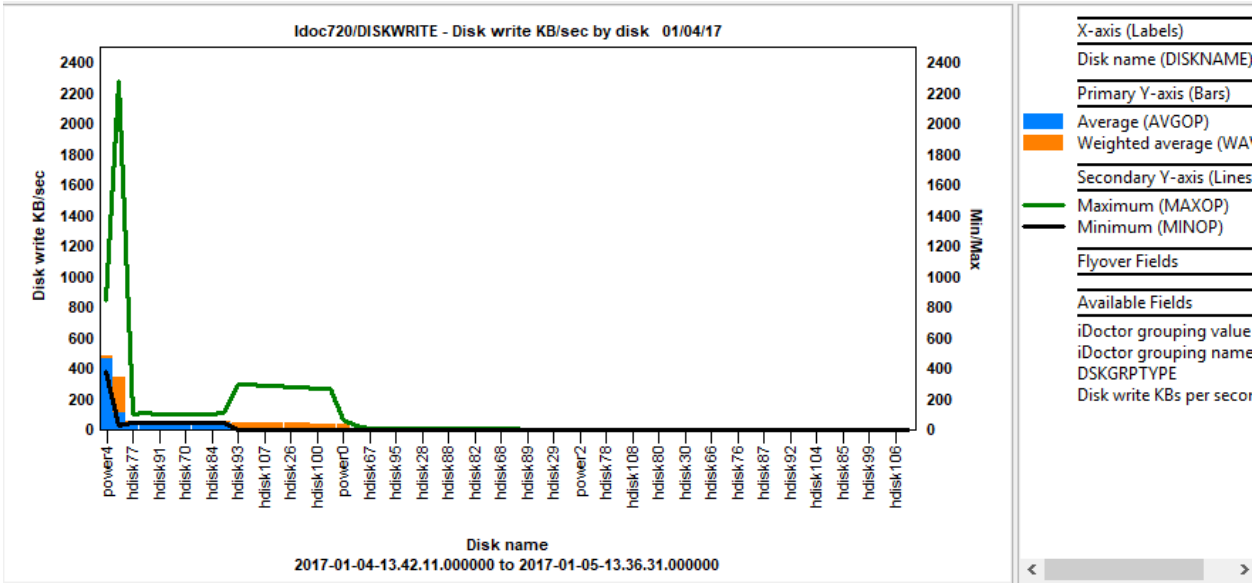
7.10.4.2.27 DISKWIO – Disk IO writes per second overview

Note: It will only show a maximum of 48 disks.



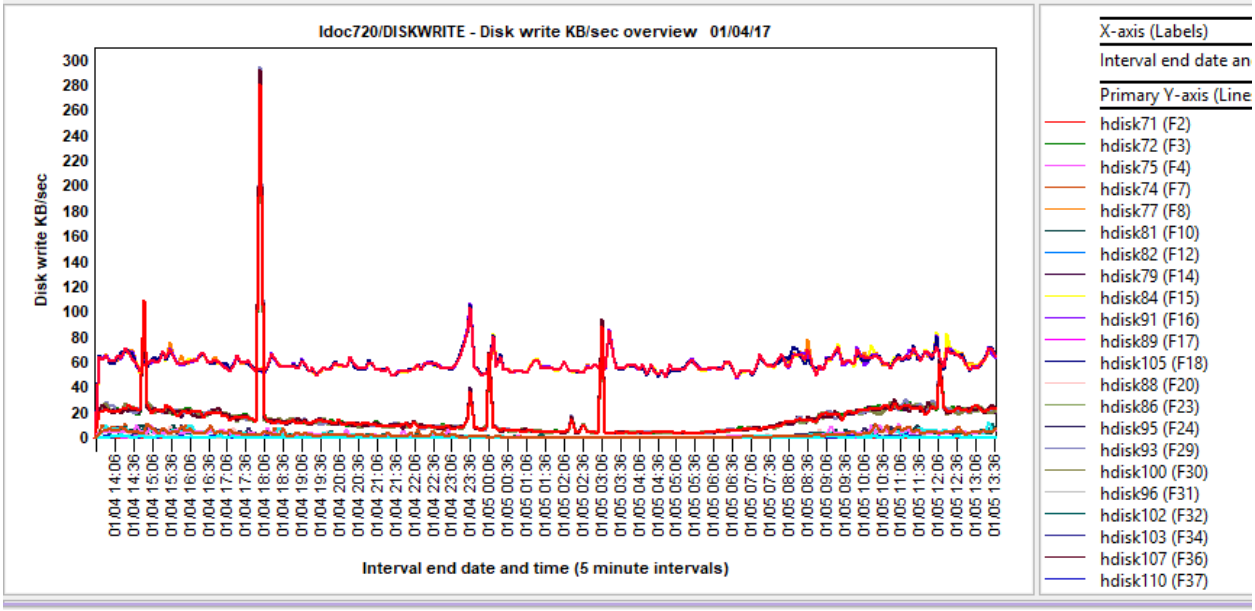
7.10.4.2.28 DISKWRITE – Disk write KB/sec by disk

This graph shows the average disk write size (in KB/sec) and compares that with the min and max values on the Y2.



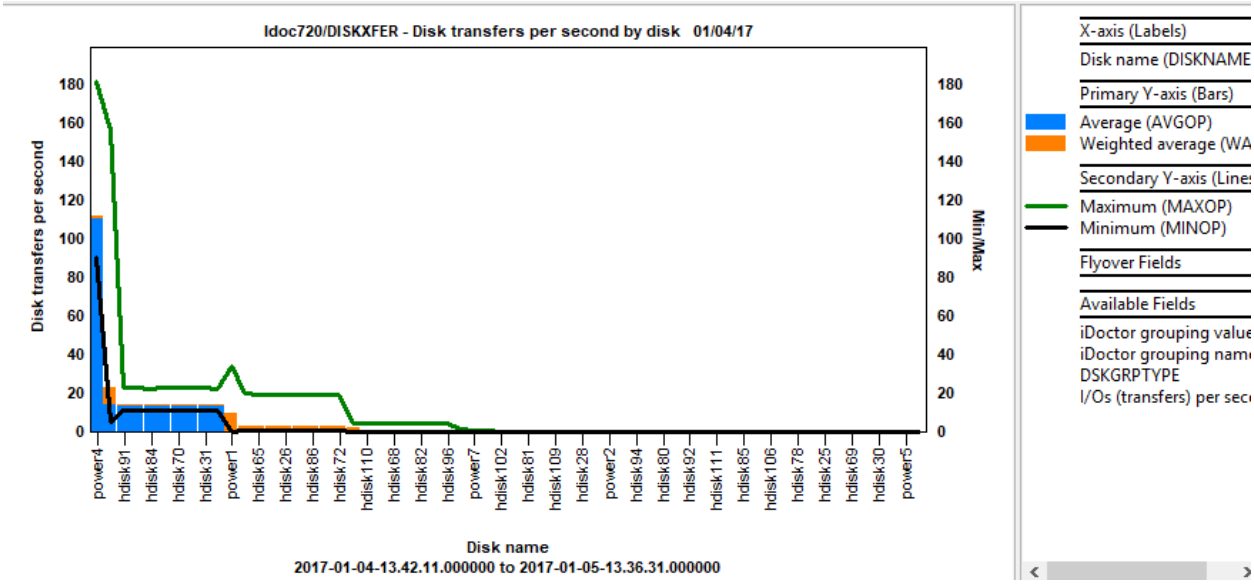
7.10.4.2.29 DISKWRITE – Disk write KB/sec overview

Note: It will only show a maximum of 48 disks.



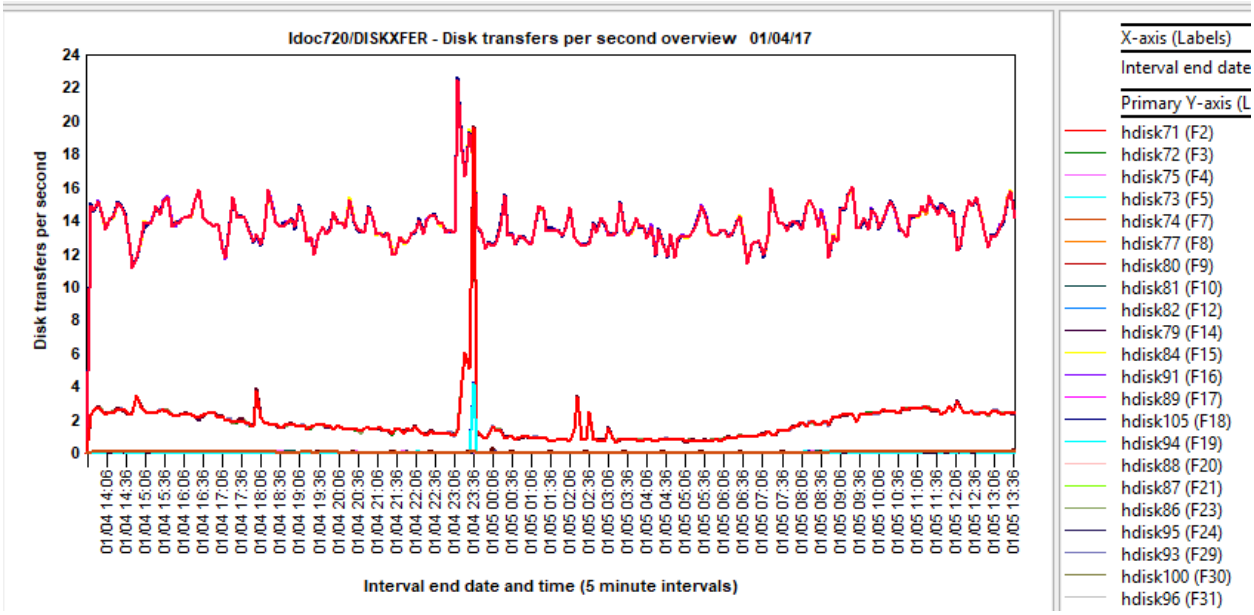
7.10.4.2.30 DISKXFER – Disk transfers per second by disk

This graph shows the average disk transfers per second and compares that with the min and max values on the Y2.



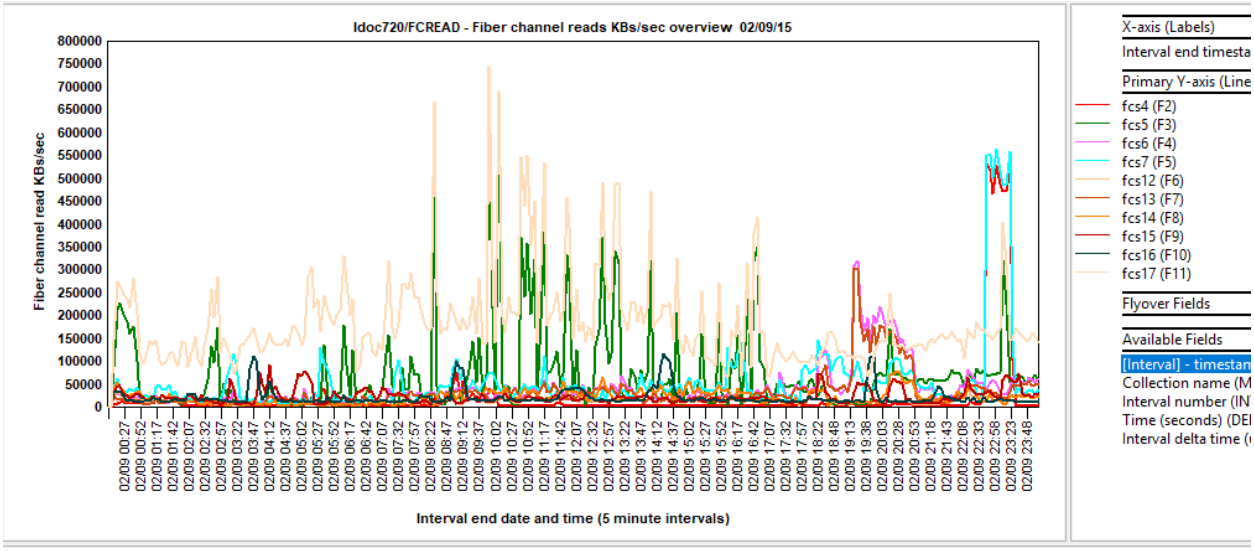
7.10.4.2.31 DISKXFER – Disk transfers per second overview

Note: It will only show a maximum of 48 disks.



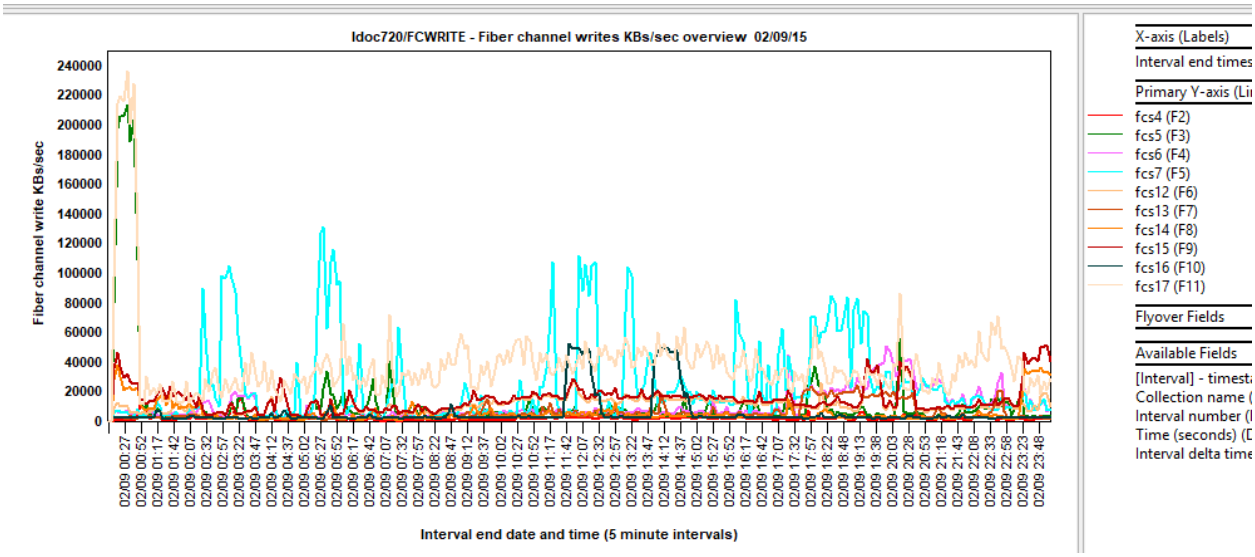
7.10.4.2.32 FCREAD – Fiber channel reads KBs/sec overview

This graph displays the data rate (KBs/sec) for read operations on each Fiber Channel adapter in the system. **Note:** It will only show a maximum of 48 adapters.



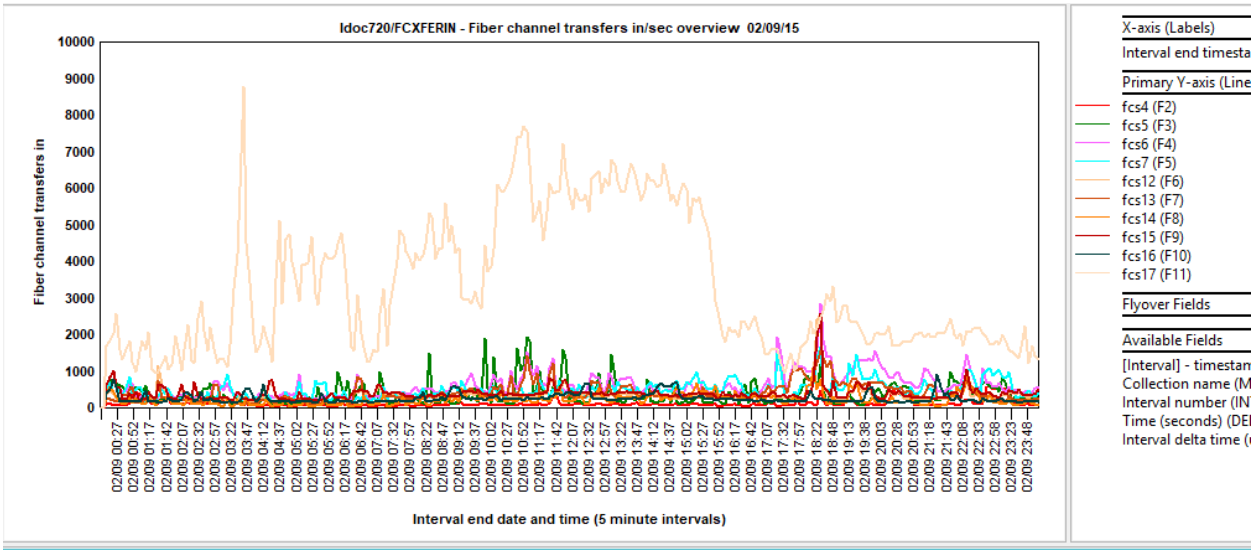
7.10.4.2.33 FCWRITE – Fiber channel writes KBs/sec overview

This graph displays the data rate (KBs/sec) for write operations on each Fiber Channel adapter in the system. **Note:** It will only show a maximum of 48 adapters.



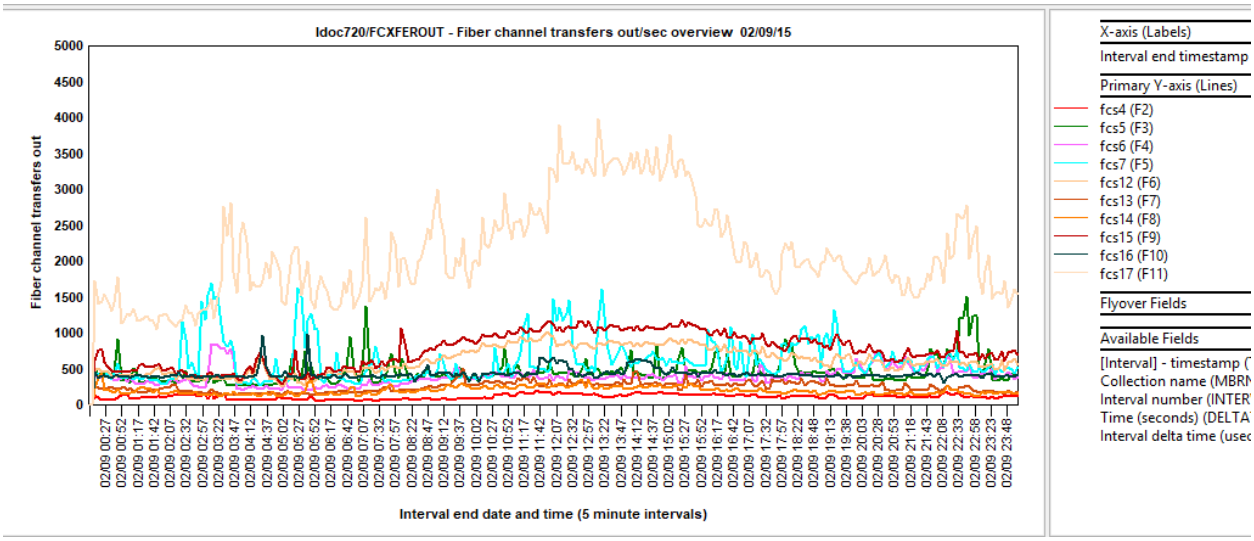
7.10.4.2.34 FCXREFIN – Fiber channel transfers in/sec overview

This graph displays the read operations per second on each Fiber Channel adapter in the system. **Note:** It will only show a maximum of 48 adapters.



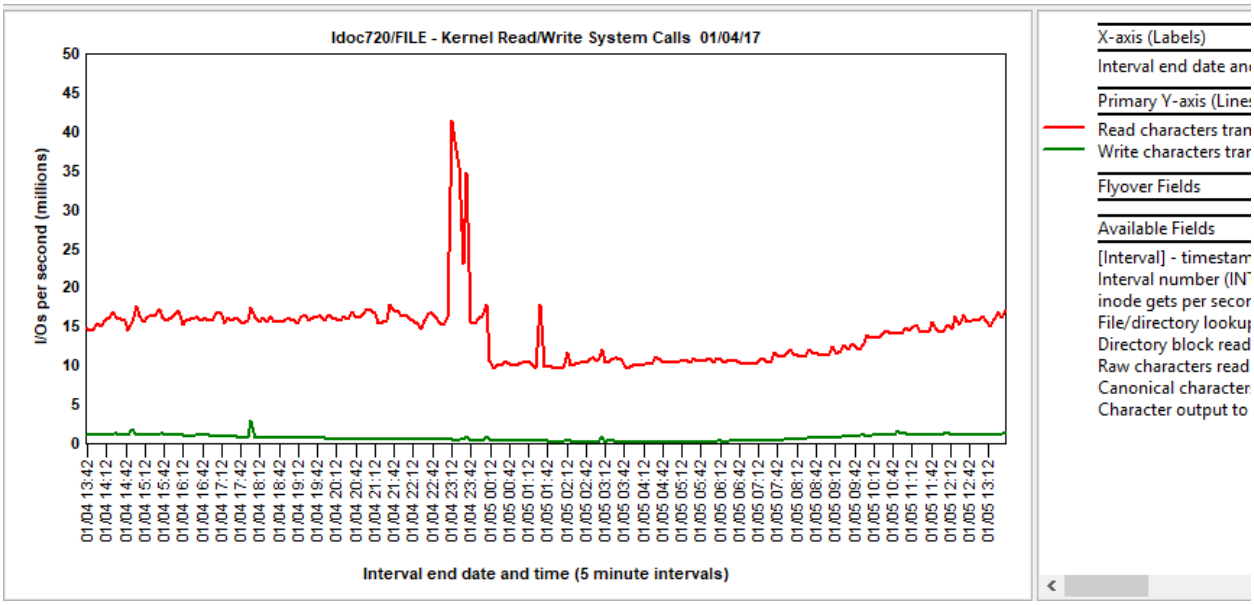
7.10.4.2.35 FCXREFOUT – Fiber channel transfers out/sec overview

This graph displays the write operations per second on each Fiber Channel adapter in the system. **Note:** It will only show a maximum of 48 adapters.



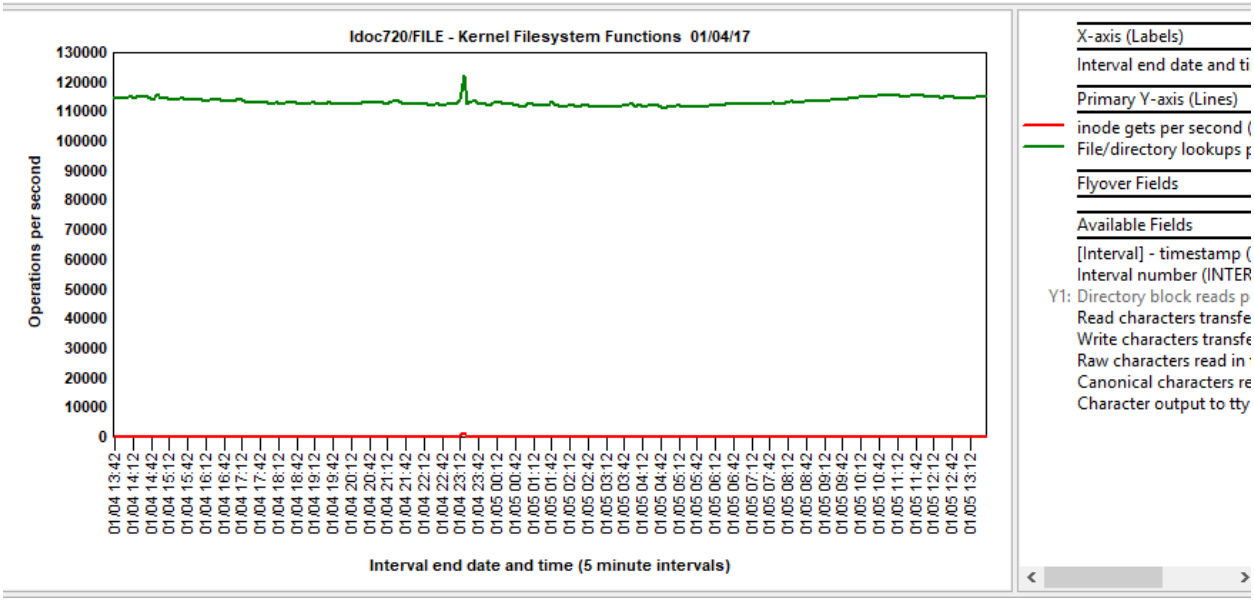
7.10.4.2.36 FILE – Kernel Read/Write System Calls

This graph displays the number of characters transferred per second (in millions) for both read and write system calls.



7.10.4.2.37 FILE – Kernel Filesystem Functions

This graph displays operations per second for inode gets, file/directory lookups and directory block reads.

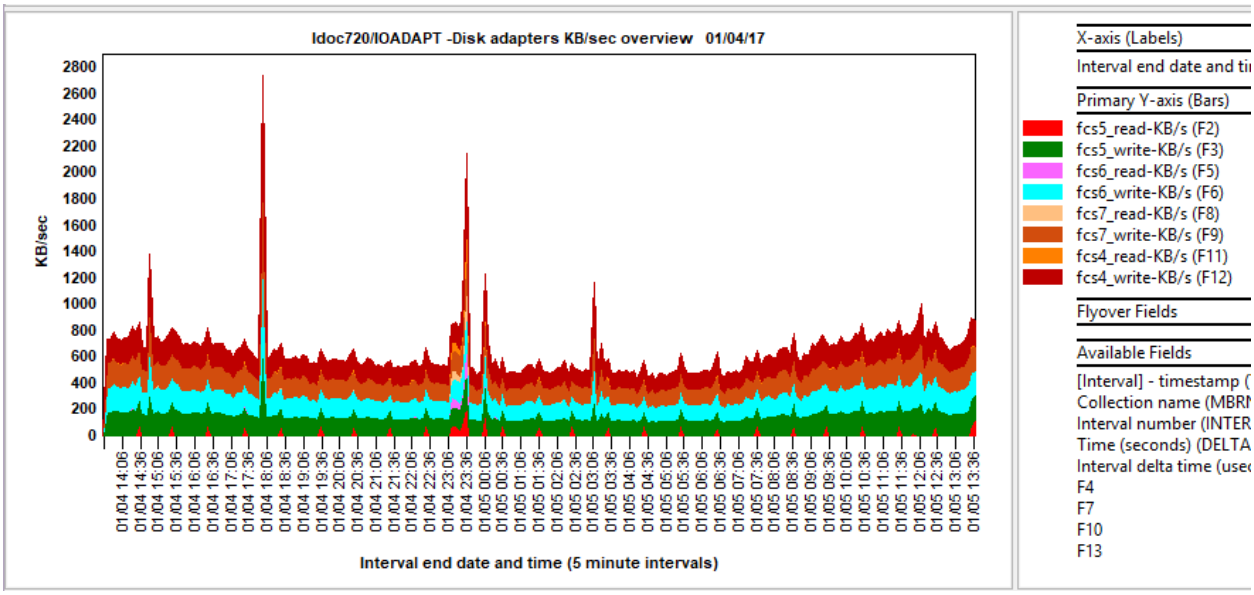


7.10.4.2.38 IOADAPT -Disk adapters KB/sec overview

For each I/O adapter listed on the BBBC sheet, contains the data rates for both read and write operations (Kbytes/sec) and total number of I/O operations performed. On AIX 5.1 and later, this information is reported by the iostat -A command.

Note: These charts can be easily converted to lines (or bars) by right-clicking the graph and using the **Graph definition -> Set Graph type** menu.

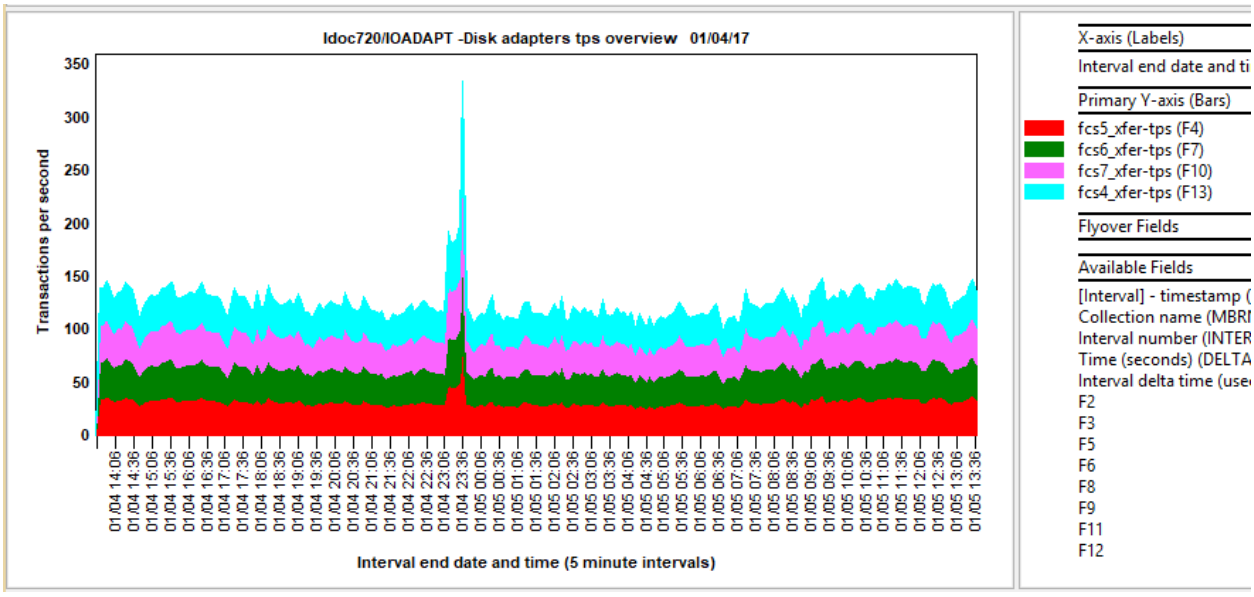
Note: It will only show a maximum of 48 adapters.



7.10.4.2.39 IOADAPT -Disk adapters tps overview

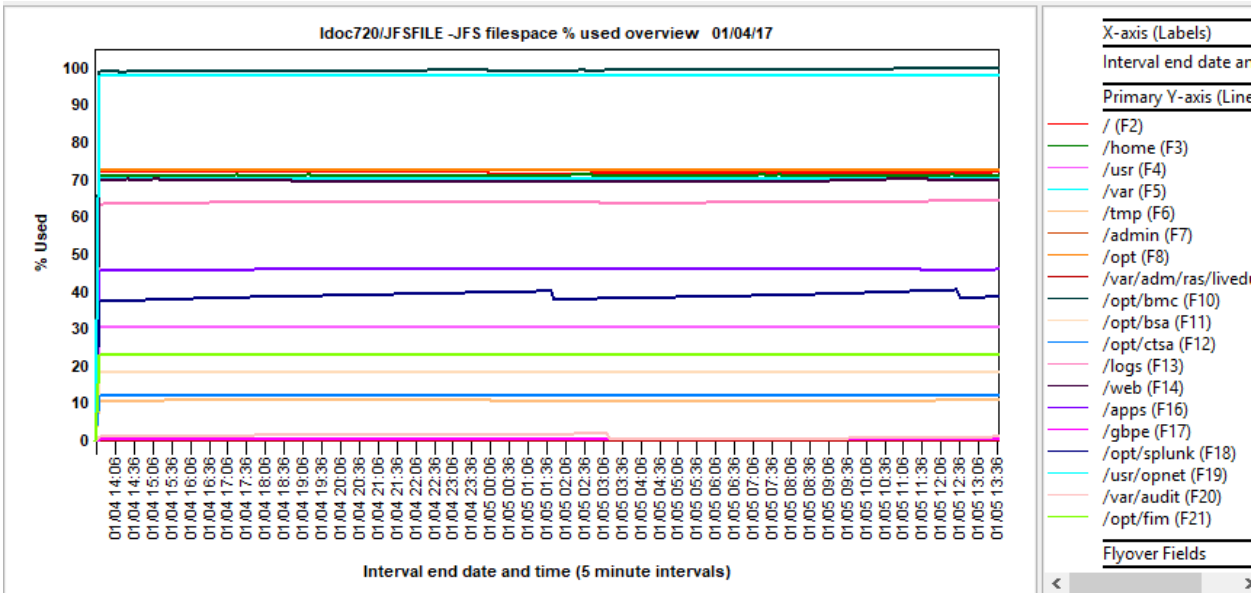
For each I/O adapter listed on the BBBC sheet, contains the data rates for both read and write I/O operations performed. **Note:** These charts can be easily converted to lines (or bars) by right-clicking the graph and using the **Graph definition -> Set Graph type** menu.

Note: It will only show a maximum of 48 adapters.



7.10.4.2.40 JFSFILE -JFS filespace % used overview

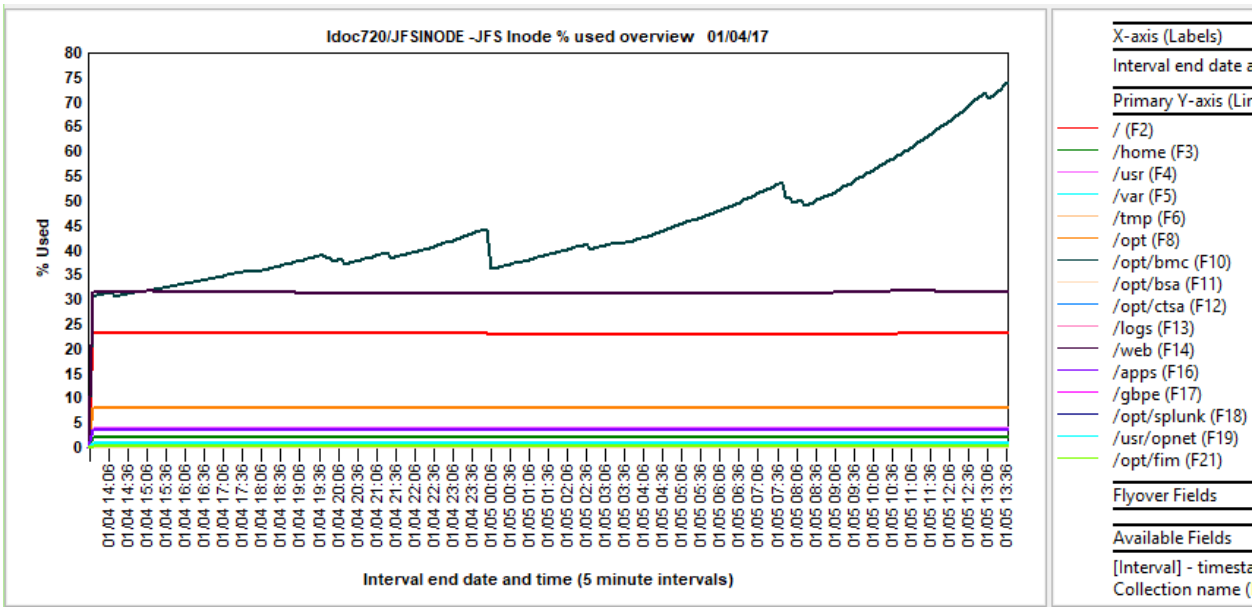
For each file system, this graph shows what percentage of the space allocation is being used during each interval. These figures are the same as the **%Used** value reported by the `df` command. The column headings show the mount point; sheet BBBC can be used to cross-reference to the file system/LV.



JFSFILE -JFS filespace % used overview

7.10.4.2.41 JFSINODE -JFS Inode % used overview

For each file system, this sheet shows what percentage of the Inode allocation is being used during each interval. These figures are the same as the **%Iused** value reported by the df command. The column headings show the mount point; sheet BBBC can be used to cross-reference to the file system/LV.



JFSINODE -JFS Inode % used overview

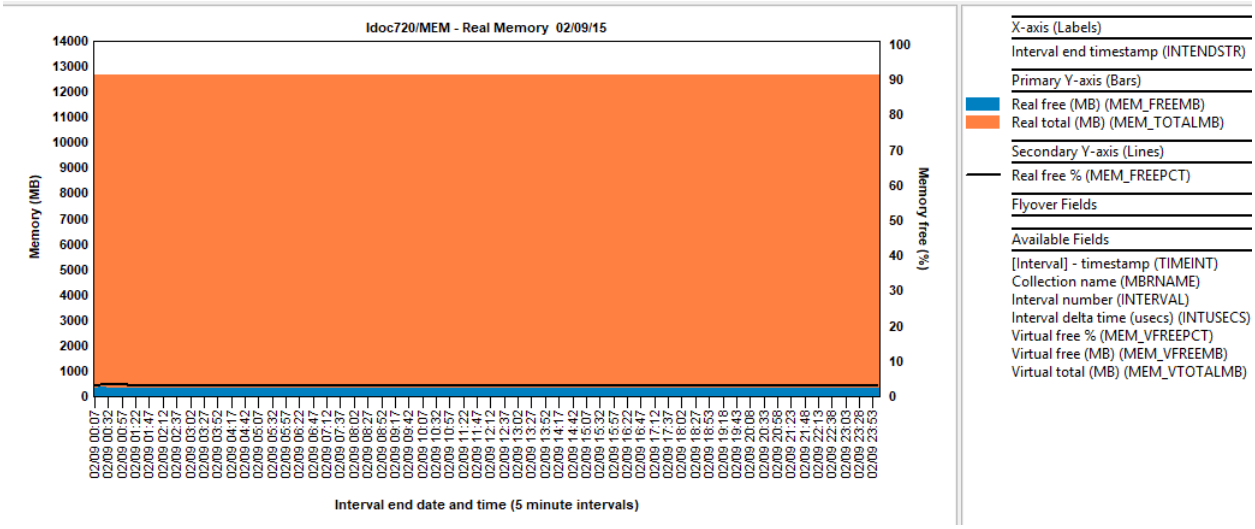
7.10.4.3 nmon Analyzer – M-Z Sheets

These graphs cover nmon Analyzer sheets that start with the letters M through Z (excluding SYS_SUMM.)

Note: Some graphs will NOT appear depending on what data exists in the original nmon file. If the data tag does not exist, then the graphs will not appear here.

7.10.4.3.1 MEM - Real Memory

This graph shows the amount of Real Free memory (and total) in Mbytes by time of day.



MEM - Real Memory

7.10.4.3.2 MEMNEW - Memory Use

The graph shows the allocation of memory split into the three major categories: pages used by user processes, file system cache, and pages used by the system (kernel).

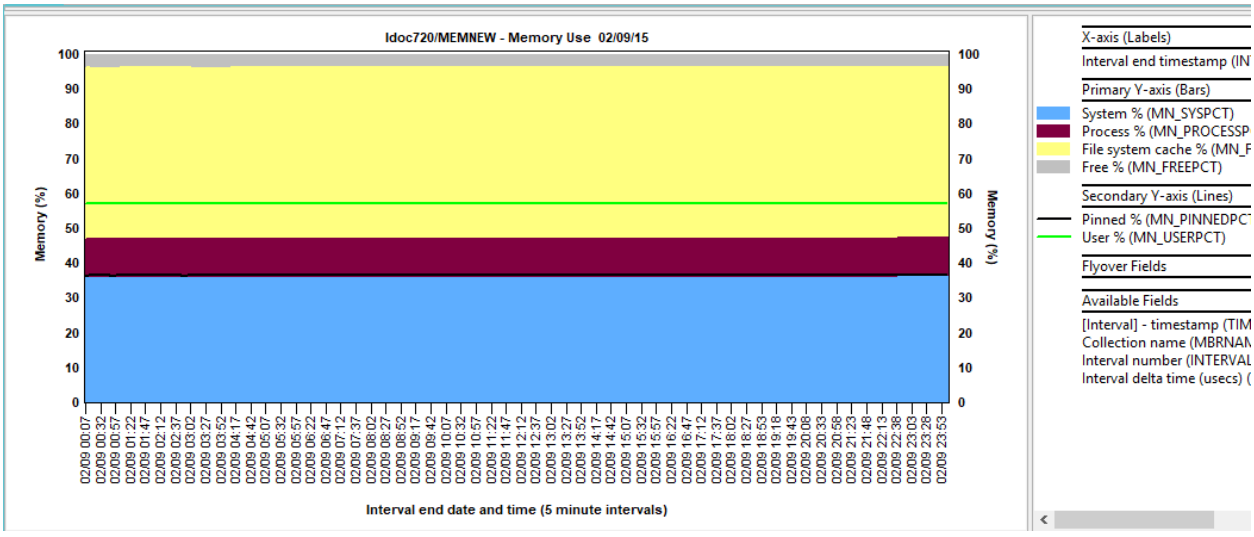
Process % the percentage of real memory allocated to user processes

FS Cache% the percentage of real memory allocated to file system cache

System % the percentage of real memory used by system segments

Free % the percentage of unallocated real memory

User % the percentage of real memory used by non-system segments



MEMNEW - Memory Use

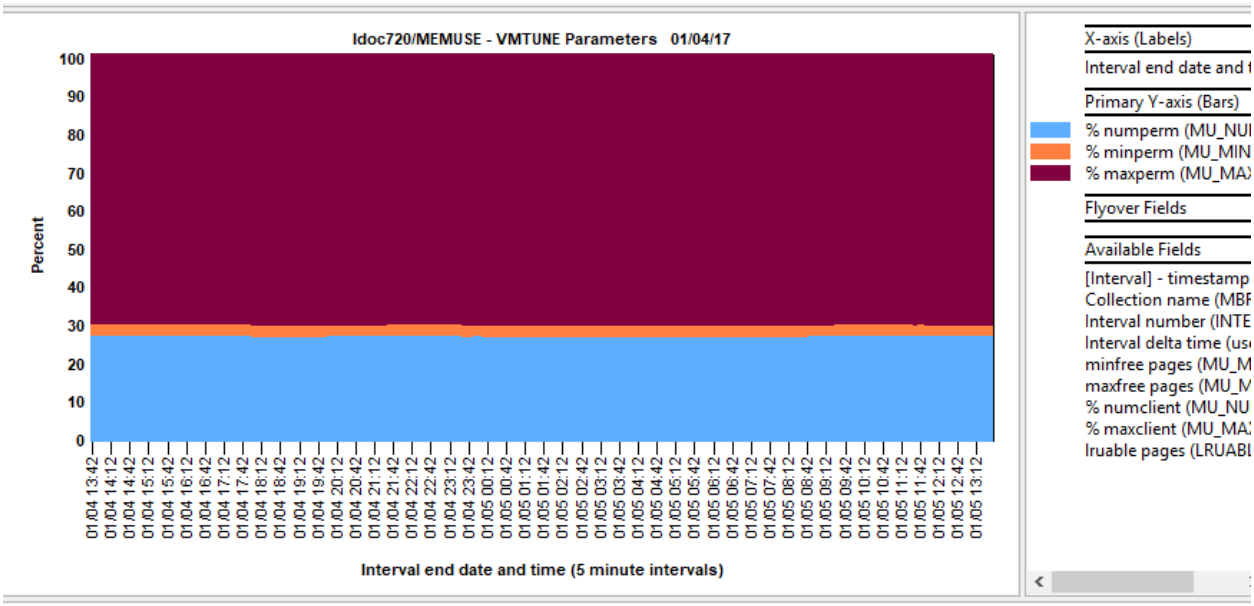
7.10.4.3.3 MEMUSE - VMTUNE Parameters

Except for **%comp**, the values on this sheet are the same as would be reported by the vmtune command.

%numperm the percentage of real memory allocated to file pages**.**

%minperm value specified on the vmtune command or system default of 20%. This will normally be constant for the run unless the vmtune or rmss commands are used during collection.

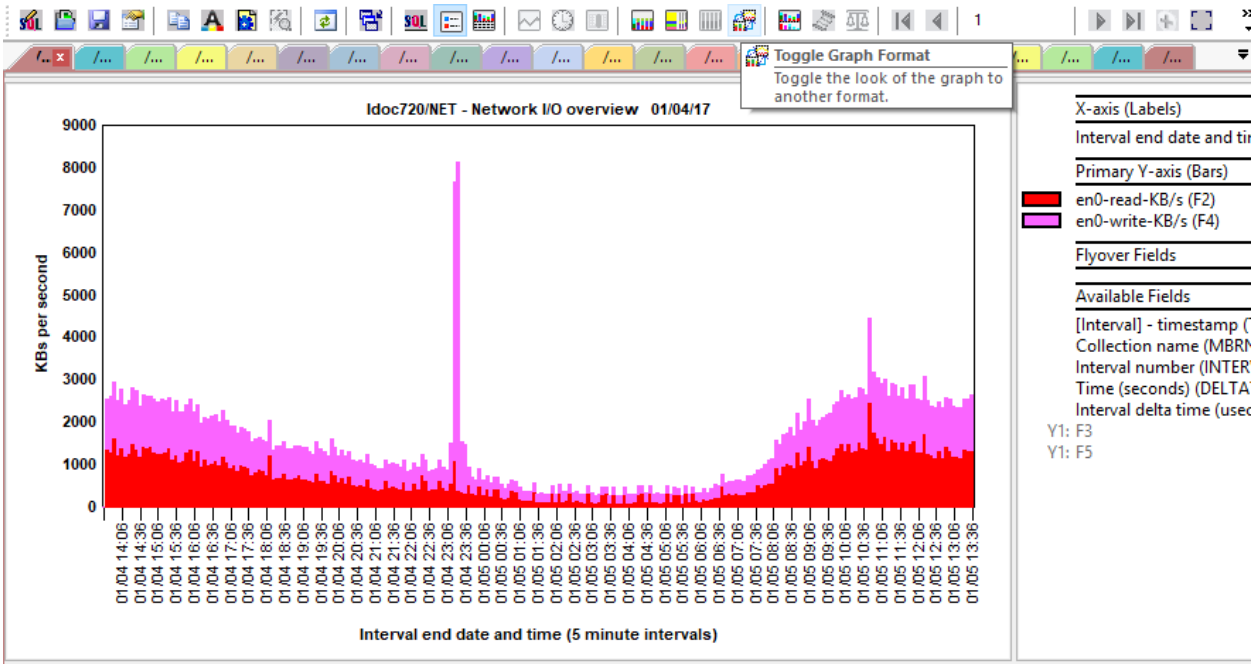
%maxperm value specified on the vmtune command or system default of 80%. This will normally be constant for the run unless the vmtune or rmss commands are used during collection.



MEMUSE - VMTUNE Parameters

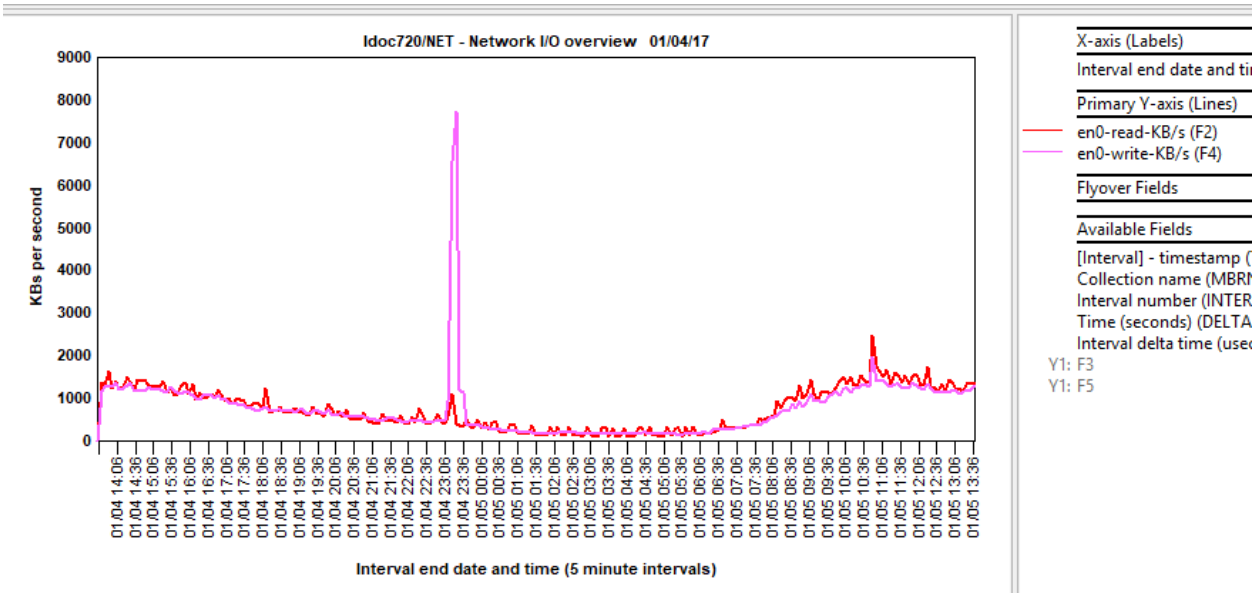
7.10.4.3.4 NET - Network I/O overview

This sheet shows the data rates, in Kbytes/sec, for each network adapter in the system (including SP switch if present). This is the same as produced by the netpmon –O dd command. The first graph shows total network traffic broken down as Total-Read and Total-Write.



NET - Network I/O overview (as bars)

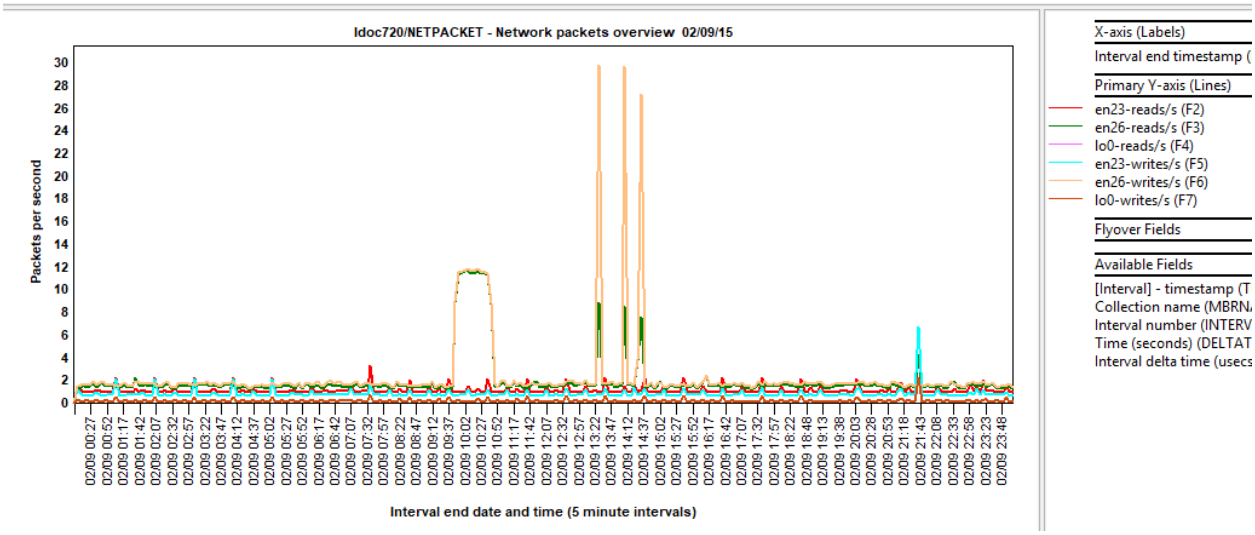
Note: These charts can be easily converted to lines (or bars) by clicking the Toggle Graph Format button on the toolbar (see previous screenshot.)



NET - Network I/O overview (as lines)

7.10.4.3.5 NETPACKET - Network packets overview

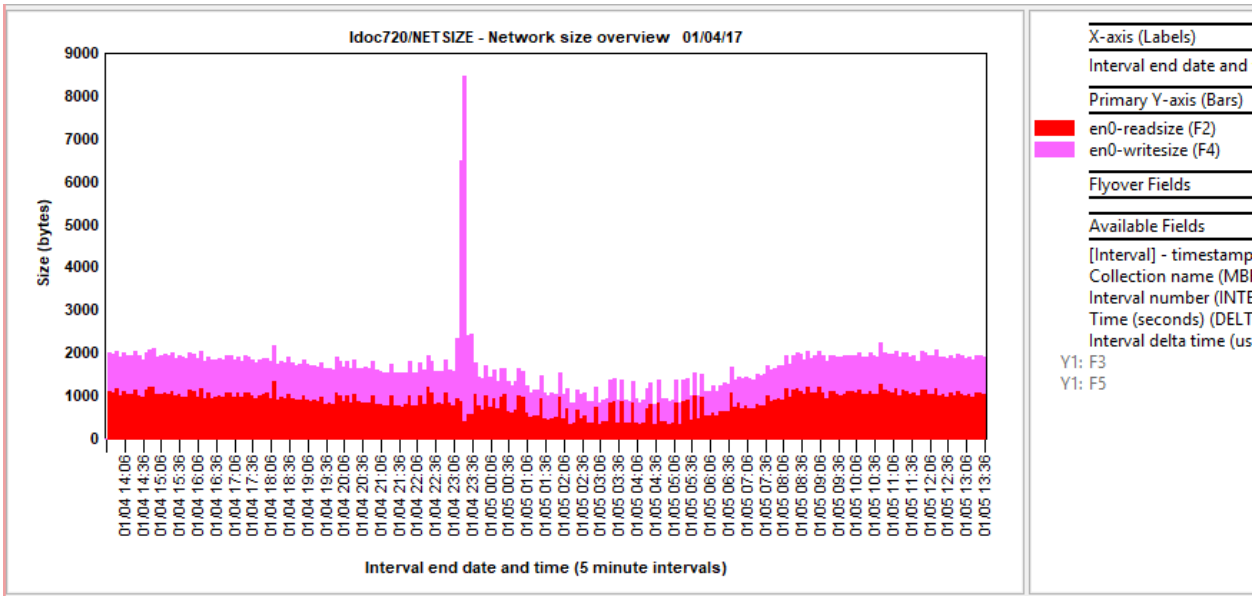
This sheet shows the number of read/write network packets for each adapter. This is the same as produced by the netpmon –O dd command.



NETPACKET - Network packets overview

7.10.4.3.6 NETSIZE - Network size overview

This sheet shows the average packet size in bytes for each network adapter in the system.



NETSIZE – Network size overview

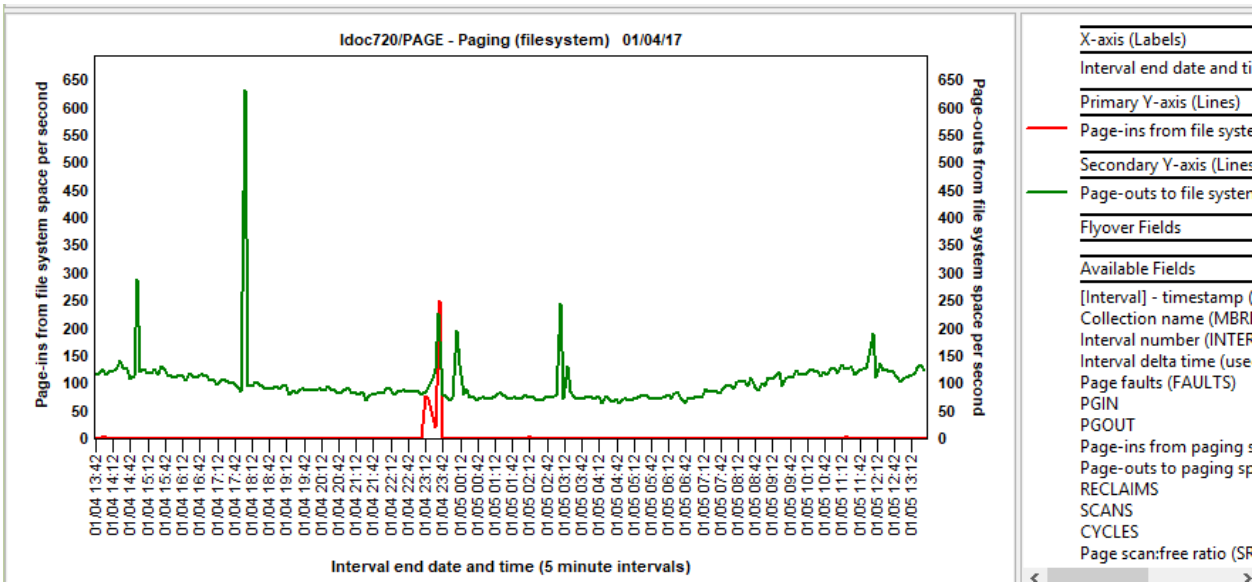
7.10.4.3.7 PAGE - Paging (pgspace)

The graph shows the total rate/sec of page-in/page-out operations to pgspace over time.

If **page-ins** is consistently higher than **page-outs** this may indicate thrashing.

7.10.4.3.8 PAGE - Paging (filesystem)

The graph shows the total rate/sec of page-in/page-out operations to file systems over time.



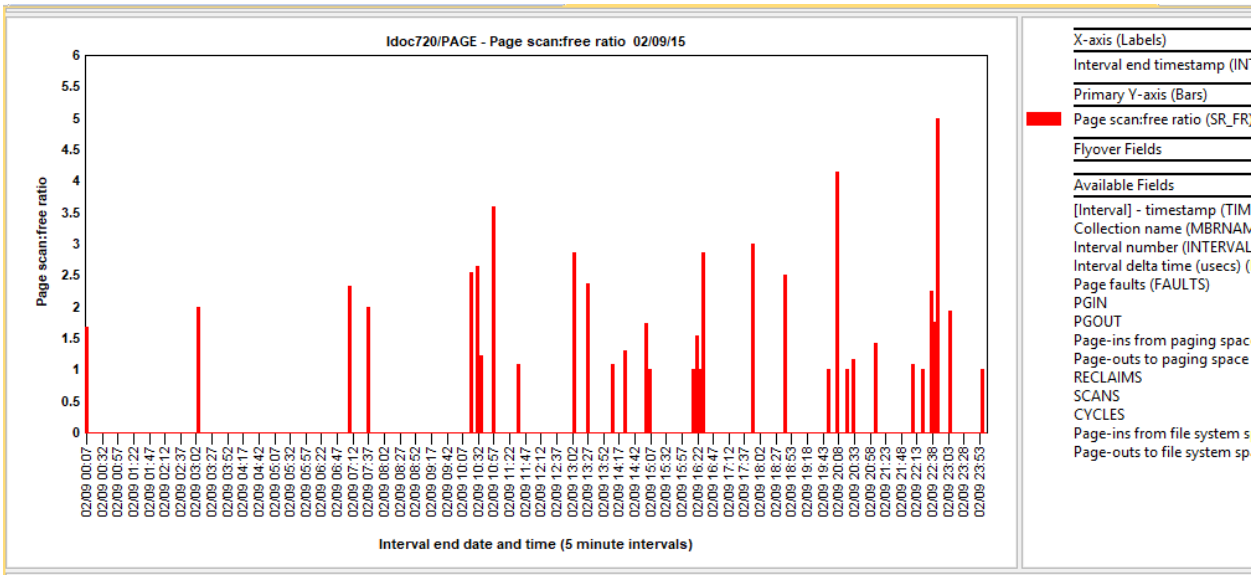
PAGE – Paging (filesystem)

7.10.4.3.9 PAGE - Page scan:free ratio

This graph shows the ration of scans to reclaims over time.

Field	Description
scans	The number of pages/sec examined by the page replacement routine. This is the same as the sr value reported by vmstat. Page replacement is initiated when the

	number of free pages falls below minfree and stops when the number of free pages exceeds maxfree.
reclaims (free)	This field is the same as the fr value reported by vmstat and represents the number of pages/sec freed by the page-replacement routine.



PAGE – Page scan:free ratio

7.10.4.3.10 PCPU - 1 overview

7.10.4.3.11 PCPU - 2 overview

7.10.4.3.12 PCPU - 3 overview

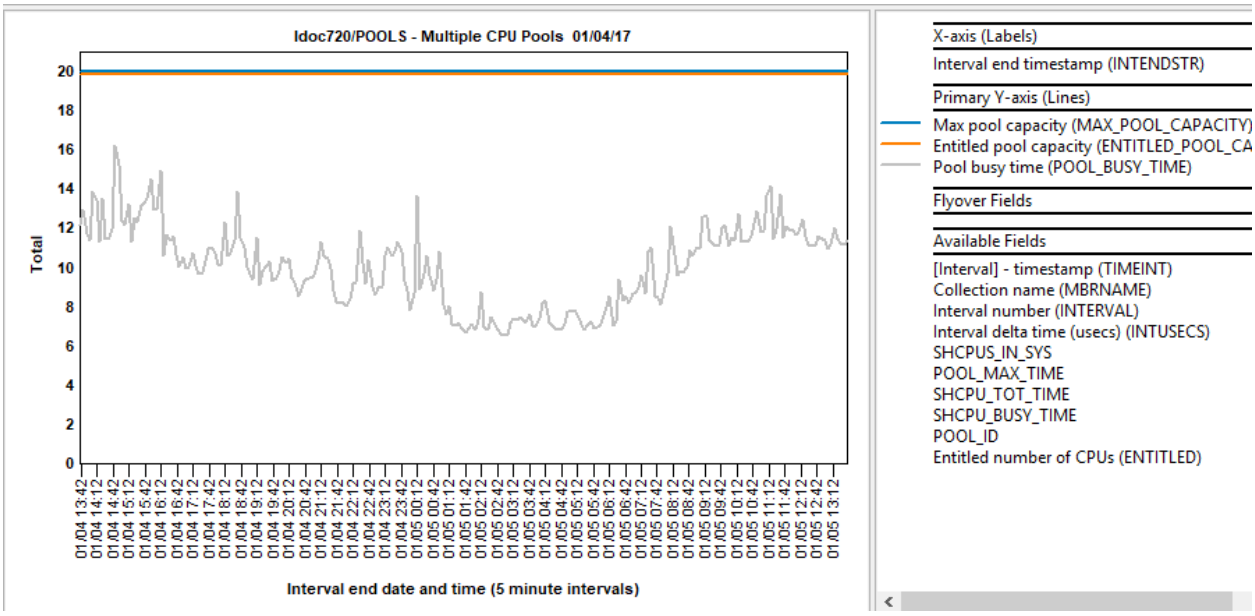
7.10.4.3.13 PCPU - 4 overview

7.10.4.3.14 PCPU_ALL - Total overview

7.10.4.3.15 POOLS - Multiple CPU Pools

This graph contains information about the shared pool in which the LPAR is running. Most of the data will only be present if “Allow performance information collection.” is set in the LPAR properties.

Field	Description
Max pool capacity	The maximum number of VPs defined for this pool
Entitled pool capacity	The entitlement for this pool (includes reserve entitlement)
Pool time	The average number of cores in use by this shared pool during the interval



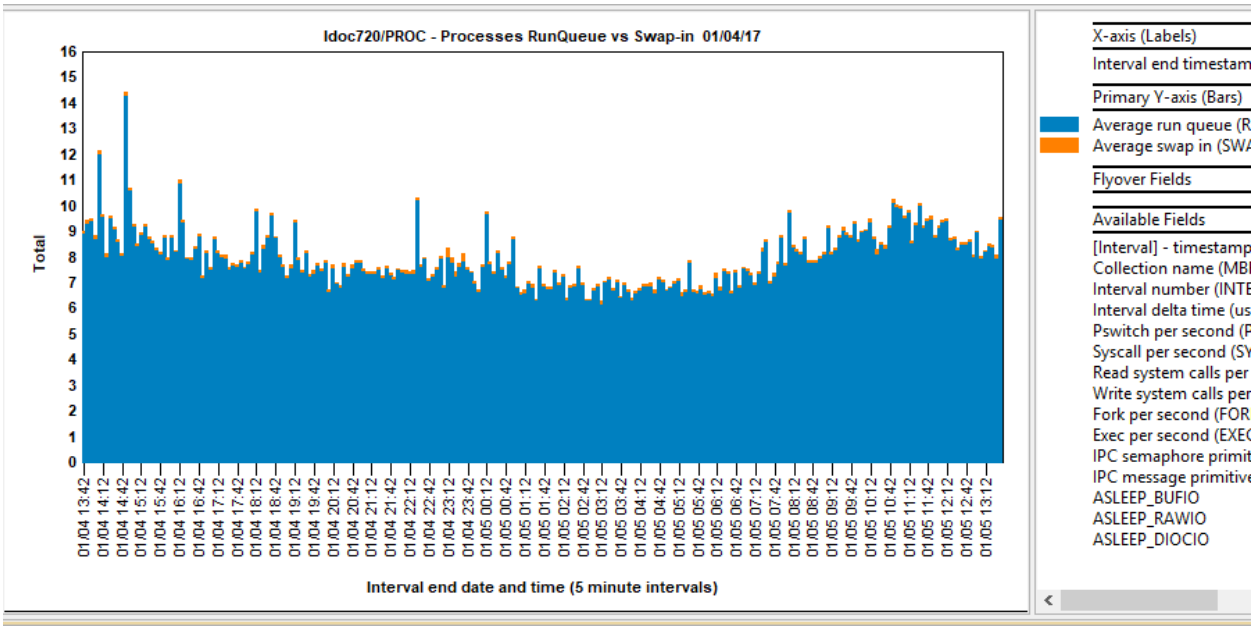
POOLS – Multiple CPU Pools

7.10.4.3.16 PROC - Processes RunQueue vs Swap-in

This sheet contains a subset of the fields reported by NMON on the Kernel Internal Statistics panel. The **RunQueue** and **Swap-in** fields are average values for the interval. All other fields are rates/sec:

RunQueue the average number of kernel threads in the run queue. This is reported as **runq-sz** by the sar -q command and is reported as **RunQueue** on the nmon Kernel Internal Statistics panel. A value that exceeds 3x the number of CPUs may indicate CPU constraint.

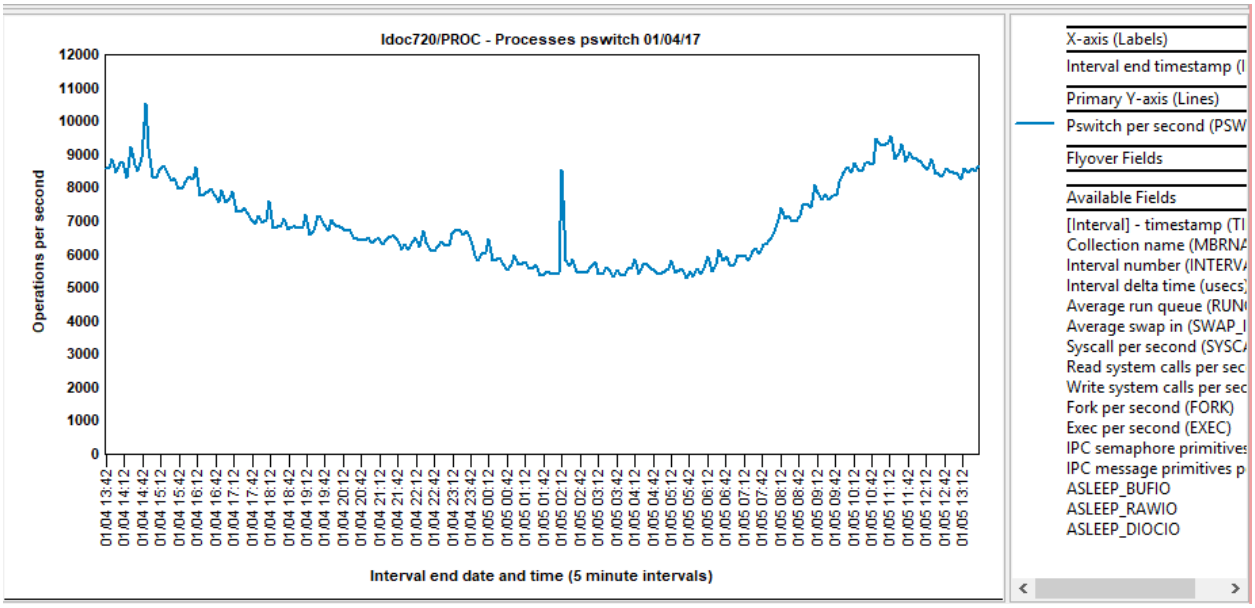
Swap-in the average number of kernel threads waiting to be paged in. This is reported as **swpq-sz** by the sar -q command.



PROC - Processes RunQueue vs Swap-in

7.10.4.3.17 PROC - Processes pswitch

This sheet contains a subset of the fields reported by NMON on the Kernel Internal Statistics panel. This graph reports the number of context switches. This is reported as **pswch/s** by the sar -w command.

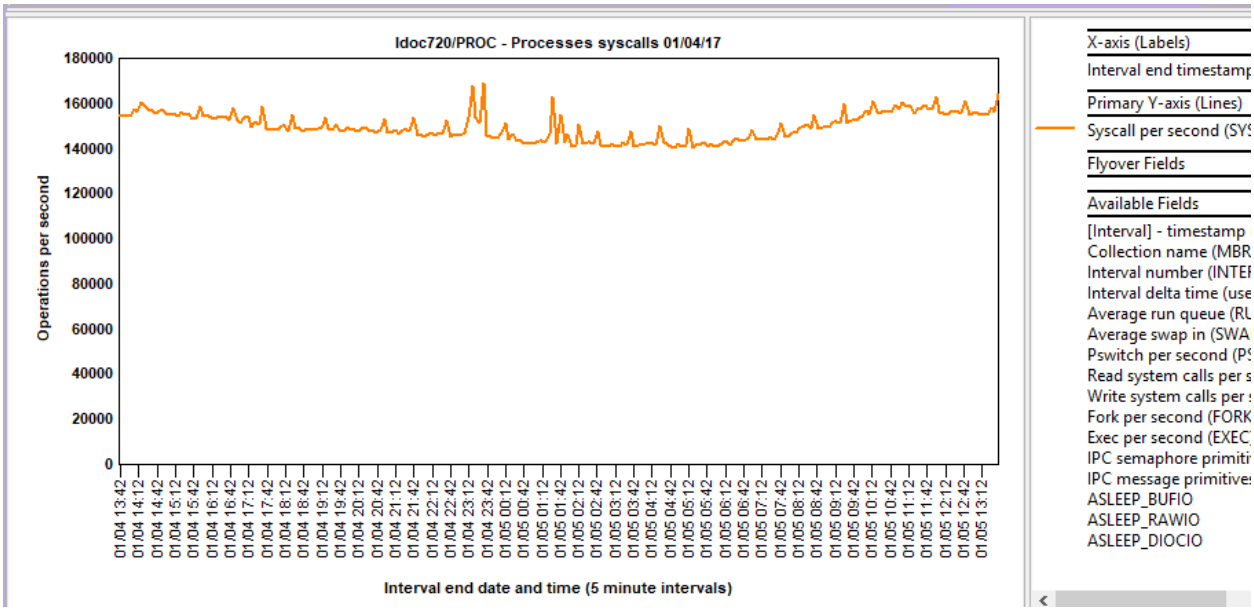


PROC - Processes pswitch

7.10.4.3.18 PROC - Processes syscalls

This sheet contains a subset of the fields reported by NMON on the Kernel Internal Statistics panel.

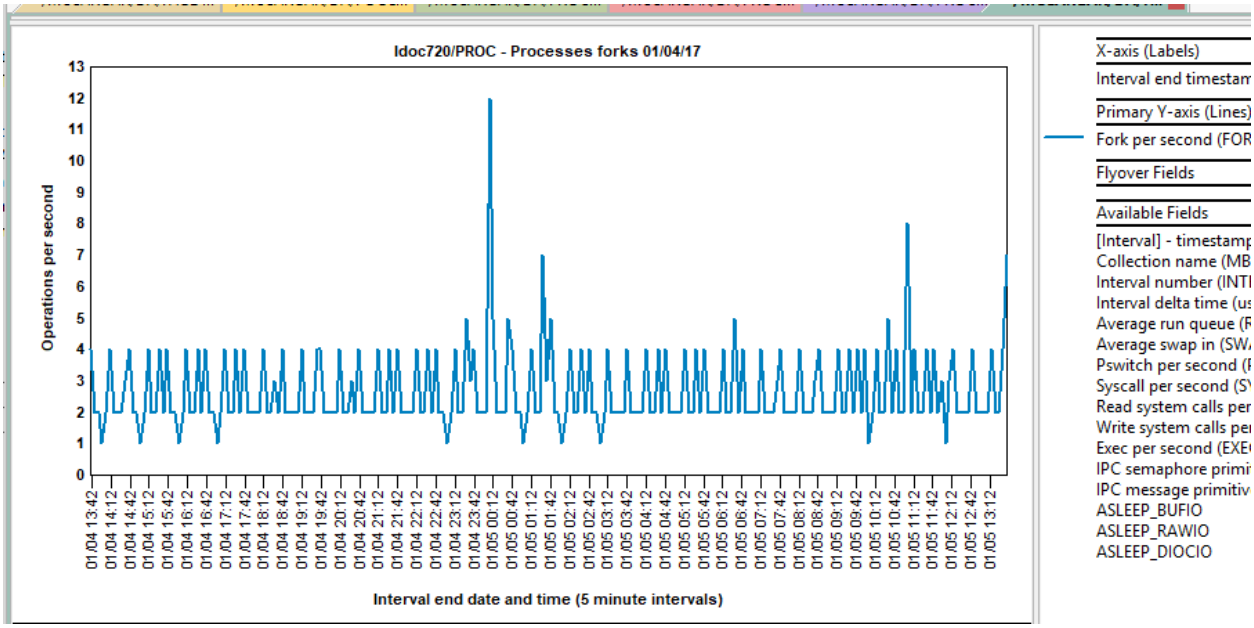
This graph reports the total number of system calls. This is reported as **scall/s** by the sar -c command.



PROC - Processes syscalls

7.10.4.3.19 PROC - Processes forks

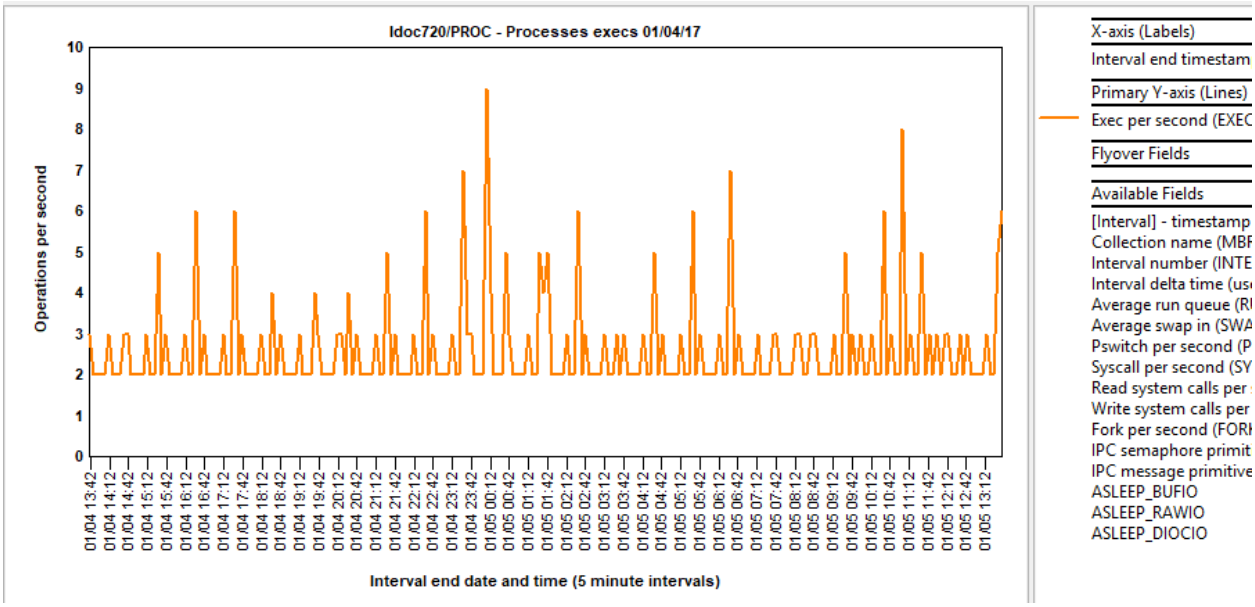
This graph reports the number of fork system calls. This is reported as **fork/s** by the sar -c command.



PROC - Processes forks

7.10.4.3.20 PROC - Processes execs

This graph reports the number of exec system calls per second. This is reported as **exec/s** by the sar -c command.

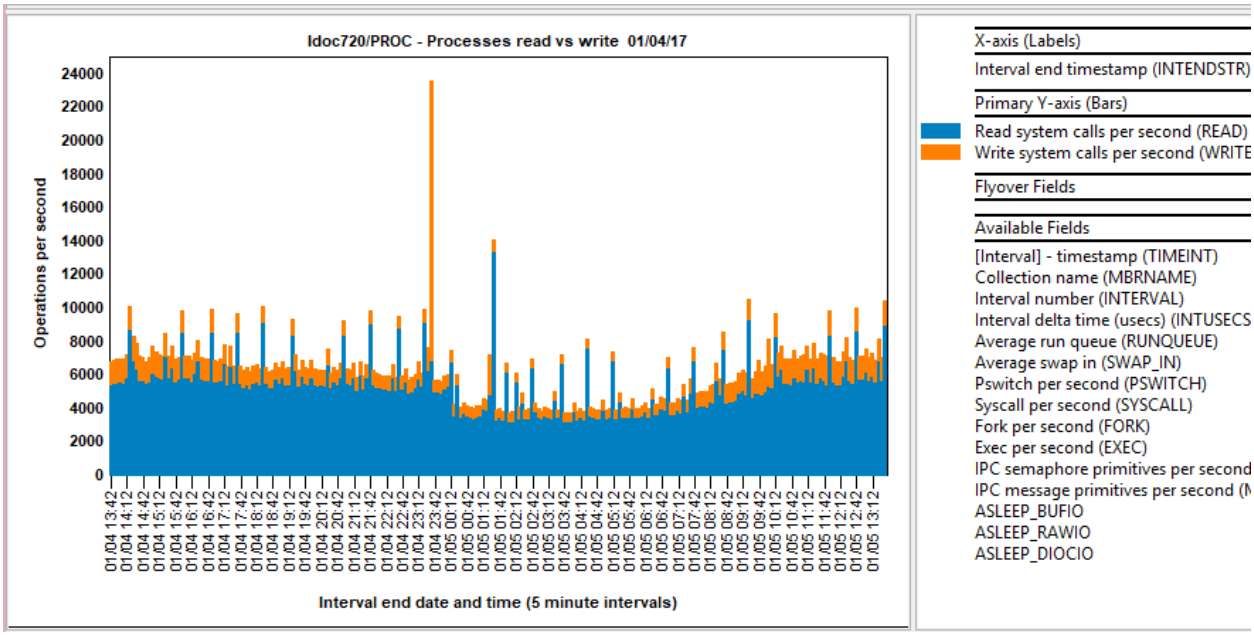


PROC - Processes execs

7.10.4.3.21 PROC - Processes read vs write

This graph compares read vs write system calls per second over time.

Field	Description
read	The number of read system calls. This is reported as **sread/s** reported by the sar -c command.
write	The number of write system calls. This is reported as **swrit/s** by the sar -c command.

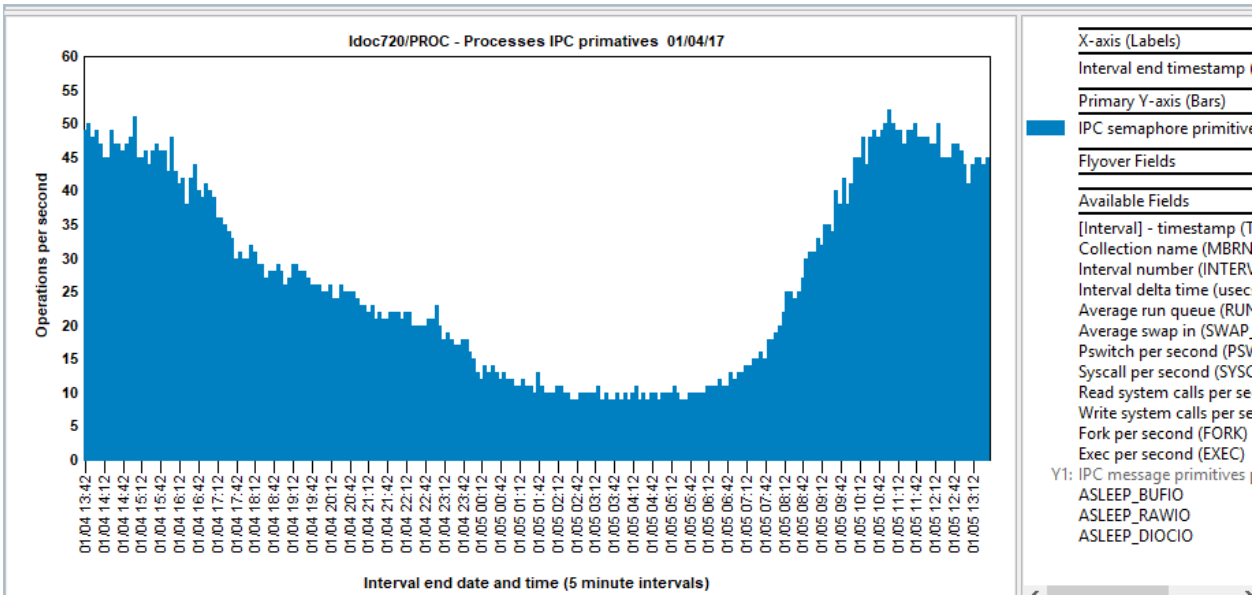


PROC - Processes read vs write

7.10.4.3.22 PROC - Processes IPC primitives

This graph compares IPC semaphore primitives vs IPC message primitives per second over time.

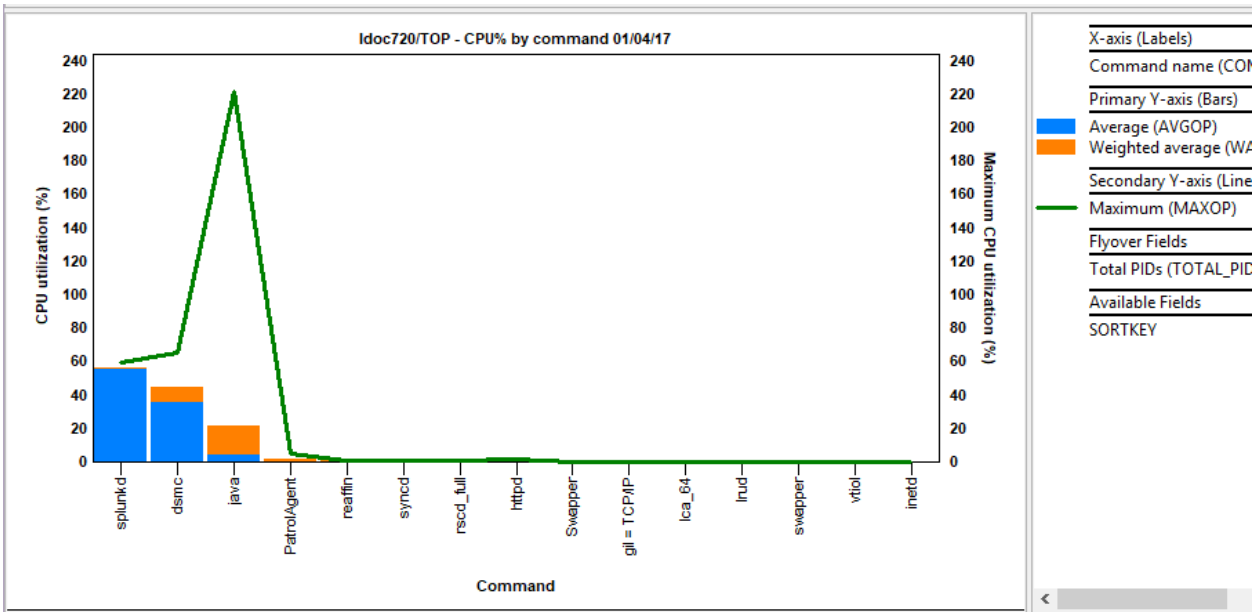
Field	Description
Sem	The number of IPC semaphore primitives (creating, using and destroying). This is reported as **sema/s** by the sar -m command.
msg	The number of IPC message primitives (sending and receiving). This is reported as **msg/s** by the sar -m command.



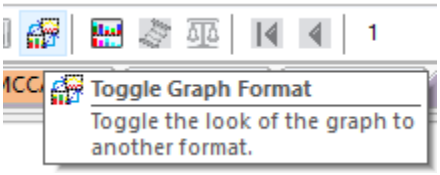
PROC – Processes IPC primatives

7.10.4.3.23 TOP - CPU% by command

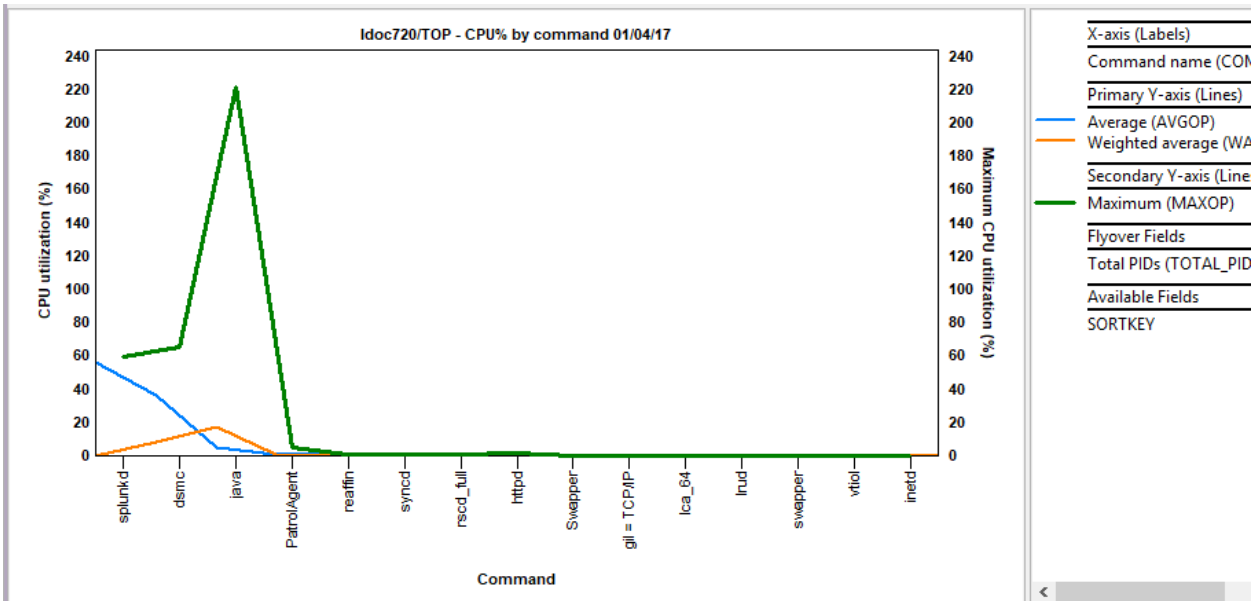
Note: data only present for processes that consumed a significant amount of CPU during an interval. The TOP sheets does **NOT** represent a complete view of the system.



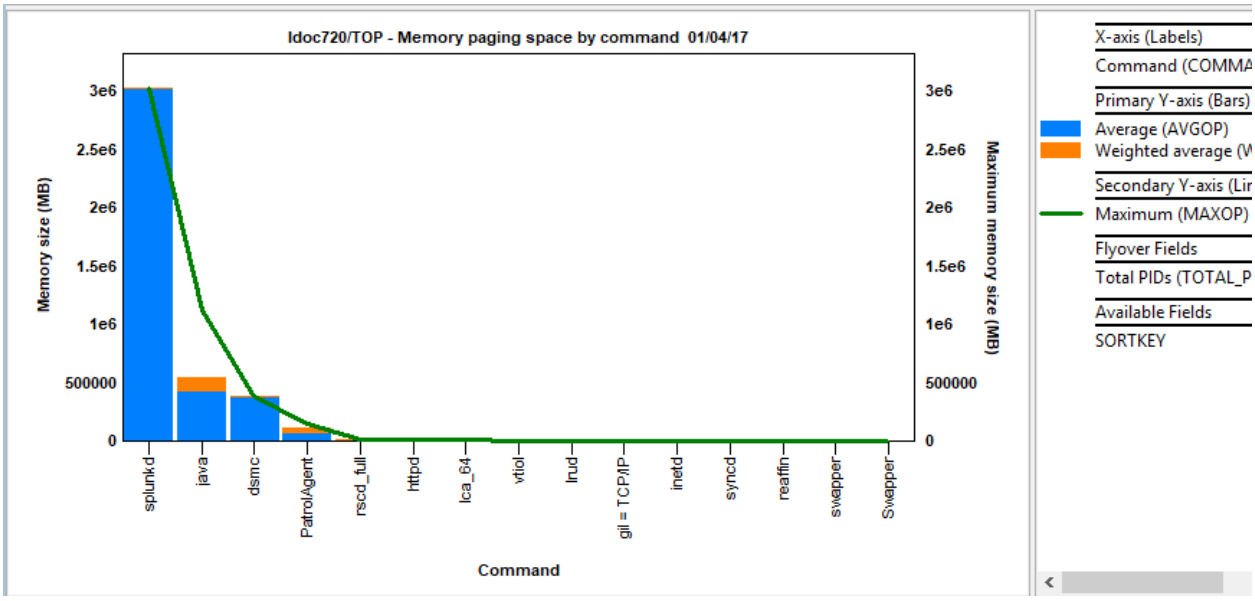
TOP – CPU % by command



Tip: Press the toggle graph format button to convert this graph to bars to lines if desired.

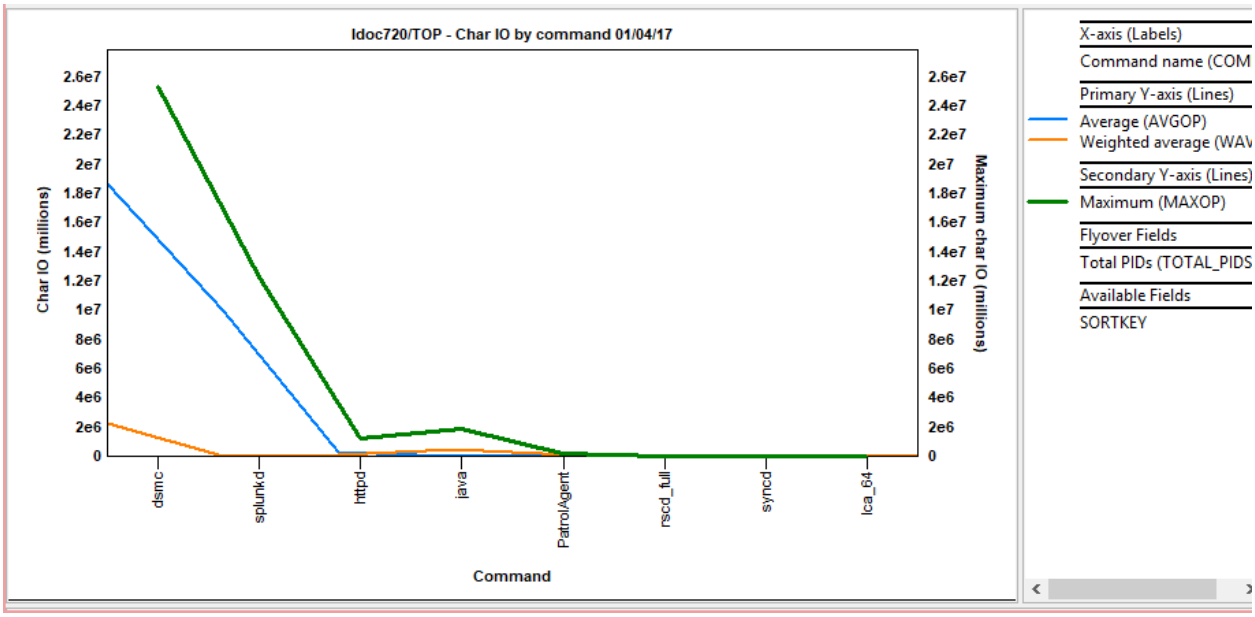


7.10.4.3.24 TOP - Memory paging space by command



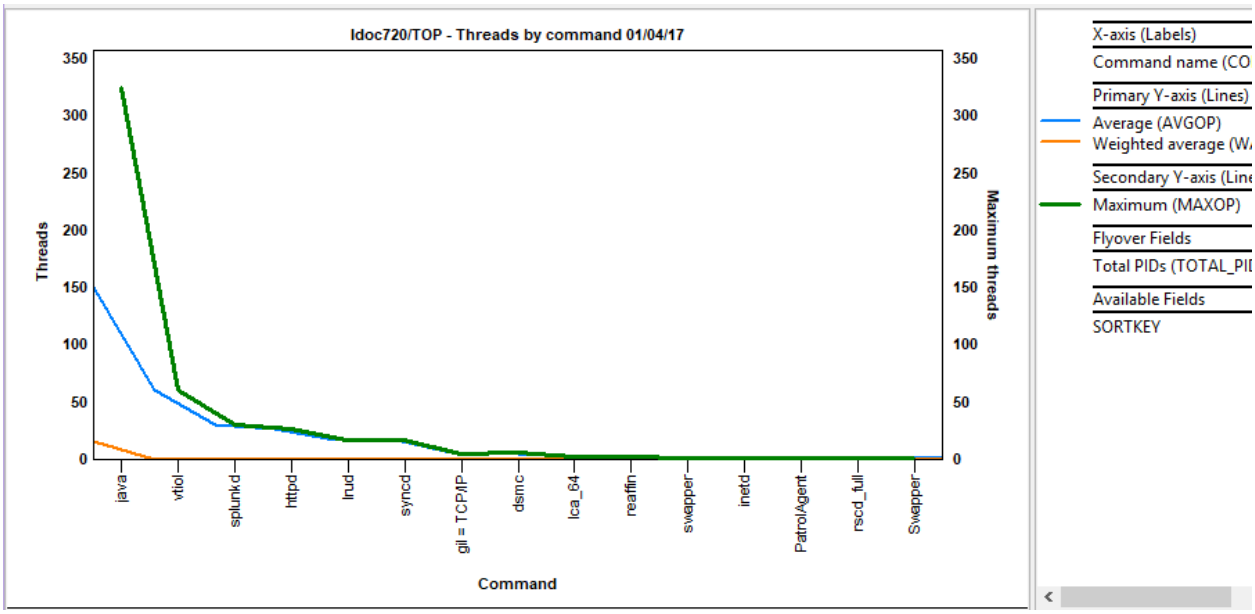
TOP – Memory paging space by command

7.10.4.3.25 TOP - Char IO by command



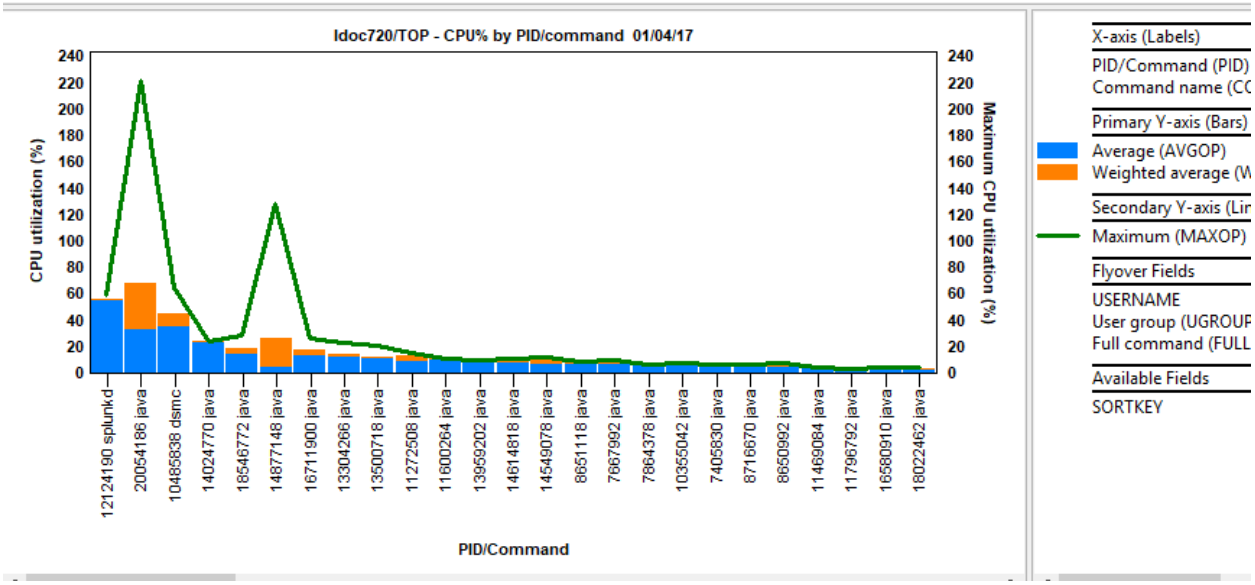
TOP – Char IO by command

7.10.4.3.26 TOP - Threads by command



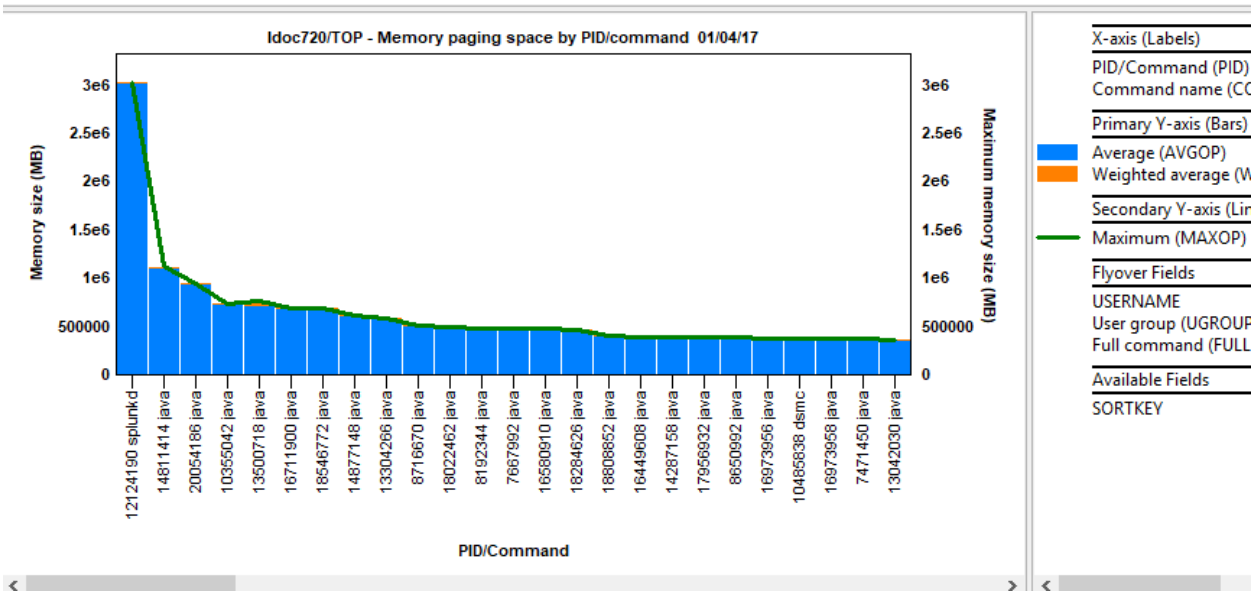
TOP – Threads by command

7.10.4.3.27 TOP - CPU% by PID/command



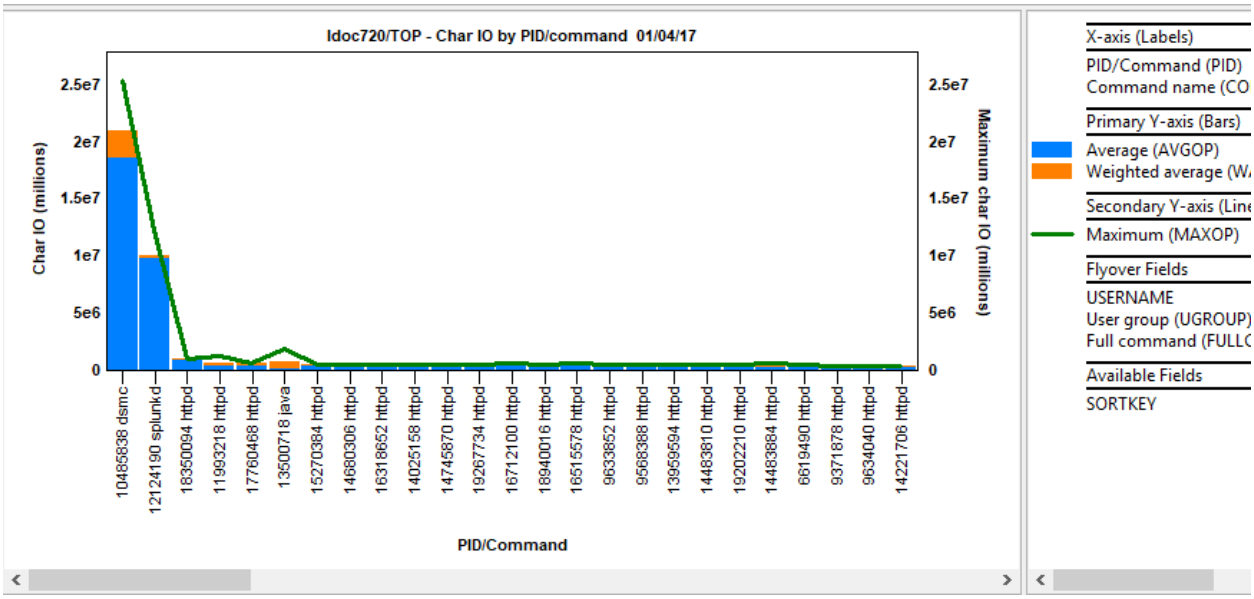
TOP – CPU % by PID/command

7.10.4.3.28 TOP - Memory paging space by PID/command



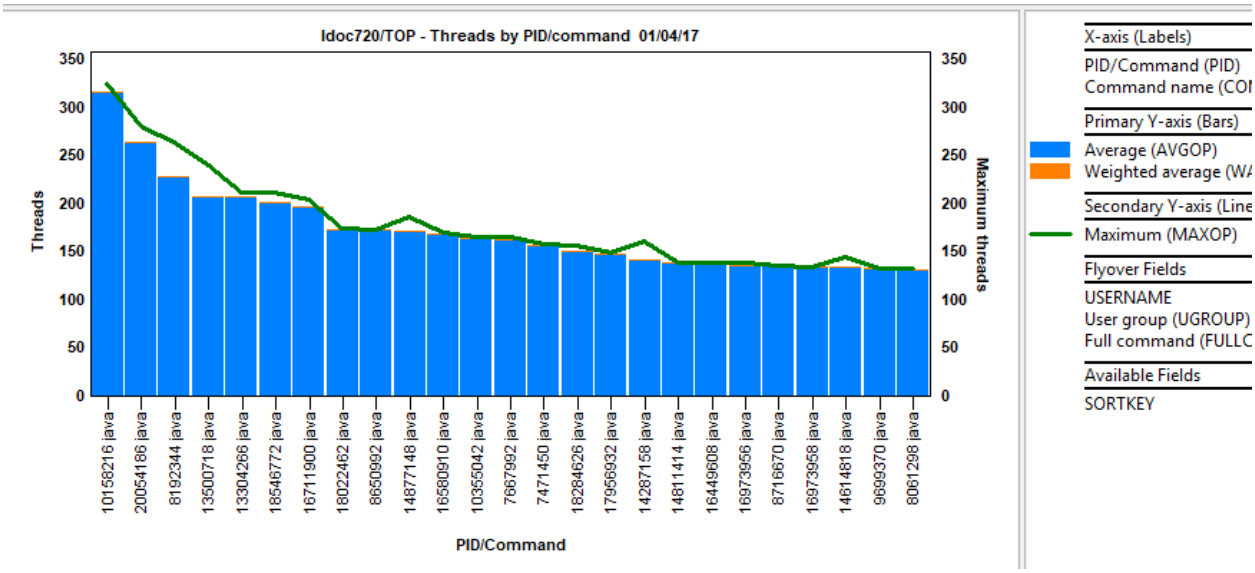
TOP – Memory paging space by PID/command

7.10.4.3.29 TOP - Char IO by PID/command



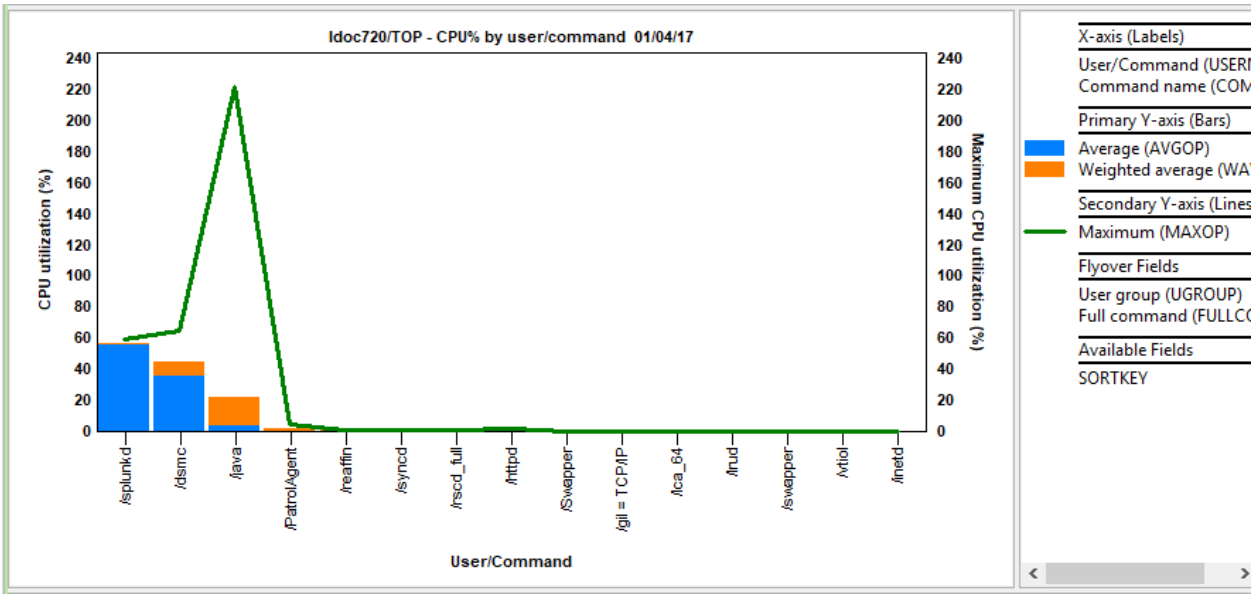
TOP – Char IO by PID/command

7.10.4.3.30 TOP - Threads by PID/command



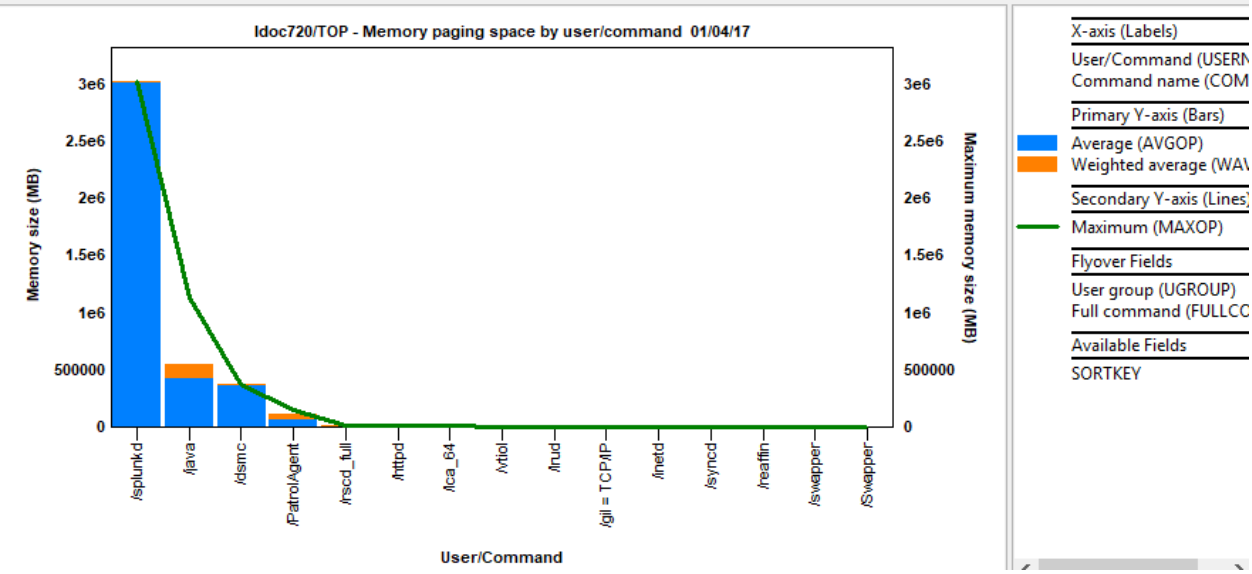
TOP – Threads by PID/command

7.10.4.3.31 TOP - CPU% by user/command



TOP – CPU % by user/command

7.10.4.3.32 TOP - Memory paging space by user/command



TOP – Memory paging space by user/command

7.10.4.3.33 TOP – Char IO by user/command

7.10.4.3.34 TOP - Threads by user/command

7.10.4.3.35 TOP - CPU% by user

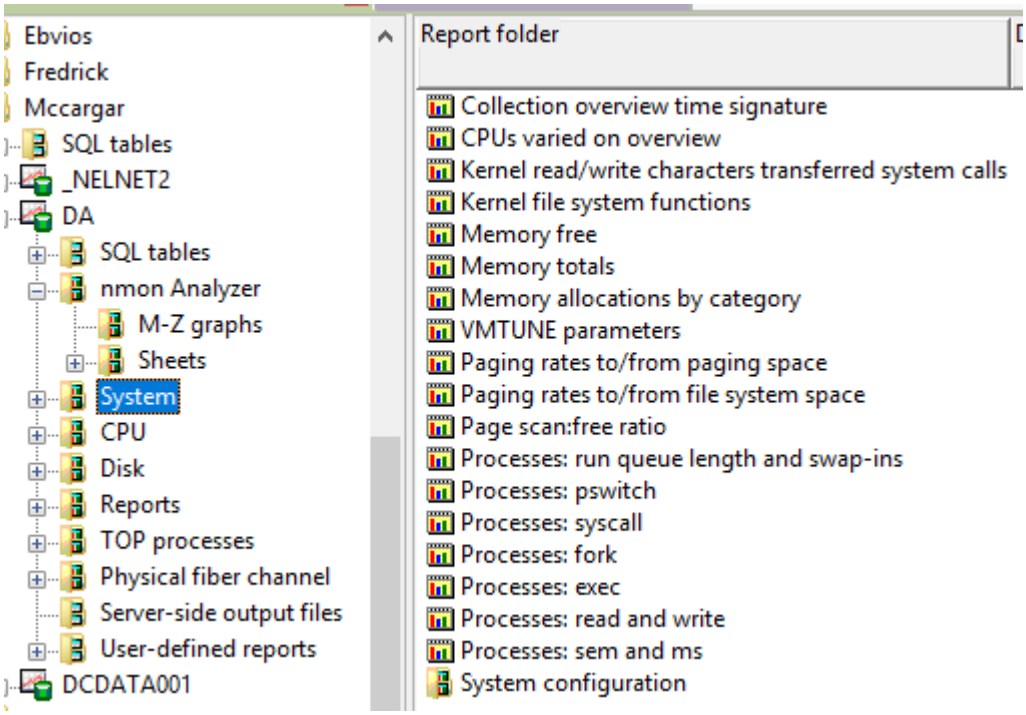
7.10.4.3.36 TOP - Memory paging space by user

7.10.4.3.37 TOP - Char IO by user

7.10.4.3.38 TOP - Threads by user

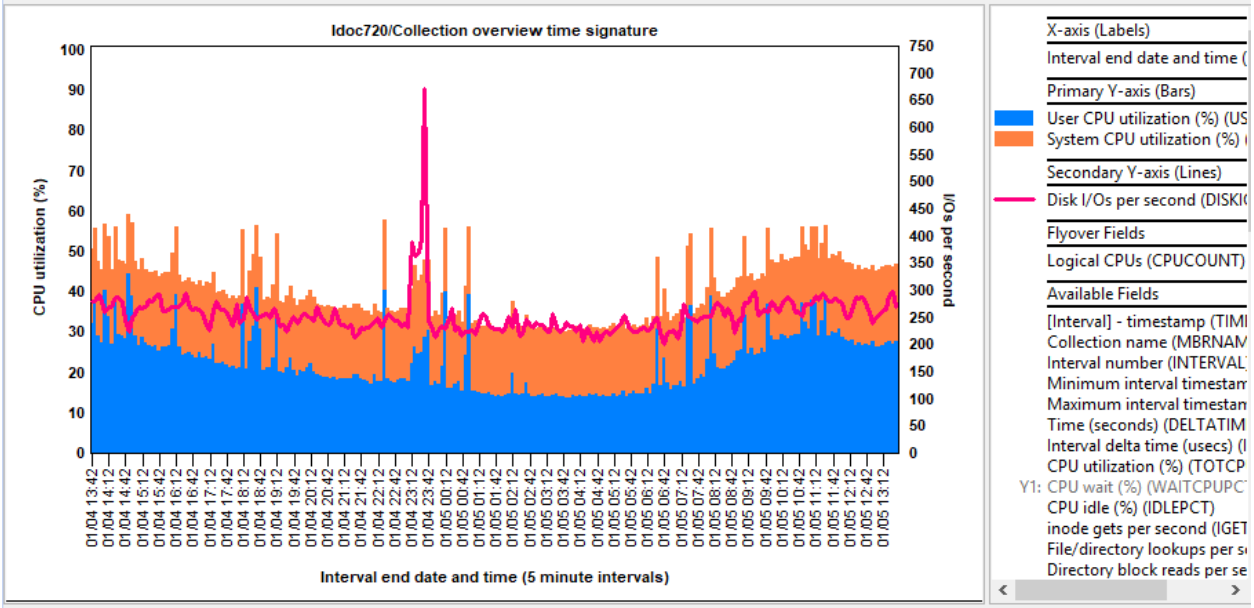
7.10.4.4 System

These graphs display several different types of statistics for the AIX/VIOS or Linux system.



Systems folder

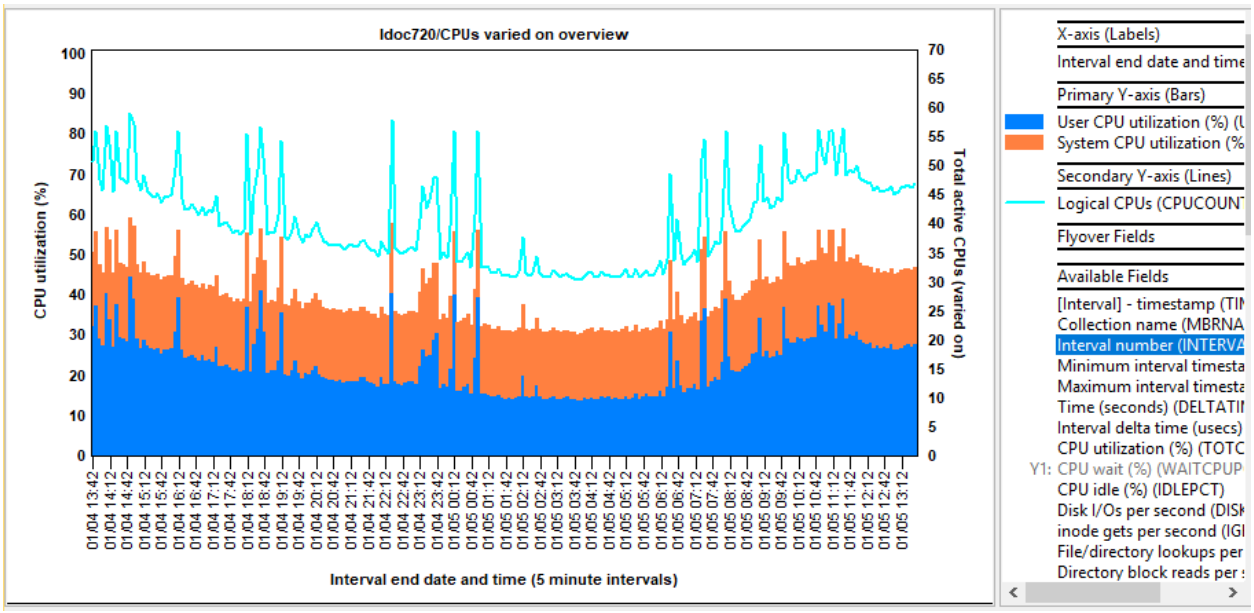
7.10.4.4.1 Collection overview time signature



Collection overview time signature

This graph shows CPU utilization and CPU wait percentage along with disk I/Os per second as the secondary Y-axis (Y2).

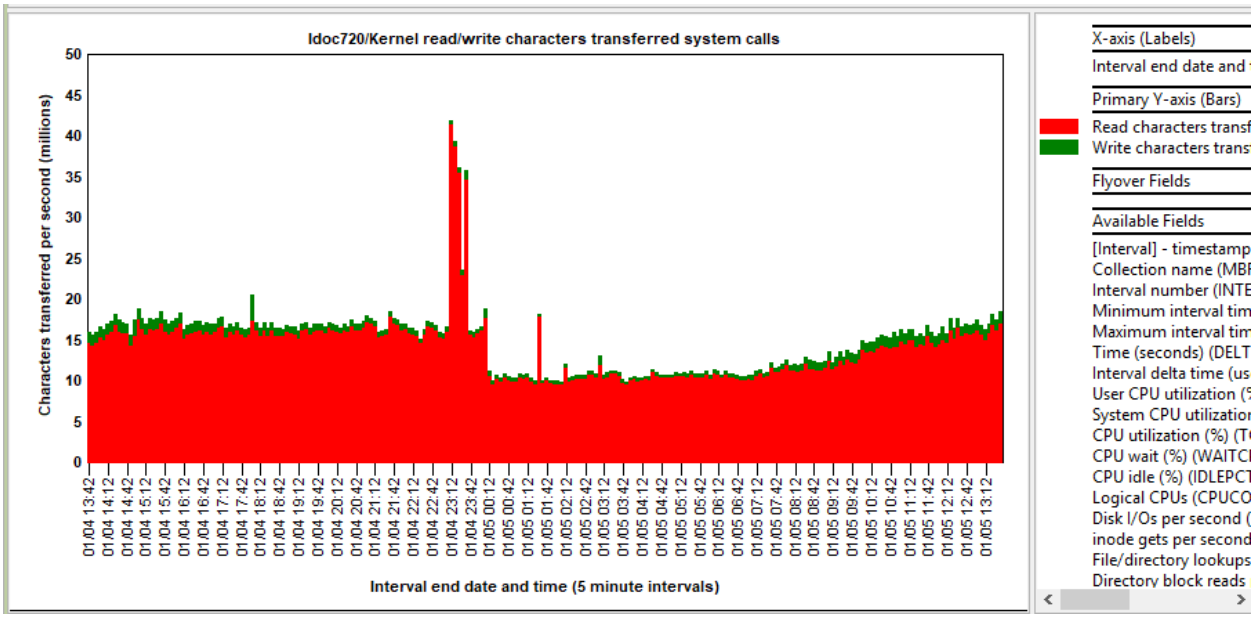
7.10.4.4.2 CPUs varied on overview



CPUs varied on overview

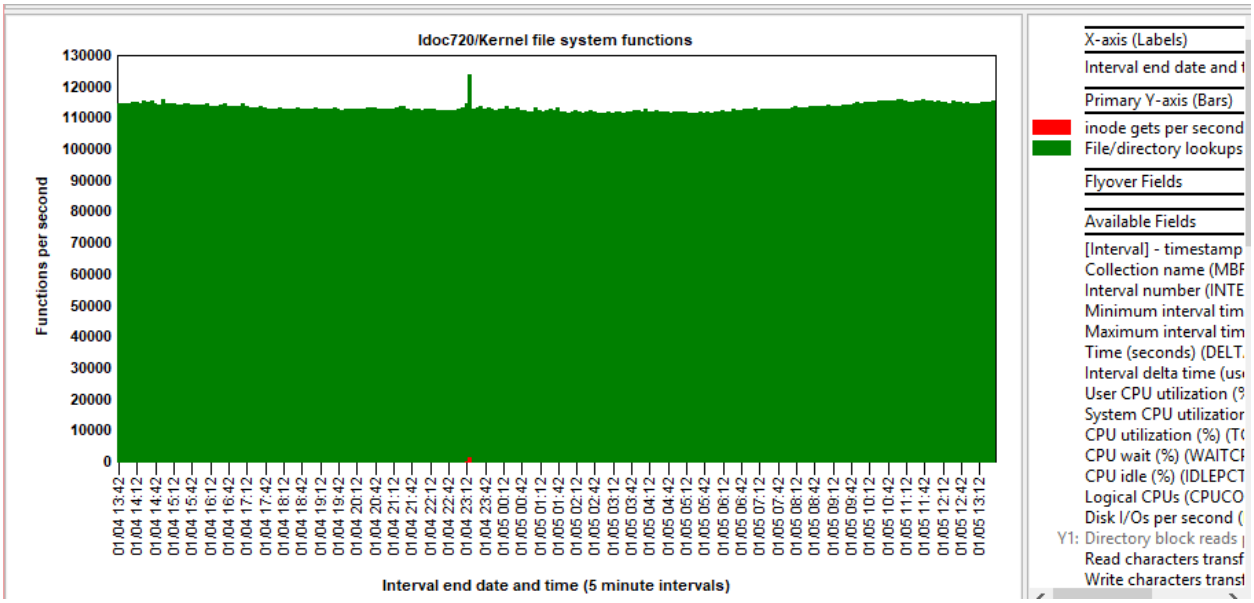
This graph shows CPU utilization and CPU wait percentage along with the number of CPUs varied on over time on the secondary Y-axis (Y2).

7.10.4.4.3 Kernel read/write characters transferred system calls



Kernel read/write characters transferred system calls

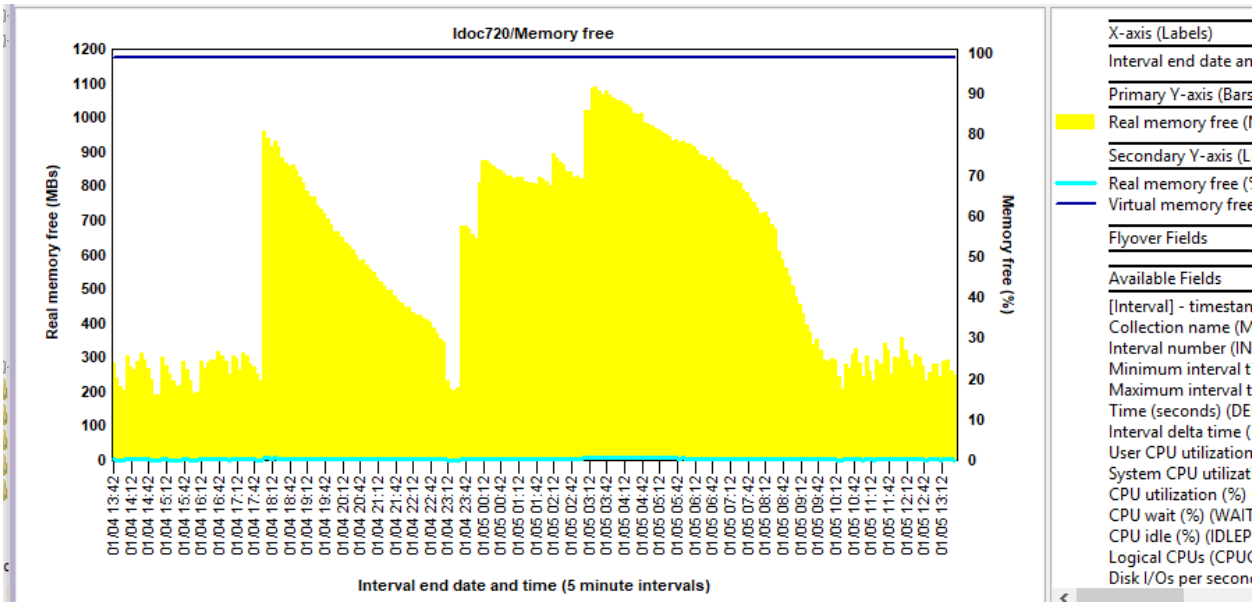
7.10.4.4.4 Kernel file system functions



Kernel file system functions

This graph displays the number of function calls per second over time for the iget, namei and dirblk functions.

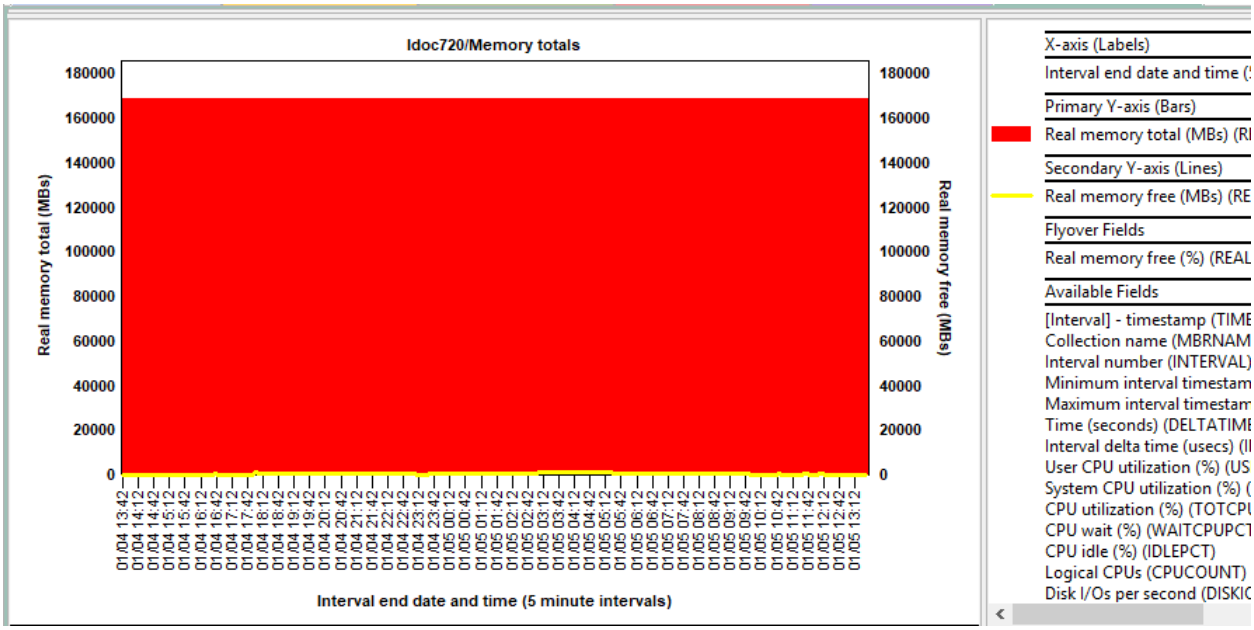
7.10.4.4.5 Memory free



Memory free

This graph shows the real memory free in megabytes and also as a percentage on the second Y-axis along with virtual memory free.

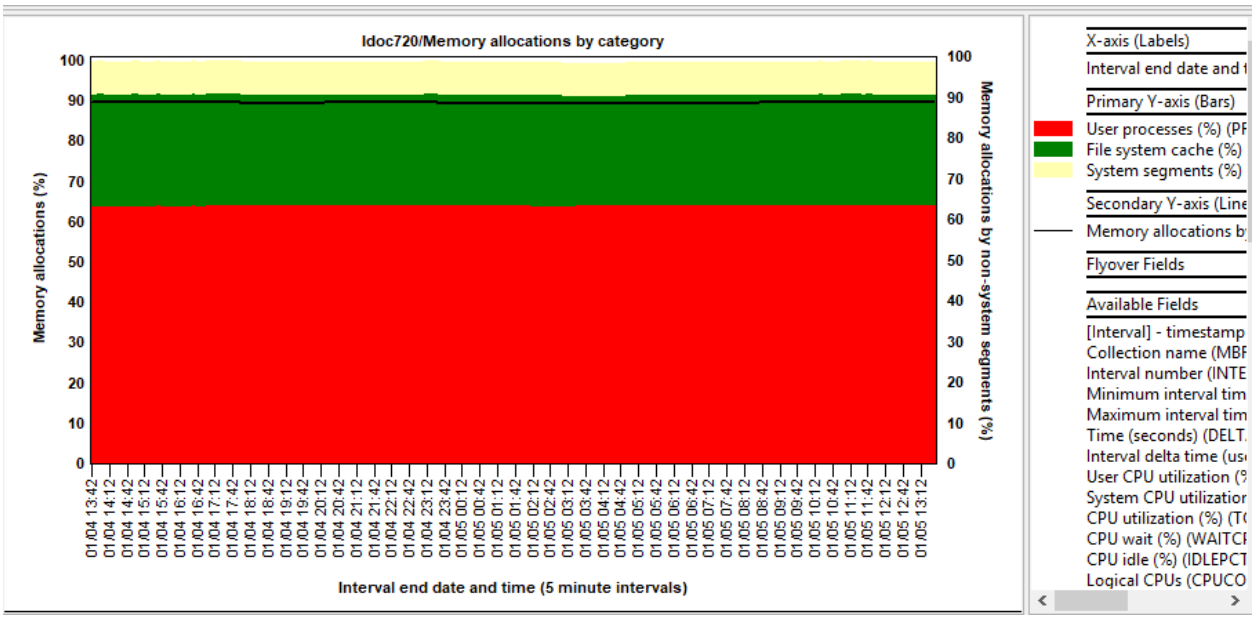
7.10.4.4.6 Memory totals



Memory totals

This graph displays total real memory along with the real memory free on the secondary Y-axis.

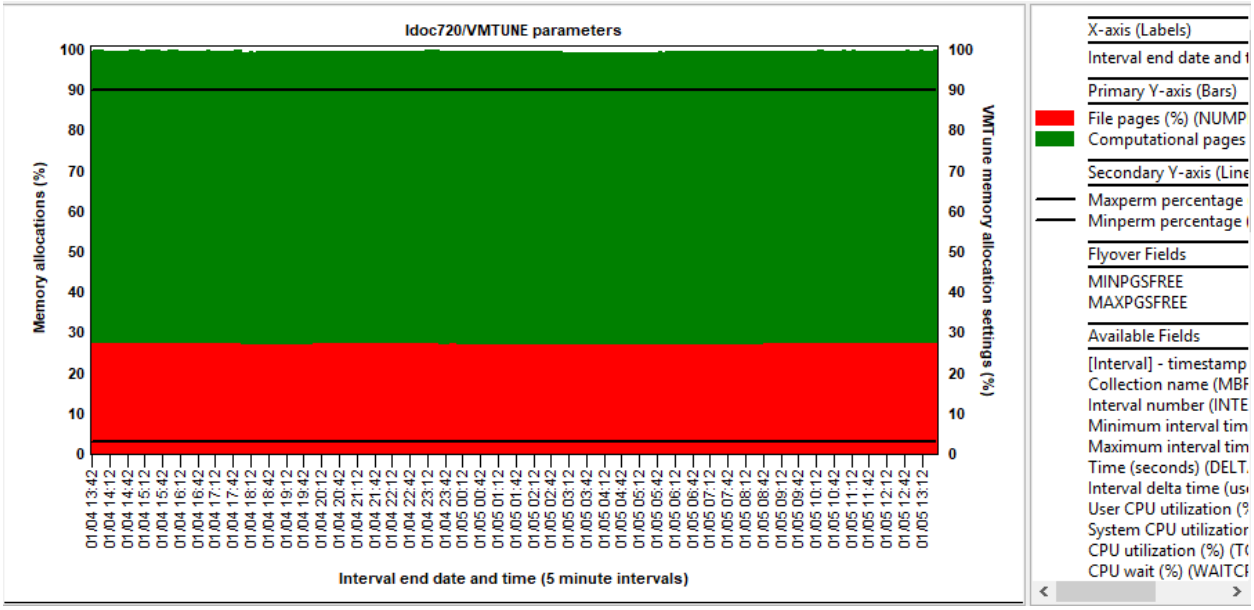
7.10.4.4.7 Memory allocations by category



Memory allocations by category

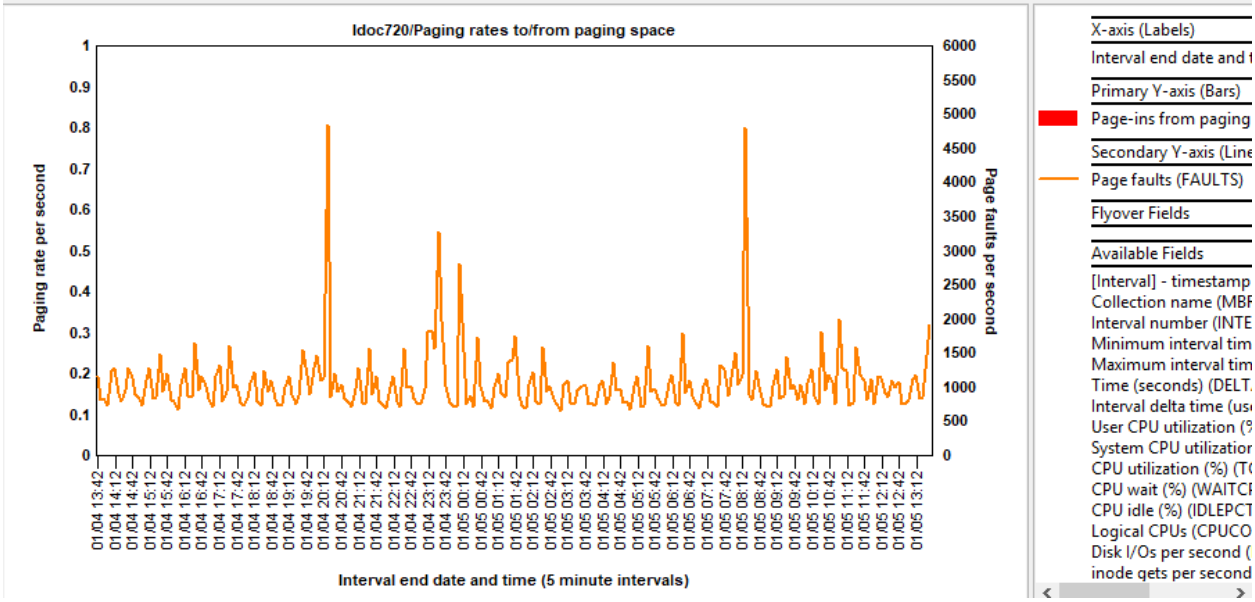
This graph gives the user an indication of the possible high-level categories that memory is being allocated to.

7.10.4.4.8 VMTUNE parameters



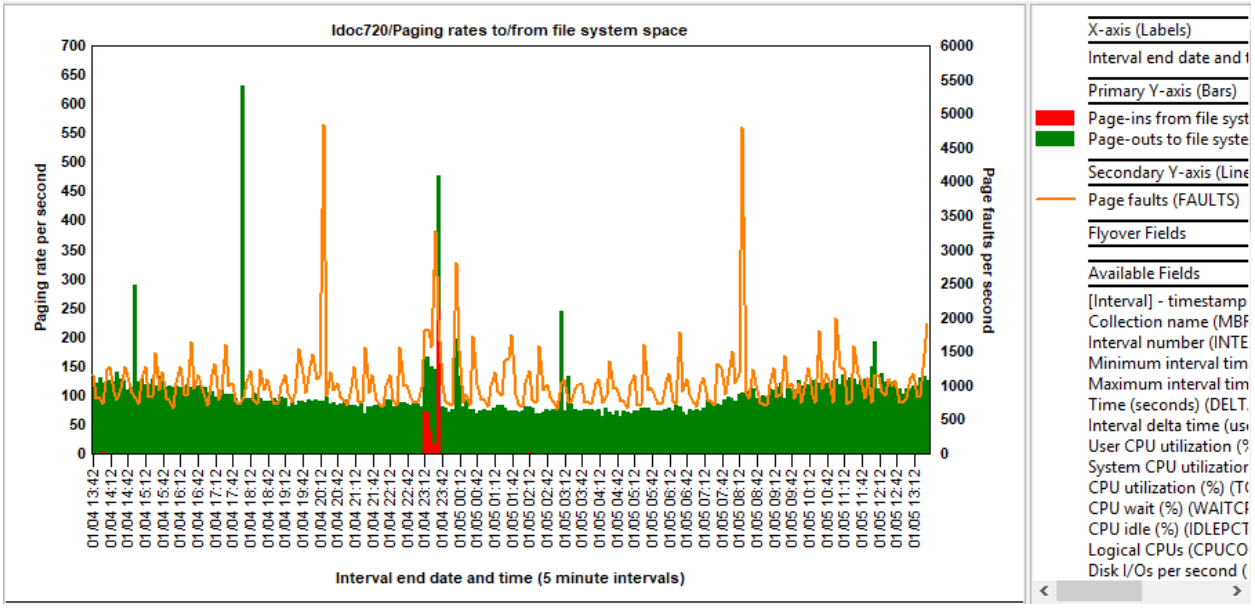
VMTUNE parameters

7.10.4.4.9 Paging rates to/from paging space



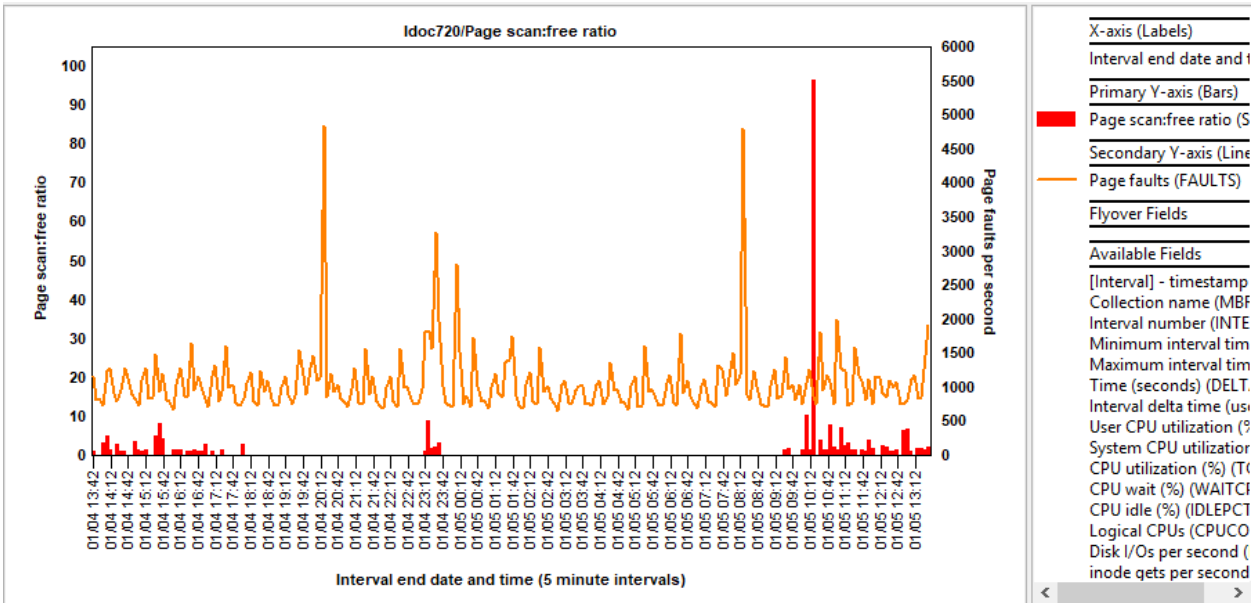
Paging rates to/from paging space

7.10.4.4.10 Paging rates to/from file system space



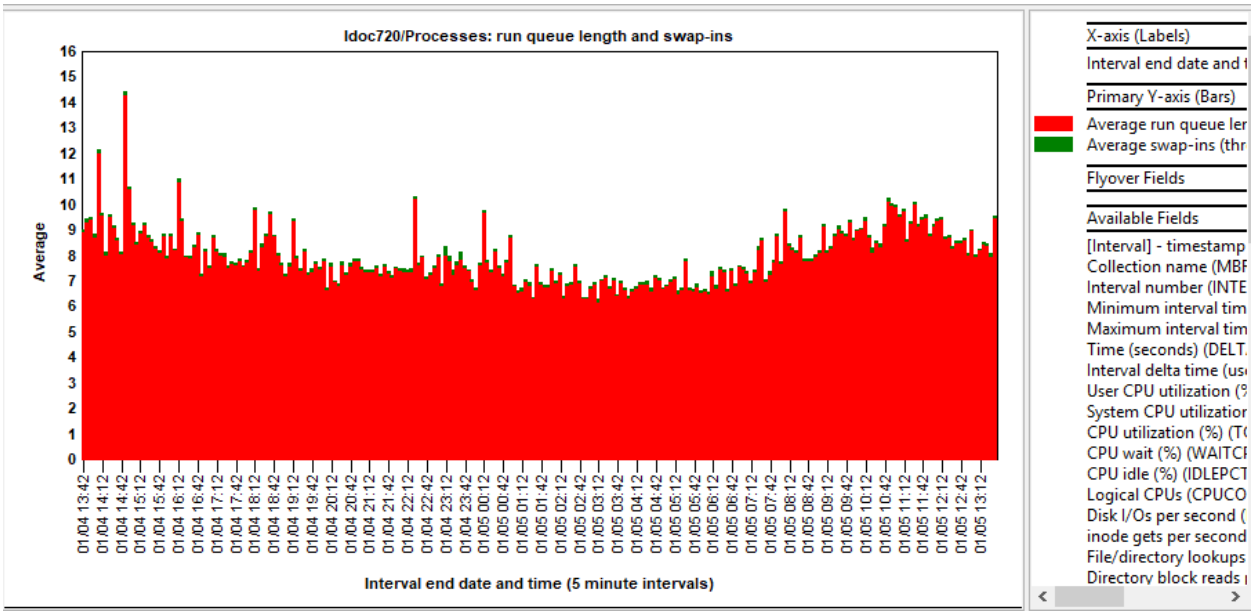
Paging rates to/from file system space

7.10.4.4.11 Page scan:free ratio



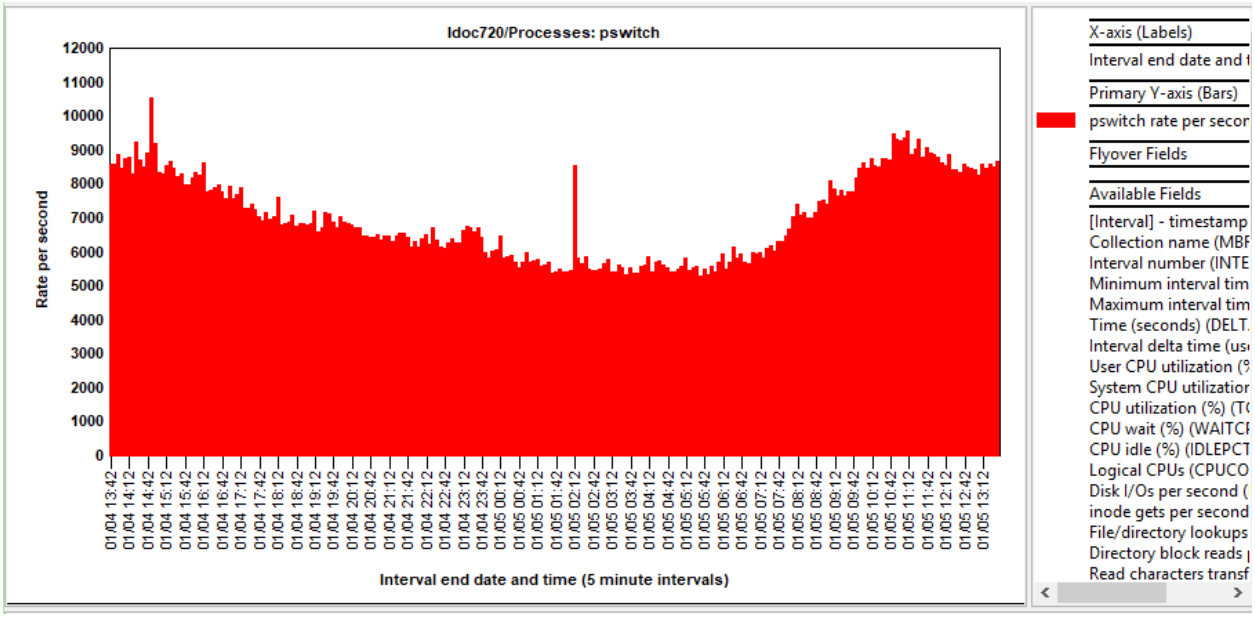
Page scan:free ratio

7.10.4.4.12 Processes: run queue length and swap-ins



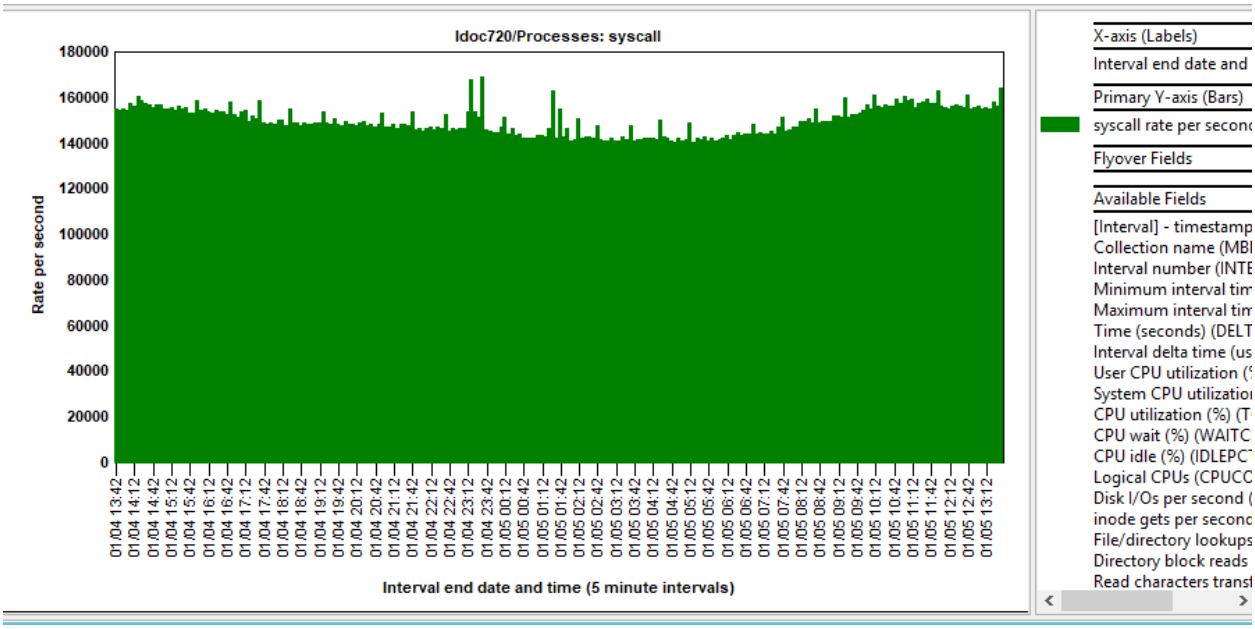
Processes: run queue length and swap-ins

7.10.4.4.13 Processes: pswitch

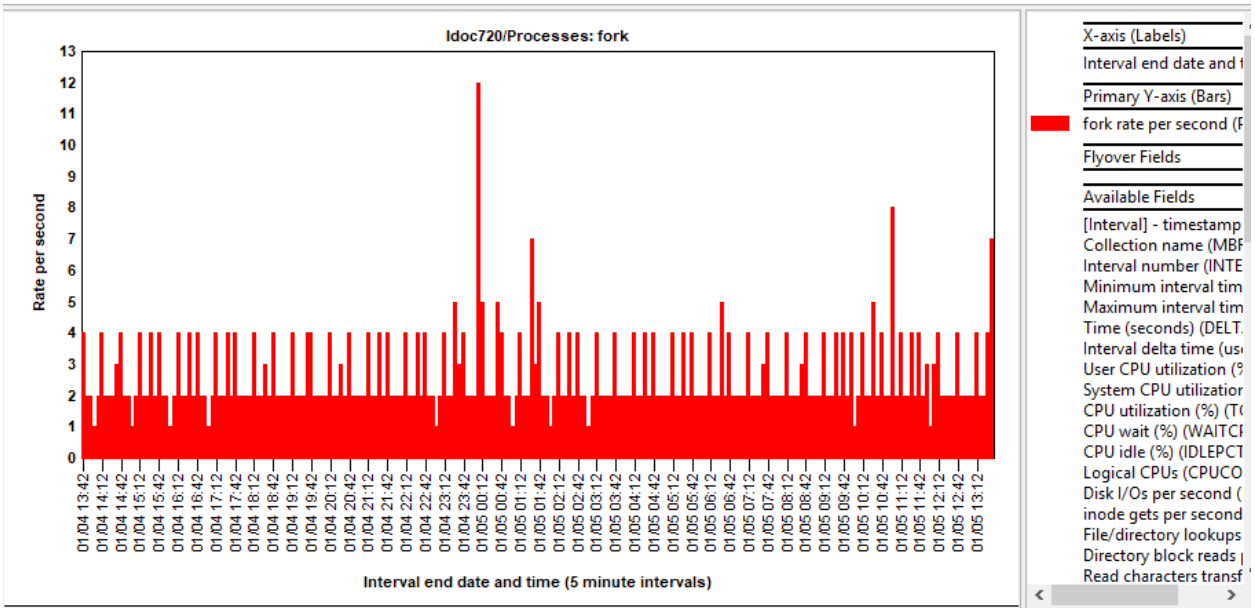


Processes: pswitch

7.10.4.4.14 Processes: syscall

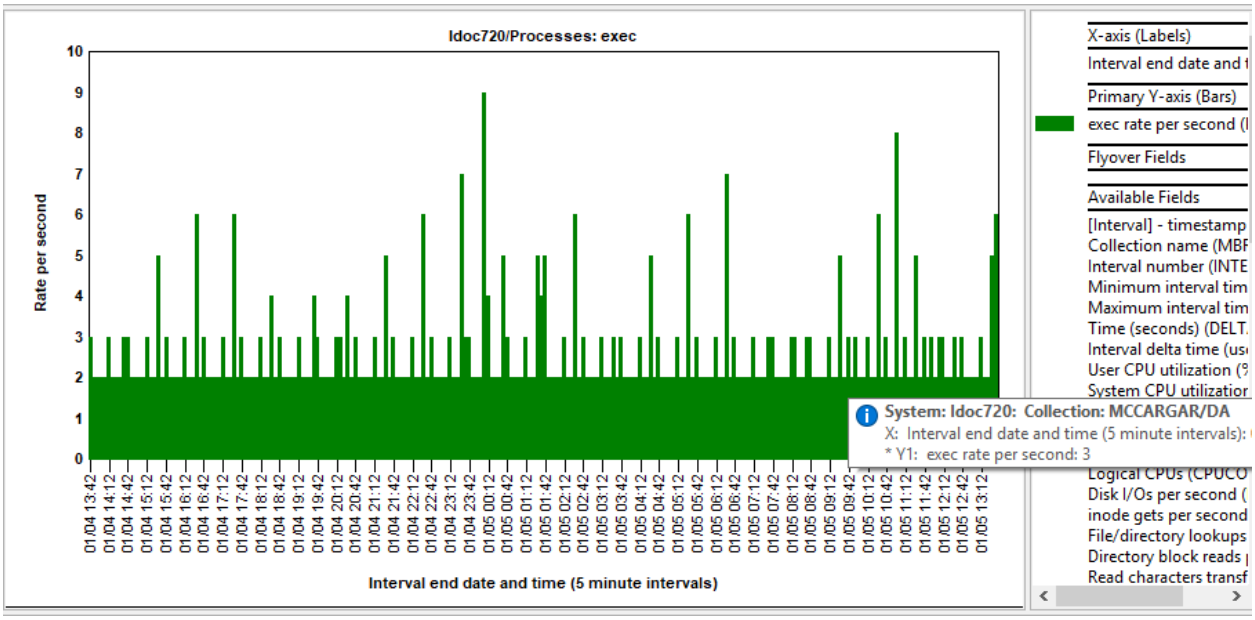


7.10.4.4.15 Processes: fork

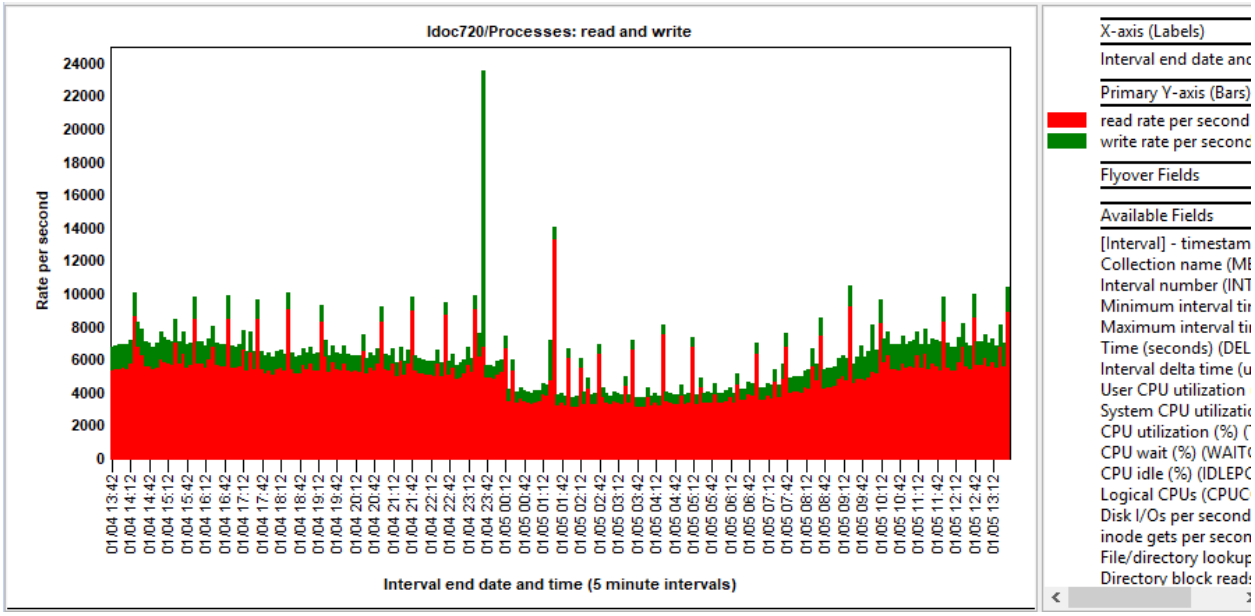


Processes: fork

7.10.4.4.16 Processes: exec

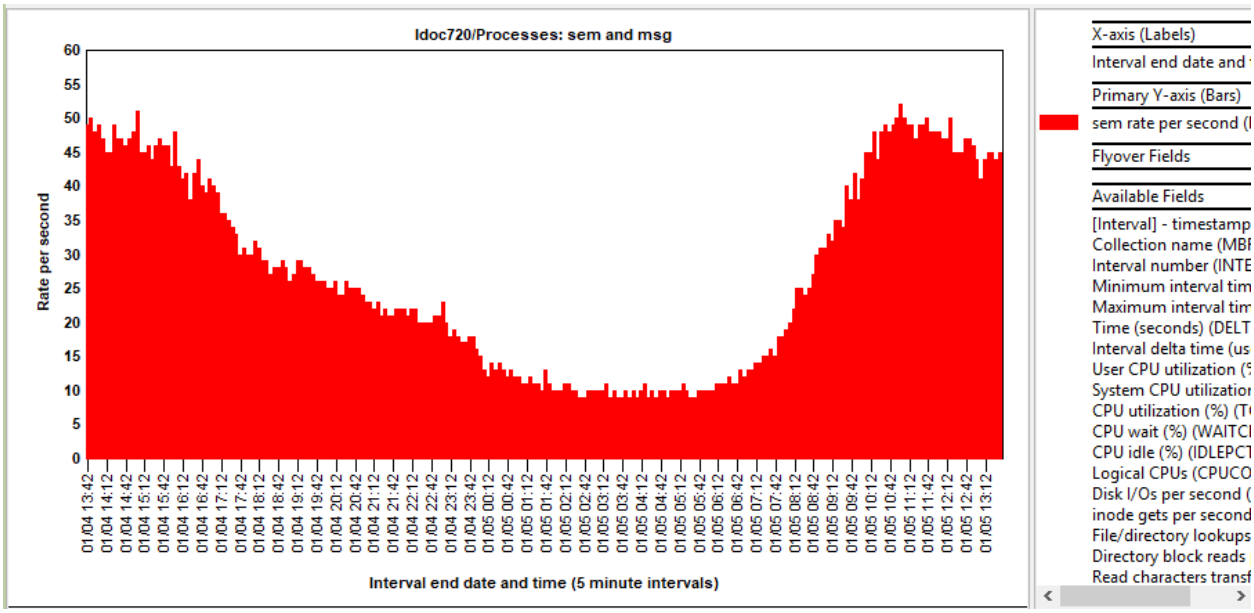


7.10.4.4.17 Processes: read and write



Processes: read and write

7.10.4.4.18 Processes: sem and msg

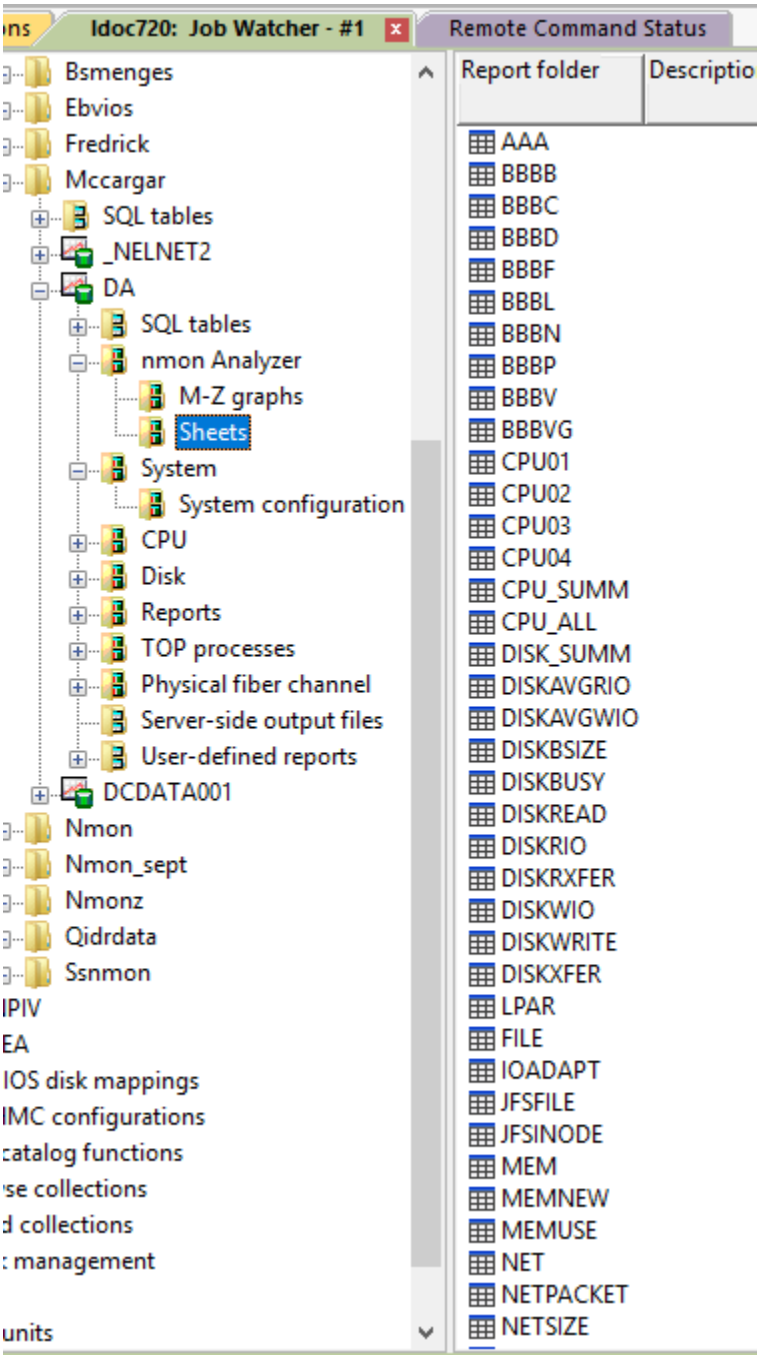


Processes: sem and msg

7.10.4.5 Sheets

This folder displays the nmon table data from nmon Analyzer as close as possible.

Note: Some reports will NOT appear depending on what data exists in the original nmon file. If the data tag does not exist, then the report will not exist.

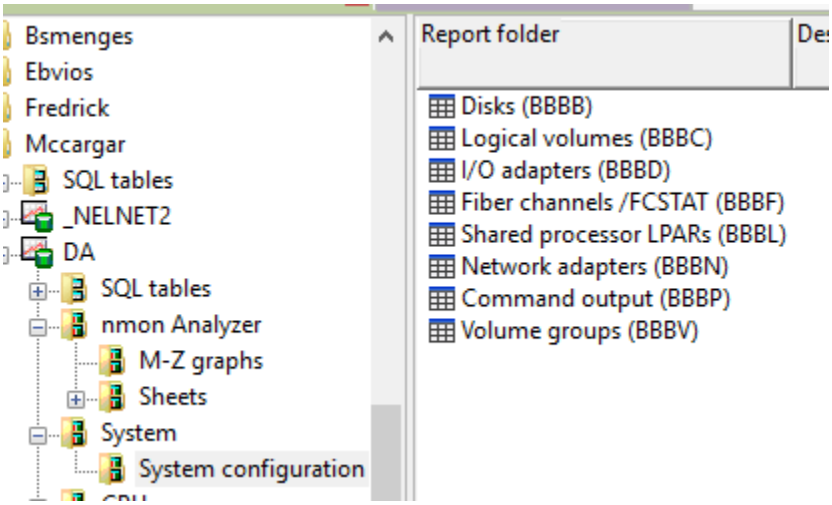


Nmon Analyzer -> Sheets folder

7.10.4.6 System configuration

These reports display information about the hardware configuration of the system as reported by nmon.

In some cases if nmon tags like BBBP are shown, then the data is directly from nmon without any modification by iDoctor (besides importing it into the database.)



System configuration folder

Note: These views do not display the disk mapping (VIOS to IBM i). For that open the QAIDRNMCOR_<<COLNAME>> table under **Server-side output files** folder.

7.10.4.6.1 Disks (BBBB)

The BBBB sheet lists all disks listed in the ODM together with the capacity (in GBs) and the adapter type (SCSI/SSA/Fibre) as reported by lsdev. Note that some fiber-attached devices do not report their capacity to AIX. **Note:** iDoctor assigns a disk ID to each disk for analysis purposes and this will be found in several iDoctor reports.

File ID (FILEI	Disk ID (within iDoctor DB) (DISKID)	Disk name (DISKNA	Size in GBs (SIZE_G	Disk attach type (DISC_ATTACH_TYPE)
1	0001	hdisk71	80.00	EMC
1	0002	hdisk72	300.00	EMC
1	0003	hdisk75	60.00	EMC
1	0004	hdisk73	32.00	EMC
1	0005	hdisk76	2.00	EMC
1	0006	hdisk74	50.00	EMC
1	0007	hdisk77	80.00	EMC
1	0008	hdisk80	32.00	EMC
1	0009	hdisk81	50.00	EMC
1	0010	hdisk83	2.00	EMC
1	0011	hdisk82	60.00	EMC
1	0012	hdisk78	80.00	EMC
1	0013	hdisk79	300.00	EMC
1	0014	hdisk84	80.00	EMC
1	0015	hdisk91	80.00	EMC
1	0016	hdisk89	60.00	EMC
1	0017	hdisk105	80.00	EMC
1	0018	hdisk94	32.00	EMC
1	0019	hdisk88	50.00	EMC
1	0020	hdisk87	32.00	EMC
1	0021	hdisk92	80.00	EMC
1	0022	hdisk86	300.00	EMC
1	0023	hdisk95	50.00	EMC

Disks (BBBB) view

7.10.4.6.2 Logical volumes (BBBC)

The BBBC sheet shows the output from the lspv command for all local disks at the start of the data collection.

File ID (FILEI	DATA
1	hdisk77:
1	LV NAME LPs PPs DISTRIBUTION MOUNT POINT
1	hd2 22 22 08..08..00..00..06 /usr
1	hd9var 12 12 00..12..00..00..00 /var
1	hd8 1 1 00..01..00..00..00 N/A
1	hd4 3 3 00..03..00..00..00 /
1	hd5 1 1 01..00..00..00..00 N/A

Logical volumes

7.10.4.6.3 I/O adapters (BBBD)

The BBBD sheet shows a list of all I/O adapters listed in the ODM together with the hdisks addressed through that adapter.

File ID (FILEI	Disk adapter number (ADAPTER_NUMBE	Disk adapter name (ADAPTER_NA	Total disks (DIS	Description (DESC)
1	0	fcs5	14	Virtual Fibre Channel Client Adapter
1	1	fcs6	14	Virtual Fibre Channel Client Adapter
1	2	fcs7	14	Virtual Fibre Channel Client Adapter
1	3	fcs4	14	Virtual Fibre Channel Client Adapter

I/O adapters (BBBD)

7.10.4.6.4 Fiber channel / FCSTAT (BBBF)

The BBBF sheet displays output from the FCSTAT command against each fiber channel adapter on the system.

File ID (FILEI	A	B	C	D
1	FC Adapter stats	FCs found	4	
1	FC5	fcs4		
1	FC5	fcs4		
1	FC5	fcs4	FIBRE CHANNEL STATISTICS REPORT	fcs4
1	FC5	fcs4		
1	FC5	fcs4	Device Type	Virtual Fibre Channel Client Adapter (adapter/vdevice/IE
1	FC5	fcs4	Serial Number	UNKNOWN
1	FC5	fcs4	Option ROM Version	UNKNOWN
1	FC5	fcs4	ZA	UNKNOWN
1	FC5	fcs4	World Wide Node Name	0xC0507608F9E40000
1	FC5	fcs4	World Wide Port Name	0xC0507608F9E40000
1	FC5	fcs4		
1	FC5	fcs4	FC-4 TYPES	
1	FC5	fcs4	Supported	0x0000010000000000000000000000000000000000000000000000
1	FC5	fcs4	Active	0x0000010000000000000000000000000000000000000000000000
1	FC5	fcs4	Class of Service	3
1	FC5	fcs4	Port Speed (supported)	UNKNOWN
1	FC5	fcs4	Port Speed (running)	8 GBIT
1	FC5	fcs4	Port FC ID	0x1cde0d
1	FC5	fcs4	Port Type	Fabric
1	FC5	fcs4	Attention Type	UNKNOWN
1	FC5	fcs4	Topology	UNKNOWN
1	FC5	fcs4		
1	FC5	fcs4	Seconds Since Last Reset	4523099
1	FC5	fcs4		

Fiber channel / FCSTAT (BBBF)

7.10.4.6.5 Shared processors (BBBL)

The BBBL sheet is only produced if the operating system is running in a partitions and contains details of the configuration of the LPAR at the start of the collection run.

File ID (FILEI	METRIC	Value
1	lparno	3
1	lparname	sv63117-cdn-ebus-weblogic-wk-p
1	CPU in sys	20
1	Virtual CPU	15
1	Logical CPU	120
1	smt threads	8
1	capped	0
1	min Virtual	1
1	max Virtual	20
1	min Logical	1
1	max Logical	160
1	min Capacity	0.5
1	max Capacity	9.0
1	Entitled Capacity	6.0
1	min Memory MB	10240
1	max Memory MB	168960
1	online Memory	168960
1	Pool CPU	20
1	Weight	128
1	pool id	0
1	Flags	LPARed DRable SMT Shared UnCapped PoolAuth Migratable Not-Donating AMSable.

Shared processors (BBBL)

7.10.4.6.6 Network adapters (BBBN)

The BBBN describes each network adapter in the system and shows the name, speed and MTU size.

File ID (FILEI	Network adapter name (NETWORK_NA	MTU size (MTU)	Speed (Mbits/sec) (MBITS)	Description (DESC)
1	en0	1500	10240	Standard Ethernet Network Interface
1	0	0	0	not available

Network adapters (BBBN)

7.10.4.6.7 Command output (BBBP)

This report displays command output from running several AIX commands at the time the NMON collection was taken. The commands include the following:

uptime, lsconf, lsps, lparstat, emstat, vmo, lssrad, mpstat, schedo, vmstat, wlm*, oslevel, ifconfig, netstat and more.

Note that to get output from these commands requires NMON to be running with root privileges.

File ID (FILEI	CMD	DATA
1	uptime	
1	uptime	01:30PM up 52 days, 8:21, 0 users, load average: 7.23, 8.18, 9.02, , , , ,
1	lsconf	
1	lsconf	System Model: IBM,8284-22A

Command output (BBBP)

7.10.4.6.8 Volume groups (BBBV)

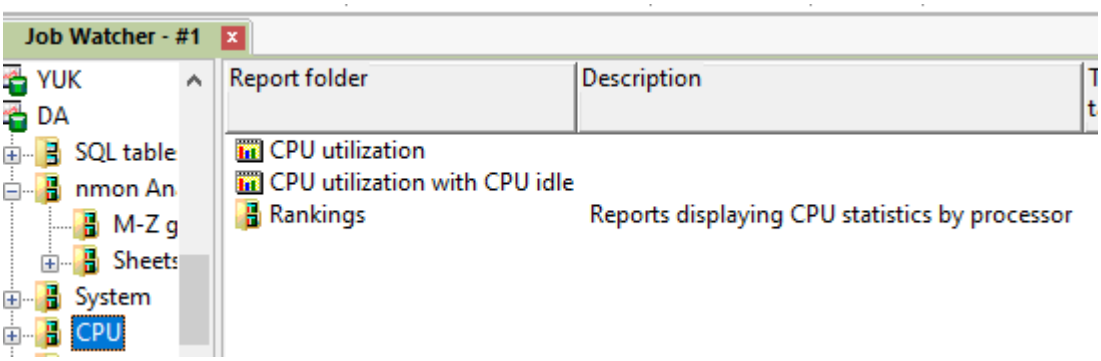
This sheet lists all of the volume groups present at the start of the collection

File ID (FILEI	DATA
1	
1	VOLUME GROUP: rootvg VG IDENTIFIER: 00f71b4000004c000000015293cae813
1	VG STATE: active PP SIZE: 512 megabyte(s)
1	VG PERMISSION: read/write TOTAL PPs: 159 (81408 megabytes)
1	MAX LVs: 256 FREE PPs: 23 (11776 megabytes)
1	LVs: 18 USED PPs: 136 (69632 megabytes)
1	OPEN LVs: 17 QUORUM: 1 (Disabled)
1	TOTAL PVs: 1 VG DESCRIPTORS: 2
1	STALE PVs: 0 STALE PPs: 0

Volume groups (BBBV)

7.10.4.7 CPU

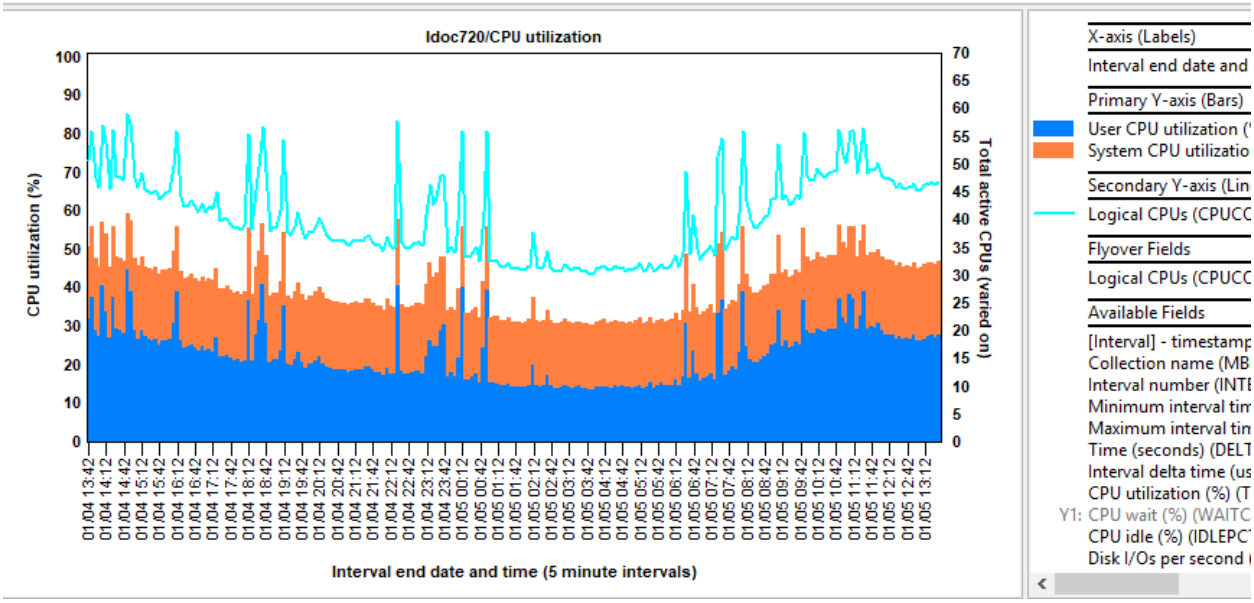
The CPU graphs display CPU utilization over time and provides an option to rank the CPU utilizations by CPU thread under the **Rankings** folder.



CPU folder

7.10.4.7.1 CPU Utilization

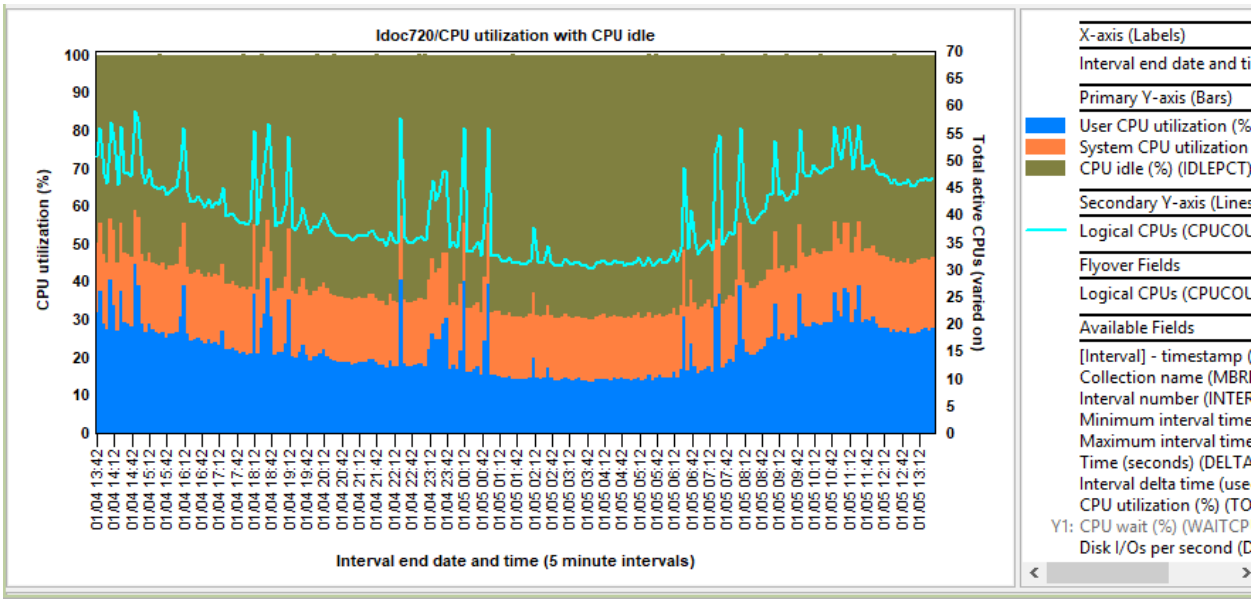
This graph shows user and system CPU utilization and CPU wait percentage along with the total varied on CPUs as the secondary Y-axis (Y2).



CPU Utilization

7.10.4.7.2 CPU Utilization with CPU idle

This is the same graph as the previous one except it also includes the CPU idle (%.)

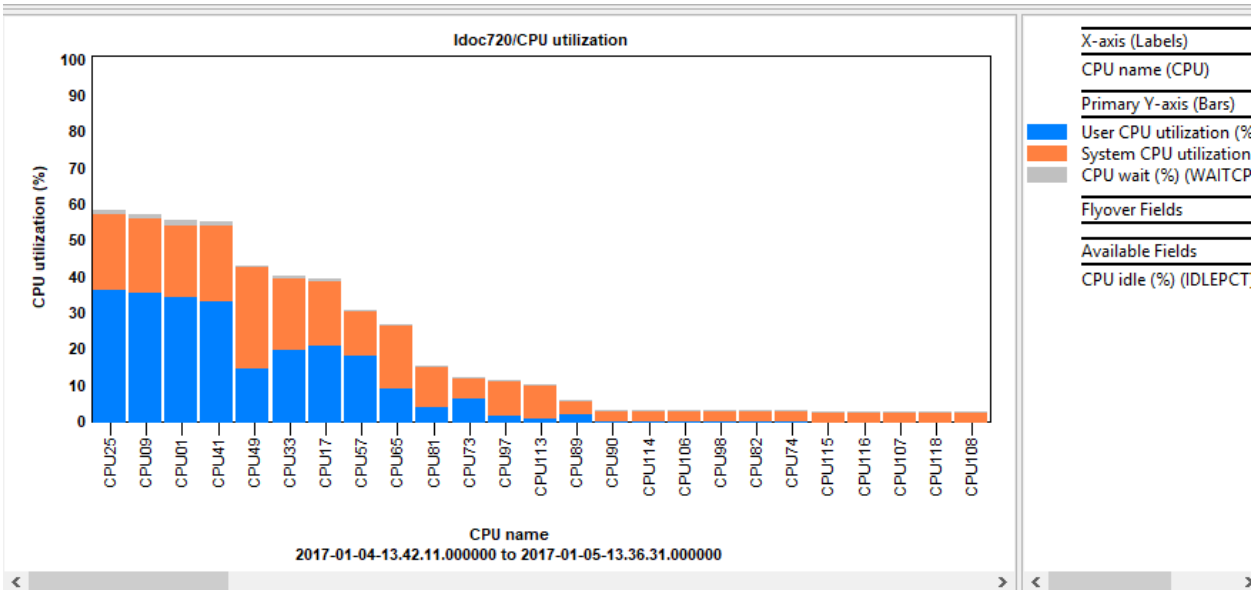


CPU Utilization with CPU idle

7.10.4.8 CPU graphs -> Rankings

The rankings subfolder contains the same graphs as described previously but ranked by CPU thread.

7.10.4.8.1 CPU Utilization

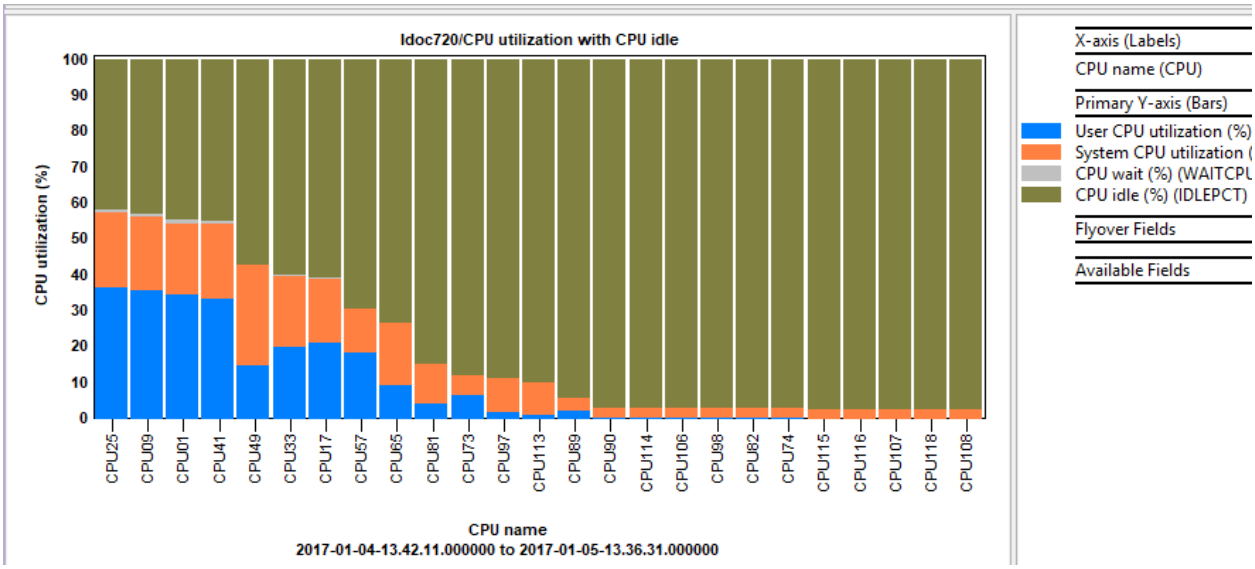


CPU Utilization

This graphs shows CPU utilization and CPU wait percentage with the CPU thread having the highest utilization shown first.

7.10.4.8.2 CPU Utilization with CPU idle

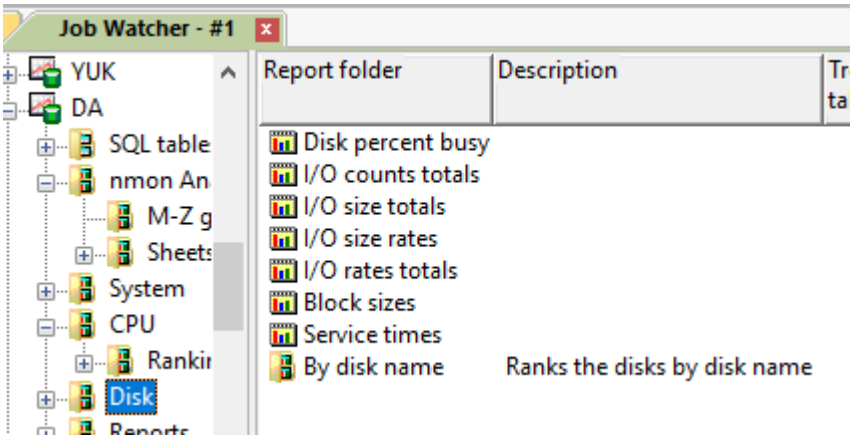
This is the same graph as the previous one except it also includes the CPU idle (%.)\



CPU Utilization with CPU idle

7.10.4.9 Disk graphs

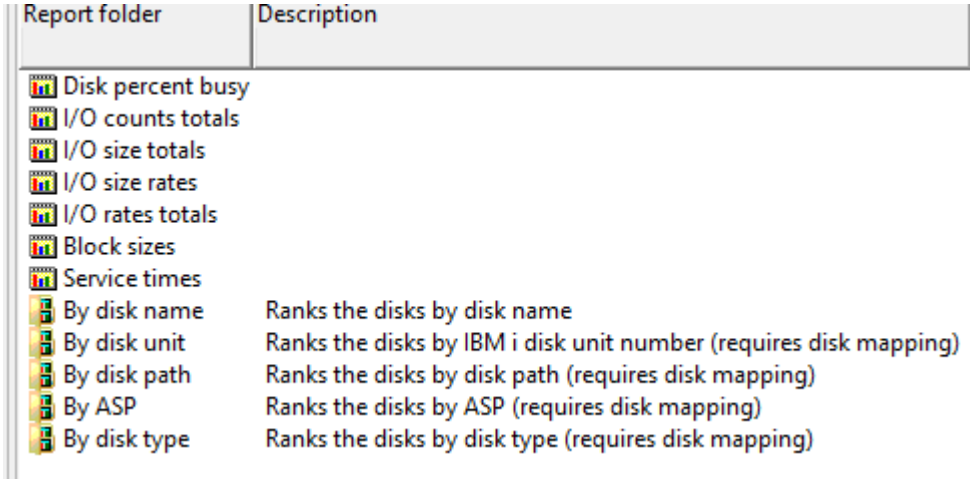
The disk graphs display the various disk statistics provided by nmon. Within the first folder are additional ranking graphs that allow the user to rank by disk name.



Nmon disk folder

If a valid VIOS to IBM i disk mapping has been provided (at analysis time when using the Analyze Data window) then the following additional ranking graphs are available:

By disk unit
By disk path
By ASP
By disk type

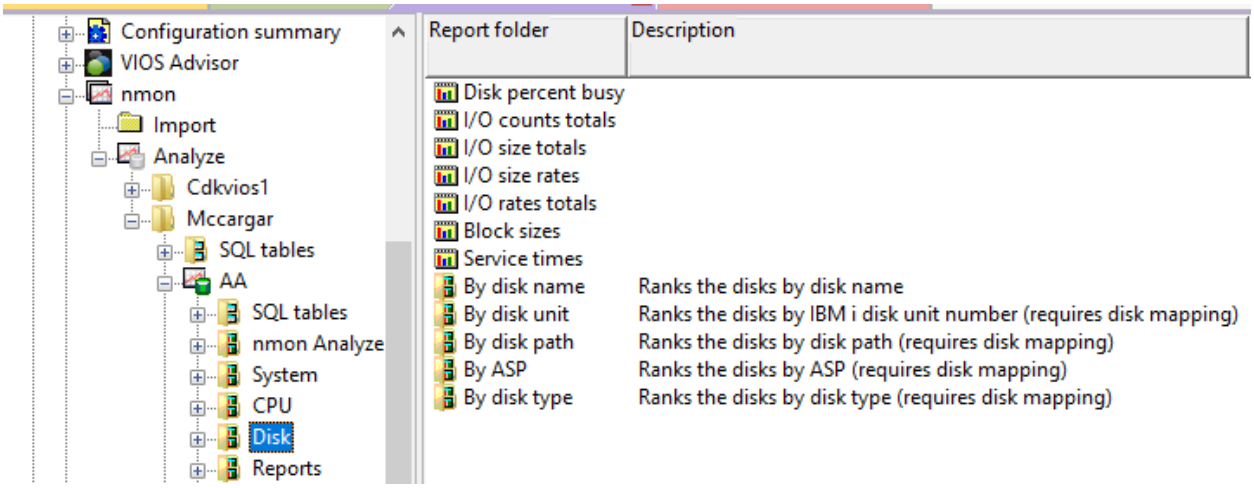


Nmon disk (graphs) folder with a disk mapping provided

Older versions of nmon do not collect all statistics shown in these graphs. In those cases, the graphs or portions of the graph may be all 0s (or show up blank.)

Tip: In the Preferences interface on the Power tab, options are available to filter these graphs by ASP or disk name. The disk name filtering option allows you to only show EMC/PowerPath disks or exclude them. After graphs have been opened, use the Change SQL Parameters menu from the graph to modify the parameters/filtering used if desired.

Note: It is highly recommended to include disk response times in your nmon data. See the section on the Power Collection Wizard for more information on how to include these. If NOT collected then you will have all 0s for the Average response times shown on these graphs.



Disk graphs folder

7.10.4.9.1 NMON Disk fields

Here is a list of the disk fields collected by NMON (taken from https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.prftools/io_stat.htm):

DISKBUSY, Disk %Busy
Percentage of time during which the disk is active.

DISKREAD, Disk Read KB/s
Total read operations from the disk in KBs per second.

DISKWRITE, Disk Write KB/s
Total write operations to the disk in KBs per second.

DISKXFER, Disk transfers per second
Number of transfers per second.

DISKRXFER, Transfers from disk (reads) per second
Number of read transfers per second.

DISKBSIZE, Disk Block Size
Total number of disk blocks that are read and written over the interval.

DISKRIO, Disk IO Reads per second
Number of disk read I/O transfers per second.

DISKWIO, Disk IO Writes per second
Number of disk write I/O transfers per second.

DISKAVGRIO, Disk IO Average Reads KBs/xfer
Average number of KBs that are read from the disk per read I/O operation.

DISKAVGWIO, Disk IO Average Writes KBs/xfer
Average number of KBs that are written to the disk per write I/O operation.

DISKSERV Disk Service Time msec/xfer
Average disk I/O service time per transfer in milliseconds.

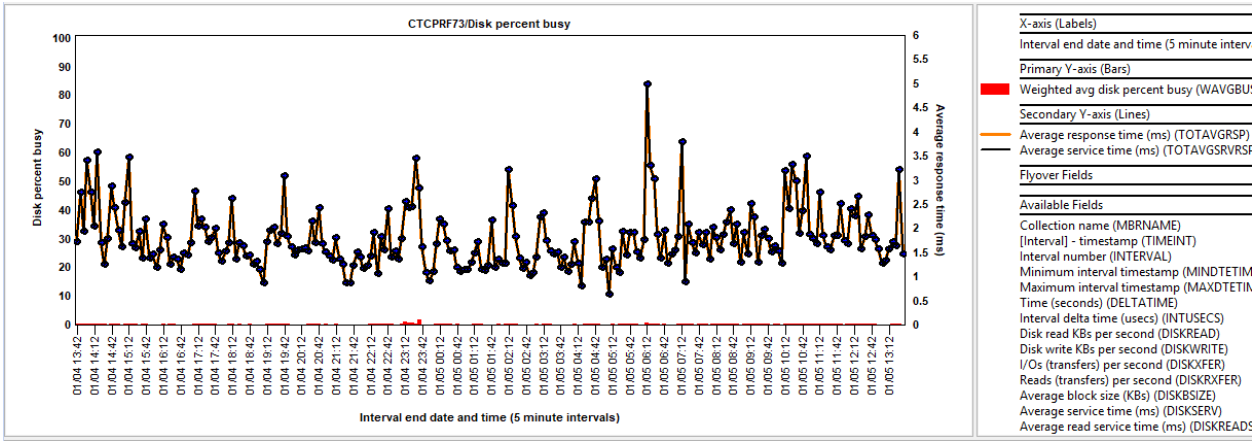
DISKREADSERV, Disk Read Service Time msec/xfer
Average read disk service time per transfer in milliseconds.

DISKWRITESERV, Disk Write Service Time msec/xfer
Average write disk service time per transfer in milliseconds.

DISKWAIT, Disk Wait Queue Time msec/xfer
Average time spent in the disk wait queue per transfer in milliseconds.

Note: iDoctor does not currently graph all of these, but additional disk graphs exist under the nmon Analyzer folder

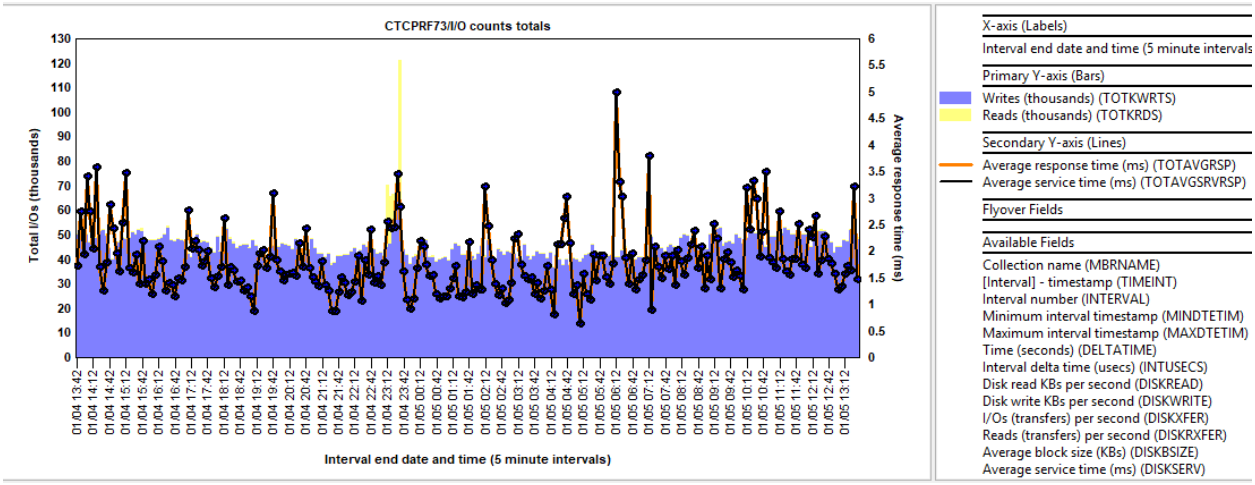
7.10.4.9.2 Disk percent busy



Disk percent busy

This graph displays the average disk percent busy (calculated as a weighted average that exclude 0 values) along with the average response time and average services times on the secondary Y-axis (Y2).

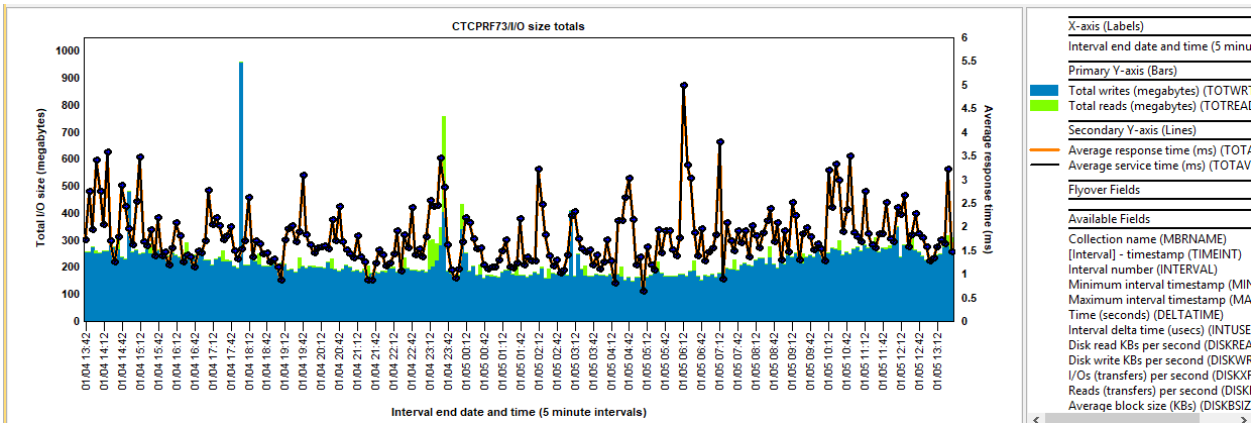
7.10.4.9.3 I/O counts totals



I/O counts totals

This graph shows the total number of reads and writes along with the average response time and average services times on the secondary Y-axis (Y2).

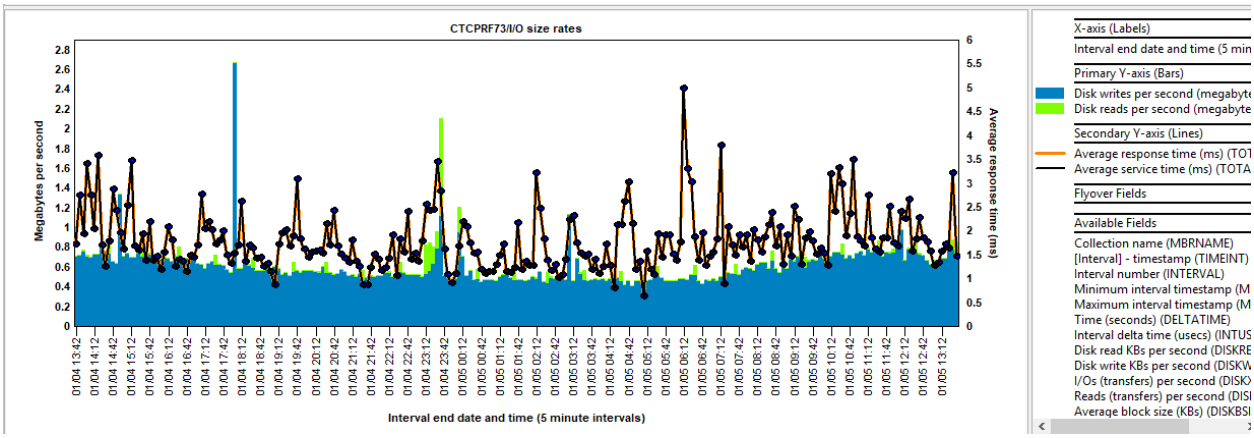
7.10.4.9.4 I/O size totals



I/O size totals

This graph shows the total I/O size (in megabytes) for both reads and writes along with the average response time and average services times on the secondary Y-axis (Y2).

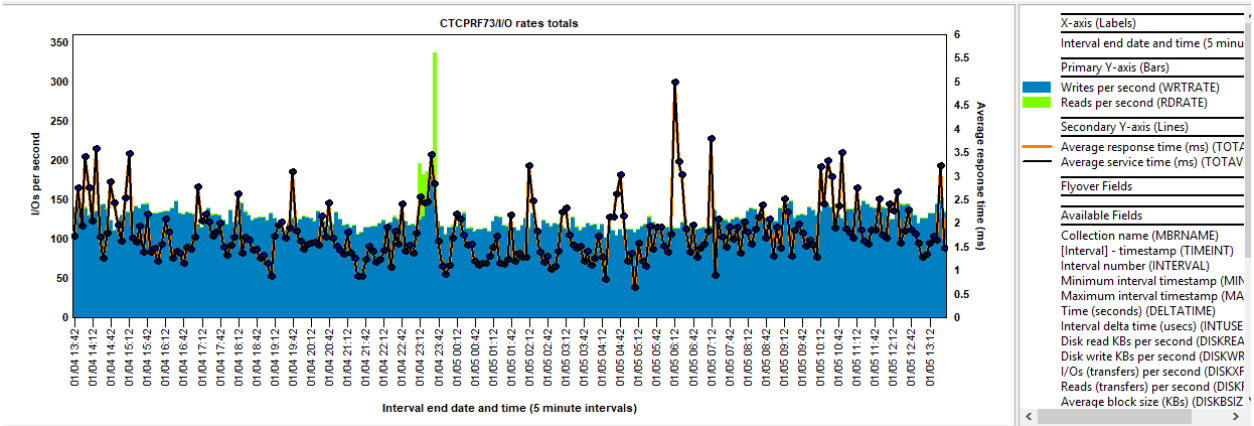
7.10.4.9.5 I/O size rates



I/O size rates

This graph shows the I/O size rates for reads and writes (in megabytes per second) along with the average response time and average services times on the secondary Y-axis (Y2).

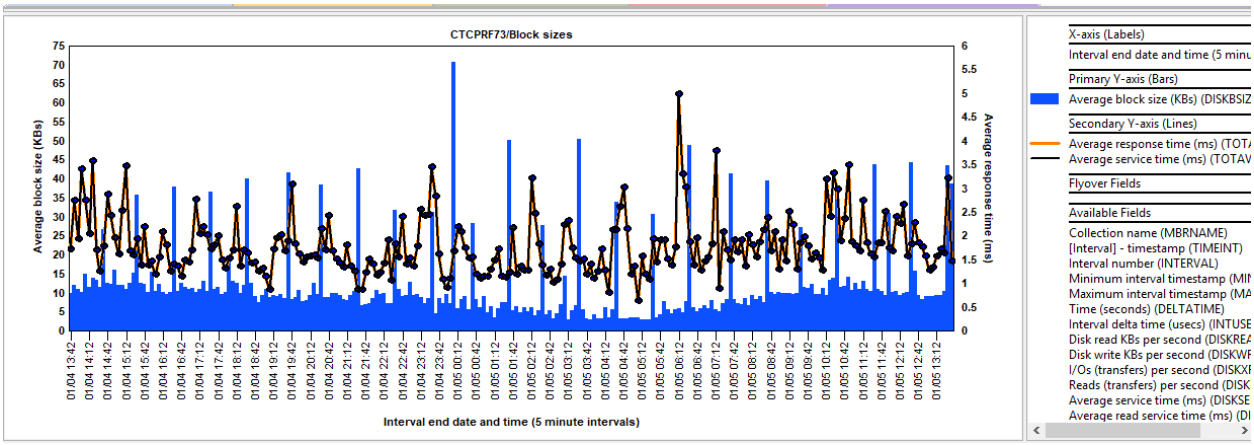
7.10.4.9.6 I/O rates totals



I/O rates totals

This graphs shows the I/O rates per second for reads and writes along with the average response time and average services times on the secondary Y-axis (Y2).

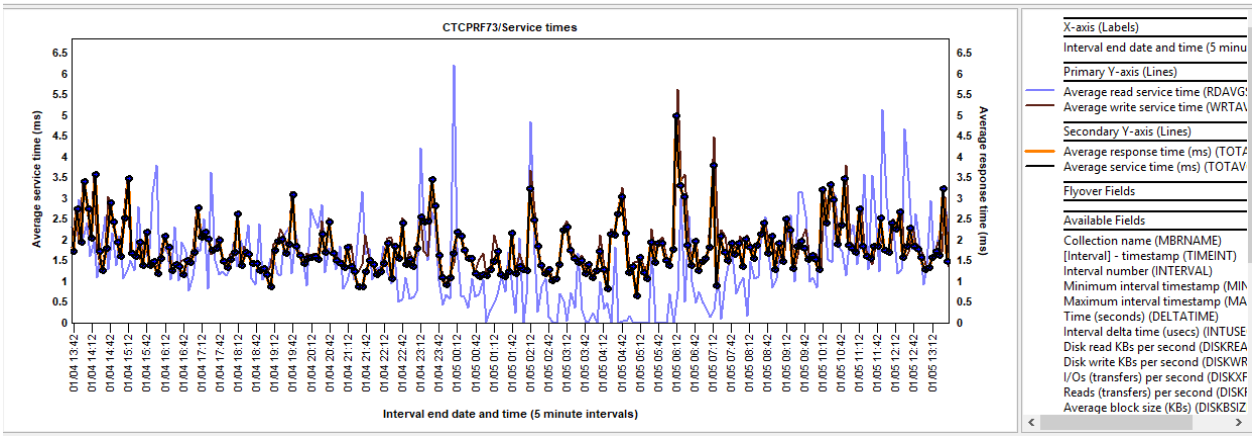
7.10.4.9.7 Block sizes



Block sizes

This graph shows the average block size along with the average response time and average services times on the secondary Y-axis (Y2).

7.10.4.9.8 Service times



Service times

This graph shows the average read and write service times along with the average response time and average services time on the secondary Y-axis (Y2).

7.10.4.9.9 By disk name

These graphs rank the NMON disk statistics by disk name.

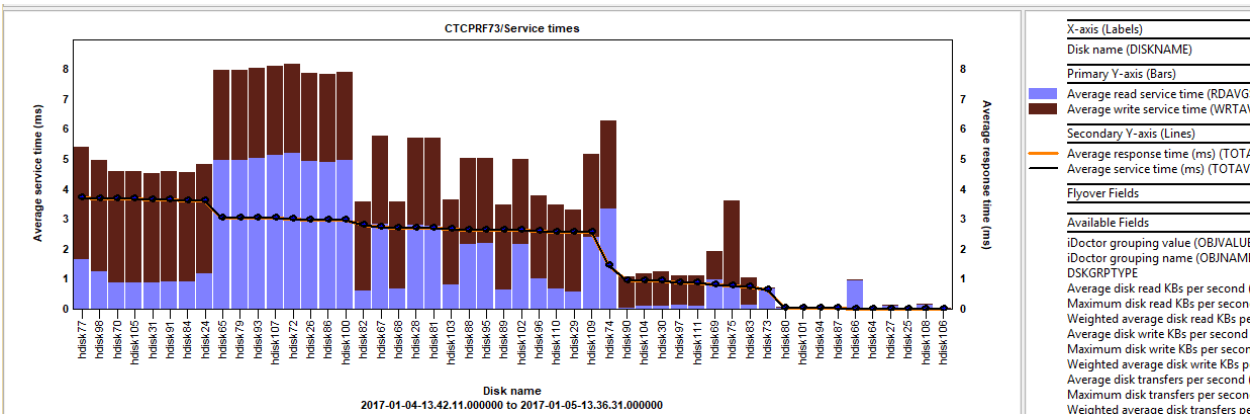
There are two possible ways to access these graphs:

1) As a drill down from the overview graphs in the previous section (in this case time filtering is used.)

2) -or- From the by disk name folder (time filtering is not used, entire collection shown.)

The graphs shown are identical to the ones described in the previously with the exception that the graph is grouped by disk name instead of time interval.

The following is an example of this type of graph.



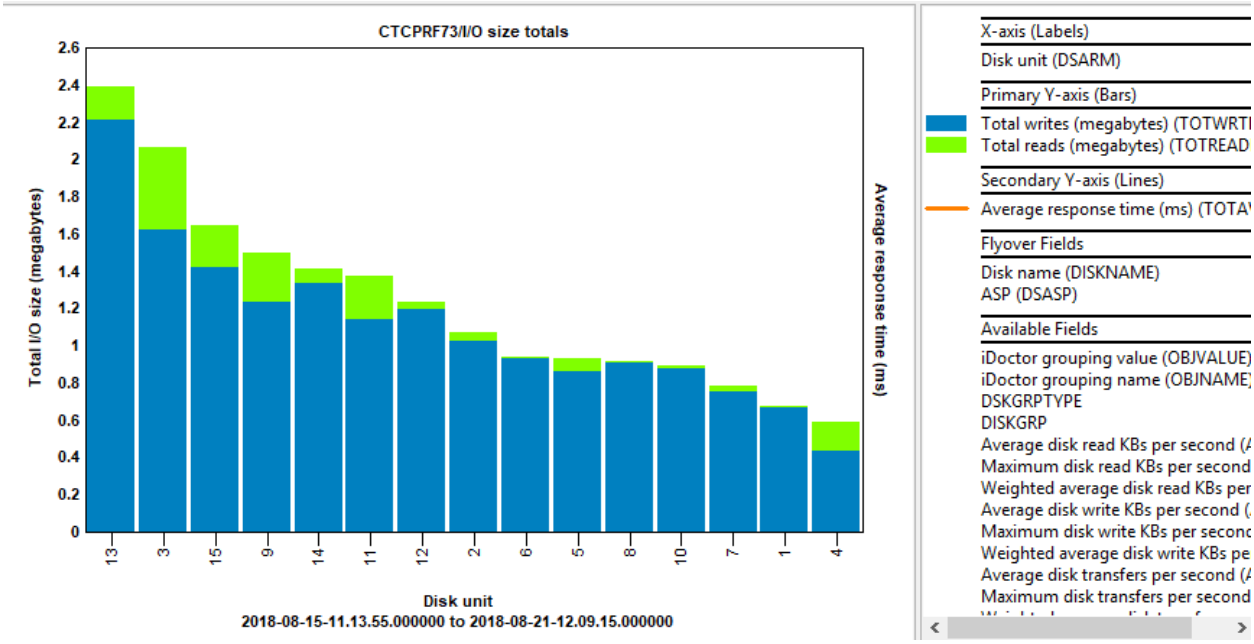
Disk (graphs) ->By disk name -> Service times example

7.10.4.9.10 By disk unit

These graphs rank the NMON disk statistics by disk unit. This type of graph requires a valid VIOS to IBM i disk mapping to be associated with the collection.

As described previously these graphs can be accessed either as a drill down or under the by disk unit folder. The graphs shown are identical to the ones described previously with the exception that the graph is grouped by disk unit instead of time interval.

The following is an example of this type of graph.



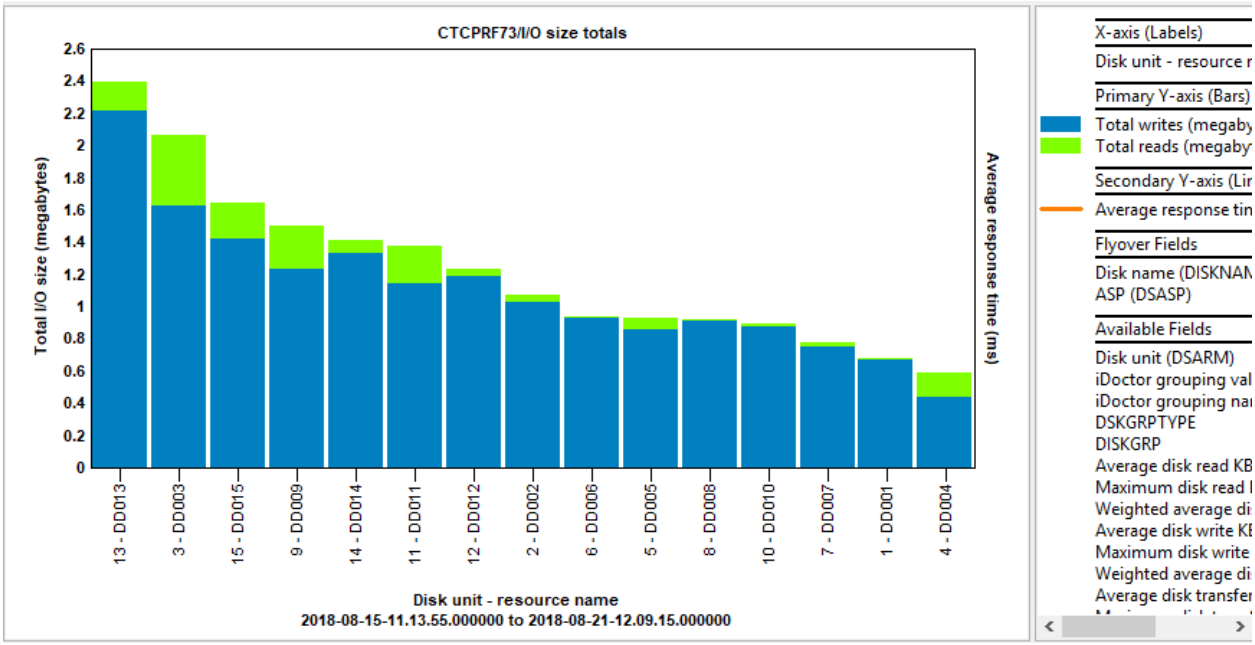
Disk (graphs) ->By disk unit -> I/O size totals example

7.10.4.9.11 By disk path

These graphs rank the NMON disk statistics by disk path (defined in Collection Services Investigator or the nmon graphs as disk unit and device resource name). This type of graph requires a valid disk mapping to be associated with the collection.

As described previously these graphs can be accessed either as a drill down or under the by disk path folder. The graphs shown are identical to the ones described previously with the exception that the graph is grouped by disk path instead of time interval.

The following is an example of this type of graph.



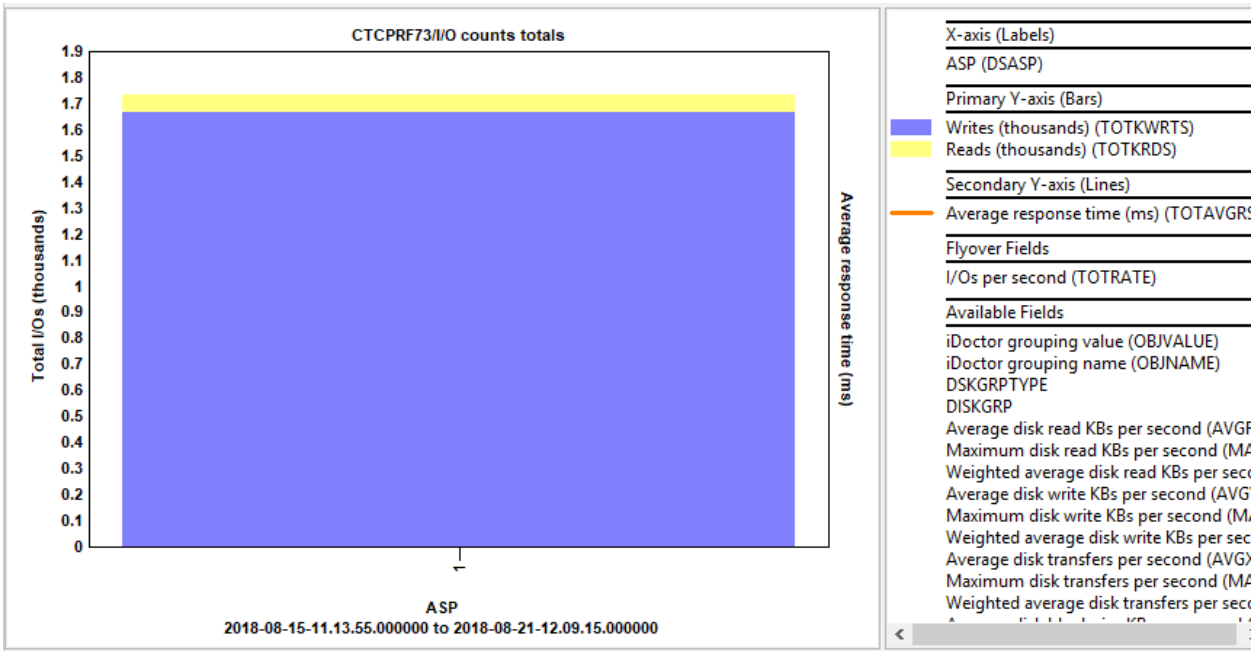
Disk graphs ->By disk path -> I/O size totals example

7.10.4.9.12 By ASP

These graphs rank the NMON disk statistics by ASP. This type of graph requires a valid disk mapping to be associated with the collection.

As described previously these graphs can be accessed either as a drill down or under the by ASP folder. The graphs shown are identical to the ones described previously with the exception that the graph is grouped by ASP instead of time interval.

The following is an example of this type of graph.



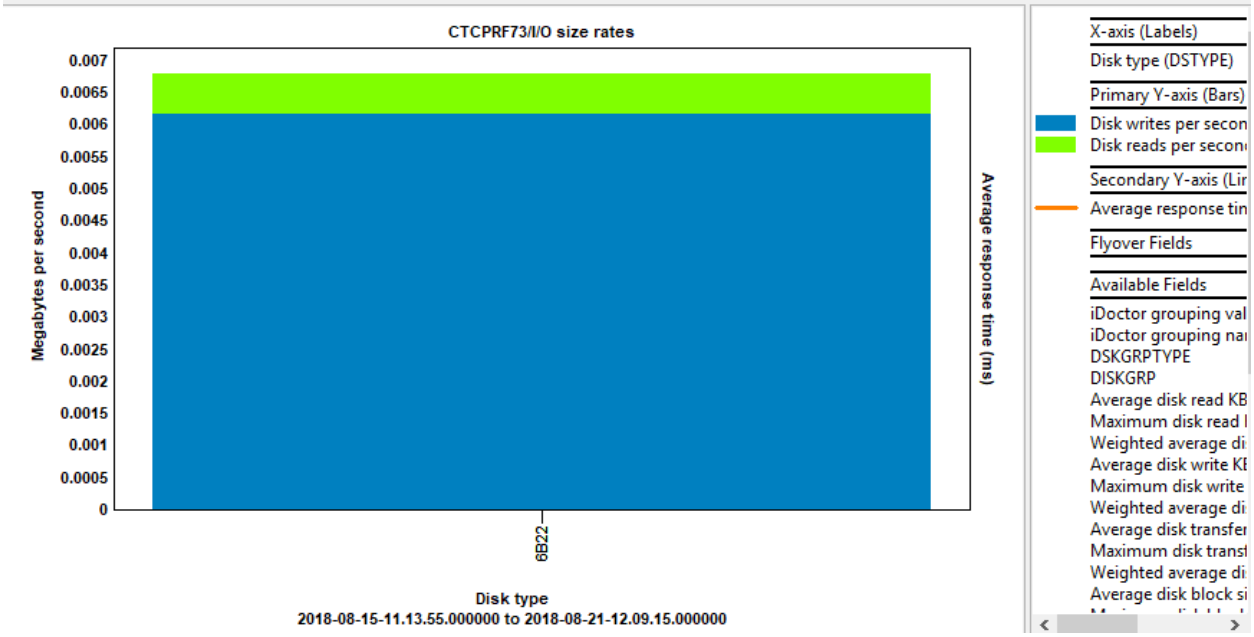
Disk (graphs) ->By ASP -> I/O counts totals example

7.10.4.9.13 By disk type

These graphs rank the NMON disk statistics by disk type. This type of graph requires a valid disk mapping to be associated with the collection.

As described previously these graphs can be accessed either as a drill down or under the by ASP folder. The graphs shown are identical to the ones described previously with the exception that the graph is grouped by ASP instead of time interval.

The following is an example of this type of graph.



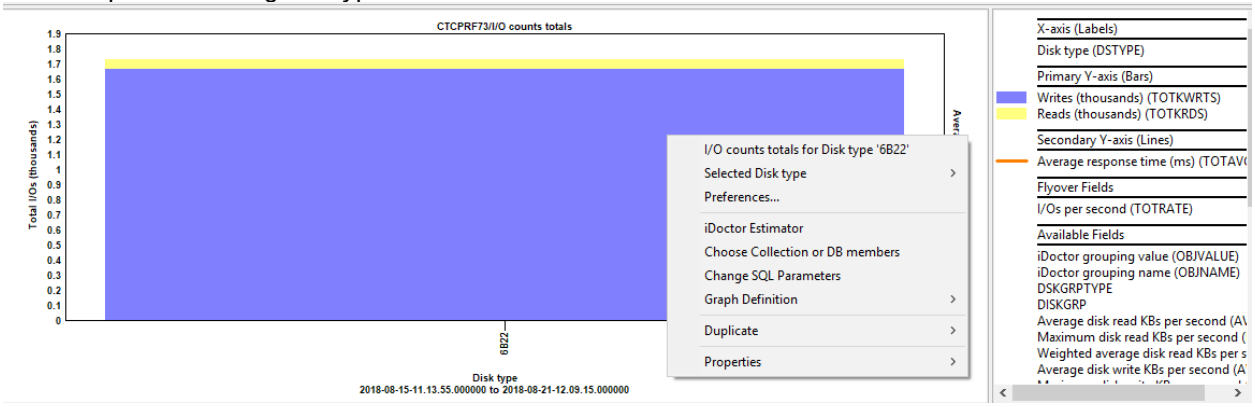
Disk (graphs) ->By disk type -> I/O size rates example

7.10.4.9.14 Disk graphs -> Single disk name, unit, path, etc over time drill down

From any of the ranking graphs described above a user can right-click the desired disk, unit, path, ASP or disk type and perform a drill down to show a graph for just the desired selection over time.

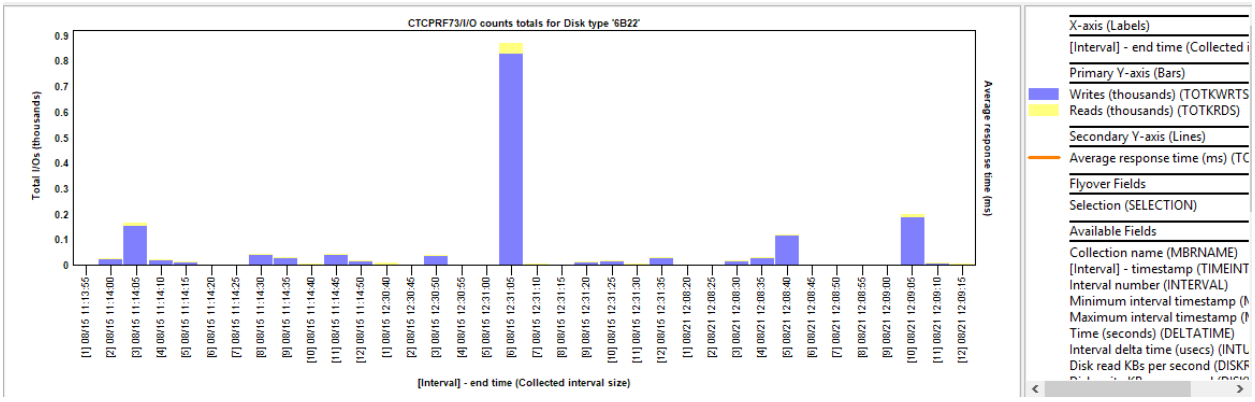
For example to drill into the previous graph shown over time, a user would right-click the bar

An example of initiating this type of drill down is shown below:



Drilling down using the Selected "Disk type" menu. Disk type could have also been disk name, disk unit, disk path or ASP depending on the type of graph.

And then an example of this type of graph is:



I/O counts totals for Disk Type 6B22

7.10.4.10 TOP graphs

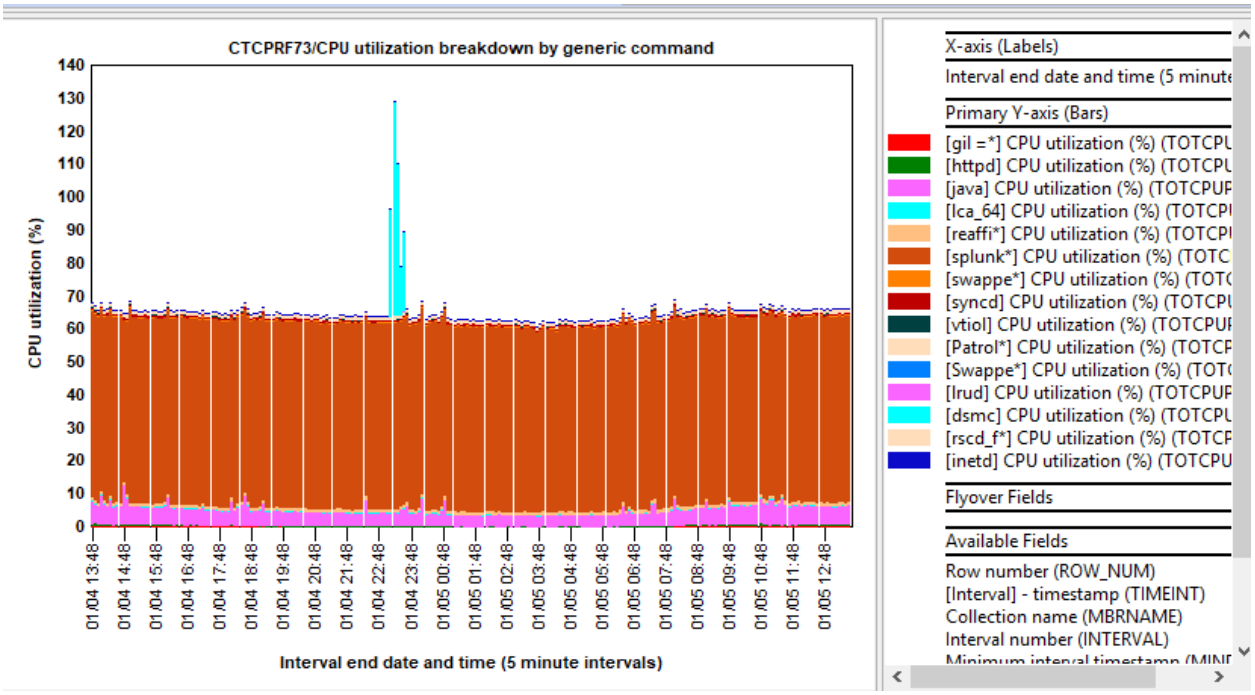
The top graphs show data only from the Top collected processes by NMON. These processes are the ones that satisfy the top processes CPU filtering parameter when creating the collection.

Note: Collectively these graphs do NOT show all metrics on the system.

These graphs display CPU utilization, paging size, character IO, memory usage over time.

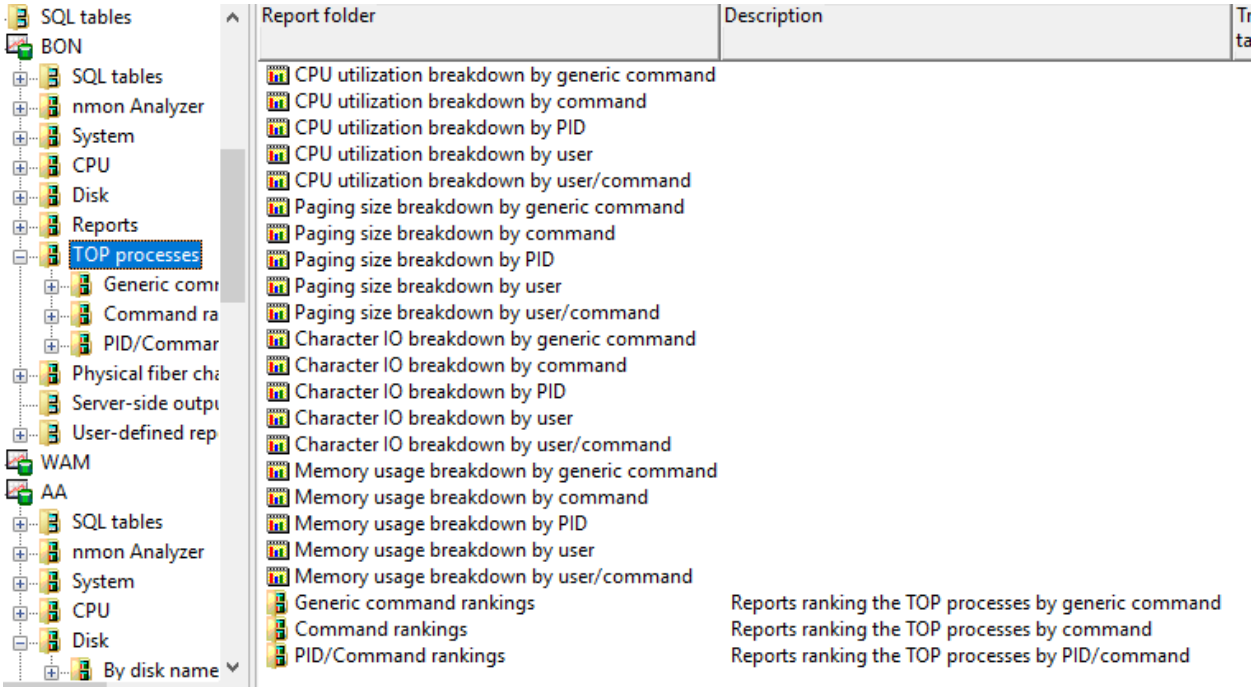
Additional graphs are also shown which show the same metrics but ranked by command or PID (process ID) or broken down by command, PID, etc.

An example of this type of graph is:



CPU utilization breakdown by generic command

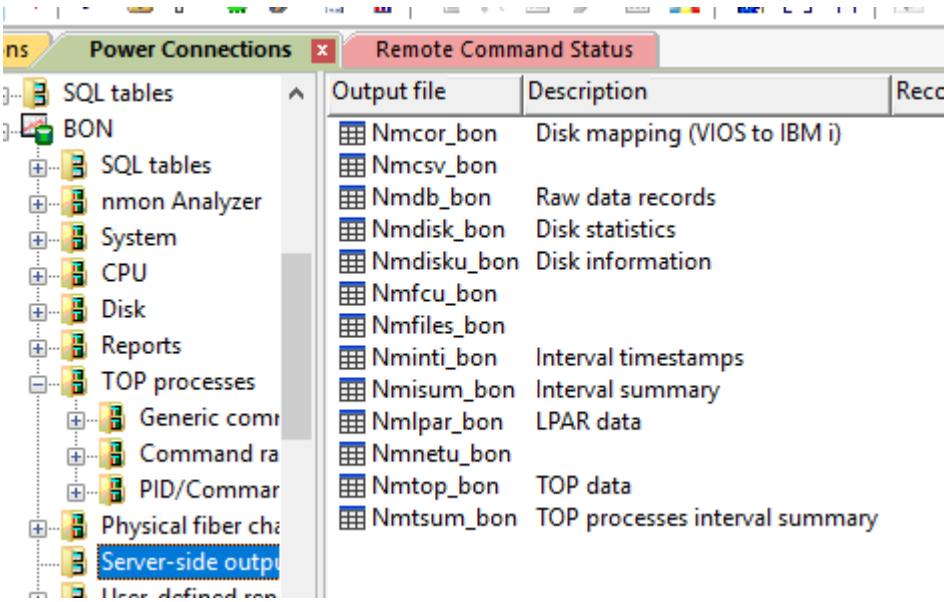
And the following lists the complete set of graphs in the TOP processes folder:



Nmon -> Top processes (graphs folder)

7.10.5 Server-side output files

This folder contains a list of tables associated with the current collection. This is the set of the tables created during the import/analysis process and they will all begin with NM* and end with the name of the current collection.



Nmon -> Server-side output files folder

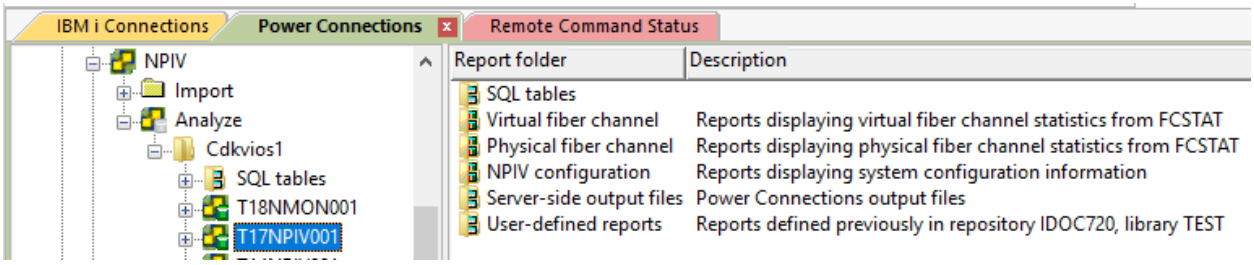
File	Description
Nmcor_<col>	The VIOS to IBM i disk mapping provided (if any) at analysis time.
Nmcsv_<col>	This file is a copy of the original .nmon file into a DB table. **Note:** This file contains just 1 32 KBs field and if the original data exceeded 32 KBs per line it will be truncated and lost during the import process.
Nmdb_<col>	This file contains the data tag for each nmon record as well as up to 49 columns to go with it. If more than 49 columns exist in the original .nmon data they will be lost during the analysis process.
Nmdisk_<col>	This file exists in order to make writing the SQL easier for the disk graphs. 1 record is created per interval per disk. Each disk is given an iDoctor generated disk ID.
Nmdisku_<col>	This file provides information about the mapping of iDoctor generated disk IDs to hdisks on the VIOS. Disk ID (within iDoctor DB) (DISKID) / Disk name (DISKNA) 1 hdisk71 2 hdisk72 3 hdisk75 4 hdisk73 5 hdisk76 6 hdisk74 7 hdisk77
Nmfcu_<col>	This file provides the fiber channel configuration (if available in the .nmon data.) The FCID is the fiber channel identifier generated by iDoctor and used to uniquely identifier each fiber channel. ON/Disk statistics - #1 / /MCCARGAR/BON/Disk information - #1 / /MCCARGAR/BON - #1 FCID / Fiber channel device name (FCDEV) / DEVICE_TYPE / WWPN / PORT_SPEED 0 fcs4 Virtual Fibre Channel Client Adapter (adapter/vdevice/IBM C0507608F9E40000 8 GBIT 1 fcs5 Virtual Fibre Channel Client Adapter (adapter/vdevice/IBM C0507608F9E40002 8 GBIT 2 fcs6 Virtual Fibre Channel Client Adapter (adapter/vdevice/IBM C0507608F9E40004 8 GBIT 3 fcs7 Virtual Fibre Channel Client Adapter (adapter/vdevice/IBM C0507608F9E40006 8 GBIT
Nmfiles_<col>	This table contains a list of original .nmon files used to create the data during the analysis process. This table only contains the filename in each record but the relative record number of each record in this table is the unique ID for each .nmon file. Other tables such as nmdb_<col> contains a FILEID field which maps to the relative record number in this file.
Nminti_<col>	This table provides a list of intervals in the NMON data captured along with a timestamp where each interval ended. The NMON graphs use this table to ensure that time periods are properly graphed.
Nmisum_<col>	This table provides an interval summary of the nmon statistics where several commonly graphed metrics are thrown together into 1 file to make the SQL statements easier to write.
NMlpar_<col>	This table contains the LPAR data records from the nmon data but placed in a field described file.
Nmnetu_<col>	This table provides information about the network configuration. File ID (FILEI / Host name (HOSTNA / NETNAME / MTU size (MTU) / Speed (Mbits/sec) (MBITS) / NETDESC 1 en6 1500 1024 Standard Ethernet Network Interface 1 lo0 16896 0 Loopback Network Interface

Nmtop_<col>	This table provides the TOP, data tags from the original .nmon file but placed in a field described file.
Nmtsum_<col>	This table provides an interval summary for all the TOP data tags.

7.11 NPIV

This section describes the options available for NPIV-type collections.

Many graphs are found within several folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the appropriate menu.



NPIV analysis options in iDoctor

Each folder contains a series of graphs. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Some graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one.

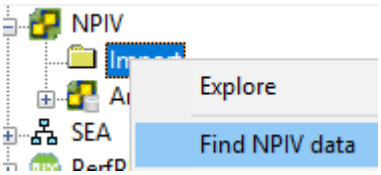
The options available under a NPIV-type collection include:

- Virtual fiber channel – Virtual fiber channel graphs summarized together or ranked by adapters.
- Physical fiber channel – Physical fiber channel graphs summarized together or ranked by adapters.
- NPIV configuration – reports that display the NPIV configuration

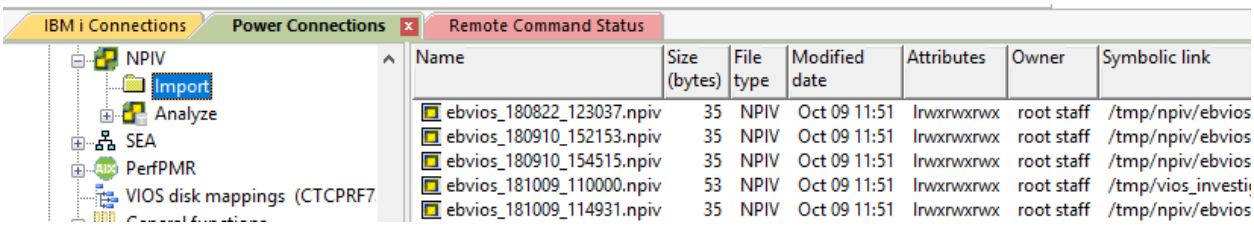
7.11.1 Import

This folder displays the available raw NPIV data found on the system that has not yet been processed into the analysis database.

Note: In order to find new data that has been sent to the system, you must use the Find NPIV data menu option.



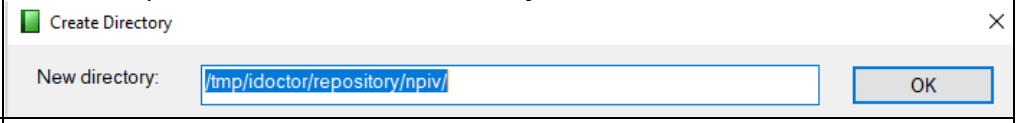
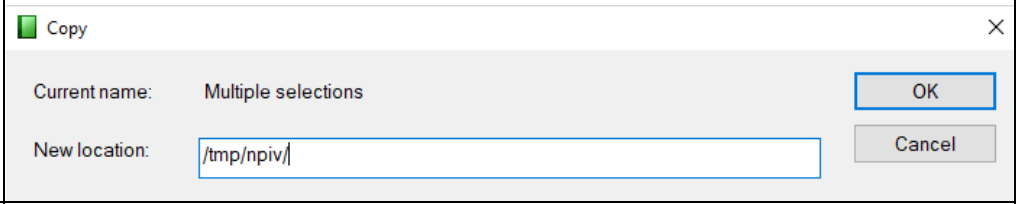
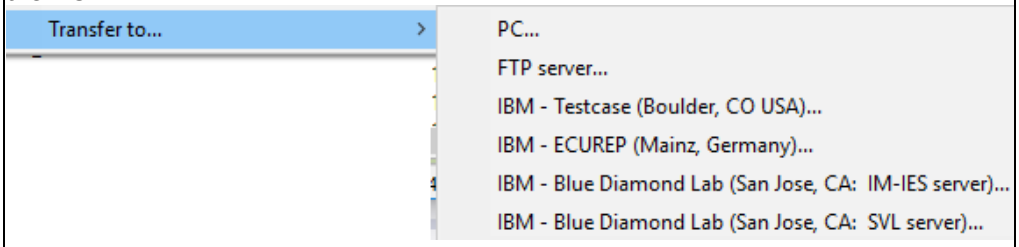
Find NPIV data menu



General functions -> Power -> NPIV -> Import folder

Note: The contents of the Import folder are cached and only rebuilt each time the Find NPIV data option is used. If these files are removed (outside of this interface) the change will NOT be immediately reflected here and only rerunning the Find option will correct this.

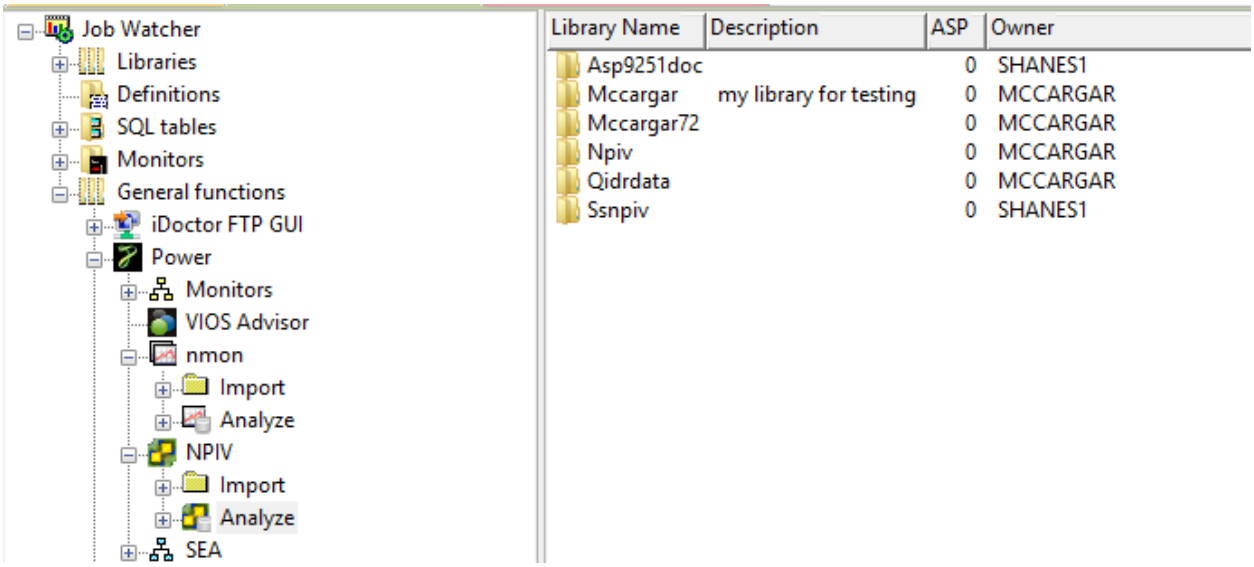
When right-clicking on an .npiv file in this folder, the following menu options are available:

Menu	Description
Open (local copy)	This option will cause the .npiv file to be downloaded to the PC and then opened using the default program specified by Windows for .npiv files.
Edit	This option will open an iDoctor editor which allows you to make changes to the file.
Analyze Data	This menu will cause the selected .npiv files to be processed by the iDoctor stored procedure that analyzes npiv data. Once complete the graphable NPIV data can be found under the NPIV -> Analyze folder.
Upload…	This option displays the Upload files from PC window which is used to transfer any previously collected .npiv files from the PC to the remote system in the desired directory.
Create Directory	Creates a directory on the remote system you are working with (IFS, VIOS or AIX.) The default path listed is the current directory for this folder. 
Copy…	This option allows you to Copy all selected files to a new directory of your choice. 
Delete	Removes the selected files from the system.
Rename	Renames the selected file.
Save	This option will Save the desired files to either a SAVF (IBM i) or tar file (AIX, VIOS) depending on the type of system being used.
Transfer to	This menu provides several choices for sending the .npiv files to another server or the PC. 

7.11.2 Analyze

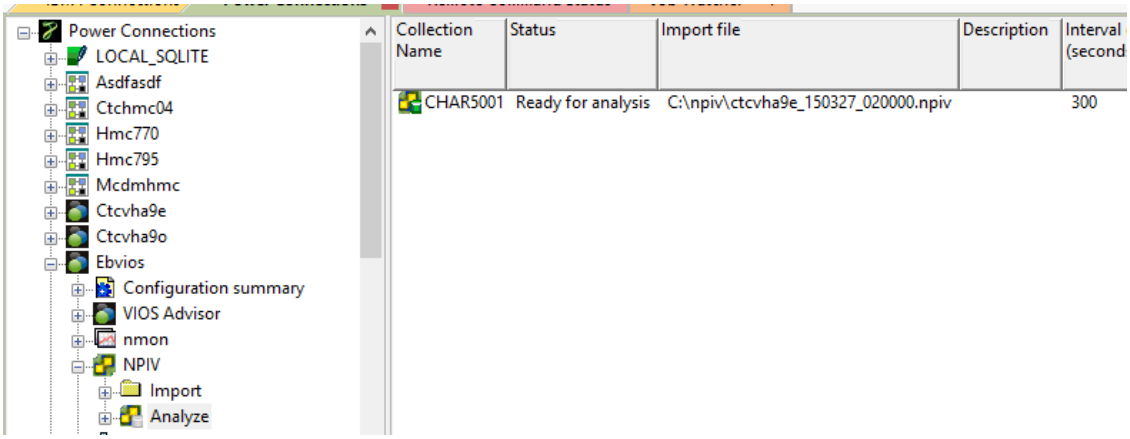
If using an IBM i as the analysis DB or connected currently to an IBM i then this folder provides a list of all libraries on the current system that contains NPIV data.

An example is:



General functions -> Power -> NPIV -> Analyze folder

Note: If using SQLite as the analysis DB then this folder contains the collections (example below.)

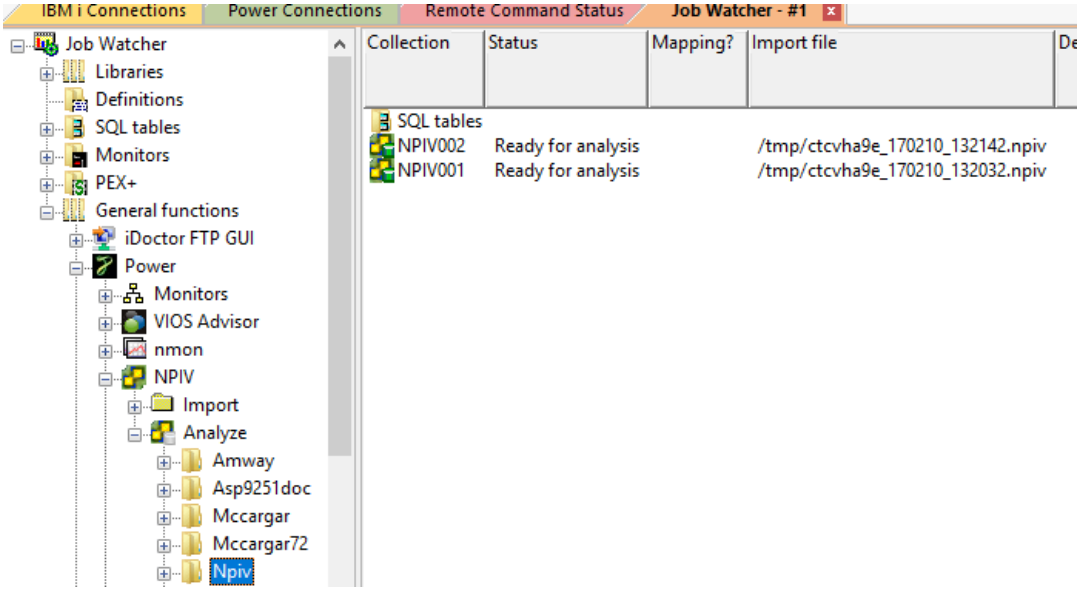


Power Connections -> VIOS -> nmon -> Analyze folder

7.11.3 Collections

Each library (or SQLite DB) contains a list of NPIV collections that have been imported into it. Each collection consists of a set of iDoctor created SQL Tables that were derived from the NPIV data and in some cases a VIOS to IBM i disk mapping (optional).

Each collection contains a detailed set of reporting options (graphs or tables) within it.



List of collections in library NPIV

7.11.3.1 Collection Fields

The list of collections contains several columns which are described below:

Field	Description
Collection	Name of the collection. This is a short unique name (10 characters or less) given to the collection at import time based on the collection name prefix value.
Status	Indicates if all necessary SQL tables have been created successfully.
Mapping?	Indicates if a disk mapping was used when creating this collection. Not having a disk mapping will mean fewer analysis options will be available.
Import File	This column displays the original file name that was used when the import occurred.
Description	A description given to the collection when the import occurred.
Interval duration	The duration of each interval within the NPIV data.
Total intervals	The total number of intervals (snapshots) taken.
Collector version	This is the iDoctor build number used when this collection was created. This by default is stored in the iDoctor scripts directory in /tmp/idoctor/qidrvrm.txt The number increases by 1 each time the iDoctor scripts we ship are updated.
Import time	The date and time the import occurred.
Start time	The date and time when the collection started.
End time	The date and time when the collection ended.
Host name	The name of the host on the system the data was collected on.
Node name	The node name for the system the data was collected on.
User name	The user name that started the collection.
Build	This returns the result of the uname -s command. On VIOS this returns AIX typically.
Command	The command used to create the NPIV data.
AIX	The version of AIX installed when the collection was created.
System serial number	The system serial number of the system where the data was created.
Mode	This indicates the import mode used on the Analyze data window when importing the data into the database. 0 = Normal 1 = Append 2 = Merge

7.11.3.2 Menus

The following menu options are available by right-clicking on a NPIV collection.

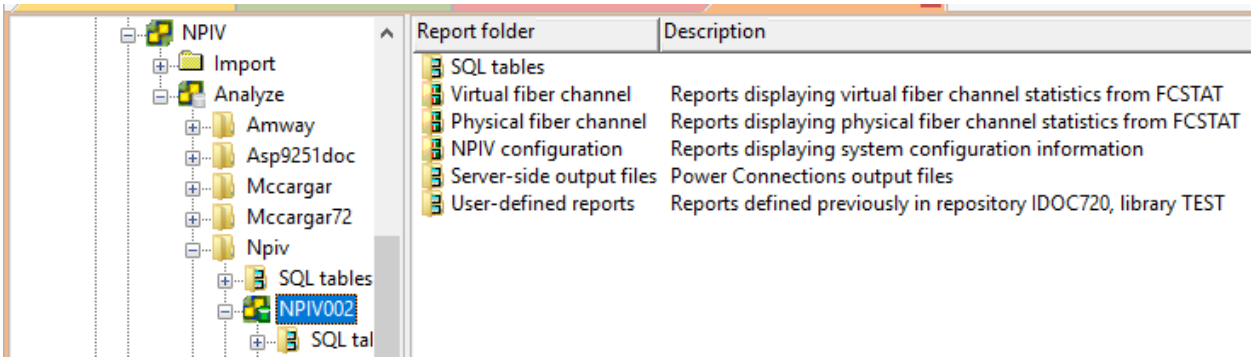
Menu	Description
Analyses	This contains the list of analysis options available to run against the selected collections. In iDoctor, these analyses are SQL stored procedures. Currently these are the options available: **Analyze Collection** – This is the interface that lets you work with all the analyses. **Run Change NPIV sensitive user data** – This will modify the data in the NPIV collection to hide/replace potential any customer sensitive data. **Note:** This option only applies when connected to an IBM i or using an IBM i as the analysis DB.
Virtual fiber channel	This contains the set of graphs for virtual fiber channel statistics. It includes (summarized) overview charts as well as several ways to rank the adapters or graph them together over time individually.
Physical fiber channel	This contains the set of graphs for physical fiber channel statistics. It includes (summarized) overview charts as well as several ways to rank the adapters or graph them together over time individually.
NPIV configuration	This contains reports that show the NPIV layout and configuration.
Generate Reports	Launches the Report Generator function that lets you create multiple reports at once.
Copy	This function will copy the desired collection to a different name in the same library or copy one or more selected collections to another library.
Delete	Removes the selected collections from the system.

Additional menu options that are common to all library folders in iDoctor are discussed here.

7.11.4 Reports

This section describes the reporting options available for NPIV collections.

Many graphs are found within several folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the appropriate menu.

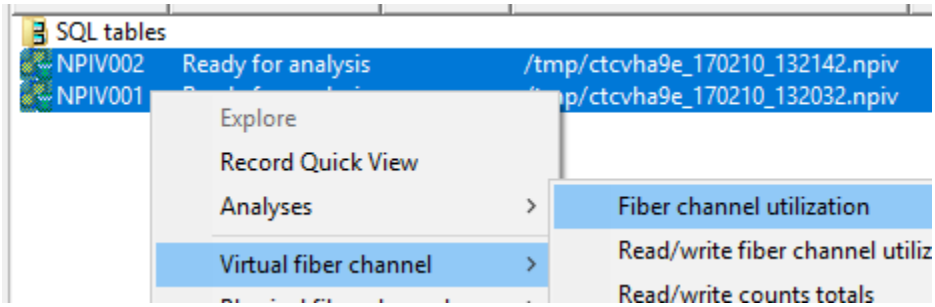


Job Watcher -> General Functions -> Power -> NPIV -> Analyze -> NPIV (lib) -> NPIV002 (collection)

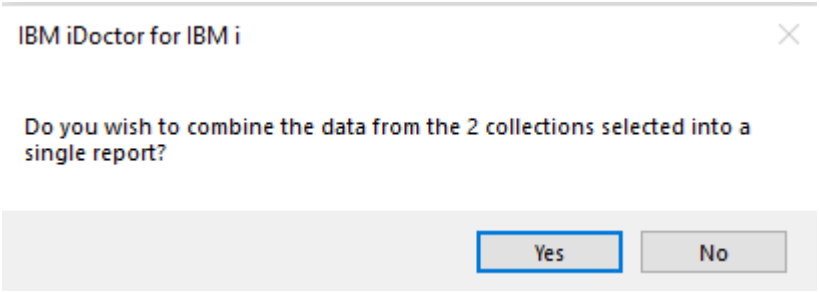
Each folder contains a series of graphs or reports. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Some graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one.

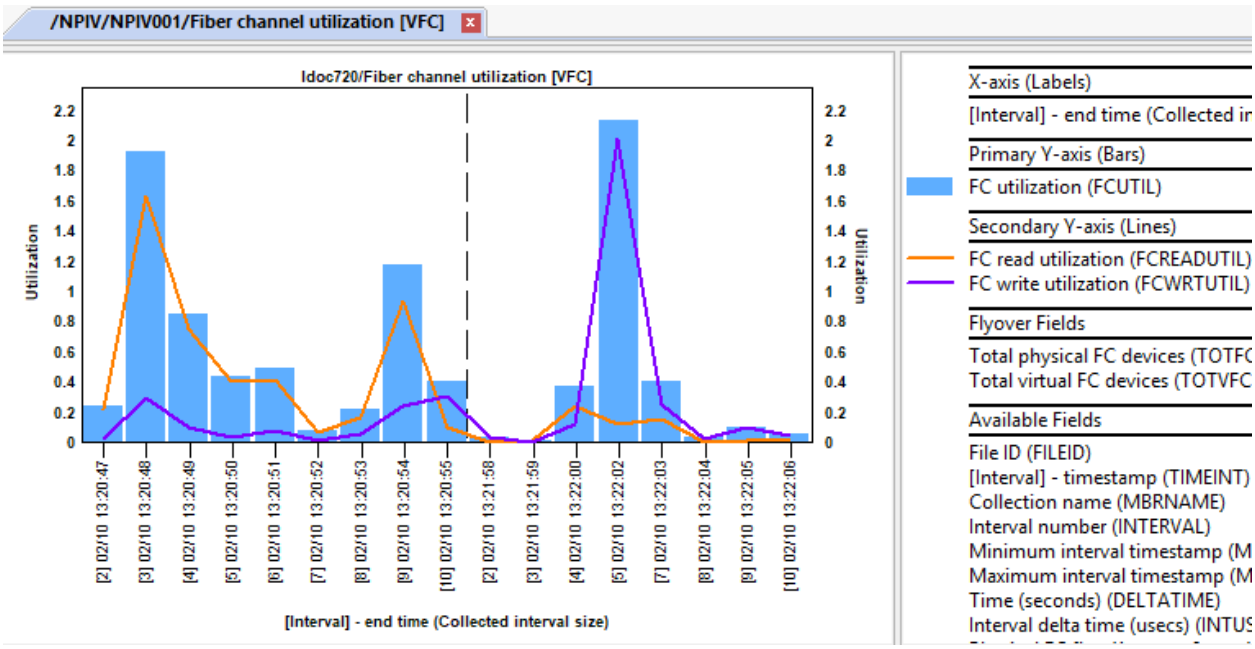
Tip: Many graphs (such as the Virtual fiber channel -> Fiber channel utilization) support graphing multiple collections at once. To do this, select all desired collections from the list of collections, then right-click and pick the desired graph from the menu.



Selecting a graph to run against multiple NPIV collections



Click Yes



Fiber channel utilization graph against 2 collections

7.11.4.1 Menus

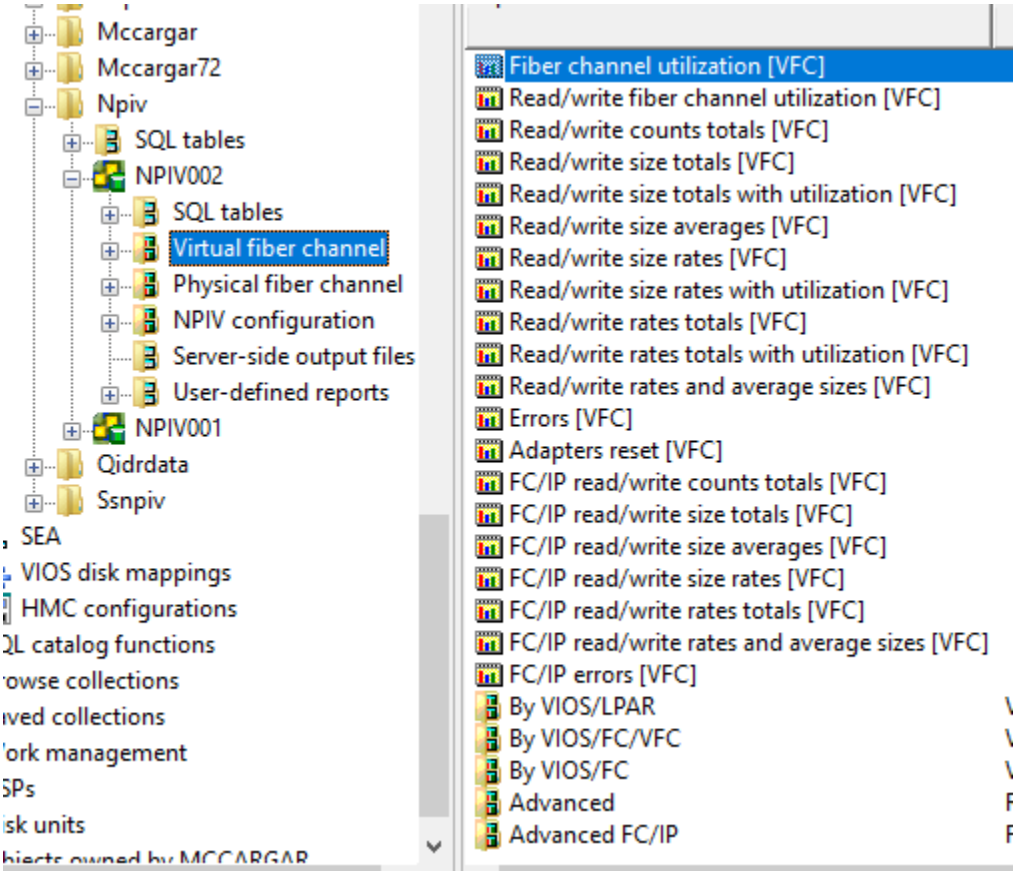
Right-clicking a graph gives a menu with the following options:

Menu	Field Description
Open graph(s)	Opens the selected graphs into a new Data Viewer or an existing one depending on the submenu available that shows the list of Data Viewers (if any are open).
Edit	This option will open the graph without running the SQL statement. The SQL Editor will be opened allowing the user to modify the query before running the SQL.

7.11.4.2 Virtual fiber channel

This set of graphs covers the **virtual** fiber channel statistics captured using the FCSTAT command and the iDoctor scripts. These graphs will use an abbreviation of [VFC] to indicate they apply to virtual fiber channel adapters.

The options available are:

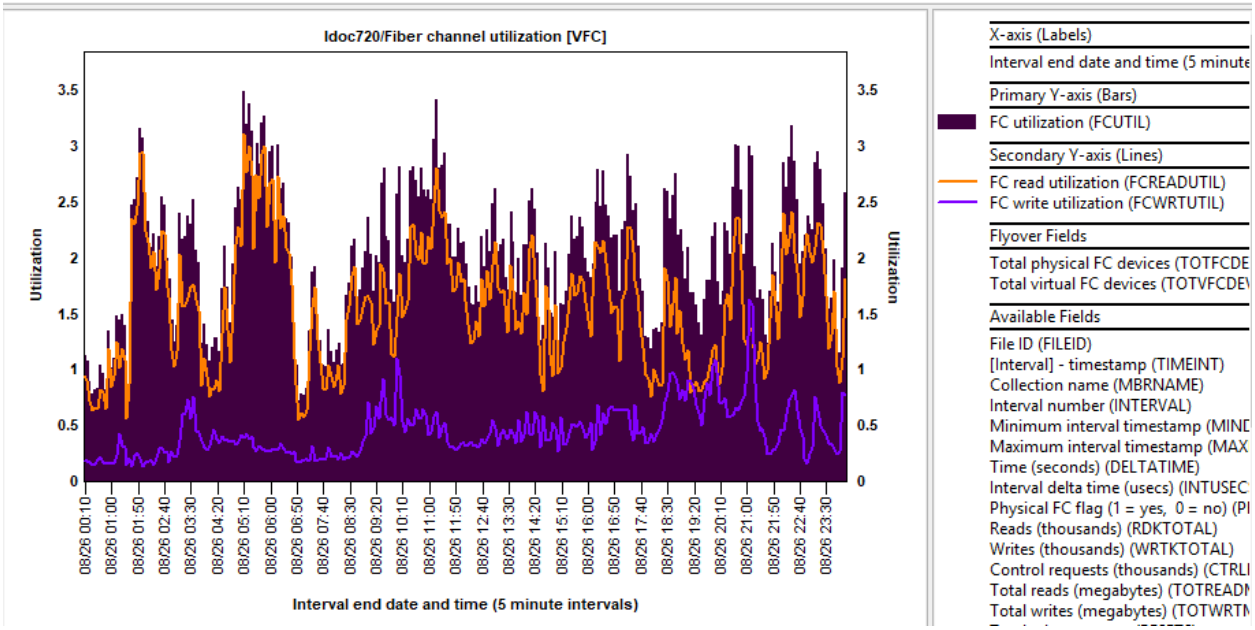


NPIV -> Virtual fiber channel folder

The By VIOS… graphs are all rankings graphs. The Advanced graphs will graph each individual adapter together over time. They will not work well if you have too many adapters (> 20 or so) and you will need to use the ranking graphs instead or a combination of the overview charts and ranking charts (filtered by time period) to do your analysis.

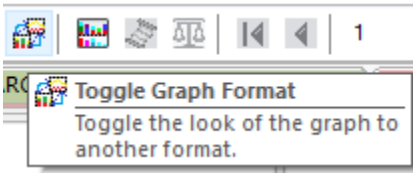
7.11.4.2.1 Fiber channel utilization [VFC]

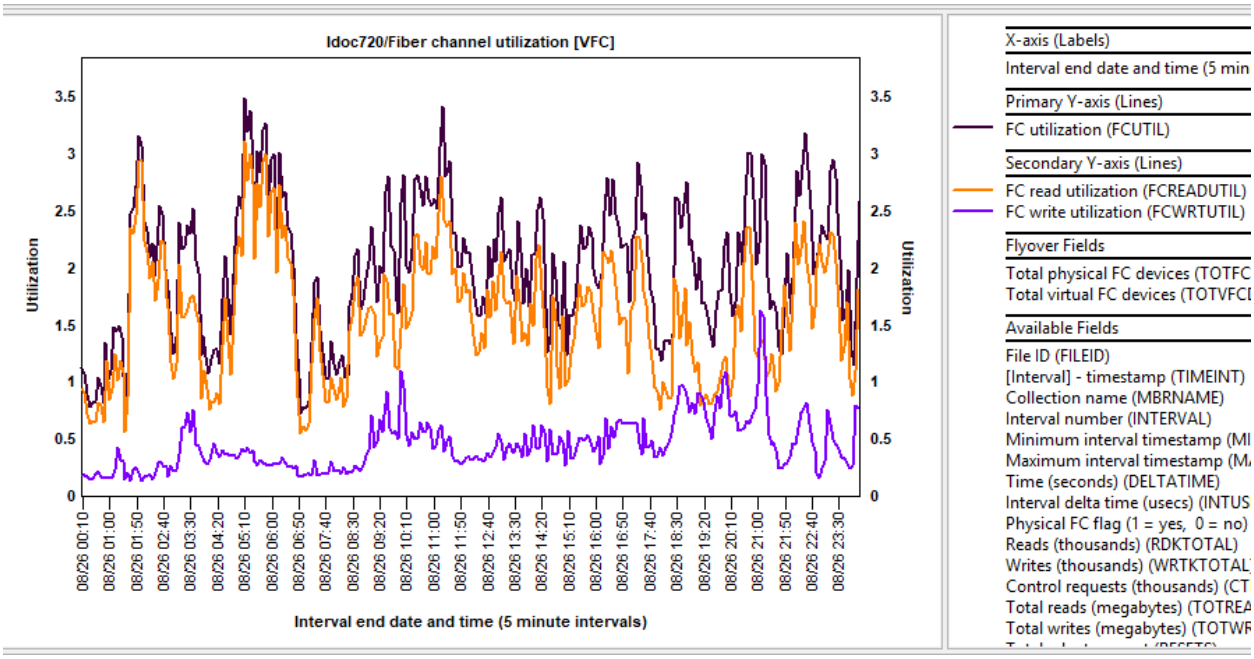
This graph summarizes the NPIV adapters together over time and displays the average fiber channel utilization (total) and read and write utilization.



Fiber channel utilization [VFC]

Tip: If you do NOT like bar graphs you can press the Toggle graph format button on the toolbar to convert these graphs to lines on the Y1-axis. Your change is remembered for next time you open this graph.

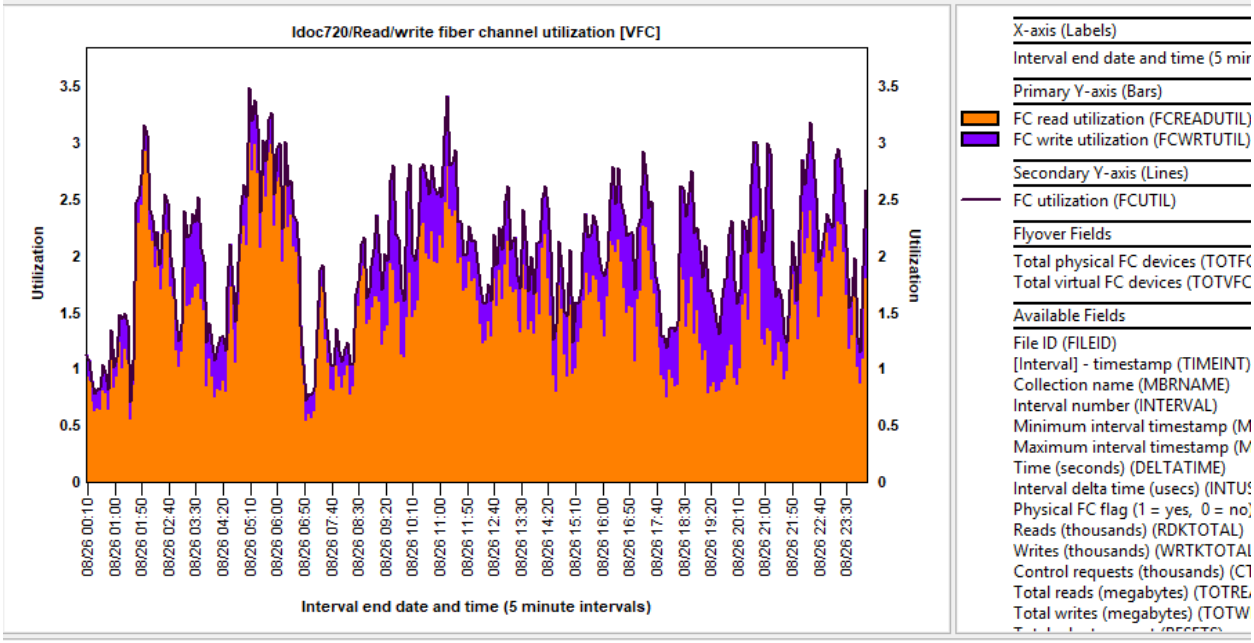




Fiber channel utilization [VFC] converted to a line graph (on Y1)

7.11.4.2.2 Read/write fiber channel utilization [VFC]

This graph shows the breakdown of read vs write utilization on the primary Y-axis (Y1) this time and puts the overall fiber channel utilization on the second Y axis (Y2).

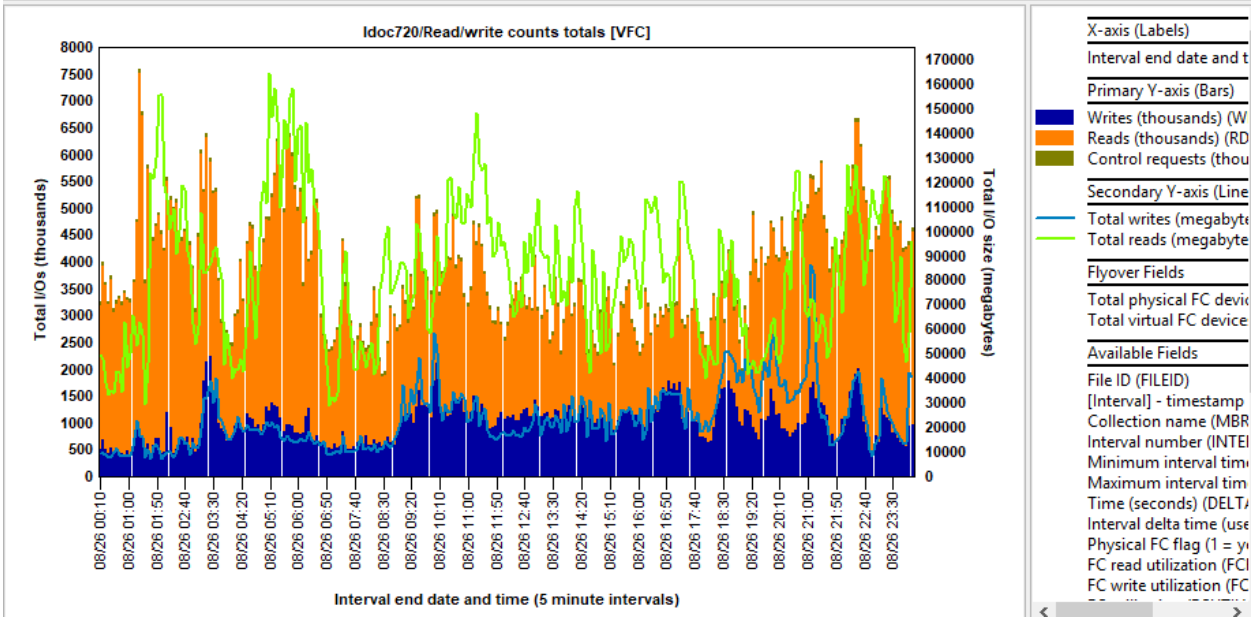


Read/write fiber channel utilization [VFC]

7.11.4.2.3 Read/write counts totals [VFC]

This graph displays the reads, writes and control requests added together for all virtual fiber channels.

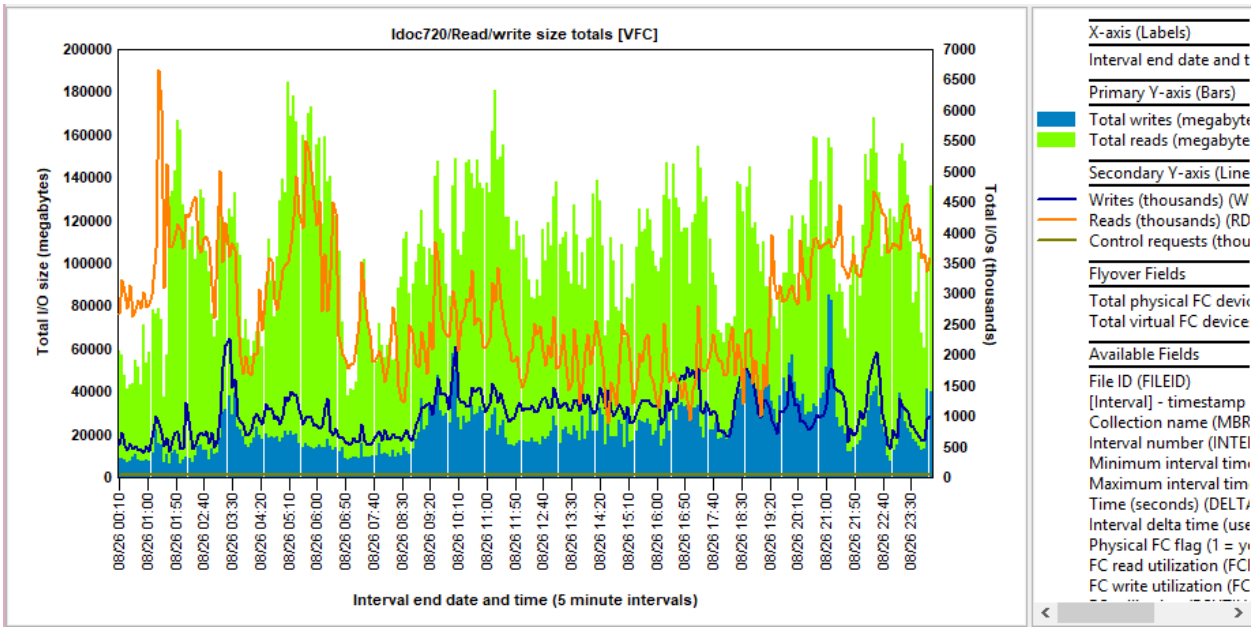
The second Y-axis shows the total I/O size (MBs) of the read and write requests separately.



Read/write counts totals [VFC]

7.11.4.2.4 Read/write size totals [VFC]

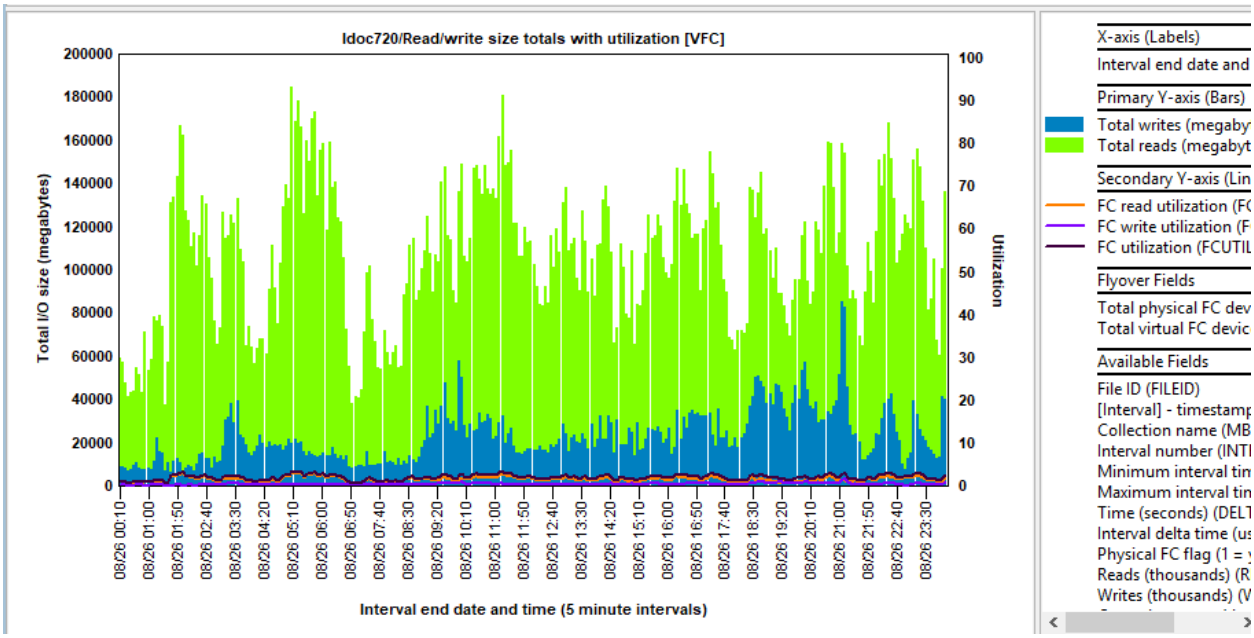
This graph is the same as the previous graph it just flips the Y1 and Y2 axis content around.



Read/write size totals [VFC]

7.11.4.2.5 Read/write size totals with utilization [VFC]

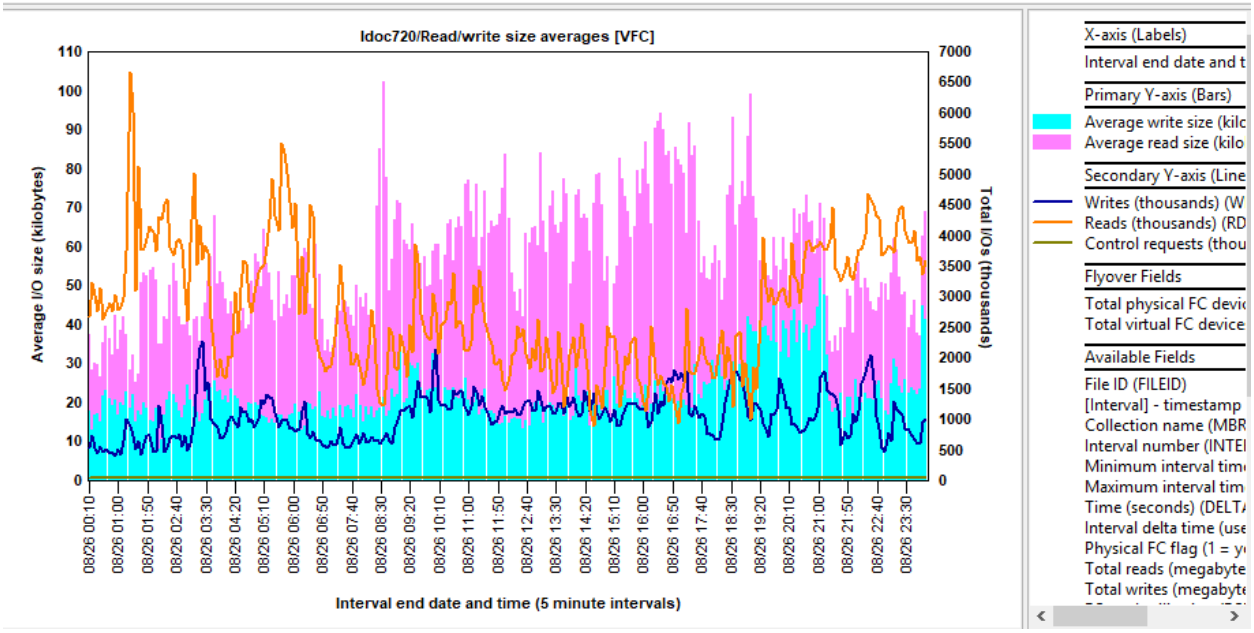
This graph shows the total I/O size (MBs) of the read and write requests separately and also displays utilization metrics on the Y2-axis.



Read/write size totals with utilization [VFC]

7.11.4.2.6 Read/write size averages [VFC]

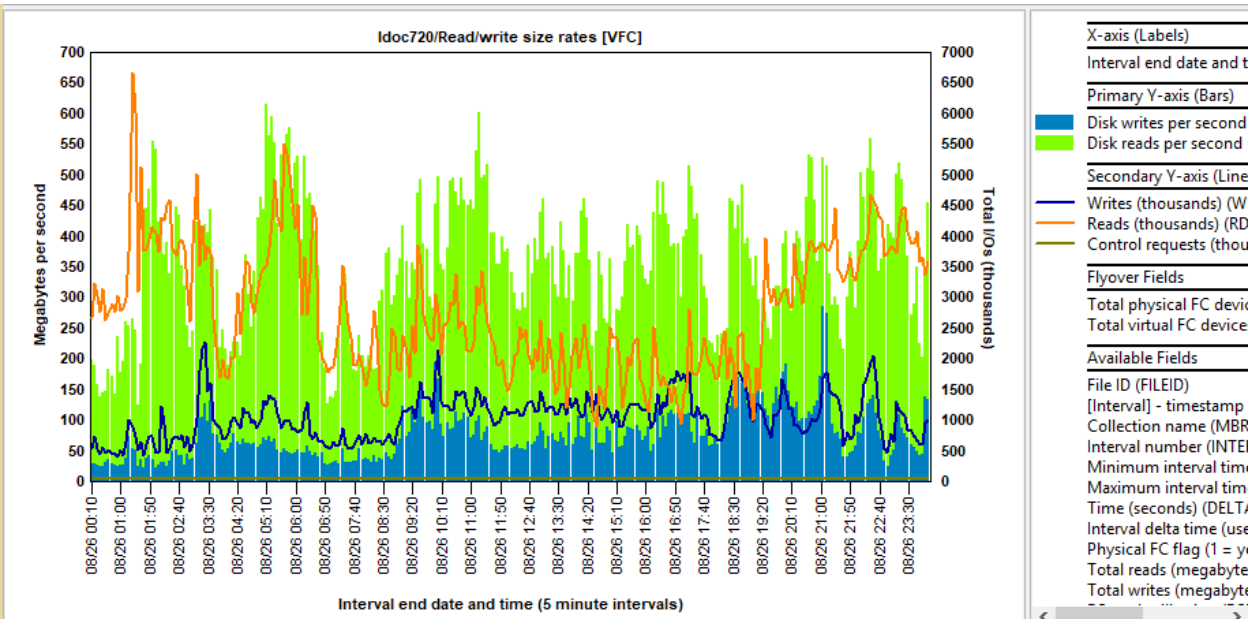
This graph displays the average size of each read/write operation and puts the total writes, reads and control requests on the Y2-axis.



Read/write size averages [VFC]

7.11.4.2.7 Read/write size rates [VFC]

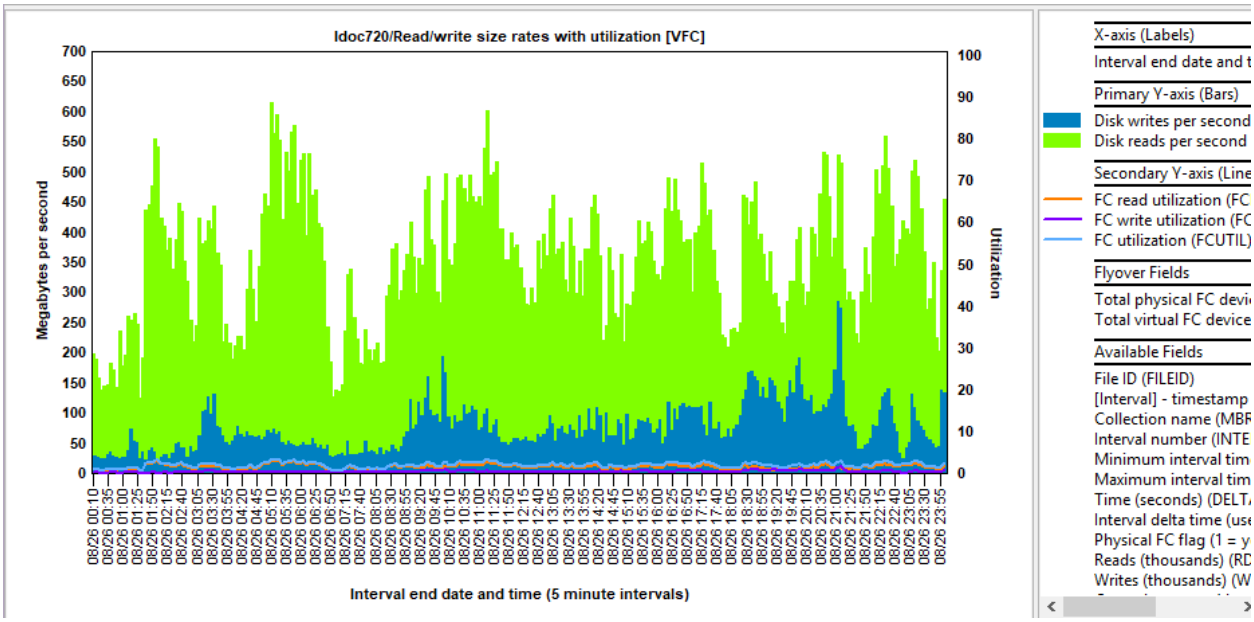
This graph displays the total read/write sizes (in MBs) per second over time.



Read/write size rates [VFC]

7.11.4.2.8 Read/write size rates with utilization [VFC]

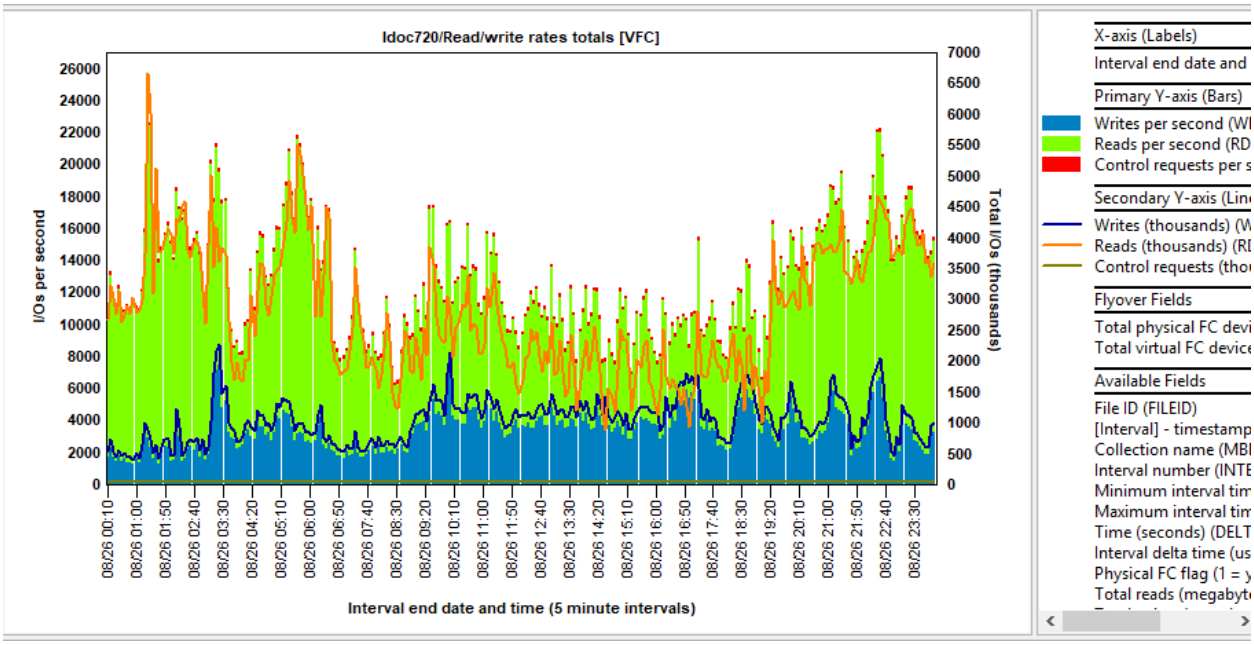
This is the same as the prior graph but replaces the Y2 counts with utilizations.



Read/write size rates with utilization [VFC]

7.11.4.2.9 Read/write rates totals [VFC]

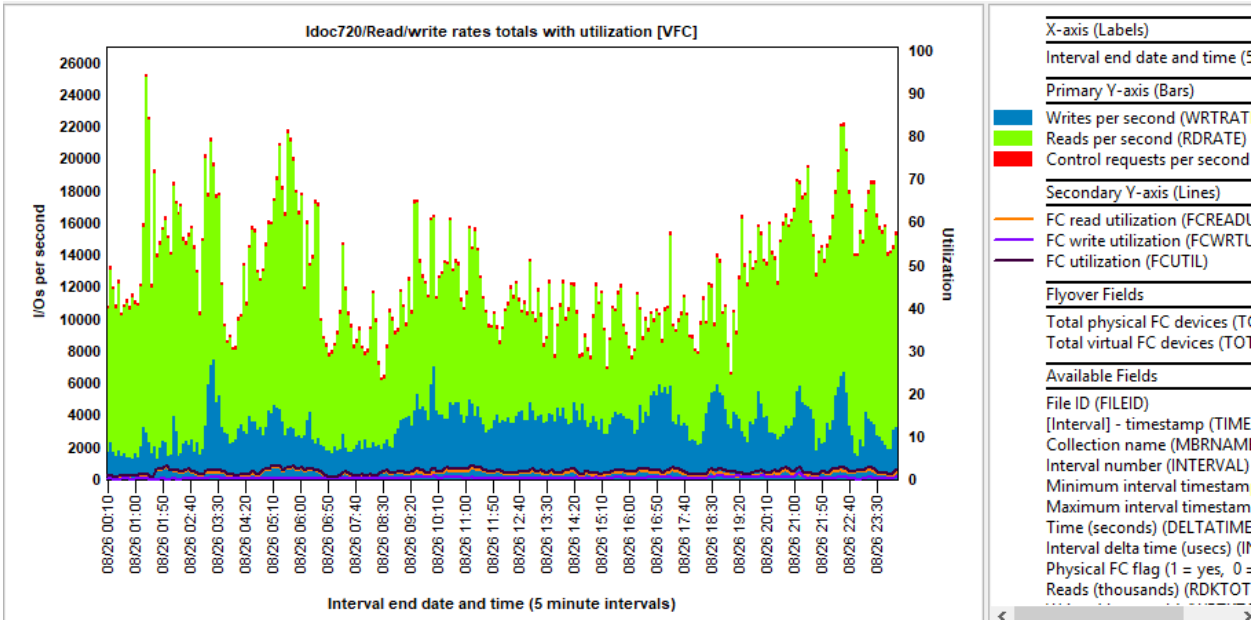
This graph dipslays the read, write and control requests rates per second with the total I/Os on the Y2-axis.



Read/write rates totals [VFC]

7.11.4.2.10 Read/write rates totals with utilization [VFC]

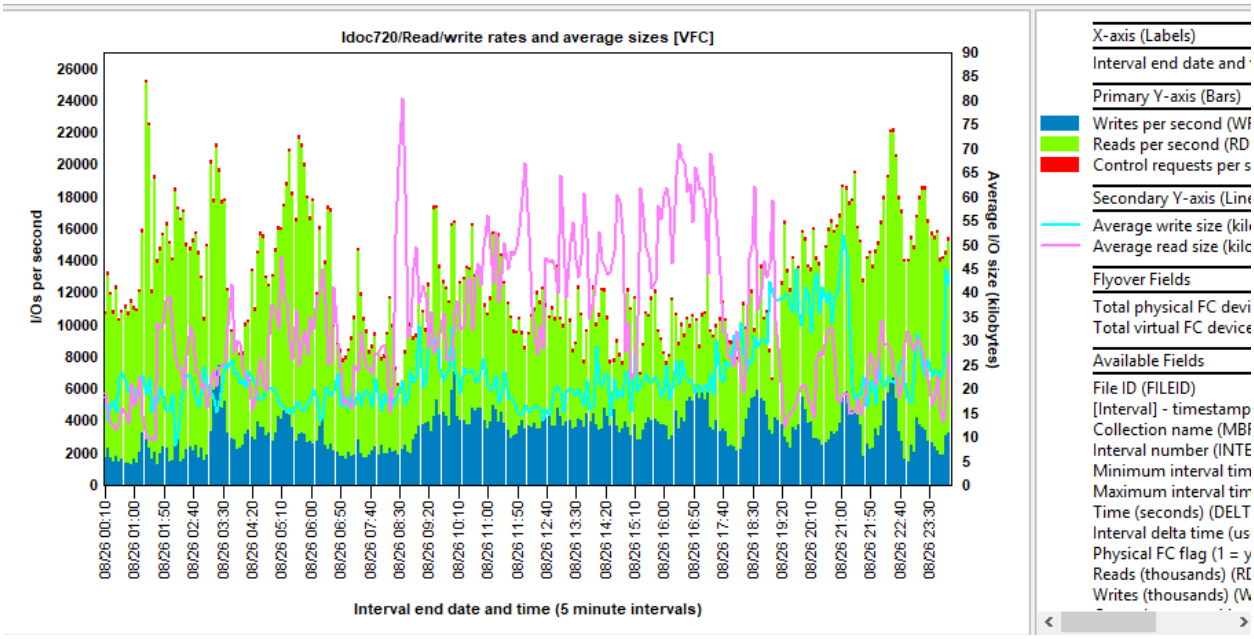
This is the same as the prior graph but includes utilizations on the Y2-axis.



Read/write rates totals with utilization [VFC]

7.11.4.2.11 Read/write rates and average sizes [VFC]

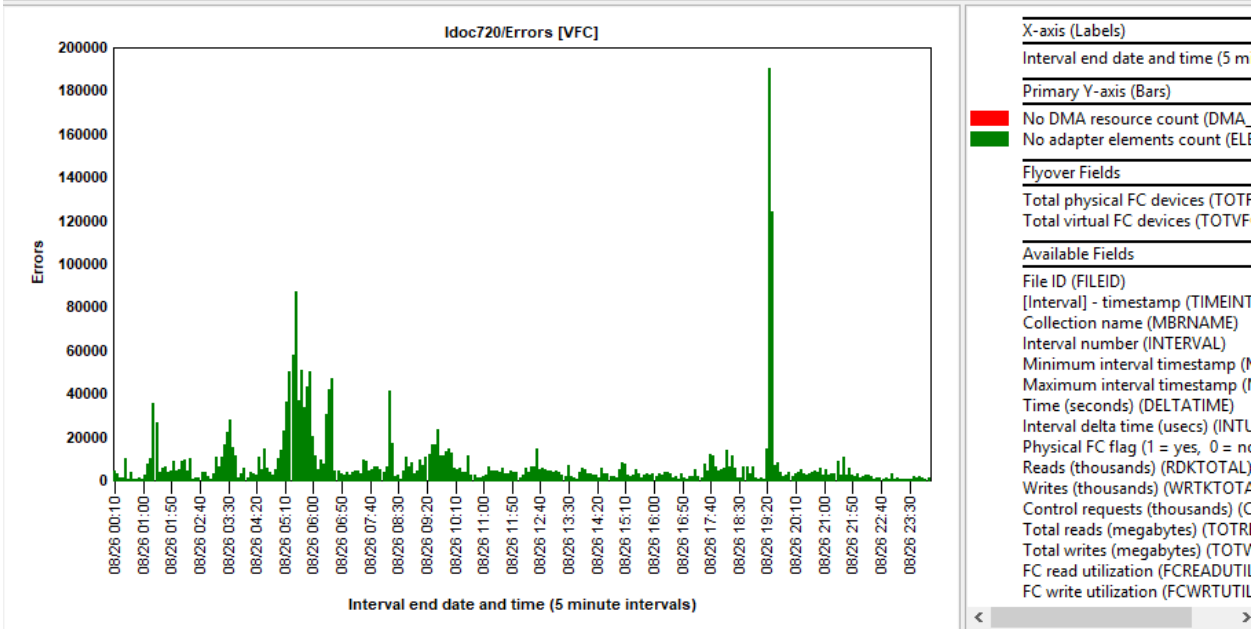
This graph displays the read, write and control requests rates per second with the average I/O sizes on the Y2-axis.



Read/write rates and average sizes [VFC]

7.11.4.2.12 Errors [VFC]

Displays the error counts over time for all virtual fiber channels.



Errors [VFC]

7.11.4.2.13 Adapters reset [VFC]

This graph displays the total adapters reset over time. The Y2-axis displays utilization.

7.11.4.2.14 FC/IP read/write counts totals [VFC]

The FC/IP graphs show fiber channel over IP statistics and are the same as the graphs already covered with similar names above.

7.11.4.2.15 FC/IP read/write size totals [VFC]

7.11.4.2.16 FC/IP read/write size averages [VFC]

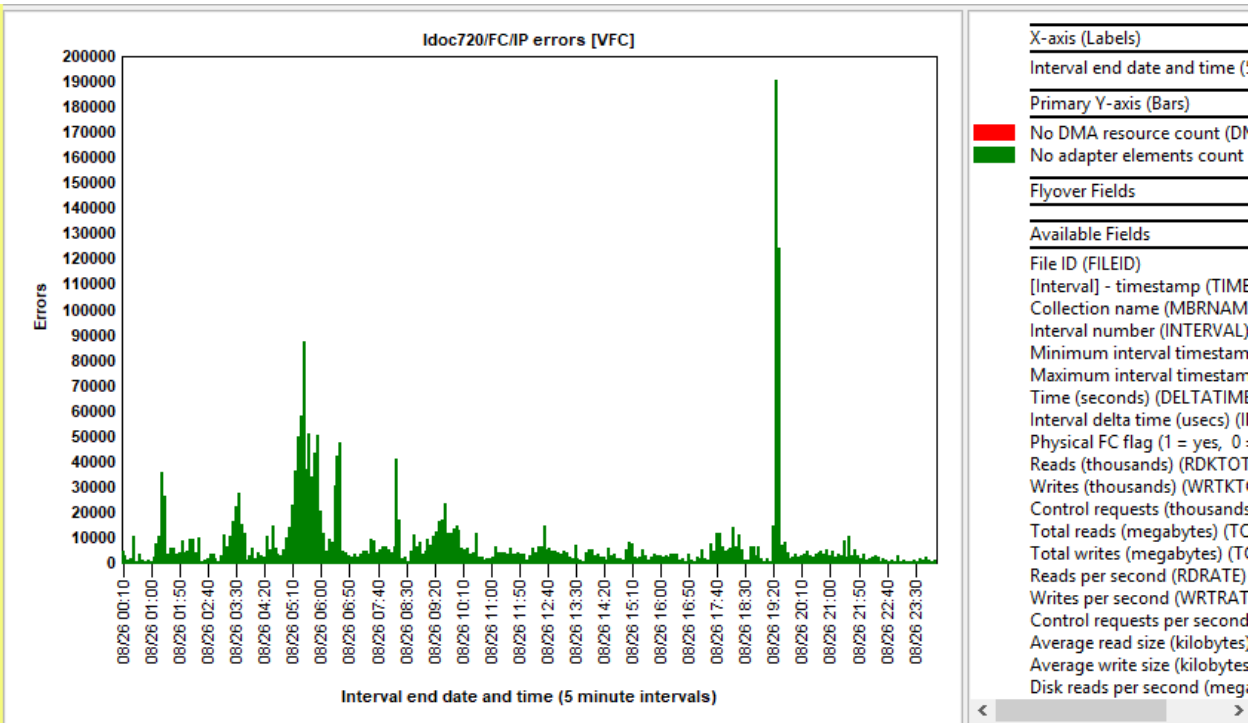
7.11.4.2.17 FC/IP read/write size rates [VFC]

7.11.4.2.18 FC/IP read/write rates totals [VFC]

7.11.4.2.19 FC/IP read/write rates and average sizes [VFC]

7.11.4.2.20 FC/IP errors [VFC]

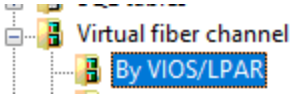
This graph shows the Fiber channel over IP errors that occurred across all virtual fiber channels.



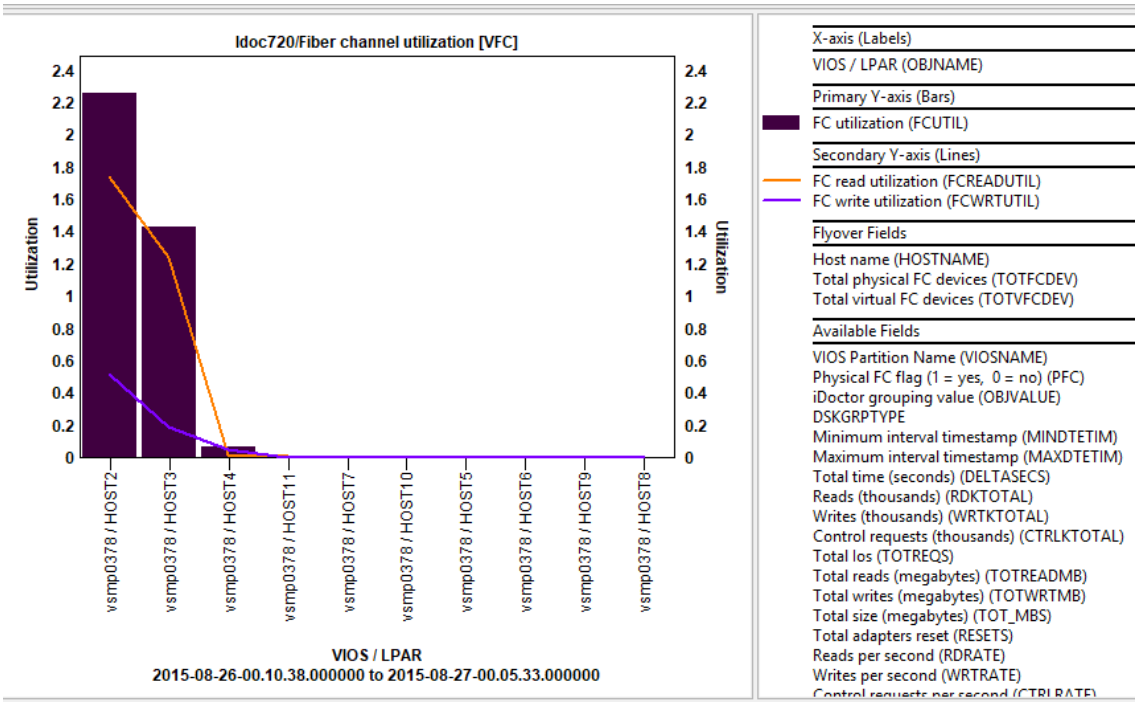
FC/IP errors [VFC]

7.11.4.2.21 By VIOS/LPAR

These graphs rank the virtual fiber channel metrics by VIOS/LPAR combination.

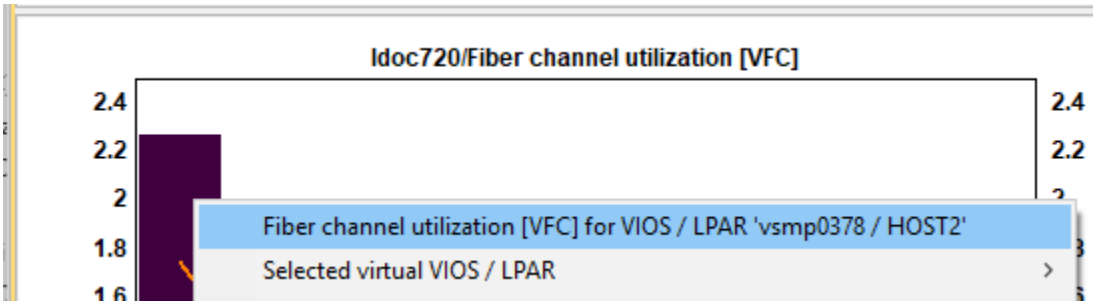


An example of this type of graph is:

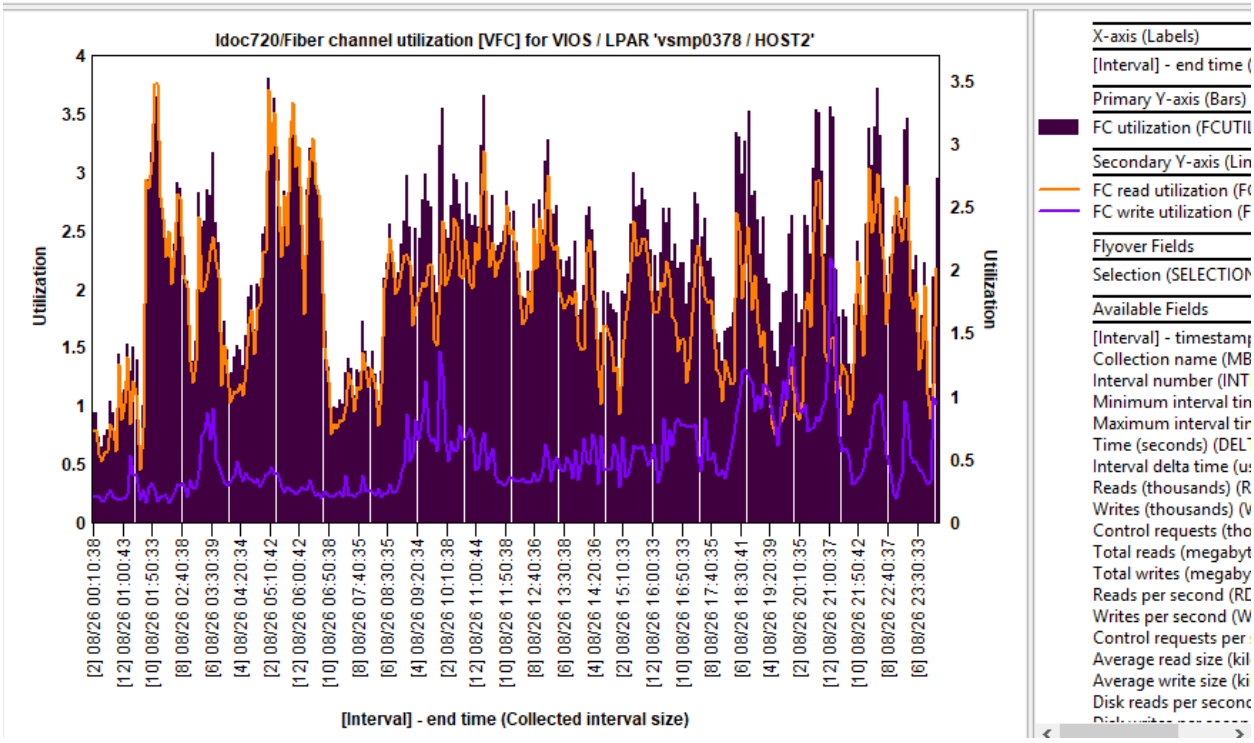


Virtual fiber channel -> By VIOS/LPAR -> Fiber channel utilization [VFC]

From this type of graph, you can right-click…



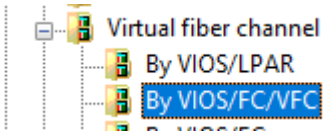
and pick the 1st option to graph the desired VIOS/LPAR virtual fiber channel stats over time.



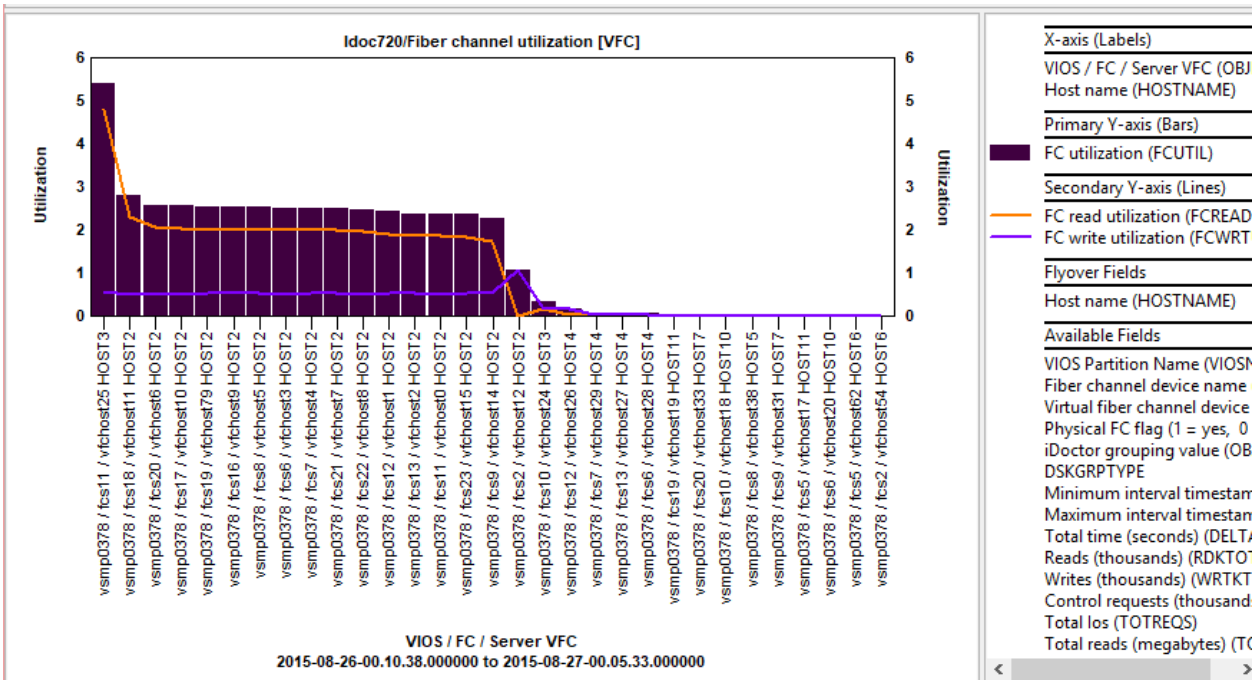
Fiber channel utilization [VFC] for VIOS / LPAR 'vsmp0378 / HOST2'

7.11.4.2.22 By VIOS/FC/VFC

These graphs rank the virtual fiber channel metrics by VIOS/FC (adapter)/VFC combination.



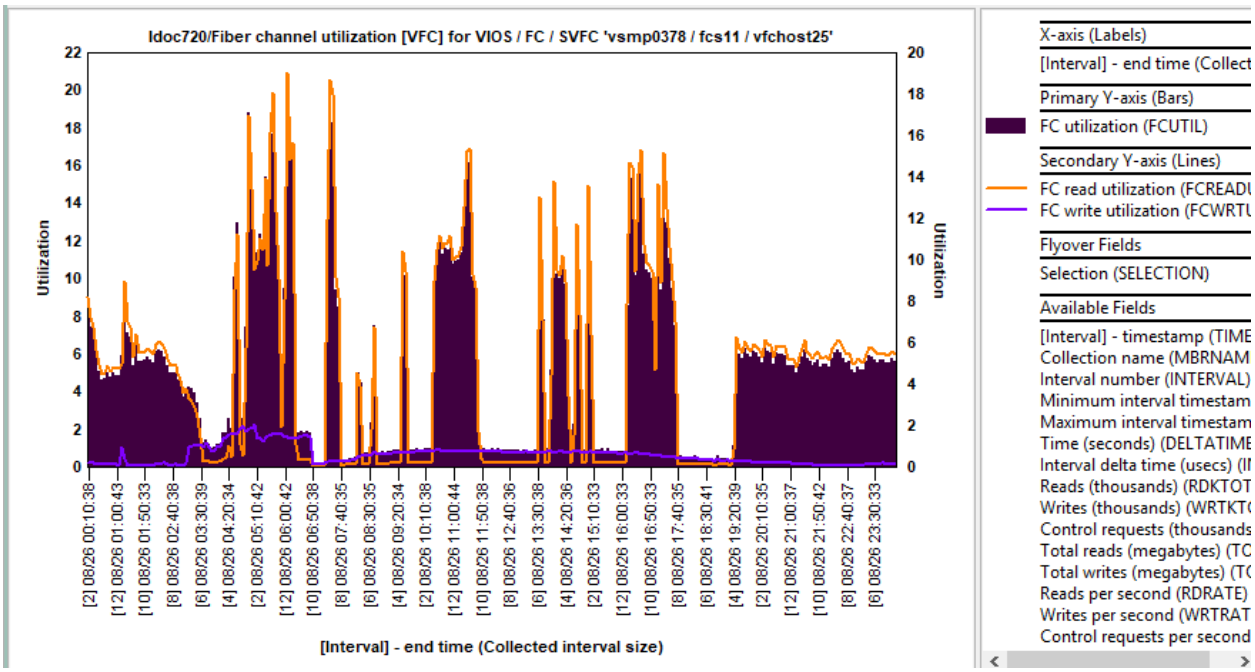
An example of this type of graph is:



Virtual fiber channel -> By VIOS/FC/Server VFC -> Fiber channel utilization [VFC]

Note: The HOSTNAME field will only be listed on the graph's X-axis if this combination of VIOS/FC/Server VFC is being served by only 1 host. If it serves multiple hosts/LPARs then it will be blank.

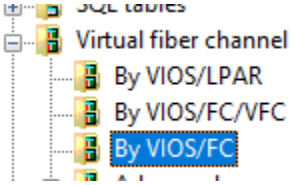
Clicking the 1st option in the graph above and drilling down would result in this graph over time for the selection:



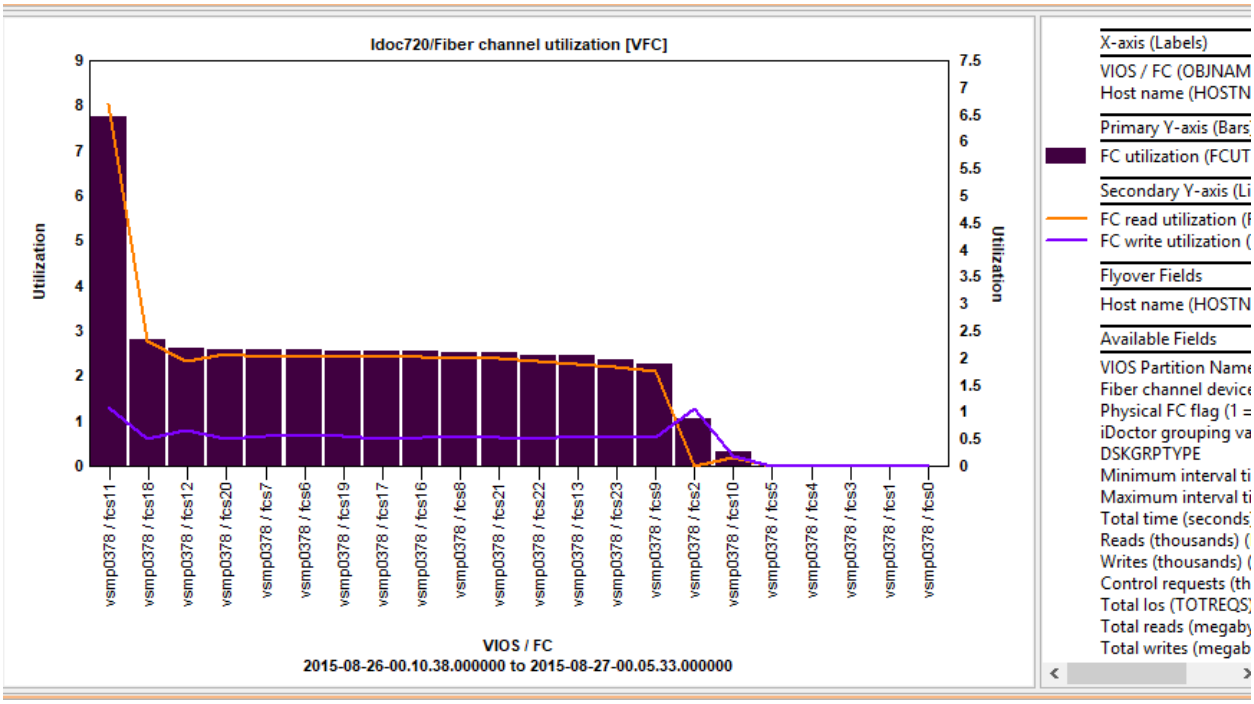
Fiber channel utilization [VFC] for VIOS / FC /SVFC 'vsmp0378 / fcs11 / vfchost25'

7.11.4.2.23 By VIOS/FC

These graphs rank the virtual fiber channel metrics by VIOS/FC combination.



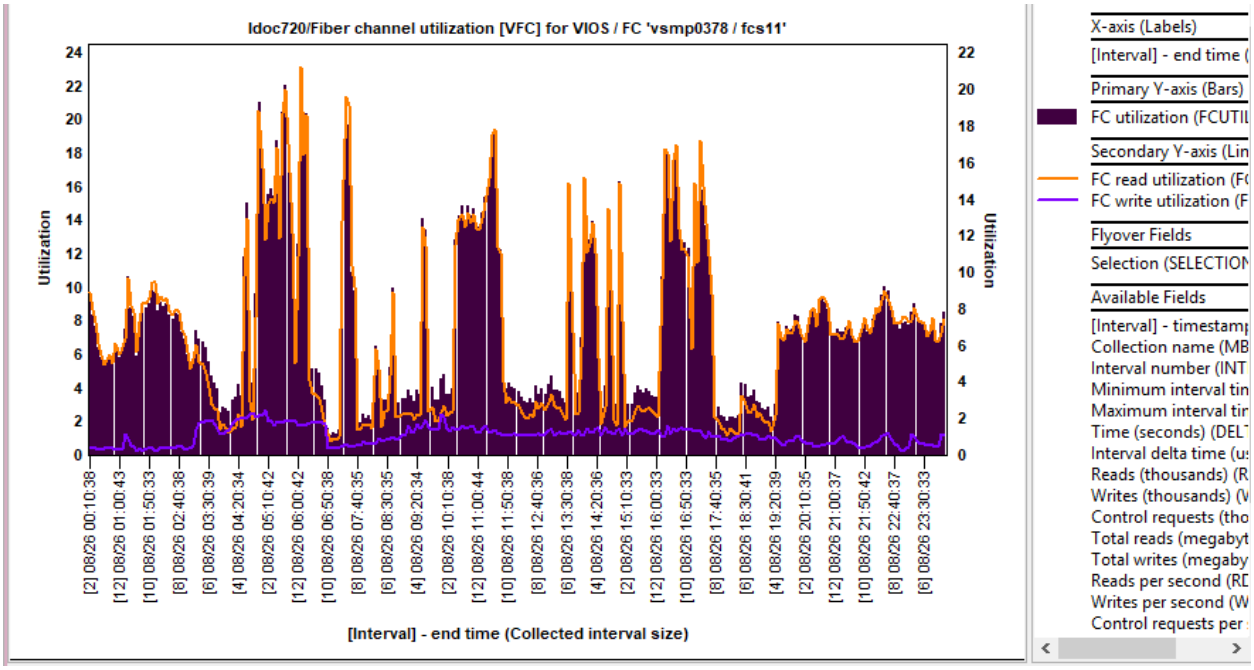
An example of this type of graph is:



Virtual fiber channel -> By VIOS/FC -> Fiber channel utilization [VFC]

Note: The HOSTNAME field will only be listed on the graph's X-axis if this combination of VIOS/FC is being served by only 1 host. If it serves multiple hosts/LPARs then it will be blank.

Clicking the 1st option in the graph above and drilling down would result in this graph over time for the selection:

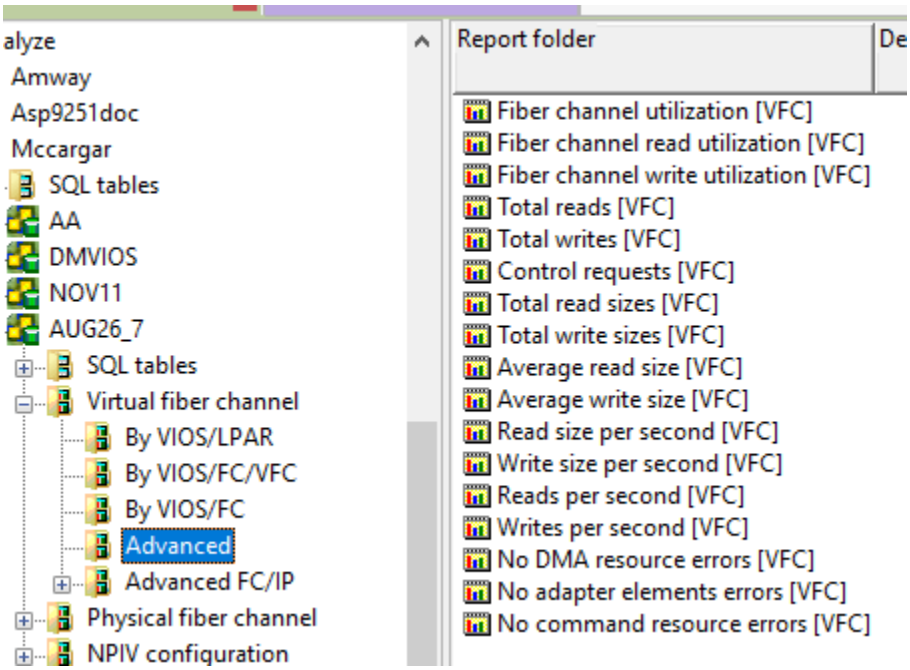


Fiber channel utilization [VFC] for VIOS / FC 'vsmp0378 / fcs11'

7.11.4.2.24 Advanced

The advanced graphs will graph each individual fiber channel adapter separately over time. Keep in mind they will only work well for a limited number of adapters (typically less than 20 works best.)

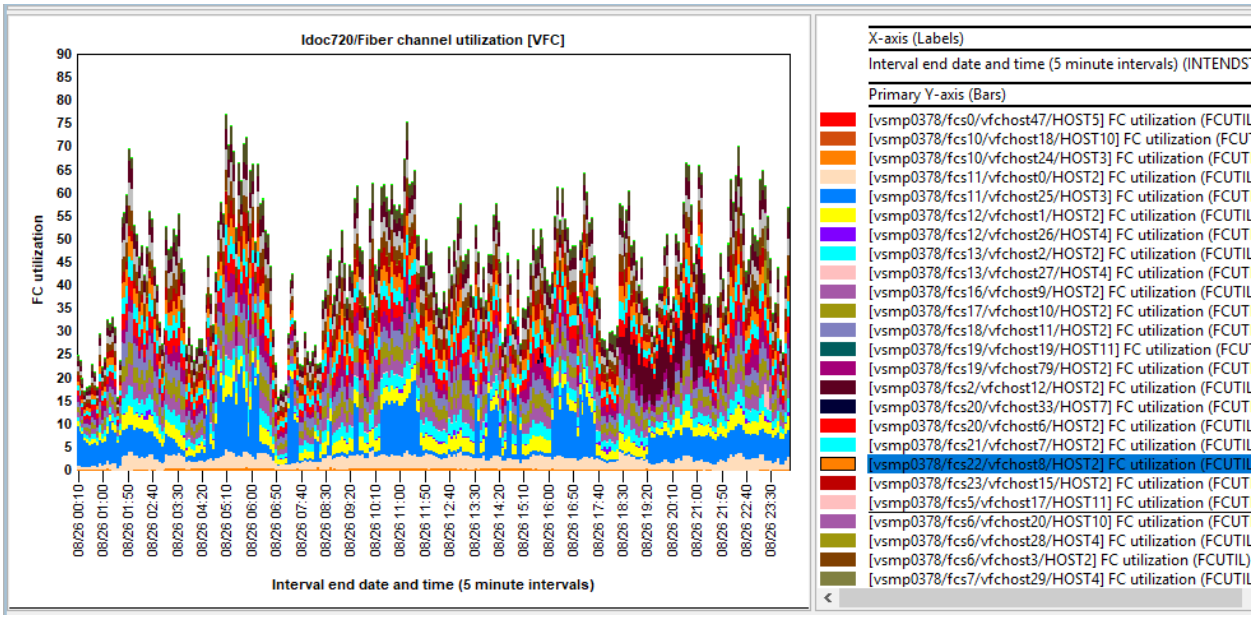
Tip: If you have hundreds of adapters it will be too slow and too hard to analyze using these graphs.



NPIV -> Virtual fiber channel -> Advanced folder

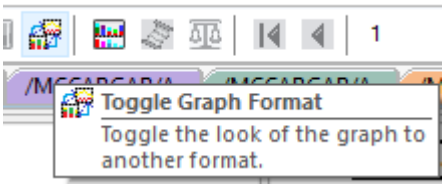
7.11.4.2.24.1 Fiber channel utilization [VFC]

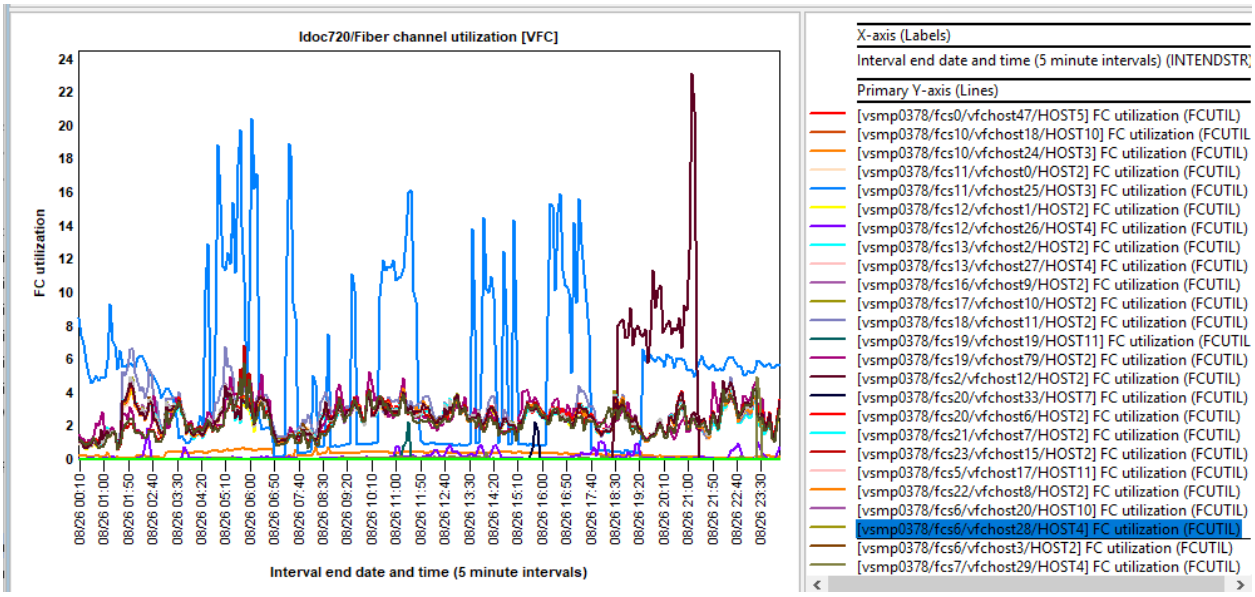
This graph displays each fiber channel's individual utilization together over time.



NPIV -> Virtual fiber channel -> Advanced -> Fiber channel utilization [VFC]

Tip: Press the Toggle graph format button on the toolbar to convert this graph to a line chart instead if desired.





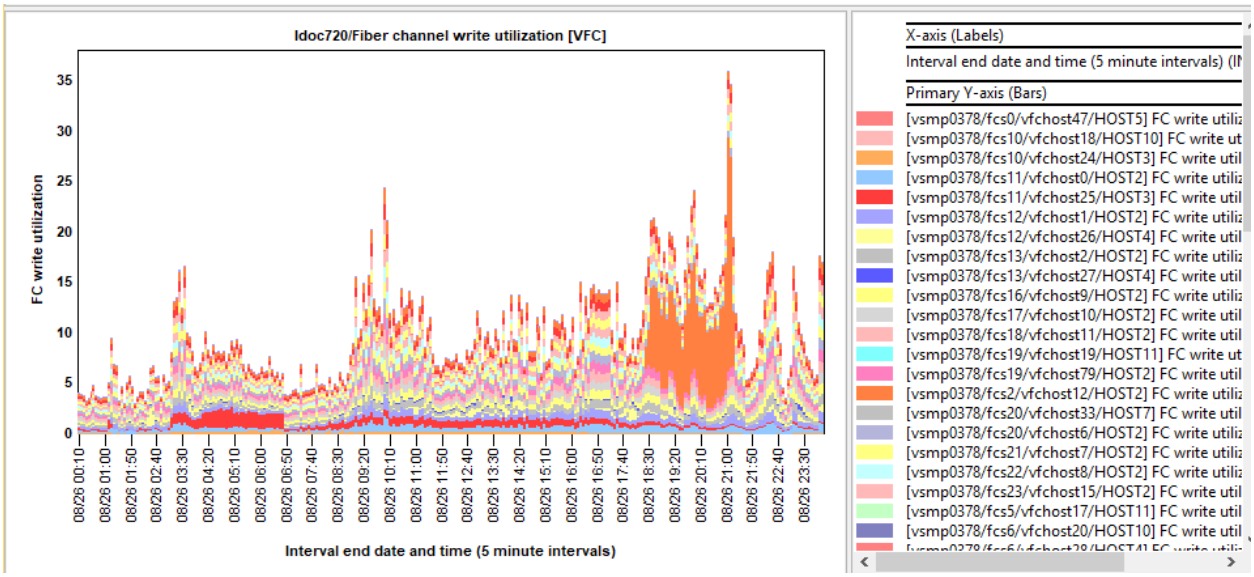
NPIV -> Virtual fiber channel -> Advanced -> Fiber channel utilization [VFC] as a line graph

7.11.4.2.24.2 Fiber channel read utilization [VFC]

This graph displays each virtual fiber channel's individual read utilization together over time.

7.11.4.2.24.3 Fiber channel write utilization [VFC]

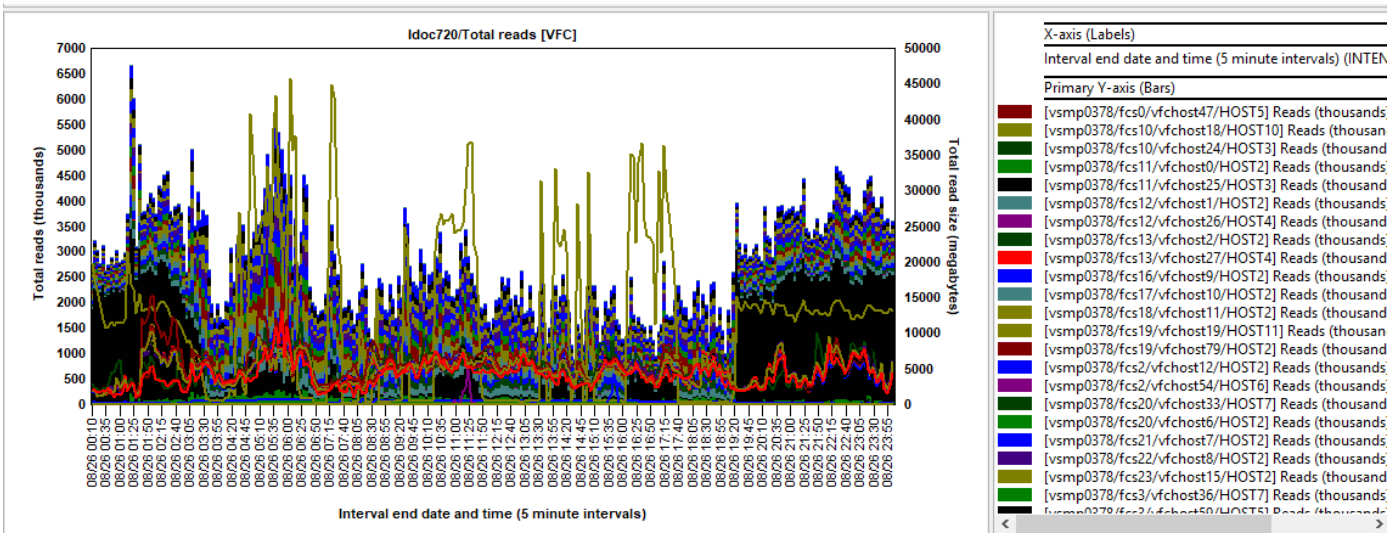
This graph displays each virtual fiber channel's individual write utilization together over time.



NPIV -> Virtual fiber channel -> Advanced -> Fiber channel write utilization [VFC]

7.11.4.2.24.4 Total reads [VFC]

This graph displays the total reads for all virtual fiber channels individually over time. The Y2-axis displays the total read size (in MBs.)



NPIV -> Virtual fiber channel -> Advanced -> Total reads [VFC]

7.11.4.2.24.5 Total writes [VFC]

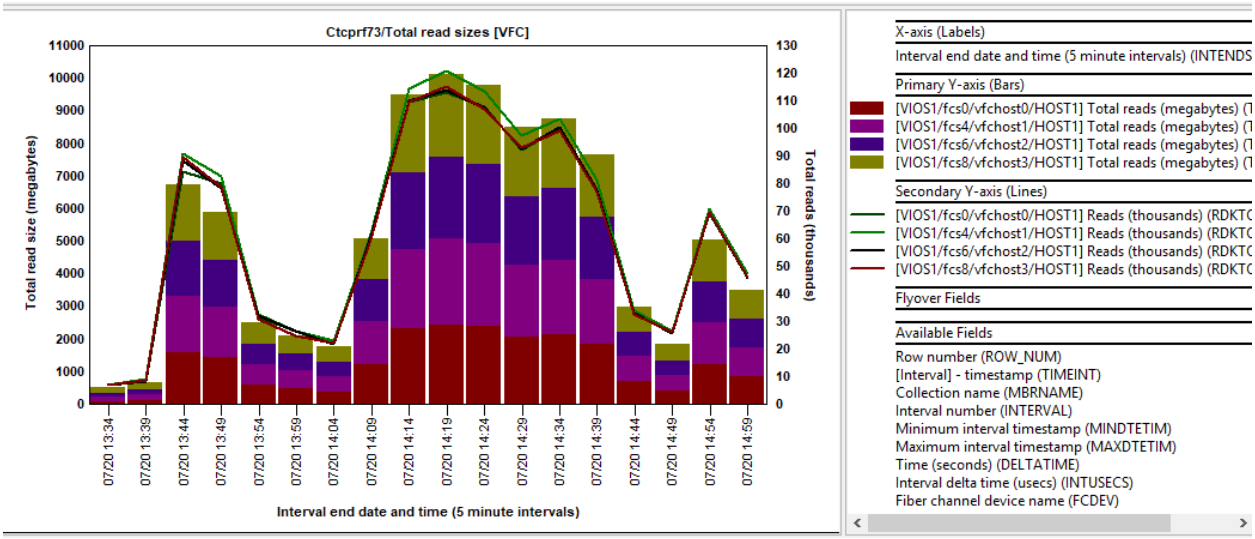
This graph displays the total writes for all virtual fiber channels individually over time. The Y2-axis displays the total write size (in MBs.)

7.11.4.2.24.6 Control requests [VFC]

This graph displays the control requests for all virtual fiber channels individually over time.

7.11.4.2.24.7 Total read sizes [VFC]

This graph displays the total read size (in MBs) for all virtual fiber channels individually over time. The Y2-axis displays the total reads (in thousands.)



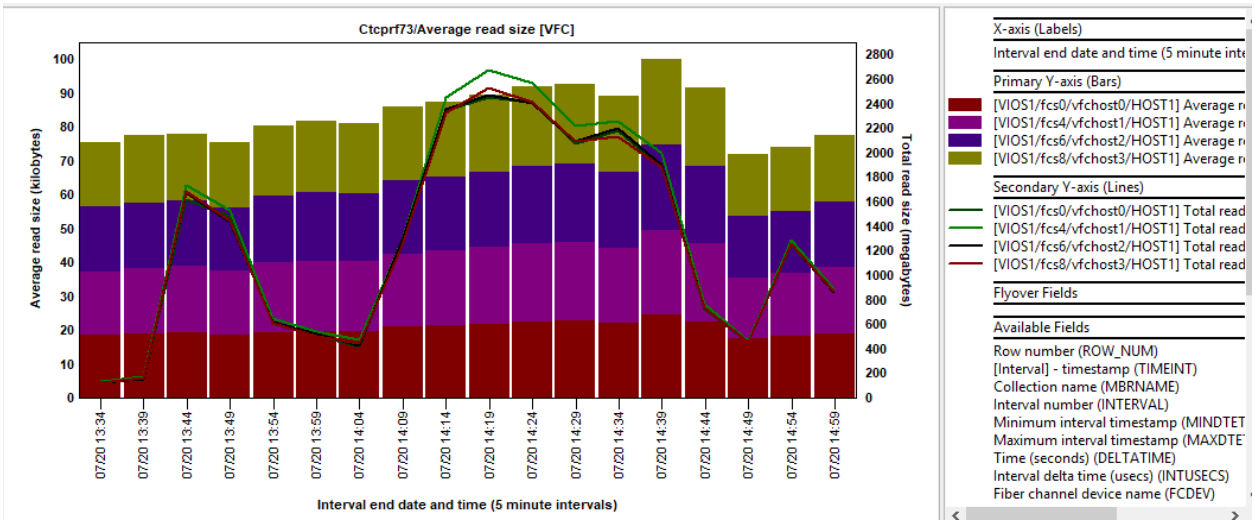
NPIV -> Virtual fiber channel -> Advanced -> Total reads sizes [VFC]

7.11.4.2.24.8 Total write sizes [VFC]

This graph displays the total write size (in MBs) for all virtual fiber channels individually over time. The Y2-axis displays the total writes (in thousands.)

7.11.4.2.24.9 Average read size [VFC]

This graph displays the average size for each read operation (in KBs) for all virtual fiber channels individually over time. The Y2-axis displays the total reads (in thousands.)



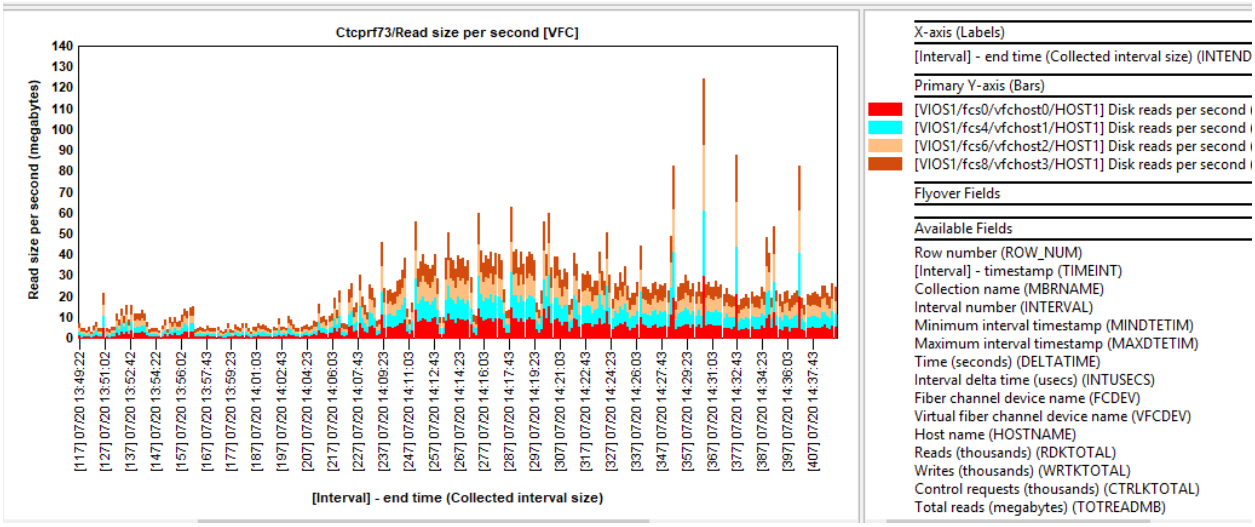
NPIV -> Virtual fiber channel -> Advanced -> Average read size [VFC]

7.11.4.2.24.10 Average write size [VFC]

This graph displays the average size for each write operation (in KBs) for all virtual fiber channels individually over time. The Y2-axis displays the total writes (in thousands.)

7.11.4.2.24.11 Read size per second [VFC]

This graph displays the total read size per second (in MBs) for all virtual fiber channels individually over time.



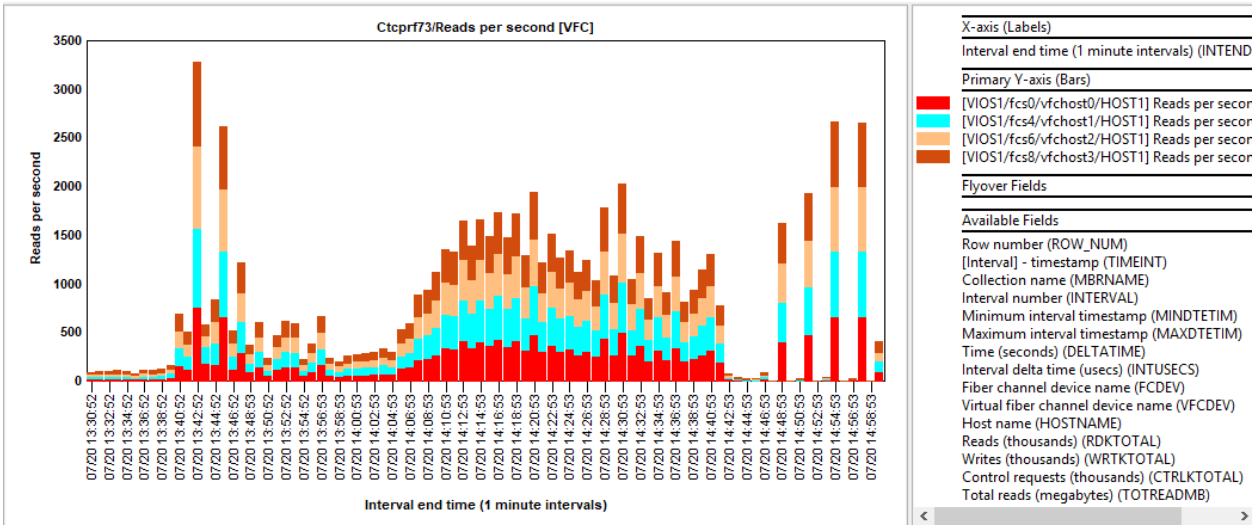
NPIV -> Virtual fiber channel -> Advanced -> Read size per second [VFC]

7.11.4.2.24.12 Write size per second [VFC]

This graph displays the total write size per second (in MBs) for all virtual fiber channels individually over time.

7.11.4.2.24.13 Reads per second [VFC]

This graph displays the total reads per second for all virtual fiber channels individually over time.



NPIV -> Virtual fiber channel -> Advanced -> Reads per second [VFC]

7.11.4.2.24.14 Writes per second [VFC]

This graph displays the total writes per second for all virtual fiber channels individually over time.

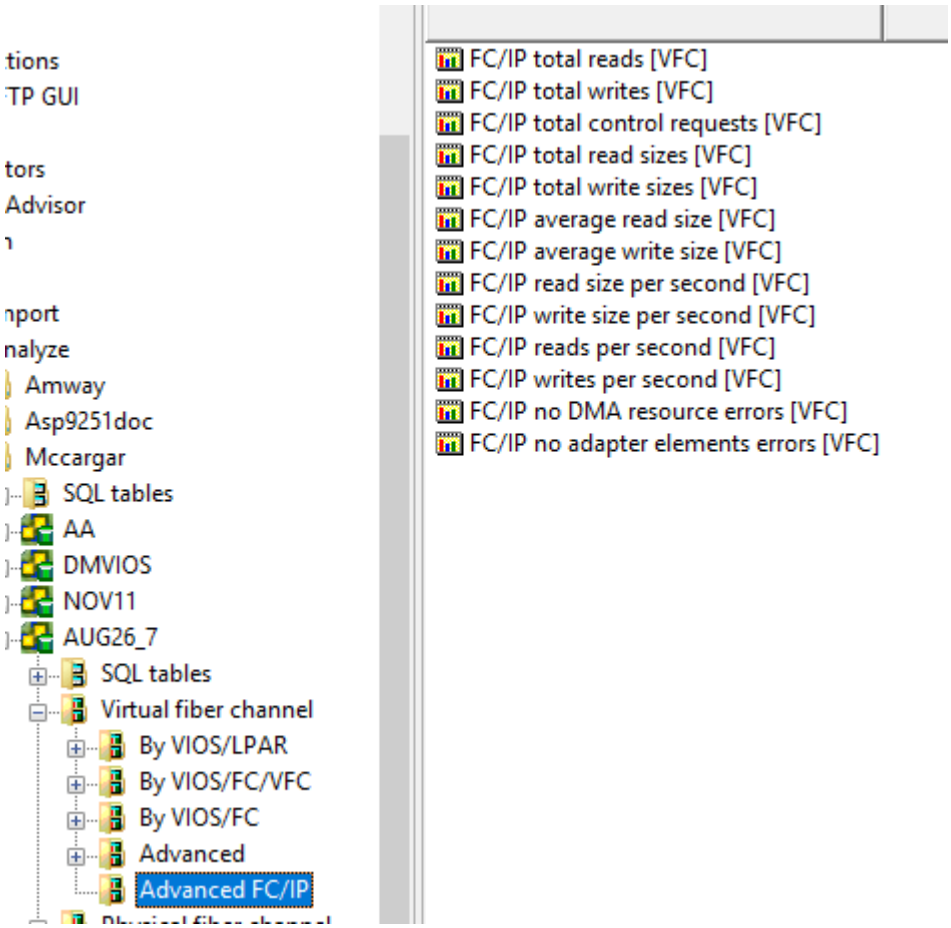
7.11.4.2.24.15 Resource error graphs

These graphs are provided to show various types of virtual fiber channel errors individually over time:

- No DMA resource errors [VFC]
- No adapter elements errors [VFC]
- No command resource errors [VFC]

7.11.4.2.25 Advanced FC/IP

This section contains the set of graphs specific to virtual fiber channel over IP configurations.



Virtual fiber channel -> Advanced FC/IP folder

They are advanced graphs and they show individual fiber channels together over time. These metrics were already covered in the previous section so look there for more information if needed.

The set of graphs available are:

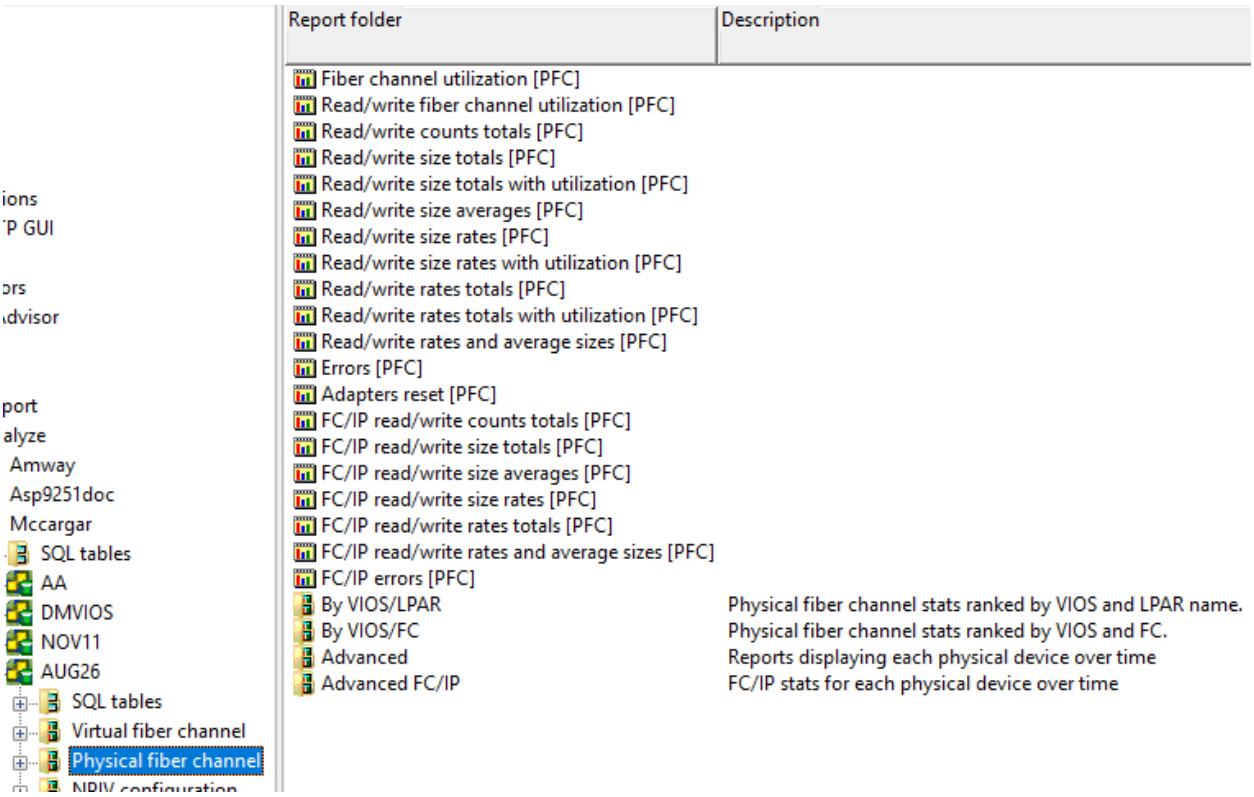
- FC/IP total reads [VFC]
- FC/IP total writes [VFC]
- FC/IP total control requests [VFC]
- FC/IP total read sizes [VFC]
- FC/IP total write sizes [VFC]
- FC/IP average read size [VFC]
- FC/IP average write size [VFC]
- FC/IP read size per second [VFC]

- FC/IP write size per second [VFC]
- FC/IP reads per second [VFC]
- FC/IP writes per second [VFC]
- FC/IP no DMA resource errors [VFC]
- FC/IP no adapter elements errors [VFC]

7.11.4.3 Physical fiber channel

This set of graphs covers the **physical** fiber channel statistics captured using the FCSTAT command and the iDoctor scripts. These graphs will use an abbreviation of [PFC] to indicate they apply to Physical Fiber Channel adapters.

The options available are:



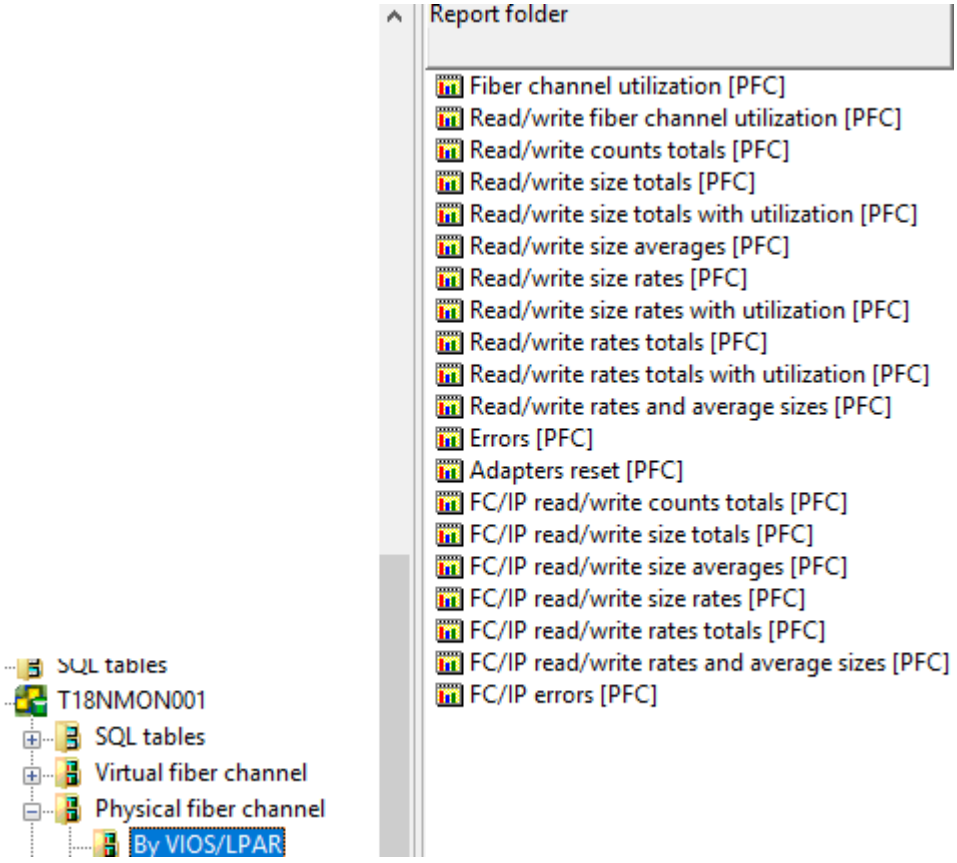
NPIV -> Physical fiber channel folder

The By VIOS… graphs are all rankings graphs. The Advanced graphs will graph each individual adapter together over time. They will not work well if you have too many adapters (> 20 or so) and you will need to use the ranking graphs instead or a combination of the overview charts and ranking charts (filtered by time period) to do your analysis.

Note: The overview graphs are identical to the ones in the Virtual fiber channel section (except they are based on the physical fiber channels) so are not covered here.

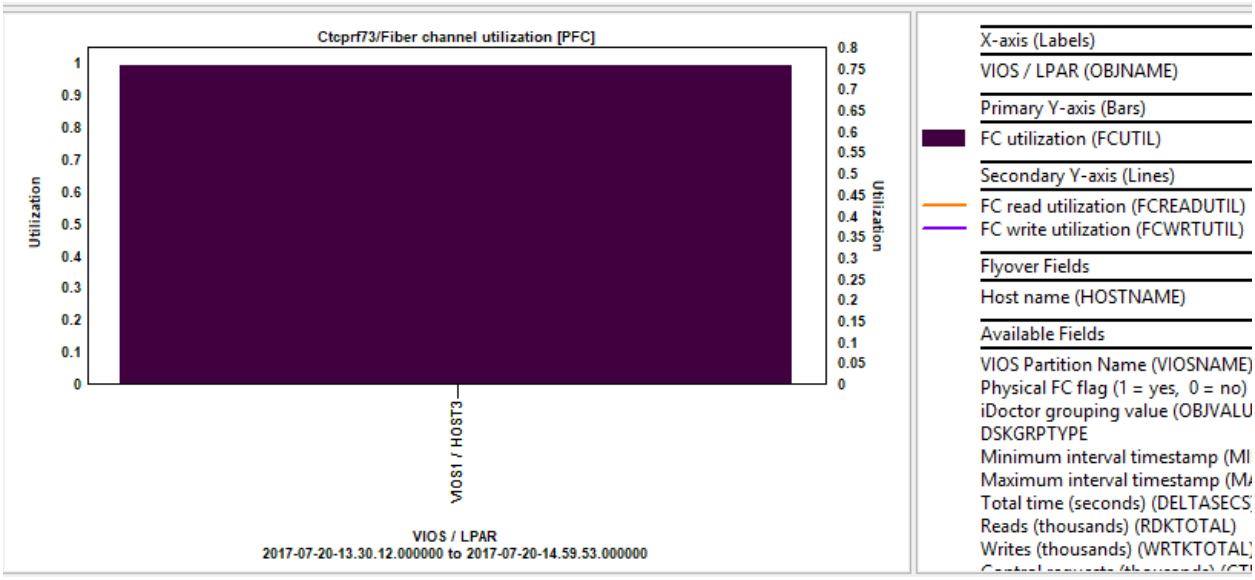
7.11.4.3.1 By VIOS/LPAR

These graphs rank the physical fiber channel metrics by VIOS/LPAR combination.



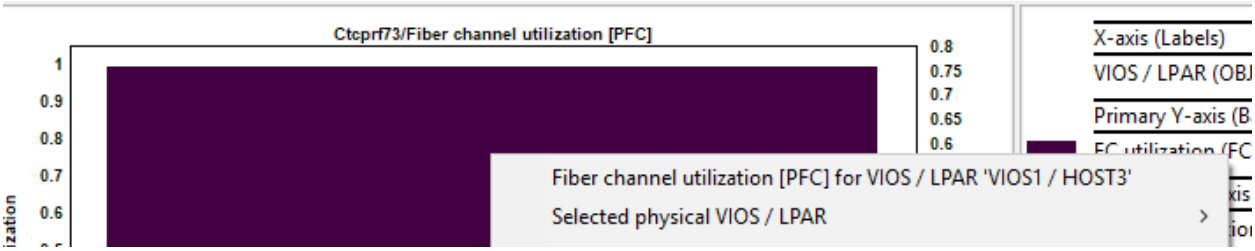
NPIV -> Physical fiber channel -> By VIOS/LPAR folder

An example of this type of graph is:

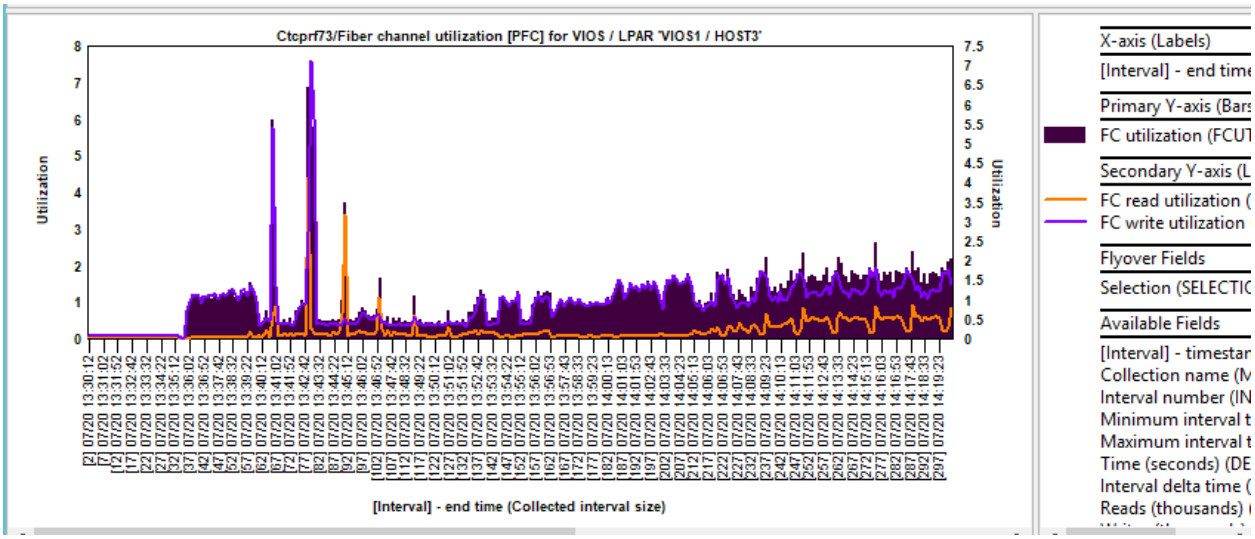


Physical fiber channel -> By VIOS/LPAR -> Fiber channel utilization [PFC]

From this type of graph, you can right-click…



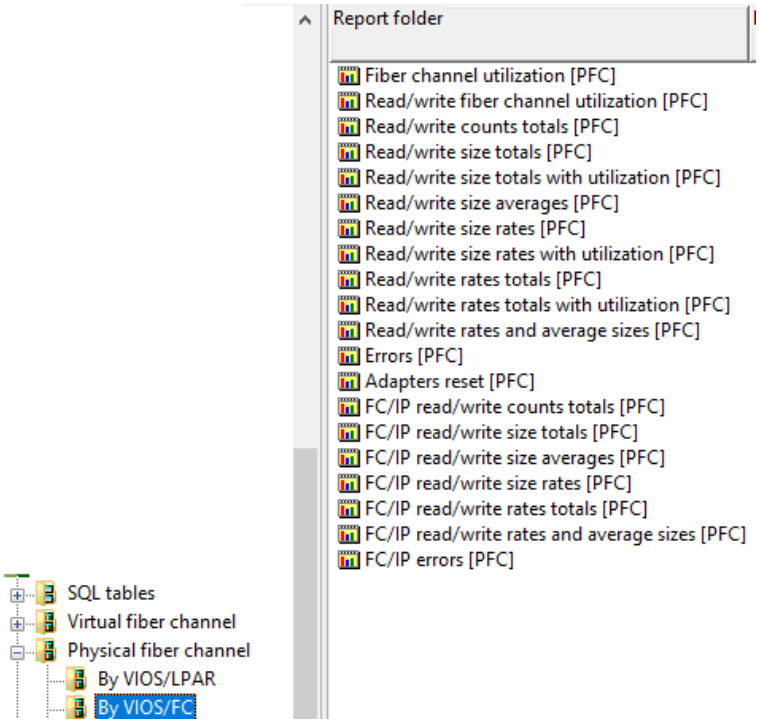
and pick the 1st option to graph the desired VIOS/LPAR physical fiber channel stats over time.



Fiber channel utilization [PFC] for VIOS / LPAR 'VIOS1 / HOST3'

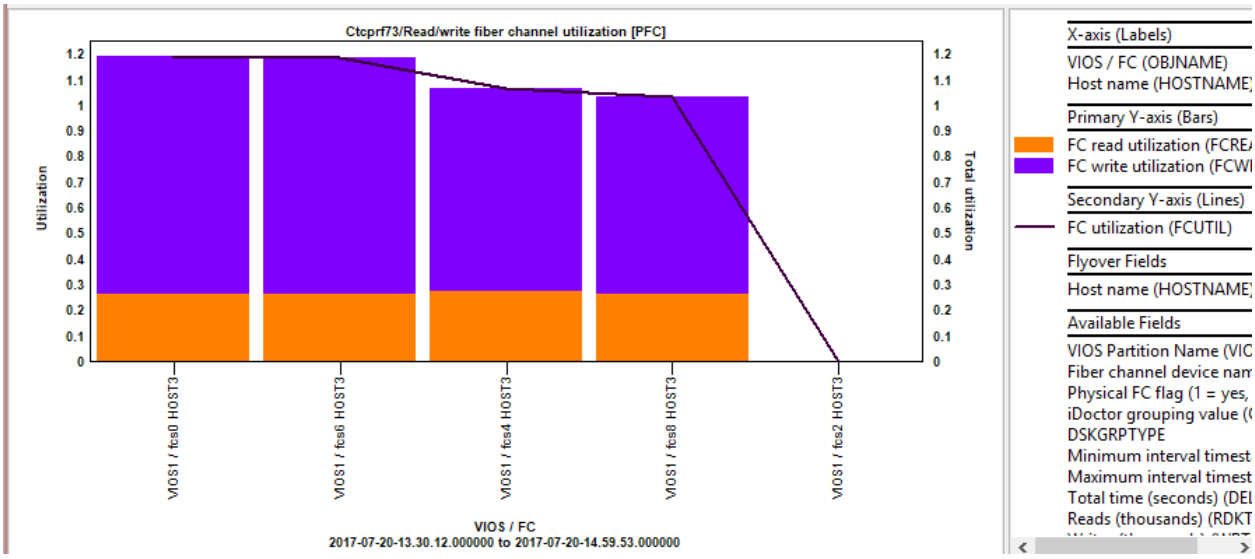
7.11.4.3.2 By VIOS/FC

These graphs rank the physical fiber channel metrics by VIOS and FC adapter combination.



NPIV -> Physical fiber channel -> By VIOS/FC folder

An example of this type of graph is:

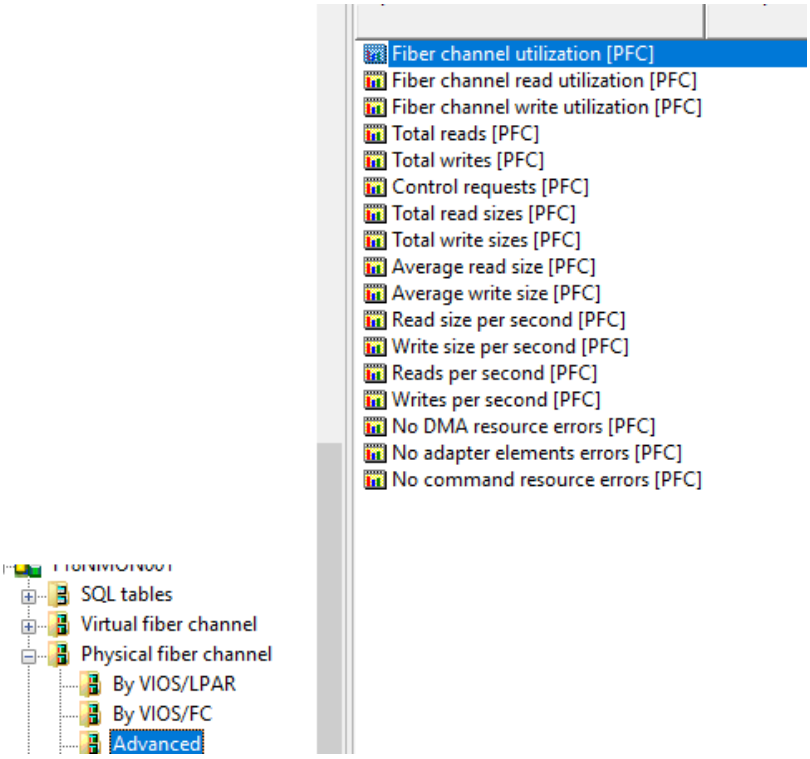


Physical fiber channel -> By VIOS/FC -> Read/write fiber channel utilization [PFC]

7.11.4.3.3 Advanced

The advanced graphs will graph each individual physical fiber channel adapter separately over time. Keep in mind they will only work well for a limited number of adapters (typically less than 20 works best.)

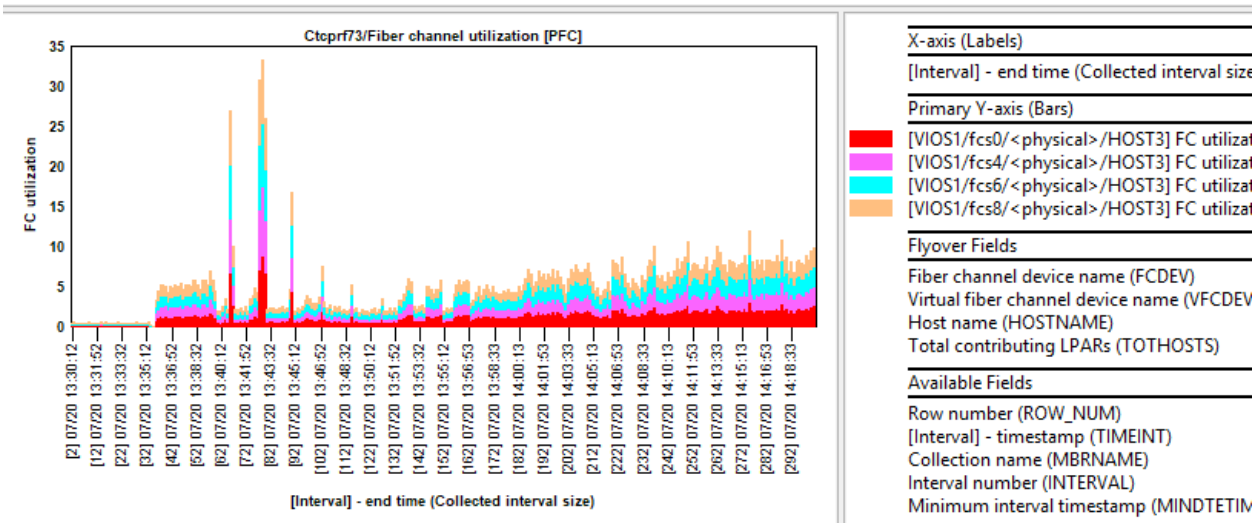
Tip: If you have hundreds of adapters it will be too slow and too hard to analyze using these graphs.



NPIV -> Physical fiber channel -> Advanced folder

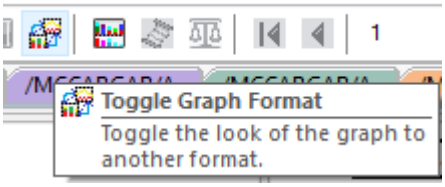
7.11.4.3.3.1 Fiber channel utilization [PFC]

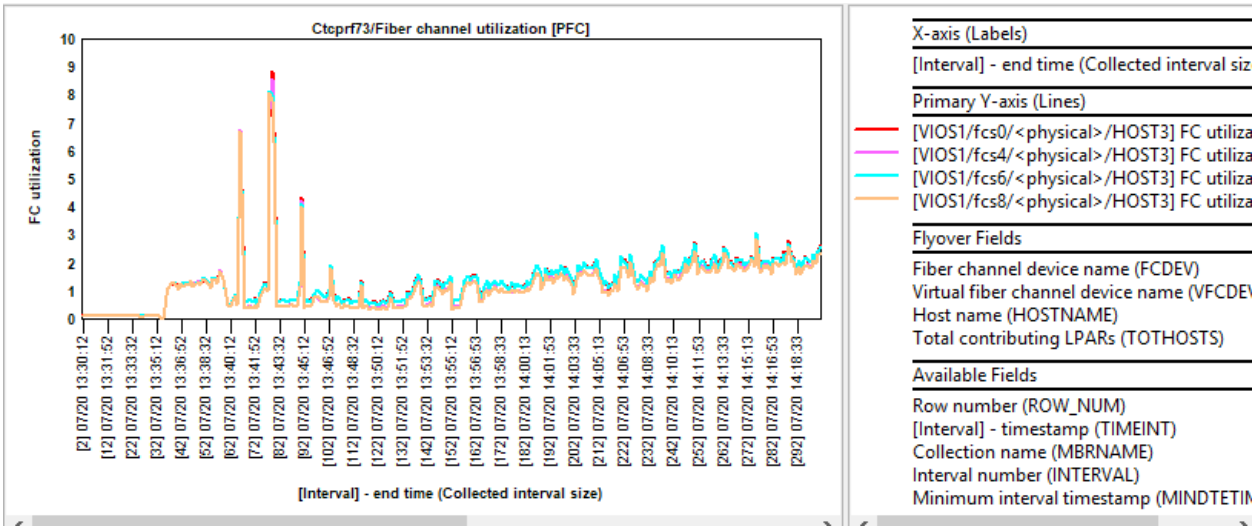
This graph displays each fiber channel's individual utilization together over time.



NPIV -> Physical fiber channel -> Advanced -> Fiber channel utilization [PFC]

Tip: Press the Toggle graph format button on the toolbar to convert this graph to a line chart instead if desired.





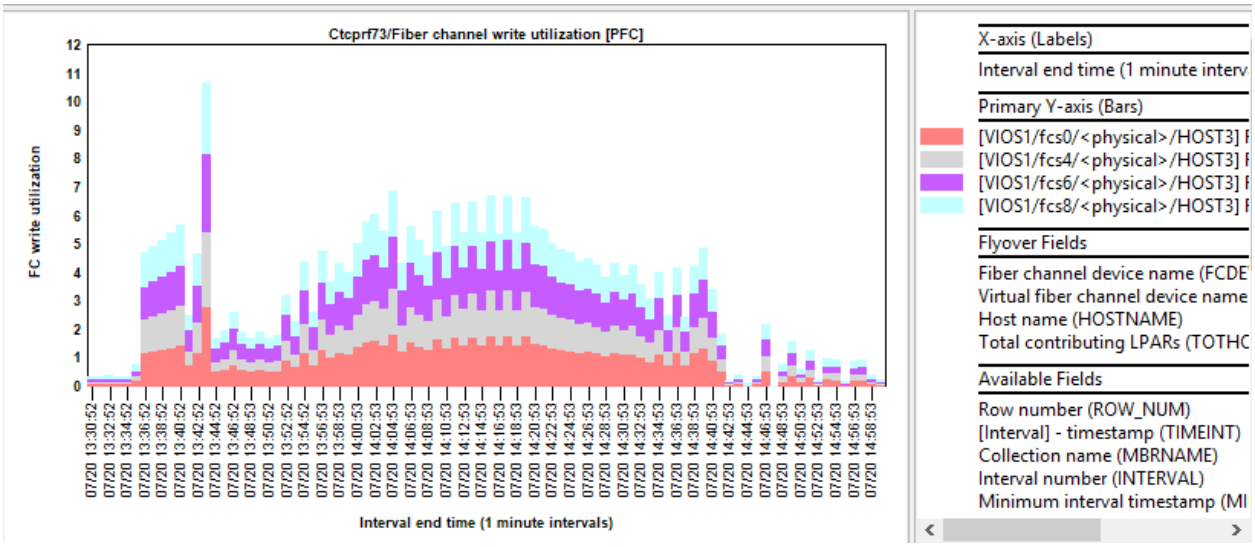
NPIV -> Physical fiber channel -> Advanced -> Fiber channel utilization [PFC] as a line graph

7.11.4.3.3.2 Fiber channel read utilization [PFC]

This graph displays each physical fiber channel's individual read utilization together over time.

7.11.4.3.3.3 Fiber channel write utilization [PFC]

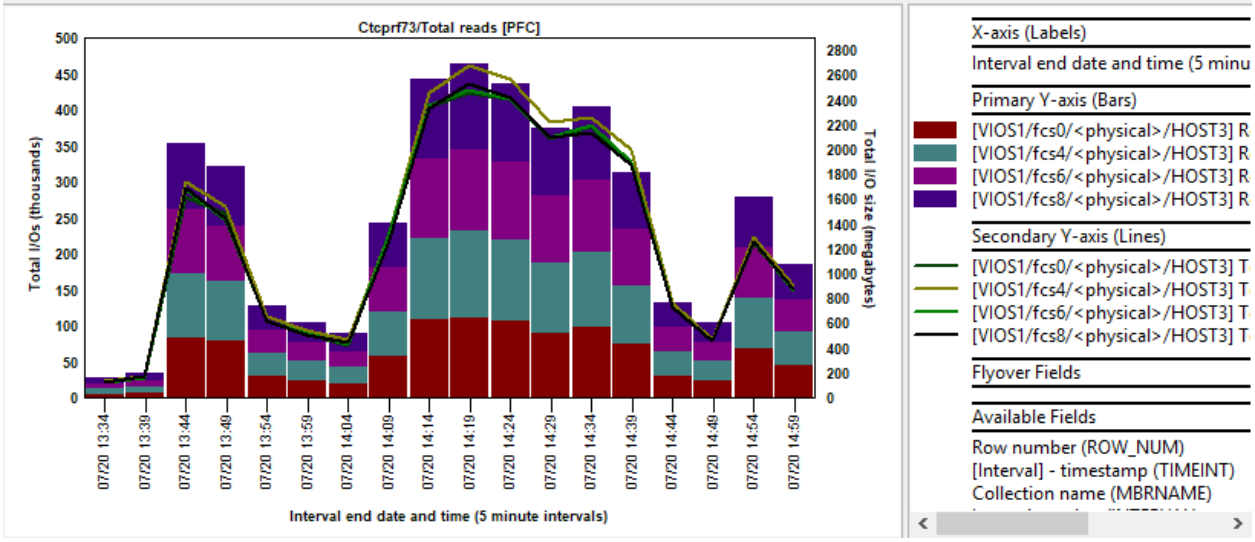
This graph displays each physical fiber channel's individual write utilization together over time.



NPIV -> Physical fiber channel -> Advanced -> Fiber channel write utilization [PFC]

7.11.4.3.3.4 Total reads [PFC]

This graph displays the total reads for all physical fiber channels individually over time. The Y2-axis displays the total read size (in MBs.)



NPIV -> Physical fiber channel -> Advanced -> Total reads [PFC]

7.11.4.3.3.5 Total writes [PFC]

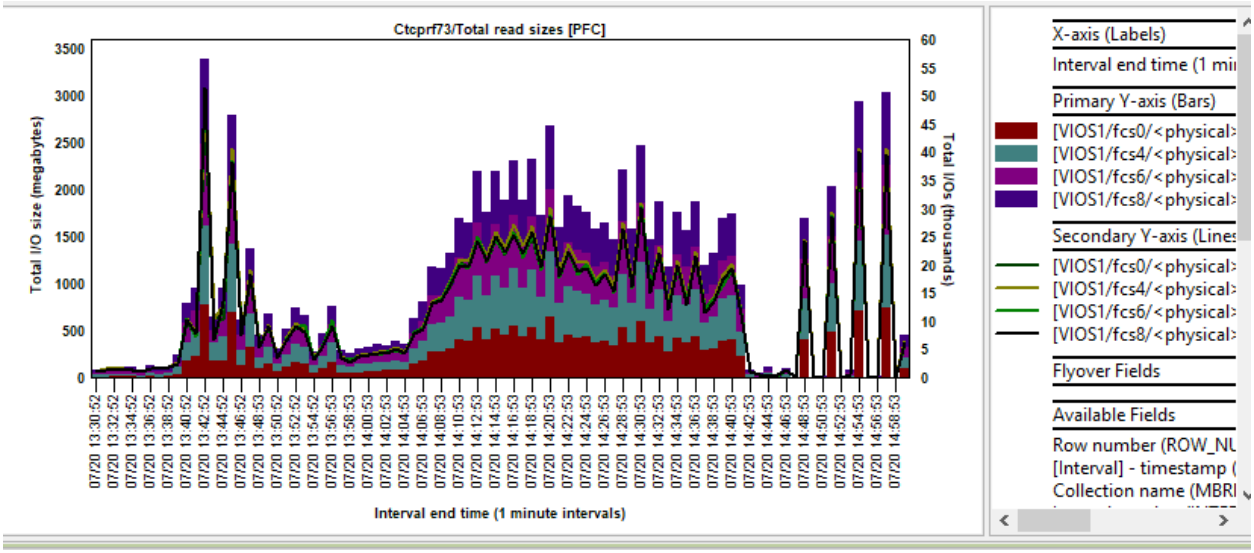
This graph displays the total writes for all physical fiber channels individually over time. The Y2-axis displays the total write size (in MBs.)

7.11.4.3.3.6 Control requests [PFC]

This graph displays the control requests for all physical fiber channels individually over time.

7.11.4.3.3.7 Total read sizes [PFC]

This graph displays the total read size (in MBs) for all physical fiber channels individually over time. The Y2-axis displays the total reads (in thousands.)



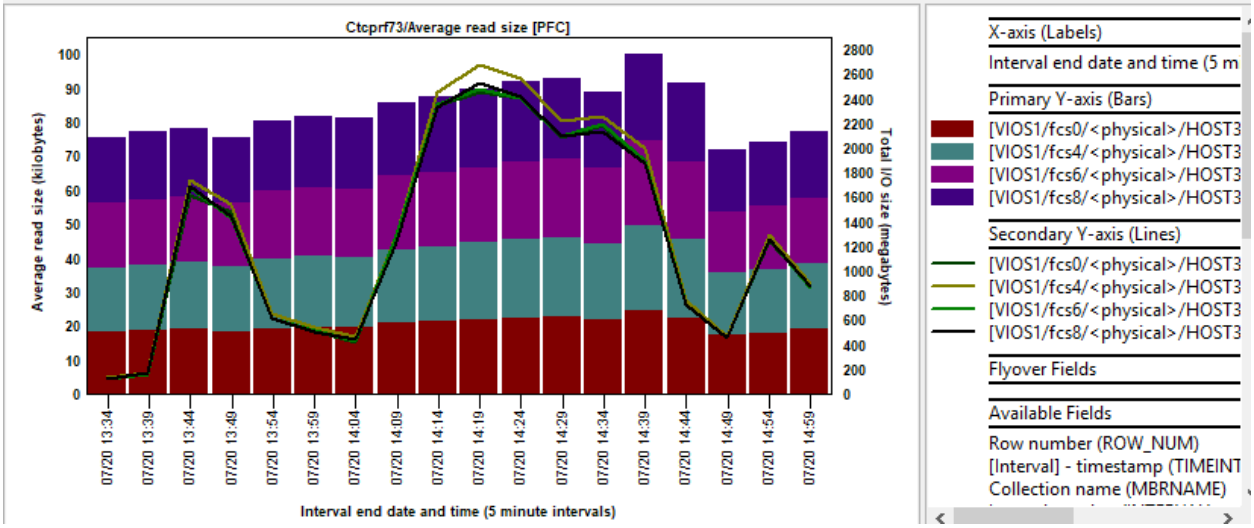
NPIV -> Physical fiber channel -> Advanced -> Total reads sizes [PFC]

7.11.4.3.3.8 Total write sizes [PFC]

This graph displays the total write size (in MBs) for all physical fiber channels individually over time. The Y2-axis displays the total writes (in thousands.)

7.11.4.3.3.9 Average read size [PFC]

This graph displays the average size for each read operation (in KBs) for all physical fiber channels individually over time. The Y2-axis displays the total reads (in thousands.)



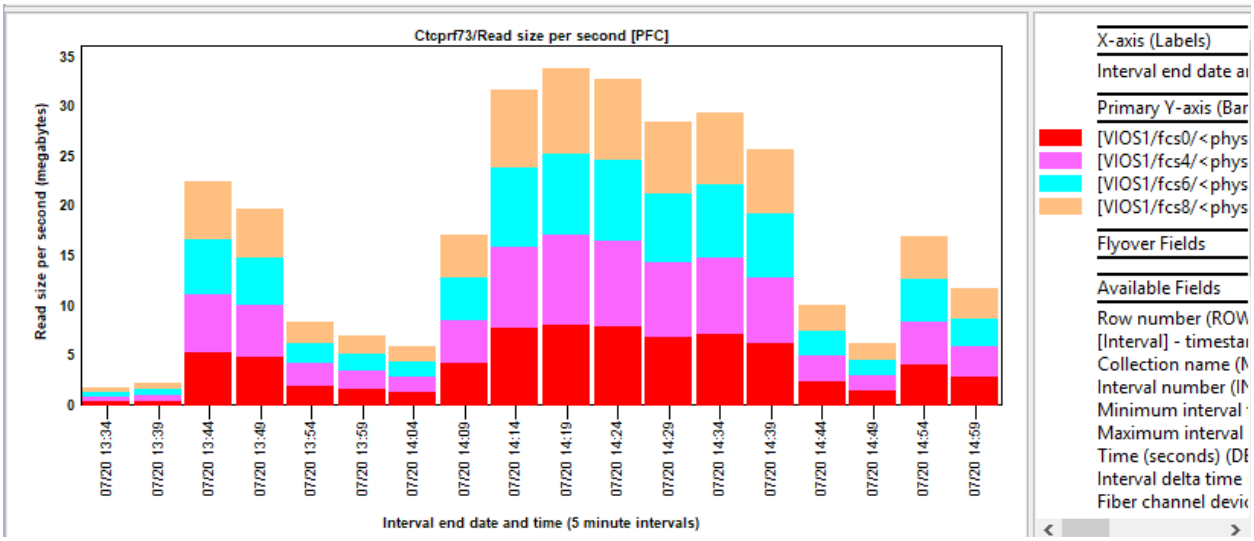
NPIV -> Physical fiber channel -> Advanced -> Average read size [PFC]

7.11.4.3.3.10 Average write size [PFC]

This graph displays the average size for each write operation (in KBs) for all physical fiber channels individually over time. The Y2-axis displays the total writes (in thousands.)

7.11.4.3.3.11 Read size per second [PFC]

This graph displays the total read size per second (in MBs) for all physical fiber channels individually over time.



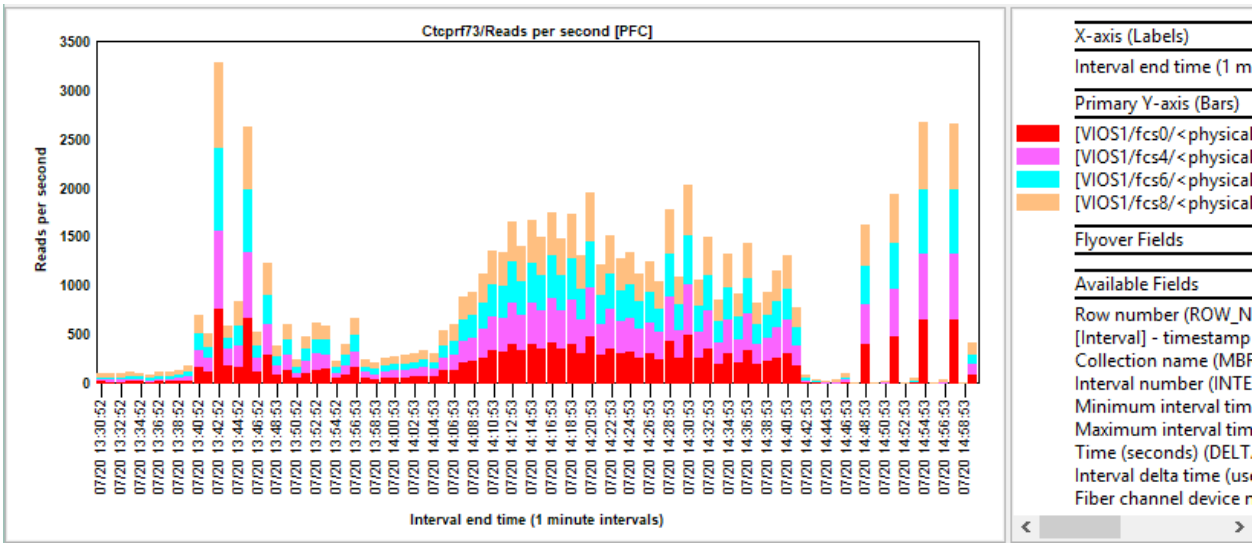
NPIV -> Physical fiber channel -> Advanced -> Read size per second [PFC]

7.11.4.3.3.12 Write size per second [PFC]

This graph displays the total write size per second (in MBs) for all physical fiber channels individually over time.

7.11.4.3.3.13 Reads per second [PFC]

This graph displays the total reads per second for all physical fiber channels individually over time.



NPIV -> Physical fiber channel -> Advanced -> Reads per second [PFC]

7.11.4.3.3.14 Writes per second [PFC]

This graph displays the total writes per second for all physical fiber channels individually over time.

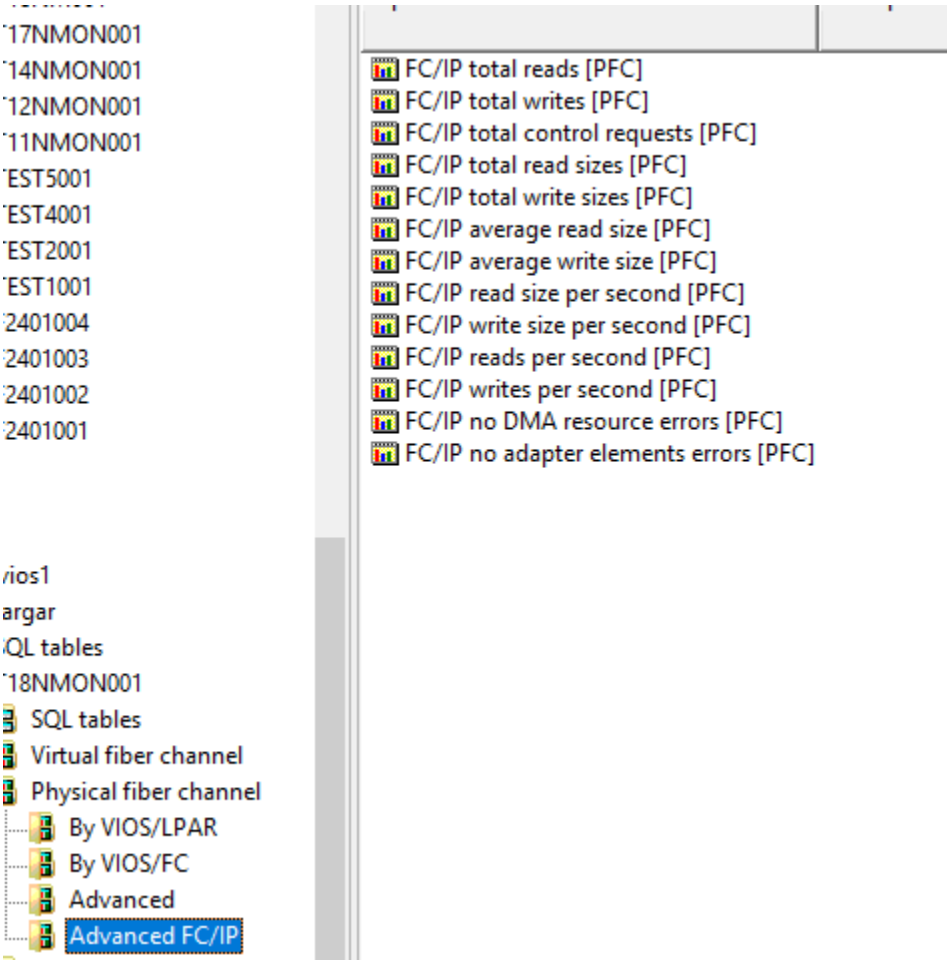
7.11.4.3.3.15 Resource error graphs

These graphs are provided to show various types of physical fiber channel errors individually over time:

- No DMA resource errors [PFC]
- No adapter elements errors [PFC]
- No command resource errors [PFC]

7.11.4.3.4 Advanced FC/IP

This section contains the set of graphs specific to physical fiber channel over IP configurations.



Physical fiber channel -> Advanced FC/IP folder

They are advanced graphs and they show individual physical fiber channels together over time. These metrics were already covered in the previous section so look there for more information if needed.

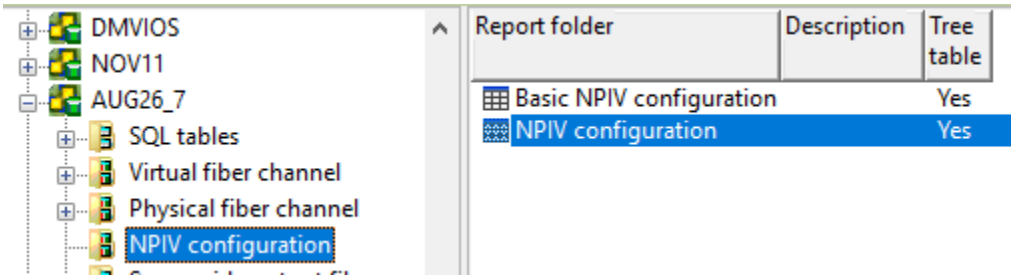
The set of graphs available are:

- FC/IP total reads [PFC]
- FC/IP total writes [PFC]
- FC/IP total control requests [PFC]
- FC/IP total read sizes [PFC]
- FC/IP total write sizes [PFC]
- FC/IP average read size [PFC]
- FC/IP average write size [PFC]

- FC/IP read size per second [PFC]
- FC/IP write size per second [PFC]
- FC/IP reads per second [PFC]
- FC/IP writes per second [PFC]
- FC/IP no DMA resource errors [PFC]
- FC/IP no adapter elements errors [PFC]

7.11.4.4 NPIV Configuration

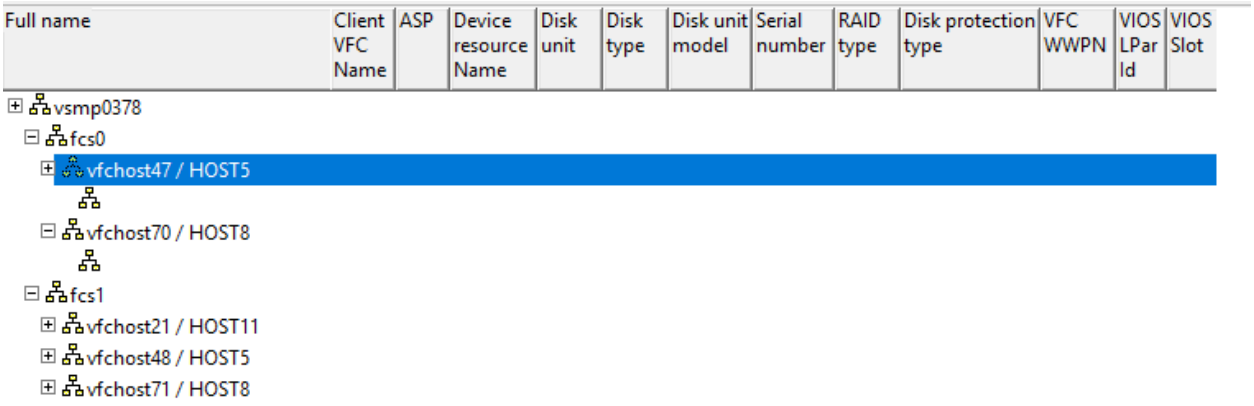
This folder contains reports that show the NPIV configuration in a tree table.



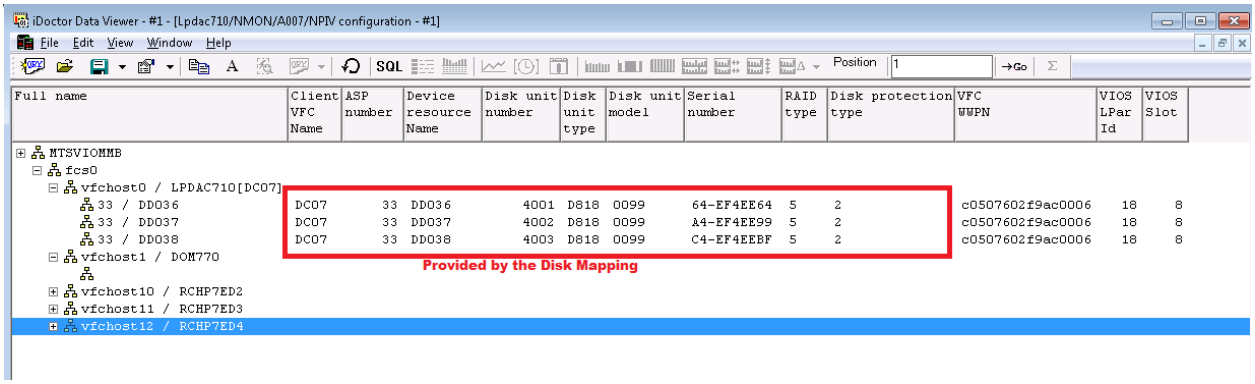
NPIV -> NPIV configuration folder

This folder contains a report (tree) that shows the NPIV configuration along with the IBM i disk mapping details (if provided at analysis time.)

If the disk mapping is NOT provided, then most of the data will not be filled in (see example below.)



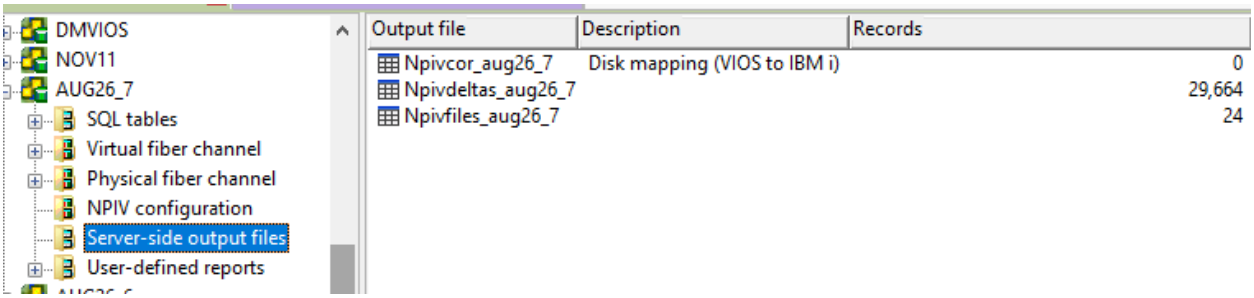
NPIV configuration (without a disk mapping)



NPIV configuration (with a disk mapping)

7.11.5 Server-side output files

This folder contains a list of tables associated with the current collection. This is the set of the tables created during the import/analysis process and they will all begin with NPIV* and end with the name of the current collection.



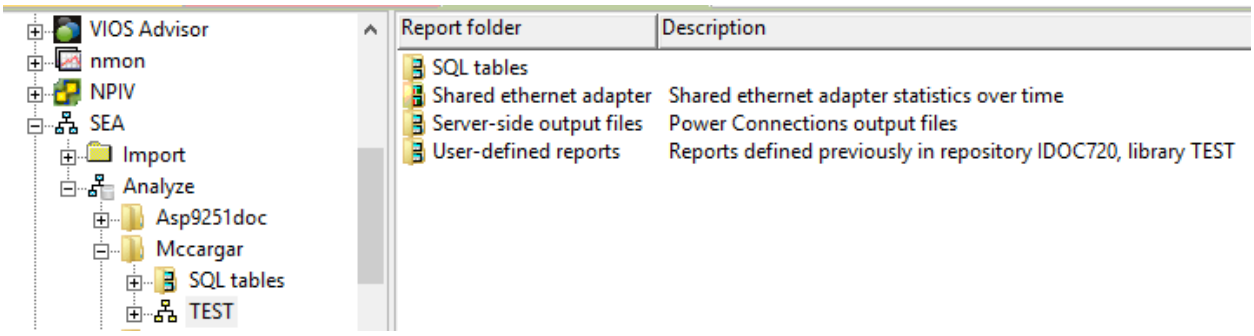
Server-side output files folder

File	Description
Npivcor_<col>	The VIOS to IBM i disk mapping provided (if any) at analysis time.
Npivdeltas_<col>	The contains the interval-based statistics captured from the NPIV scripts but stored in a field described file.
Npivfiles_<col>	This table contains a list of original .npiv files used to create the data during the analysis process. This table only contains the filename in each record but the relative record number of each record in this table is the unique ID for each .npiv file. Other tables such as npivdeltas_<col> contains a FILEID field which maps to the relative record number in this file.

7.12 SEA

This section describes the options available for SEA-type (Shared Ethernet Adapter) data collections.

Several graphs are found within the folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the appropriate menu.



SEA analysis options in iDoctor

Each folder contains a series of graphs. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Some graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one.

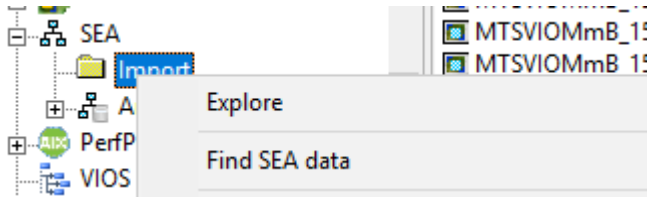
The options available under a SEA-type collection include:

- Shared Ethernet adapter – Includes SEA packet total, size totals, rates as well as ways to rank the SEA devices.

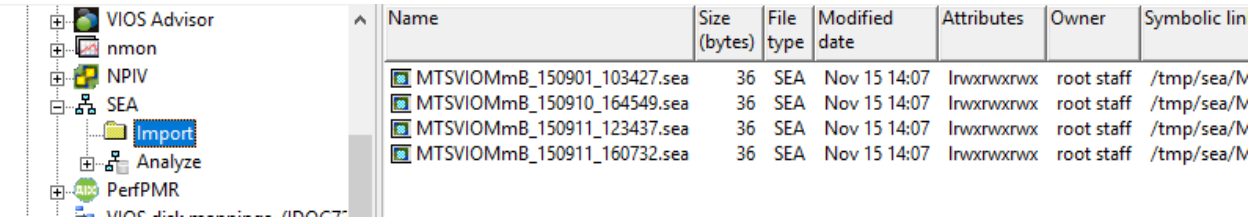
7.12.1 Import

This folder displays the available raw SEA data found on the system that has not yet been processed into the analysis database.

Note: In order to find new data that has been sent to the system, you must use the Find SEA data menu option.



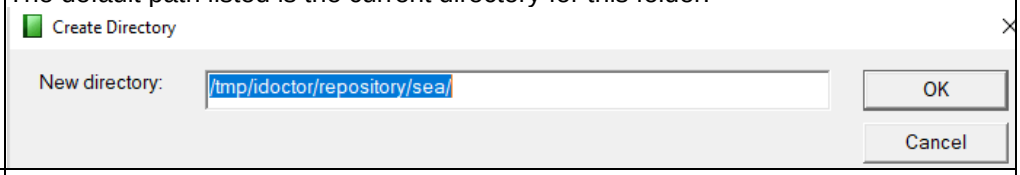
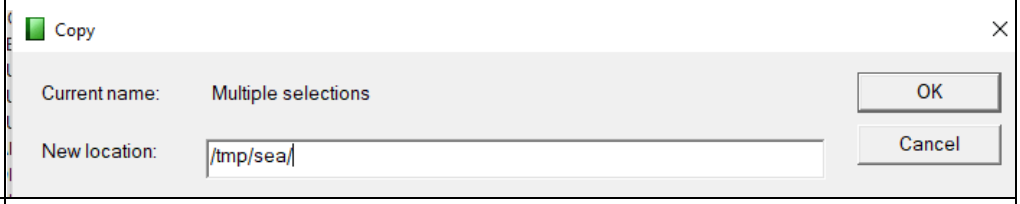
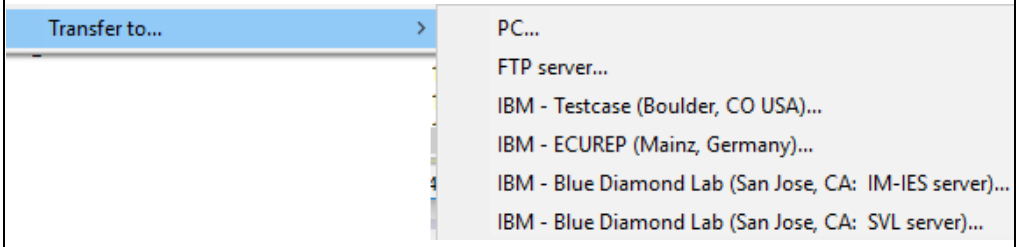
Find SEA data menu



Power Connections -> VIOS1 -> SEA -> Import folder

Note: The contents of the Import folder are cached and only rebuilt each time the Find SEA data option is used. If these files are removed (outside of this interface) the change will NOT be immediately reflected here and only rerunning the Find option will correct this.

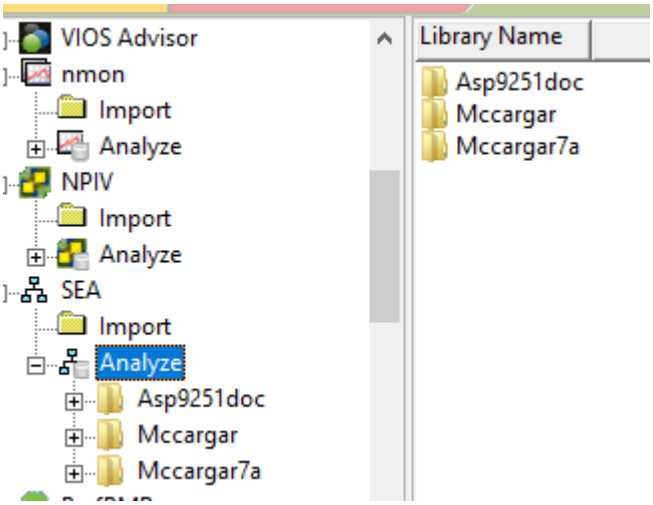
When right-clicking on an .sea file in this folder, the following menu options are available:

Menu	Description
Open (local copy)	This option will cause the .sea file to be downloaded to the PC and then opened using the default program specified by Windows for .sea files.
Edit	This option will open an iDoctor editor which allows you to make changes to the file. **Tip:** Double-clicking the file in the list will also cause this to happen.
Analyze Data	This menu will cause the selected .sea files to be processed by the iDoctor stored procedure that analyzes sea data using the Analyze Data window. Once complete the graphable SEA data can be found under the NPIV -> Analyze folder.
Upload…	This option displays the Upload files from PC window which is used to transfer any previously collected .sea files from the PC to the remote system in the desired directory.
Create Directory	Creates a directory on the remote system you are working with (IFS, VIOS or AIX.) The default path listed is the current directory for this folder. 
Copy…	This option allows you to Copy all selected files to a new directory of your choice. 
Delete	Removes the selected files from the system.
Rename	Renames the selected file.
Save	This option will Save the desired files to either a SAVF (IBM i) or tar file (AIX, VIOS) depending on the type of system being used.
Transfer to	This menu provides several choices for sending the .sea files to another server or the PC. 

7.12.2 Analyze

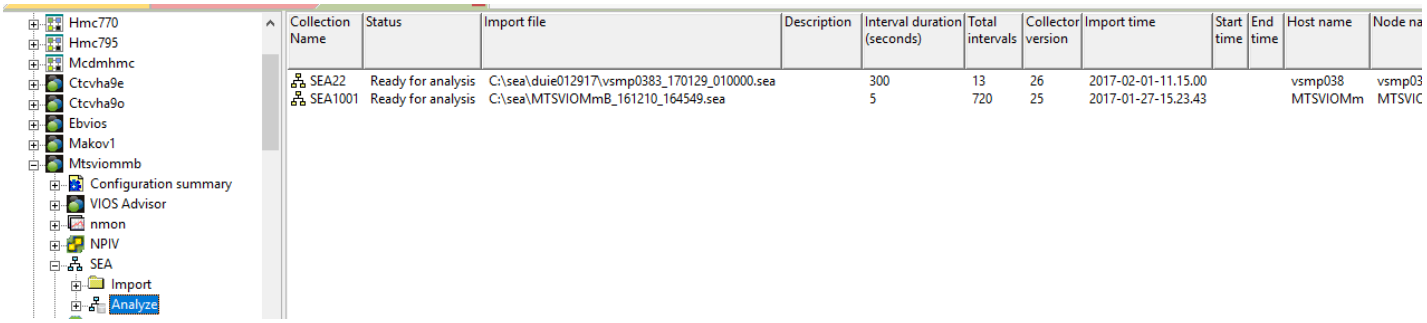
If using an IBM i as the analysis DB or connected currently to an IBM i then this folder provides a list of all libraries on the current system that contains SEA data.

An example is:



Power Connections -> VIOS1 -> SEA -> Analyze folder

Note: If using SQLite as the analysis DB then this folder contains the collections (example below.)

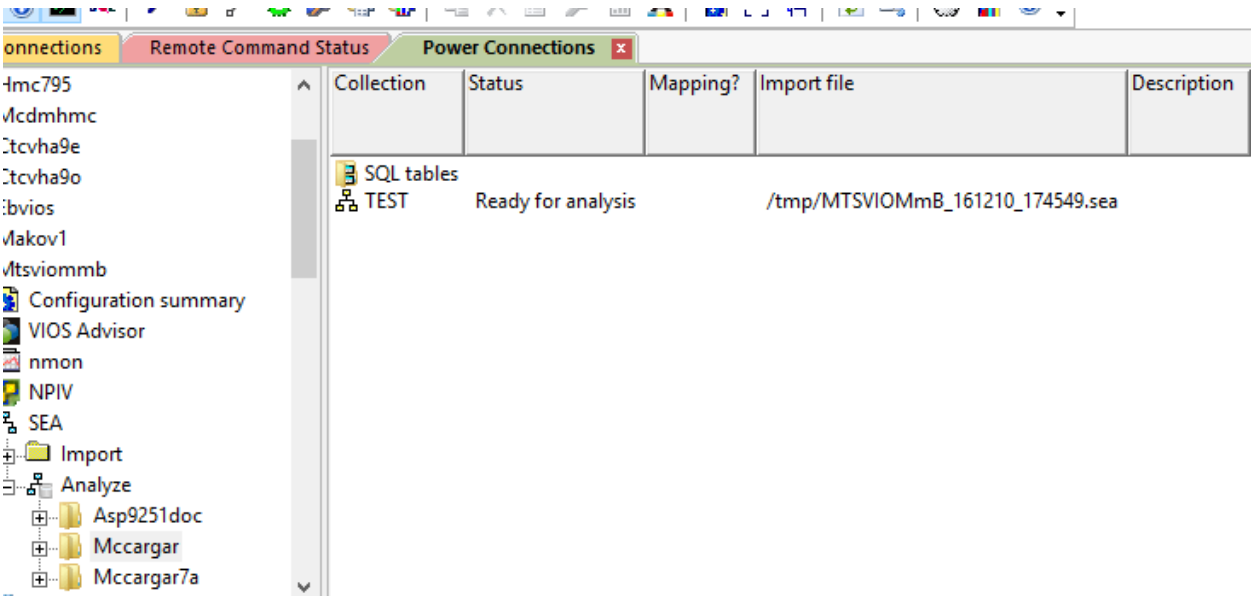


Power Connections -> VIOS -> SEA -> Analyze folder

7.12.3 Collections

Each library (or SQLite DB) contains a list of SEA collections that have been imported into it. Each collection consists of a set of iDoctor created SQL Tables that were derived from the NPIV data and in some cases a VIOS to IBM i disk mapping (optional).

Each collection contains a detailed set of reporting options (graphs or tables) within it.



List of collections in library mccargar

7.12.3.1 Collection Fields

The list of collections contains several columns which are described below:

Field	Description
Collection	Name of the collection. This is a short unique name (10 characters or less) given to the collection at import time based on the collection name prefix value.
Status	Indicates if all necessary SQL Tables have been created successfully.
Mapping?	Indicates if a disk mapping was used when creating this collection. Not having a disk mapping will mean fewer analysis options will be available.
Import File	This column displays the original file name that was used when the import occurred.
Description	A description given to the collection when the import occurred.
Interval duration	The duration of each interval within the SEA data.
Total intervals	The total number of intervals (snapshots) taken.
Collector version	This is the iDoctor build number used when this collection was created. This by default is stored in the iDoctor scripts directory in /tmp/idoctor/qidrvrm.txt The number increases by 1 each time the iDoctor scripts we ship are updated.
Import time	The date and time the import occurred.
Start time	The date and time when the collection started.
End time	The date and time when the collection ended.
Host name	The name of the host on the system the data was collected on.
Node name	The node name for the system the data was collected on.
User name	The user name that started the collection.
Build	This returns the result of the uname -s command. On VIOS this returns AIX typically.
Command	The command used to create the SEA data.
AIX	The version of AIX installed when the collection was created.
System serial number	The system serial number of the system where the data was created.
Mode	This indicates the import mode used on the Analyze data window when importing the data into the database. 0 = Normal 1 = Append 2 = Merge

7.12.3.2 Menus

The following menu options are available by right-clicking on a SEA collection.

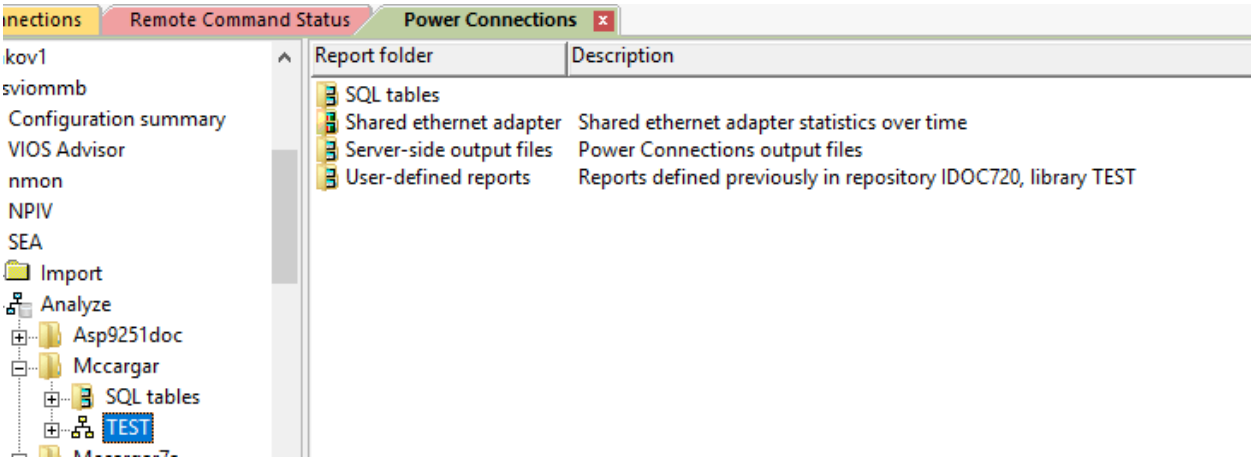
Menu	Description
Analyses	This contains the list of analysis options available to run against the selected collections. In iDoctor, these analyses are SQL stored procedures. Currently these are the options available: **Analyze Collection** – This is the interface that lets you work with all the analyses. **Run Change SEA sensitive user data** – This will modify the data in the SEA collection to hide/replace potential any customer sensitive data. **Note:** This option only applies when connected to an IBM i or using an IBM i as the analysis DB.
Shared ethernet adapter	This contains a set of graphs for the Shared Ethernet Adapter statistics. Also includes a sub folder to rank the SEA statistics by device name.
Generate Reports	Launches the Report Generator function that lets you create multiple reports at once.
Copy	This function will copy the desired collection to a different name in the same library or copy one or more selected collections to another library.
Delete	Removes the selected collections from the system.

Additional menu options that are common to all library folders in iDoctor are discussed here.

7.12.4 Reports

This section describes the reporting options available for NPIV collections.

Many graphs are found within several folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the appropriate menu.

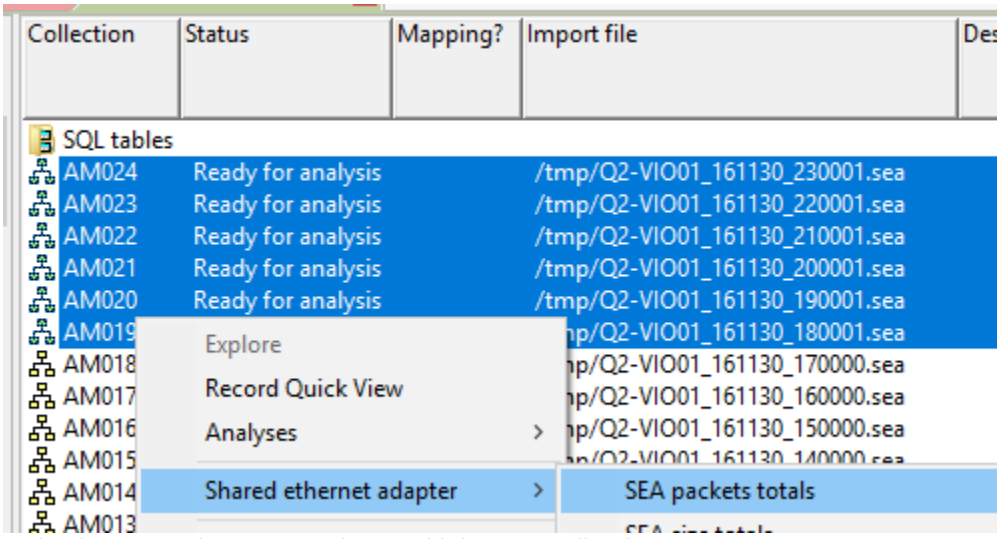


Power Connections -> VIOS1 -> SEA -> Analyze -> MCCARGAR (lib) -> TEST (collection)

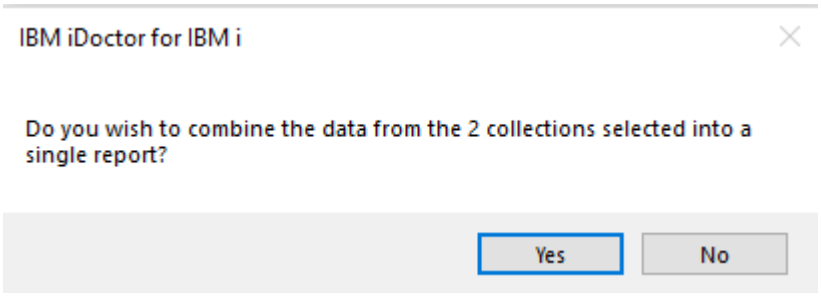
Each folder contains a series of graphs or reports. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Some graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one.

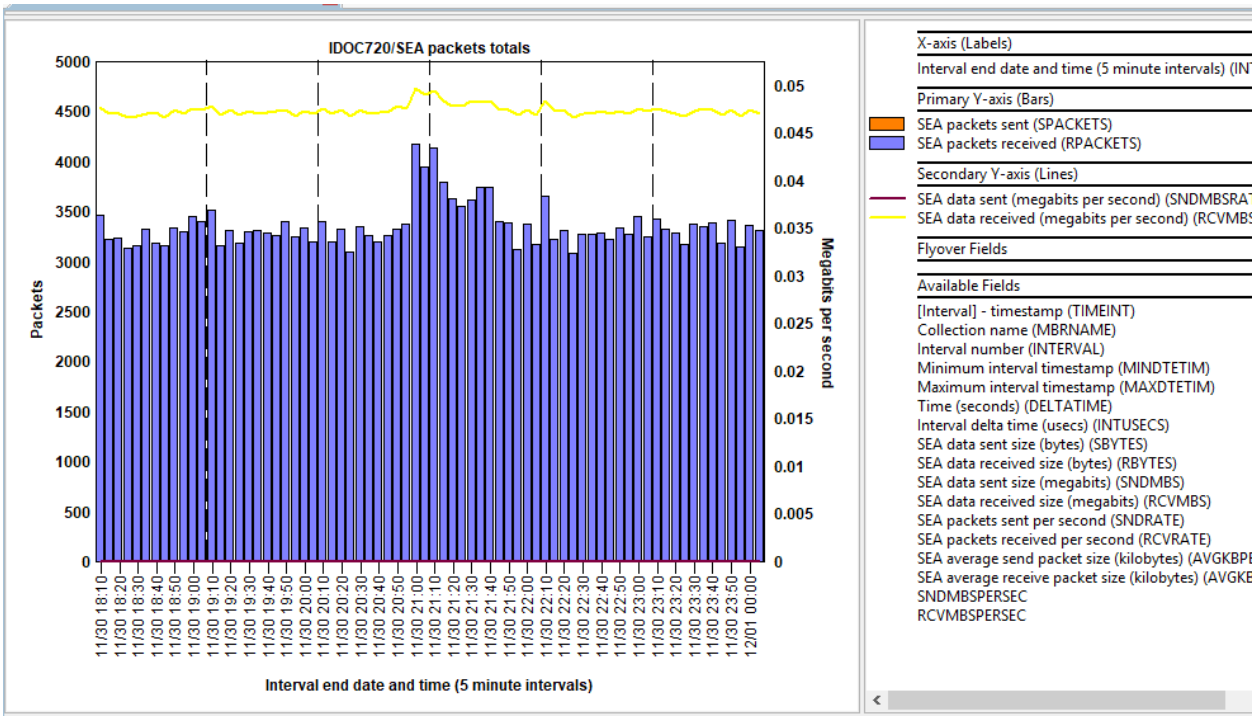
Tip: Many graphs (such as the Shared ethernet adapter -> SEA packets totals) support graphing multiple collections at once. To do this, select all desired collections from the list of collections, then right-click and pick the desired graph from the menu.



Selecting a graph to run against multiple SEA collections



Click Yes



Fiber channel utilization graph against 6 SEA collections

7.12.4.1 Menus

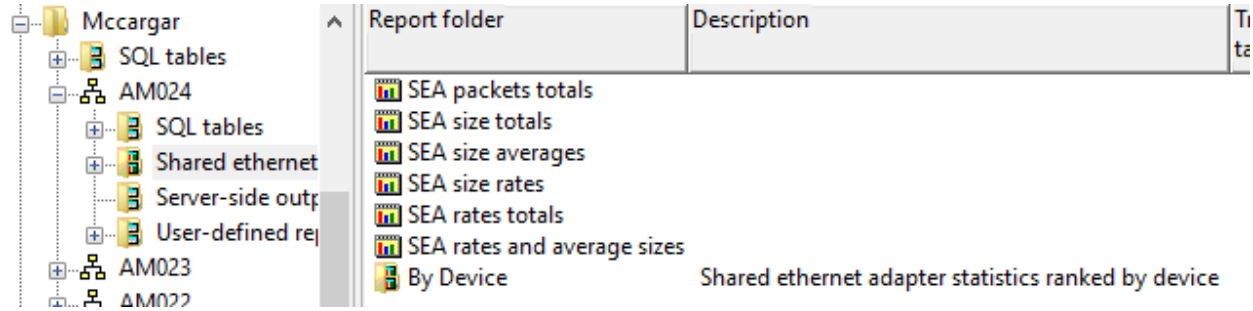
Right-clicking a graph gives a menu with the following options:

Menu	Field Description
Open graph(s)	Opens the selected graphs into a new Data Viewer or an existing one depending on the submenu available that shows the list of Data Viewers (if any are open).
Edit	This option will open the graph without running the SQL statement. The SQL Editor will be opened allowing the user to modify the query before running the SQL.

7.12.4.2 Shared ethernet adapter

This set of graphs displays the summarized statistics for all shared ethernet adapters found in the data.

The options available are:

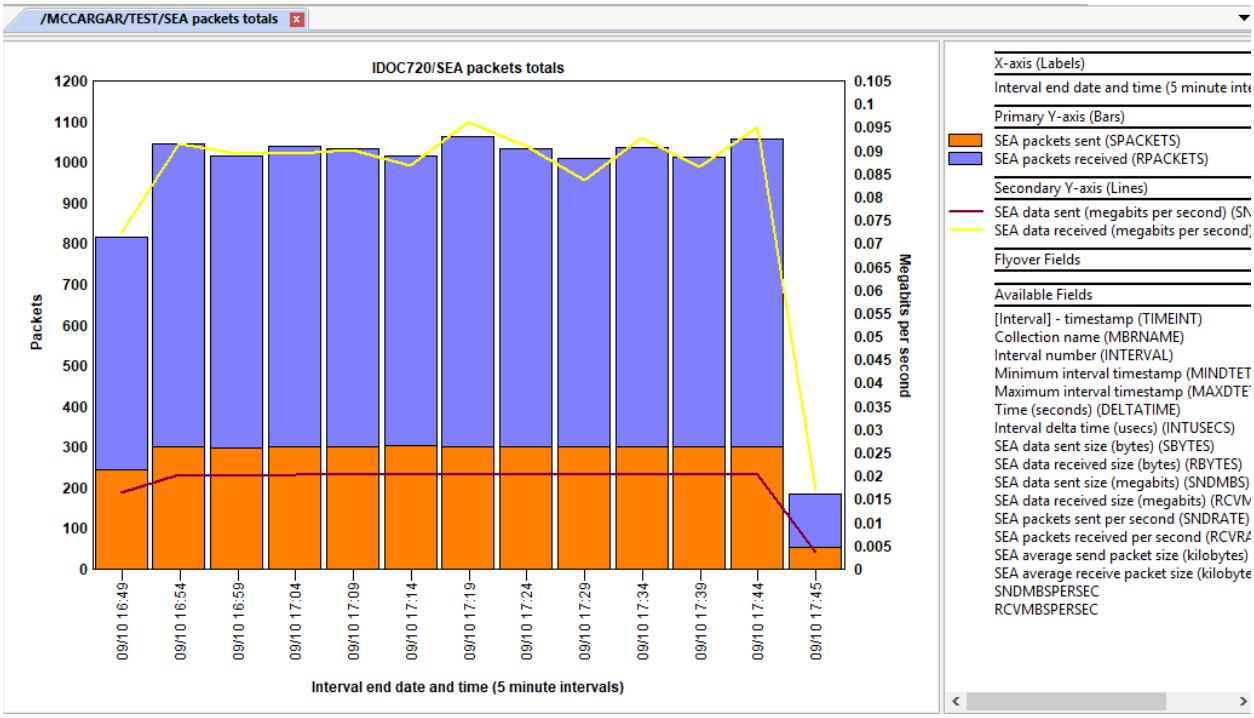


SEA -> Shared ethernet adapter folder

Note: The **By Device** folder graphs contains the same set of graphs shown here but those are rankings graphs per device.

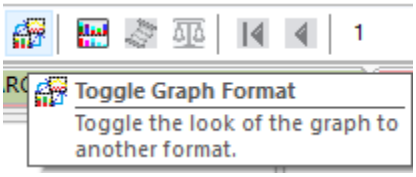
7.12.4.2.1 SEA packets totals

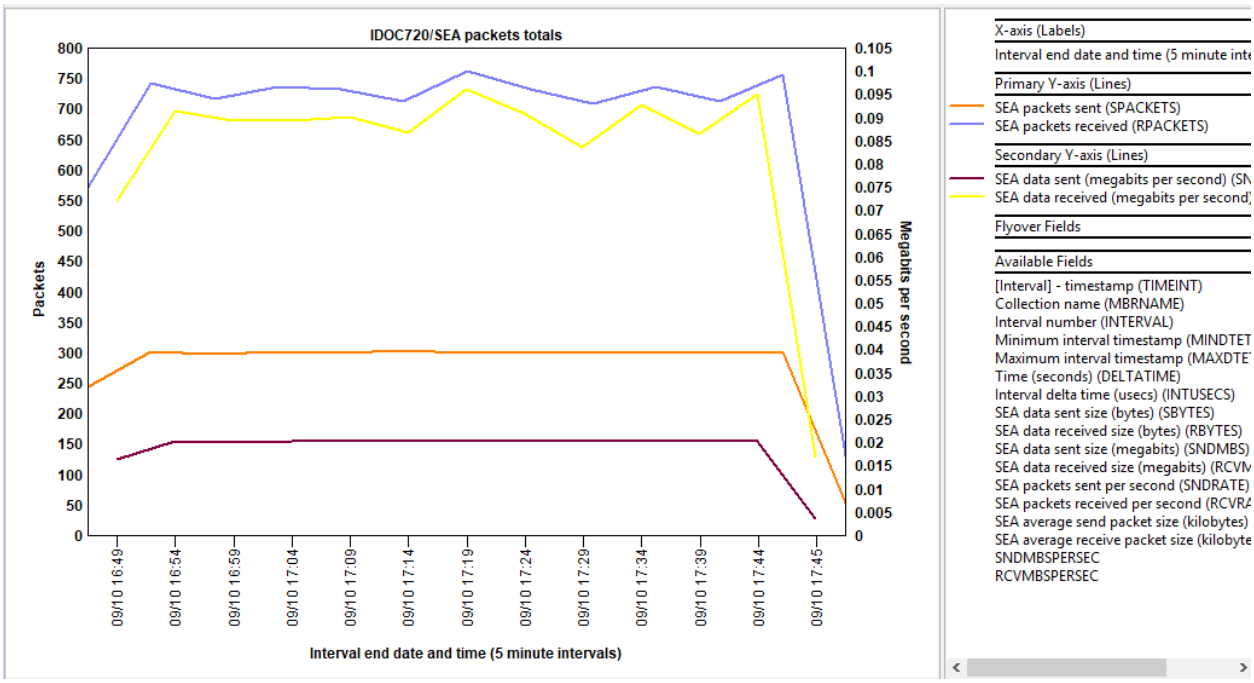
This graph summarizes the SEA adapters together over time and displays the total packets sent and received as well as sent/received megabytes per second.



SEA packets totals

Tip: If you do NOT like bar graphs you can press the Toggle graph format button on the toolbar to convert these graphs to lines on the Y1-axis. Your change is remembered for next time you open this graph.

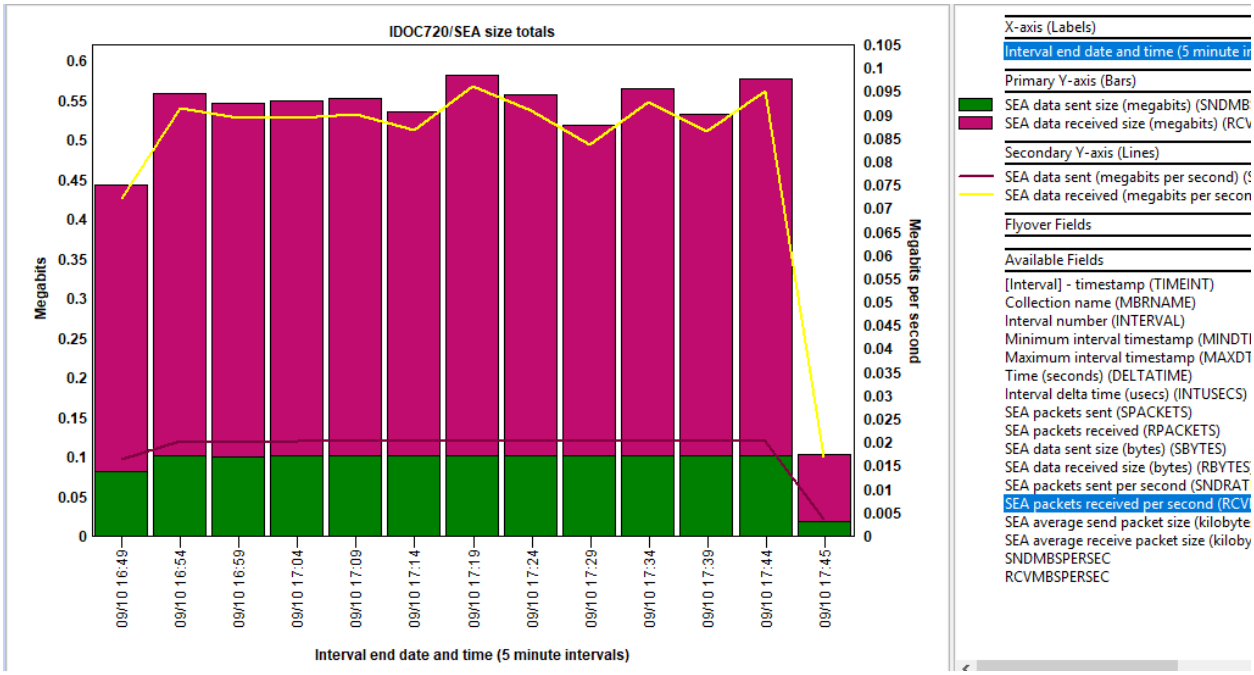




SEA packet totals converted to a line graph (on Y1)

7.12.4.2.2 SEA size totals

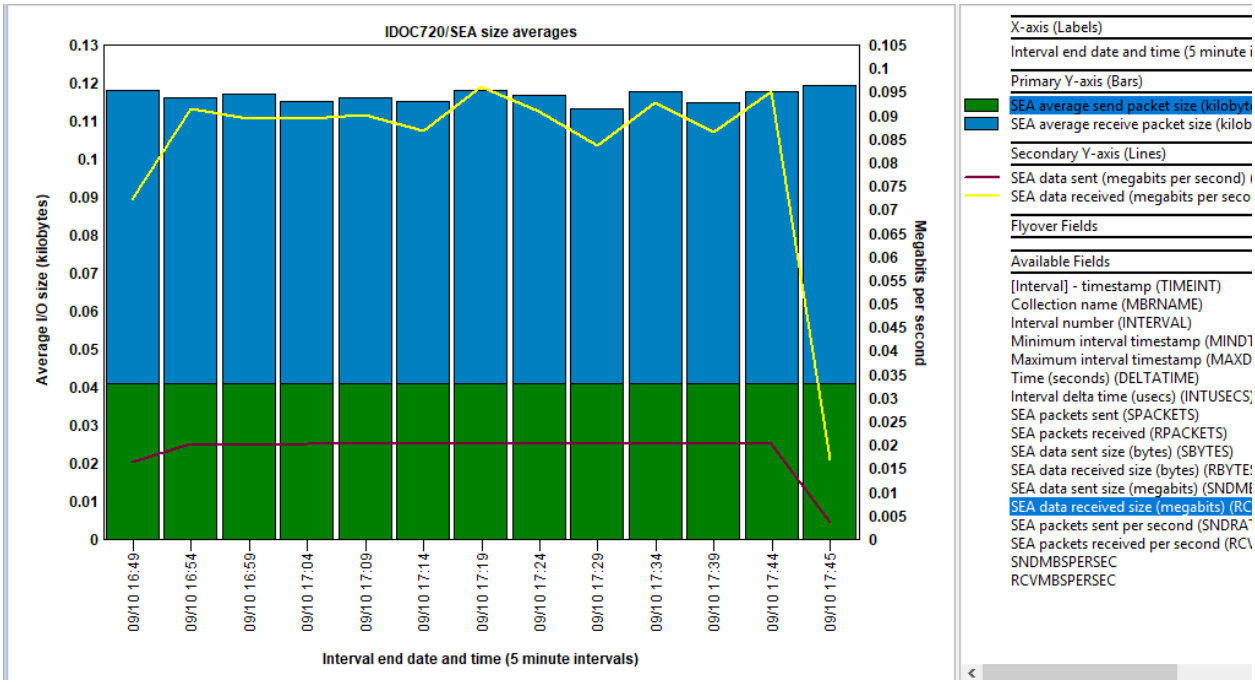
This graph shows the data size sent and received in megabits on the Y1 and also the rate on the second Y-axis (Y2).



SEA size totals

7.12.4.2.3 SEA size averages

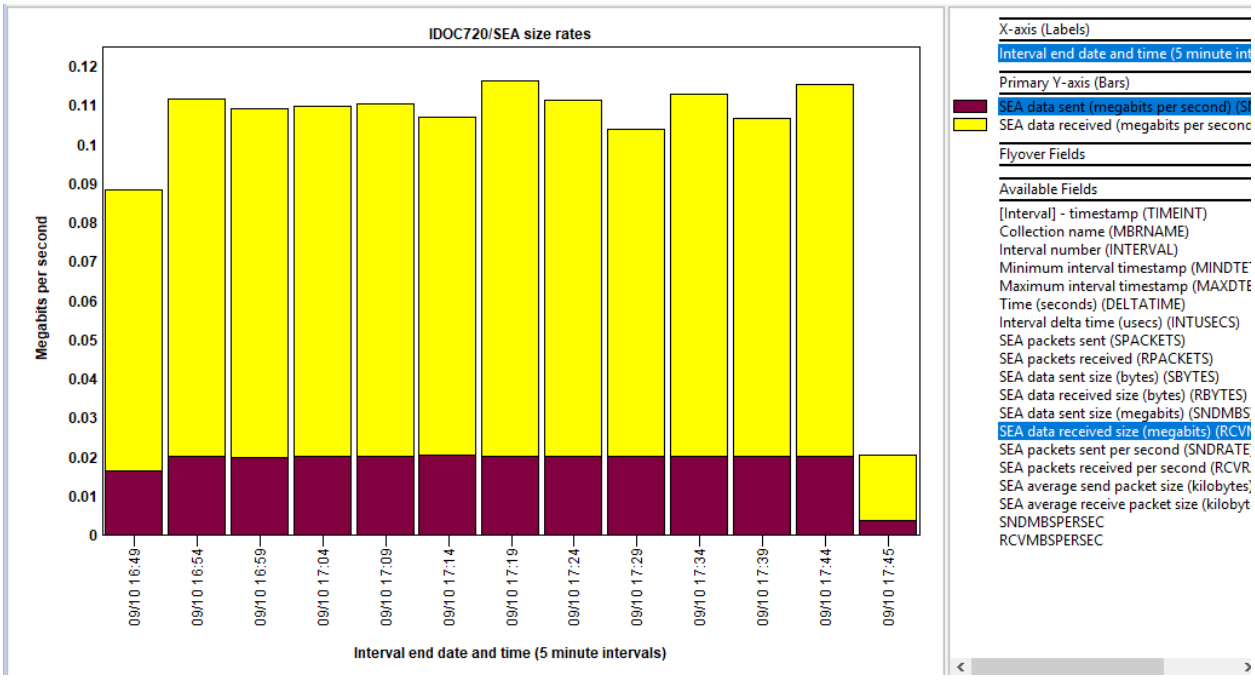
This graph displays the average packet size (in kilobytes) as well as the megabits sent/received per second (Y2.)



SEA size averages

7.12.4.2.4 SEA size rates

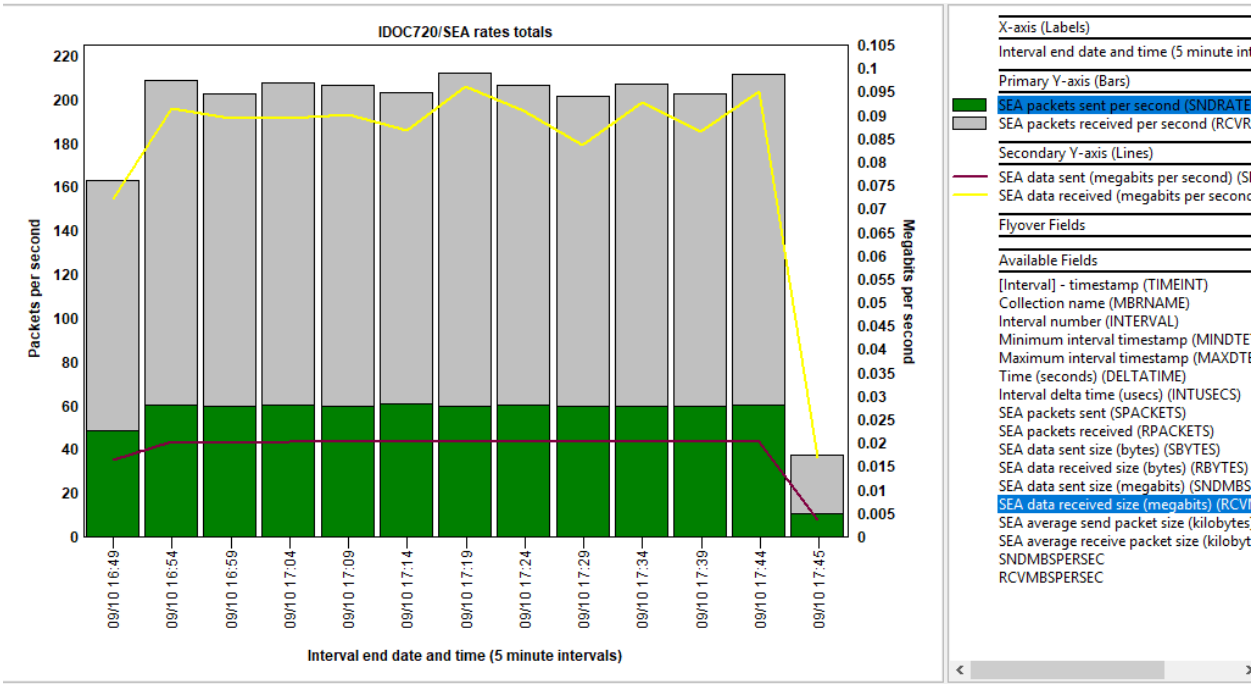
This graph displays the megabits per second (sent/received) on the primary Y-axis (Y1.)



SEA size rates

7.12.4.2.5 SEA rates totals

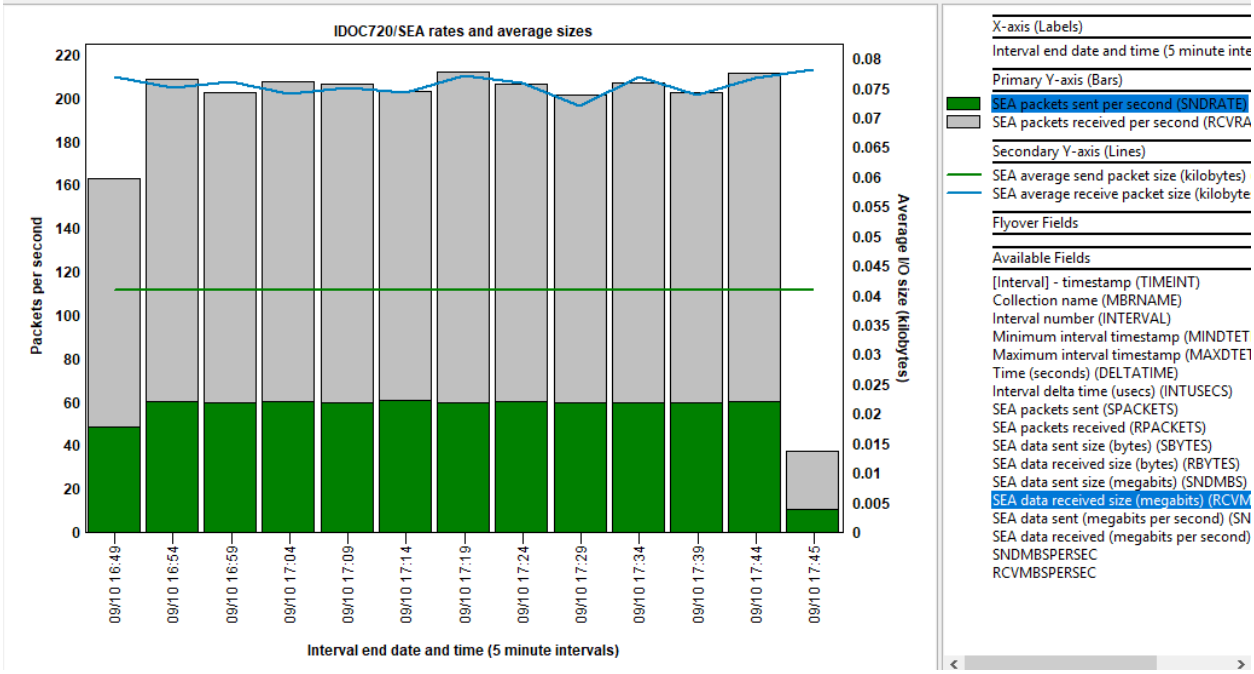
This graph displays the SEA packets sent/received per second (Y1) and the megabits per second sent/received on the Y2.



SEA rates totals

7.12.4.2.6 SEA rates and average sizes

This graph displays the SEA packets sent/received per second (Y1) and the average I/O size on the Y2.

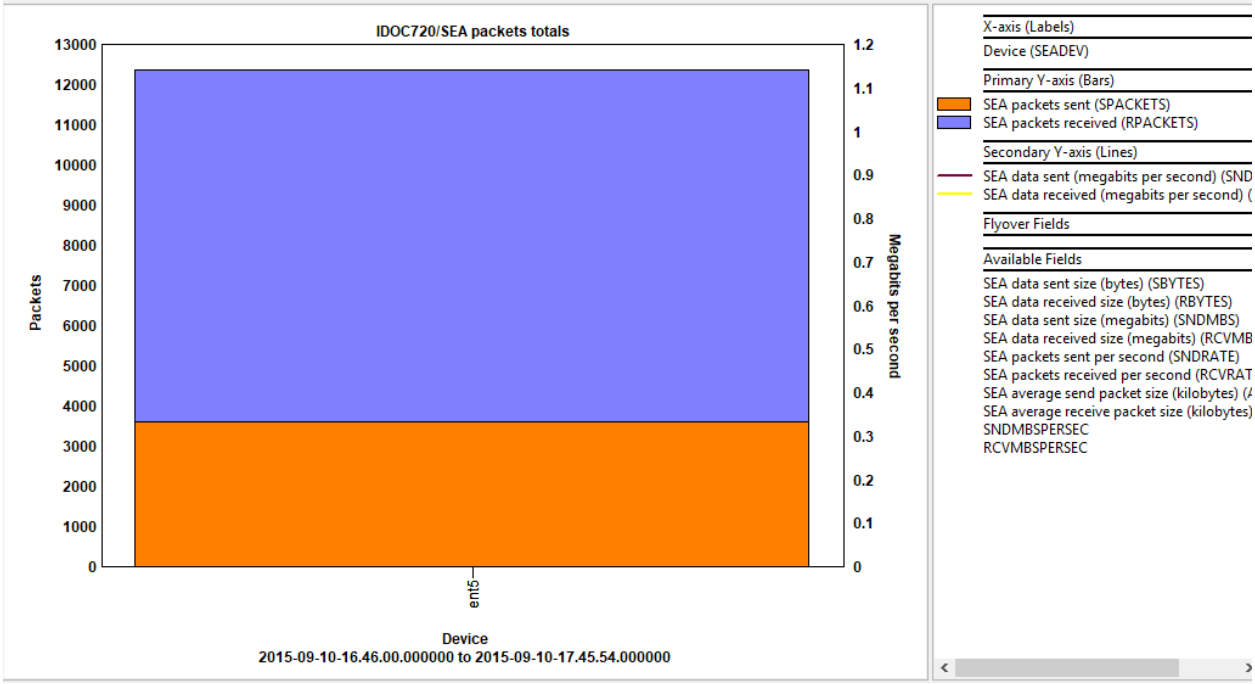


SEA rates and average sizes

7.12.4.2.7 By Device

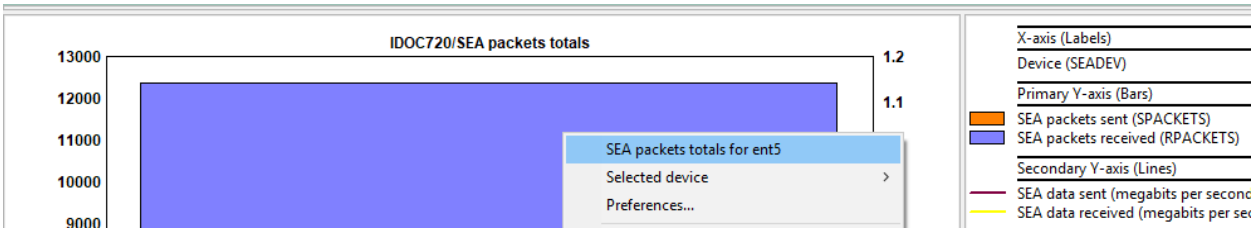
These graphs rank the virtual fiber channel metrics by shared ethernet adapter.

An example of this type of graph is:

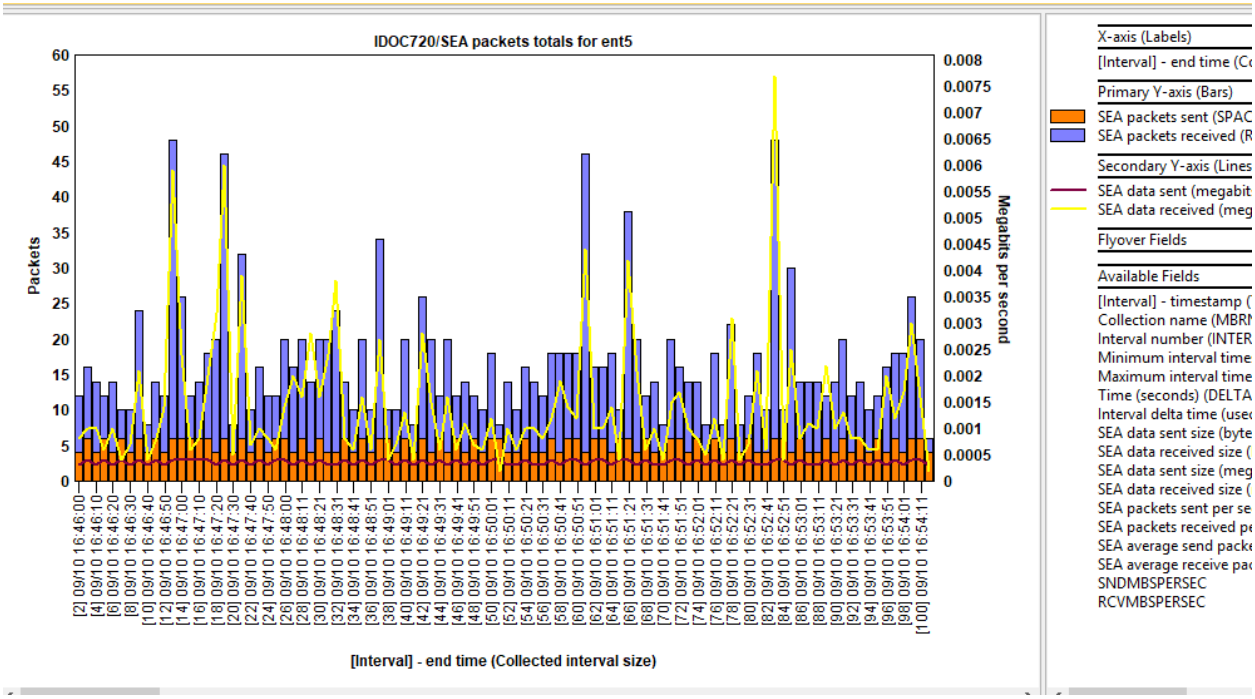


Shared ethernet adapter -> By Device -> SEA packets totals

From this type of graph, you can right-click…



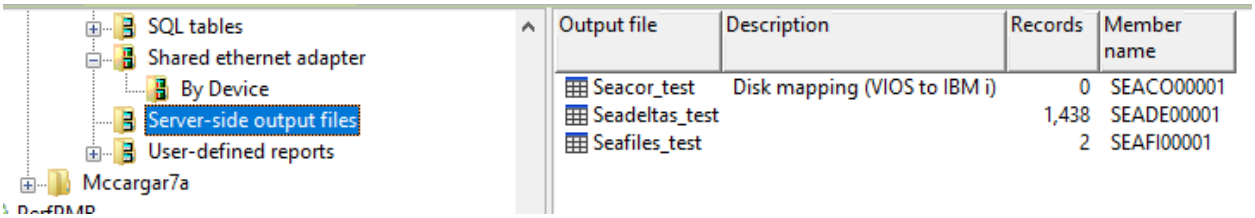
and pick the 1st option to graph the desired device over time.



SEA packet totals for ent5

7.12.5 Server-side output files

This folder contains a list of tables associated with the current collection. This is the set of the tables created during the import/analysis process and they will all begin with SEA* and end with the name of the current collection.



Server-side output files folder

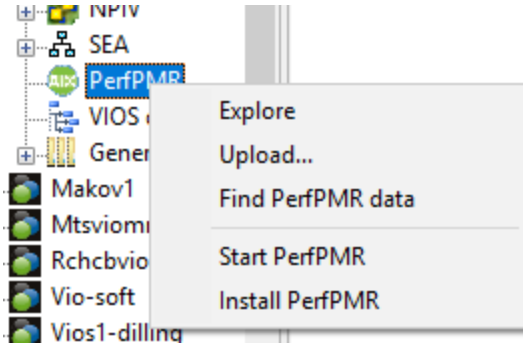
File	Description
SEAcor_<col>	The VIOS to IBM i disk mapping provided (if any) at analysis time.
SEAdeltas_<col>	The contains the interval-based statistics captured from the SEA scripts but stored in a field described file.
SEAfiles_<col>	This table contains a list of original .sea files used to create the data during the analysis process. This table only contains the filename in each record but the relative record number of each record in this table is the unique ID for each .sea file. Other tables such as seadeltas_<col> contains a FILEID field which maps to the relative record number in this file.

7.13 PerfPMR

This folder lets you work with any PerfPMR data directories found on the current VIOS.

NOTE: This option is only available if you are working with a VIOS within the Power Connections interface.

Use the Find menu option to locate existing or new data not shown in the PerfPMR folder.

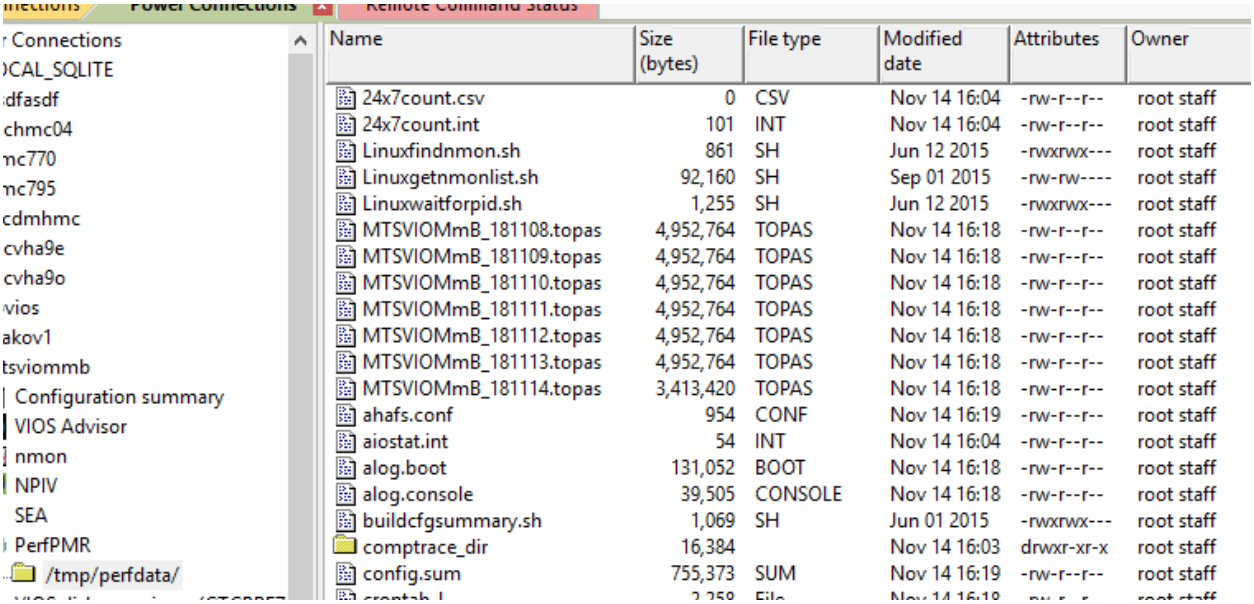


After the Find function completes refresh the PerfPMR folder to display any PerfPMR created data found on the VIOS.

Directory	Location	Partition collected on	Time	Partition Number	Type	Mode	Entitled Capacity	Online Virtual CPUs	Online Memory	Variable Capacity Weight	Data Directory	PerfPMR version
/tmp/perfdata/	MTSVIOMmB	MTSVIOMMB	Wed Nov 14 16:04:25 CST 2018	18	Shared-SMT-4	Uncapped	4.00	4	8192 MB	128	/tmp/idoctor	610

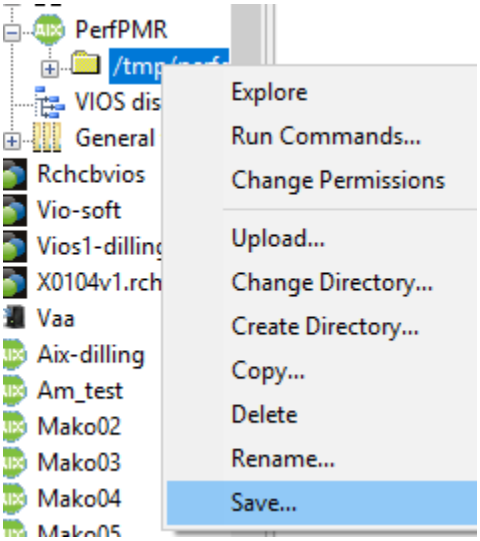
PerfPMR folder contents

Expand the folder to view the contents (if desired):

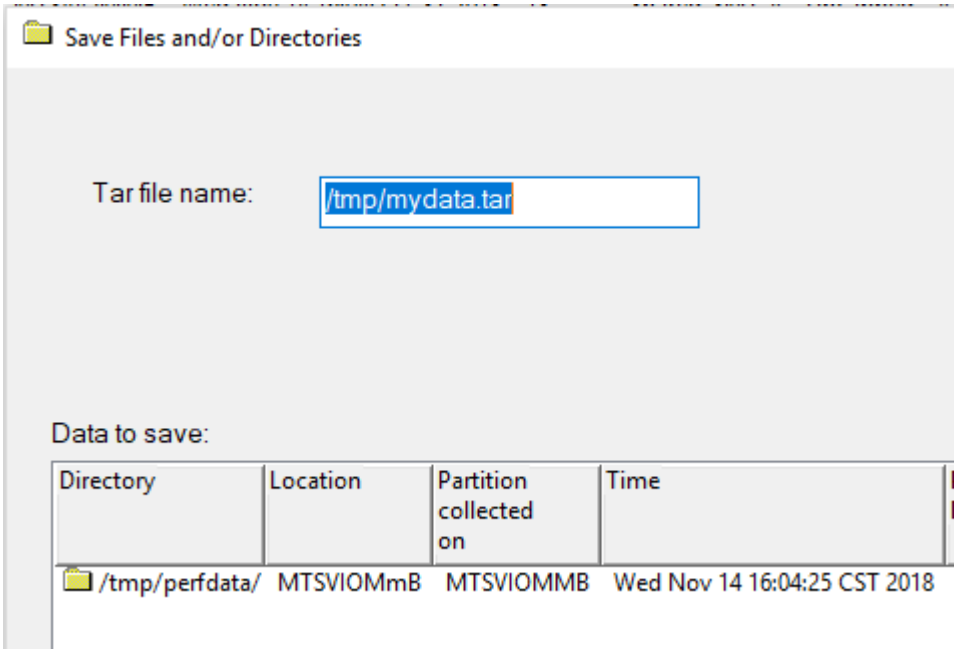


PerfPMR /tmp/perfdata/ contents

To send the data to IBM, first use the Save function on the directory and give it a tar file name.

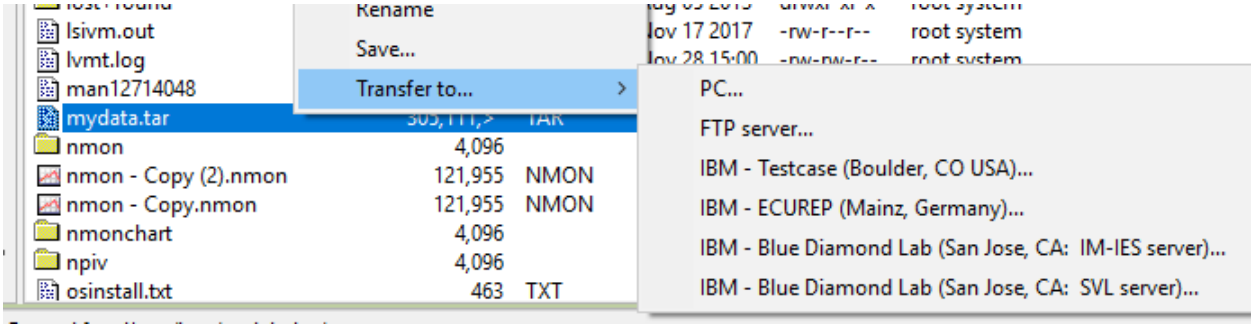


PerfPMR -> /tmp/perfdata/ -> Save menu option



Save Files and/or Directories

Then once the tar file has been created you can transfer that to IBM.



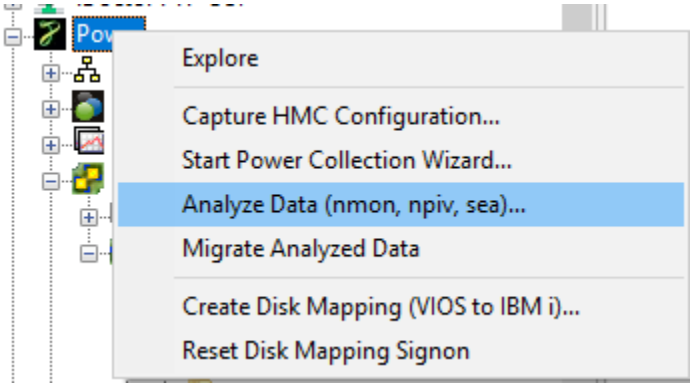
Transfer menu options for a .tar file

7.14 VIOS to IBM i disk mappings

When connected to a VIOS system you will have the option to create a disk mapping between the VIOS and an IBM i of your choice. The IBM i should be associated with the VIOS via the HMC for this feature to work.

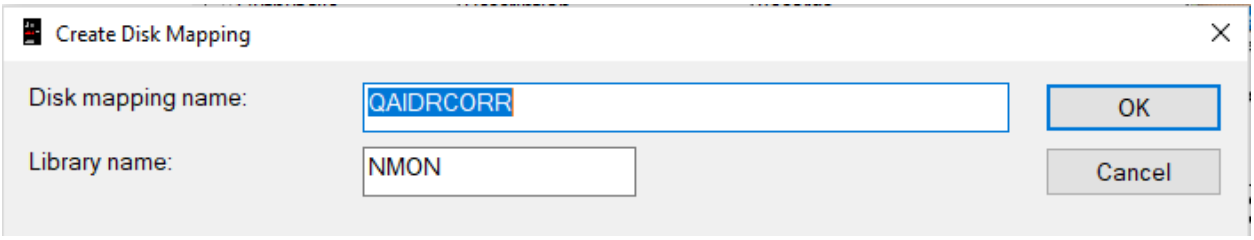
7.14.1 Create Disk Mapping Window

You can create a new disk mapping for a VIOS by right-clicking the General functions -> Power folder or right-clicking on a VIOS connection within the Power Connections view. Then use the Create Disk Mapping menu option to launch the interface.



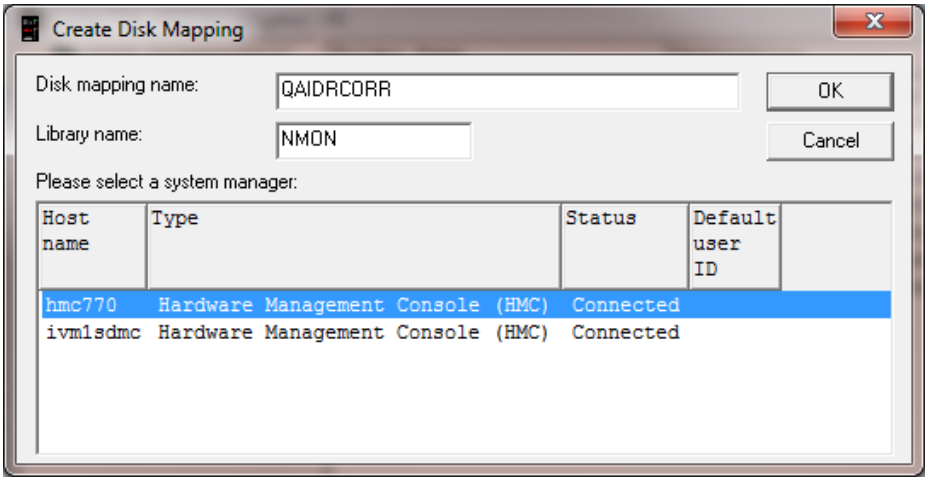
Power menu options

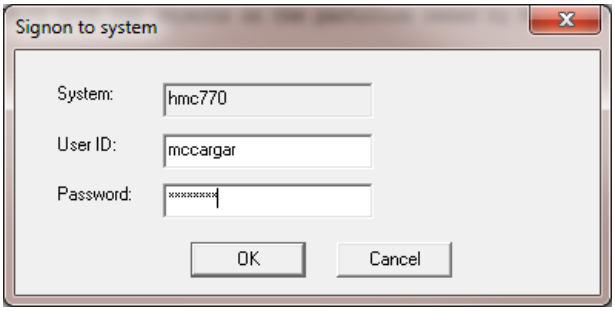
This window allows a user to add a new disk mapping to their system by using data from the HMC and the current IBM i you are working with. If you are connected to a VIOS within Power Connections, then this means the IBM i specified as the Analysis DB.



Create Disk Mapping Window Step 1 (Specify table and library name)

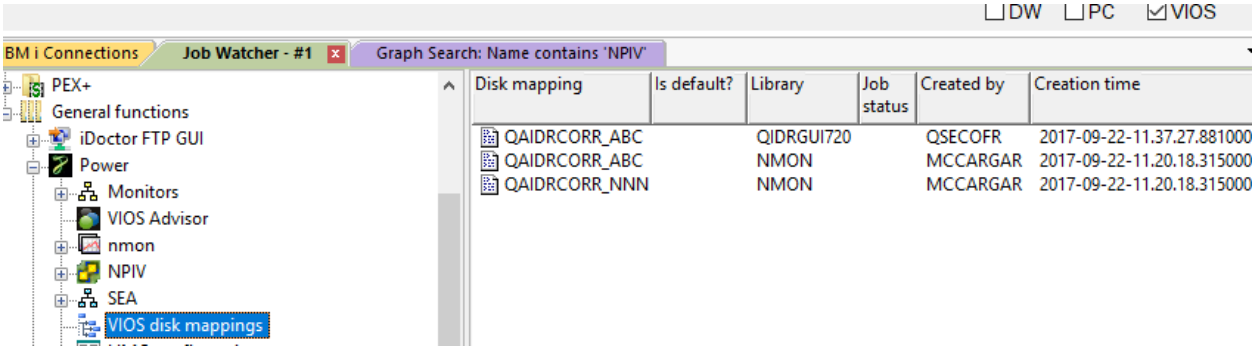
First, this window prompts for the SQL table name and library name to create the new disk mapping table into. After clicking OK, the user will need to select the desired HMC (if more than one was detected) and sign on to the desired HMC to gather the disk information from.



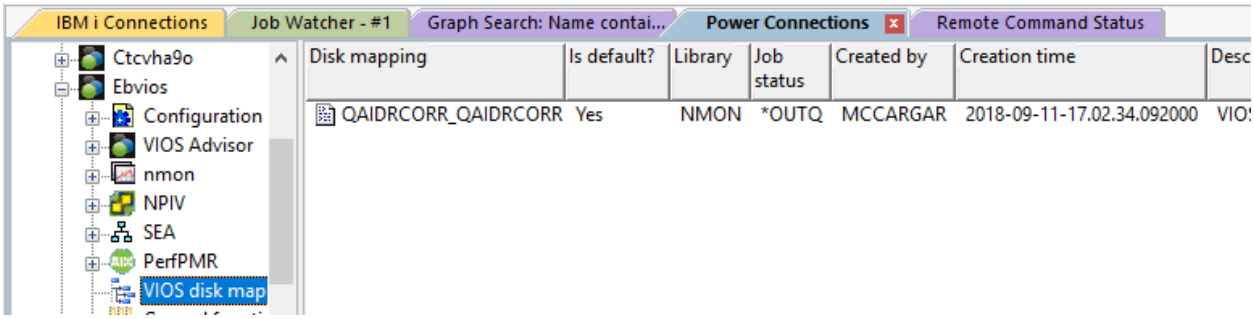


Create Disk Mapping Window Step 2 (Select the HMC and signon)

After signing on to the HMC successfully, information about the virtual disks are gathered. A batch job on the IBM i is started and will gather the necessary information and fill the SQL table. To view the progress of this batch job, expand the VIOS Disk mappings folder under General functions -> Power or under your VIOS connection if using the Power Connections interface..



IBM i –> Job watcher -> Power -> VIOS disk mappings folder



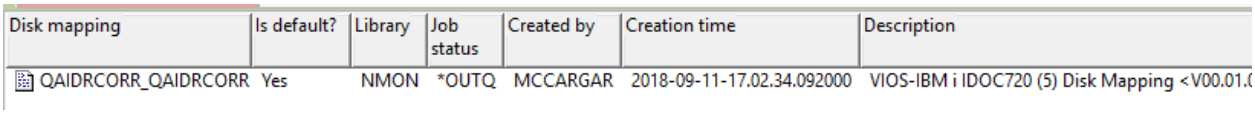
Power Connections -> VIOS ebvios -> VIOS disk mappings folder

Note: Successful signing on to the HMC using this interface will automatically create a text file called SysMgrs in the IFS under the directory /home/"your_user_profile"/QIBM/iDoctor. This file keeps track of the system managers available and the user ID to connect with. The creation of this file will cause step 2 to be skipped next time, but if you wish to switch to use a different HMC (or if you have problems with the process) deleting this text file would reset the process. A menu option called "Remove Disk Mapping SysMgrs file from IFS" has been provided to do this.

Once the mappings have completed you can double-click them to view them. Also on the Analyze Data window you will indicate the desired disk mapping(s) you wish to use when analyzing new .nmon or .npiv data files.

7.14.2 VIOS Disk Mappings Folder

This folder contains the list of disk mappings on the current IBM i. If working with the Power Connections view and connected to a VIOS then this folder contains the disk mappings found on the (IBM i) analysis DB.



VIOS Disk Mappings Folder

Double-click a disk mapping from the list in order to view it.

Column	Description
Disk mapping	This is the name of the SQL table / disk mapping.
Is default?	Indicates 'Yes' if this is the default disk mapping to use for the current system. You can right-click an entry and use the Set as default menu in order to set this.
Library	The library name that contains the SQL table.
Job status	This is the status of the job that created or is currently creating the disk mapping. *OUTQ status means the job has finished.
Created by	The user profile on the IBM i that created the disk mapping
Creation time	The time the job started that created the disk mapping.
Description	The description for the disk mapping.
File	The physical file object name (10 characters max) for the disk mapping.
Job creating disk mapping	This is the job name that created or is currently creating the disk mapping on the system. These jobs are named QIDRESCORR.

8 Job Watcher

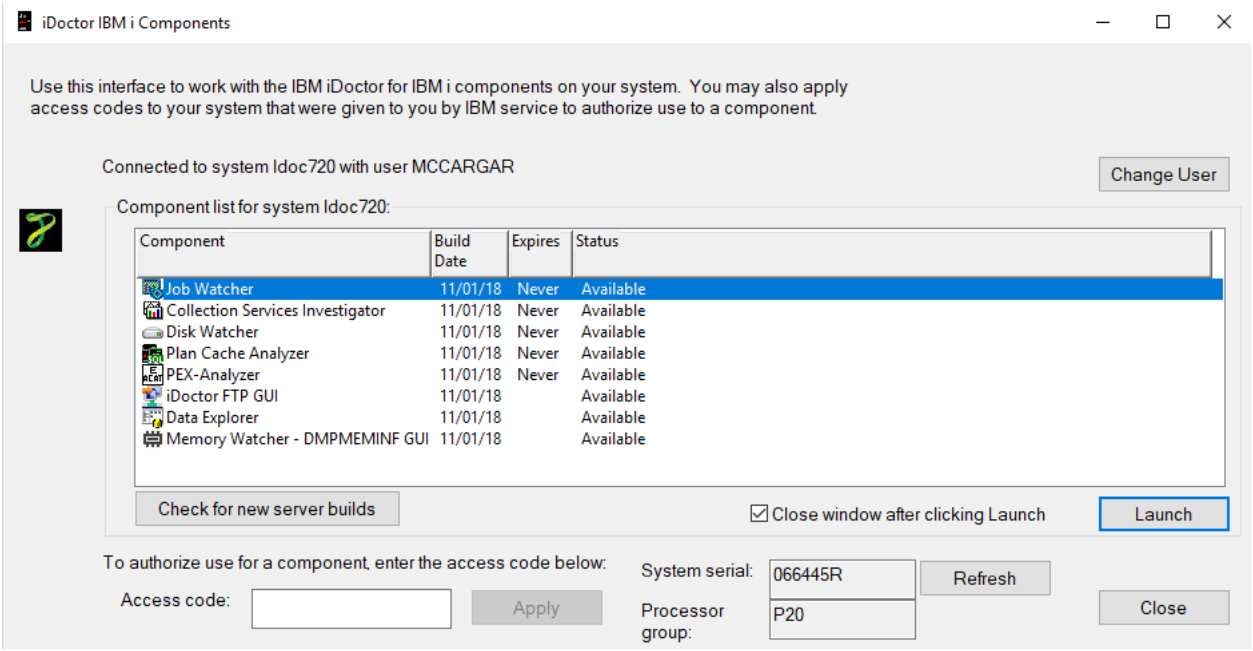
This chapter provides an overview of the interfaces within iDoctor's Job Watcher component.

Note: iDoctor Job Watcher is a completely different offering than what is provided in IBM i 5770PT1 – Job Watcher option and the Performance Data Investigator (web GUI within IBM Navigator for i). These are two different products and purchased and supported separately within IBM.

8.1 Starting Job Watcher

Job Watcher is a component of the iDoctor suite of tools. iDoctor can be started using the Start menu: Start->IBM iDoctor for IBM i. Once the IBM iDoctor for IBM i application appears, the Job Watcher component is started from the IBM i Connections View by double-clicking on the desired system. If there are none you will need to add one to the list first.

A list of available components will appear on the next window. Double-click on the Job Watcher component or select Job Watcher and click the Launch button in order to continue

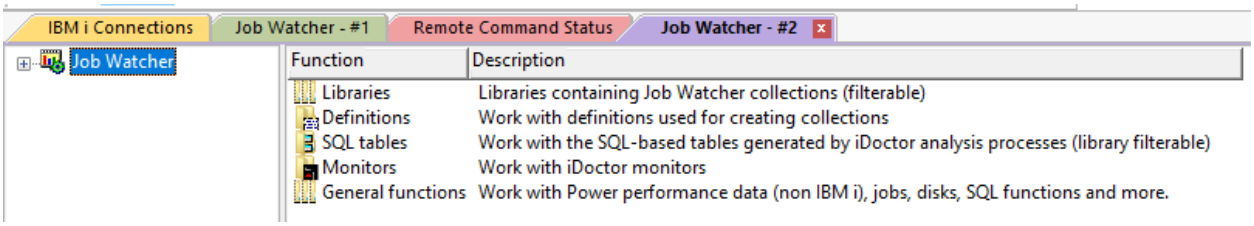


iDoctor IBM i Components Window

Note: Collection Services Investigator, Plan Cache Analyzer and Disk Watcher will only be available if Job Watcher is installed correctly and a valid access code for Job Watcher has been applied. These components are included with the Job Watcher license.

8.2 Job Watcher Component View

The Job Watcher view is the interface used to add definitions, start new collections or monitors, or work with existing data.



Job Watcher Component View

The 'Job Watcher' folder contains a list of folders, each providing different features available. Collections can be displayed in various ways, either under the Libraries folder on a per library basis, or under the Monitors folders for Job Watcher collections that exist within a monitor.

The General functions folder contains several additional options for working with the IFS, browsing objects, working with the disk units and ASPs or working with any non-IBM i data (VIOS/HMC/AIX) that has been collected and moved to this system. The options for working with non-IBM i data are stored in the General functions -> Power folder.

8.2.1 Menu Options

The following Job Watcher specific menu options are available by right clicking on the 'Job Watcher' icon in the component view above:

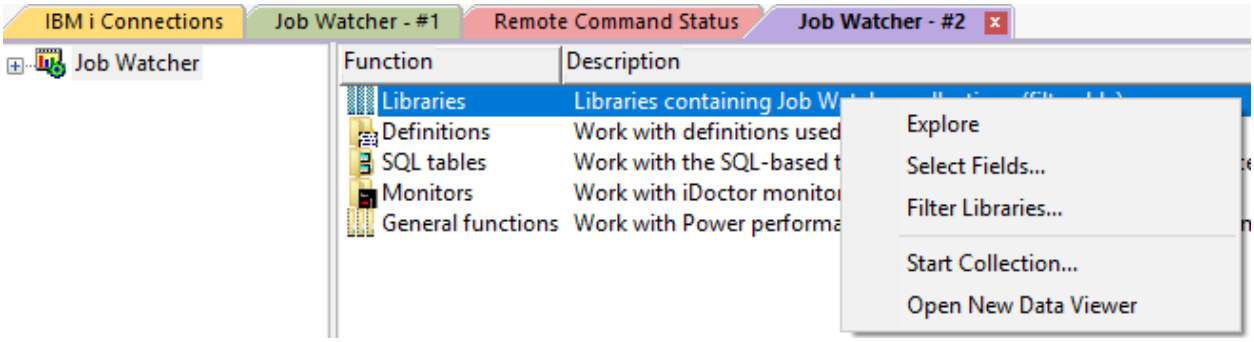
Menu Item	Description
Filter libraries…	This option allows you to filter the libraries shown in the Libraries and SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of the list if the system contains many libraries containing collections (and/or SQL tables).
Add Definition	This option displays the Add Job Watcher Definition Wizard. The definition defines characteristics about the collection such as which data options to collect.
Start Collection	This menu will open the Start Job Watcher Collection Wizard where the user can kick off a collection using the desired Job Watcher definition. **Note:** A Job Watcher definition must be defined first before creating a collection unless you wish to use an IBM-supplied definition.
Start Monitor	This menu will open the Start iDoctor Monitor Wizard where the user can start a Job Watcher, PEX Analyzer or Disk Watcher monitor. Monitors are designed to provide 24x7 collection of performance data.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.
Set User-Defined Reports Database	This option allows the user to view/modify the currently used user-defined reports database. The database can either be an MS Access file or a library on an IBM i. The database stores the information needed to build the user-defined tables and graphs shown in iDoctor.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Work with iDoctor scheduled jobs	This option is a shortcut to the General functions –> Work management -> Scheduled jobs folder. It shows all the iDoctor created scheduled jobs that exist on the current IBM i system.
Collections database	The iDoctor collection database identifies all collections on the system and can be used to facilitate the drill down from one component to another in some situations.
Properties	Use this menu to display version information for the current component installed on the current system. The build level of the GUI is also displayed here.

8.3 Libraries

This folder contains the libraries on the system that contain Job Watcher data. This list displays each library's name and description. By clicking on a library in the tree you will see any Job Watcher collections that exist within the library.

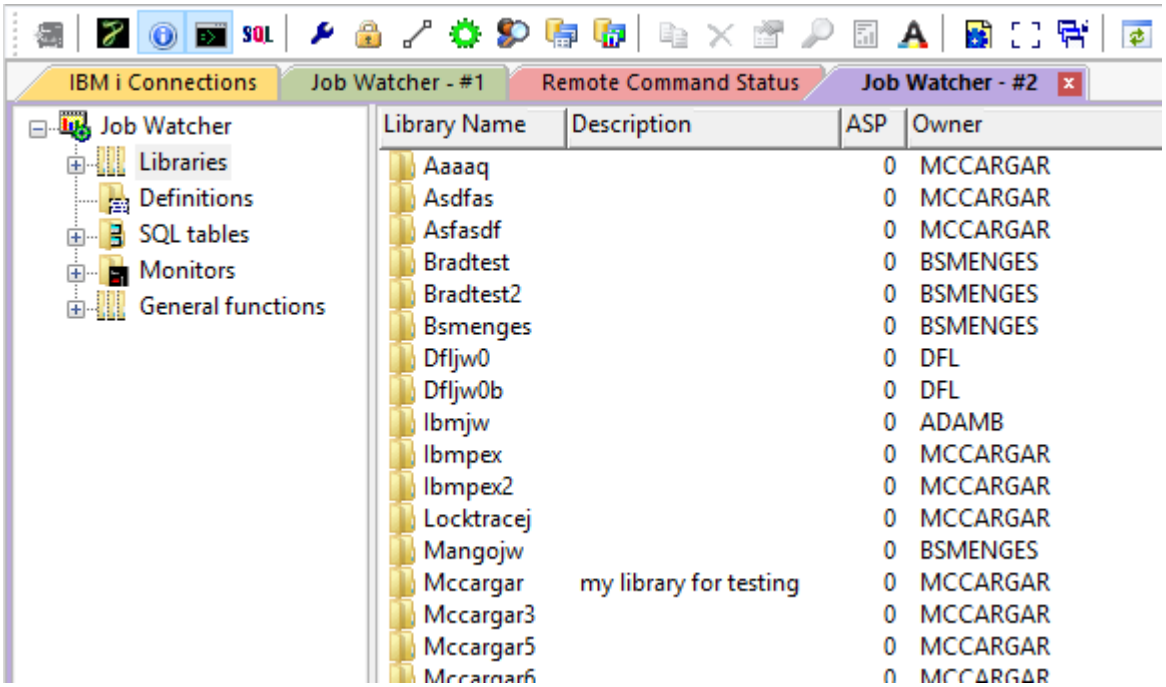
8.3.1 Menu Options

Right-clicking on the Libraries folder provides these options:



Libraries folder menu options

Menu	Description
Explore	This option will display the list of libraries on the system matching any library filters defined by the Filter libraries window.
Select fields...	This will display the Field selection for 'Libraries' window and allows the user to control which columns appear in the list of Libraries and their order.
Filter libraries…	This option allows you to filter the libraries shown in the Libraries and SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of the list if the system contains many libraries containing collections (and/or SQL tables).
Start Collection	This menu will open the Start Job Watcher Collection Wizard where the user can kick off a collection using the desired Job Watcher definition. **Note:** A Job Watcher definition must be defined first before creating a collection unless you wish to use an IBM-supplied definition.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.



Libraries folder in the Job Watcher Component View

8.3.2 Library Menu Options

The following Job Watcher specific menu options are available by right clicking on a library in the component view.

Menu Item	Description
Start Collection	This menu will open the Start Job Watcher Collection Wizard where the user can kick off a collection using the desired Job Watcher definition. **Note:** A Job Watcher definition must be defined first before creating a collection unless you wish to use an IBM-supplied definition.

Additional menu options that are common to all library folders in iDoctor are discussed here.

8.4 Monitors

Job Watcher monitors allow for 24x7 collection of Job Watcher data on a system. They run continuously storing only the most recent collections desired. Job Watcher monitors will run until ended manually by the user.

The following green screen commands are used in library QIDRWCH to work with Job Watcher monitors:

Command	Description
STRJWMON	This will start or restart a Job Watcher monitor.
HLDJWMON	This will hold the Job watcher monitor. The monitor job remains active, but no new data will be captured until the RLSJWMON command is used to release it.
RLSJWMON	This command is used to release a Job Watcher monitor that has been previously held.
DLTJWMON	This command is used to remove a Job Watcher monitor and all the collections within it from the system.

For more information about Job Watcher monitors in the GUI, see the section on Monitors in chapter 4.

8.5 SQL Tables

This folder contains all the SQL tables that exist on the system generated by Job Watcher analyses.

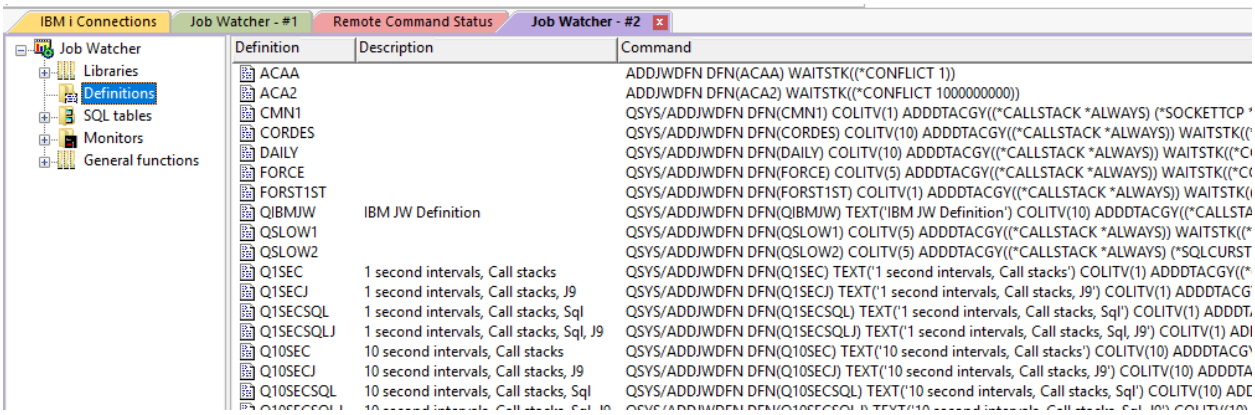
The folder exists in 3 places and each will filter the contents appropriately based on where it is located:

Location	Description
Under Job Watcher	Entire system
Under a library	All collections in the library
Under a collection	Only this collection

For more information see the SQL Tables section in chapter 4.

8.6 Definitions

A Definitions folder is provided in Job Watcher to allow the user to work with the Job Watcher definitions that exist on the current system. An example of this interface is:



Job Watcher Definitions Folder

The fields shown in this view are as follows:

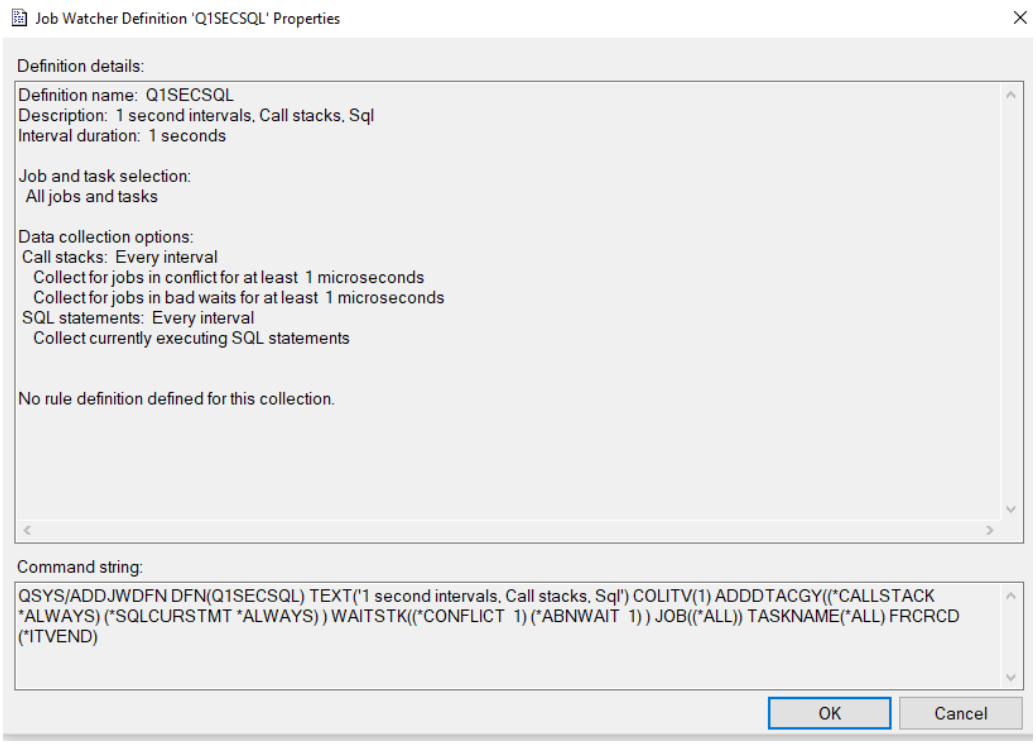
Field	Description
Definition	Name of the definition. IBM-supplied definitions begin with Q.
Description	An optional description given to the Job Watcher definition
Command	The command string used to create the definition.

The following options are available when right clicking on one or more definitions in the list:

Field	Description
Change Definition	Opens the Add Job Watcher Definition Wizard and loads the selected definition into it so it can be changed.
Add Definition	Opens the Add Job Watcher Definition Wizard in order to create a new definition.
Start Collection	Opens the Start Job Watcher Collection Wizard using the selected definition.
Start Monitor	Opens the Start iDoctor Monitor Wizard using the selected definition.
Delete	Removes the selected definitions from the system.
Properties	Displays the properties for the selected Job Watcher definition.

8.6.1 Properties

Double-clicking on a definition or using the Properties menu from the Job Watcher Definitions View displays all parameters that were used when creating the definition. An example of this interface is:



Job Watcher Definition Properties

Advanced users can change the command string that defines the definition if desired. If changes have been made to the command string, pressing the OK button will remove the existing definition from the system and replace it using the command string specified.

Note: IBM-supplied definitions cannot be changed.

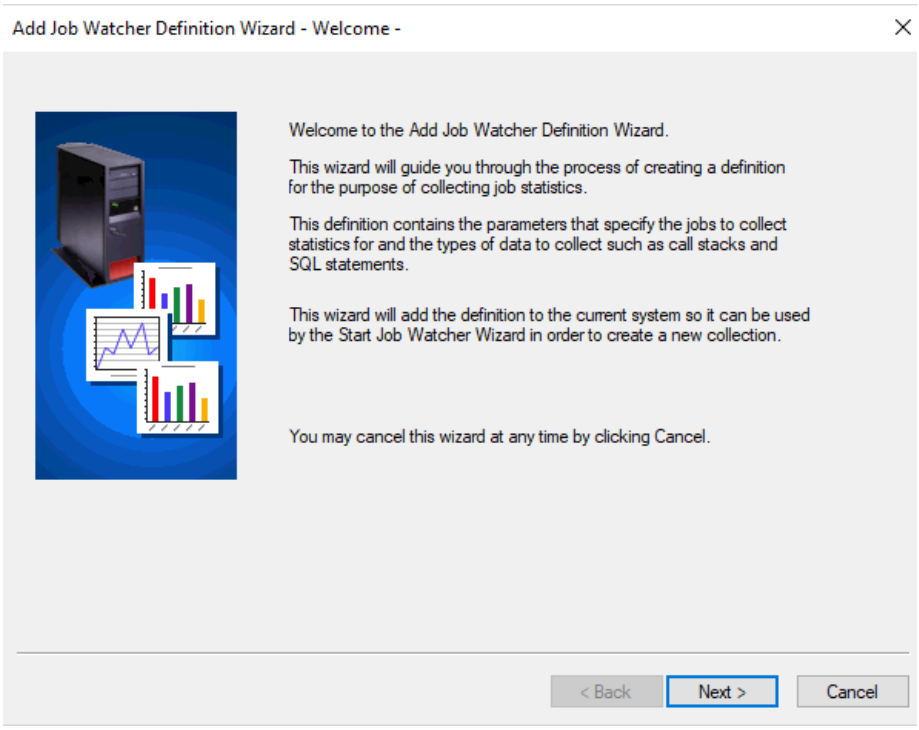
8.7 Add Job Watcher Definition Wizard

Collection Job Watcher data requires the use of a Job Watcher definition. These definitions include all the parameters that define which types of data to include in the collection. A user can use either the IBM-supplied definitions or create their own by adding a definition to the system using this interface.

Note: This is an interface over the IBM i command ADDJWDFN.

8.7.1 Welcome

The Welcome page in the Add Job Watcher Definition Wizard introduces the user to the wizard and explains what the wizard will do.



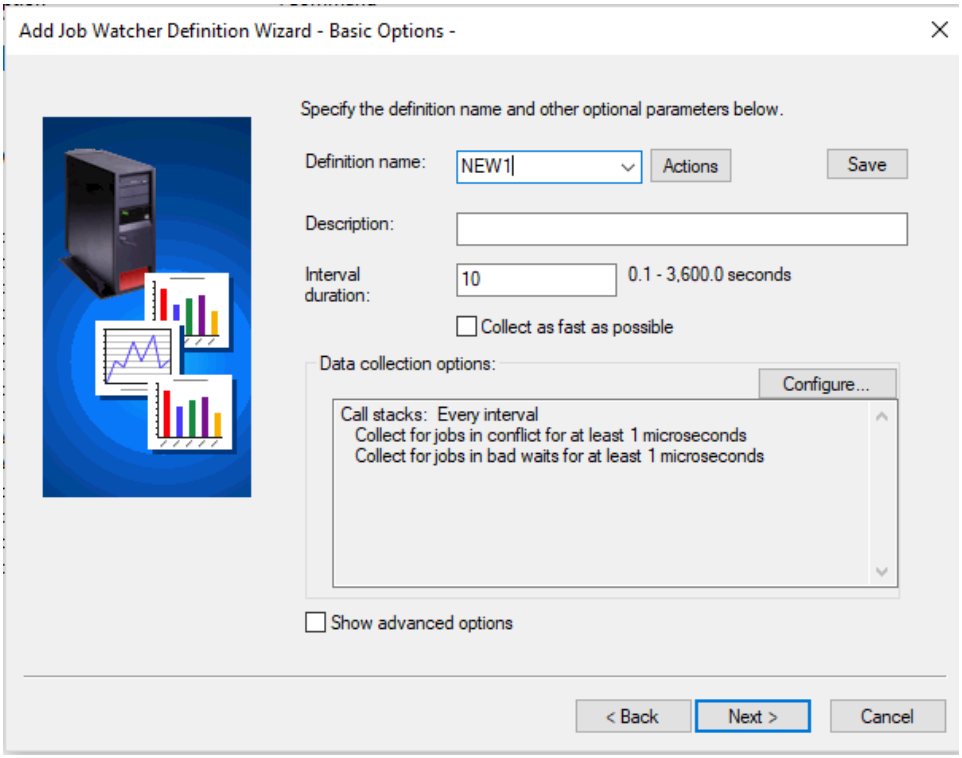
Add Job Watcher Definition Wizard – Welcome

8.7.2 Basic Options

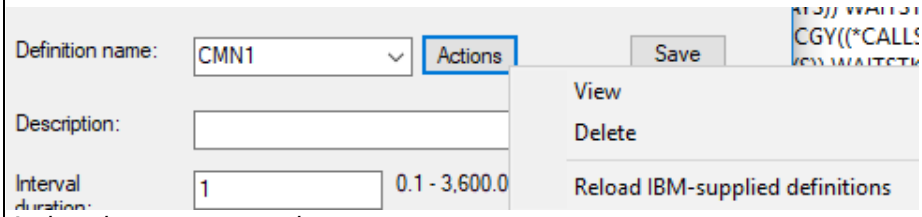
The basic options page in the Wizard allows you to enter the definition's name, description and interval duration. It also provides information about the data collection options selected with a button to configure them.

If you wish you can change a definition using this interface by selecting a different definition on the system from the drop-down list. This action will discard all changes made into this interface and load the parameters for the definition selected into the Wizard.

An example of the Basic Options panel is shown below:



Add Job Watcher Definition Wizard – Basic Options

Option	Description
Definition name	This field allows you to either provide the definition name of the definition to add to the system or select from the existing definitions on the system in order to make updates the definition.
Actions button	Click this button to display a menu of options relating to Job Watcher definitions:  *Actions button menu options* View – Displays the properties for the definition. Only works if changing an existing definition. Delete – Removes the definition from the system. Reload IBM-supplied definitions – Deletes and replaces all IBM-supplied Job Watcher definitions on the system.
Save	The save button will add or update the current definition on the system in its currently defined state within this interface. This also occurs automatically when going through the wizard and hitting Finish on the last page.
Description	The 50-character text description that describes the definition.
Interval duration	The size of each sample of data in seconds. **Note:** If the collect as fast as possible option is checked then this value is greyed out and is not applicable.
Collect as fast as possible	Check the collect as fast as possible button to collect the next snapshot immediately after the previous one finishes (no delay). Keep in mind this option can be very resource intensive so use caution!
Data collection options	This section lists the data collection options that are currently defined for the current definition and a button to enter the Data Collection Options window in order to configure them.
Show advanced options	When checked the Advanced options page of the Wizard will be shown when the Next button is pressed. This screen contains options that are less commonly used.

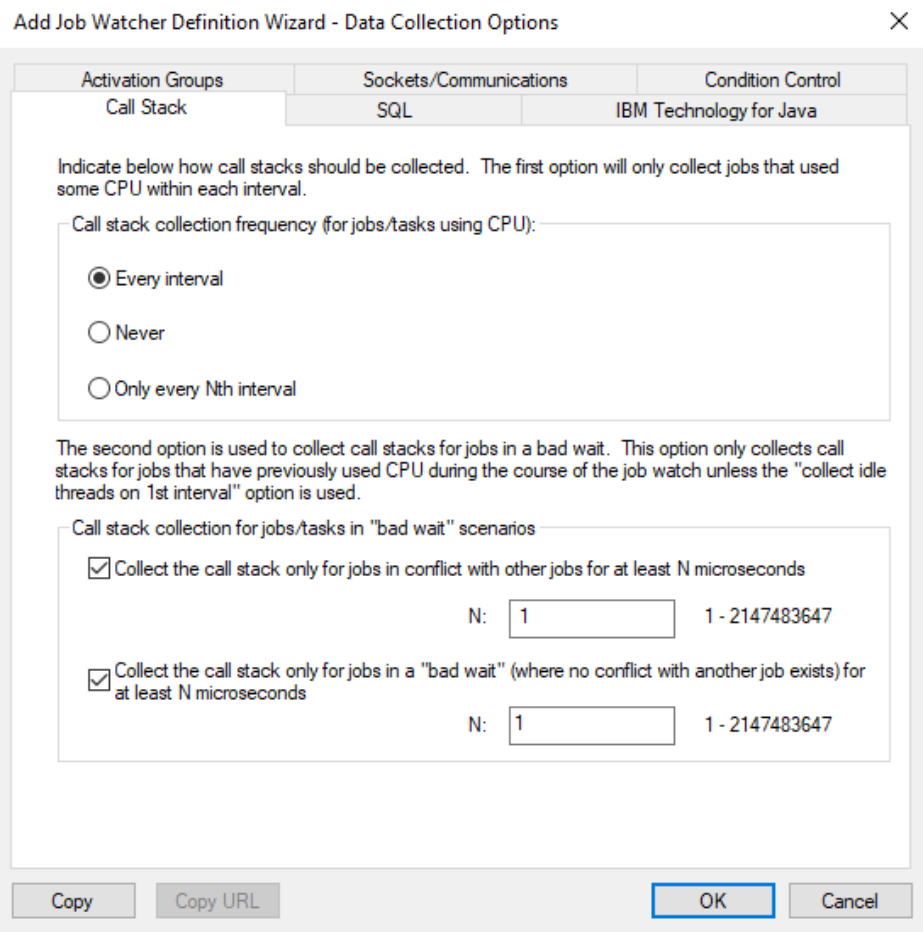
8.7.3 Data Collection Options

The Data Collection Options interface allows you to specify which types of data Job Watcher should collect. The types of data that may be collected are broken up into several different panels.

8.7.3.1 Call Stack

The call stack tab allows you to specify whether call stacks should be collected and how often. Job Watcher normally only collects call stacks for jobs that used CPU during the interval collected. You may also indicate if call stacks should be collected for jobs that are experiencing performance issues even though no CPU was used.

An example of this interface is the following:



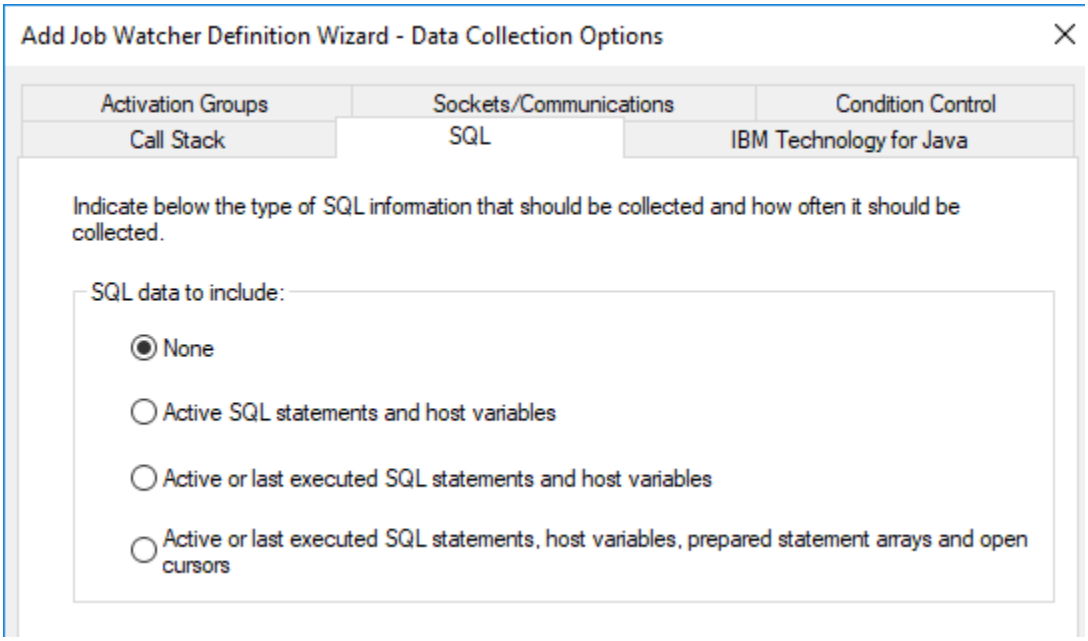
Add Job Watcher Definition Wizard – Data Collections Options – Call Stack

Option	Description
Every interval	The call stack will be harvested every interval for every job in the collection that used CPU during each interval.
Never	The call stack will not be collected.
Only every Nth interval	The call stack will only be harvested for jobs using CPU every Nth interval. Selecting this option will display a field where the value for N can be entered. If the value for N is 5 then only jobs that used CPU every 5th interval of the collection will include call stacks.
Collect call stacks for jobs in conflict	This option indicates if call stacks should be collected for jobs that are in conflict with other jobs. The value N defines how long the job needs to have been in conflict for the call stack to be collected. N is specified in microseconds.
Collect call stacks for jobs in bad waits	This option indicates if call stacks should be collected for jobs that are in bad waits. The value N defines how long the job needs to have been in a bad wait in order for the call stack to be collected. N is specified in microseconds.

8.7.3.2 SQL

This page allows the user to define the options for collecting SQL statements for jobs included in the collection.

SQL statements are created into file QAPYJWSQL. Host variables for SQL statements are created in QAPYJWSQLH. QAPYJWSQLO and QAPYJWSQLP contain open cursor lists and prepared statement areas if the most detailed choice is selected.



Add Job Watcher Definition Wizard – Data Collections Options – SQL

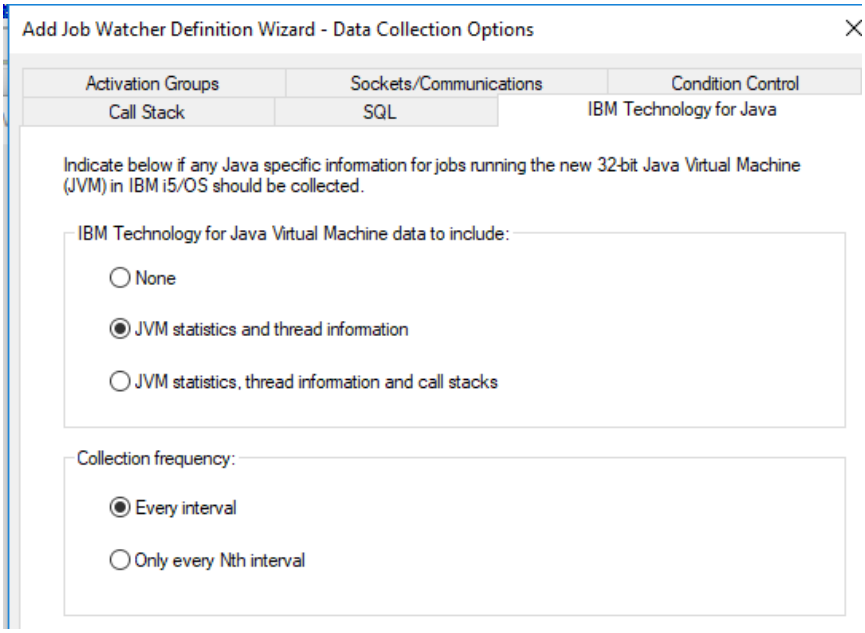
Option	Description
None	No SQL statements collected. This is the default.
Active SQL statements and host variables	SQL statements will be collected for any jobs that are currently running SQL statements (at the moment each interval is harvested) within the collection. If this option is used it's quite possible not to get any SQL information if the statements that are running complete
Active or Last executed SQL statements and host variables	This option will collect the last executed SQL statement and host variable for every job in the collection, for every interval the job is active. **Tip:** For most users, this is the recommended choice if you wish to collect SQL statements.
Last executed SQL statements, host variables, prepared statement areas	This option will collect the last executed SQL statement and host variable for every job in the collection, for every interval the job is active. In addition, this option will collect information about the prepared statement areas and open cursors for the job running the SQL statement.
SQL collection frequency	If one of the above SQL collection options is selected, this option allows the user to determine how often the SQL data should be collected.

8.7.3.3 IBM Technology for Java

This page allows for the collection of IBM Technology for Java Virtual Machine statistics and thread data. IBM Technology for Java is also known as J9 and is the new 32-bit JVM.

JVM statistics for J9 are written to file QAPYJWIJVM. JVM thread data is written to file QAPYJWIJVT.

If J9 call stacks are collected they are written to file QAPYJWIJVS. Call stacks for J9 jobs are not collected in the regular call stack file QAPYJWSTK.

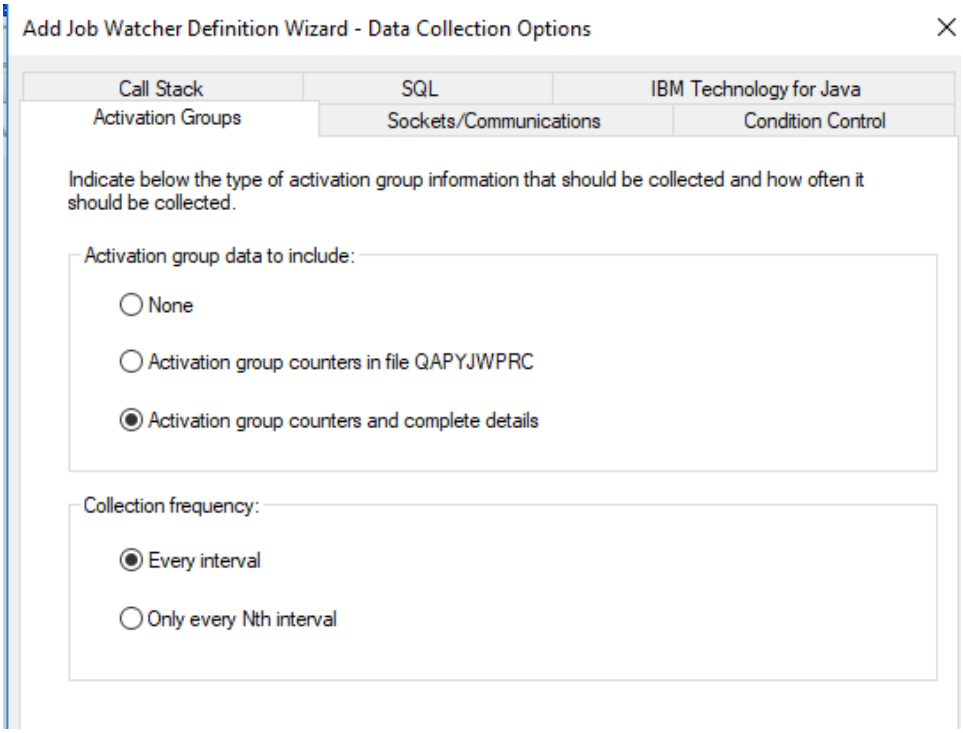


Add Job Watcher Definition Wizard – Data Collections Options – IBM Technology for Java

Option	Description
None	No J9 JVM information will be collected. This is the default.
JVM statistics and thread information	J9 JVM statistics and thread information will be collected.
JVM statistics, thread information and call stacks	J9 JVM statistics, thread information and J9 call stacks will be collected.
Collection frequency	If one of the above J9 collection options is selected, this option allows the user to determine how often the J9 data should be collected.

8.7.3.4 Activation Groups

This page allows the user to define the options for collecting activation group information for jobs included in the collection.



Add Job Watcher Definition Wizard – Data Collections Options – Activation Groups

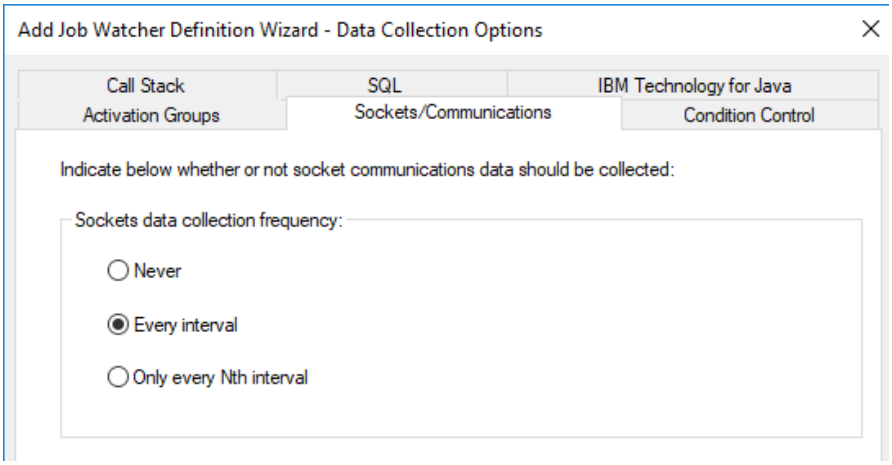
The following table describes the parameters available on this page of the Wizard.

Option	Description
None	No activation group data collected
Activation group counters in file QAPYJWPRC	If this option is selected, the counters in file QAPYJWPRC (the job/process information file) will be filled. The fields that will be filled are: CURNUMACTG (current number of activation groups) and CURNUMACT (current number of activations)
Activation group counters and complete details	This option will collect the activation group counters in the QAPYJWPRC file as well as additional files containing complete information about the activation groups for all jobs included in the collection. The files filled by this option are: QAPYJWAIGP - general activation group information QAPYJWAIHP - activation group heap sizes and counts QAPYJWAIPA - list of programs in each activation group collection
Collection frequency	If one of the above activation group collection options is selected, this option allows the user to determine how often the activation group data should be collected.

8.7.3.5 Sockets/Communications

This page allows the user to capture communications and socket information for jobs running in the collection. Socket data is collected into files QAPYJWSKTC and QAPYJWSKJB.

An example of this window is shown below:

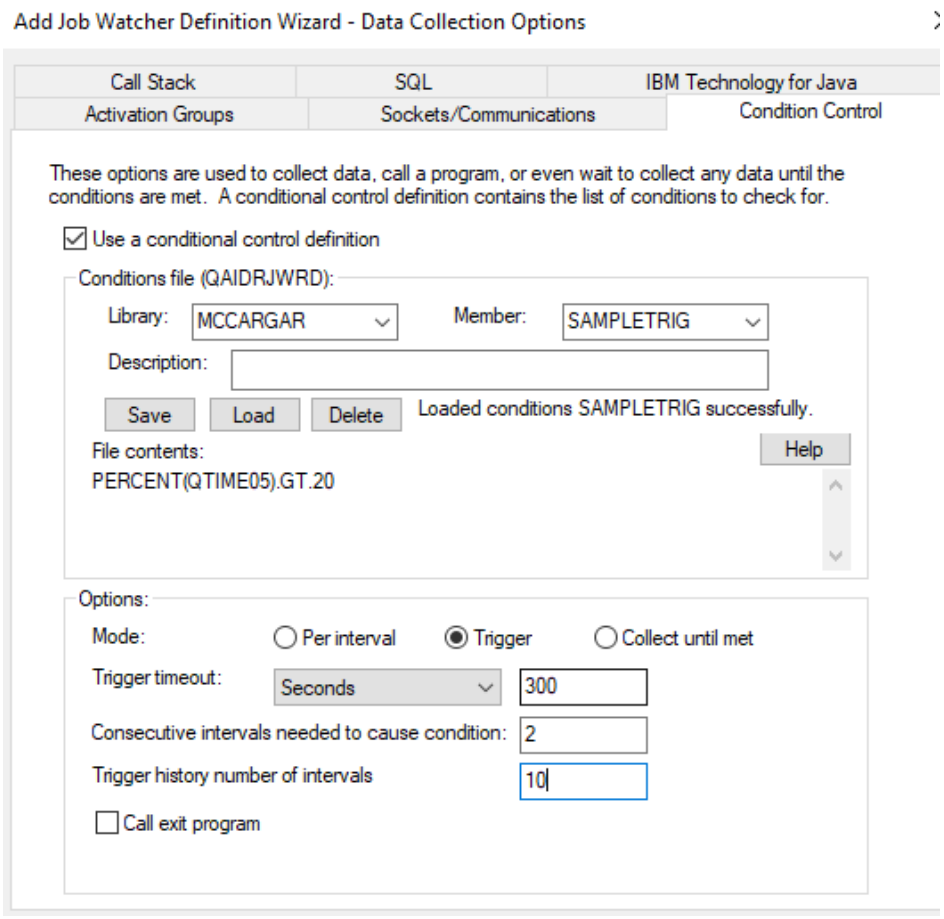


Add Job Watcher Definition Wizard – Data Collections Options – Sockets/Communications

Option	Description
None	No socket data will be collected
Every interval	Socket information will be collected every interval
Only every Nth interval	Socket information will be collected every Nth interval

8.7.3.6 Condition

This page is used to define a conditional control file for the collection. This file is used to collect data or perform unique actions based on criteria encountered during collection. The control file definitions are saved into file QAIDRJWRD. An example of creating a rule definition via the green screen is available in file QAIDRJWRD in library QIDRWCH.



Add Job Watcher Definition Wizard – Data Collections Options – Condition Control

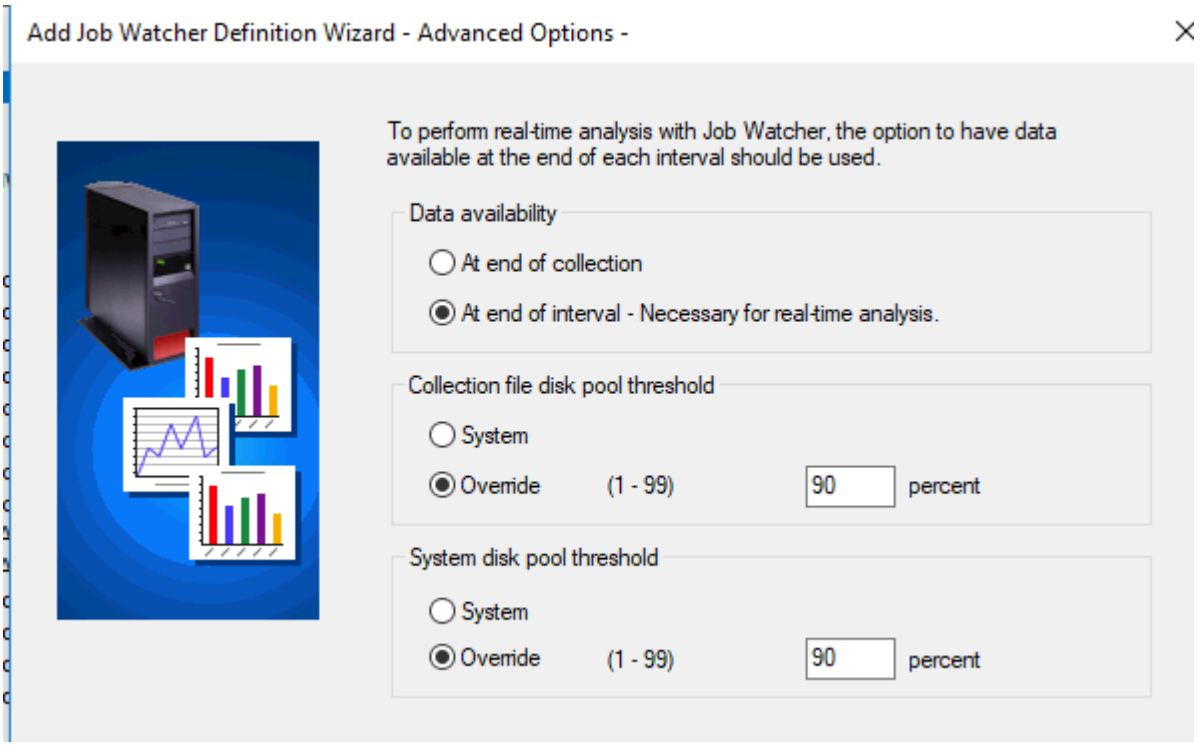
Option	Description
Use a conditional control	Check the box to define a conditional control file to use for this Job Watcher definition. Uncheck a box will remove it.
Library	List of the libraries found on the system containing existing conditional control files named QAIDRJWRD. The value is editable. To save the rule definition into a new library, type the library name into this field before pressing the Save button.
Member	Within the current library selected, the definition (member) names that were found. The value is editable. To save a new rule definition, provide the name into this field before pressing the Save button.
Description	Description of the conditional control member.
Save	This will save the QAIDRJWRD file member containing the contents of the File Contents text box currently shown.
Load	This will load the currently specified QAIDRJWRD file member into the File contents field.
Delete	This will remove the currently specified QAIDRJWRD file member from the library indicated.
Help	This will display additional information about how to define the conditions file. The contents of this information are copied below: The conditions control file must be a source physical file and may be used to specify conditions that Job Watcher will use to limit data collection. If a file is specified on this parameter, the data collected will be compared against the conditions defined in the file. Conditions must be specified in a specific format. A description of valid conditions and formats follows: Direct field comparison Syntax: FIELDNAME.COMPARAND.VALUE Example 1: The condition will be met when more than 75 synchronous database writes occur in the interval SYNDBWRT.GT.75 Example 2: The condition will be met when the wait time in bucket 6 is between 30 and 80 microseconds. QTIME06.GE.30.AND.QTIME06.LE.80 Rate condition Syntax: RATE(FIELDNAME).COMPARAND.VALUE Example: The condition will be met when the rate of synchronous database writes is greater than 10 per second RATE(SYNDBWRT).GT.10 Percent condition (applies to time spent in a particular wait bucket) Syntax: PERCENT(FIELDNAME).COMPARAND.VALUE Example: The condition will be met when more than 10 percent of time spent waiting was counted in bucket 9 PERCENT(QTIME09).GT.10 Average condition Syntax: AVERAGE(FIELDNAME1,FIELDNAME2).COMPARAND.VALUE where FIELDNAME1 is a time and FIELDNAME2 is a corresponding count

	Note: At this time the only time/count combinations reported in Job Watcher are the wait bucket times and counts reported in the QAPYJWTDE file. Example: The condition will be met when the average wait time for a wait counted in bucket 5 is greater than 50 microseconds AVERAGE(QTIME05,QCOUNT05).GT.50
File contents	This is the contents of the conditional control file. This should list one or more conditions to check for in the Job Watcher data based on the Help file covered previously.
Mode	Displays and configures the type of conditional collection to use. **Per interval** The specified condition will be checked in every interval. In this type of collection, data will only be written to the database files for intervals in which the condition was satisfied. If an exit program is specified on the on the User exit program (EXITPGM) parameter it will be called in each interval where the condition was satisfied. **Trigger** The specified condition will be checked in each interval until the condition is satisfied. Once the condition has been met, data will be unconditionally written to the database files for the remainder of the written to the database files for the remainder of the collection. If an exit program is specified on the on the User exit program (EXITPGM) parameter it will be called one time in the interval where the condition was satisfied. **Collect until met** Data will be unconditionally written to the database files until the condition is satisfied. Once the condition has been met, the collection will end. If an exit program is specified on the on the User exit program (EXITPGM) parameter it will be called one time before the collection ends.
Trigger timeout	When using Trigger mode, this specifies how long the collection should run without writing any data to the database files. The value is specified in either seconds or intervals.
Consecutive intervals needed to cause condition	This indicates how many intervals are needed to cause the condition to occur.
Trigger history number of intervals	Specifies the amount of data (in intervals) that should be buffered as history during the conditional collection. The specified amount of data will be maintained until the condition has been satisfied, at which time all buffered data will be written to the database files along with the data from the current interval.
Call exit program	Check the box to indicate that a user-defined program will be called. By checking the box, you can then enter the library / program name as well as the desired value to pass to parameter 1 of the program.

8.7.4 Advanced Options

The Advanced Options page in the Add Job Watcher Definition Wizard allows the user to configure options that are normally only needed in rare circumstances.

An example of this screen is the following:



Add Job Watcher Definition Wizard – Advanced Options

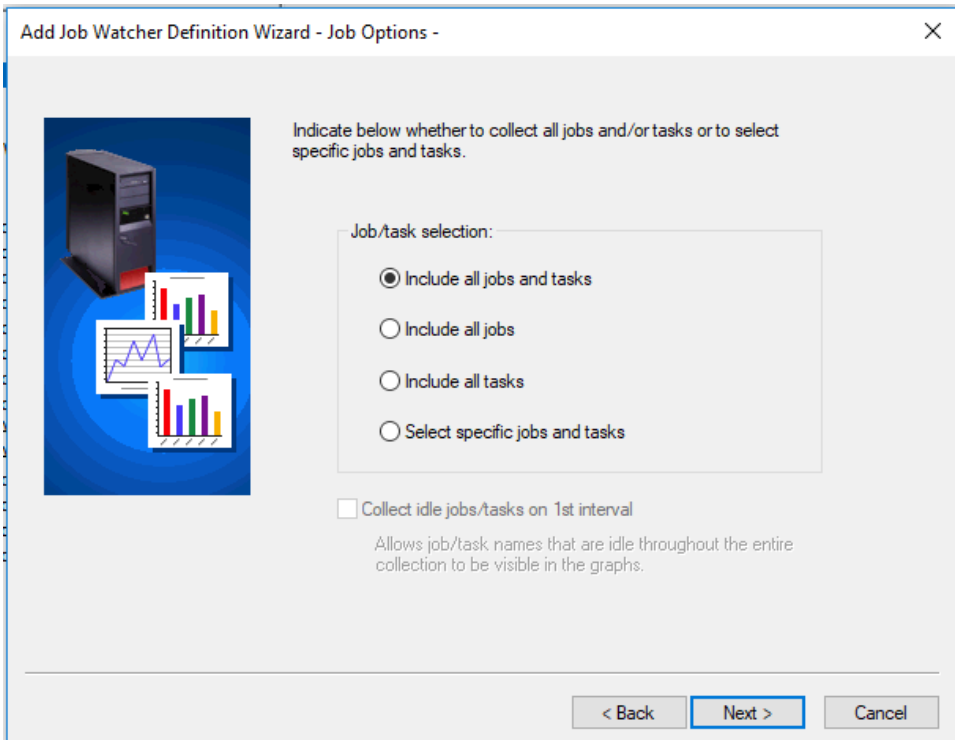
GUI Element	Description
Data availability	Indicates how soon the collection data will be ready for use. Job Watcher has the capability to collect data for several intervals before actually writing any data to the database files. There are some slight performance gains possible in the collection by specifying “At end of collection”, but the downside to doing this is the data may only exist in the database files until after the collection has ended.
Collection file disk pool threshold	Specifies the percentage of the auxiliary storage pool (ASP) that contains the Job Watcher database files that can be used before the collection is forced to end. Use the Change Storage Threshold function of the Start System Service Tools (STRSST) command in order to change the system threshold for an ASP.
System disk pool threshold	Specifies the percentage of the system auxiliary storage pool (ASP) which can be used before the collection is forced to end. Use the Change Storage Threshold function of the Start System Service Tools (STRSST) command in order to change the system threshold for the system ASP.

8.7.5 Job Options

This page allows the user to determine whether all jobs/tasks should be collected, or if specific jobs and tasks should be collected. If the option "Select specific jobs and tasks" is selected then the job/task

selection page will be shown next in order for the user to define which jobs and/or tasks should be collected.

An example of this window is shown below:



Add Job Watcher Definition Wizard – Job Options

The following table describes the parameters available on this page of the Wizard.

Note: Active jobs/tasks are defined as those jobs or tasks that used the CPU for each interval collected.

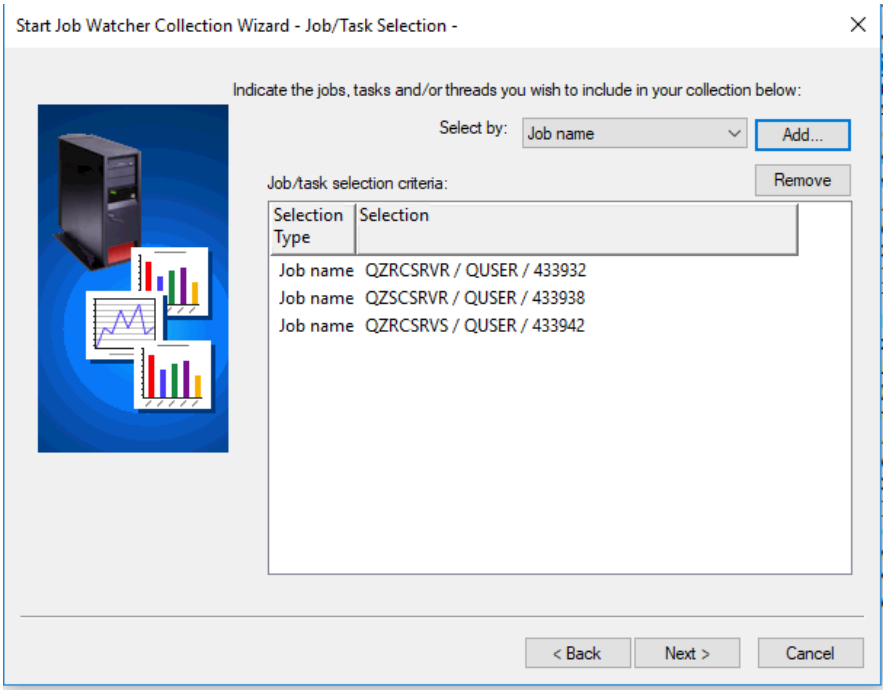
Option	Description
Include all jobs and tasks	All "active" jobs and tasks running on the system will be collected.
Include all jobs	All "active" jobs running on the system will be collected
Include all tasks	All "active" tasks running on the system will be collected.
Select specific jobs and tasks	Selecting this option will display the Job/task selection page when the 'Next' button on the Wizard is pressed. This window provides many ways to select or filter which jobs/tasks to collect among the jobs/tasks running on the system.
Collect idle jobs/tasks on 1st interval	This option will collect an interval of data for every job/task found within the collection regardless if the job/thread/task used CPU or not. Normally data is not collected for jobs and tasks that did not use CPU during an interval. If a job never uses CPU throughout the entire collection the job name and thread ID will not be displayable in the reports unless this option is used.

8.7.6 Job/task selection

This window provides the user with the ability to select the jobs and tasks to include in the collection. There are six different ways to select the jobs/tasks to use in the collection: Job name, task name, current user profile, subsystem, pool ID, and taskcount. These options are listed within the select by drop

down list. After making the selection in the list, pressing the Add... button will display the appropriate interface in order to make the selection and add it to the list of job/task selection criteria.

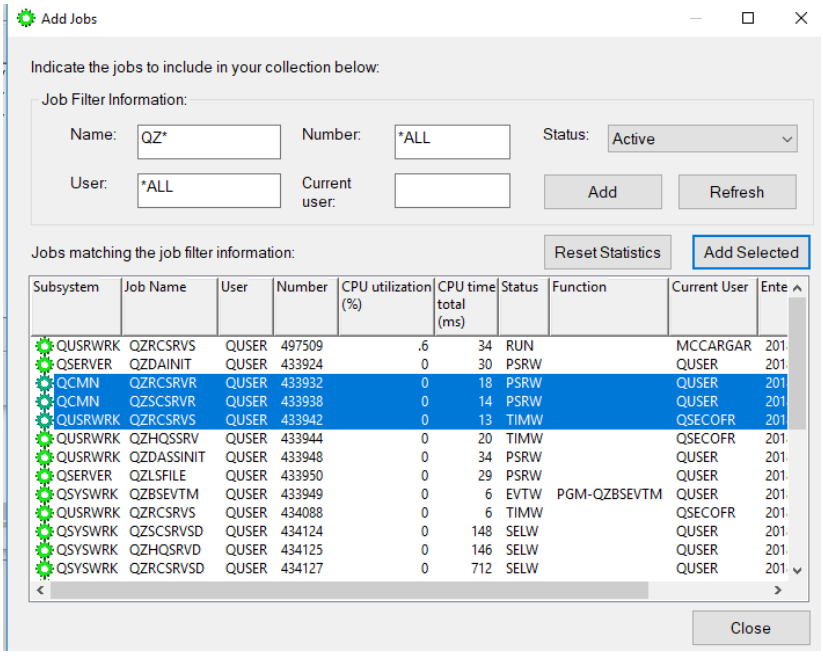
An example of this page of the Wizard is:



Add Job Watcher Definition Wizard – Job/Task Selection

8.7.6.1 Job name selection

Pressing the Add... button while "Job name" is selected in the Select by drop down list will display the following window:



Add Job Watcher Definition Wizard – Add Jobs Window

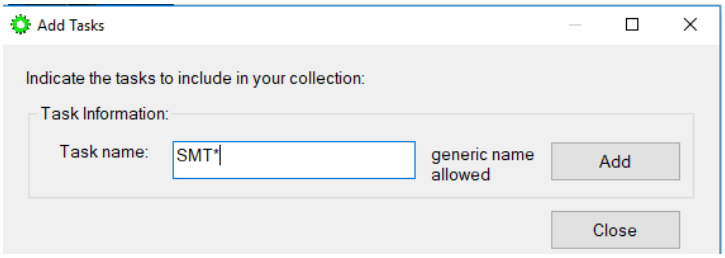
This window displays the list of jobs on the system and allows the user to add generic or specific job names to the job/task selection criteria list on the job/task selection page of the Wizard.

The following table describes the fields on this window:

Option	Description
Job Filter information: Job Name	This field is used to specify a generic job name. This job name may be used to either display a list of active jobs running on the system that match the generic name (by pressing the Refresh button), or add a job/task selection criteria using a generic name (by pressing the Add button).
Job Filter information: Job User	This field is used to specify a generic job user name. This job user name along with the job name filter may be used to either display a list of active jobs running on the system that match the generic job user name (by pressing the Refresh button), or add a job/task selection criteria using a generic job user name (by pressing the Add button).
Job Filter information: Job Number	This field is used to specify the job number to use when either filtering the list of active jobs or adding a job selection criteria to the job/task selection page of the Wizard.
Job Filter information: Current user	Indicates the current user profile to use when displaying the list of active jobs. This option only applies to the "Refresh" button for updating the active list of jobs to select from and does not apply to the Add... button (can't select jobs by current user profile using the Add button). To select all jobs for a specific user profile use the "current user profile" selection type on the Job/Task selection page of the Wizard.
Add	This button will add the currently specified job information filter (job name, job user and job number) to the list of job/task selection criteria on the Job/Task selection page of the Wizard. This option does not apply to the current user field.
Refresh	This button will update the list of "jobs matching the job filter information".
Jobs list	This is the list of jobs matching the job name, job user, job number and current user profile specified. This list may be used to select individual jobs to collect in the job watch.

8.7.6.2 Task name selection

Pressing the Add... button while "Task name" is selected in the Select by drop down list will display the following window.



Add Job Watcher Definition Wizard – Add Tasks Window

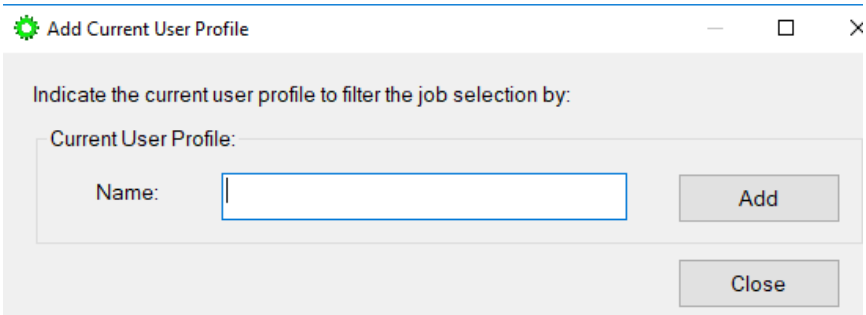
This window displays a field to specify a generic task name to include in the job/task selection criteria list on the job/task selection page of the Wizard.

The following table describes the fields on this window:

Option	Description
Task name	This field is the generic task name. Pressing the Add button will add the generic task name to the list on the Job/task selection page of the Wizard. This field could also contain a specific task name if it is keyed in correctly, but there is not an option to view the list of active tasks from this window.

8.7.6.3 Current user profile selection

Pressing the Add... button while "Current user profile" is selected in the Select by drop down list will display the following window.



Add Job Watcher Definition Wizard – Add Current User Profile Window

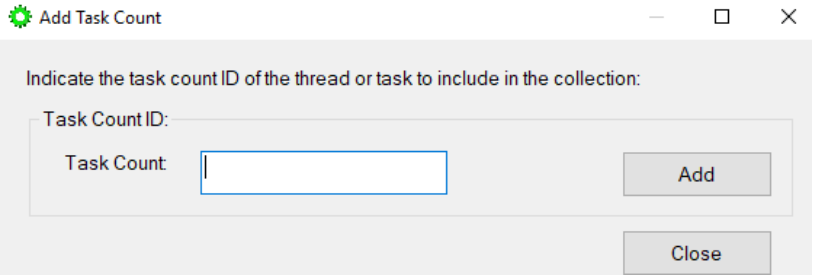
This window displays a field to specify a current user profile name to include in the job/task selection criteria list on the job/task selection page of the Wizard.

The following table describes the fields on this window:

Option	Description
Current user profile name	This field is for entering the current user profile to collect job information for. Generic names are not allowed for this field.

8.7.6.4 Task count selection

Pressing the Add... button while "Task count" is selected in the Select by drop down list will display the following window:



Add Job Watcher Definition Wizard – Add Task Count Window

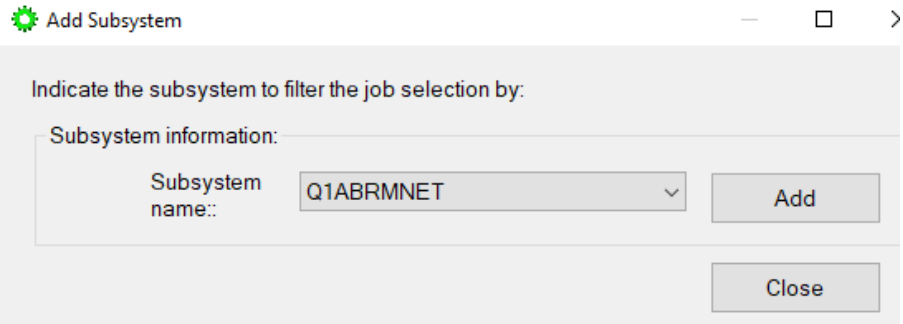
This window displays a field to specify the task count to include in the job/task selection criteria list on the job/task selection page of the Wizard.

The following table describes the fields on this window:

Option	Description
Task count	This field is the task count which uniquely identifies a job/thread or task on a system. The task count must be entered in 16 character HEX format.

8.7.6.5 Subsystem name selection

Pressing the Add... button while "Subsystem" is selected in the Select by drop down list will display the following window:



Add Job Watcher Definition Wizard – Add Subsystem Window

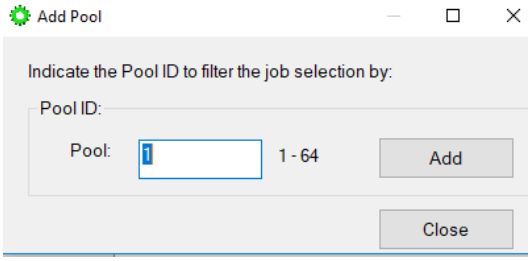
This window displays a list of subsystems that are running on the system to select from. By selecting a subsystem this indicates that all jobs that running in that subsystem will be included in the collection (if not filtered out by other parameters which may also be used).

The following table describes the fields on this window:

Option	Description
Subsystem	Contains a list of active subsystems. Clicking the Add button will add the selected subsystem to the list on the Job/task selection page.

8.7.6.6 Pool ID selection

Pressing the Add... button while "Pool ID" is selected in the Select by drop down list will display the following window:



Add Job Watcher Definition Wizard – Add Pool Window

This window allows the user to select the jobs/tasks to include in the job watch by the pool the jobs/tasks are running in.

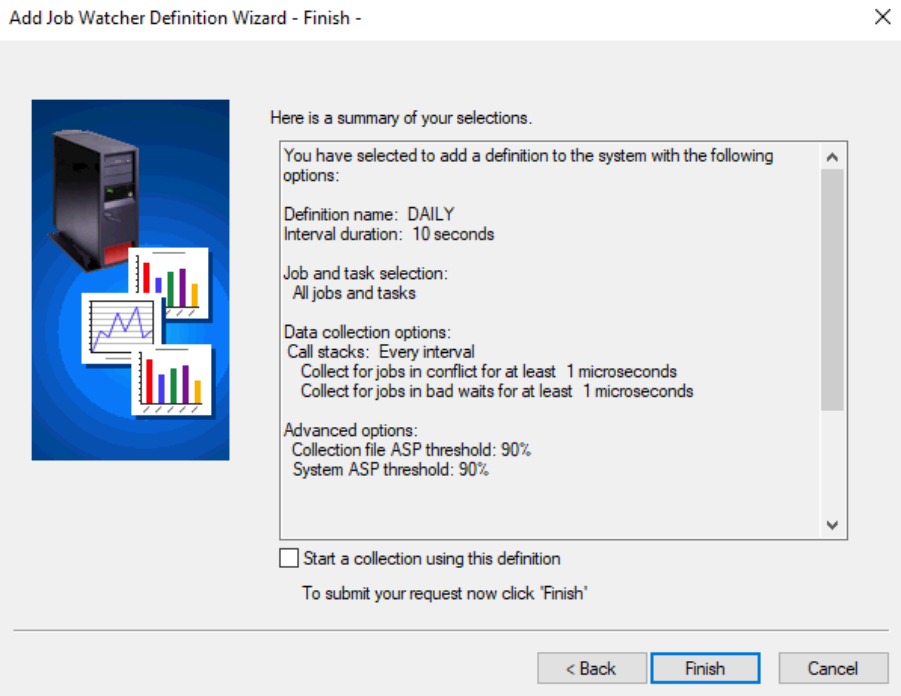
The following table describes the fields on this window:

Option	Description
Pool ID	This field contains the desired pool ID to collect job/task/threads from. Clicking the Add button will add the selected pool information to the list on the Job/task selection page of the Wizard.

8.7.7 Finish

The Finish page provides complete details about all selections made in the wizard. If anything listed doesn't look right, use the Back button to go back and make any changes necessary. After clicking 'Finish' the command (ADDJWDFN) to add the definition to the system will be issued. The command string is listed at the bottom this page and can be copied to a green screen session and modified if necessary.

An example of this interface is:



Add Job Watcher Definition Wizard – Finish

Note: Click the “Start a collection using this definition” option in order to launch the Start Job Watcher Collection Wizard right after the definition is added to the system.

8.8 Start Job Watcher Collection Wizard

Job Watcher provides the capability to collect detailed information about all jobs and tasks on the system.

This section covers the creation of a collection using the Start Job Watcher Collection Wizard. The Wizard is accessible via the Start Collection menu on the Job Watcher or library folder icons. This Wizard guides the user step by step through the process of creating a collection. Each page is covered in detailed within the next sections.

Tip: If Job Watcher data already exists in the library it must match the currently installed OS release of IBM i or you will be unable to collect more data in that library. You cannot combine data of different releases in the same library.

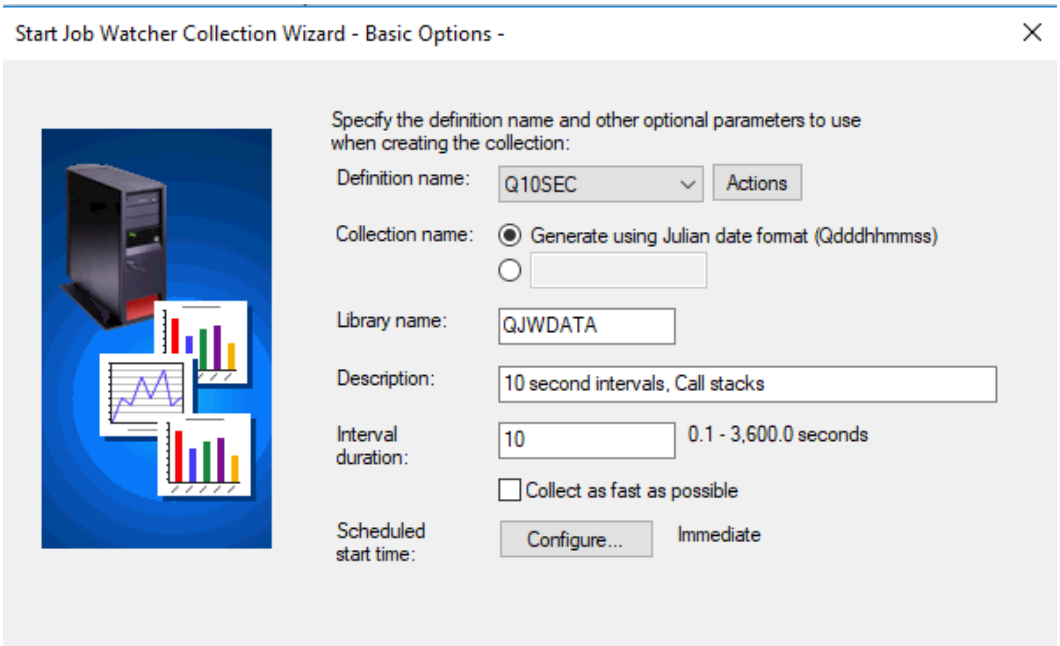
8.8.1 Welcome

The Welcome page in the Start Job Watcher Wizard introduces the user to the wizard and offers information about what it will do.

Tip: Starting a collection requires a definition. Use the Add Job Watcher Definition Wizard first if you do not wish to use the IBM-supplied definitions.

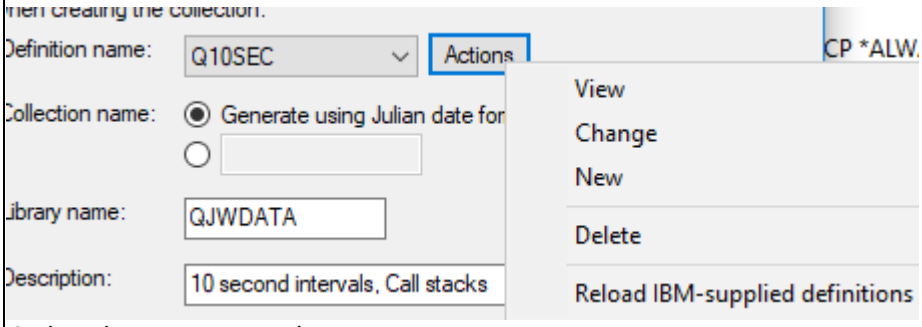
8.8.2 Basic Options

The Basic Options Page allows the user to specify the collection name, definition name, library, interval duration, and description as well as scheduling options. The following is an example of this interface:



Start Job Watcher Collection Wizard – Basic Options

The following table provides details about each of the parameters on this page:

Field	Description
Definition name	The name of the definition to create the collection with. A Job Watcher definition contains most of the data collection options that define the types of information to include in the collection. IBM-supplied definitions begin with Q.
Actions button	Click this button to display a menu of options relating to Job Watcher definitions:  *Actions button menu options* **View** – Displays the properties for the definition. Only works if changing an existing definition. **Change** – This will open the Add Job Watcher Definition Wizard and load the selected definition's settings into it. **New** – This will open the Add Job Watcher Definition Wizard in order to create a new definition. **Delete** – Removes the definition from the system. **Reload IBM-supplied definitions** – Deletes and replaces all IBM-supplied Job Watcher definitions on the system.
Collection name	The name of the collection to create. This name matches the member name used when creating the Job Watcher QAPYJW* database files on the server. If you want the system to generate a collection name for you, use the "Generate using Julian date format" option.
Library	The name of the library to create the collection in. If the library does not exist, the GUI will ask if it should be created. **Note:** If the library contains Job Watcher collections files from a older release or record format then you will only be able to contain if all Job Watcher files are removed from the library. If you need to keep the data, then use another library.
Description	A description to give the collection.
Interval duration	The size of each sample of data in seconds. The interval duration from the definition will be preloaded as the default value for this field. **Note:** If the collect as fast as possible option is checked then this value is greyed out and is not applicable.
Collect as fast as possible	Check the collect as fast as possible button to collect the next snapshot immediately after the previous one finishes (no delay). Keep in mind this option can be very resource intensive so use caution!
Scheduled start time	This option allows you to schedule when the collection should start.

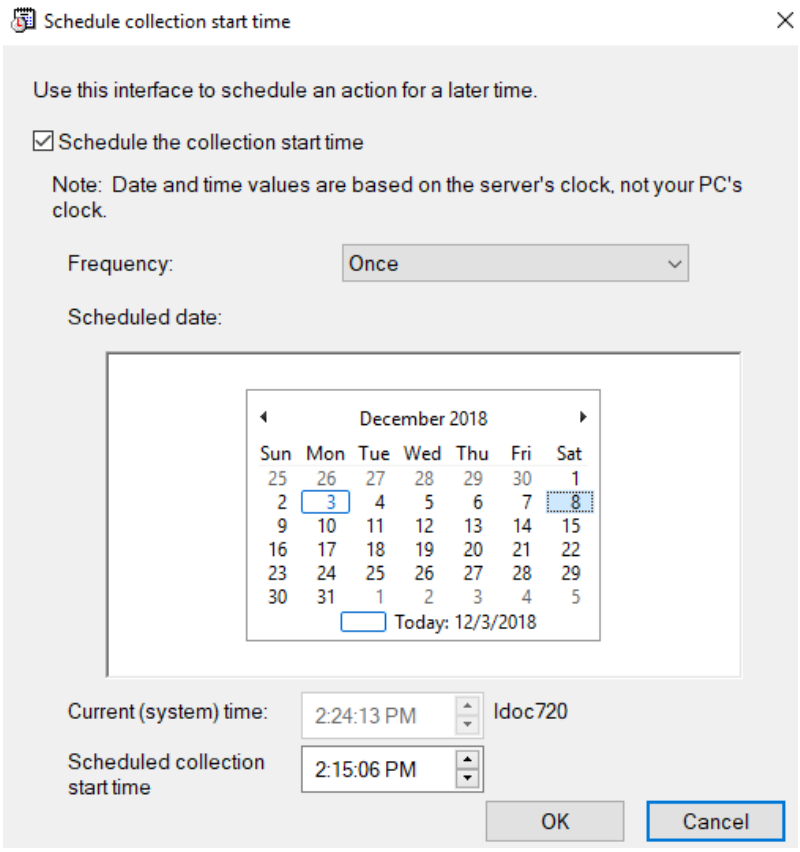
8.8.3 Scheduling Options

This page allows the user to determine a specific date and time for the collection to begin collecting data. By clicking the checkbox, the user can optionally include a date/time to schedule the collection. This option will create a scheduled job on the system.

Use the iDoctor Scheduled Jobs window to check the status of scheduled iDoctor jobs on the system. Access that window by right clicking the Job Watcher icon in the Job Watcher component view.

Tip: To configure the default scheduled time (number of days and hours in advance) preference, see the Preferences -> Scheduling interface.

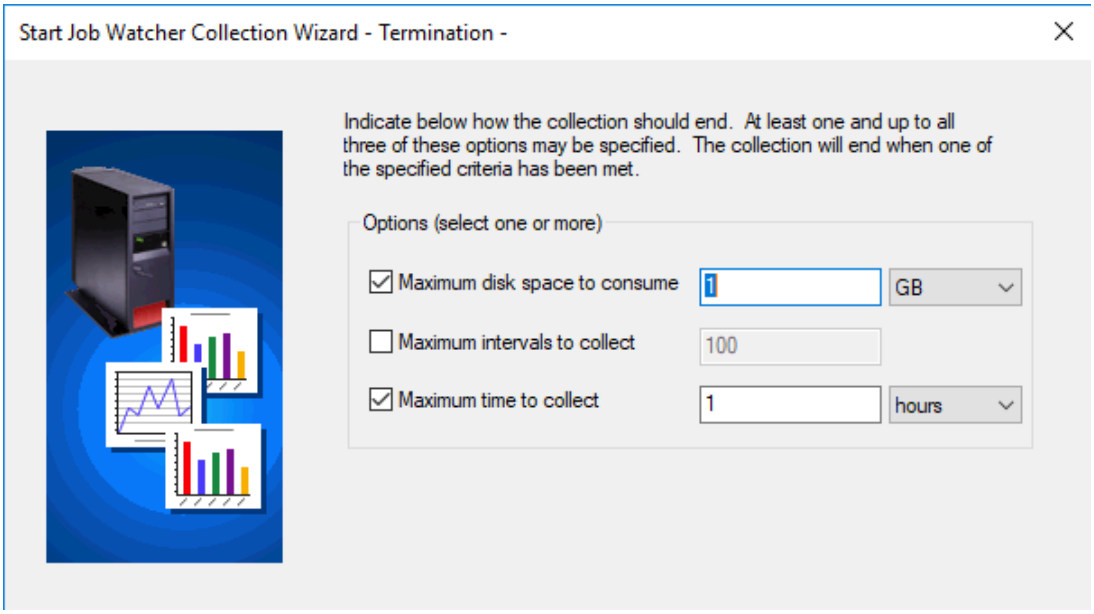
An example of this page of the Wizard is:



Start Job Watcher Collection Wizard – Schedule Collection Start Time

8.8.4 Termination

The Termination Page allows the user to specify what conditions should cause the collection to end. Whichever option is satisfied first, will cause the collection to end immediately.



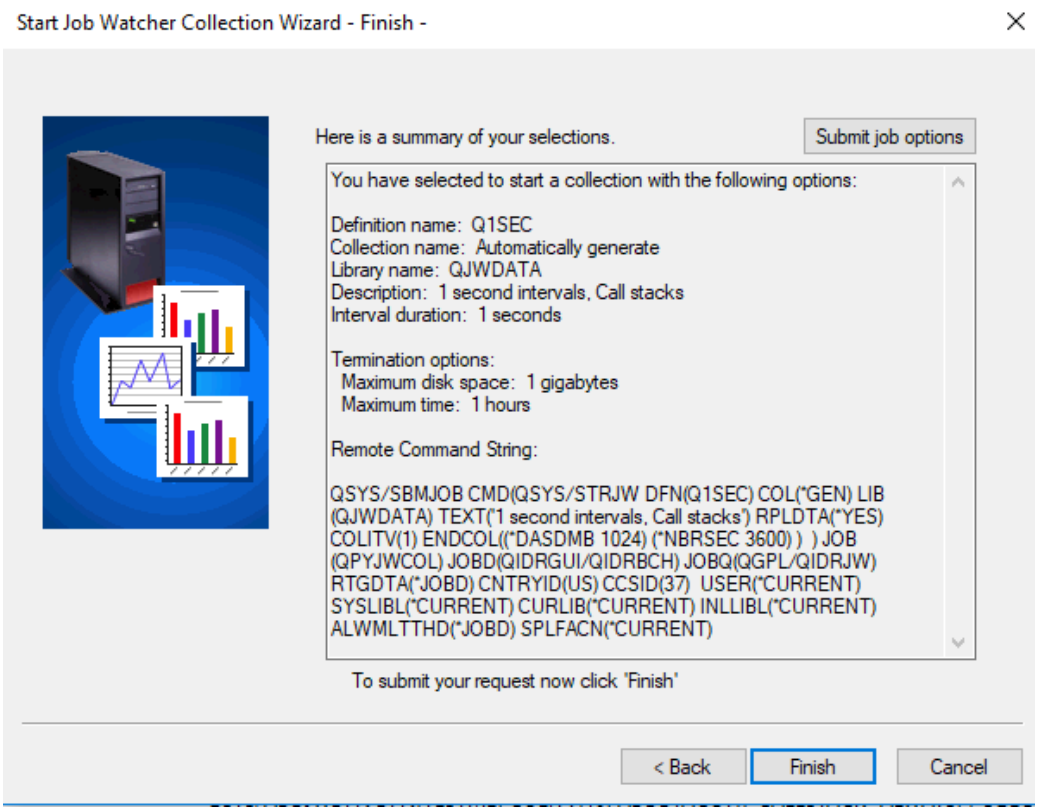
Start Job Watcher Collection Wizard – Termination

Field	Description
Maximum disk space to consume	If checked, the collection will end if the amount of disk space consumed by the collection reaches the amount specified.
Maximum intervals to collect	If checked, the collection will end once the specified number of intervals has been collected.
Maximum time to collect	If checked, the collection will end once the specified amount of time has elapsed. The time value may be entered in seconds, minutes or hours.

8.8.5 Finish

The Finish page provides complete details about all selections made in the wizard. If anything listed doesn't look right, use the Back button to go back and make any changes necessary. After clicking 'Finish' a STRJW command will be issued to start the collection. This command is listed at the bottom this page and can be copied to a green screen session and modified if necessary.

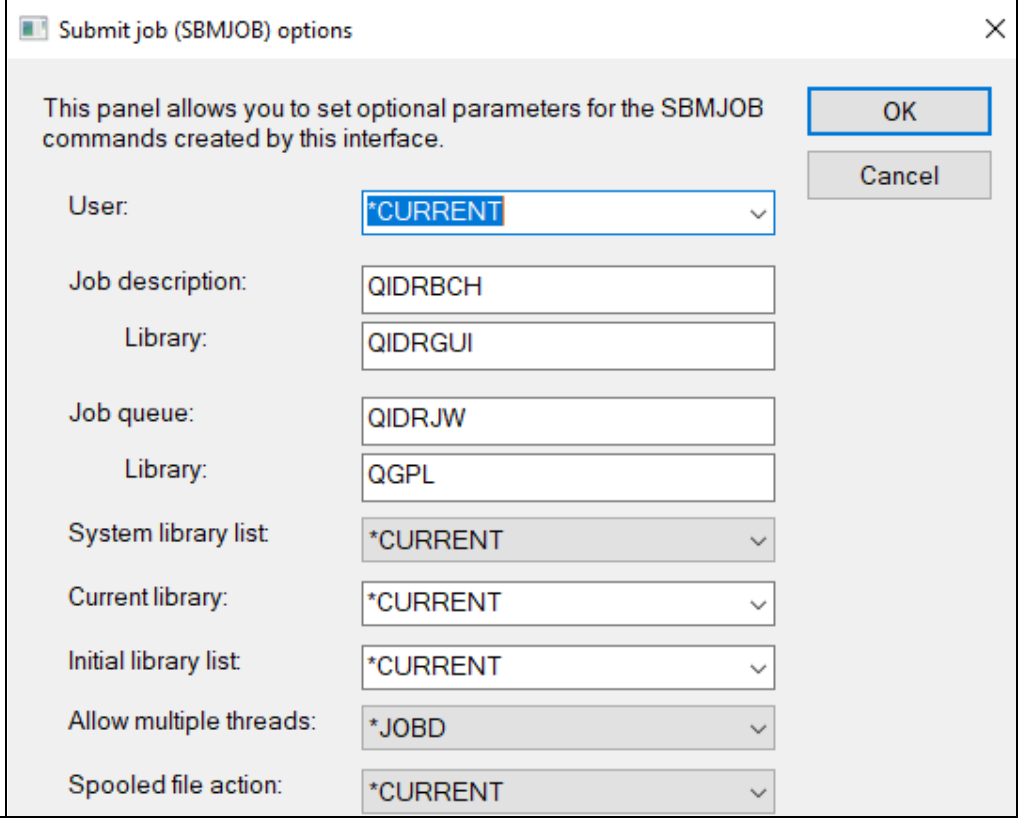
After the collection is started will take several seconds before anything appears in the GUI while the collection is initialized. Use F5 to refresh the list of collection in the collection library in order to work with the new collection.



Start Job Watcher Collection Wizard – Finish

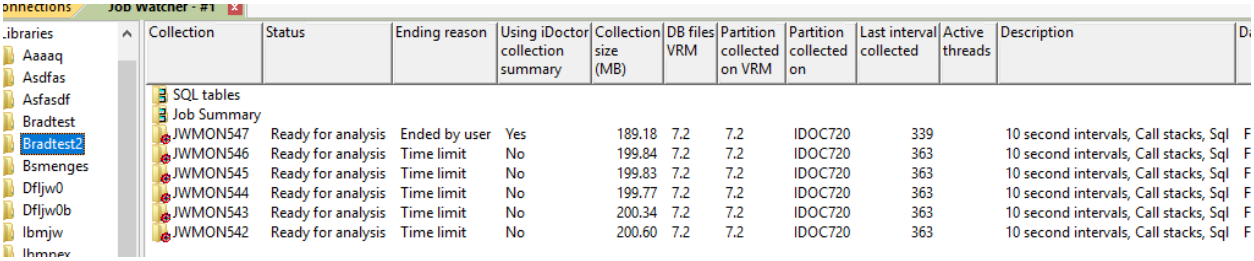
The following section lists the parameters available on this interface:

Option	Description
Submit job options	This button allows you to change parameters on the SMBJOB command. For more information see the help text for the SBMJOB command on the IBM i.



8.9 Collections

Moving down the tree within each Library folder are one or more collections that have been created (or are currently being created) within the current library. The green icons indicate active collections and red icons indicate collections that have completed. The status field is used to indicate if any errors occurred during collection or the current status of an active collection.



Job Watcher Collections in a Library

Each collection has a status field indicating whether it is currently running. You can also tell the status by the color of the icon: Green = active, Red = not active.

8.9.1 Collection Fields

The list of collections displays the collection name, description, status as well as several additional fields.

Each collection in the list has a set of fields available which can be optionally reordered and displayed. To change the current field selections for the collection list, use the Select fields... menu from the library folder. A listing of the available fields and a short description is provided in the table below:

Field	Description
Collection	Name of the collection. This name matches the member name used in the database files named QAPYJW* that exist in the current library.
Status	The status field indicates the status of the job on the system running the collection (if active) or if not active the status indicates whether the collection is ready for use.
Ending reason	This field indicates what caused the collection to end. There are several possible reasons a collection may end as described below: **Size limit** – The collection exceeded the maximum disk space allowed as described in the definition. **Interval limit** – The collection stopped when the maximum intervals to collect was met. **Time limit** – The collection stopped when the maximum time limit to collect was met. **ASP limit** – The system ASP limit as defined in SLIC service tools has been exceeded causing the collection to end. **Ended by user** – Job Watcher detected that the user ended the collection manually. **Rule time limit** – A conditional control file was defined but not met within the specified time limit. **Rule interval limit** – A conditional control file was defined that was not met within the specified interval limit. **Rule trigger program error** – A rule was defined where the conditions were met but when Job Watcher attempted to call the program associated with the rule, the program did not run correctly.
Using iDoctor collection summary	Indicates if the collection has been summarized or not. If this is No, then many of the graphs will take longer to run and fewer analysis options will be presented. **Tip:** Use the Analyses -> Run Collection Summary menu for a collection to create the summary tables.
Collection size (MB)	Displays the approximate size of the collection in megabytes. This size does not include the size of the summary files.
DB files VRM	This is the (IBM i) version of the Job Watcher database files found in the current library. **Tip:** Do NOT use CVTPFRCOL command to convert performance data for use by iDoctor, it can cause problems. Contact idoctor@us.ibm.com for advice if needed.
Partition collected on VRM	The version of IBM i that was used to create this collection.
Partition collected on	Indicates the name of the system the data was originally collected on.
Last interval collected	This value shows the last interval collected. If the collection is not running, this value indicates the total number of intervals that were collected.
Active threads/tasks	The total number of active jobs/threads (meaning used CPU in the last interval) detected in this collection. A value is only shown when the collection is actively running.
Description	A description for the collection specified at creation time.
Start time	The date/time the collection started.
End time	The date/time the collection ended (if not active)
Job creating collection	The fully qualified job that created (or is currently creating) the collection.

8.9.2 Menu Options

The table below outlines the different types of operations that may be performed by right clicking on a collection within the Job Watcher component view.

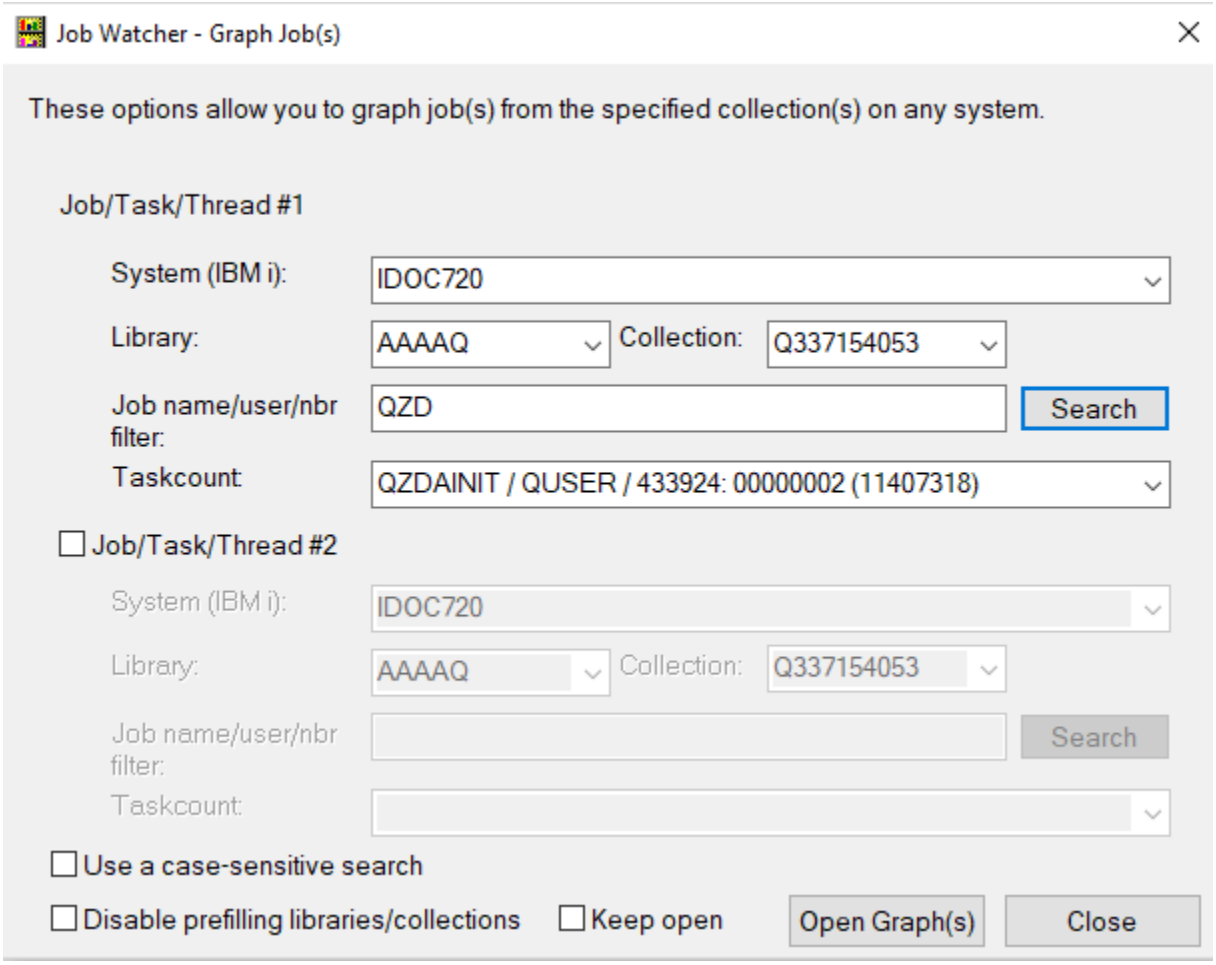
Menu Item	Description
Explore	Displays the contents of the collection folder in the right pane of the Job Watcher component view.
Record Quick View	Displays the fields for a collection in the list view vertically for easier viewing. Not available from the tree side.
Analyses -> Analyze Collection	Displays the Analyze Collection window showing the available analyses that can be ran against the desired collection(s). Data generated by these analyses are stored in SQL tables which are accessible under the SQL Tables folder.
Analyses -> Run ALL default analyses	If checked, the Run ALL default analyses option will be used. All default analyses will be executed for each collection after it completes. This is NOT all analyses but only a select few that are most commonly needed.
Analyses -> Run XYZ	This lists you run a specific analysis which varies by component and VRM of the collection. See the Analyses section for a list of those available.

Favorites	This list of graphs are the ones most commonly used and are great starting points.
Waits	Contains collection-wide wait summary graphs either by time interval or by thread. If unsure of where to investigate first, the Collection overview time signature under the by time interval subfolder is the best place to start.
CPU	Contains collection-wide CPU graphs showing CPU utilization along with Dispatched CPU and CPU queuing.
Job counts	These graphs help you see how many total jobs / tasks and threads existed on the system during the collection. Note that the counts given for each type are normally less than they truly were because many jobs/tasks/threads never use CPU during the collection and therefore the type of work (job, task or thread) running is unknown. These are added up together in an "Always idle jobs/threads/tasks" bucket. However, if the collection uses the "force 1st interval" option then this problem will go away. But use caution as this option can take a long time in some environments and may not be advised. Contact support first if you need advice on this! **Note:** This folder will NOT appear until the Collection Summary analysis has been ran on the collection.
Memory	These graphs in provide information about 3 job level metrics available in Job Watcher: pages marked easy to steal, page frames and memory page demand.
Temporary storage	These graphs provide metrics related to job temporary storage allocations or temporary pages allocated. **Note:** This folder only appears at 7.2+.
Physical disk I/Os	These graphs cover disk I/O metrics, page faults. synchronous response and more.
Logical DB	These graphs show logical I/O metrics of various types.
IFS	This folder provides collection-wide summary graphs showing IFS activity by time interval.
Top consumers	These graphs show the job, generic jobs or users that used the most CPU or spent the most time in any of the wait buckets.
Opens	These reports can help give an idea of which programs are causing opens. Because of the way Job Watcher works they are NOT always useful or accurate. These reports are based on the call stacks only captured at the end of each Job Watcher interval. The smaller your intervals and the more snapshots you have the more samples you get and more accurate these reports will be. **Note:** It is very common to have no data after running these reports and that is not a bug!
SQL	These graphs all show metrics related to SQL metrics.
J9 JVM graphs	These graphs summarize the JVM statistics for all J9 JVMs (IBM Technology for Java) found in the collection. These graphs only appear if the data was optionally collected.
Other graphs	This option provides collection-wide summary graphs showing other types of information such as state transitions and transactions by time interval.
System tasks explorer	These graphs are used to show wait bucket contributions for system tasks only. This can be used to compare with the Collection overview time signature.
Collection size	These reports are used to display information about the size of each collection.

Graph Job(s)	This option allows you to search for or specify a job and graph it over time using the wait bucket data. You can also specify a 2nd job in order to do a comparison of 2 jobs at once.
Search...	Performs a search over the entire collection looking for a specific piece of data specified by the user.
Generate Reports…	This option can be used to build a report of the desired set of Job Watcher tables and graphs. The report consists of a screenshot of each graph along with its title and collection information. The reports are built into a HTML page and displayed in the web browser when completed.
Copy URL	Creates a link to the component, library and collection that can be accessed later, or sent to another user.
Copy...	Copies one or more collections to another library. Selecting multiples is only available from the list side of the Job Watcher component view.
Delete...	Deletes a collection. Select multiple collections in order to delete more than one at a time. Selecting multiples is only available from the list side of the Job Watcher component view.
Rename	This option will rename the collection in the current library.
Save	Saves the selected collections to a save file on the system. The save file will be added to the list under the Saved collections folder under the Job Watcher icon.
Split	Divides a collection into multiple pieces based on an interval range or a time range. This can be used to focus on a particular set of data or to improve performance of the graphs if the collection is very large.
Transfer to...	FTP one or more collections to the PC, another IBM, another server or to IBM. Selecting multiples is only available from the list side of the Job Watcher component view.
Stop	Ends an active collection by issuing the ENDJW command. Once a collection is stopped it cannot be restarted again.
Properties	Use this menu to display the property pages for the collection. The property pages provide quick access to additional summary information about the collection.

8.9.3 Graph Jobs

This option allows you to search for or specify a job and graph it over time using the wait bucket data. You can also specify a 2nd job in order to do a comparison of 2 jobs at once. The data can come from two different systems if desired.

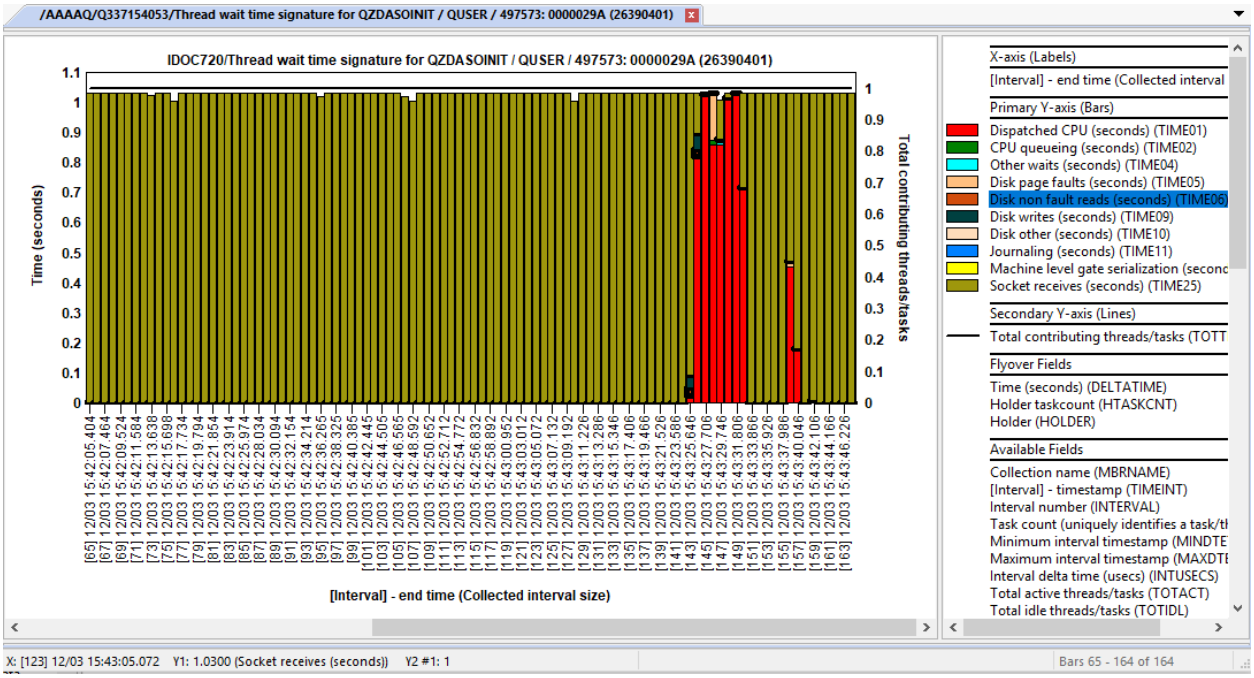


Job Watcher – Graph Job(s) window

Option	Description
Job #1 - System (IBM i)	The name of the IBM i to use for Job number 1.
Job #1 - Library	The Job Watcher collection library name. The drop-down list contains all libraries on the system that contain Job Watcher data (unless the disable prefilling libraries/collections is checked!)
Job #1 - Collection	The Job Watcher collection name to use for job #1. The drop-down list contains all collections found in the specified library.
Job #1 - Job name/user/nbr filter	This option allows you to enter in part of the job name, user, number or thread ID to search for. Pressing the Search button will build the list of taskcounts (jobs) that match the search criteria. Leave the value blank to return the 1st 1000 taskcounts in the collection.
Job #1 - Taskcount	The list of taskscounts matching the search criteria in the Job name/user/nbr filter field.

Note: The same information applies for Job #2 (see the previous table.) By selecting 2 jobs you will be automatically presented with a comparison graph of both jobs together over time.

Option	Description
Use a case-sensitive search	This option is used to perform a case-sensitive search. For some system task names this is required.
Disable prefilling libraries/collections	In some environments if you have many collections (hundreds) the drop down lists that prefilter the libraries and collections may be too slow to work effectively. Check this box and the drop-down lists will no longer be prefilled with data after making changes on this interface.
Keep open	This option if checked will cause the graph(s) to be created but this window will continue to remain open. This can be useful if you wish to kick off the graphs for multiple jobs at once before reviewing them.
Open graph(s)	This will open the specified graph's Thread wait time signature graph for the job(s) indicated.



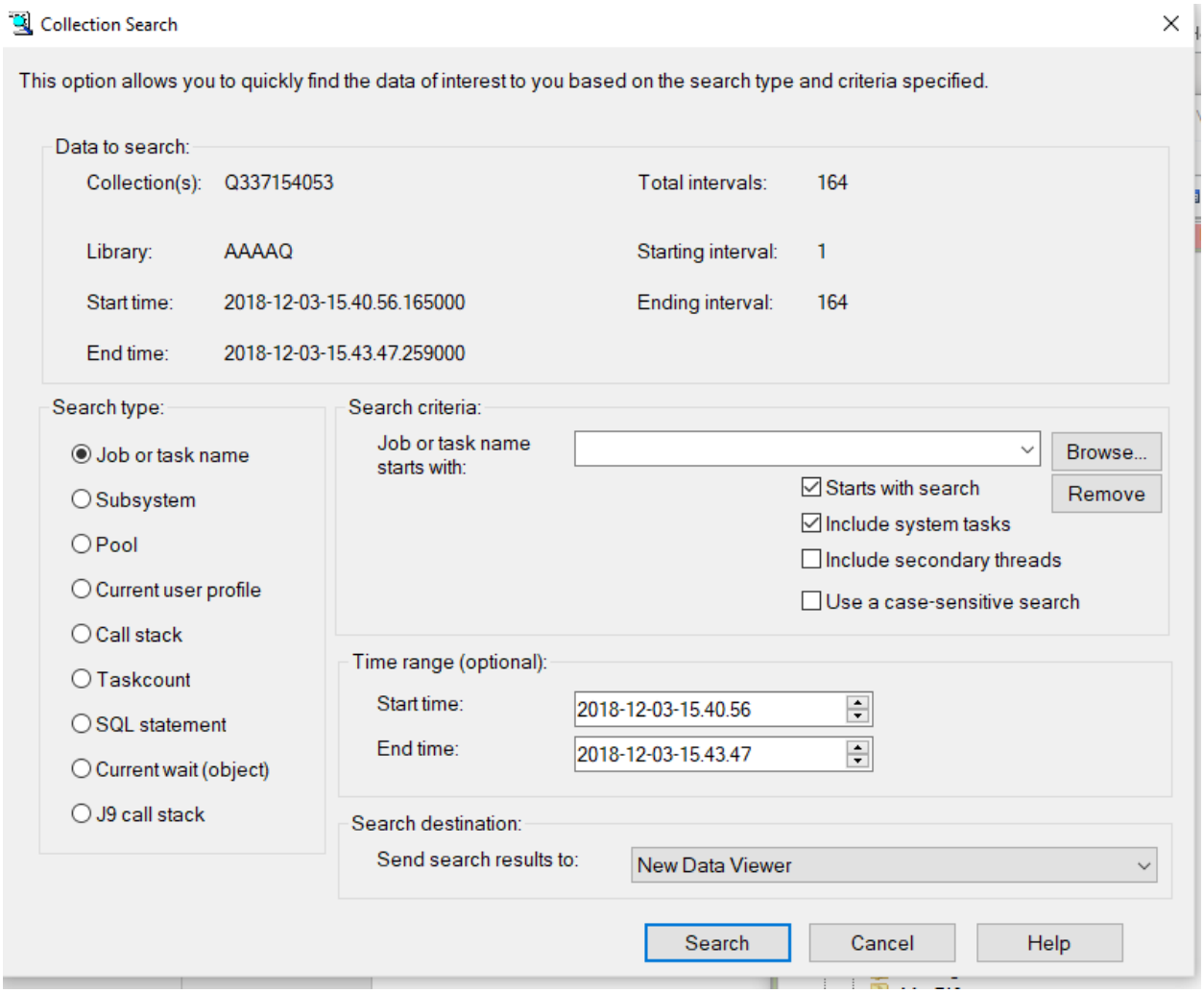
Thread wait time signature graph opened from the Graph Job(s) window

8.9.4 Search

The Search function in Job Watcher allows the user to look for a known job name, program name, subsystem, pool, user profile, or even part of an SQL statement in order to build a report for the detailed data found in the collection that matches the search criteria. The window offers a browse function, so the unique values found in the collection for each type can be selected from if desired.

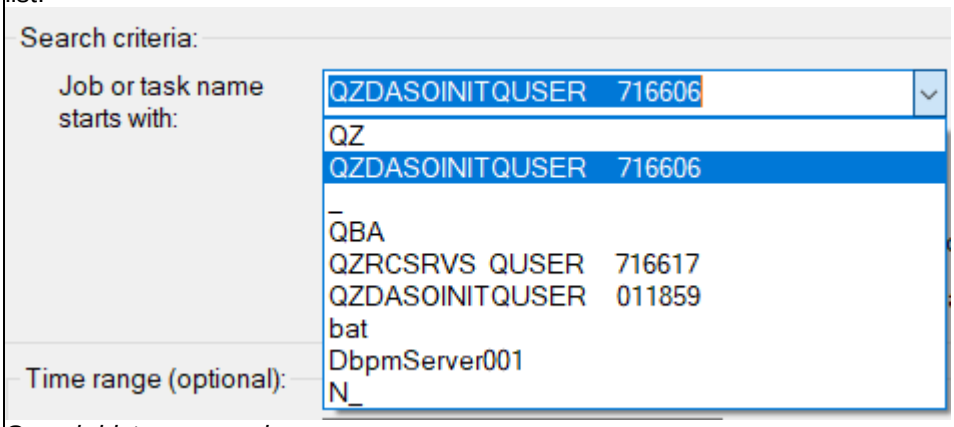
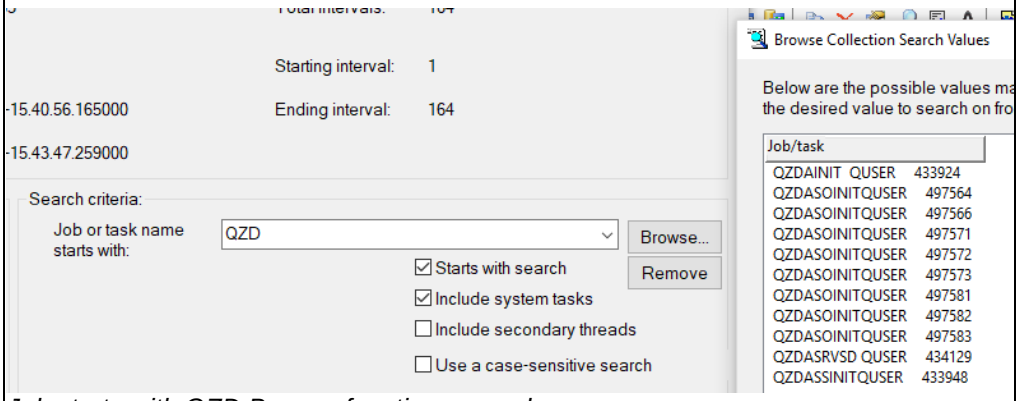
You can search over a single collection in the library or **multiple collections** in the same library if you select multiples before right-clicking them and then use the Search menu.

An example of this interface is:



Collection Search Window

The following table describes the fields on this window:

Options	Description
Collection(s)	Name of the collection(s) to search.
Library	Library name of the collection(s).
Start time	The date and time the earliest collection started.
End time	The date and time the oldest collection ended.
Total intervals	Total number of intervals found in the collection(s).
Starting interval	The smallest interval number found in the collection(s).
Ending interval	The highest interval found in the collection(s).
Search type	The search type allows you to pick which kind of data you want to search for. Changing the selection will change the fields shown in the Search criteria area of the window as appropriate.
Search criteria	The content of this section varies depending on the search type selected. You can use the Browse option to find the possible values in the collection(s) for the desired search type. The value entered acts as a filter for the Browse function as well. Clear the text box in order to show all available data in the Browse feature. The search drop-down lists contain search values used from previous searches. By selecting one and pressing the Remove button you can remove the entry from the list.  *Search history example*
Browse	The browse option is used to see the possible values matching the field you are searching on. After selecting a value from the window, the search text box value will be updated. **Note:** The data is prefiltered based on the value entered in the text box at the time the Browse button is pressed. An example of this interface is:  *Job starts with QZD Browse function example*
Starts with search	This checkbox is used to indicate if the search should be on the beginning of the data being searched or just contains anywhere in the field.

	For example, if you want to search for jobs starting with QZD **only** then check the starts with search box. But if you also want to include jobs named HYZQZD then uncheck the box.
Include system tasks	This option is used to include or exclude system tasks from the search results when searching by Job name, subsystem or pool. The default value is to include system tasks in the search results.
Include secondary threads	This option indicates if secondary threads should be included in the search results or not. This option applies only when searching by Job name, subsystem or pool.
Use a case-sensitive search	If you need to search on a mixed case system task name, then check this box.
Time range	The time range fields allow you to narrow you search based on the desired time range. The default time values shown match the start time and end times shown at the top of the window.
Search destination	This drop-down list allows you to pick which Data Viewer to send the search results to if multiple Data Viewers are active.

The Search criteria section allows you to enter the values appropriate for the search type selected. The fields available to search on change based on search type picked.

The Time range section allows you to narrow the search to only a specific time period. You may use this for example if a user reports a system slow down within a certain time for example.

For example, performing a search using search type “Job or task name starts with” with the value QZDA would give a report like this:

/AAAAQ/Q337154053/Job or task name search - #1

Library name (LIBNA	Collection name (MBRNAME)	Job name/user/number: thread ID (JTTHREAD)	Interval number (INTNU	Interval number (INTERVA	Time of day at ending snapshot start (STARTOD)	Reserved (TRESERV	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Elapsed interval time in microseco (TDEUSECS)
AAAAQ	Q337154053	QZDAINIT / QUSER / 433924: 00000002	1	1	2018-12-03-15.40.59.637000		11,407,318	4,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	1	1	2018-12-03-15.40.56.166000		26,390,389	2,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	2	2	2018-12-03-15.41.00.554000		26,390,389	4,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	3	3	2018-12-03-15.41.01.584000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	4	4	2018-12-03-15.41.02.614000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	5	5	2018-12-03-15.41.03.644000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	6	6	2018-12-03-15.41.04.674000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	7	7	2018-12-03-15.41.05.697000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	8	8	2018-12-03-15.41.06.727000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	9	9	2018-12-03-15.41.07.757000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	10	10	2018-12-03-15.41.08.787000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	11	11	2018-12-03-15.41.09.817000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	12	12	2018-12-03-15.41.10.847000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	13	13	2018-12-03-15.41.11.878000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	14	14	2018-12-03-15.41.12.908000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	15	15	2018-12-03-15.41.13.938000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	16	16	2018-12-03-15.41.14.968000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	17	17	2018-12-03-15.41.15.998000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	18	18	2018-12-03-15.41.17.028000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	19	19	2018-12-03-15.41.18.053000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	20	20	2018-12-03-15.41.19.083000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	21	21	2018-12-03-15.41.20.113000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	22	22	2018-12-03-15.41.21.143000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	23	23	2018-12-03-15.41.22.173000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	24	24	2018-12-03-15.41.23.203000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	25	25	2018-12-03-15.41.24.233000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	26	26	2018-12-03-15.41.25.263000		26,390,389	1,
AAAAQ	Q337154053	QZDASOINIT / QUSER / 497564: 00000BB3	27	27	2018-12-03-15.41.26.293000		26,390,389	1

iDocJW.mdb QAIDRSQL table SUM 300

Job or task name search results

From this report there are drill down options available to view graphs for any job and interval selected. For this example, right click and choose an option under the “Selected threads” menu.

The other search types such as call stack provide different outputs and drill down options as applicable.

8.9.5 Generate Reports

The Job Watcher Generate Reports function is used to create the desired series of reports and save each one as an image that can be reviewed later. The reports can be over one or more collections in the desired library. Selecting multiple collections will cause each graph produced to be over all collections selected (assuming the graph picked supports this feature.)

In addition to the reports generated this function will also build a summary of all collections.

Job Watcher
Idoc720, Library QJWDATA

Collection	Status	Ending reason	Using iDoctor collection summary	Collection size (MB)	DB files VRM	Partition collected on VRM	Partition collected on	Last interval collected	Active threads	Description	Day	Start time	End time	Job creating collection
RUN022	Ready for analysis	Ended by user	Yes	171.97	7.2	7.2	IDOC720	541		5 second intervals, Call stacks	Wednesday	2016-10-12-13.16.06.206000	2016-10-12-14.01.27.122000	Qpyjwcol / Mccargar / 037340
RUN021	Ready for analysis	Ended by user	Yes	228.98	7.2	7.2	IDOC720	722		5 second intervals, Call stacks	Wednesday	2016-10-12-12.15.44.981000	2016-10-12-13.16.16.346000	Qpyjwcol / Mccargar / 037337
RUN020	Ready for analysis	Time limit	Yes	228.94	7.2	7.2	IDOC720	723		5 second intervals, Call stacks	Wednesday	2016-10-12-11.15.23.438000	2016-10-12-12.15.59.468000	Qpyjwcol / Mccargar / 037334

Generate reports collection summary information for 3 collections

In addition, the creation settings and definition information for each collection is listed next in the report.

Collection	Creation settings	Definition
RUN022	**Creation settings:** Definition name: Q5SEC Collection name: RUN022 Library name: QJWDATA Description: *DFN Interval duration: *DFN seconds Termination options: Maximum disk space: 4096 megabytes Maximum time: 0000003630 seconds **STRJW Remote Command String:** QSYS/STRJW DFN(Q5SEC) COL(RUN022) LIB(QJWDATA) TEXT (*DFN) RPLDTA(*YES) COLITV(*DFN) ENDCOL((*NBRSEC 0000003630) (*DASDMB 4096))	**Defintion** Definition name: Q5SEC Description: 5 second intervals, Call stacks Interval duration: 5 seconds Job and task selection: All jobs and tasks Data collection options: Call stacks: Every interval Collect for jobs in conflict for at least 1 microseconds Collect for jobs in bad waits for at least 1 microseconds Advanced options: Collection file ASP threshold: 90% System ASP threshold: 90% No rule definition defined for this collection. **ADDJWDFN Remote Command String:** QSYS/ADDJWDFN DFN(Q5SEC) TEXT('5 second intervals, Call stacks') COLITV(5) ADDDTACGY((*CALLSTACK *ALWAYS)) WAITSTK((*CONFLICT 1) (*ABNWAIT 1)) JOB((*ALL)) TASKNAME(*ALL) FRCRCD(*ITVEND) TOASPTHLD(90) SYSASPTHLD(90)
		Defintion

Generate reports creation settings and definition information.

And then finally a section next will list information about the system for each collection selected.

System information:

Collection	System name	Operating system VRM	System serial number	System type	System model	Number of processors	Cycles per microsecond	File level
RUN022	IDOC720	V7R2M0	067B660	9179	MHD	1	512	8
RUN021	IDOC720	V7R2M0	067B660	9179	MHD	1	512	8
RUN020	IDOC720	V7R2M0	067B660	9179	MHD	1	512	8

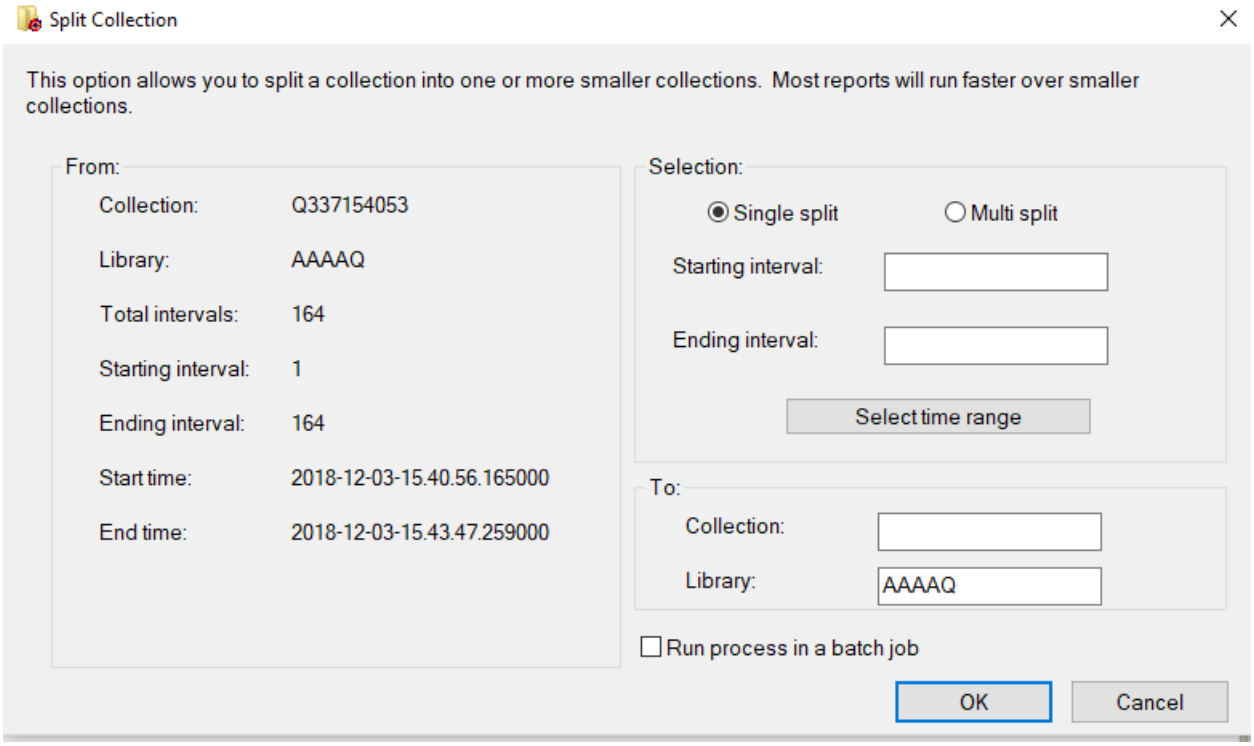
See the Generate Reports section in Chapter 4 for additional information on using this interface.

8.9.6 Split

Job Watcher provides a function that allows a user to split a large collection into one or more smaller collections. This is sometimes useful if the time range of interest within a collection is known and you

wish to isolate the data for only that time period.

An example of this interface is:



Split Collection Window – Single Split Mode

When performing only a single split, some of the options on the screen are different than when performing multiple splits. The following tables describe the elements shown on this interface:

From Frame Element	**Description**
Collection	Name of the collection to search.
Library	Library name of the collection.
Total intervals	Total number of intervals found in the collection.
Starting interval	The 1st interval found in the collection.
Ending interval	The last interval found in the collection.
Start time	The date and time the collection started.
End time	The date and time the collection ended.

Selection Frame Element	Description
Single split/Multi split	Toggles the split type between single and multiple splits
Starting interval	The 1st interval to include in the generated collection.
Ending interval	The last interval to include in the generated collection.
Starting time	The starting time to include in the generated collection.
Ending time	The ending time to include in the generated collection.
Select time range/ Select interval range	This is a toggle button that switches from selecting by interval numbers or by timestamp.
Intervals per collection	If multi split mode is enabled this field allows you to enter how many intervals each generated collection should contain.
Minutes per collection	If multi split mode is enabled this field allows you to enter how many minutes each generated collection should contain.

To Frame Element	Description
Collection	Name of the collection to generate. When generating multiple collections the name must be less than 8 characters.
Library	Library name to place the generated collections into

Note: The split function does NOT copy and split any data in the SQL tables folder. All analyses must be reran in the newly created "split" collection.

8.9.7 Stop

An active collection can be stopped by using the Stop menu found by right clicking on a collection within the Job Watcher component view.

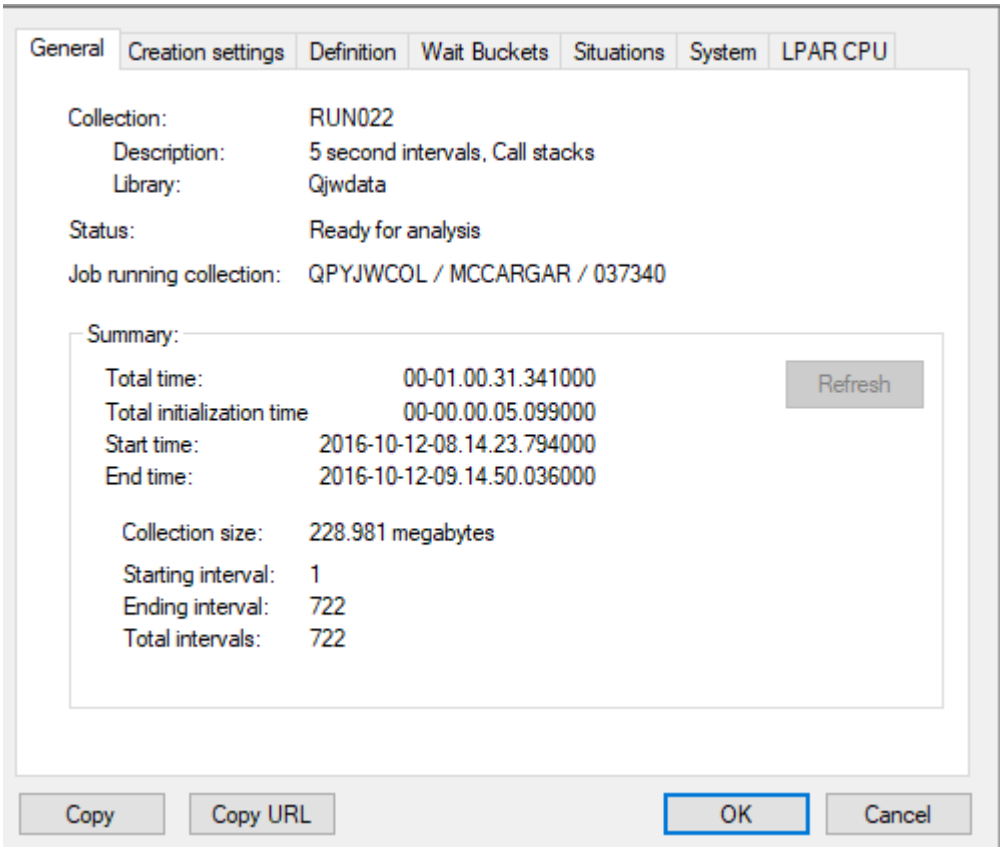
This option will issue an ENDJW command in order to end the collection.

8.9.8 Properties

This section covers the property pages for a collection. Access the property pages by right clicking on a collection and choosing the Properties menu.

8.9.8.1 General

The General property page provides basic information about the collection such as when it was created.



Collection Properties - General

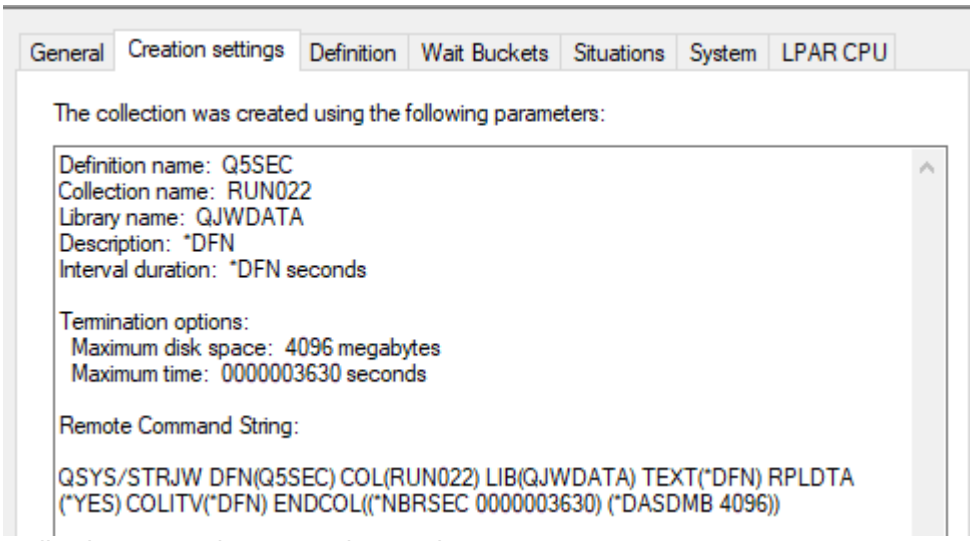
The following information is displayed on the General property page:

Option	Field Description
Collection	Name of the collection. This matches the member name used in the QAPYJW* files on the server in the library specified.
Description	Description of the collection.
Library	Library the collection resides in.
Status	The status of the collection. This could indicate if the job running the collection failed or that the collection is ready for analysis.
Job running collection	Displays the name of the job that created or is currently creating the collection. If the job log is available a button will be shown to display it.

Total time	Displays the total run time of the collection in timestamp format.
Total initialization time	Displays the estimated initialization time for the collection in timestamp format. This is an estimate of the amount of time it took between the collection being started and the 1st interval of data being collected
Start time	The time the collection was started.
End time	The time the collection ended (if it has ended).
Collection size	The total size of the collection. This number does NOT include any SQL tables generated.
Starting interval	The 1st interval number detected in the collection.
Ending interval	The last interval number detected in the collection.
Total interval	The total number of intervals found in the collection.

8.9.8.2 Creation Settings

The Creation settings property page provides details about the parameters that were used when creating the collection.

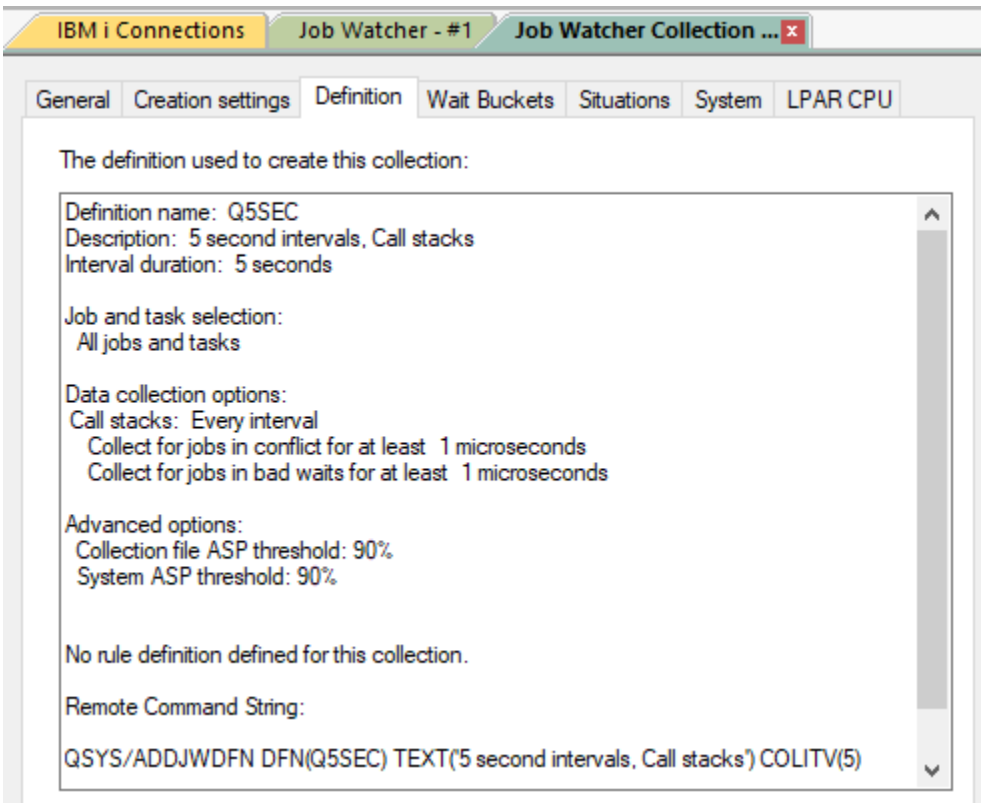


Collection Properties – Creation Settings

The information shown on this window matches the Finish page of the Start Job Watch Wizard when the collection was created.

8.9.8.3 Definition

The definition page displays the parameters that were defined in the definition used to create the collection.



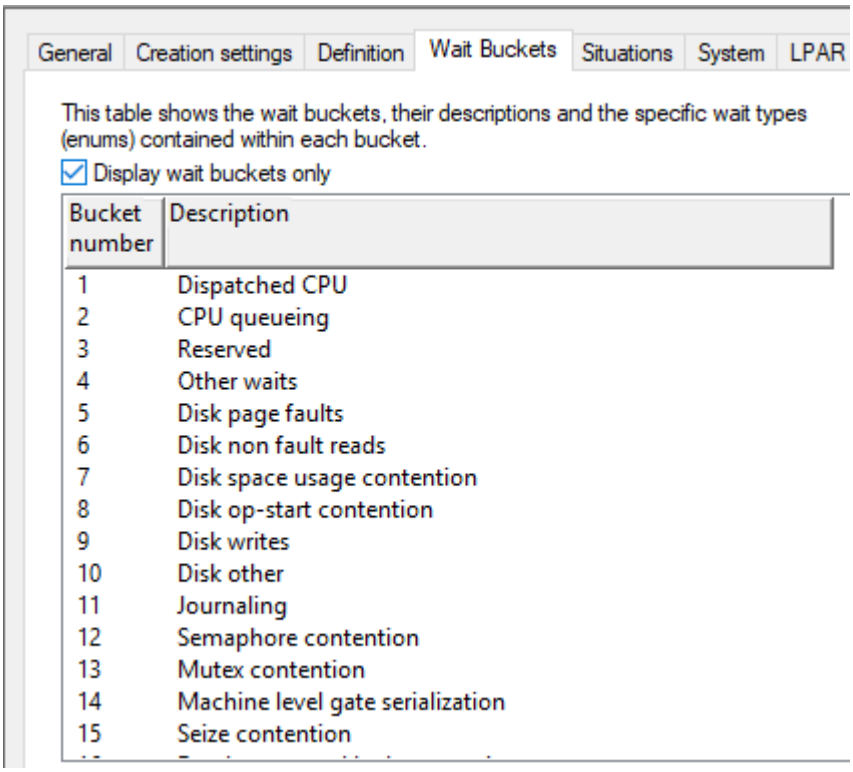
Collection Properties – Definition

8.9.8.4 Wait Buckets

In IBM i, Collection Services and Job Watcher utilize the same 32 wait buckets to identify the types of waits occurring on the system.

Each specific type of wait is identified by an enum (a wait point on the system) and each enum is given a wait bucket. In Job Watcher we can tell how much time was spent in each wait bucket for each thread during each interval. We can also tell what enum (wait) each thread was in at the end of interval and how long the thread was in that wait (the current wait).

The wait bucket page displays the wait bucket and enums within each bucket that were used during creation of the collection. Typically, the wait bucket mapping never changes except at release boundaries. These wait buckets are necessary for the wait graphs shown in Job Watcher and Collection Services Investigator.



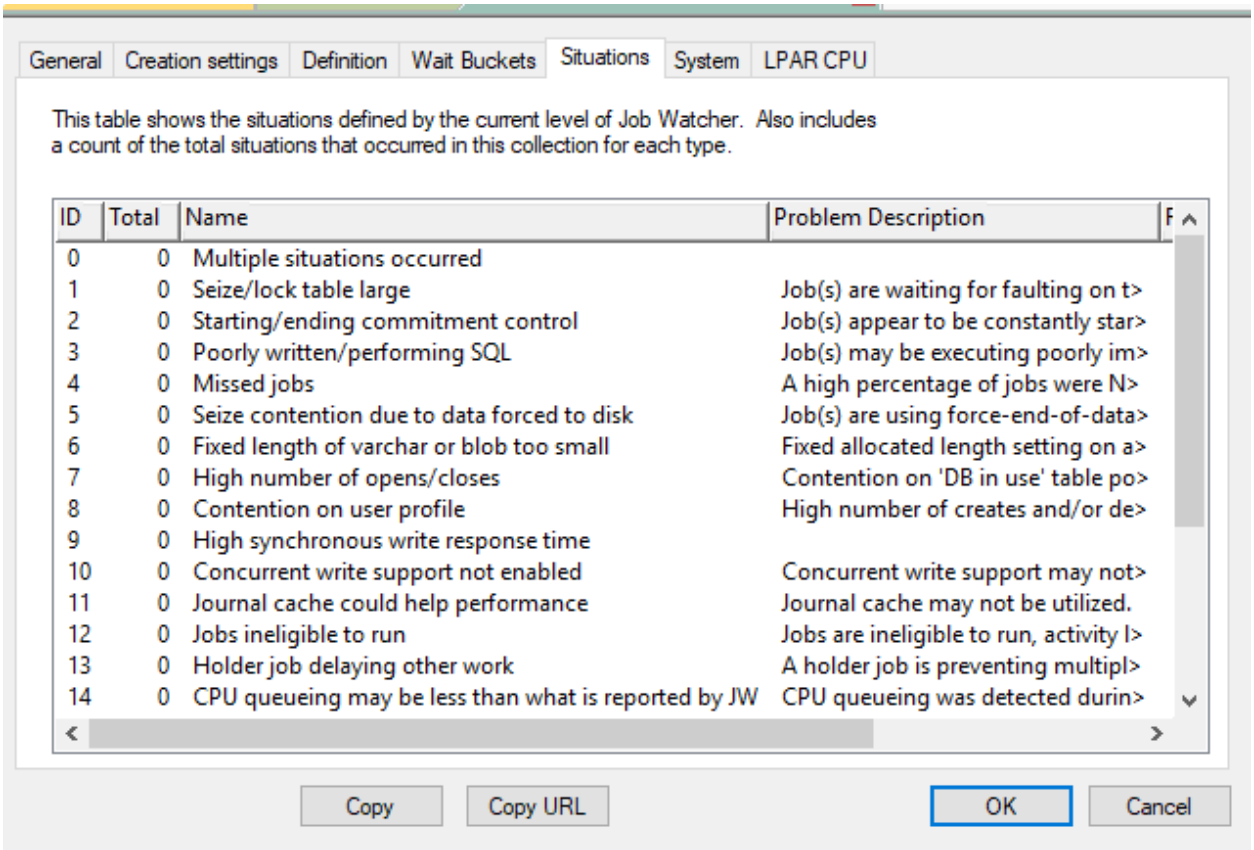
Collection Properties – Wait Buckets

Tip: Check the box “Display wait buckets only” if you just want to see a list of all the wait buckets without showing any specific enums within each bucket.

8.9.8.5 Situations

The Situations panel shows the Job Watcher Situational Analysis situations that have been defined by iDoctor. From here you can see the ID # of each situation, its name, as well as the problem and resolution descriptions.

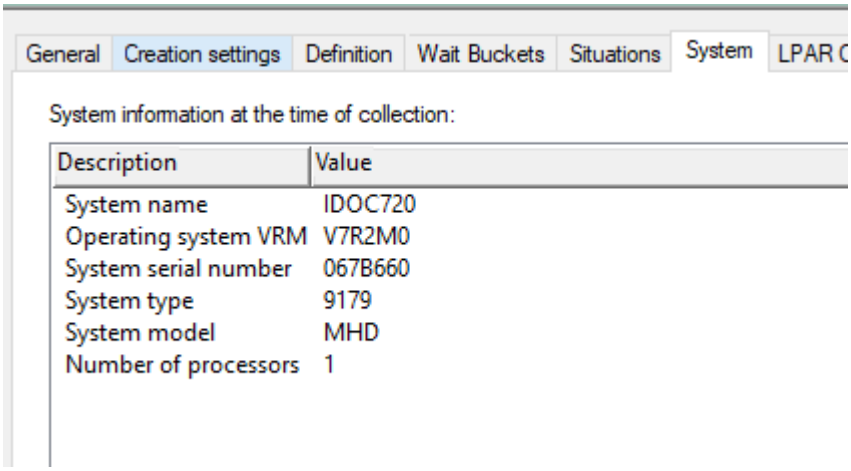
This page also shows the number of times each situation occurred during the collection via the Total column. Please note that if the collection has not been summarized and the Situational Analysis not yet ran then the 0 values may not be accurate.



Collection Properties – Situations

8.9.8.6 System

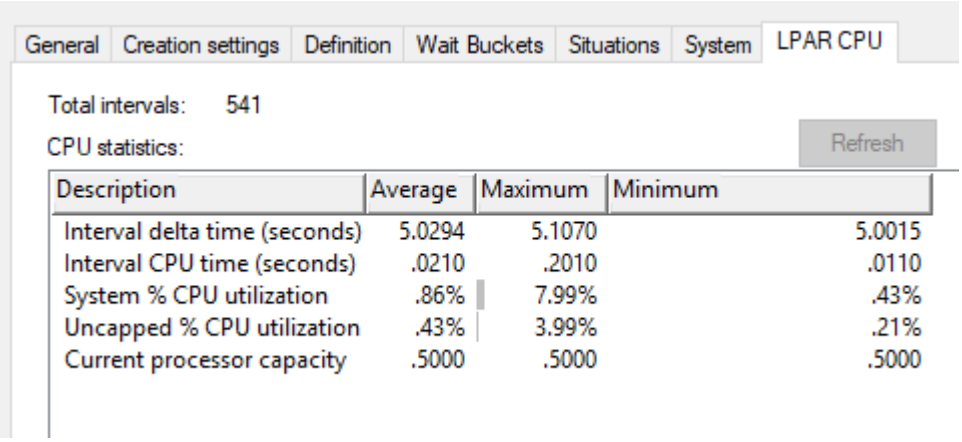
The system property page displays details about the system the collection was created on. This information includes the type, model, operating system VRM and the number of processors.



Collection Properties – System

8.9.8.7 LPAR CPU

The LPAR CPU property page provides details about the CPU utilization on the current partition during collection as well as the current processor capacity (CPC) value.



Collection Properties – LPAR CPU

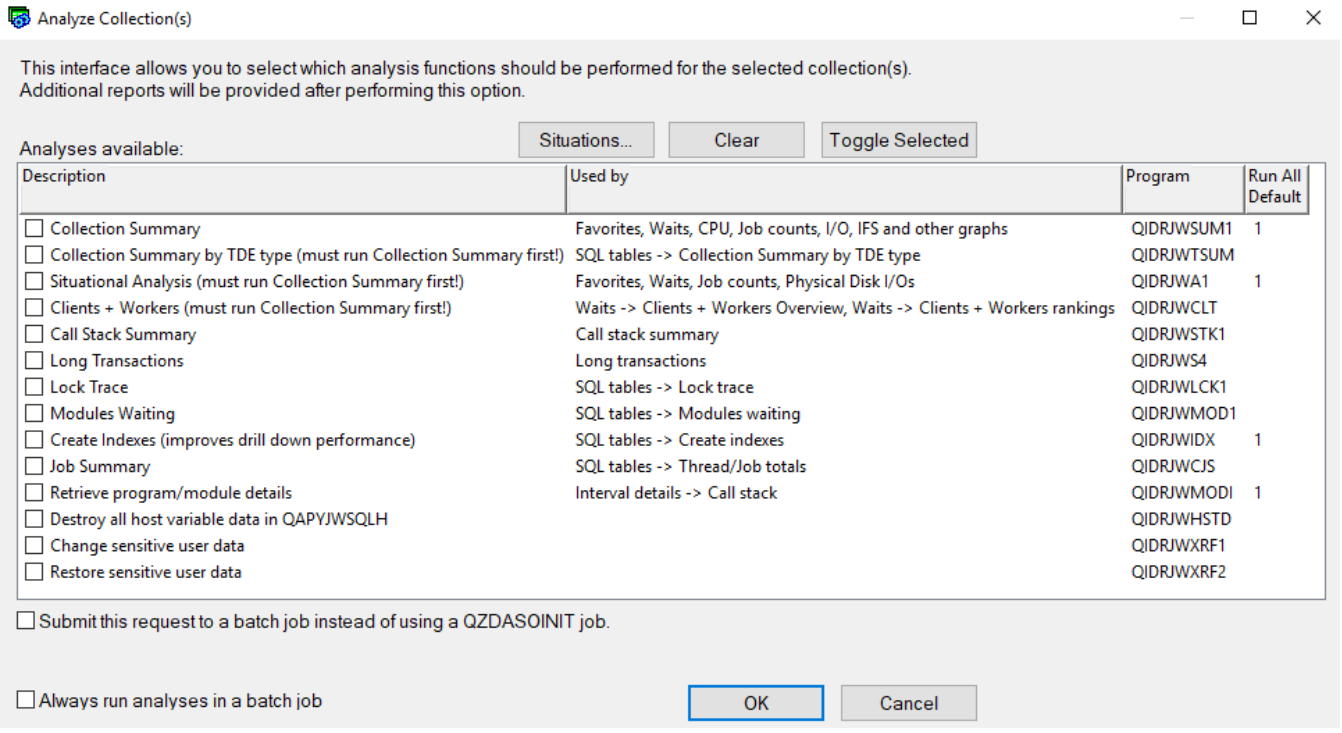
8.10 Analyses

The available Analyses in Job Watcher and what they provide is described in this section.

All analyses are written as SQL stored procedures and are initiated from the Analyses menu after selecting one or more collections and right-clicking. Each analysis has a 'fast path' option that allows it to be ran without visiting the Analyze Collection window.

8.10.1 Analyze Collection Window

The Analyze Collection window presents the user with a list of available analyses that can be ran over the currently selected collection(s). It is opened using the Analyses -> Analyze Collection... menu.



Job Watcher Analyze Collection(s) Window

Each available analysis is presented to the user on this screen. Special options for Situational Analysis such as creating your own situations or modifying the parameters used by the IBM defined situations are accessible by clicking the Situations button.

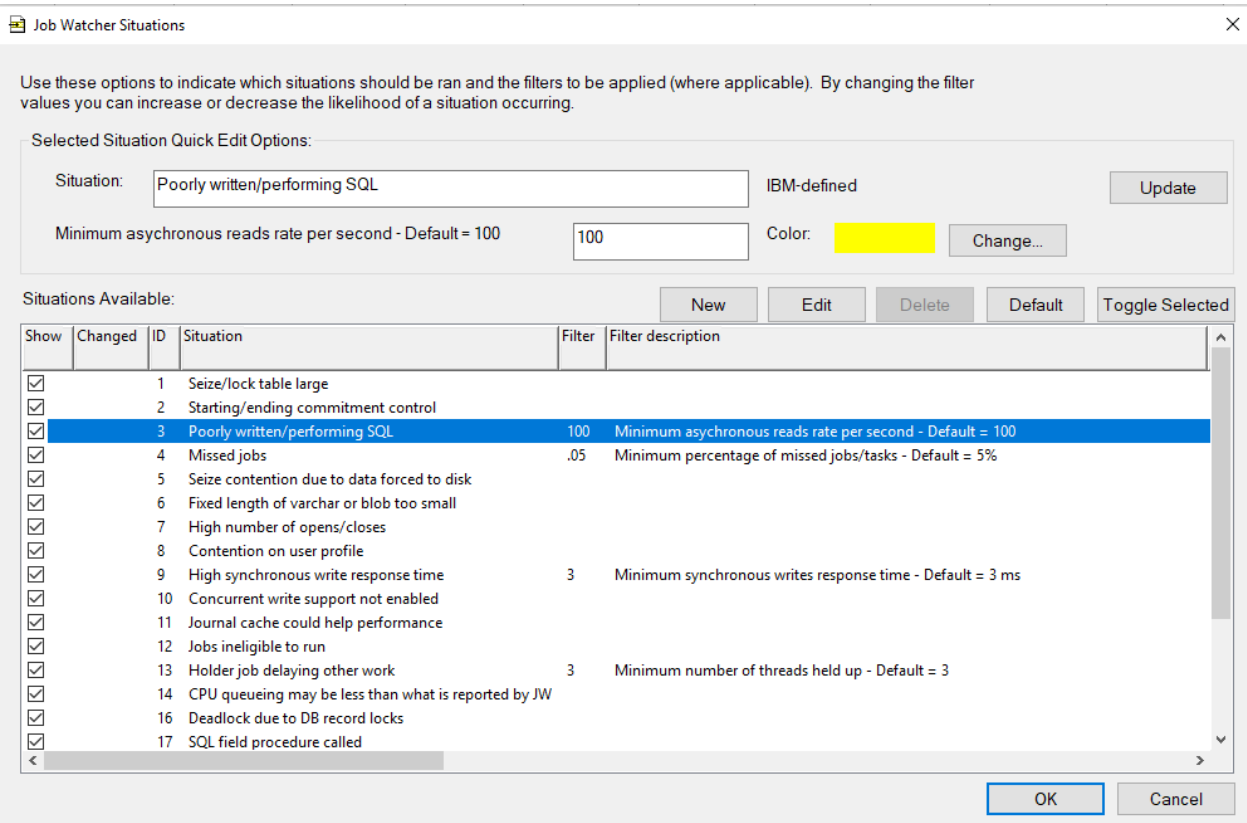
The controls on this interface and what they do is described in more detail in the following table:

Option	Description
Situations... button	Opens the Job Watcher Situations window which allows the user to modify the parameters used by the IBM-defined situations or create new ones.
Clear button	This button clears all selections.
Toggle selected button	This button changes the checked state of all selected analyses in the list.
Analyses available list	This is the list of the Job Watcher analyses currently available. The analyses available can vary from collection to collection depending on the OS VRM of the system the data was collected on and the data found in the collection. Checking an analysis name indicates that it will be ran when the OK button is pressed.
Submit this request to a batch job	If this option is used an SQL script will be created on the server and ran on the server in a new submitted job. The default behavior is to run the analyses from the Remote SQL Statement Status View which uses a QZDASOINIT job created by the GUI. Though this gives better status of the progress of the analysis processes, it is not ideal if the analysis is expected to take a very long time.
Always run analyses in a batch job	This option is a preference linked with Preferences -> Miscellaneous tab -> “Always run analyses in a batch job”. If checked the analysis will run in a batch job instead of a Remote SQL Statement Status View.

8.10.1.1 Job Watcher Situations Window

This window is accessed by clicking the Situations… button on the Analyze Collection(s) window in Job Watcher. It allows you to turn on or off any situations you want to disable or enable or change their settings.

Note: When changing are made to an iDoctor-defined situation those changes are saved in the windows registry.

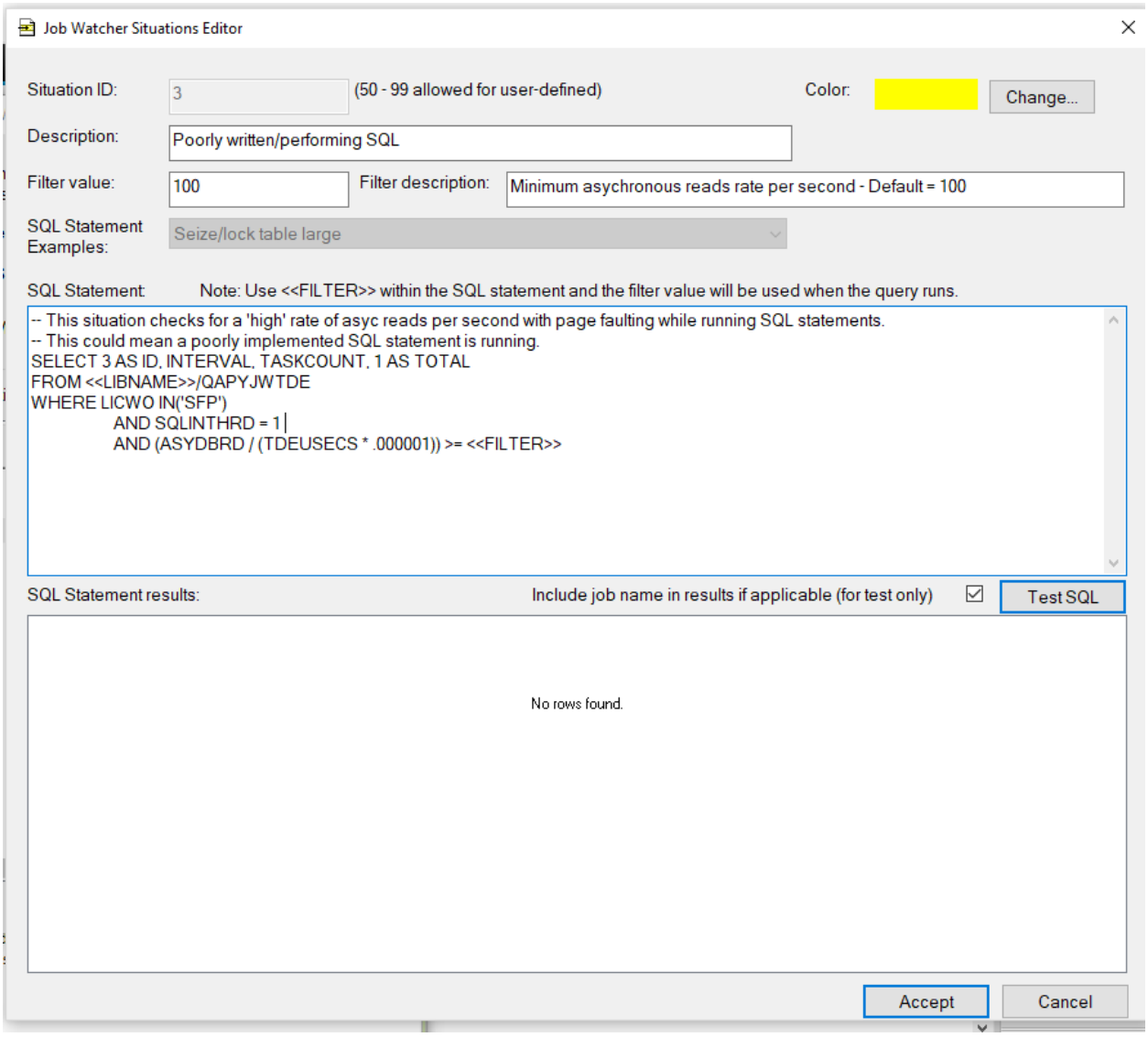


Job Watcher Situations

Option	**Description**
Situation (name)	This field displays or allows you to modify the current situation name.
Filter description	This displays the description of the filter available (if applicable, not all situations use a filter). The description cannot be changed here.
Filter value	If applicable this text field allows the user to modify the value for the filter. Within the SQL statement the <<FILTER>> parameter will be changed at run-time to use the value specified here.
Color	This option displays the (background) color to use for this situation. Press the Change.. button to modify it.
Update	Press this button to save any changes made to the situation name, filter value or color. To change other aspects of the situation you will need to select it from the list and then press the Edit button.
New	This button will allow you to create a new situation.
Edit	This option allows you to modify any of the details for the selected situation in the list.
Delete	The delete button will remove any user-defined situations. You cannot remove iDoctor-defined situations, but you can press the Show column's checkbox in order to stop using it.
Default	This option will discard all changes made to the iDoctor-situations and removes all user-defined situations from Job Watcher. Use with caution!
Toggle selected button	This button changes the Show state of all selected situations in the list.
Situations available	This is the list of iDoctor defined and user-defined situations. By selecting one its values will be shown at the top of the screen.

8.10.1.2 Job Watcher Situations Editor

This screen is shown when creating a new situation or editing an existing one.



8.10.2 Collection Summary

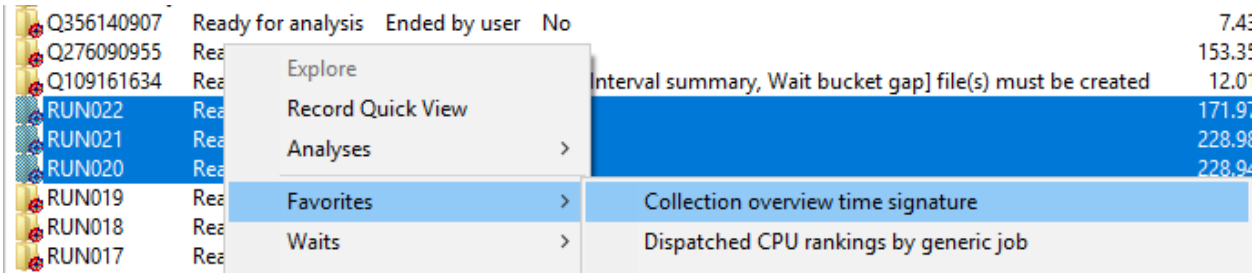
The Collection Summary analysis summarizes the job and wait bucket data in order to improve performance of graphs shown in Job Watcher and to offer more graphing options. The statistics are added up on a per interval basis and idle wait metrics from file QAPYJWSTS are expanded into an iDoctor SQL table (QAIDRJWGAP_<<COLNAME>>) for easier processing.

Note: When running this option, you may be prompted (depending on a preference) for any desired filtering you wish to perform against the data. Filtered data on this analysis can ONLY be analyzed under the SQL tables interface and no graphs are available over filtered Collection Summary analysis output. Do not filter the data if you want to analyze it with the graphs.

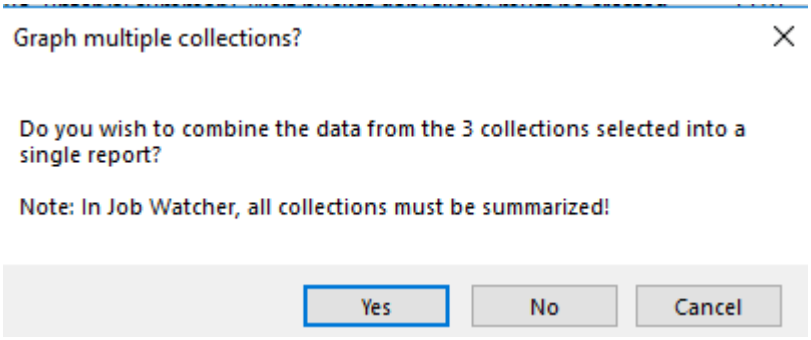
After running the analysis, new features become available in Job Watcher and are described in the next sections.

8.10.2.1 Graphing multiple collections

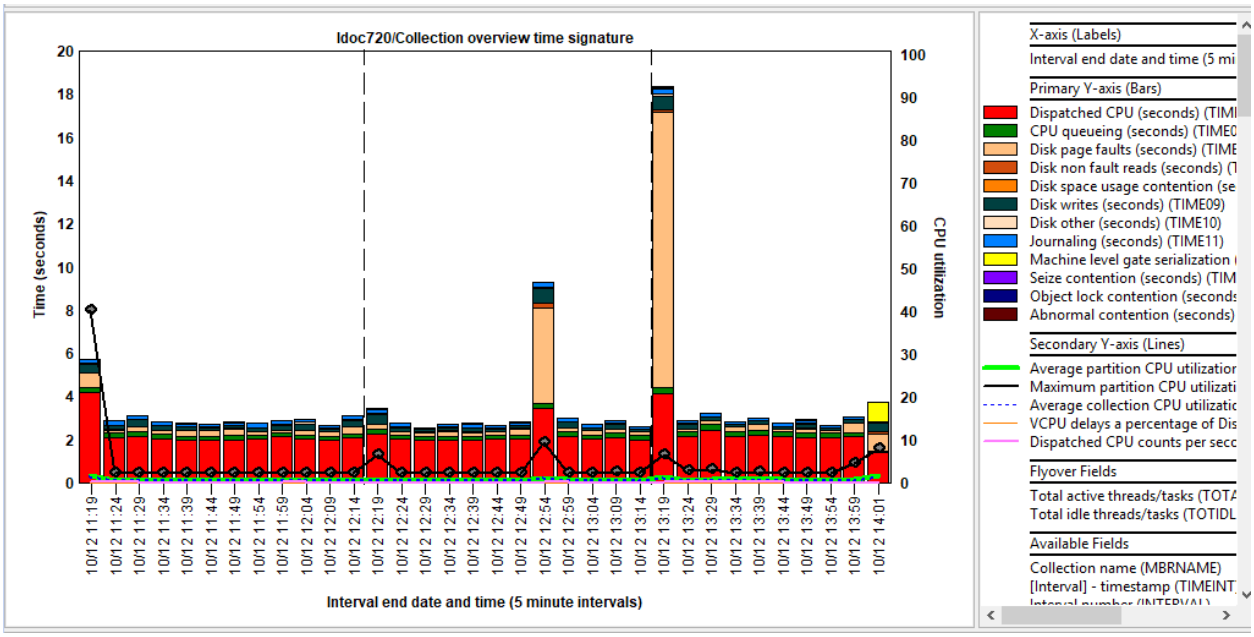
The ability to graph multiple collections within a single chart is available for many of the Job Watcher graphs. To initiate this process select them in the list of collections then right-click and pick the desired graph. **Tip:** Time breaks between collections will be indicated by vertical dashed lines.



Executing a graph against 3 collections



Graph multiple collections prompt



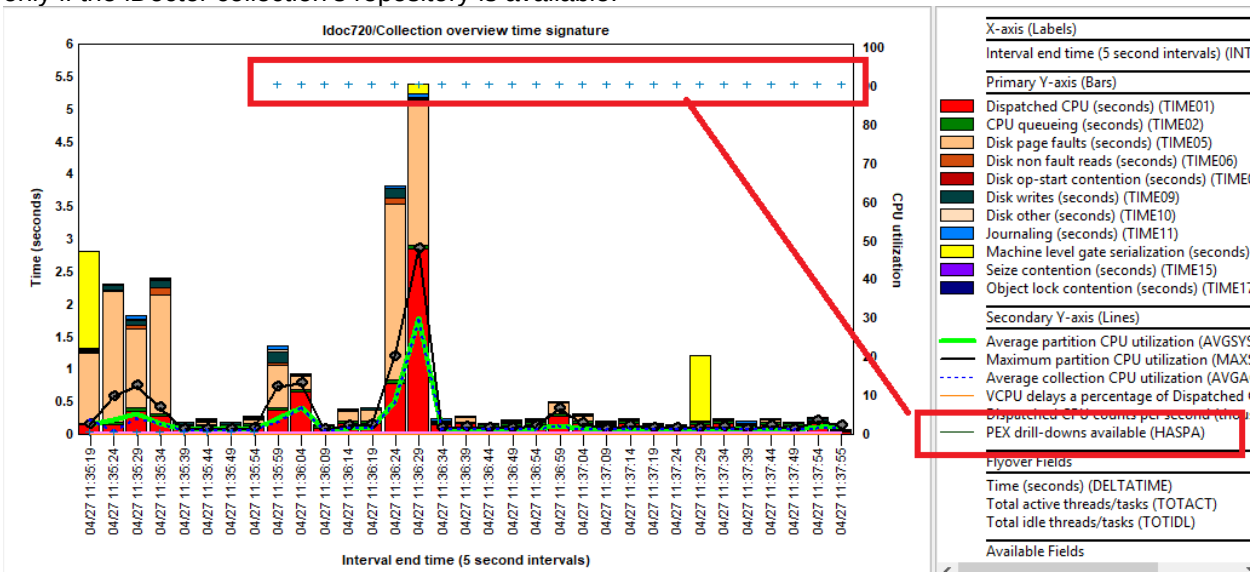
Collection overview time signature

8.10.2.2 Search over multiple collections

By selecting multiple collections and using the Search menu option you will be able to perform a search over multiple collections at once.

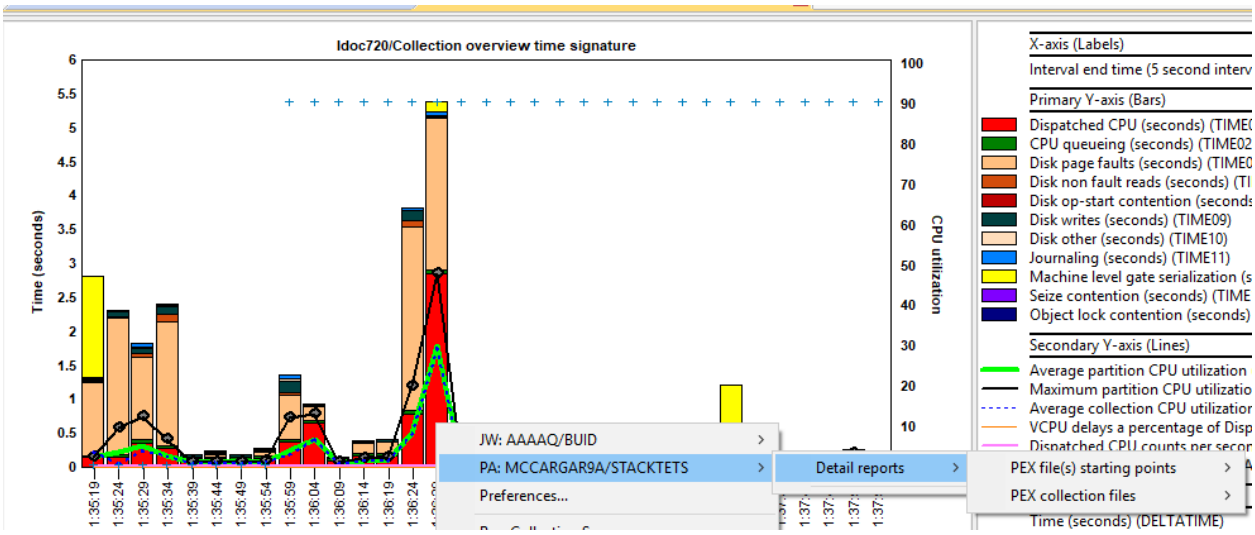
8.10.2.3 Intra-component drill downs

The Collection overview graph shows and provides options to drill down between components. Widgets are used to show that drill downs into CSI, Disk Watcher or PEX exist from Job Watcher. **Note:** This is only if the iDoctor collection's repository is available!



+ sign widget indicates that PEX drill downs are available in these intervals

Right-click on data in one of these intervals in order to have the option to view PEX Data for the selected period.

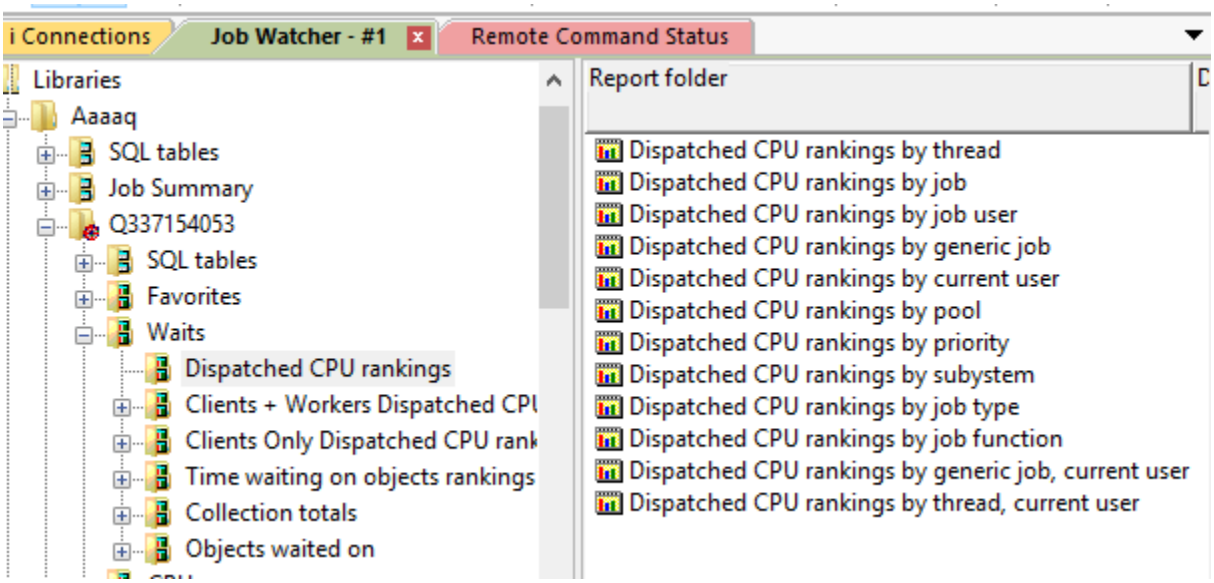


Drill-down into PEX example from Job Watcher

8.10.2.4 Additional ranking graphs

Many additional ranking graphs will become available after the collection summary analysis is ran.

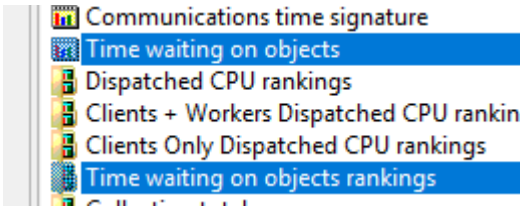
If the collection summary analysis is NOT ran, then only the by thread ranking graph option will be available!



Waits -> Dispatched CPU rankings folder

8.10.2.5 Waits -> Time waiting on objects

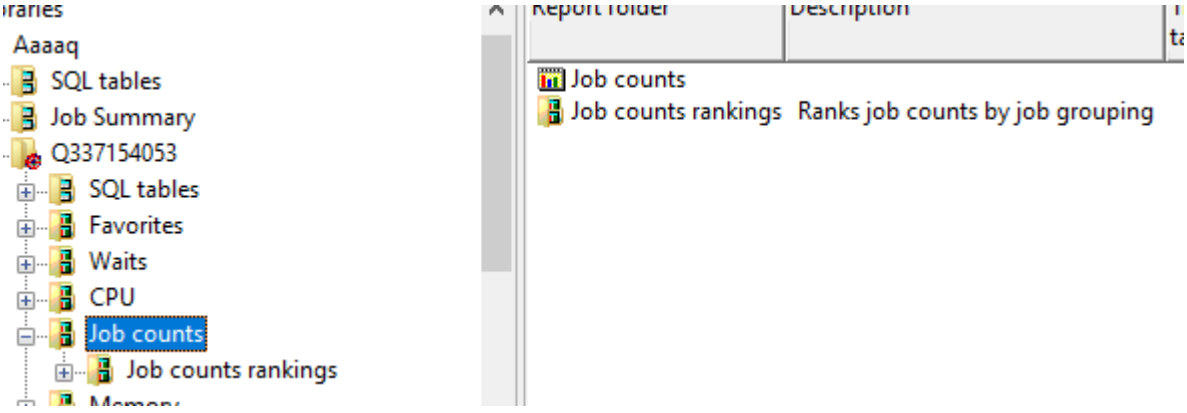
An additional set of graphs showing current wait duration times for wait objects found (at the end of each Job Watcher snapshot interval only) are added up. These show the type of (interesting) waits occurring and can be useful in some situations. Both an overview graph and ranking graphs are provided.



Waits -> Time waiting on objects graphs

8.10.2.6 Job counts graphs

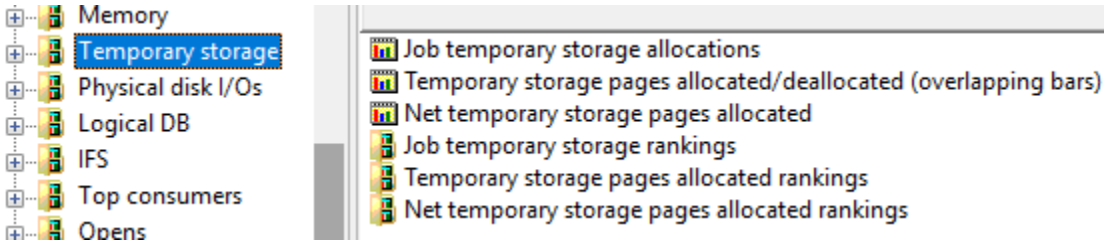
The Job counts folder will appear providing additional graphs.



Job counts folder

8.10.2.7 Temporary storage graphs

A set of temporary storage graphs will become available at 7.2+.



Temporary storage folder

8.10.2.8 Additional analyses can be ran

The following additional analyses can now be executed successfully: Situational analysis, Clients + Workers and Collection summary by TDE type.

8.10.2.9 Generate reports for multiple collections

The Generate reports feature can now be used against multiple collections.

8.10.2.10 Clients + Workers

The Clients + Workers analysis will also run and its data will become available.

8.10.2.11 SQL Tables Generated

The list of SQL tables generated by the analysis are shown below:

Table description	SQL table
Collection summary (Interval summary)	QAIDRJWSUM_<<MBRNAME>> -or- if any filtering/parameters are defined then the name becomes: QAIDRJWSUM_<<MBRNAME>>_<<JOBINFO>>_<<CURRENTUSER>>_<<SBS>>_<<STARTTIME>>_<<ENDTIME>>
Active and idle wait bucket times	QAIDRJWGAP_<<MBRNAME>>
List of identified taskcounts	QAIDRJWTL_<<MBRNAME>>
Additional tables for Clients + Workers also generated (see next section)	

8.10.3 Clients + Workers

This analysis is automatically kicked off when running the Collection Summary analysis. But this can also be executed separately from the Analyses menu if needed (i.e. the tables it creates get deleted).

The analysis looks specifically at the work done by *QSQSRVR* jobs (workers) and the jobs that kicked them off to perform work on their behalf (clients). It will also analyze any other jobs that may have a client/worker relationship defined on the system. If this exists it will be present in the Job Watcher data in file QAPYJWPRC in these columns:

- **CLIENTJOBN** CHAR 28
  Field text . . . . . : Client job name

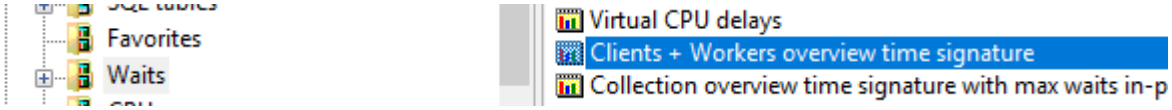
- **CLIENTTHD** BINARY 18
  Field text . . . . . : Client thread task count

But currently the only known jobs with this relationship are QSQSRVR jobs.

After running the analysis, new features become available in Job Watcher and are described in the next sections.

8.10.3.1 Waits -> Clients + Workers overview time signature graph

A new graph showing only wait bucket contributions from clients + worker jobs will appear.



Clients + Workers overview time signature graph

8.10.3.2 Waits -> Clients + Workers Dispatched CPU rankings

These graphs rank the client + worker jobs found in the data.

8.10.3.3 Waits -> Clients Only Dispatched CPU rankings

These graphs rank the client jobs found in the data.

8.10.3.4 Client + worker selected job over time

From the ranking graphs options, you can drill down on the selected client or worker using the Selected client or Selected worker menu options. These allow you to graph the client and/or worker over time. In some cases, a QSQSRVR jobs could be working for multiple clients (or vice versa), so these graphs will help show those relationships.

8.10.3.5 SQL Tables Generated

The list of SQL tables generated by the analysis are shown below:

Table description	SQL table
Client and worker interval summary file	Qaidrjwcltsum_<<MBRNAME>>
Client and worker taskcounts	Qaidrjwclt_<<MBRNAME>>

8.10.4 Collection Summary by TDE type

This option is used to produce an SQL table that summarizes the job statistics into 4 types of work per interval: Idle (I), Jobs (primary threads - P), Tasks (T) and Secondary threads (S).

There are currently no graphs over this output and the raw data can be viewed under the **SQL tables -> Collection Summary by TDE type** folder.

Interval number (INTERVA	Job (P) or task (T) or secondary thread (S) (TDETYPE)	Total contributing threads/tasks (TOTTDES)	Total CPU time (usecs) (TOTCPU)	Total IO (TOTIO)	Total synchronous IO requests (TOTSYN)	Total asynchronous IO requests (TOTASY)	Total read requests (TOTRD)	Total writes requests (TOTWRT)
1	I	1,203						
1	P	48	61,177	875	836	39	753	122
1	S	137	12,473	73	46	27	1	72
1	T	52	2,431	4	4	0	4	0
2	I	1,203						
2	P	45	20,273	799	727	72	538	261
2	S	129	4,925	19	15	4	0	19
2	T	62	214	0	0	0	0	0
3	I	1,203						
3	P	45	11,217	140	90	50	0	140
3	S	132	3,966	21	17	4	0	21
3	T	56	996	1	1	0	1	0
4	I	1,203						
4	P	44	7,072	85	58	27	0	85
4	S	131	4,221	20	16	4	1	19
4	T	57	669	0	0	0	0	0
5	I	1,203						

SQL tables -> Interval summary by TDE type

8.10.4.1 SQL Tables Generated

The list of SQL tables generated by the analysis are shown below:

Table description	SQL table
Interval summary by TDE type	QAIDRJWTSUM_<<MBRNAME>>

8.10.5 Situational Analysis

Situational Analysis is a function in iDoctor that looks for performance problems in a collection as an optional analysis. When ran and if situations (problems) are found in the data, they will be highlighted as background colors on the collection's high level (over time) graphs. Each situation identifies the job(s) associated with the situation and offers drill down options to view those jobs.

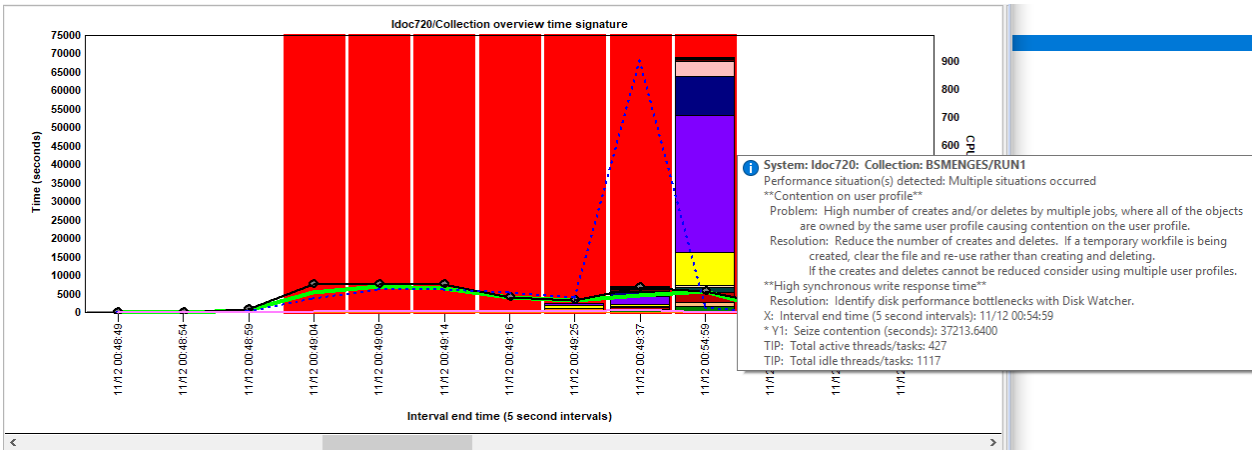
Note: Not all overview graphs will show the situations. Only some of the graphs in the Favorites, Waits and Job counts folders will show the situations currently. They are also shown in the Synchronous Response graph in the Physical Disk I/O folder.

Users can control the Situations executed when running this analysis by using the Situations... button on the Analyze Collection window.

After running the analysis, the following features become available in Job Watcher:

8.10.5.1 Situations displayed in the backgrounds of graphs

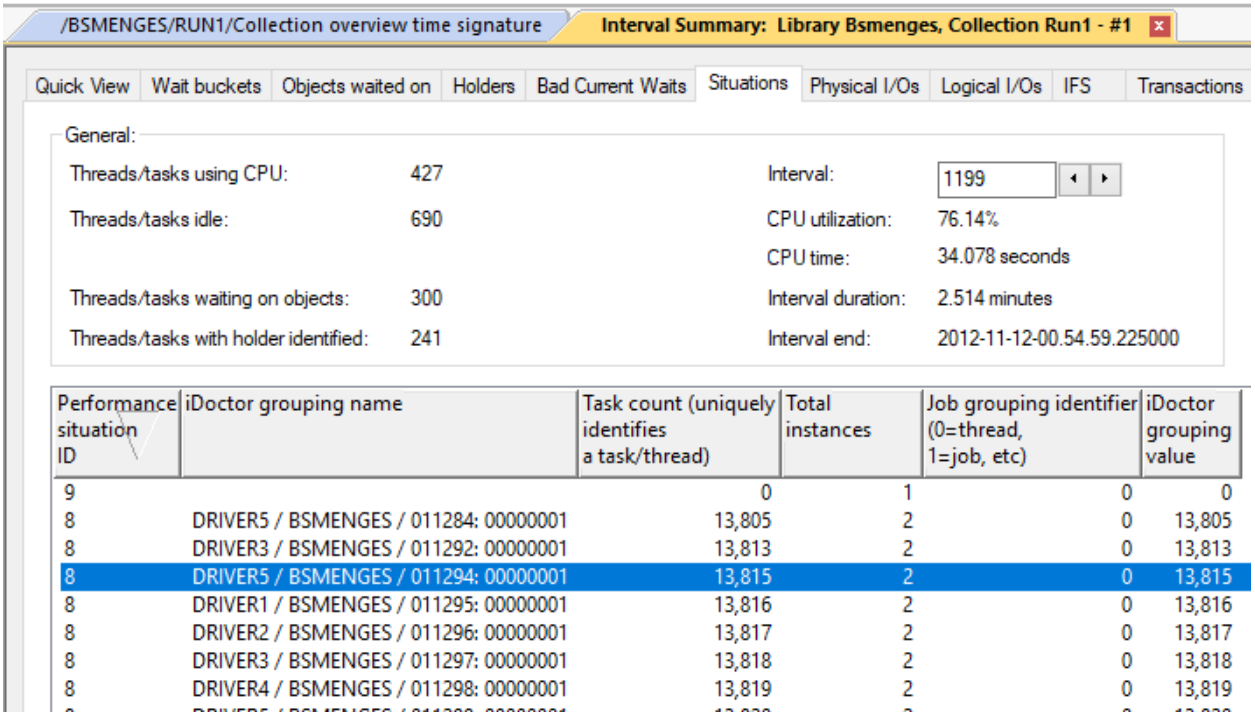
Many over time graphs in Job Watcher will display any situations that occurred in the background.



Job Watcher situations example

8.10.5.2 Interval summary -> Situations tab

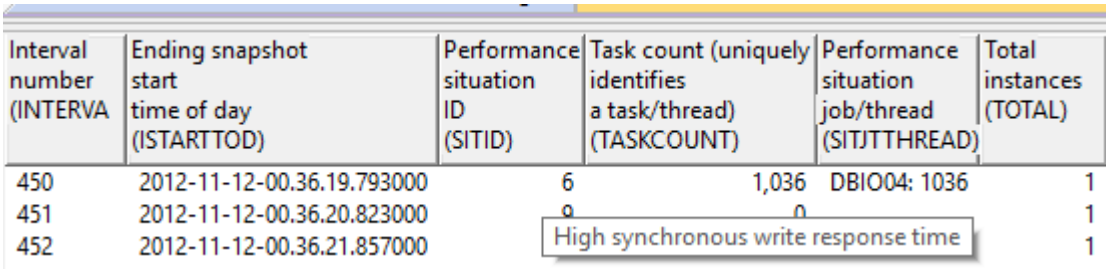
This interface is found by double-clicking on an interval in any time overview graph. It displays all the situations that were found in the interval.



Performance situation ID	iDoctor grouping name	Task count (uniquely identifies a task/thread)	Total instances	Job grouping identifier (0=thread, 1=job, etc)	iDoctor grouping value
9		0	1	0	0
8	DRIVER5 / BSMENGES / 011284: 00000001	13,805	2	0	13,805
8	DRIVER3 / BSMENGES / 011292: 00000001	13,813	2	0	13,813
8	DRIVER5 / BSMENGES / 011294: 00000001	13,815	2	0	13,815
8	DRIVER1 / BSMENGES / 011295: 00000001	13,816	2	0	13,816
8	DRIVER2 / BSMENGES / 011296: 00000001	13,817	2	0	13,817
8	DRIVER3 / BSMENGES / 011297: 00000001	13,818	2	0	13,818
8	DRIVER4 / BSMENGES / 011298: 00000001	13,819	2	0	13,819

8.10.5.3 Detail reports -> Situations

Additional drill down options will appear under the Detail reports -> Situations menu.



Interval number (INTERVA	Ending snapshot start time of day (ISTARTTOD)	Performance situation ID (SITID)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Performance situation job/thread (SITJTTHREAD)	Total instances (TOTAL)
450	2012-11-12-00.36.19.793000	6	1,036	DBIO04: 1036	1
451	2012-11-12-00.36.20.823000	9	0		1
452	2012-11-12-00.36.21.857000				1

Detail reports -> Situations -> Situation details report

8.10.5.4 SQL Tables Generated

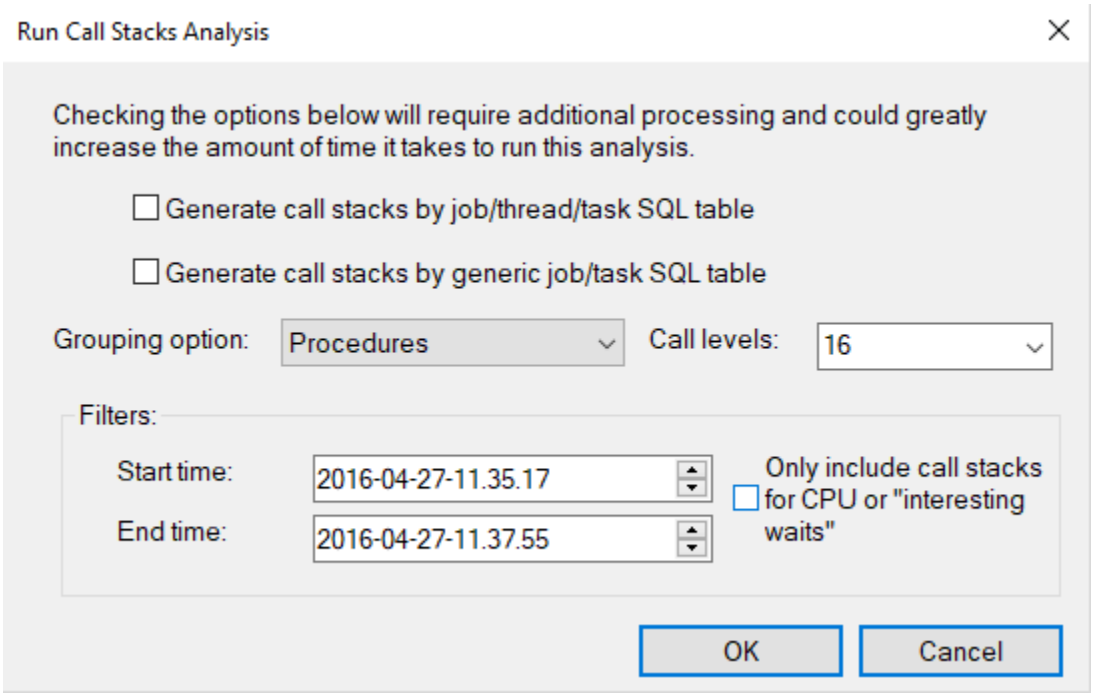
The list of SQL tables generated by the analysis is shown below:

Table description	SQL table
Situational Analysis Detail file	QAIDRJWANL_DTL_<<MBRNAME>>

8.10.6 Call Stack Summary

This analysis is used to analyze the call stack data captured in Job Watcher to look for stacks that are common or associated with certain performance characteristics.

When running the analysis, you may be prompted with the following interface:



Note: This interface will only appear if the **Preference -> Confirm -> Prompt for filtering options when running most PEX and JW analyses** is checked.

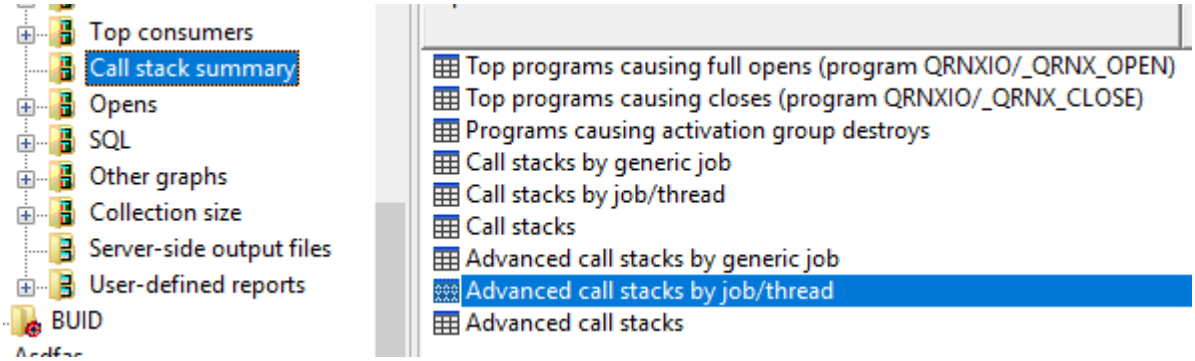
If the interface is not shown the default options are those shown above (16 call level groupings by procedure with no job grouping options checked.)

Option	Description
Generate call stacks by job/thread/task SQL table	This checkbox will cause an additional report to be created that groups the call stacks by job/thread/task.
Generate call stacks by generic job/task SQL table	This checkbox will create an additional report that groups the call stacks by generic job or task name.
Grouping option	This option controls whether the grouping of the call stacks is based on the **procedure** or instruction address.
Call levels	The maximum number of call levels to group the call stacks by in the report. Call stacks that have differences beyond this many call levels will all be grouped together. Any value can be entered from 16 to 50.
Start time	The start time to filter the data by.
End time	The end time to filter the data with.
Only include call stacks for CPU or “interesting waits”	This will cause the call stacks returned to be reduced and only return call stacks that occurred while interesting waits or CPU occurred. This will avoid returning call stacks in the reports that are often for idle jobs and not of interest. **Tip:** The “interesting waits” are those wait buckets that are shown on the Collection overview time signature graph.

After running the analysis, the following features become available in Job Watcher:

8.10.6.1 Call stack summary folder

The Call stack summary folder appears under the collection containing several reports depending on the options selected on the prompt described earlier.



Call stack summary folder

8.10.6.2 SQL Tables Generated

Table description	SQL table
Call stack summary file	QAIDRJWSTKSUM_<<MBRNAME>>
Call stack keys	QAIDRJWSTKSUMKEYS_<<MBRNAME>>
Call stacks stats	QAIDRJWSTKSUMSTATS_<<MBRNAME>>
Call stack build helper	QAIDRJWSTKSUMTMP_<<MBRNAME>>
J9 call stack keys	QAIDRJWSTKJVASTACK_<<MBRNAME>>
J9 call stack PROCI	QAIDRJWSTKJVAPROCI_<<MBRNAME>>
Generic job call stack summary file	QAIDRJWSTKGENJOB_<<MBRNAME>>
Generic job call stack keys	QAIDRJWSTKGENJOBKEYS_<<MBRNAME>>
Generic job call stack stats	QAIDRJWSTKGENJOBSTATS_<<MBRNAME>>
Job call stack summary file	QAIDRJWSTKJOB_<<MBRNAME>>
Job call stack keys	QAIDRJWSTKJOBKEYS_<<MBRNAME>>
Job call stack stats	QAIDRJWSTKJOBSTATS_<<MBRNAME>>

8.10.7 Long Transactions

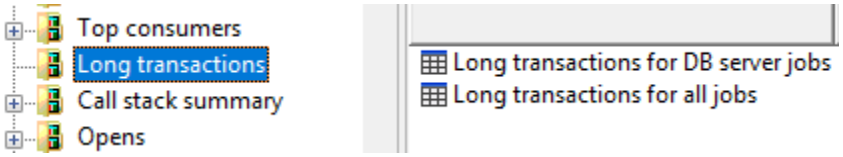
The long transactions analysis is perhaps poorly named. This analysis does not look for long running 5250 transactions but instead **looks for time periods in the job data where no normally 'idle' waits occurred**. It identifies time where jobs spent exclusively doing real work without pause. It also identifies long running SQL statements (assuming those SQL statements also did not experience any 'idle' waits during their execution).

From the SQL table generated by this analysis a user to view the longest periods of activity in the collection and the jobs that caused them.

After running the analysis, the following features become available in Job Watcher:

8.10.7.1 Long transaction folder

This folder will appear and contains the reports for this analysis.



Long transactions folder

/BSMENGES/RUN1/Long transactions for DB server ...

Total intervals (INTERVAL	Job/task name (TDEJOBNAME)		Thread ID (THREADID)	Job user profile (if constant) (JOBCURRUP)	Interval number (INTERVA	Ending interval (MAXINT)	Task count (uniquely identifies a task/thread) (TASKCOUNT)
27	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	2,812	2,839	13,711
26	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	2,840	2,866	13,711
22	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,596	1,618	13,711
17	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,414	1,431	13,711
17	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,621	1,638	13,711
13	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	2,710	2,723	13,711
13	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	2,726	2,739	13,711
9	QZDASOINITQUSER	011866	000000000000002D	-1	2,102	2,111	14,440
8	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,346	1,354	13,711
7	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,335	1,342	13,711
7	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,686	1,693	13,711
6	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,435	1,441	13,711
6	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,474	1,480	13,711
5	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,212	1,217	13,711
5	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,221	1,226	13,711
5	QZDASOINITQUSER	011866	000000000000002D	-1	2,096	2,101	14,440
4	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,482	1,486	13,711
4	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,697	1,701	13,711
3	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,355	1,358	13,711
3	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	1,487	1,490	13,711
2	QZDASOINITQUSER	011234	0000000000000025	BSMENGES	677	679	13,711
2	QZDASOINITQUSER	011866	000000000000002D	-1	2,091	2,093	14,440
2	QZDASOINITQUSER	011866	000000000000002D	-1	2,112	2,114	14,440
2	QZDASOINITQUSER	011866	000000000000002D	-1	2,115	2,117	14,440

Long transactions for DB server jobs report

8.10.7.2 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

Table description	SQL table
Job Watcher long transactions	QAIDRJWTXNSUM_<<MBRNAME>>

8.10.8 Lock Trace

This analysis is very specialized and requires that a PEX collection containing lock events has also been collected at the same time as the Job Watcher collection in order to produce any PEX data. It looks at jobs and call stacks in the Job Watcher data that identify certain wait conditions associated with performance problems.

Note: This analysis requires that the browse collections repository contains the associated PEX data needed Use General Functions -> Browse Collections (right-click) -> Full rebuild option.

The analysis will:

1) identifies jobs/call stacks that have specific lock wait criteria (enum = 5, CURRSTATE = WAIT and call stack contains both rmtmDelay* and program QDB*)

2) summarizes the time periods in #1 by lic wait object handle and taskcount, for each of these results, pex lock data is retrieved

3) for each summarized time period in the JW data, build a table of PEX lock event results for the same time period and job.

After running the analysis, no new features will appear in Job Watcher. Access the data from the SQL tables folder.

8.10.8.1 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

Table description	SQL table
Job Watcher jobs lock trace summary	QAIDRJWLCKSUM_<<MBRNAME>>
Job Watcher jobs lock trace details	QAIDRJWLCK_<<MBRNAME>>
PEX Analyzer lock trace data **Note:** This will only be generated if PEX data is found!	QAIDRJWLCKPEX_<<MBRNAME>>

8.10.9 Modules Waiting

This analysis identifies the top XPF program/modules/procedure and LIC procedures found in the call stacks captured by wait object type and LICWO.

Running this analysis will currently just produce an SQL table under the collection in the SQL tables -> Modules waiting folder.

8.10.9.1 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

Table description	SQL table
Modules waiting table	QAIDRJWMOD_<<MBRNAME>>

TOTAL	Total jobs/tasks/threads (JOBCNT)	Total intervals (INTERVAL	Objtype type (hex) and LIC wait object (LWAITOBJ)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Traceback table address (TBTADDR)
2447	109	164	0AQMo			QmRealDequeueMiQueue__FR11QmDeqPrefixPcR5MiPtr13QmDequeueType	FFFFFFFF39291430
2244	37	164	00SLW			sleep__6CfSyncFUIRUIQ2_4Rmpr18InterruptLevelTypeQ2_8TDQSEnum4Enum>	FFFFFFFF541D8700
1804	11	164	0AQMo	QWTMCMNL			3EB17045B100C870
1233	13	164	1AQQu	QP0SSRV2	QP0SWAIT	qp0swait__FP13qp0ssigwait_t	2D2666DDB8008E48
1233	13	164	1AQQu			waitOnSignal__8PxsgAreaFR17PxsgSignalMessageR15PxsgWaitOptionsRUI	FFFFFFFF544C9600
1204	25	164	00SLW	QP0LLIB1	QP0LLIB1	select	28689C6607050728
1032	11	164	00SLW	QSOSRV1	QSOSYS	poll	3B55BFF53E010F60
820	5	164	1AU62	QP2USER2	QP2API	runpase_common__FiPvT2	13AE087589017E70
820	5	164	1AU62			nsleep_common__FUIT1i	FFFFFFFF54B18020
779	124	164	00U74			do_sleepWait__12PpPaseThreadFQ2_8TDQSEnum4EnumUI	FFFFFFFF54B18CC8
779	124	164	00U74	QP2USER2	QP2API	runpase_common__FiPvT2	13AE087589017E70
331	5	164	0AQMd			QmRealDequeueMiQueue__FR11QmDeqPrefixPcR5MiPtr13QmDequeueType	FFFFFFFF39291430

Modules waiting SQL table

8.10.10 Create Indexes

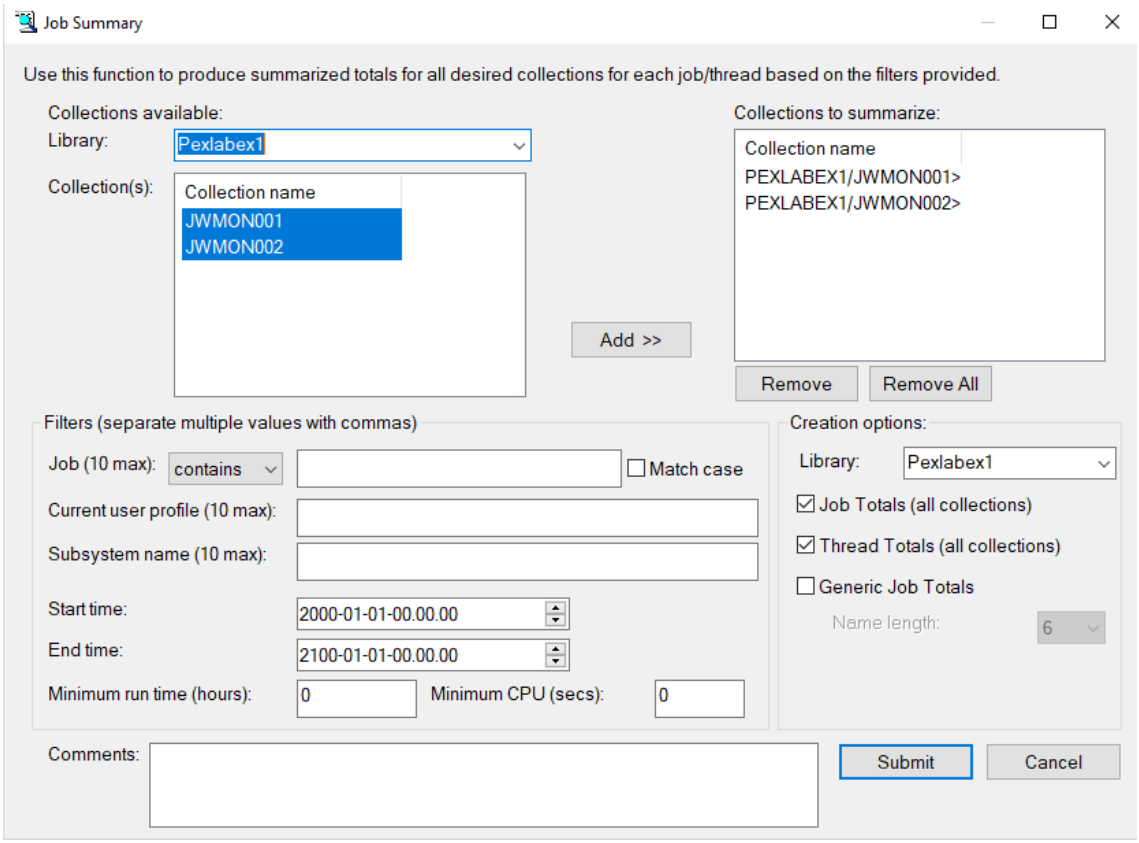
This analysis is used to build indexes that will improve performance of some of the graphing options in iDoctor. This does add SQL indexes to some of the Job Watcher files on the system. To remove these later, right-click the SQL Tables -> Create indexes folder and use the Delete… option.

8.10.11 Job Summary

The Job Summary analysis allows a user to build tables that add up job statistics across 1 or more collections. These summaries are stored in the SQL Tables folder in iDoctor.

Note: If the collections specified have not already been summarized (i.e. the Collection Summary analysis has not yet been ran), it will be ran automatically by running this analysis.

An example of this interface is:



Job Summary Analysis Window

The following table describes the various components of this window.

Option	Description
Library	The name of the library to select available collections from.
Collection(s) list	Displays the list of collections available in the current library to select from.
Add >>	Adds the selected collections in the Collection(s) available list to the Collections to summarize list.

Job (10 max)	This field indicates which jobs to include in the reports by specifying a portion of the job name. If you leave this field blank all jobs will be included in the report. Up to 10 job name values may be entered. Separate multiple values with a comma. The drop-down list allows you to specify if each value used should be a “contains” comparison or a “starts with” comparison.
Current user profile (10 max)	This is the list of up to 10 current user profiles to filter the analysis output on.
Subsystem name (10 max)	This is the list of up to 10 subsystem names to filter the analysis output on.
Start and end time	Use these fields to filter the data by time.
Minimum run time (hours)	If you wish to filter the job data by a minimum time the job ran, then enter a value in hours.
Minimum CPU (secs)	If you wish to filter the job data by a minimum CPU time used, then enter a value in seconds.

Comments	The comments field is used to apply a comment to all SQL tables generated by this analysis.
Collections to summarize	The list of collections to include in the analysis.
Remove / Remove All buttons	Removes collections from the Collections to Summarize list.
Creation options: library	This field allows the user to specify a different library than the current one for the SQL tables generated.
Job totals (all collections)	If checked a report will be generated that summarizes the data across all collections by job.
Threads totals (all collections)	If checked a report will be generated that summarizes the data across all collections by thread/taskcount.
Generic job totals	This option must be checked in order to include additional reports that summarize the metrics by generic job name. The number of characters to use in the generic job name can be specified using the **Name length** drop-down list.

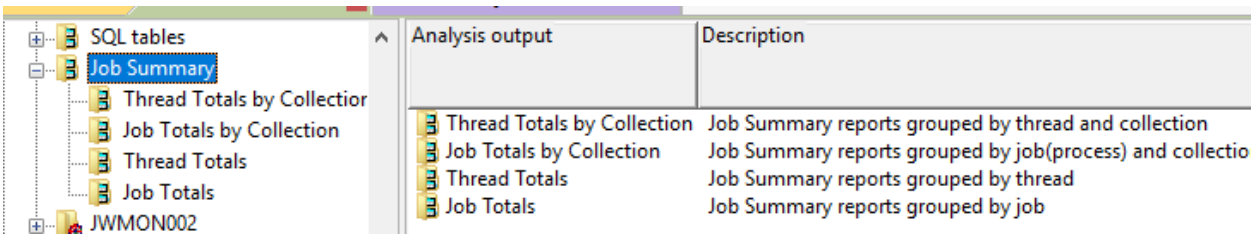
Pressing the Submit button will run the analysis over the desired collections and place the results in the SQL Tables repository. One folder is created for each of the SQL table creation options checked.

When running the analysis the progress is shown in the Remote SQL Statement Status View.

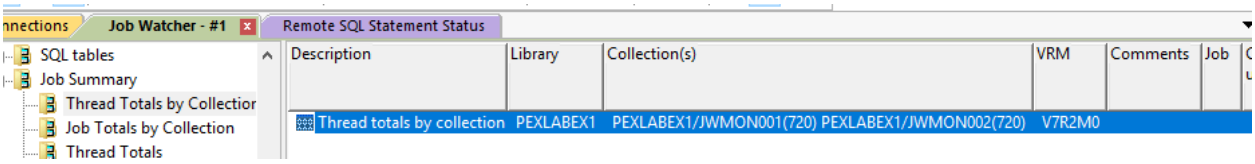
After running the analysis, the following features become available in Job Watcher under the SQL tables or Job Summary folder.

8.10.11.1 Job Summary report folders

Under the Job Summary folder you will find the report folders based on the types generated when running the analysis.



Clicking on one of these will show the collection(s) used on the Job Summary as well as any parameter filtering used when creating the analysis.



Thread Totals by collection folder

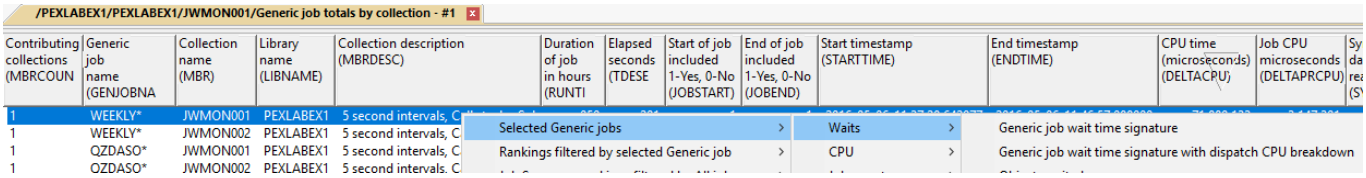
8.10.11.2 SQL Tables Generated

This function generates 1 table for each type of report created (where <<X>> is a unique number):

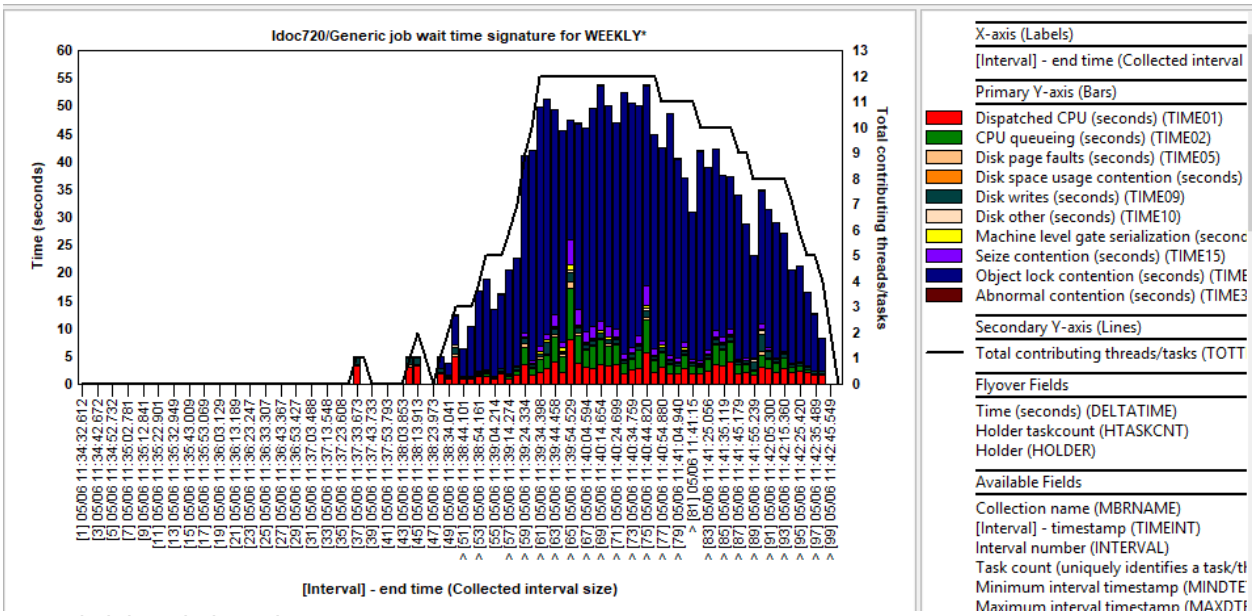
Table description	SQL table
Generic job totals	QAIDRJW7SUM_<<X>>
Job totals	QAIDRJW4SUM_<<X>>
Thread totals	QAIDRJW3SUM_<<X>>
Generic job totals by collection	QAIDRJW6SUM_<<X>>
Job totals by collection	QAIDRJW2SUM_<<X>>
Thread totals by collection	QAIDRJW1SUM_<<X>>

8.10.11.3 Selected (Job/Thread/Generic Job) drill down

By opening any of the Job Summary SQL tables and selecting one or more rows and right-clicking you will have an option to graph the desired selections over time. If the analysis was done over multiple collections and currently using a report not “by collection” then you will be able to scroll and see the data across multiple collections.



Selected Generic jobs drill down options

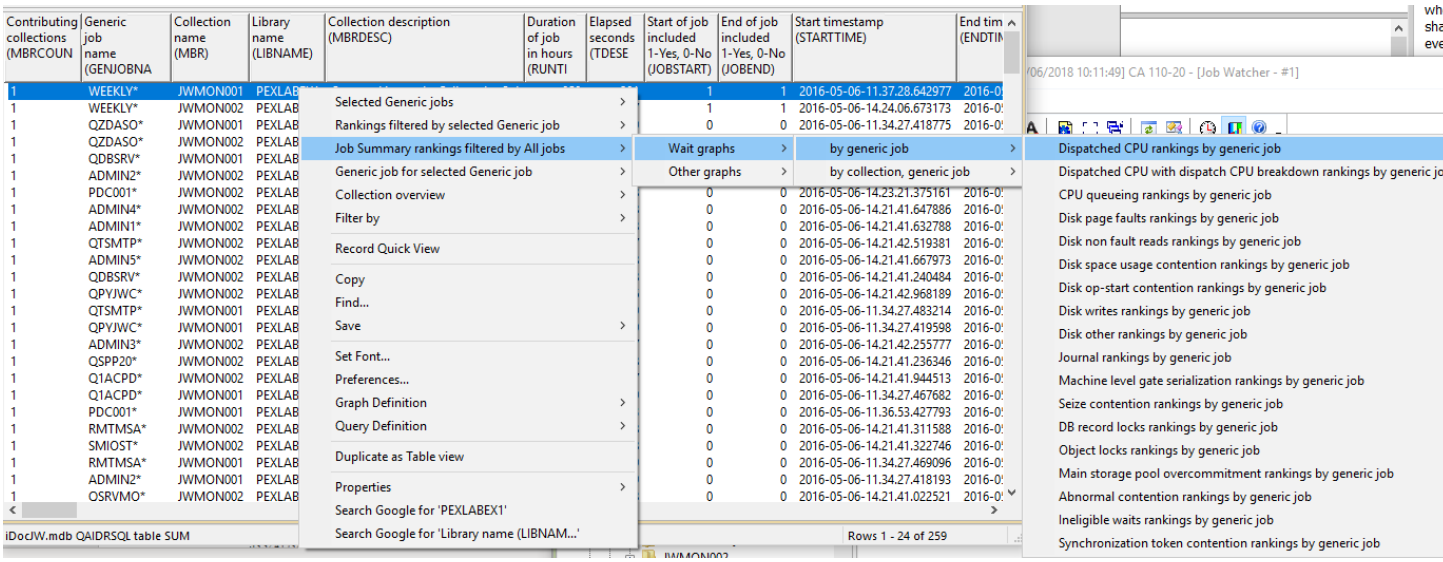


Generic job wait time signature

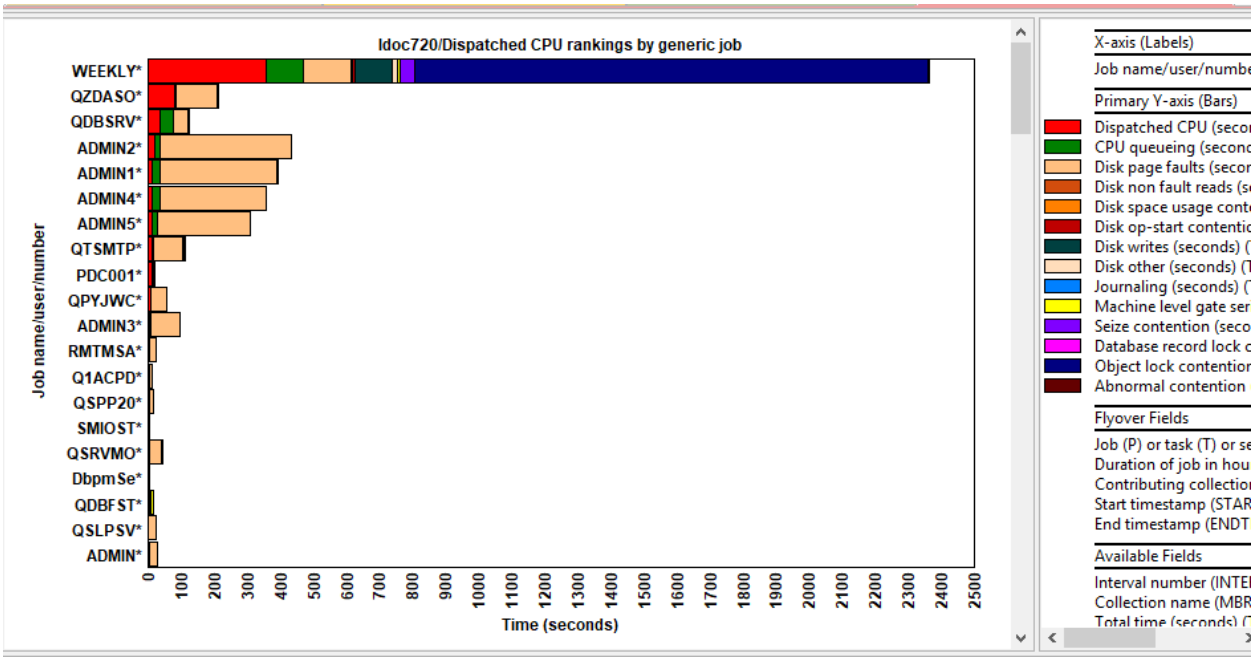
8.10.11.4 Job Summary rankings filtered by X graphs

These options provide wait bucket and other statistics for the desired jobs. What is displayed is controlled via the Filter by -> menu and could be either:

1) All jobs
2) Selected jobs
3) Prompt the user for a generic job name

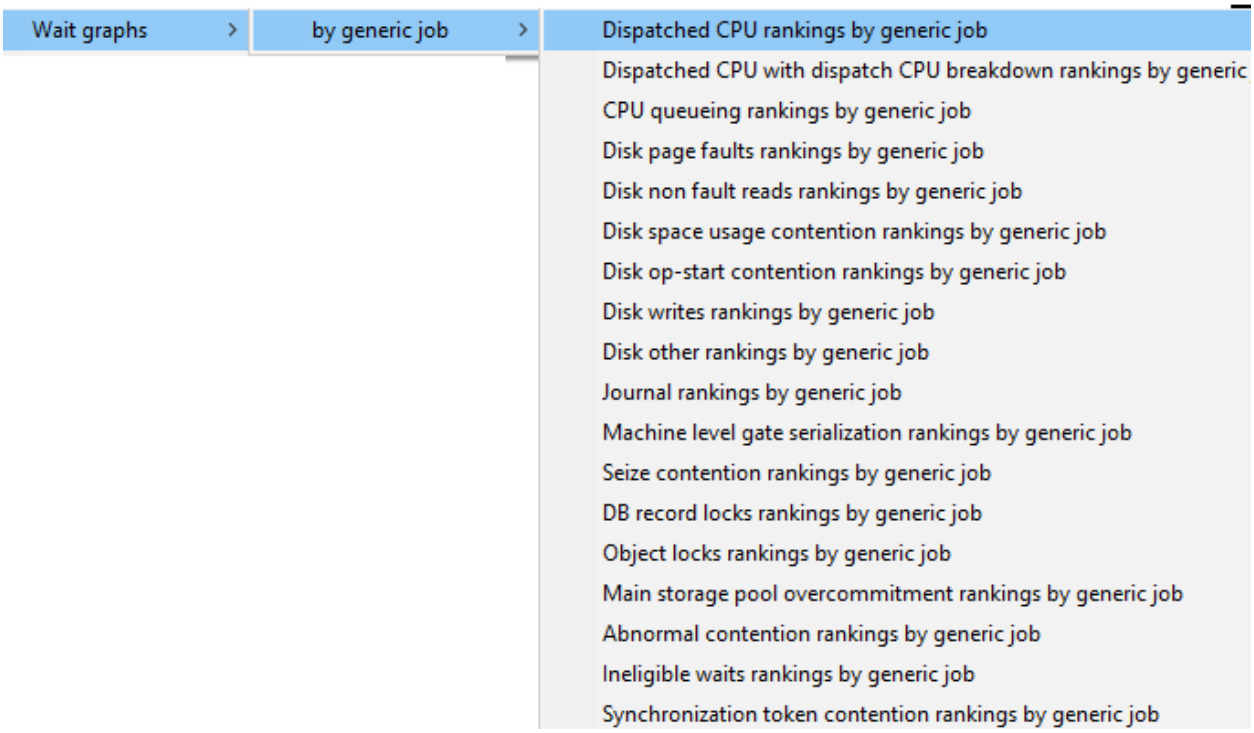


Job Summary rankings filtered by All jobs example



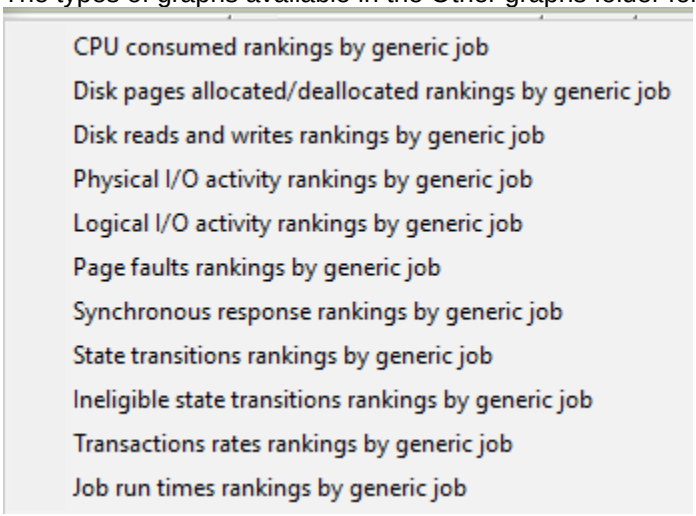
Dispatched CPU rankings by generic job (for all jobs)

The types of rankings wait graphs that can be generated are based on CPU and all the “interesting waits”. An example is:

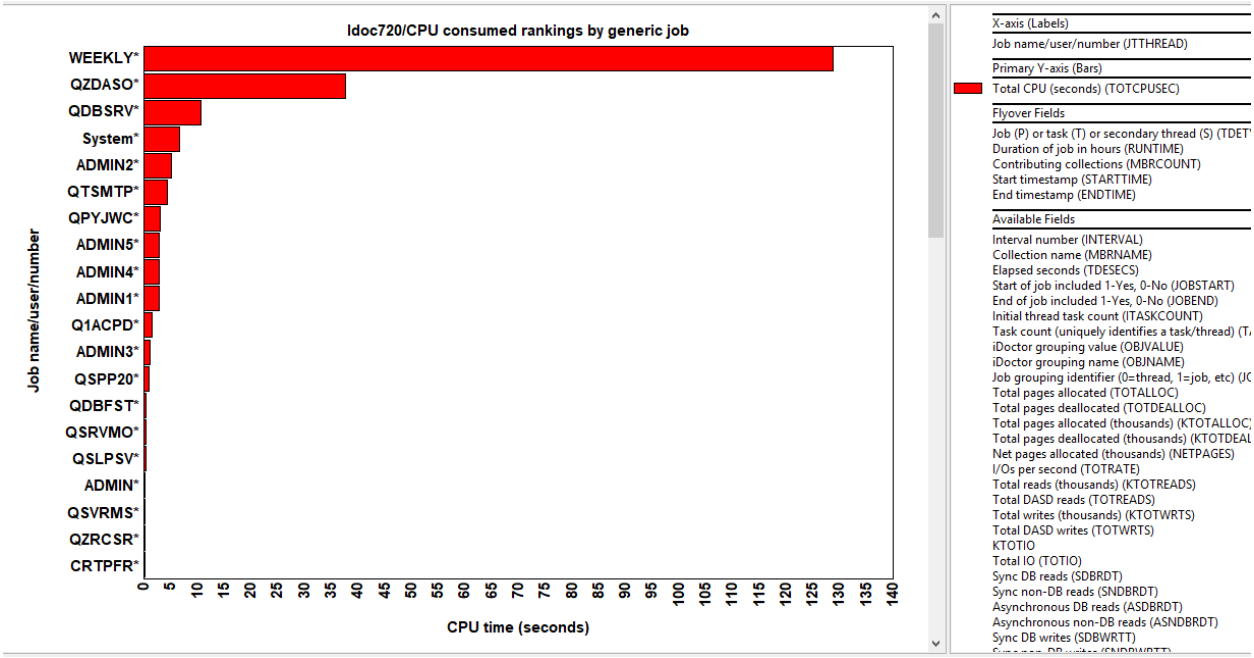


Job Summary rankings filtered by X -> Wait graphs

The types of graphs available in the Other graphs folder for this option are:



Job Summary rankings filtered by X -> Other graphs



CPU consumed rankings by generic job

8.10.12 Retrieve program/modules details

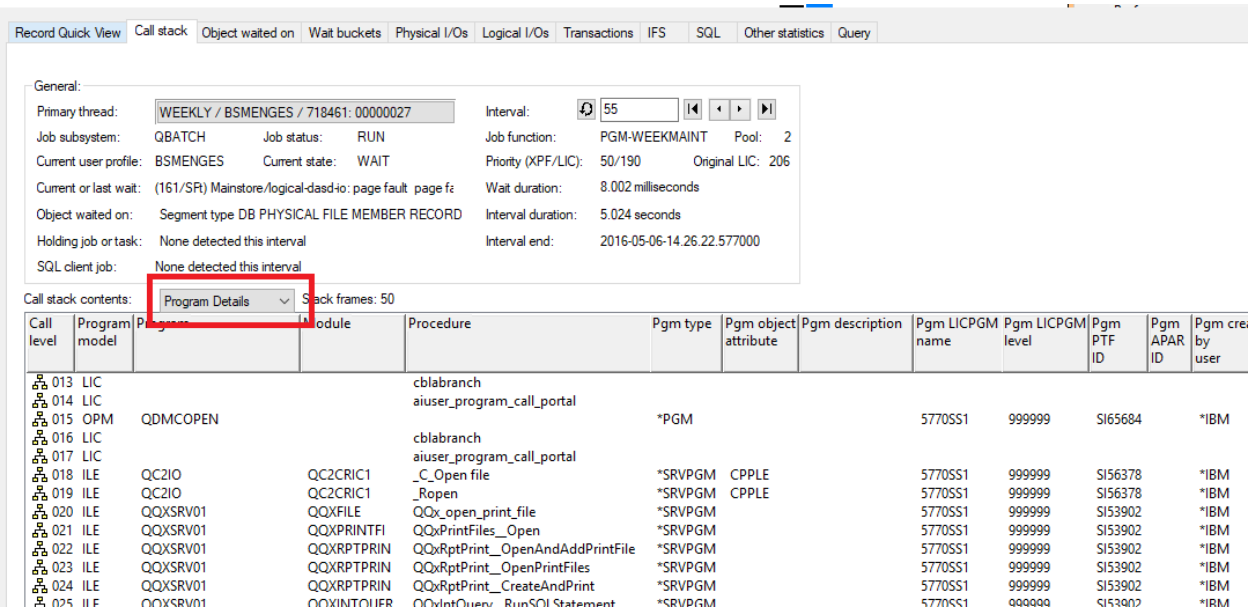
This analysis will capture additional details about the programs and modules found in the call stack data in Job Watcher. It must be executed on the same system that the data was collected on or no data will be generated. If this data is available, then the Interval Details -> Call stack window provides an additional mode to view these details with the call stack which includes things like the PTF number associated with each program found on the call stack.

Note: This analysis can consume a large amount of disk space in some environments depending on the number of programs and modules found in the call stacks. Because of many IBM i API calls made could leave many userspaces in QRPLOBJ library. It could also take a long time to complete.

Tip: This analysis only needs to be executed once per collection in a library. However, if programs have changed on the system, then newer collections would not reflect the latest program changes on the system.

8.10.12.1 Call stack – program details mode

If this analysis has been executed, then the Program details mode will become available within the call stack window. By using this option several additional columns will be added to the end of each call stack call level.



Call stack with Program Details mode enabled

8.10.12.2 SQL Tables Generated

This analysis creates physical file/members in the following tables where the member name matches the Job Watcher collection name.

Table description	SQL table
JW program/service program module information	QAIDRJWMOD
Outfile for DSPOBJD	QAIDRJWOBJ
JW service program information	QAIDRJWSVC

8.10.13 Destroy all host variable data in QAPYJWSQLH

This option **permanently** removes all host variable data in the Job Watcher collection. In some cases, if sensitive data may exist within that file this may be necessary before sending the data to IBM or others.

No features or data is generated by this analysis.

8.10.14 Change/Restore sensitive user data

These options are used to hide or replace potentially sensitive data in Job Watcher. It will update the following things in the Job Watcher files:

1) User programs names and library names
2) User module names
3) User program/module procedure names
4) System name and system serial number
5) User job names, current user profiles, wait object names and holders.

Note: This data is changed but not lost, it can be restored again using the Restore sensitive user data analysis. This will update the Job Watcher files QAPYJWTDE, QAPYJWRUNI and QAPYJWPROCI to their original state.

8.10.14.1 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

Table description	SQL table
TDE cross reference file	QAIDRJWXRFTDE_<<MBRNAME>>
RUNI cross reference file	QAIDRJWXRFRUNI_<<MBRNAME>>
PROC cross reference file	QAIDRJWXRFPROC_<<MBRNAME>>

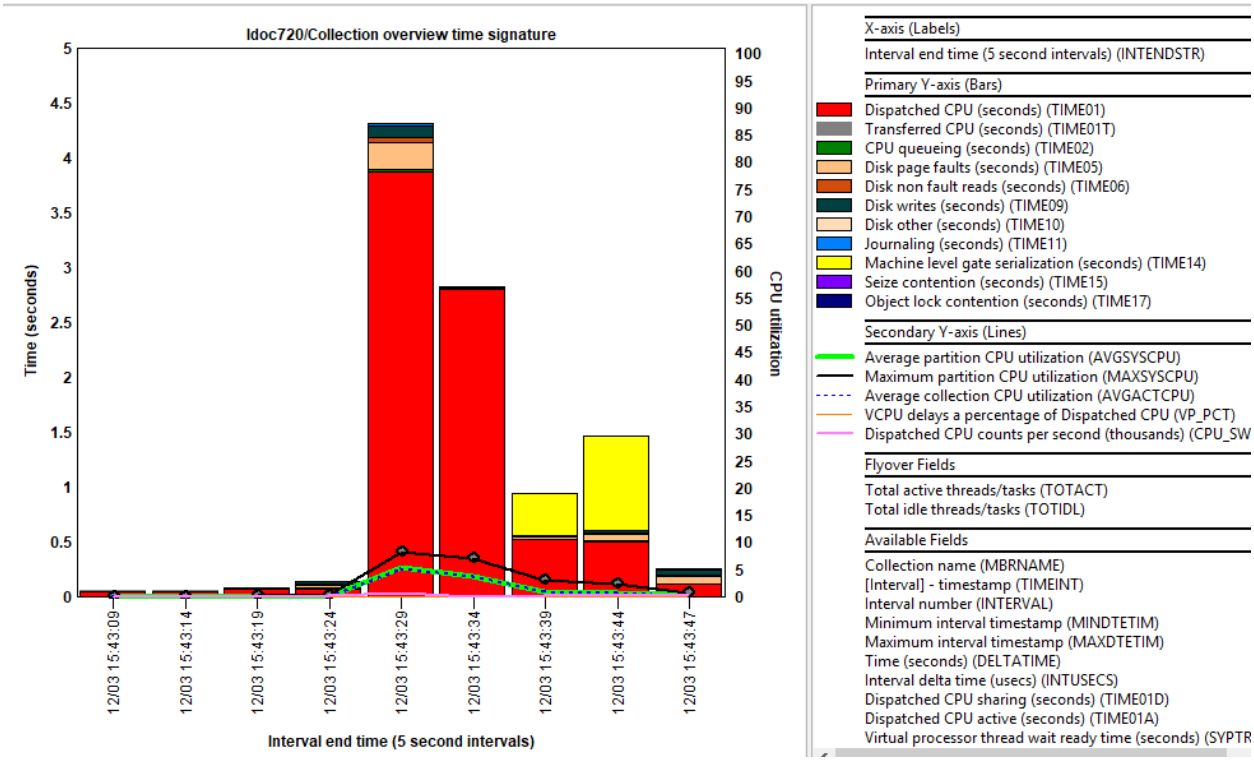
8.11 Graph notes

This section discusses the types of graphs found in a Job Watcher collection and how to use them.

It also covers briefly a discussion on “interesting” vs “idle” waits and on CPU related fields shown in many Job Watcher graphs.

8.11.1 CPU notes

Many of the Overview graphs show different types of CPU utilization.



Collection overview time signature

Average partition CPU utilization (green line) – This is the average CPU utilization for each summarized interval as collected by the LPAR.

Maximum partition CPU utilization (black line) – Because each bar in the graph could contain data from several intervals, this is the highest partition average CPU utilization that occurred. It is not a true maximum that occurred but the maximum of the average that occurred in those intervals being summarized.

Average collection CPU utilization (blue dashed line) – This is CPU utilization as taken ONLY from the Job Watcher jobs captured during the collection.

A Job Watcher collection is unlikely to contain all jobs on the system because Job Watcher will 'miss' job CPU contributions if they live and die within a single Job Watcher interval. These contributions don't get recorded in the main JW file QAPYJWTDE.

To account for this Job Watcher also collects CPU statistics for the entire system while the collection is running. We call that "partition CPU utilization" above.

Given the reasons above, sometimes collection CPU utilization will be less than the partition CPU utilization which can be interesting. (Possibly lots of short-lived threads/tasks started up and were within the "collection CPU" statistic).

Avg vs maximum is used because the time interval grouping can be configured to something greater than the collected interval size (1 min, 10 min, etc).

If you are looking at the collected interval size however, they will be the same and the green and black lines merge together.

VCPU delays as a percentage of Dispatched CPU – This value shows the amount of time virtual processors spent waiting to run in relation to the dispatched CPU time used.

The formula for this is (((SYPTREADY + SYPTLATEN) / DOUBLE(1000)) / TIME01) * 100.00 and is multiplied by 100 so it will graph well with the other metrics on the secondary Y-axis.

SYPTREADY	Virtual processor thread wait ready time. The elapsed time in milliseconds that ready to run threads of the partition's virtual processors waited to be dispatched while entitled capacity was exhausted.
SYPTLATEN	Virtual processor thread dispatch latency. The elapsed time in milliseconds that ready to run threads of the partition's virtual processors waited to be dispatched while entitled capacity was not exhausted and a physical processor was not available.
TIME01	This is the Dispatched CPU time spent (in seconds)

Dispatched CPU counts per second – This value indicates the number of times the CPU entered a run state as a rate per second (in thousands)

8.11.2 “Interesting” wait buckets

Job Watcher (and Collection Services Investigator) both make use the IBM i wait buckets instrumented on the system. There are 32 wait buckets, and these cover all possible wait states that a job can be in.

Many of these buckets are associated with idle waits and generally are not of any interest from a performance standpoint. For that reason, many iDoctor graphs only show the wait buckets that are interesting for a performance perspective.

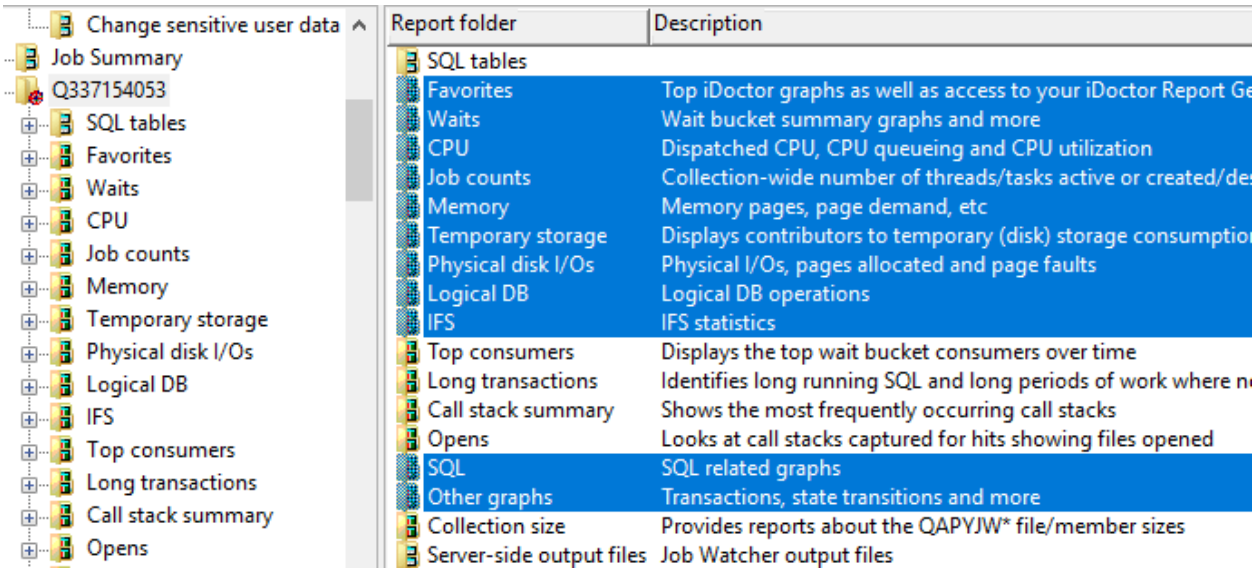
The interesting wait buckets are:

Bucket	Description
1	Dispatched CPU
2	CPU queueing
5	Disk page faults
6	Disk non-fault reads
7	Disk space usage contention
8	Disk op-start contention
9	Disk writes
10	Disk other
11	Journaling
14	Machine level gate serialization
15	Seize contention
16	Database record lock contention
17	Object lock contention
18	Ineligible waits
19	Main storage pool overcommitment
20	Journal save while active (7.2+ only)
31	Synchronization token contention
32	Abnormal contention

8.11.3 Overview graphs

The overview graphs are time-based and summarize the data in various ways over time. These show the data over the entire collection.

These graphs are contained within several folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the desired menus.



Overview Graphs in Job Watcher (selected)

Each folder contains a series of graphs. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Often most of these graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one. You can also use the Graph Compare icon on the toolbar of the Main Window in order to perform comparisons between graphs.

Tip #1: Use the clock icon on the toolbar to change the default Time interval size. This is useful if you have many thousands of intervals and wish to group those intervals into fewer bars than would be shown if you graphed at the Collected interval size.

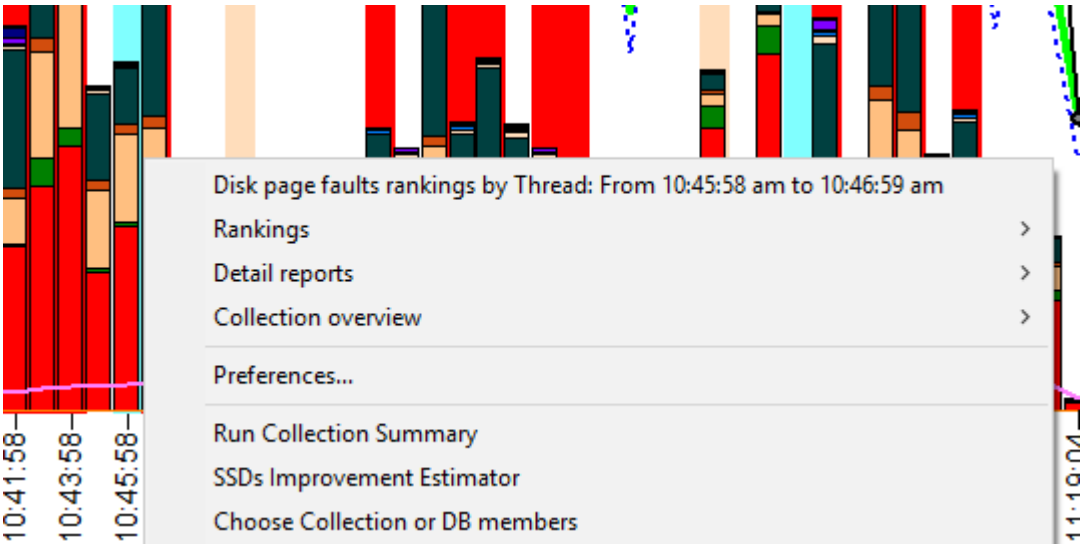
Tip #2: You can make time selections by clicking the 1st bar, hold down the shift key and clicking the last bar of the desired time range and right-click a bar and wait bucket color within that time period in order to drill down into the jobs experiencing the most amount of time in that wait bucket you right-clicked on.

Tip #3: If you wish to group job data in the ranking graphs with groupings larger than thread (by generic job, etc) you must run the Collection Summary analysis first.

8.11.3.1 Drilling down into Rankings graphs

When drilling down into ranking graphs (from Collection-wide graphs) you can select the desired time period of interest by holding down the shift key and clicking the 1st and last bars of the desired time period. Then right-click on one of the bars in the time period and pick the desired drill down graph.

This action will look something like this:



Drilling down from a Collection Overview Time Signature graph into ranking graphs

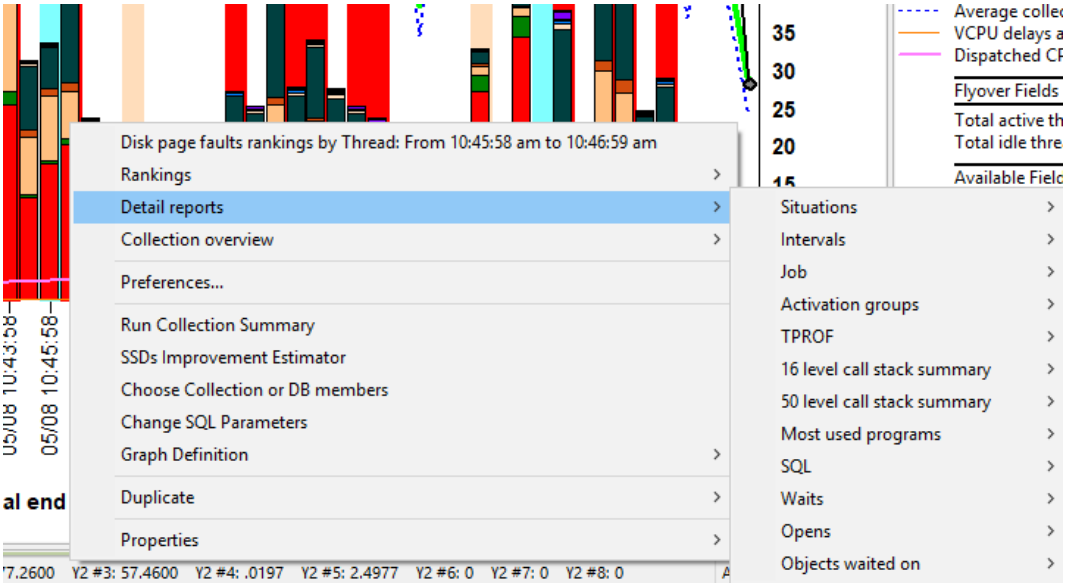
The default action at the top is Thread signatures ranked by disk page faults. This is because the bucket right-clicked on is the disk page faults time bucket. If it had been CPU time then it would give a default drilldown of Thread signatures ranked by Dispatched CPU instead.

However in the event that you don't want to drill down into wait buckets, you can use one of the Rankings menus and pick from there the desired graph you want.

8.11.3.2 Drilling down into Detail reports

Another drilldown option from the Collection-wide graphs is found under a menu called "Detail reports". This menu offers a series of table views that provide quick access to much of the raw data found in the collection.

An example of this menu and list of report categories it contains is:



Detail reports menu options

These reports are based on either the single interval or time range selected.

8.11.3.3 Run Job Summary option

The Create Job Summary menu option allows the user to create a Job Summary for only the time period selected.

PLEASE NOTE: This option is only shown if a time range (> 1 interval) has been selected.

8.11.3.4 Split Collection option

The Split Collection option allows the user to create a new collection from the selected time period in the current one.

PLEASE NOTE: This option is only shown if a time range (> 1 interval) has been selected.

8.11.3.5 Run Collection Summary

This option allows the user to run a filtered Collection Summary analysis for the currently selected interval or time period.

See the section on the Run Collection Summary interface for more information.

8.11.3.6 SSDs Improvement Estimator

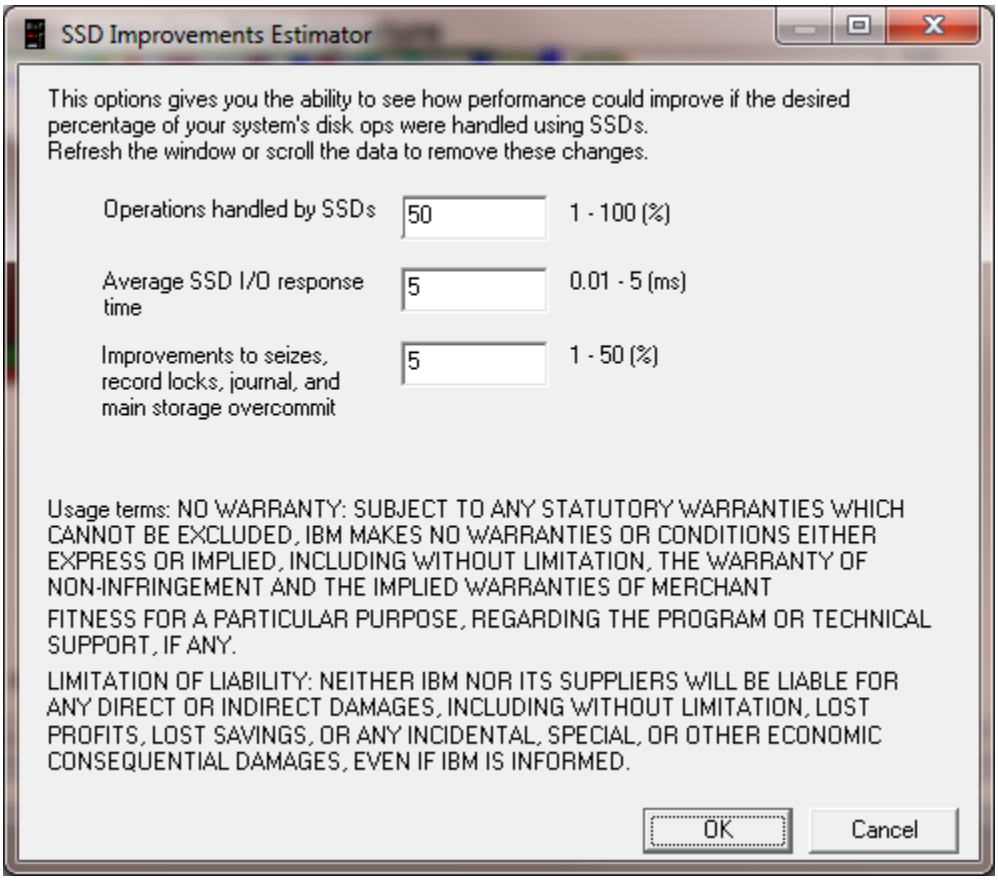
The SSDs improvement estimator is intended to show the user how a wait bucket (over time) graph might look if SSDs were added to the current system. It alters wait bucket graphs in CSI/JW to show possible disk time reductions if SSDs were installed based on % of SSDs, avg SSD I/O response time and estimated improvements to other types of waits besides disk.

Here are a few tips when using this option:

1) This function primary modifies disk wait times (buckets Disk page faults and disk non fault reads) If these times are not present on the graph then don't bother.

2) Use the clock icon to summarize the data so that all the data you wish to estimate improvements for is on the current page of the graph. As soon as you scroll the graph the changes are lost.

An example of the interface follows:



The estimations given are truly that and may vary significantly from the data captured after SSDs are installed.

8.11.4 Ranking graphs

In Job Watcher the ranking graphs are ranking jobs by a desired metric using a job grouping such as by thread, by job, by user name, etc. Job Watcher currently has 12 job groupings used. All 12 are available only if the Collection summary analysis has been ran on a collection. Otherwise the only ranking graph type available is **by thread**.

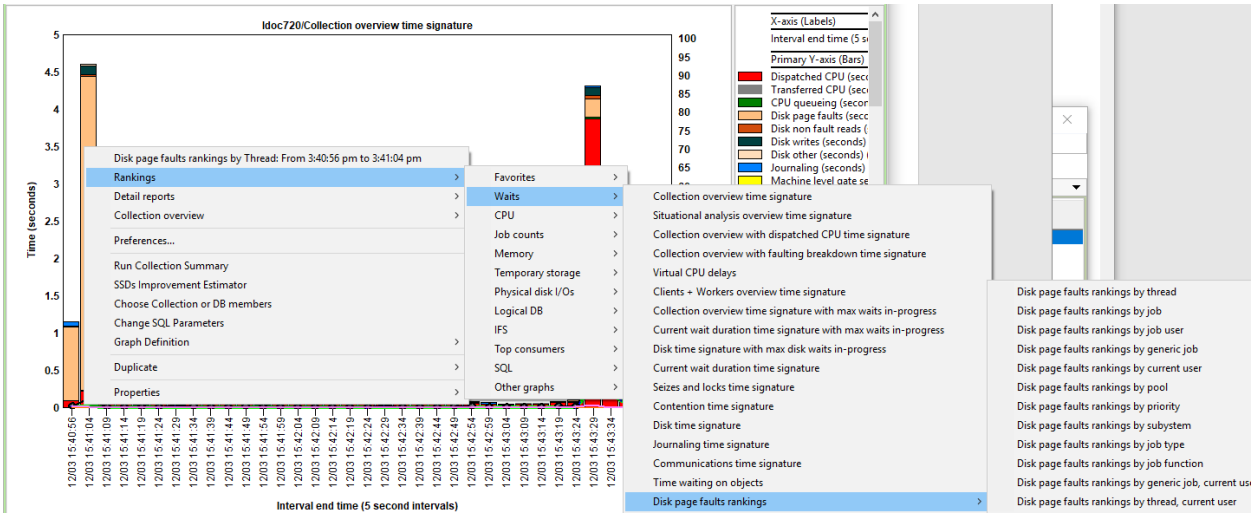
The types of job groupings available in Job Watcher are:

- by thread
- by job
- by job user
- by generic job
- by current user
- by pool
- by priority
- by subystem
- by job type
- by job function
- by generic job, current user
- by thread, current user

Job groupings list

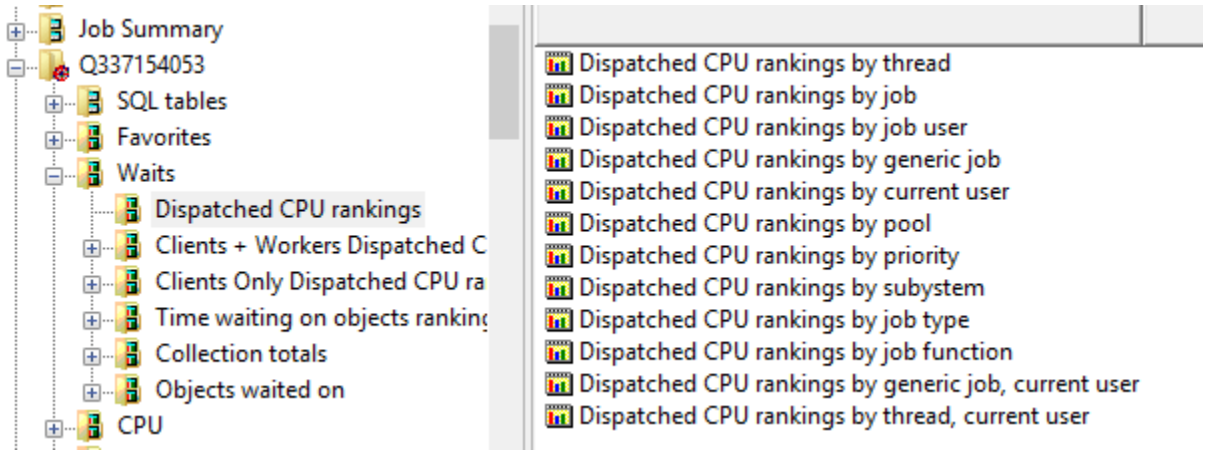
Ranking graphs are accessed in 1 of 2 ways:

1) As a drill-down from an overview graph under the Rankings -> (Pick a metric) rankings -> By job grouping X menu:



Rankings drill down options from a Collection overview time signature

2) Directly under the collection within one of the ranking graphs folders such as Waits -> Dispatched CPU rankings.



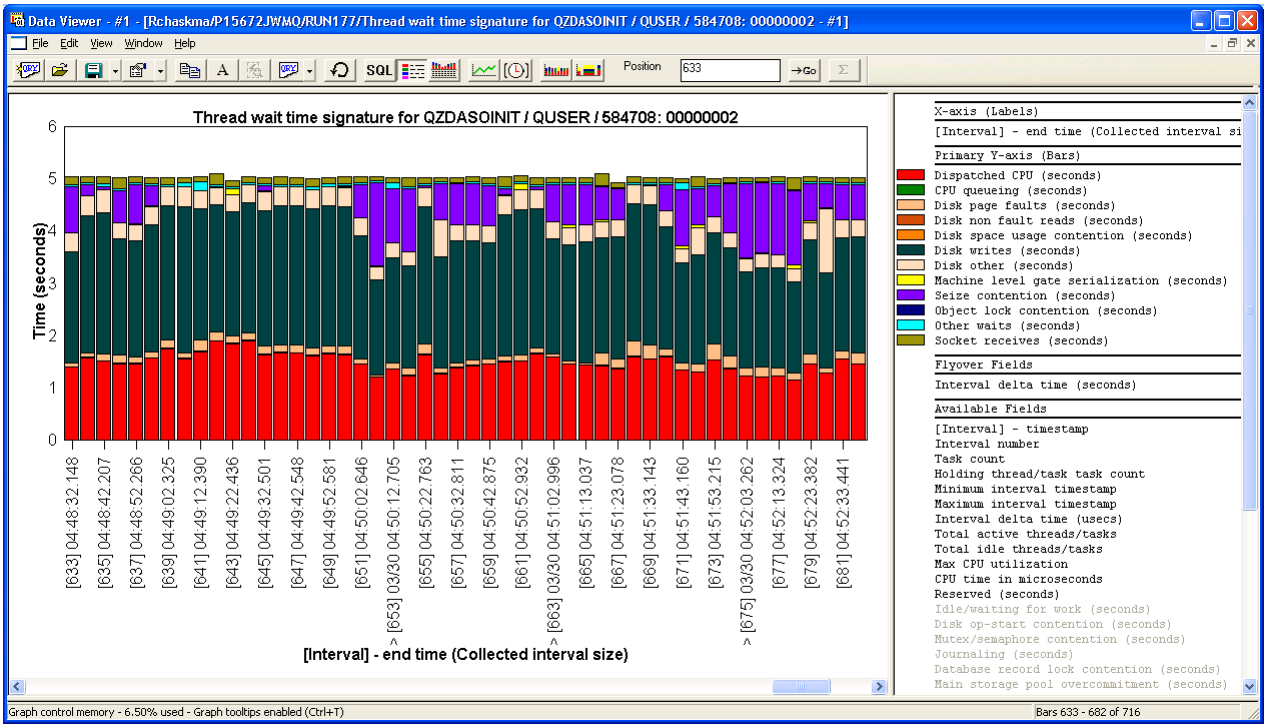
Waits -> Dispatched CPU rankings folder

From a ranking graph, users can select one or more job groupings and graph that selection over time.

8.11.5 Selection over time graphs

This graph type shows the currently selected thread (or job, generic job, user, etc) over time. The size of time interval is configurable (greater than the collected size only) by using the clock icon on the toolbar.

By default, when opening this type of Job Watcher graph the data will be shown at the collected interval size for the best level of granularity. However, if this requires too much scrolling to the see a big picture view of this job, use the clock icon to effectively “zoom out”.



Selected Thread graph for a QZDASOINIT job

If viewing a Thread this type of graph contains a visual indicator that the thread had a holder for a particular interval. This is a ^ symbol on the X axis. There are 3 shown in the example above.

From any of these intervals you can double-click to go to the Interval details and view the call stack and holder information. Or you can also right-click the desired interval and a “Holder” menu will appear allow you to view the Holder’s call stack, perform a Holder chase or to graph the holder job over time.

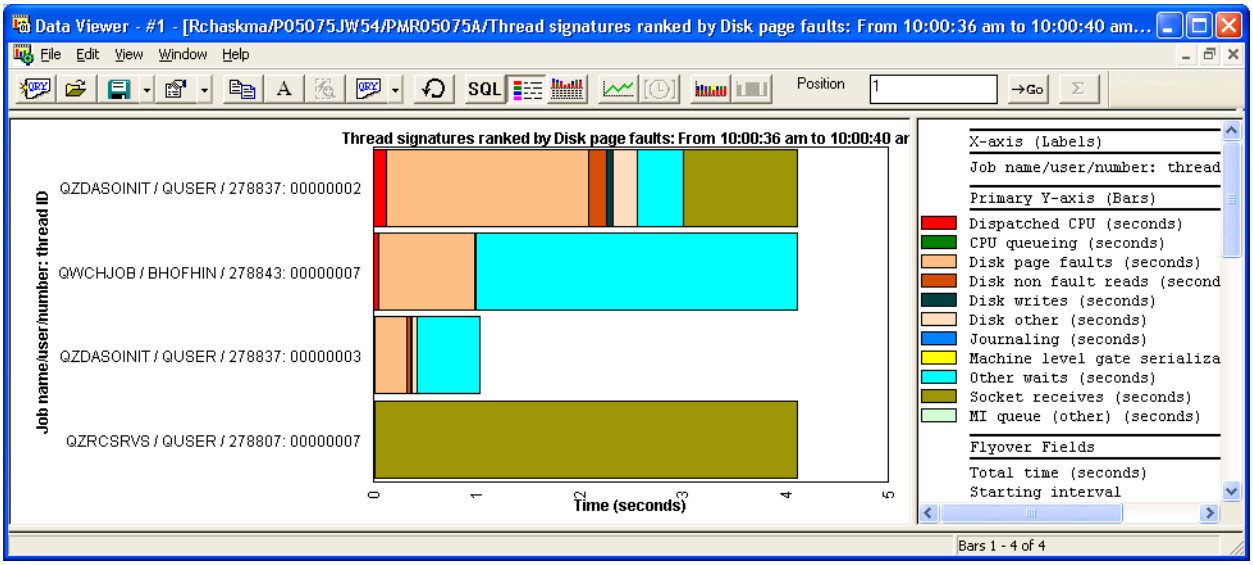
8.11.5.1 Drilling up

From the Selection over time graphs thread graphs you can select a time period of interest and right-click to have the same Rankings graphs options available to you but over the new time period. This allows you to navigate through your job over time, find something of interest and then compare that time period with the rest of the jobs on the system.

8.12 Rankings Graphs (via the Collection-Wide graphs)

This section covers the ranking graphs in Job Watcher that are available as drilldowns from any of the Collection-wide graphs.

Ranking graphs show a list of objects (jobs, threads, units, etc) ranked by the desired metric.



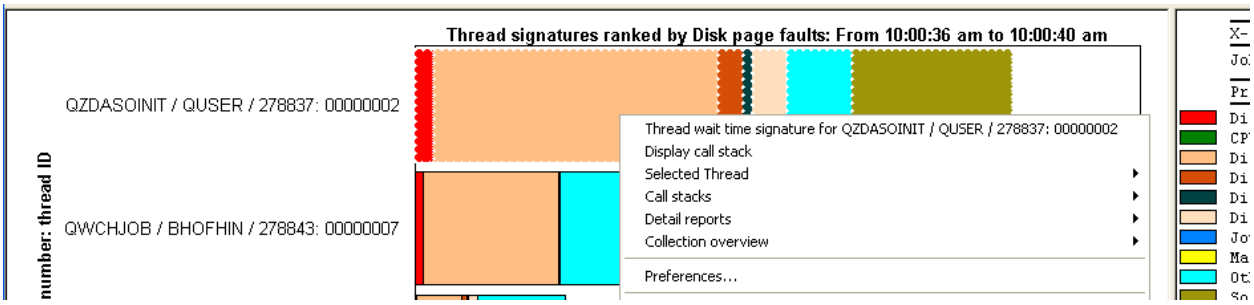
Threads ranked by disk page faults

Note: If the collection has NOT been summarized then the only ranking graphs available are the Thread ranking graphs.

8.12.1 Drilling down to Selected Thread/Job/etc graphs

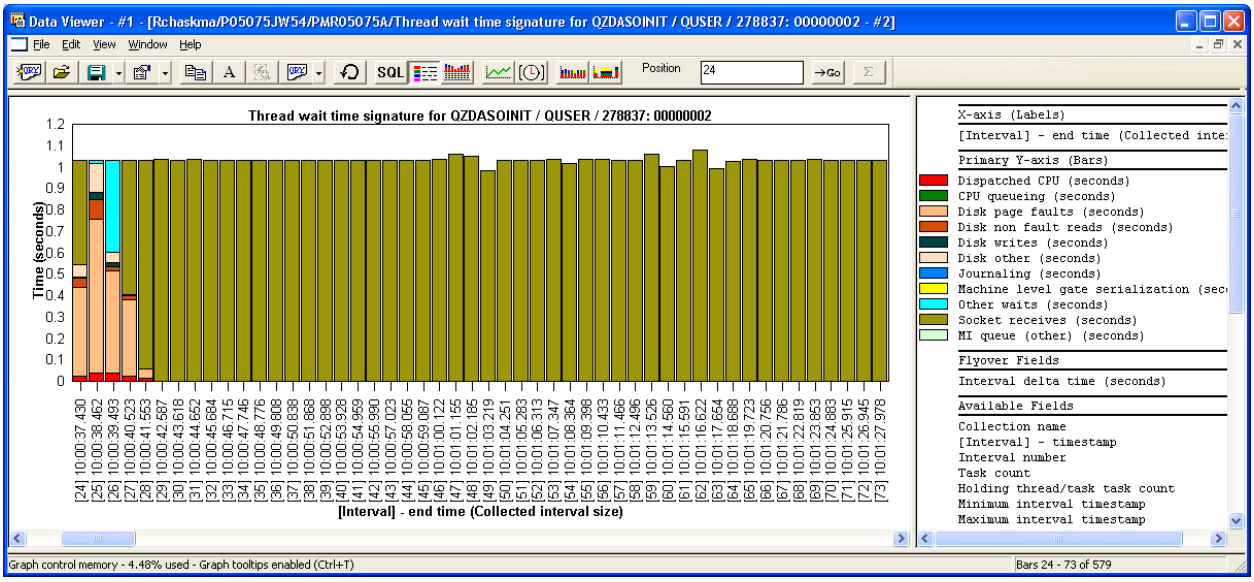
When drilling down from rankings graphs the default (top) menu is to show the current thread/job/job user/generic job, etc in a time interval graph. By default the time interval size is set to the collected interval size even if this is not your default preference. This was done deliberately since most users prefer to work with the collected interval size when dealing with this type of graph.

However, you can easily change this if desired by clicking the clock icon on the Data Viewer's toolbar.



Drilldown from a thread rankings graph to show a single job over time

To drill down simply right-click a job/thread/etc and choose the desired option. Typically the top menu is what you want, but you could also use the Selected Thread menu (name varies depending on the graph grouping) if you need to select a different graph than the one you are currently working with.



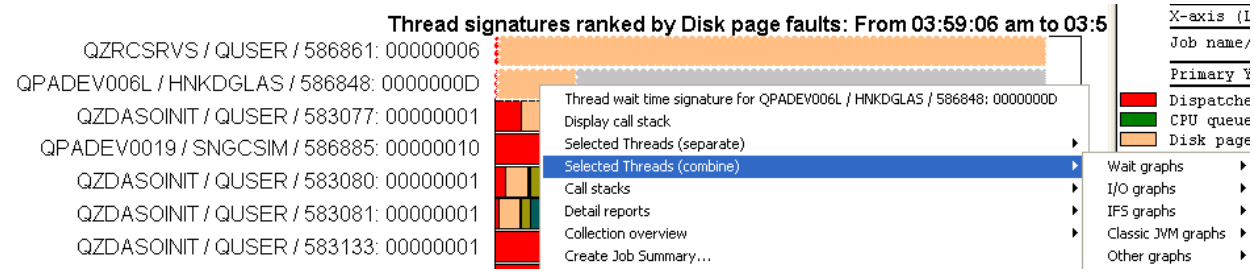
Thread wait time signature for a specific thread

In the example above the graph is showing the wait bucket times for a single thread over time. Also the graph is automatically scrolled to the start of the time interval displayed in the previous rankings graph.

8.12.2 Analyzing multiple threads/jobs/etc

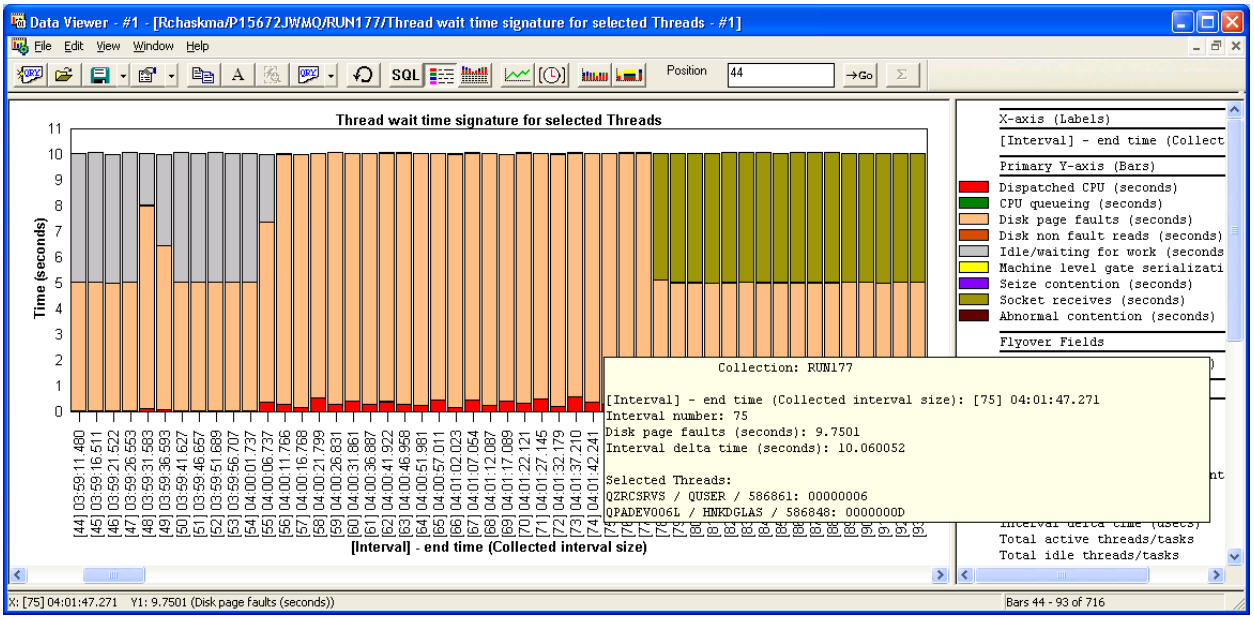
From a rankings graph you can hold down the Ctrl key and click multiple jobs/threads (depending on the graph grouping) and then when right-clicking you will have the option to combine the data from multiple jobs/threads/etc into a single graph. Just choose the menu option called Selected Thread (combine).

A result of this action looks like this:



Selected Threads (combine) menu

The resulting wait graph over time for these threads will combine the wait times from all selections into a single graph.



Graph combining data from multiple threads

The flyover will include the selections that make up the graph.

Also if you wish to select multiple jobs and have a different graph created for each one, (in one step), use the Selected Thread (Separate) menu.

8.12.3 Display call stack menu

The Display Call Stack menu from a rankings graph allows the user to quickly go to the call stack for the 1st interval in the time period indicated at the top of the rankings graph for the selected thread/task.

8.12.4 Call stacks menu

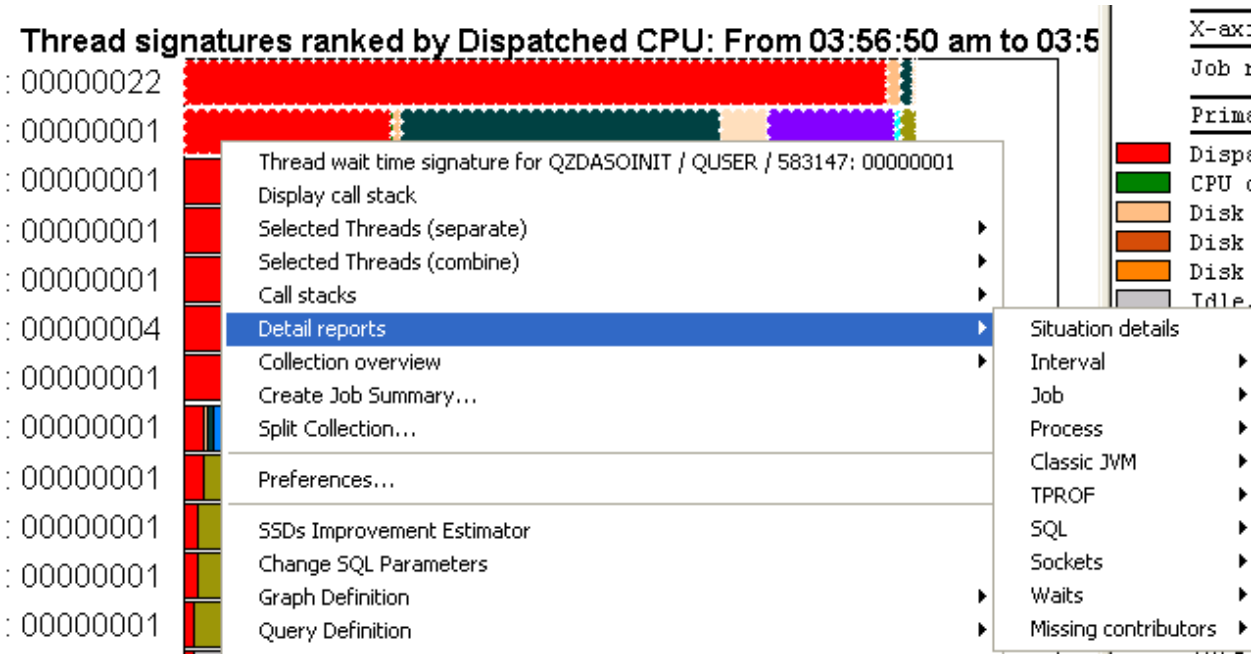
The Call Stacks menu is a special option that allows the user to display all the call stacks for all threads in the selected job from a rankings graph in various ways. This option was initially designed for use when analyzing JVM (Java), but could be used for any type of job if you pick the appropriate menu option from the list.

Menu	Description
Call stacks for selected job, interval X	This report shows all threads call stacks for the current job for the first interval of the time range selected (in the rankings graph)
Call stacks for selected job, interval X to Y	This report shows all threads call stacks for the current job for the entire time period selected in the rankings graph. Be careful not to pick too large of a time period for this one as the report could take a long time to generate.
Classic JVM call stacks for selected job, interval X	This report shows all threads call stacks for the current job for the first interval of the time range selected (in the rankings graph). This report contains extra fields specific to the classic JVM data. If the job selected is not a classic JVM job then this option won't provide data.

8.12.5 Drilling down into Detail reports

Another drilldown option from the Rankings graphs is found under a menu called "Detail reports". This menu offers a series of table views that provide quick access to many of the raw data found in the collection.

An example of this menu and list of report categories it contains is:



All of these reports are filtered on the time period selected by the rankings graph as well as the current selections (jobs/threads) from the graph.

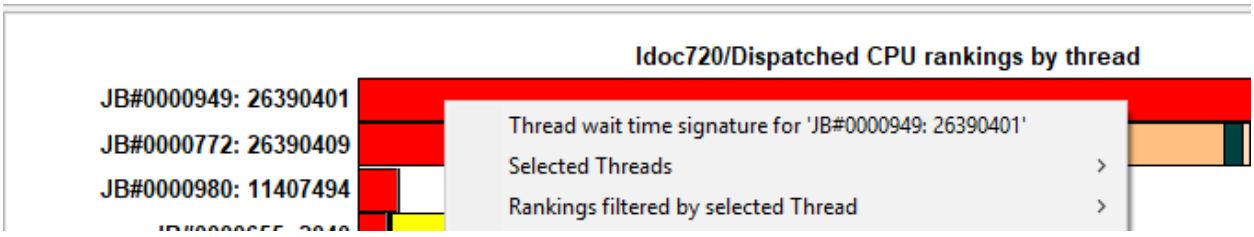
PLEASE NOTE: This drill down option is only visible if the preference on the Job Watcher tab called "Display advanced reporting options" is enabled.

8.12.6 Collection overview menu

This is identical to the menu described previously.

8.12.7 Job selection overtime graphs

These graphs are initiated as a drill-down from any ranking graph. Typically, these can be most-easily accessed by right-clicking on a job in the ranking graph and using the first menu option that appears. This will by default show the same graph but for the selection and over time.



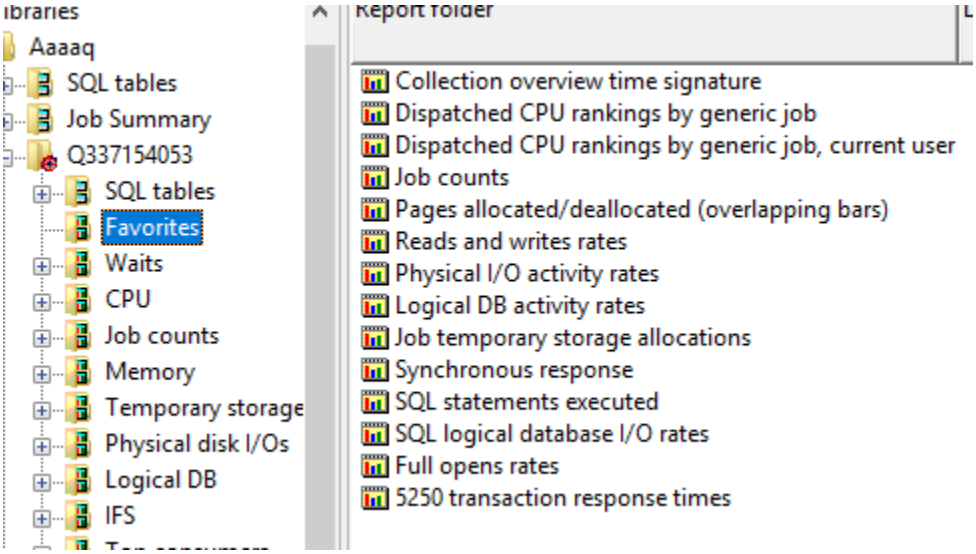
Dispatched CPU rankings drill down options

Tip: If another graph type is desired then use the Selected Threads menu or Rankings filtered by selected Thread options.

8.13 Favorites

This folder contains a list of graphs most commonly used and are great starting points. If you are new to iDoctor these are your recommended graphs to use.

Note: Some graphs only appear at certain releases or require the Collection Summary analysis to be ran.



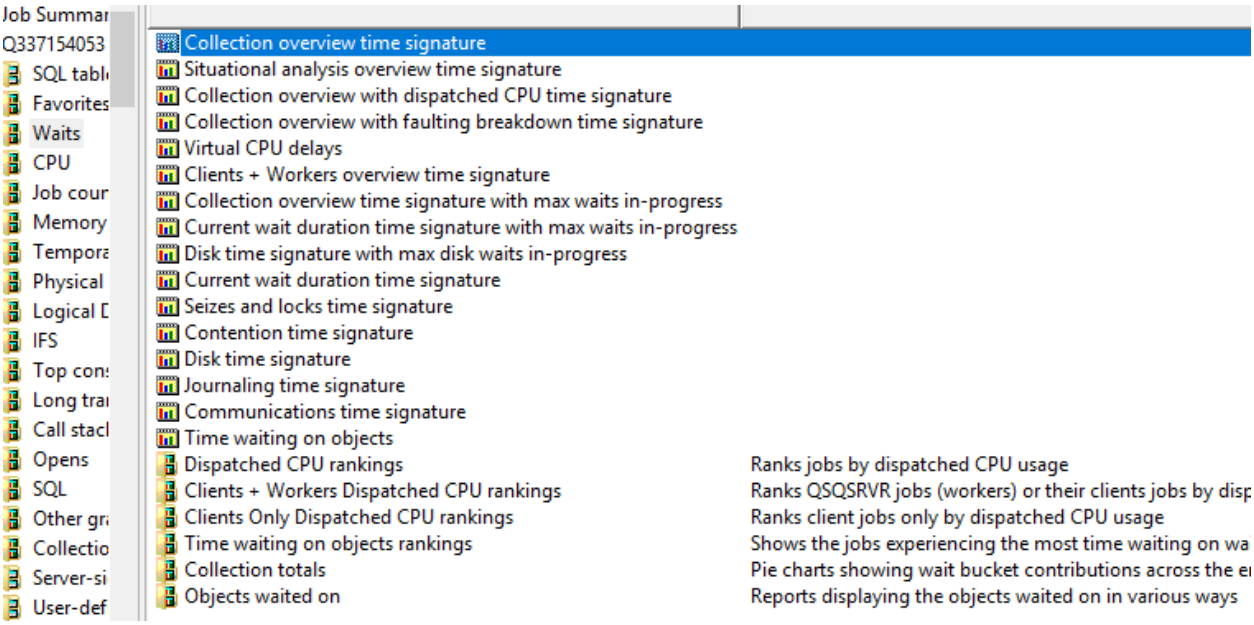
Favorites folder

Graph name	Notes
Collection overview time signature	
Dispatched CPU rankings by generic job	Requires Collection summary
Dispatched CPU rankings by generic job, current user	Requires Collection summary
Job counts	Requires Collection summary
Pages allocated/deallocated - overlapping bars	Requires Collection summary
Read and writes rates	
Physical I/O activity rates	
Logical DB activity rates	
Job temporary storage allocations	Requires Collection summary
Synchronous response	Requires Collection summary
SQL statements executed	7.2+
SQL logical database I/O rates	7.2+
Full opens rates	7.2+
5250 transaction response times	

8.14 Waits

These graphs show running and waiting time across all jobs in the collection over time. These graphs are wait bucket graphs which divides up the wait times into various buckets. These buckets contains enums which are the individual types of common waits for each bucket. The wait buckets and enums are visible from the Wait Buckets tab of the Collection's Properties.

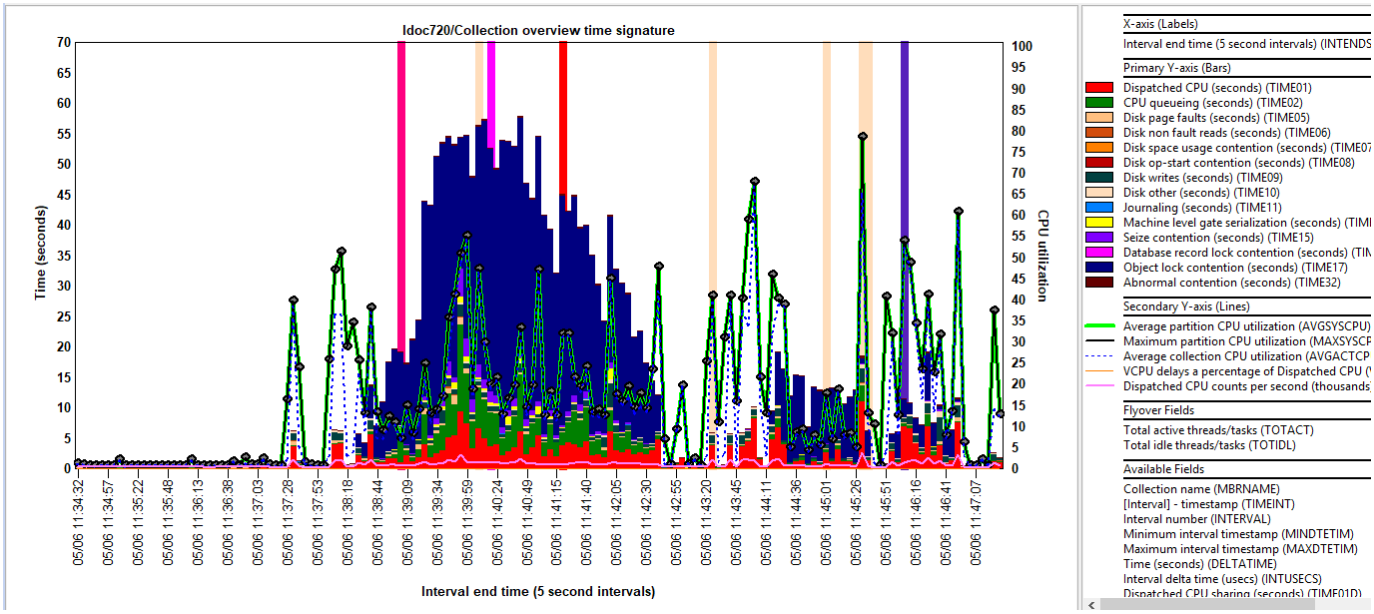
The folder contains many types of graphs related to the wait buckets including job ranking graphs and graphs over the wait object information captured by Job Watcher.



Waits folder

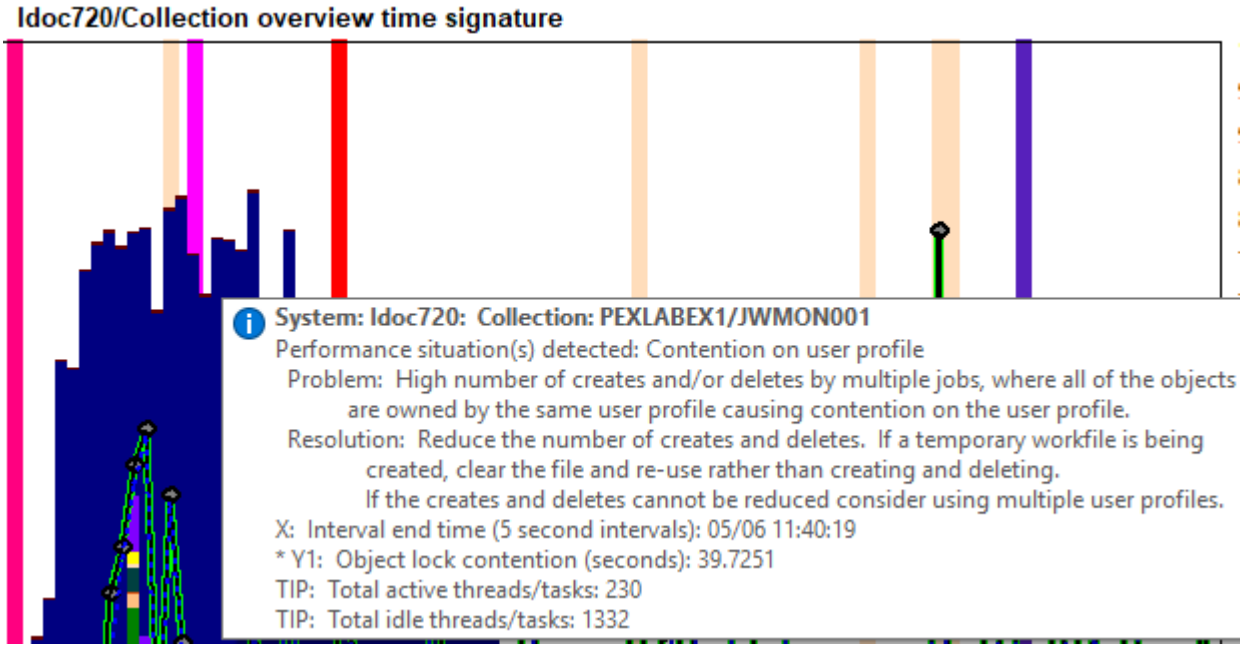
8.14.1 Collection overview time signature

This graph shows CPU time and the [“interesting” wait bucket](#) times added together across all jobs on the system. In the example above, a user could right-click intervals where the blue object lock time is showing and drill down in order to view the jobs which experienced the highest amount of object lock contention time for the selected time period.



Collection Overview Time Signature

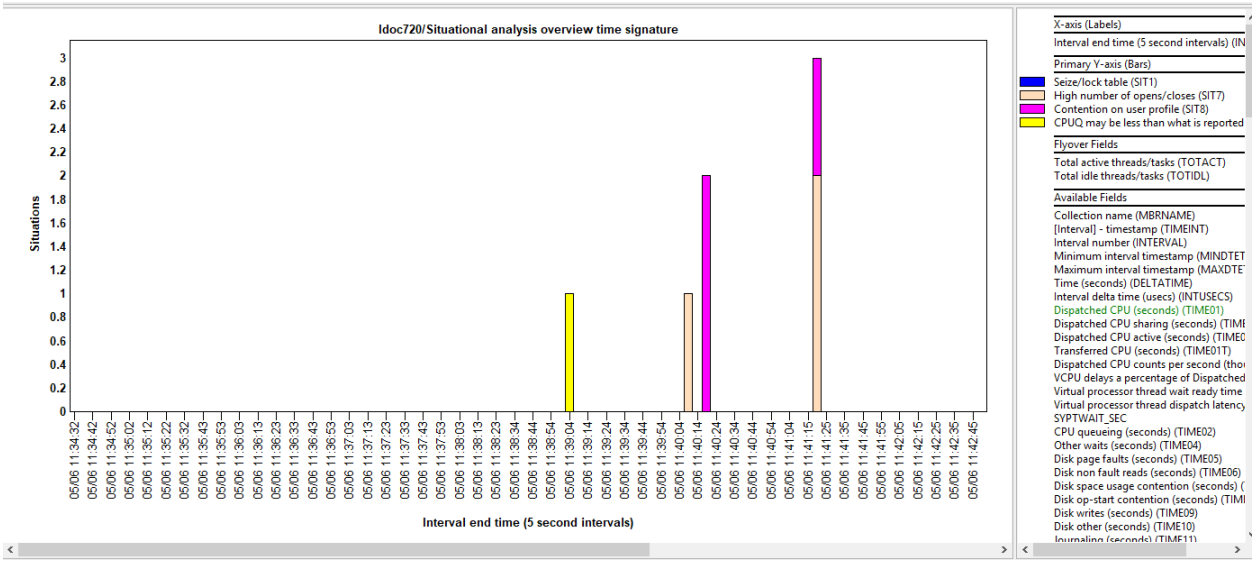
This graph is also displaying situations in the background for several intervals. Placing the mouse over the interval's data will provide more information. **Note:** Situations will only appear if the Collection Summary analysis and Situational Analysis have been ran first.



Situation details

8.14.2 Situational analysis overview time signature

This graph shows the number of times each type of situation occurred in the collection over time. Both the Collection summary analysis and the Situational analysis must be ran in order for this graph to appear.



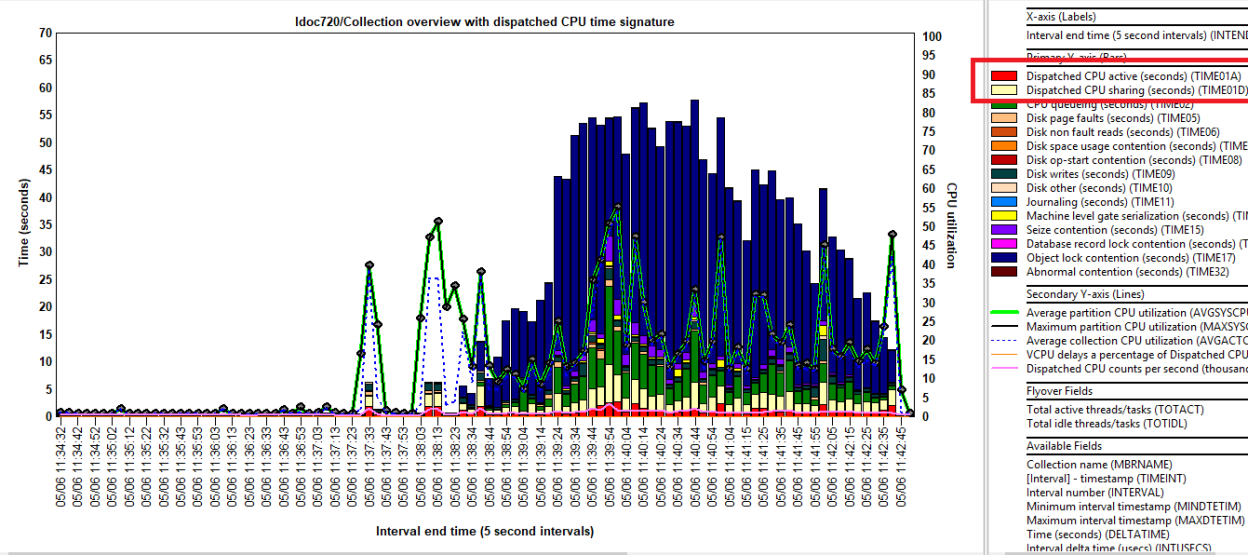
Situational Analysis Overview Time Signature

8.14.3 Collection Overview with dispatched CPU Time Signature

This graph is identical to the Collection overview time signature graph except the CPU time is divided into 2 different buckets:

Dispatched CPU active (red) – This is time spent burning CPU.

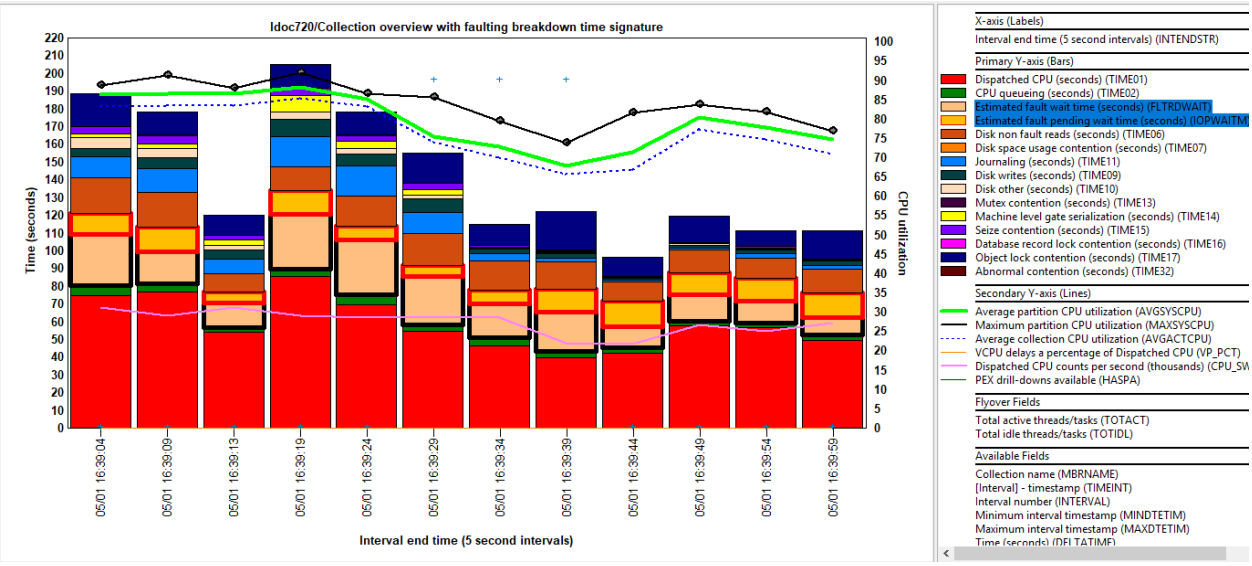
Dispatched CPU sharing (light yellow) – This is a type of time we can measure where we are dispatched to the processor but NOT actually burning CPU. It is being shared with other tasks. This is perfectly normal to have a large amount of time spent in this bucket and does not likely indicated a performance problem. It is provided here for advanced users.



Collection Overview with dispatched CPU Time Signature

8.14.4 Collection Overview with faulting breakdown Time Signature

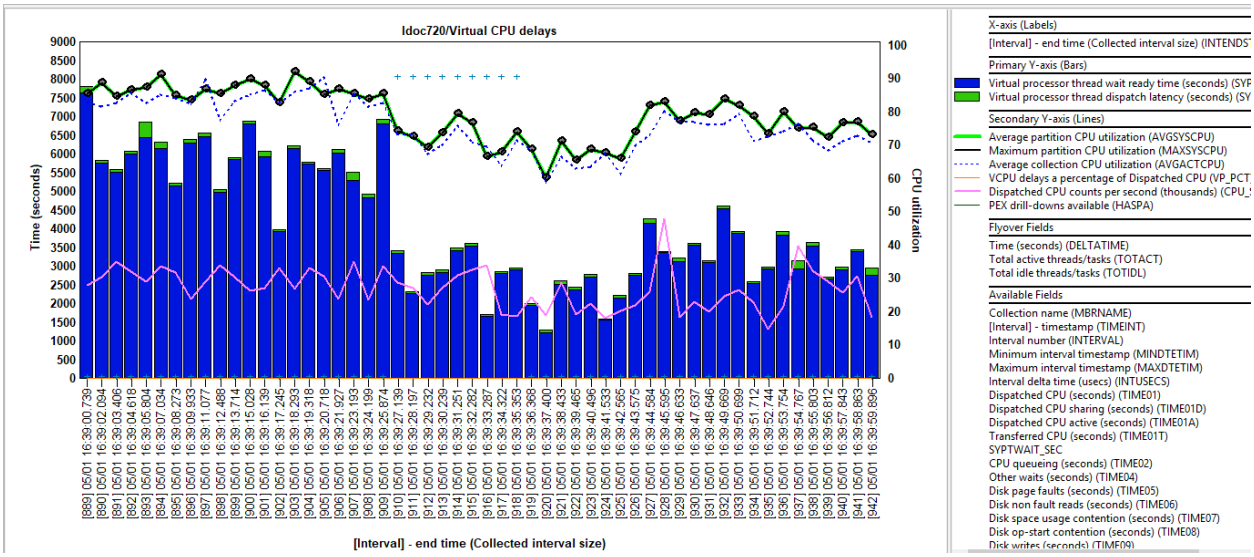
This graph is identical to the Collection Overview Time Signature except the disk page faults time is divided into 2 different buckets in order to distinguish between page fault waits vs IO pending page faults.



Collection Overview with faulting breakdown Time Signature

8.14.5 Virtual CPU delays

This graph provides a summarized look at the virtual processor thread wait ready and thread dispatch latency times across the entire LPAR.

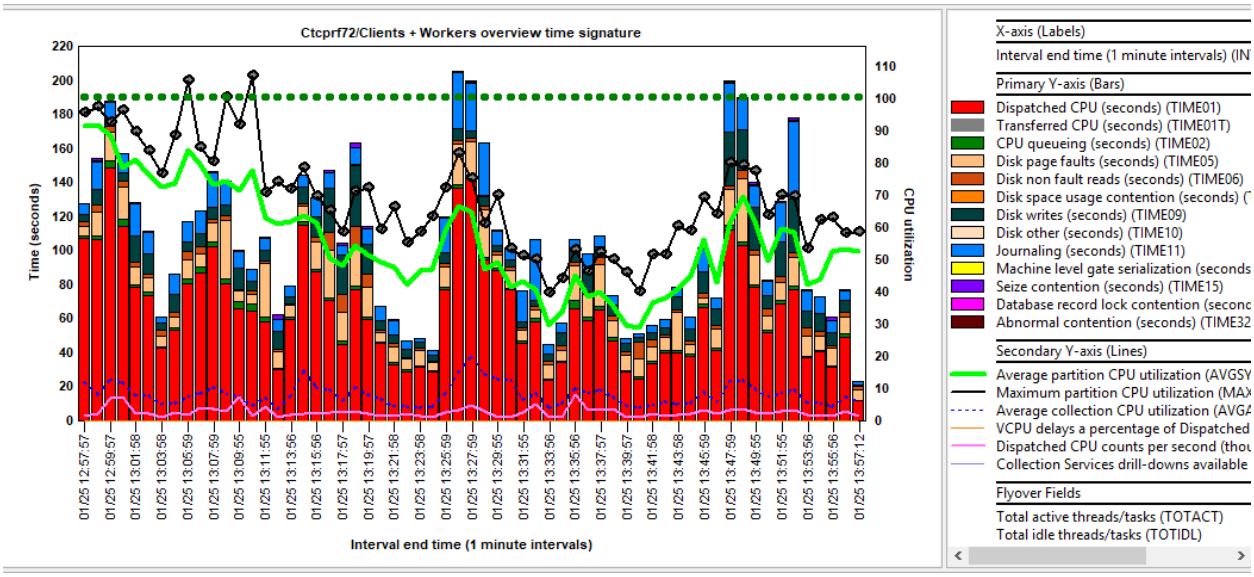


Virtual CPU delays

8.14.6 Clients + Workers overview time signature

This graph shows only the contributions of QSQSRVR jobs and the client jobs that initiated them. This graph will only appear if the Collection Summary analysis has been executed.

To drill-down and see only these same contributions use the Rankings -> Waits -> Clients + Workers rankings or Rankings -> Waits -> Clients Only Rankings menu options.

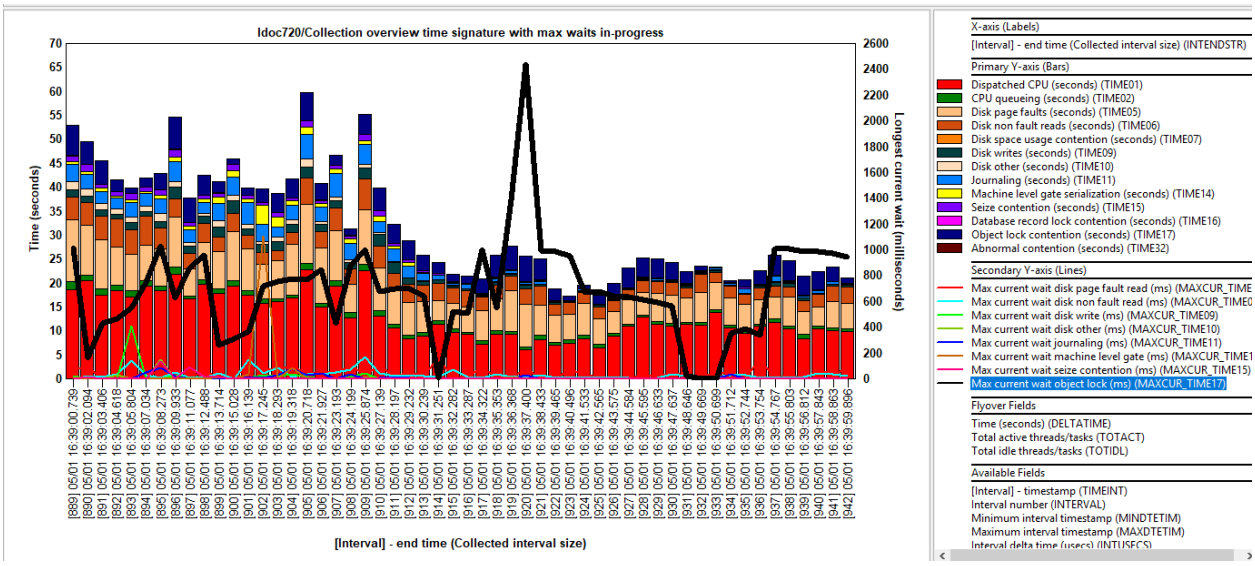


Clients + Workers overview time signature

8.14.7 Collection overview time signature with max waits in-progress

This graph is the same as the Collection Overview Time Signature except the longest waits that occurred in any job are shown on the 2nd Y-axis for any of the “interesting” types of waits. These longest waits are captured from the current wait duration field (CURRWTDUR) in the QAPYJWTDE file.

If the values are increasing over multiple intervals this can indicate a performance problem.



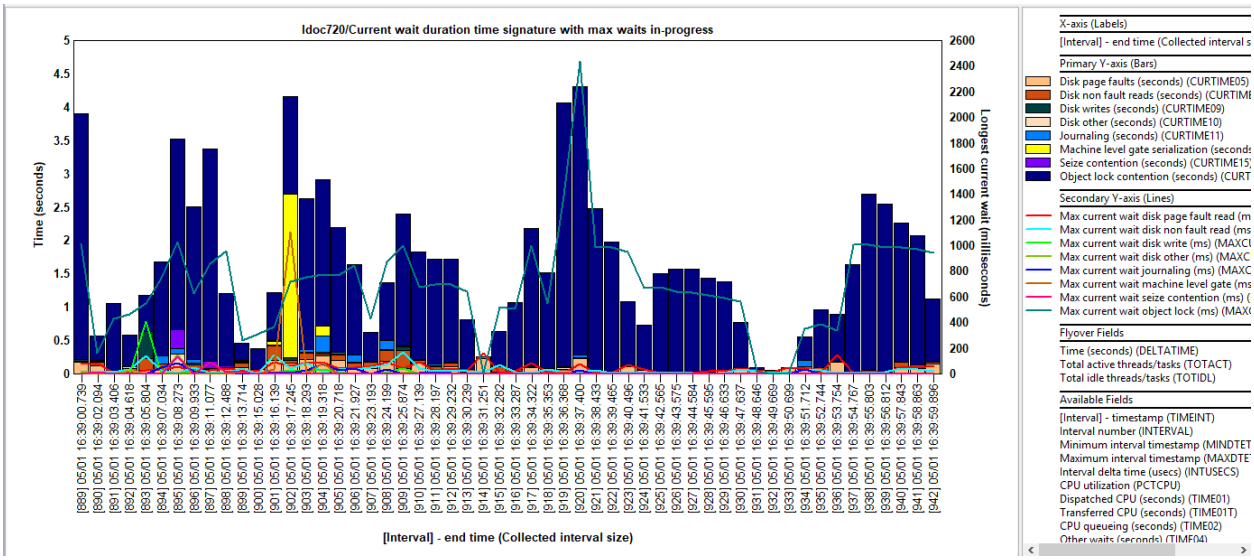
Collection overview time signature with max waits in-progress

8.14.8 Current wait duration time signature with max waits in-progress

This graph is unlike the others in this folder since it only shows the wait times for the “current wait” added up for all jobs.

The “current wait” is the wait time that occurs at the end of every snapshot interval for every job. By adding these waits together and only showing the “interesting wait buckets” across all jobs we may begin to see patterns or situations of interest that would not be otherwise readily apparent.

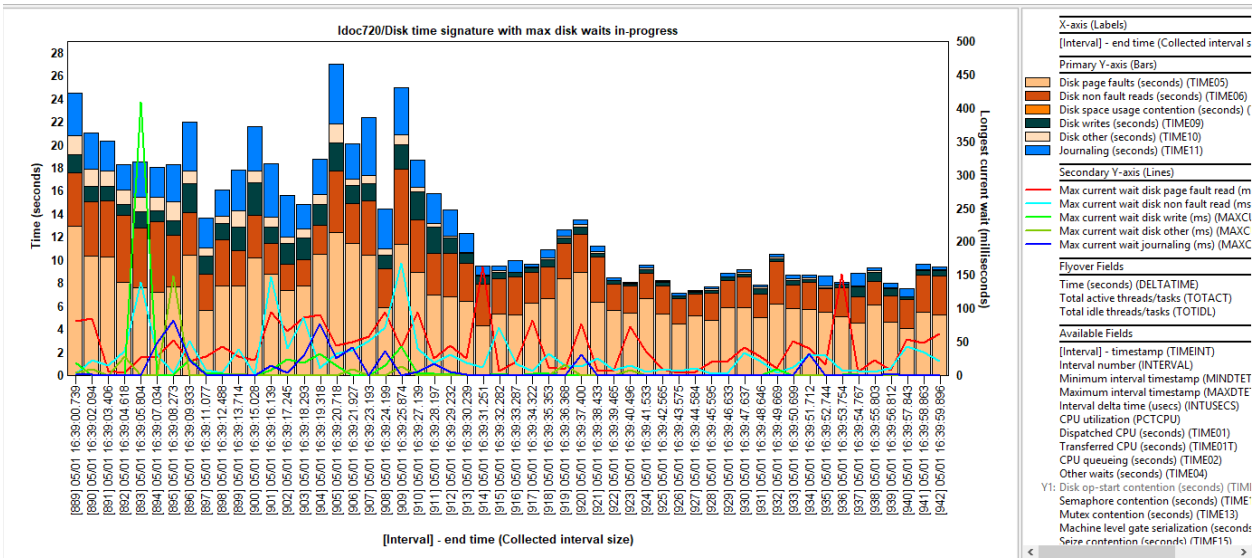
The 2nd Y-axis on this graph shows the longest single job current wait duration instead.



Current wait duration time signature

8.14.9 Disk time signature with max disk waits in-progress

This graph is like the Collection overview time signature with max waits in-progress except it only shows fields related to disk times and journal times. (wait buckets 5 – 11.)

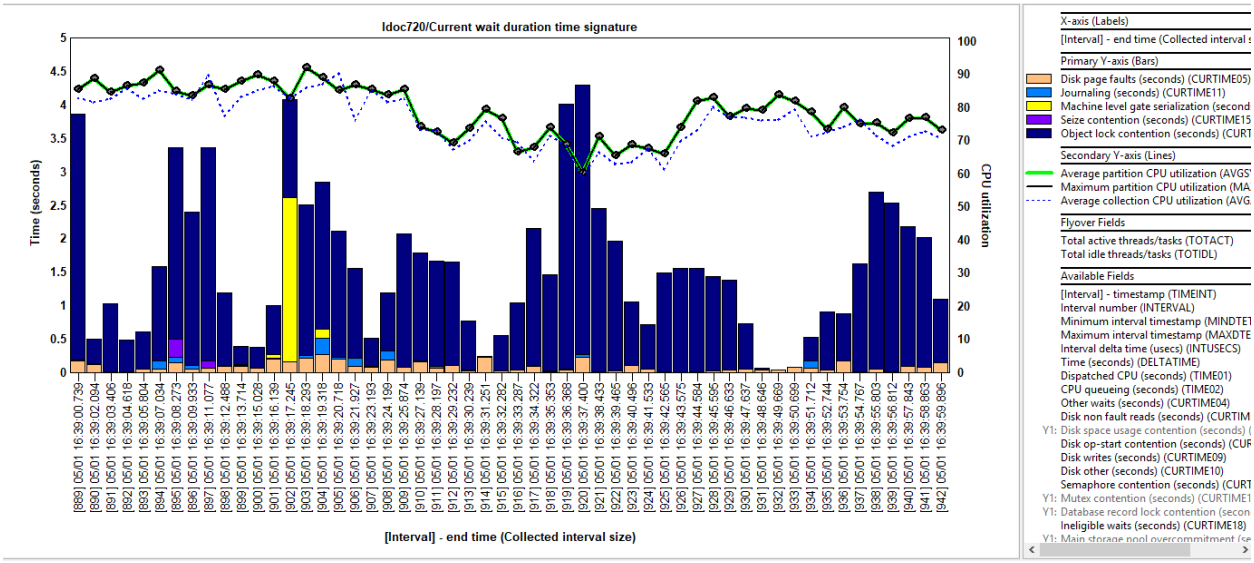


8.14.10 Current wait duration time signature

This graph is unlike the others in this folder since it only shows the wait times for the “current wait” added up for all jobs.

The “current wait” is the wait time that occurs at the end of every snapshot interval for every job. By adding these waits together and only showing the “interesting wait buckets” across all jobs we may begin to see patterns or situations of interest that would not be otherwise readily apparent.

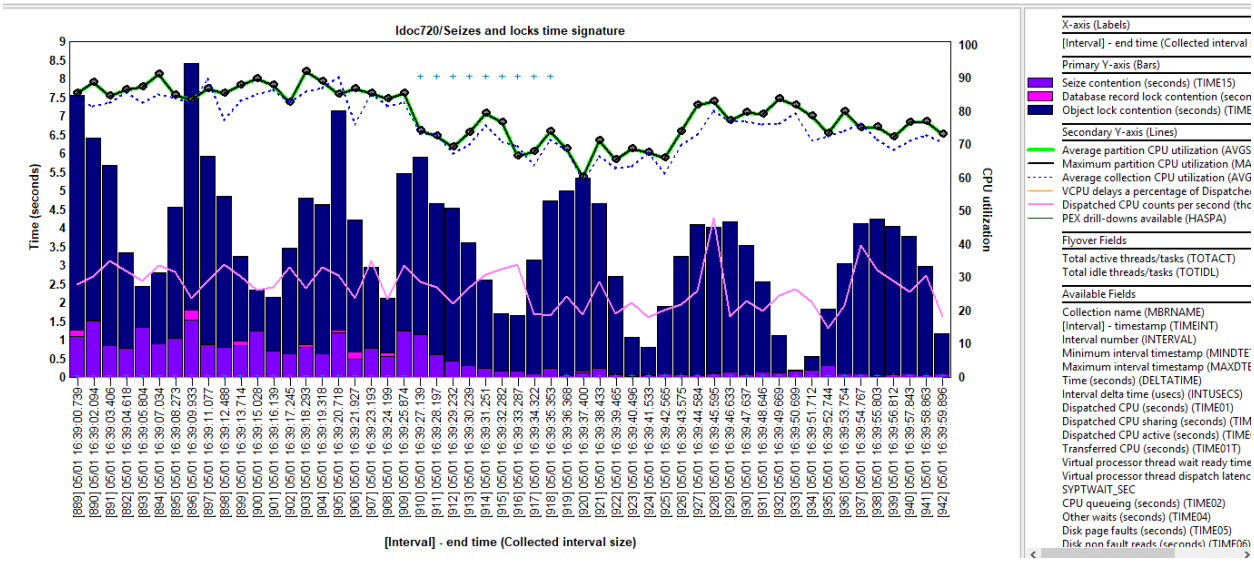
The 2nd Y-axis on this graph shows the CPU utilization.



Current wait duration time signature

8.14.11 Seizes and locks time signature

This graph shows only the seizes, record locks and object lock times as well as CPU utilization on the Y2-axis.



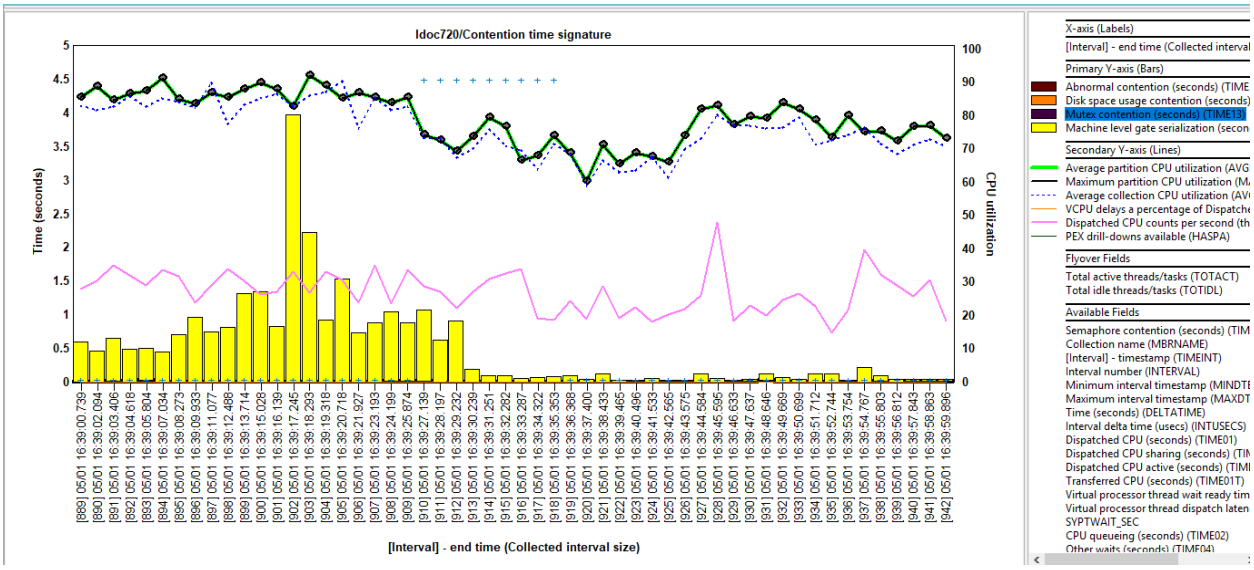
Seizes and locks time signature

8.14.12 Contention time signature

This graph only shows several wait buckets that are often associated with some type of contention on the system. It shows the following wait buckets:

1) Abnormal contention (32)
2) Disk space usage contention (7)
3) Disk op-start contention (8)
4) Mutex contention (13)

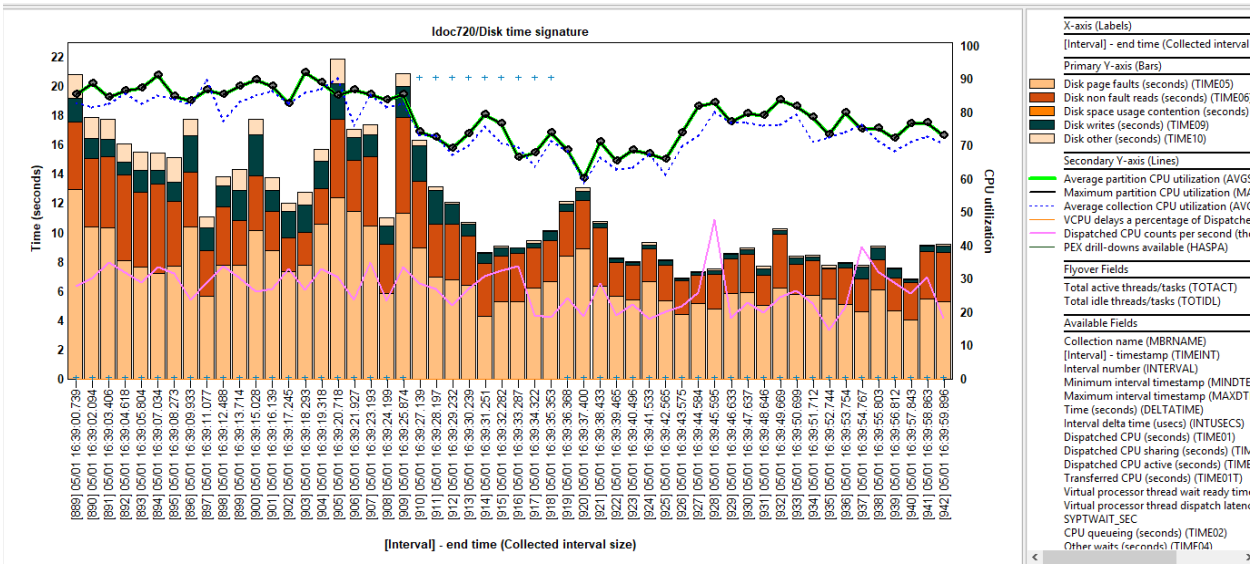
5) Machine level gate serialization (14)



Contention time signature

8.14.13 Disk time signature

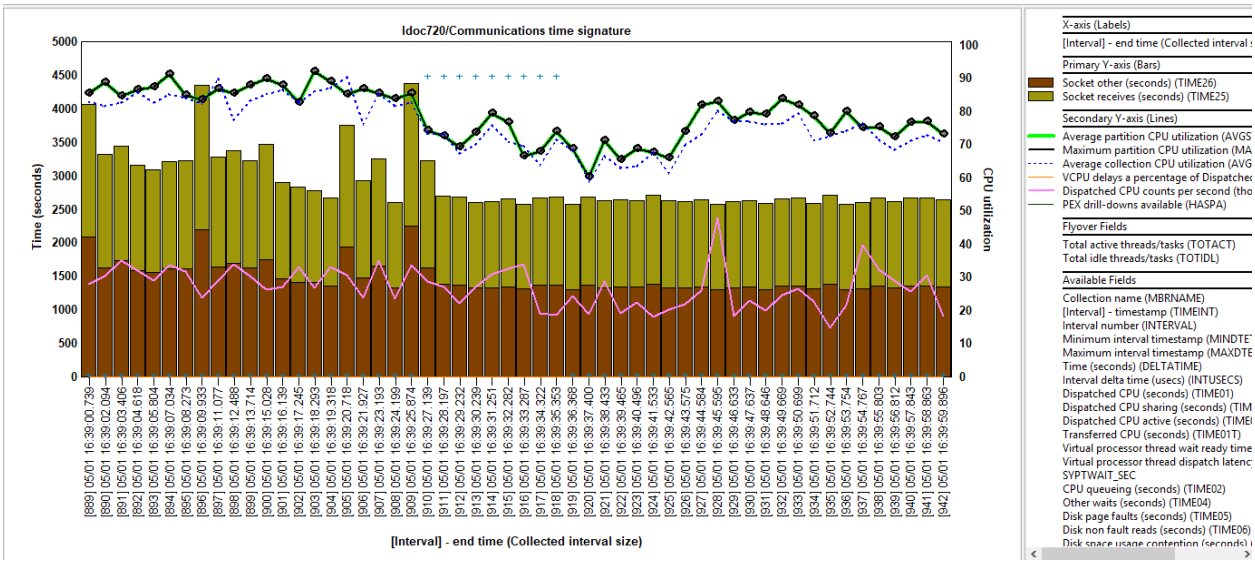
This graph shows only wait buckets associated disk related times.



Disk time signature

8.14.14 Communications time signature

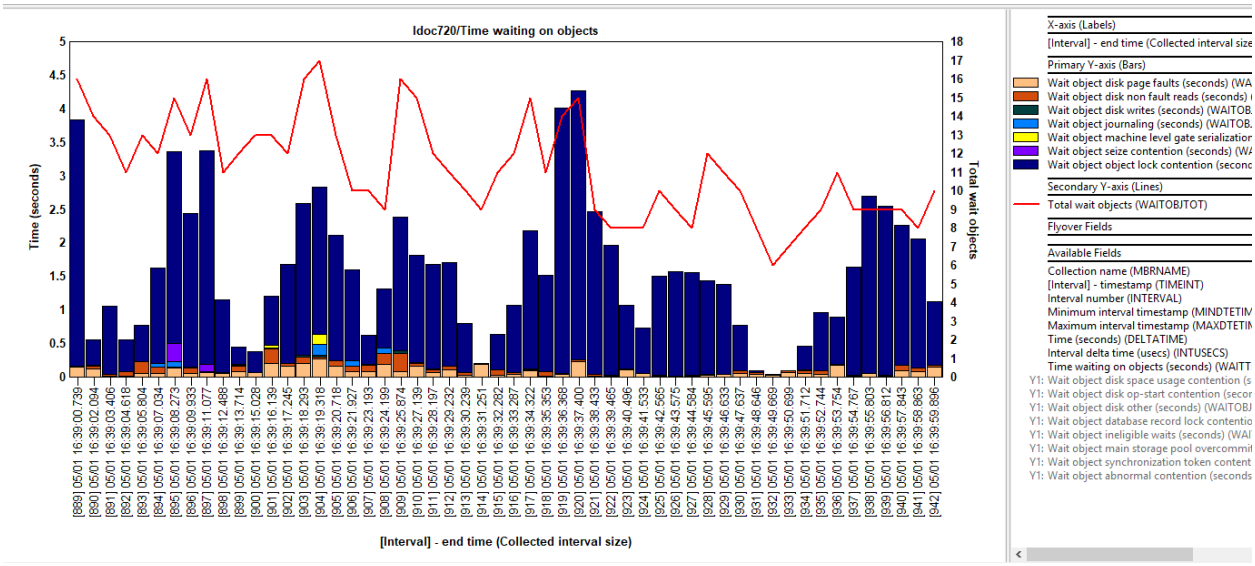
These waits indicate time waiting to receive or send data or other types of socket waits. An example of a socket receive is what a QZRCSRVS job (iDoctor remote command job servicing the GUI) will do when it is idle waiting for requests from the PC. Once the job receives data over the comm line other types of ***non-idle*** waits (CPU, disk IO) will be shown.



Communications time signature

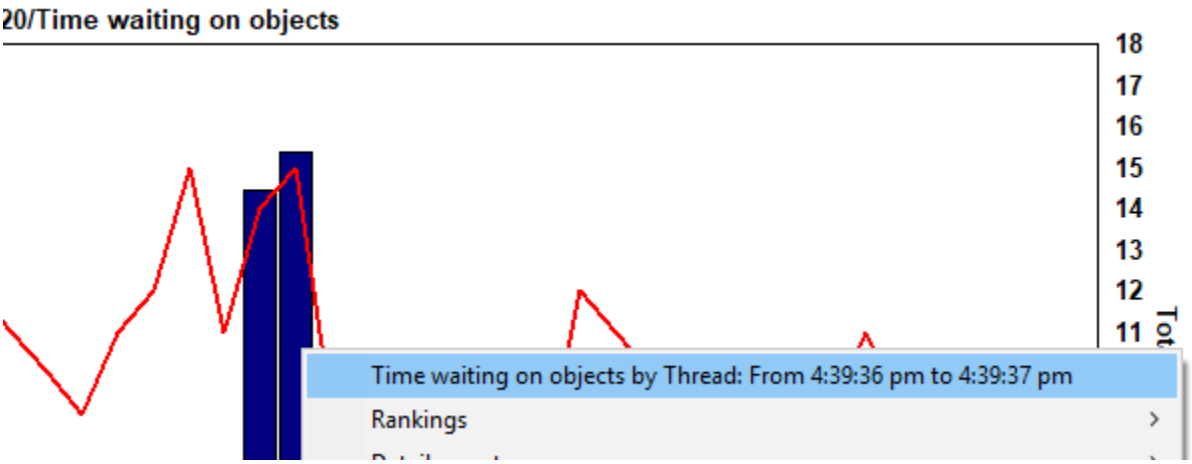
8.14.15 Time waiting on objects

This graph adds up the current wait duration times only where a wait object was associated with each “interesting” wait bucket times and graphs the results. The y2-axis displays the total wait objects found per interval.



Time waiting on objects

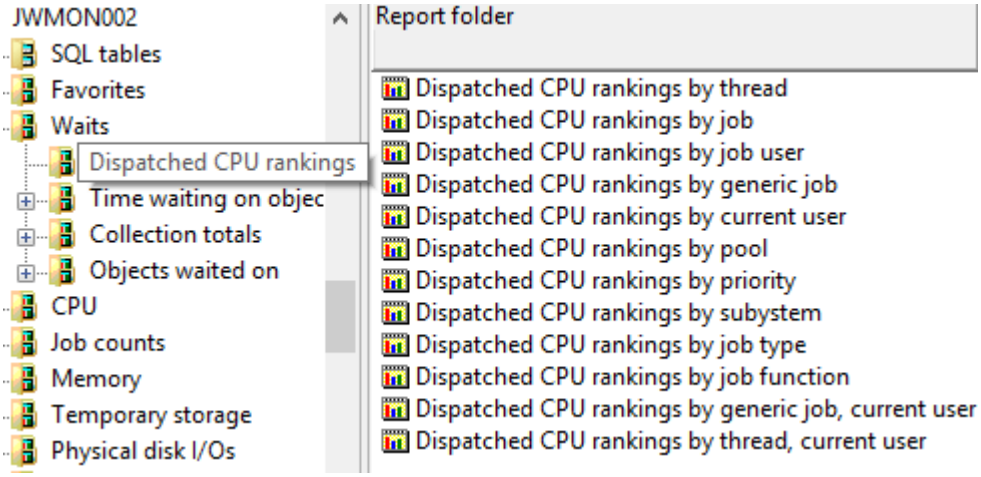
You can drill down from here to see the jobs and wait objects detected for the desired time period.



Time waiting on objects by thread drill-down option

8.14.16 Dispatched CPU rankings

This folder contains the set of ranking graphs that rank job data by Dispatched CPU time. The jobs with the most Dispatched CPU time will be listed first.

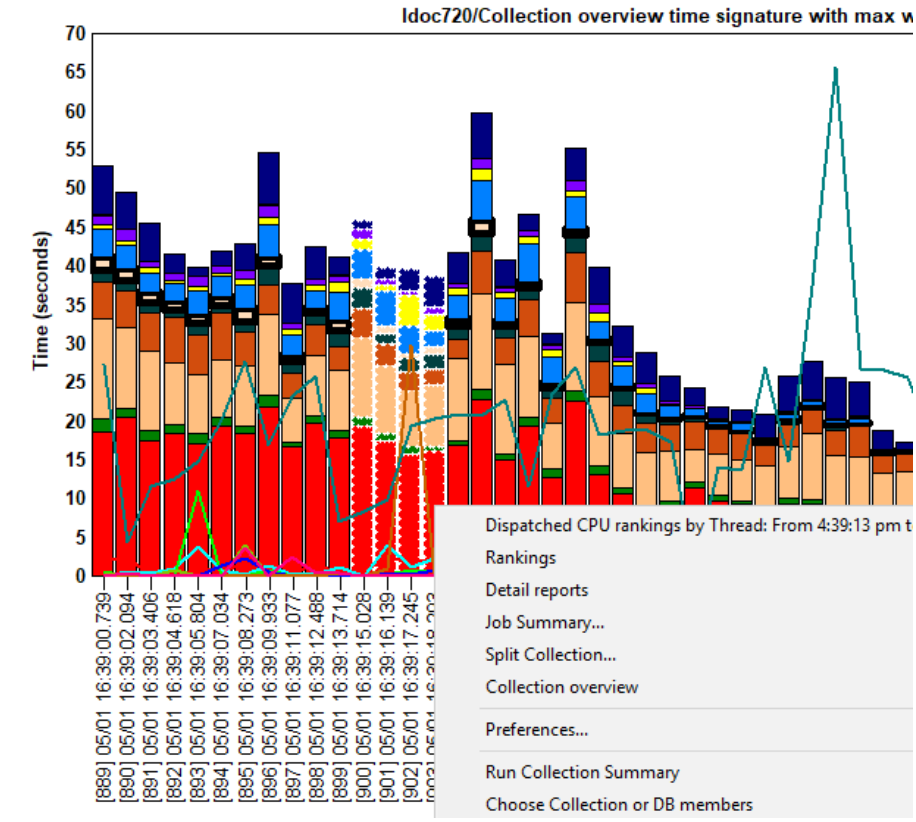


Waits -> Dispatched CPU Rankings

Note: Only the Dispatched CPU rankings by thread will exist in this folder unless the Collection summary analysis has been ran.

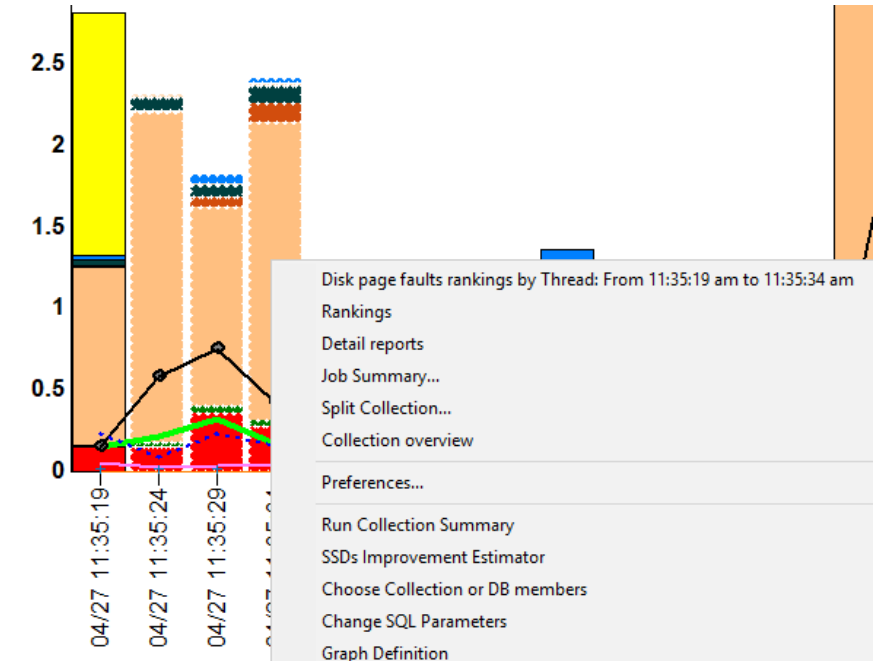
These graphs when accessed under the collection will perform the calculation against the entire collection. But these same graphs can also be used as a drill-down from the Overview graph by using the Rankings -> Waits -> Dispatched CPU rankings option or simply use the first option in the menu which would typically be Dispatched CPU rankings by thread.

From these ranking graphs you can right-click the desired job and graph its wait buckets over time.



Dispatched CPU rankings by thread drill-down menu

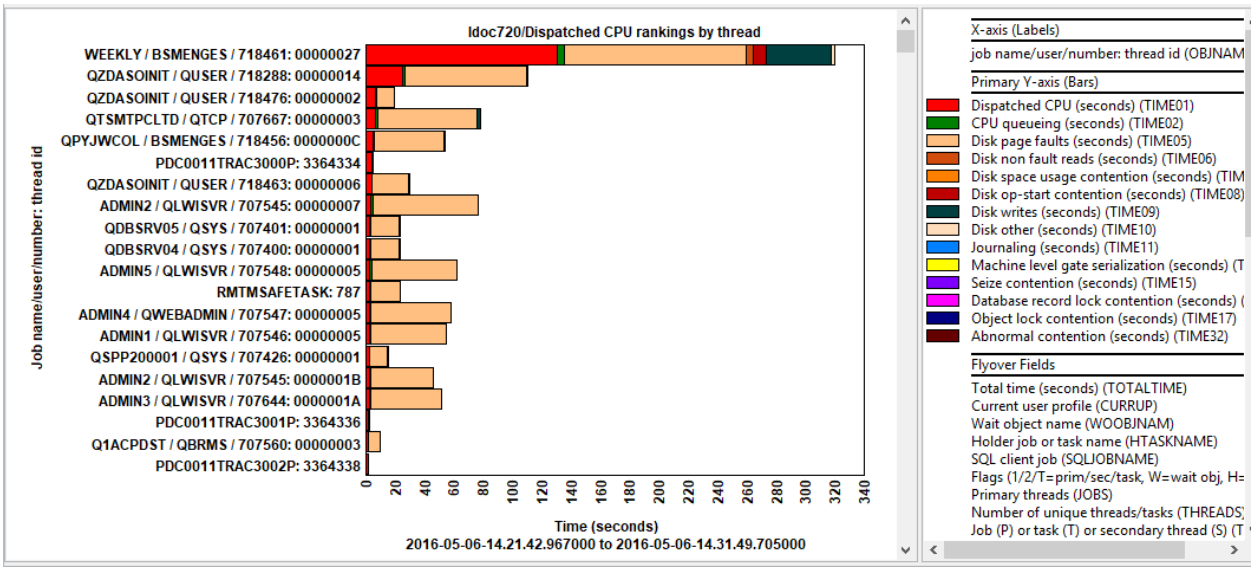
Note: By right-clicking on a different wait bucket you can change the graph you end up with to be ranked by the bucket right-clicked on.



Disk page faults rankings by thread drill-down menu

8.14.16.1 Dispatched CPU rankings by thread

This graph ranks the selected time period's wait bucket data by Dispatched CPU and thread or task.

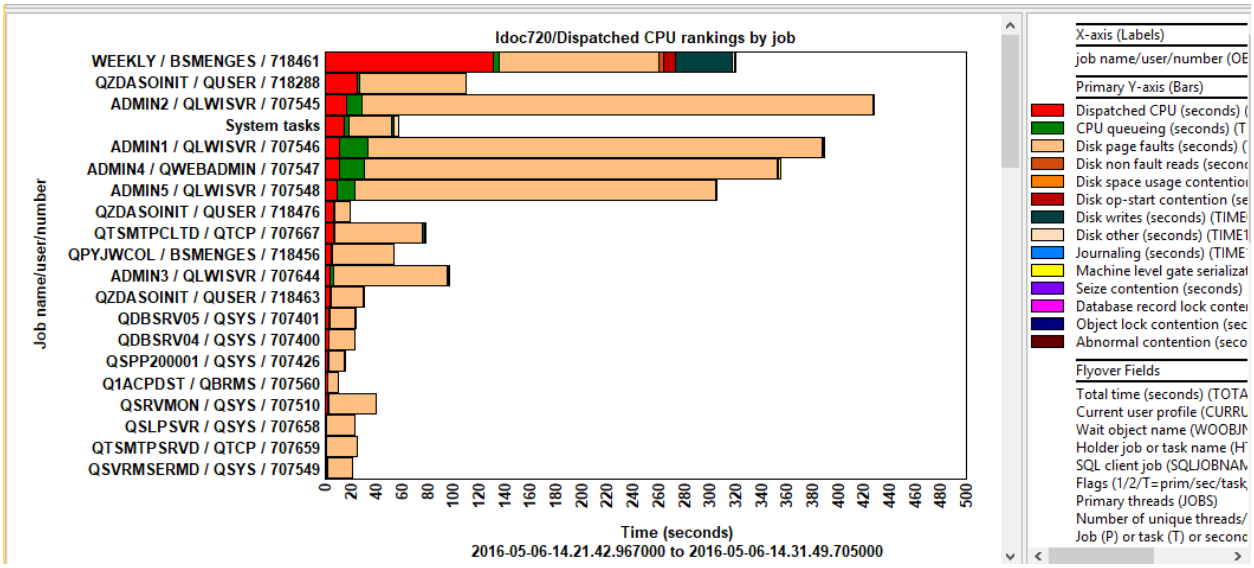


Dispatched CPU Rankings by thread

8.14.16.2 Dispatched CPU rankings by job

This graph ranks the selected time period's wait bucket data by Dispatched CPU and primary thread. Jobs that have multiple threads are all added up together.

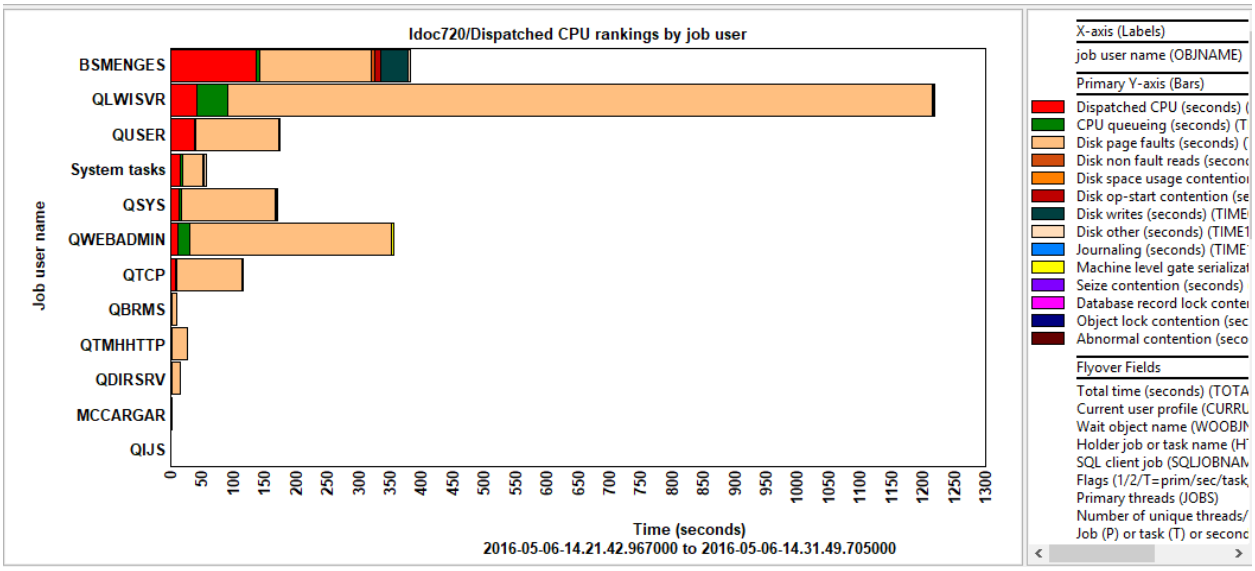
Note: All system tasks are grouped together into one "**System tasks"** record within this report.



8.14.16.3 Dispatched CPU rankings by job user

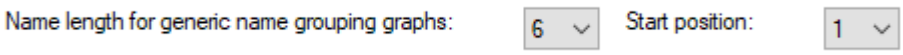
This graph ranks the selected time period's wait bucket data by Dispatched CPU and job user.

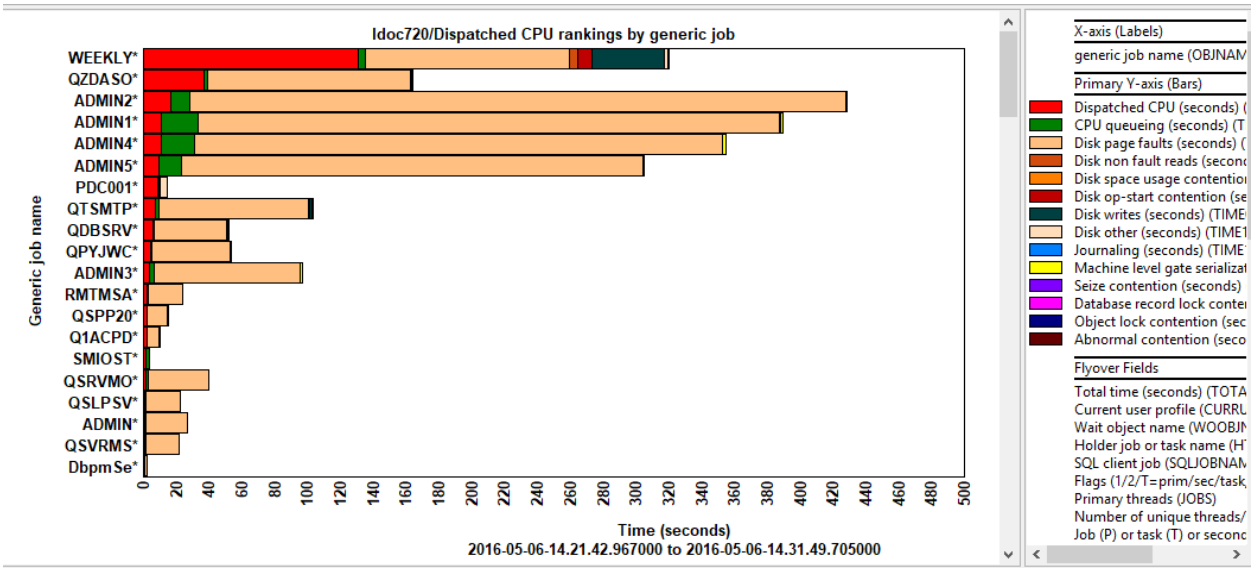
Note: All system tasks are grouped together into one "**System tasks"** record within this report.



8.14.16.4 Dispatched CPU rankings by generic job

This graph ranks the selected time period's wait bucket data by Dispatched CPU and generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

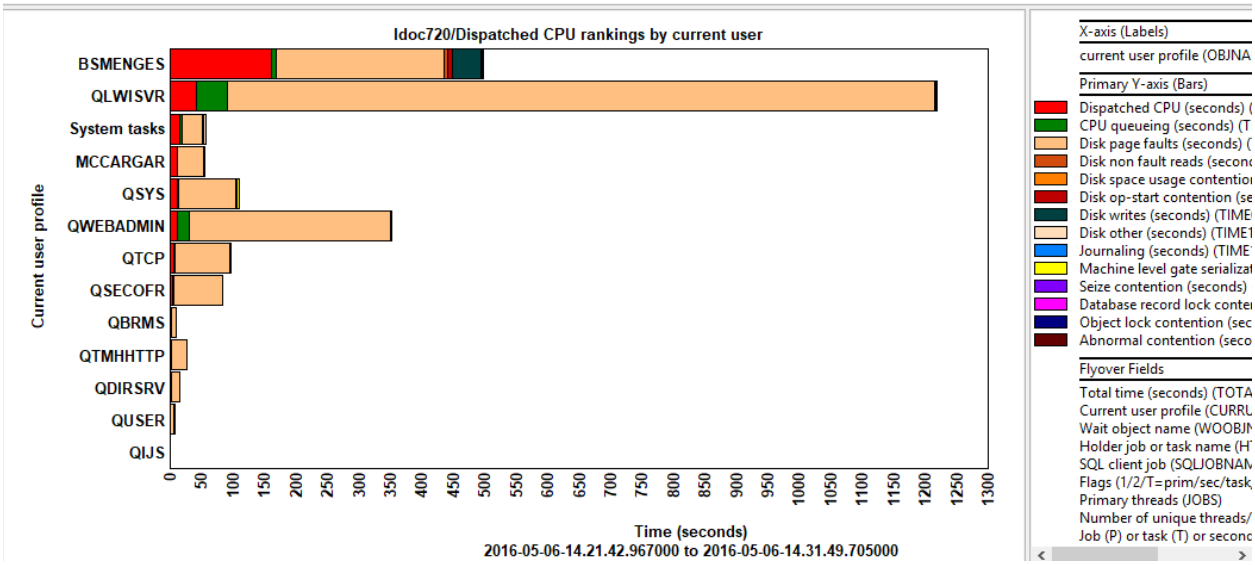




Dispatched CPU rankings by generic job

8.14.16.5 Dispatched CPU rankings by current user

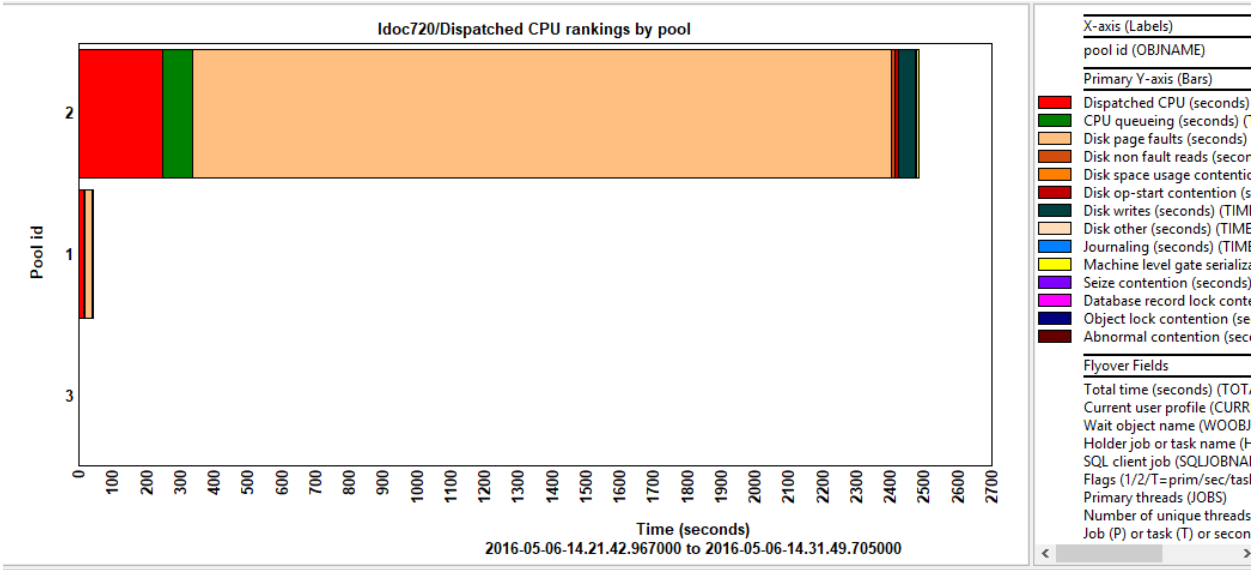
This graph ranks the selected time period's wait bucket data by Dispatched CPU and current user profile.
Note: All system tasks are grouped together into one "**System tasks"** record within this report.



Dispatched CPU rankings by current user

8.14.16.6 Dispatched CPU rankings by pool

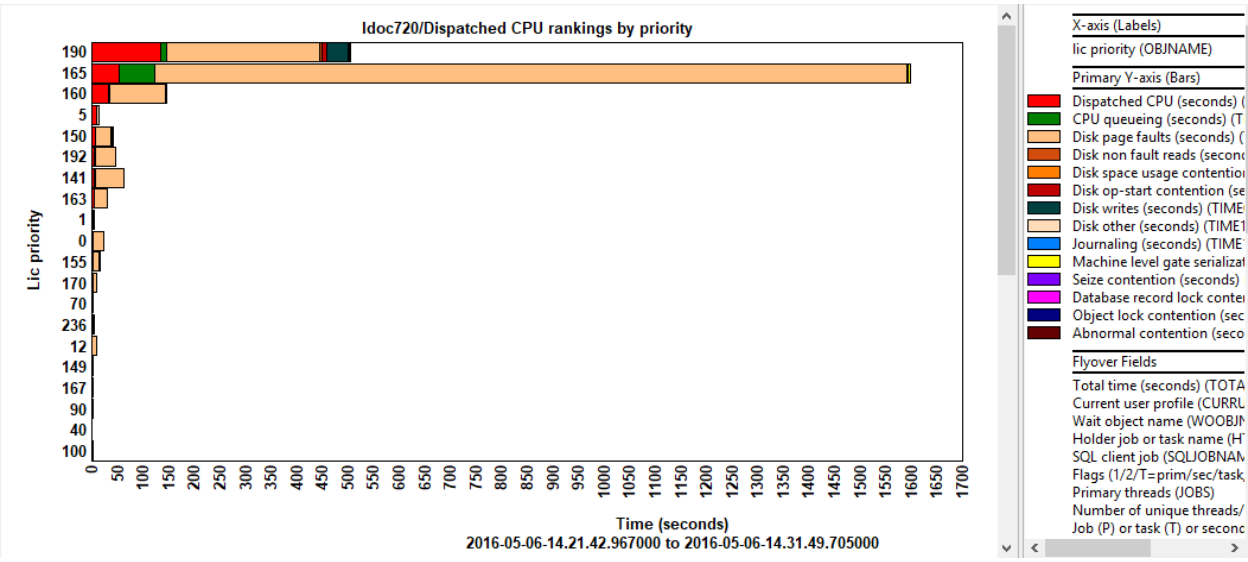
This graph ranks the selected time period's wait bucket data by Dispatched CPU and memory pool.



Dispatched CPU rankings by pool

8.14.16.7 Dispatched CPU rankings by priority

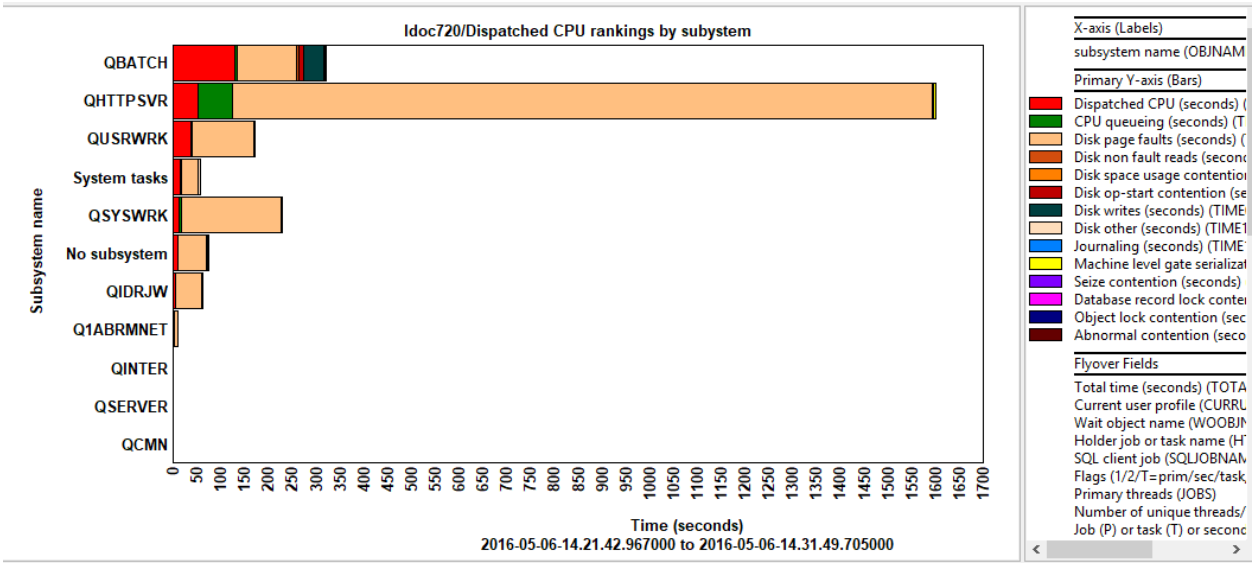
This graph ranks the selected time period's wait bucket data by Dispatched CPU and LIC priority. XPF priority can be calculated by subtracting 140 for those values shown exceeding 140.



Dispatched CPU rankings by LIC priority

8.14.16.8 Dispatched CPU rankings by subsystem

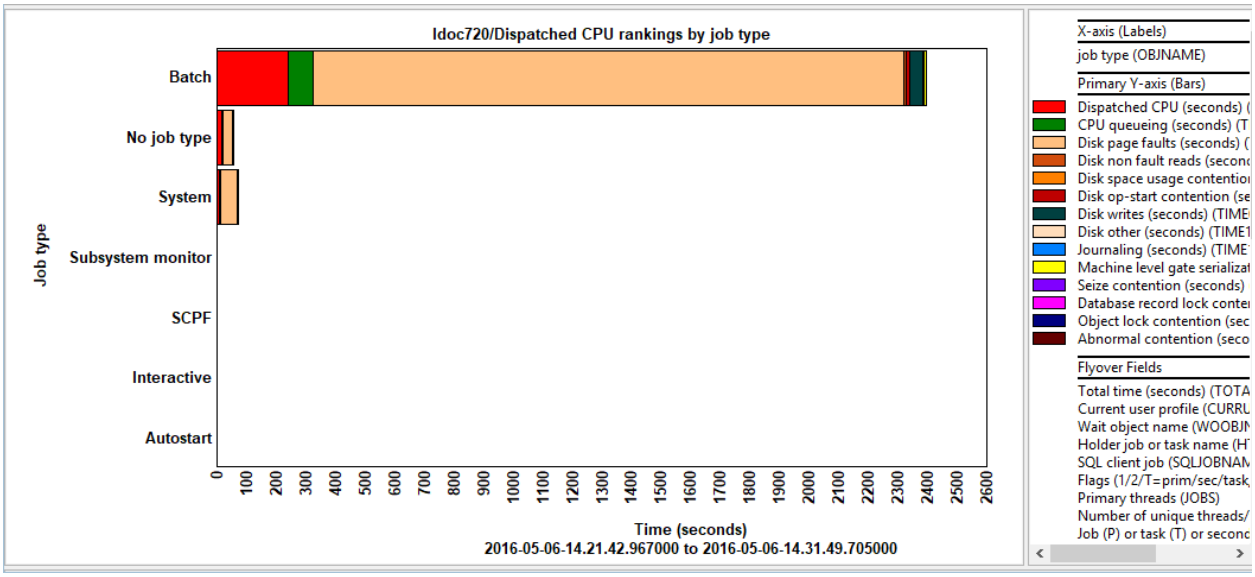
This graph ranks the selected time period's wait bucket data by Dispatched CPU and subsystem. Job times that had no subsystem listed are grouped into 1 bar called "No subsystem". **Note:** All system tasks are grouped together into one "**System tasks"** bar within this report.



Dispatched CPU rankings by subsystem

8.14.16.9 Dispatched CPU rankings by job type

This graph ranks the selected time period's wait bucket data by Dispatched CPU and job type (such as Batch, interactive, autostart, etc.) Job wait bucket times that had no job type associated with it are grouped into 1 bar called "No job type".

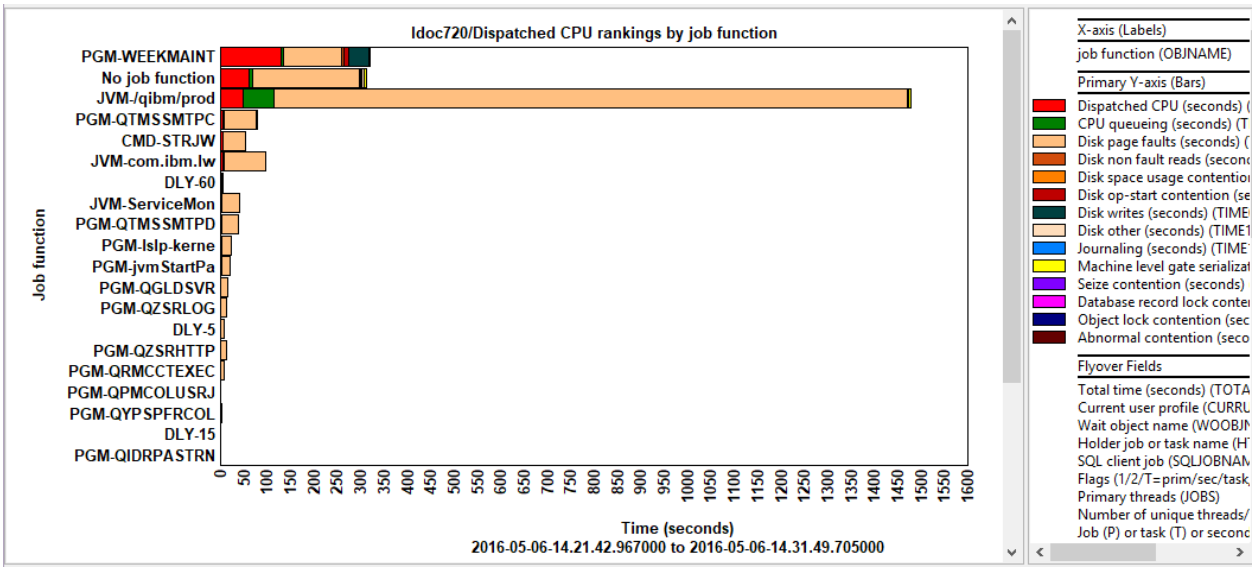


Dispatched CPU rankings by job type

8.14.16.10 Dispatched CPU rankings by job function

This graph ranks the selected time period's wait bucket data by Dispatched CPU and job function. Job function is the same as what you would see normally on the WRKACTJOB command's Function column. Any wait bucket times that had no job function associated with it are grouped into 1 bar called "No job function".

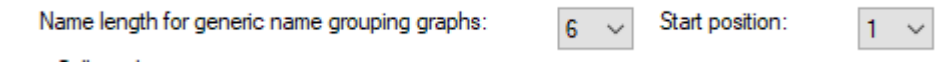
Note: This graph can be misleading in some situations because the current job function for each process is only captured at the end of each Job Watcher interval. The times given are not necessarily 100% from each function listed.



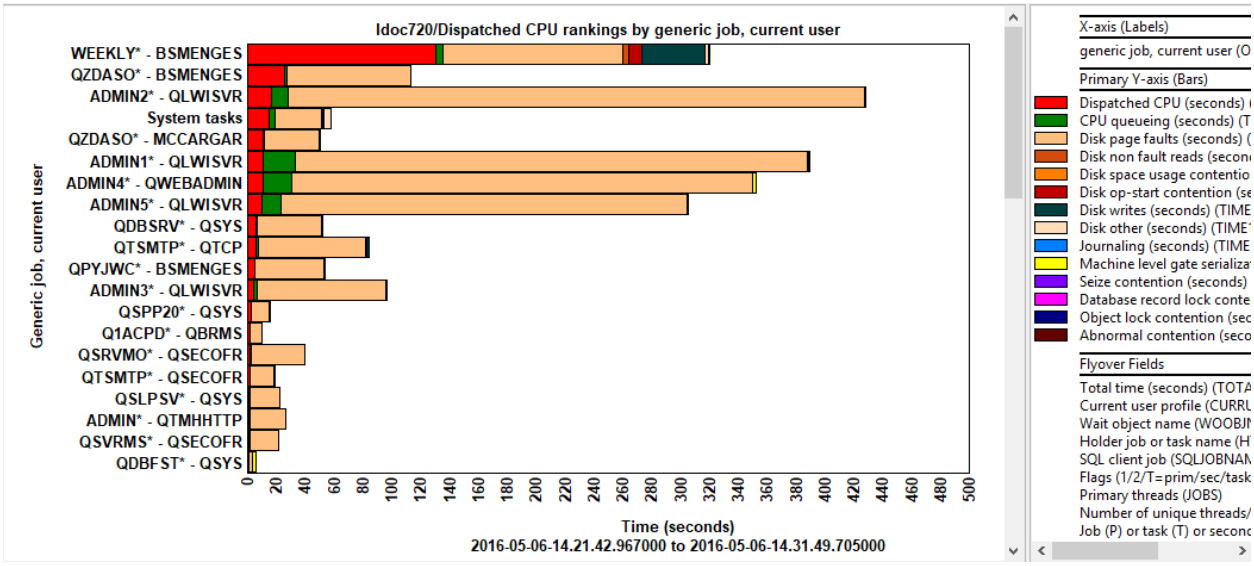
Dispatched CPU rankings by job function

8.14.16.11 Dispatched CPU rankings by generic job, current user

This graph ranks the selected time period's wait bucket data by Dispatched CPU and generic job/current user profile combination. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:



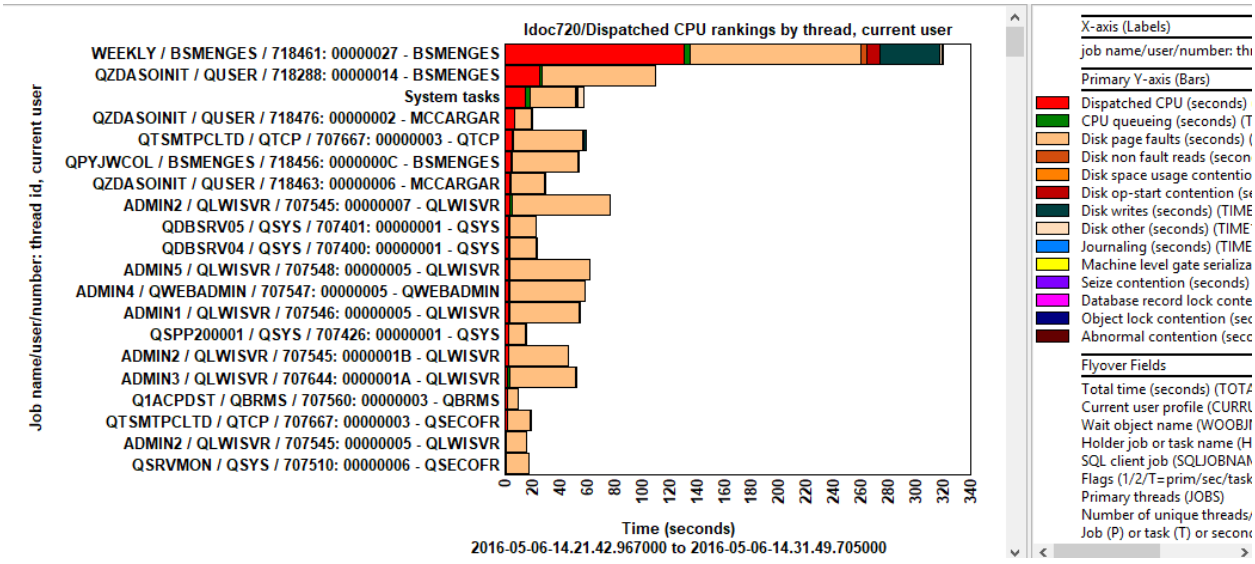
Note: All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



Dispatched CPU rankings by generic job, current user

8.14.16.12 Dispatched CPU rankings by thread, current user

This graph ranks the selected time period's wait bucket data by Dispatched CPU and thread/current user profile combination. **Note:** All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



8.14.17 Clients + Workers dispatched CPU rankings

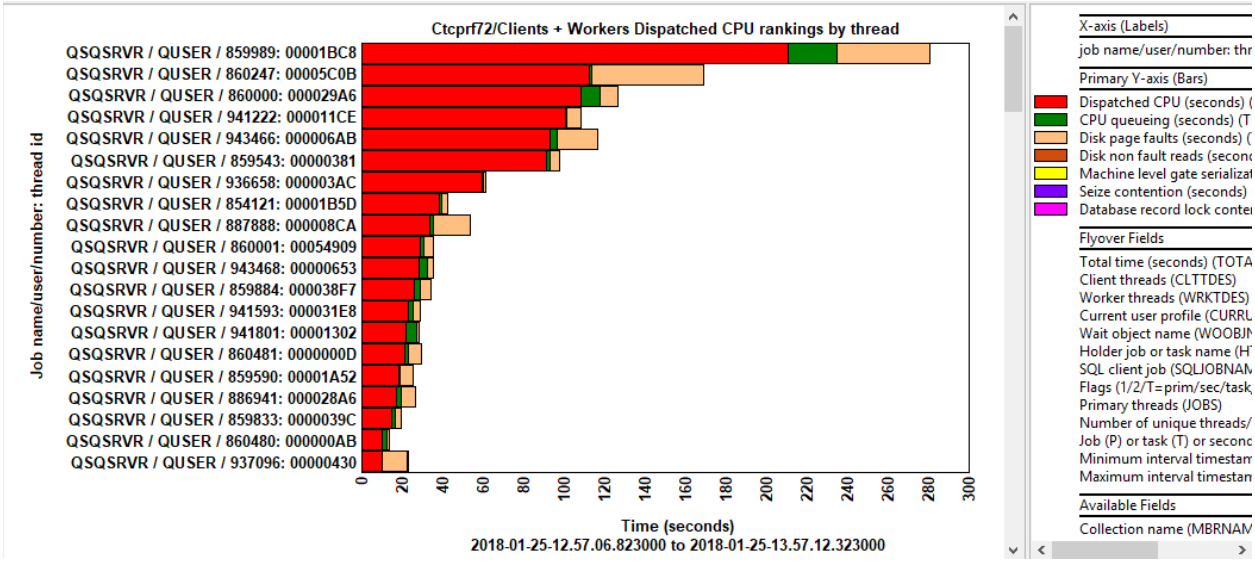
This folder provides the ranking graphs for QSQSRVR jobs and their client jobs that initiated them.

Note: These graphs require the Collection summary analysis to be ran in order to appear!

Clients + Workers dispatched CPU rankings

8.14.17.1 Clients + Workers dispatched CPU rankings by thread

This graph ranks the QSQSRVR jobs and their clients that initiated them by Dispatched CPU time and thread.



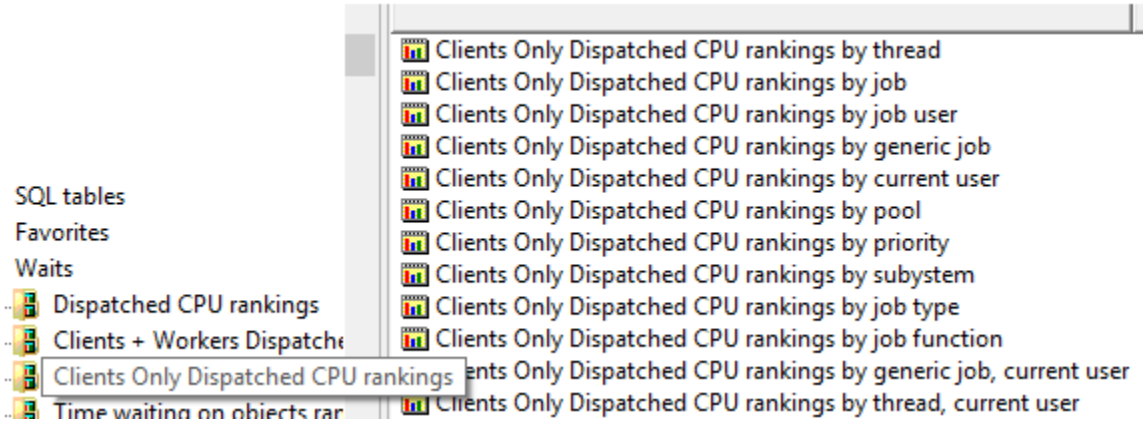
Clients + Workers dispatched CPU rankings by thread

Note: The rest of the ranking graphs of this type will follow the same format as those described previously in the **Waits -> Dispatched CPU rankings folder**.

8.14.18 Clients only dispatched CPU rankings

This folder provides the ranking graphs for the client jobs that initiated QSQSRVR jobs. The QSQSRVR jobs are not included in these reports.

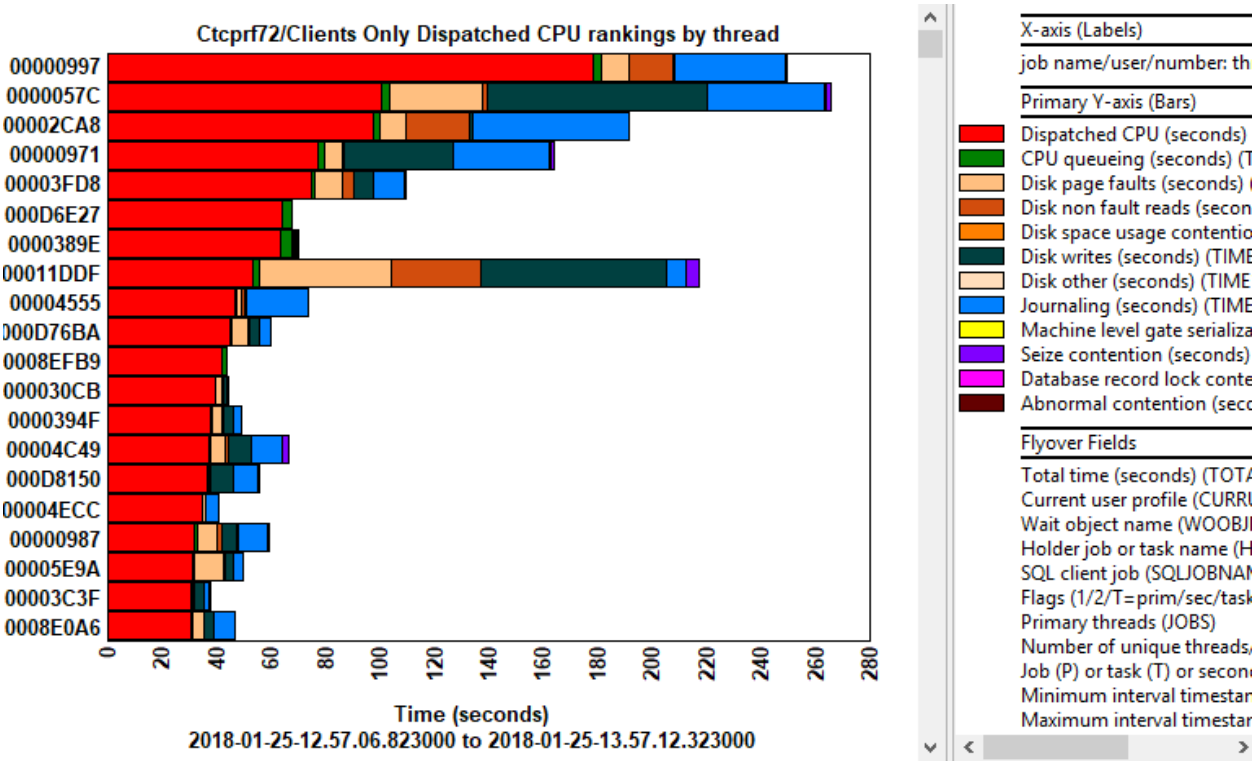
Note: These graphs require the Collection summary analysis to be ran in order to appear!



Clients only dispatched CPU rankings by thread

8.14.18.1 Clients Only dispatched CPU rankings by thread

This graph ranks the client jobs that initiated QSQSRVR jobs by Dispatched CPU time and thread.



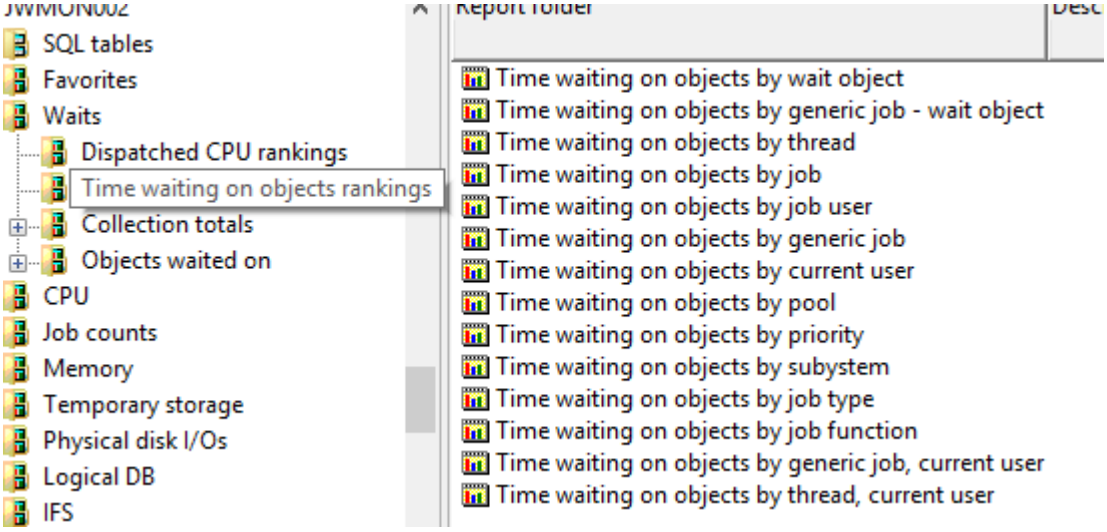
Clients only dispatched CPU rankings by thread

Note: The rest of the ranking graphs of this type will follow the same format as those described previously in the **Waits -> Dispatched CPU rankings folder**.

8.14.19 Time waiting on objects rankings

This folder contains the set of ranking graphs that rank current wait times for jobs experiencing interesting waits AND having a wait object associated with it. The jobs are ranked by total time in the interesting waits captured.

Note: These graphs require the Collection summary analysis to be ran in order to appear!



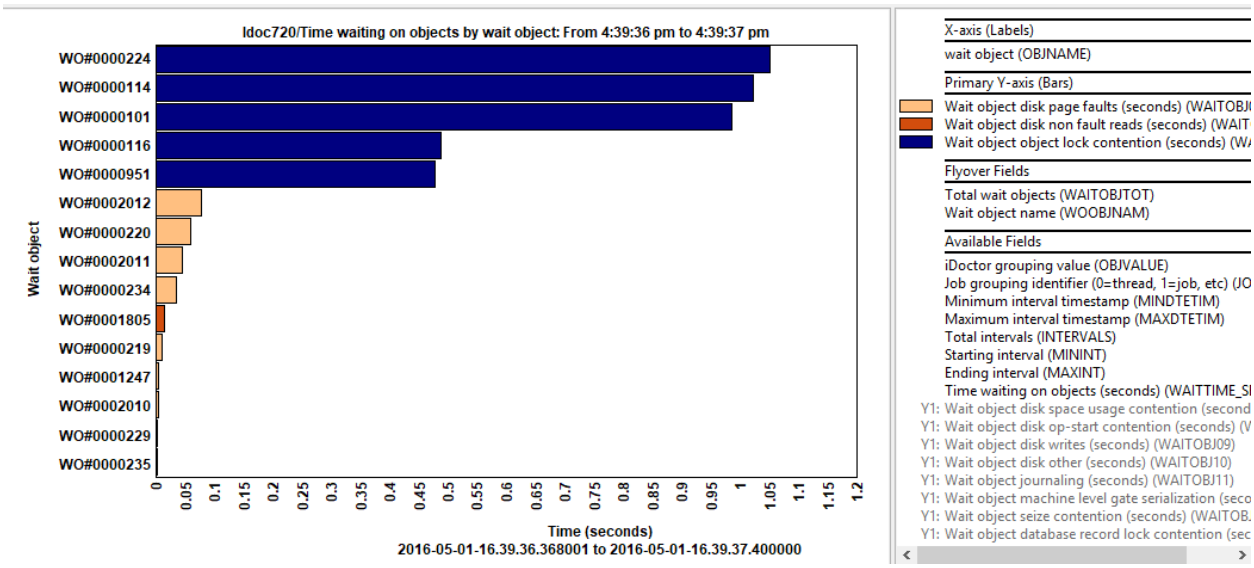
Waits -> Time waiting on objects rankings

These graphs when accessed under the collection will perform the calculation against the entire collection. But these same graphs can also be used as a drill-down from the Overview graph by using the Rankings -> Waits -> Time waiting on objects rankings option or simply use the first option in the menu which would typically be Time waiting on objects by wait object.

From these ranking graphs you can right-click the desired bars and graph their contributions over time.

8.14.19.1 Time waiting on objects by wait object

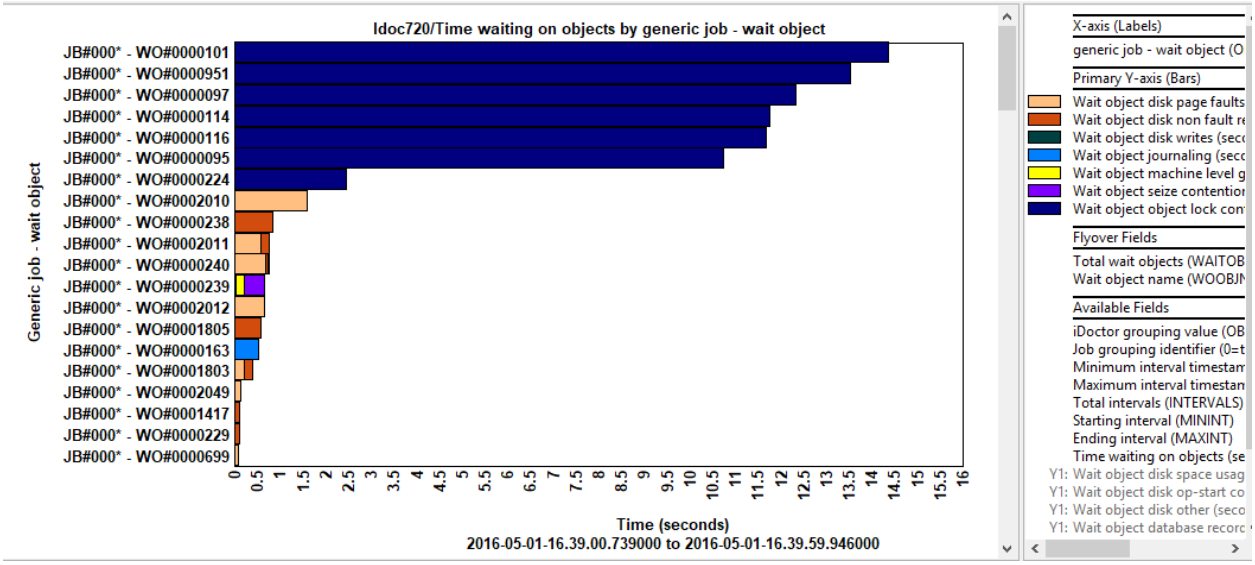
This graph ranks the selected time period's current wait times for job's experiencing interesting waits and having a wait object. This grouping is by wait object name only so multiple jobs could have contributed to each bar shown.



Time waiting on objects by wait object

8.14.19.2 Time waiting on objects by generic job, wait object

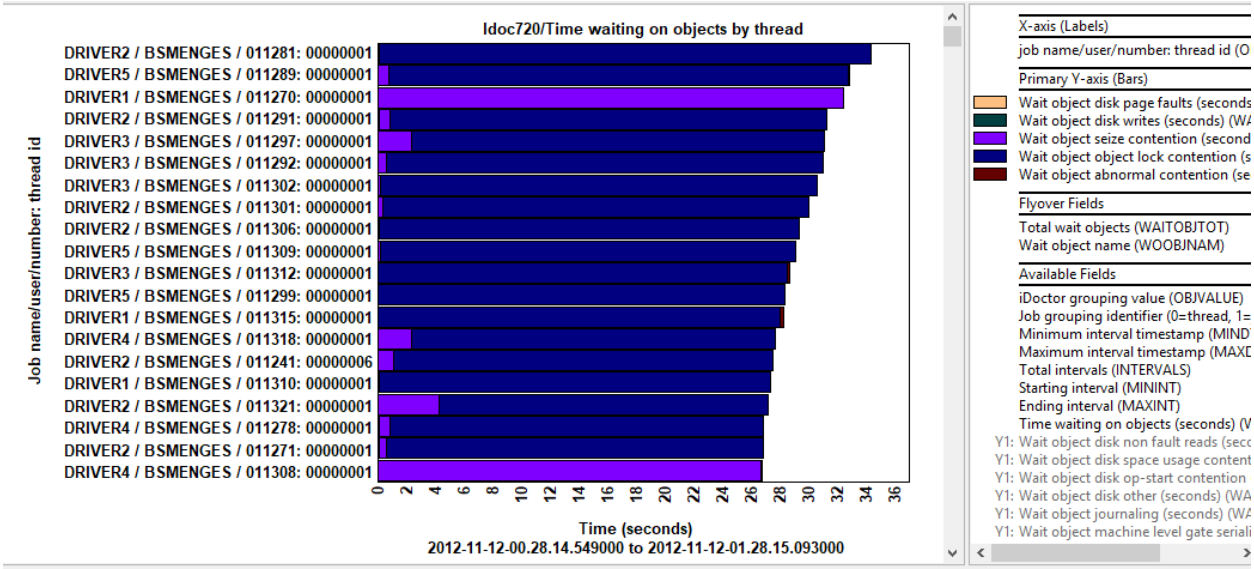
This graph ranks the selected time period's current wait times for job's experiencing interesting waits and having a wait object. This grouping is by generic job and wait object combination.



Time waiting on objects by generic job, wait object

8.14.19.3 Time waiting on objects by thread

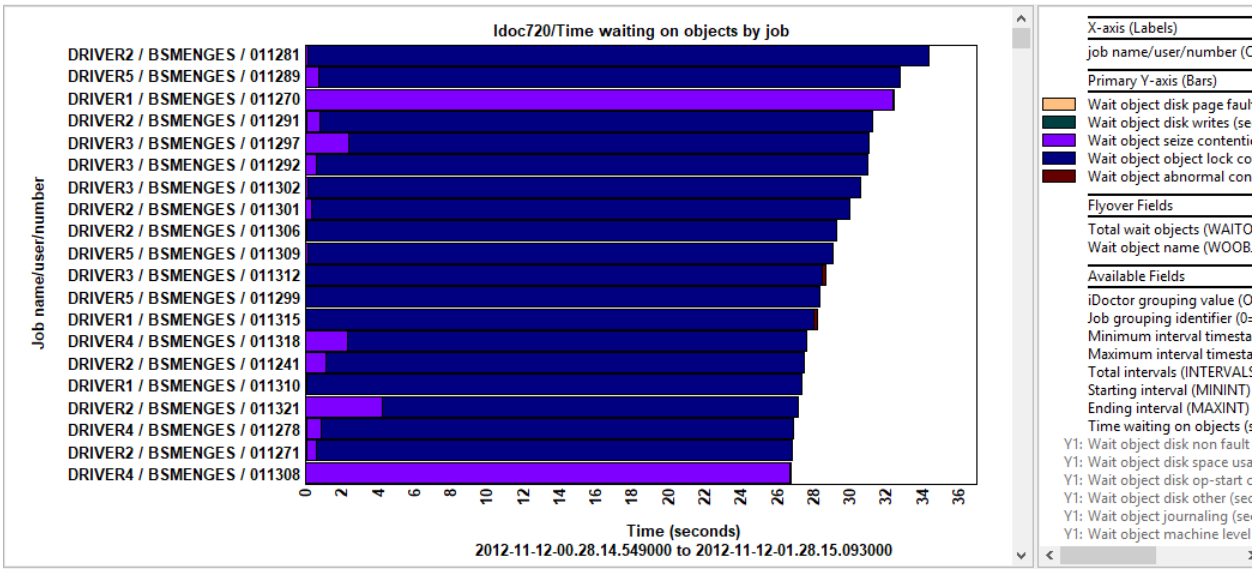
This graph ranks the selected time period's current wait times for job's experiencing interesting waits and having a wait object. This grouping is by job thread/task.



Time waiting on objects by thread

8.14.19.4 Time waiting on objects by job

This graph ranks the selected time period's current wait times for job's experiencing interesting waits and having a wait object. This grouping is by job name.



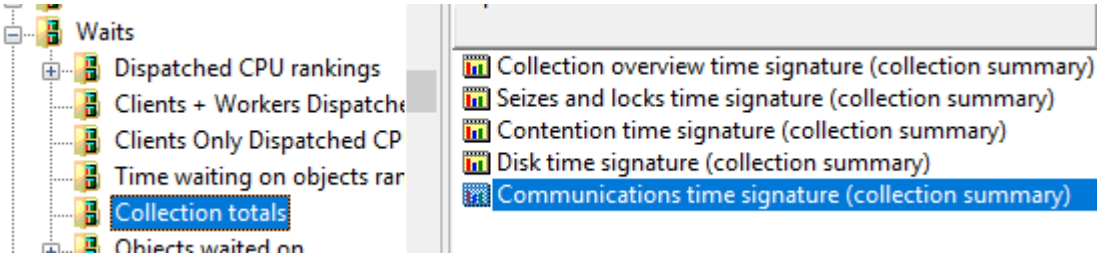
Time waiting on objects by job

The rest of the ranking graphs of this type will follow the same format as those described previously in the **Waits -> Dispatched CPU rankings folder**.

8.14.20 Collection totals

Under the Collection totals subfolder are a set of pie charts showing some of the same graphs from the Wait graphs folder except the times are added together across the entire collection instead of on a per interval basis.

Note: These graphs require the Collection summary analysis to be ran in order to appear!



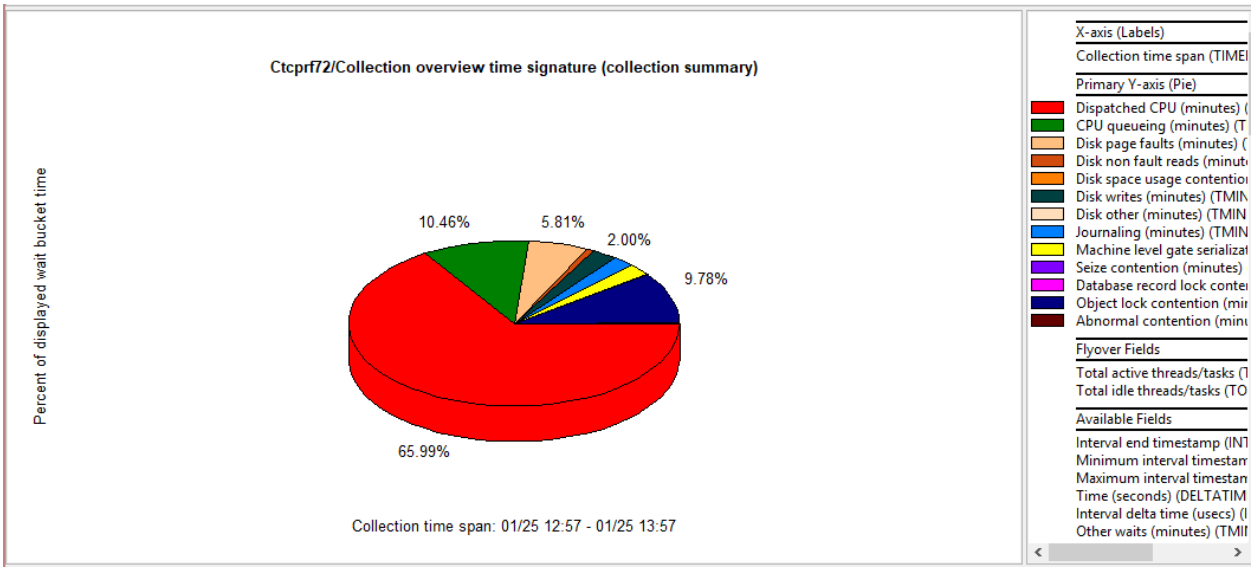
Waits -> Collection totals folder

8.14.20.1 Collection overview time signature (collection summary)

This graph shows interesting wait bucket times added up and show as a percentage among ONLY the interesting types of wait buckets.

It's important to realize that these percentages shown are only based on the waits given in the legend and NOT for all possible wait types. The wait types that are typically NOT of interest like idle communications or PASE waits are excluded. You can modify the legend to add or remove fields by right-clicking the desired bucket or use drag and drop.

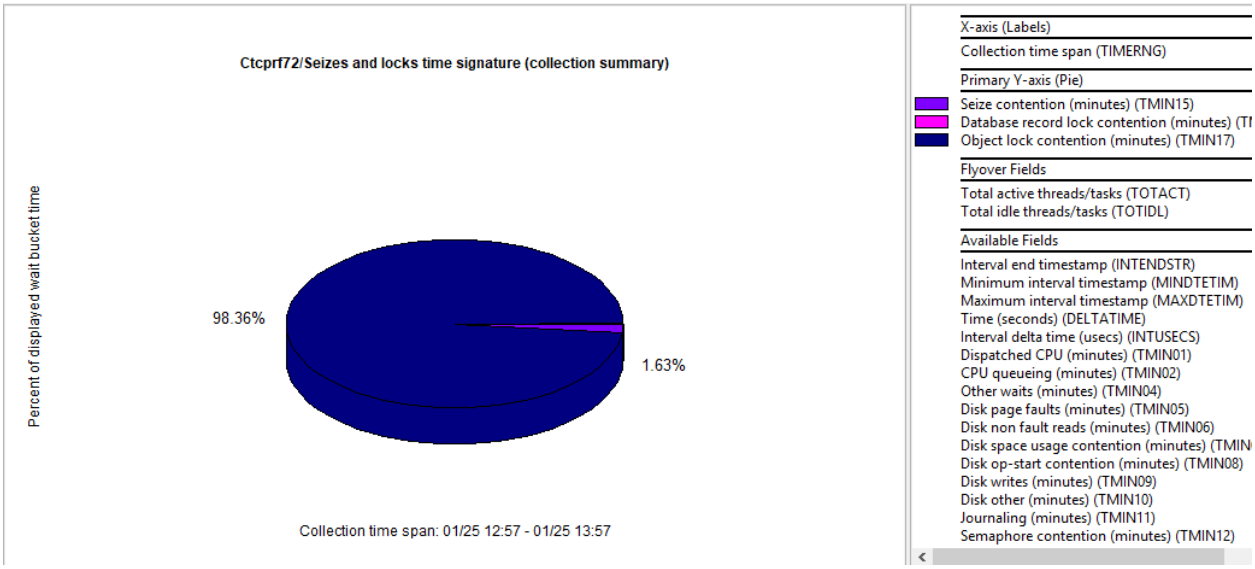
Tip: You can drill down from these graphs in the same way that you drill down from the Collection overview time signature graph, but keep in mind that all drill downs will be against the entire collection.



Collection overview time signature (collection summary)

8.14.20.2 Seizes and locks time signature (collection summary)

This graph displays only the wait bucket times related to seizes, DB record locks and object lock contention for the entire collection.



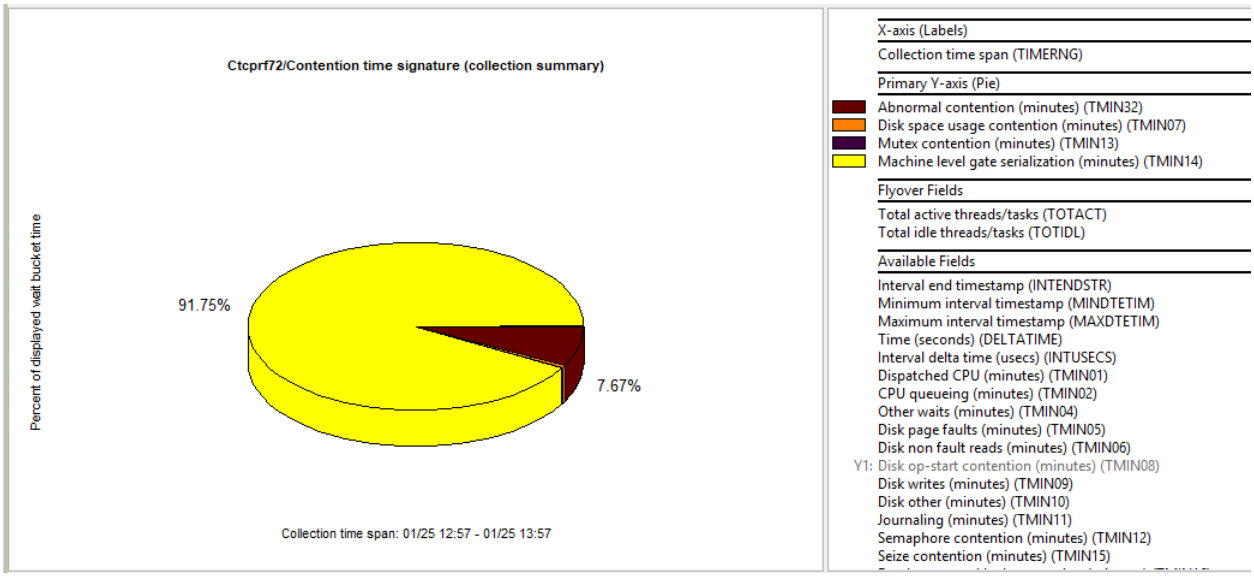
Seizes and locks time signature (collection summary)

8.14.20.3 Contention time signature (collection summary)

This graph only shows several wait buckets that are often associated with some type of contention on the system. It shows the following wait buckets:

1) Abnormal contention (32)
2) Disk space usage contention (7)

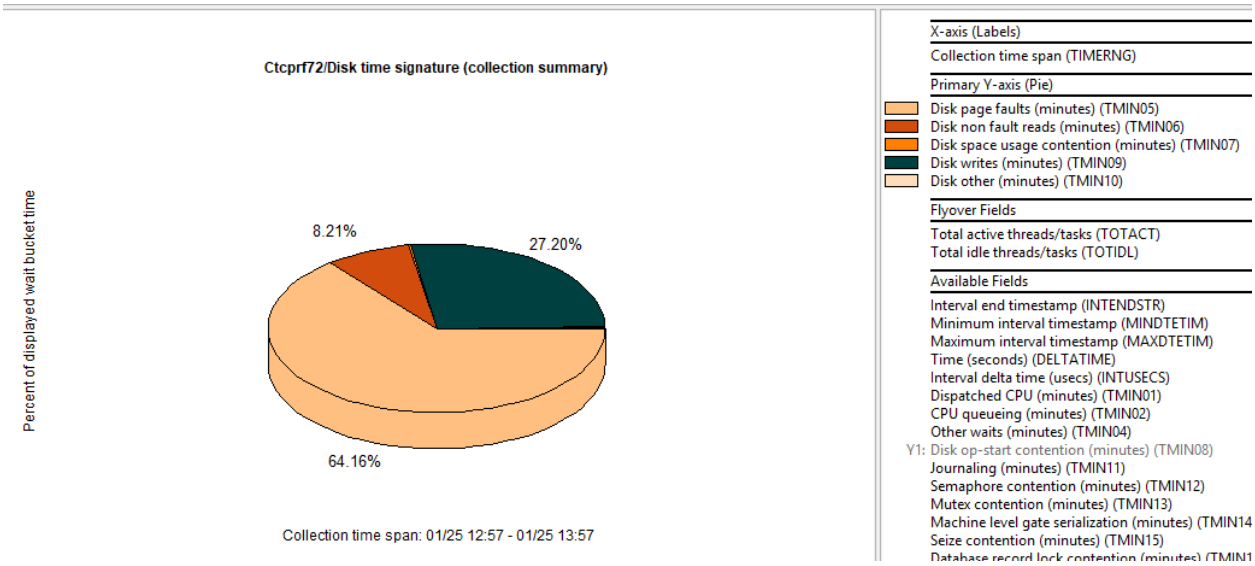
3) Disk op-start contention (8)
4) Mutex contention (13)
5) Machine level gate serialization (14)



Contention time signature (collection summary)

8.14.20.4 Disk time signature (collection summary)

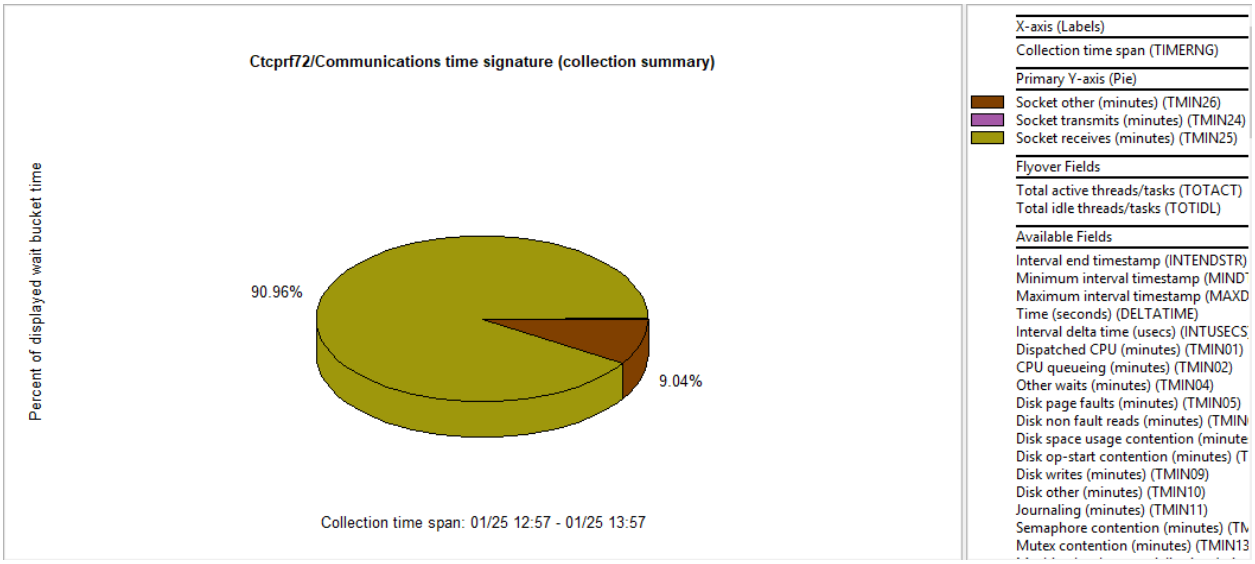
This graph shows only wait buckets associated disk related times.



Disk time signature (collection summary)

8.14.20.5 Communications time signature (collection summary)

This graph displays wait bucket times related to socket communications.

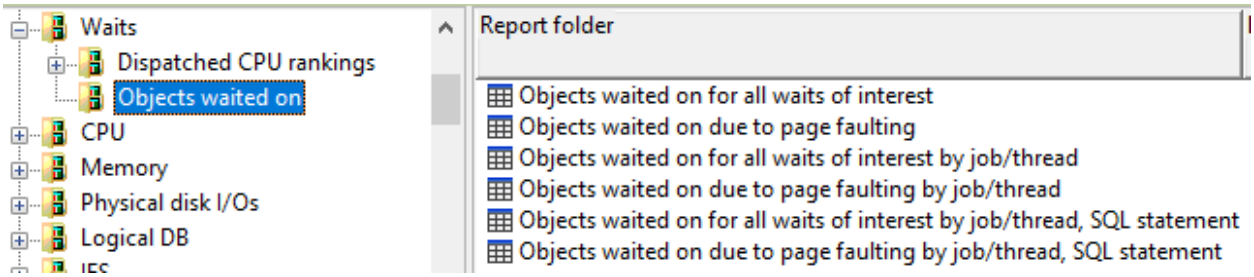


Communications time signature (collection summary)

8.14.21 Objects waited on

This folder contains a set of reports that summarize information about the wait objects found in the current collection.

Note: These graphs require the Collection summary analysis to be ran in order to appear!



Waits -> Objects waited on folder

Some of the columns shown in these reports are described further in the table below:

Column description	Notes
Total intervals	This is the total time periods found in the data where the given wait object/job and/or SQL statement was detected.
Total wait time (ms)	This is the total time spent waiting (in milliseconds) in an "interesting" wait associated with the wait object shown at the snapshots collected. It includes time for intervals where no CPU was used. It would NOT include short lived time waiting on object between snapshot boundaries not captured by Job Watcher.
Total waits at snapshot	This count is the bare minimum number of waits that occurred for this wait object. The true value is unknown and possibly more than this.
Total wait counts (could be for other wait objects)	This count is the maximum number of waits that occurred for this wait object. Because of the way Job Watcher works this count may not entirely apply to the wait object shown and could be much higher than reality.
Maximum wait time (ms)	The maximum time waiting (in milliseconds) on a wait object.
Average wait time (ms)	The average time waiting (in milliseconds) on a wait object.
Total jobs/tasks/threads	The number of taskcounts (jobs/tasks/threads) associated with the detected wait object.

8.14.21.1 Objects waited on for all waits of interest

This report only shows wait objects detected and associated with the “interesting” waits included with the Collection overview time signature graph. The report sorts the wait objects/wait buckets detected by the total intervals and total wait time.

/BSMENGES/RUN1/Objects waited on for all waits of interest - #1

Total intervals (INTERVAL	Total wait time (ms) (WAITTIME	Total waits at snapshot (TOTWAITS_SNAP)	Total wait counts (could be for other wait objects) (TOTWAITCOUNTS)	Maximum wait time (ms) (MAXWAIT	Average wait time (ms) (AVGWAITM	Wait object name (WOOBJNAM)	Wait object type description (WOOBJTYPD)	Bucket number (BUCKETNU	Description (BUCKETDESC)	Total jo (JOBCN
136	1,(17.201 minutes	3,810	64,545	31,172.3670	271.8729	BSMENGES	USER PROFILE	15	Seize contention	
55	72,638.3870	566	2,246	689.0640	128.3721	SLOT13 SLOT13	PHYSCIAL FILE MBR - DATA PART	17	Object lock contention	
52	672,714.2220	459	1,896	34,239.8200	1,465.6083	SLOT12 SLOT12	PHYSCIAL FILE MBR - DATA PART	17	Object lock contention	
49	4,307.1490	87	889	348.4360	49.5075	QQXTEMP1	FILE FORMAT	5	Disk page faults	
40	4,188.0590	79	466	501.7090	53.0134	QXCFQDT	SPACE OBJECT	5	Disk page faults	
29	331,527.9040	219	1,561	18,610.0180	1,454.8922	QDBSHR	USER PROFILE	15	Seize contention	
21	678.0840	28	229	137.8260	24.2173	SLOT13 SLOT13	PHYSCIAL FILE MBR - DATA PART	15	Seize contention	
20	535.5190	25	1,094	150.9130	21.4208	BSMENGES	USER PROFILE	9	Disk writes	
18	371.4400	28	1,313	124.1580	13.2657	QQXTEMP1	FILE	9	Disk writes	
17	1,766.3190	24	374	461.1320	73.5966		DATA SPACE	5	Disk page faults	
16	1,155.4770	28	414	135.4060	41.2670	SYSIXADV SYSIXADV	PHYSCIAL FILE MBR - DATA PART	9	Disk writes	
15	453.3630	24	2,026	233.3130	18.8901	QQXTEMP1 QQXTEMP	PHYSCIAL FILE MBR - DATA PART	9	Disk writes	
12	1,208.7970	15	1,443	613.1850	80.5865	QQXTEMP1	FILE FORMAT	9	Disk writes	
11	585.3880	17	114	133.0610	34.4346	SLOT12 SLOT12	PHYSCIAL FILE MBR - DATA PART	15	Seize contention	
10	2,605.8070	70	64	124.2160	37.2258	SLOT12 SLOT12	DB2 FILE MEMBER	17	Object lock contention	
10	1,822.6280	49	59	130.6320	37.1965	SLOT13 SLOT13	DB2 FILE MEMBER	17	Object lock contention	
10	551.4730	11	1	134.0460	50.1339	QQXTEMP1 QQXTEMP	PHYSCIAL FILE MBR - DATA PART	15	Seize contention	
9	856.2910	9	1,737	722.8320	95.1434	QAPYJWTDE RUN1	PHYSCIAL FILE MBR - DATA PART	5	Disk page faults	
8	634.5690	8	21	367.3450	79.3211	SYSIXADV SYSIXADV	PHYSCIAL FILE MBR - DATA PART	16	Database record lock contention	
7	313.1910	7	1,011	185.7770	44.7416	QJOBMSGQ	JOB MESSAGE QUEUE	9	Disk writes	

Objects waited on for all waits of interest

8.14.21.2 Objects waited on due to page faulting

This report displays the time spent waiting on page faulting with a wait object present in the Job Watcher data.

/BSMENGES/RUN1/Objects waited on for all waits of interest - #1 /BSMENGES/RUN1/Objects waited on due to page faulting - #1

Total intervals (INTERVAL	Total wait time (ms) (WAITTIME	Total page faults at snapshot (TOTFLTS_SNAP)	Total page faults (could be for other wait objects) (TOTFLTS_OTHERS)	Maximum wait time (ms) (MAXWAIT	Average wait time (ms) (AVGWAITM	Wait object name (WOOBJNAM)	Wait object type description (WOOBJTYPD)	Total jobs/tasks/threads (JOBCNT)
49	4,307.1490	87	889	348.4360	49.5075	QQXTEMP1	FILE FORMAT	70
40	4,188.0590	79	466	501.7090	53.0134	QXCFQDT	SPACE OBJECT	70
17	1,766.3190	24	374	461.1320	73.5966		DATA SPACE	23
9	856.2910	9	1,737	722.8320	95.1434	QAPYJWTDE RUN1	PHYSCIAL FILE MBR - DATA PART	1
5	1,288.3560	5	192	911.1820	257.6712	QAPYJWSTK RUN1	PHYSCIAL FILE MBR - DATA PART	1
5	22.5000	5	781	12.9620	4.5000	PCS	TEMPORARY - PROCESS CTL SPACE	2
4	20.5920	4	290	8.9510	5.1480	QSYS	USER PROFILE	2

Objects waited on due to page faulting

8.14.21.3 Objects waited on for all waits of interest by job/thread

This report only shows wait objects and job/threads detected and associated with the “interesting” waits included with the Collection overview time signature graph. The report sorts the wait objects/wait buckets/jobs detected by the total intervals and total wait time.

/BSMENGES/RUN1/Objects waited on for all waits of interest - #1 | /BSMENGES/RUN1/Objects waited on due to page faulting - #1 | /BSMENGES/RUN1/Objects waited on for all waits of interest by job/thread - #1

Total intervals (INTERVAL	Total wait time (ms) (WAITTIME	Total waits at snapshot (TOTWAITS_SNAP)	Total wait counts (could be for other wait objects) (TOTWAITCOUNTS)	Maximum wait time (ms) (MAXWAIT	Average wait time (ms) (AVGWAITM	Wait object name (WOOBJNA	Wait object type description (WOOBJTYPD)	Bucket number (BUCKETNU	Description (BUCKETDESC)	Job name/user/number: thread ID (JTTHREAD)	Task count identifies a task/thre (TASKCOU
42	9,595.8780	41	156	1,	33.1720	BSMENGES	USER PROFILE	15	Seize contention	QPADEV0003 / BSMENGES / 011021: 00000038	
25	7,341.2060	23	160	2,136.5330	340.7224	BSMENGES	USER PROFILE	15	Seize contention	DRIVER5 / BSMENGES / 011790: 00000002	
24	4,187.0390	23	165	1,273.6820	183.2477	BSMENGES	USER PROFILE	15	Seize contention	DRIVER5 / BSMENGES / 011795: 00000006	
23	5,817.4040	21	174	1,729.0290	285.1277	BSMENGES	USER PROFILE	15	Seize contention	DRIVER5 / BSMENGES / 011785: 00000002	
23	4,507.5860	23	149	765.9810	195.9820	BSMENGES	USER PROFILE	15	Seize contention	DRIVER2 / BSMENGES / 011767: 00000006	
23	4,252.1270	23	156	744.9280	184.8751	BSMENGES	USER PROFILE	15	Seize contention	DRIVER2 / BSMENGES / 011762: 00000005	
22	5,458.6570	21	150	1,127.2210	252.1128	BSMENGES	USER PROFILE	15	Seize contention	DRIVER4 / BSMENGES / 011804: 00000006	
22	5,174.7130	20	183	2,455.2140	314.8657	BSMENGES	USER PROFILE	15	Seize contention	DRIVER4 / BSMENGES / 011794: 00000004	

1.274 seconds

Objects waited on for all waits of interest by job/thread

8.14.21.4 Objects waited on due to page faulting by job/thread

This report shows where page faulting was detected as occurring the most by job/thread/wait object.

/BSMENGES/RUN1/Objects waited on for all waits of intere... | /BSMENGES/RUN1/Objects waited on due to page faulting ... | /BSMENGES/RUN1/Objects waited on for all waits of intere... | /BSMENGES/RUN1/Objects waited

Total intervals (INTERVAL	Total wait time (ms) (WAITTIME	Total page faults at snapshot (TOTFLTS_SNAP)	Total page faults (could be for other wait objects) (TOTFLTS_OTHERS)	Maximum wait time (ms) (MAXWAIT	Average wait time (ms) (AVGWAITM	Wait object name (WOOBJNAM)	Wait object type description (WOOBJTYPD)	Job name/user/number: thread ID (JTTHREAD)	Task cou identifies a task/th (TASKCO
9	856.2910	9	1,737	722.8320	95.1434	QAPYJWTDE RUN1	PHYSCIAL FILE MBR - DATA PART	QZDASOINIT / QUSER / 011234: 00000025	
5	1,288.3560	5	192	911.1820	257.6712	QAPYJWSTK RUN1	PHYSCIAL FILE MBR - DATA PART	QDBFSTCCOL / QSYS / 009542: 00000005	
3	307.2180	3	36	157.7510	102.4060	QQXTEMP1	FILE FORMAT	DRIVER4 / BSMENGES / 011533: 00000003	
3	17.6380	3	409	12.9620	5.8793	PCS	TEMPORARY - PROCESS CTL SPACE	QWCTJOBS / QSYS / 009544: 00000001	
3	13.9610	3	205	8.9510	4.6537	QSYS	USER PROFILE	QWCPJOBS / QSYS / 009543: 00000001	
2	590.9420	2	23	550.2060	295.4710	QIDRPATPRF	PROGRAM	QZDASOINIT / QUSER / 011866: 0000002D	
2	343.5790	2	23	319.4670	171.7895	QXCFQDT	SPACE OBJECT	DRIVER2 / BSMENGES / 011536: 00000002	
2	334.7140	2	23	316.6770	167.3570		DATA SPACE	DRIVER2 / BSMENGES / 011536: 00000002	

Objects waited on due to page faulting by job/thread

8.14.21.5 Objects waited on for all waits of interest by job/thread, SQL statement

This report only shows wait objects and job/threads with an SQL statement detected and associated with the “interesting” waits included with the Collection overview time signature graph. The report sorts the data found by the total intervals and total wait time.

/BSMENGES/RUN1/Objects waited on for all w... | /BSMENGES/RUN1/Objects waited on due to ... | /BSMENGES/RUN1/Objects waited on for all w... | /BSMENGES/RUN1/Objects waited on due to ... | /BSMENGES/RUN1/Objects waited on for a...

Total intervals (INTERVAL	Total wait time (ms) (WAITTIME	Total waits at snapshot (TOTWAITS_SNAP)	Total wait counts (could be for other wait objects) (TOTWAITCOUNTS)	Maximum wait time (ms) (MAXWAIT	Average wait time (ms) (AVGWAITM	Wait object name (WOOBJNAM)	Wait object type description (WOOBJTYPD)	Bucket number (BUCKETNU	Description (BUCKETDESC)	Job name/user/number: thread ID (JTTHREAD)	Task count identifies a task/threa (TASKCO...	SQL statement (SQLSTMT)
5	50.8820	5	761	16.4440	10.1764	QAPYJWTDE RUN1	PHYSCIAL FILE MBR - DATA PART	5	Disk page faults	QZDASOINIT / QUSE>	13,711	select count(>
3	195.4970	3	27	113.2560	65.1657	SLOT12 SLOT12	PHYSCIAL FILE MBR - DATA PART	17	Object lock content>	DRIVER2 / BSMENGES>	14,331	INSERT INTO>
2	715.7150	2	6	564.5040	357.8575	SLOT13 SLOT13	PHYSCIAL FILE MBR - DATA PART	17	Object lock content>	DRIVER1 / BSMENGES>	14,392	INSERT INTO>
2	519.1250	2	6	508.7910	259.5625	SLOT13 SLOT13	PHYSCIAL FILE MBR - DATA PART	17	Object lock content>	DRIVER2 / BSMENGES>	14,387	INSERT INTO>
2	467.3960	2	6	366.5420	233.6980	SLOT12 SLOT12	PHYSCIAL FILE MBR - DATA PART	17	Object lock content>	DRIVER4 / BSMENGES>	14,354	INSERT INTO>
2	446.1640	2	9	288.7650	223.0820	SLOT12 SLOT12	PHYSCIAL FILE MBR - DATA PART	17	Object lock content>	DRIVER4 / BSMENGES>	14,308	INSERT INTO>

Objects waited on for all waits of interest by job/thread, SQL statement

8.14.21.6 Objects waited on due to page faulting by job/thread, SQL statement

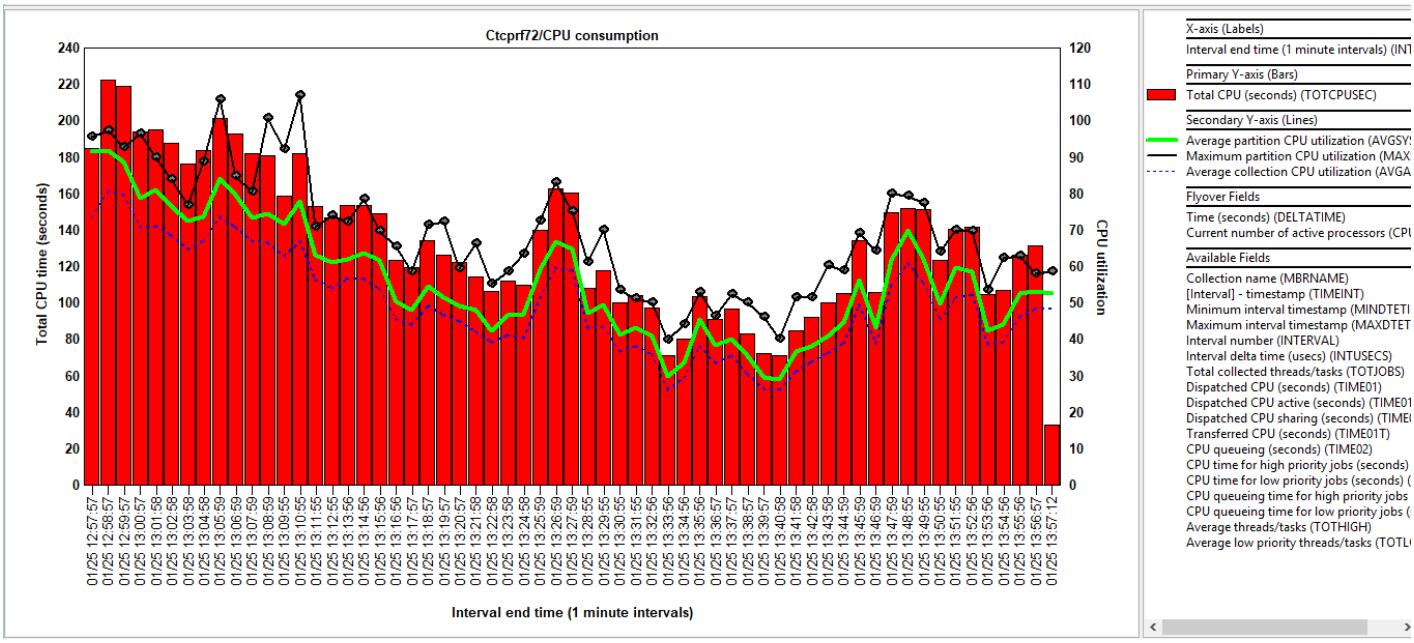
This report shows where page faulting was detected as occurring the most by job/thread/wait object and SQL statement.

8.15 CPU

These graphs show CPU utilization and CPU times for the collection in various ways.

8.15.1 CPU consumption

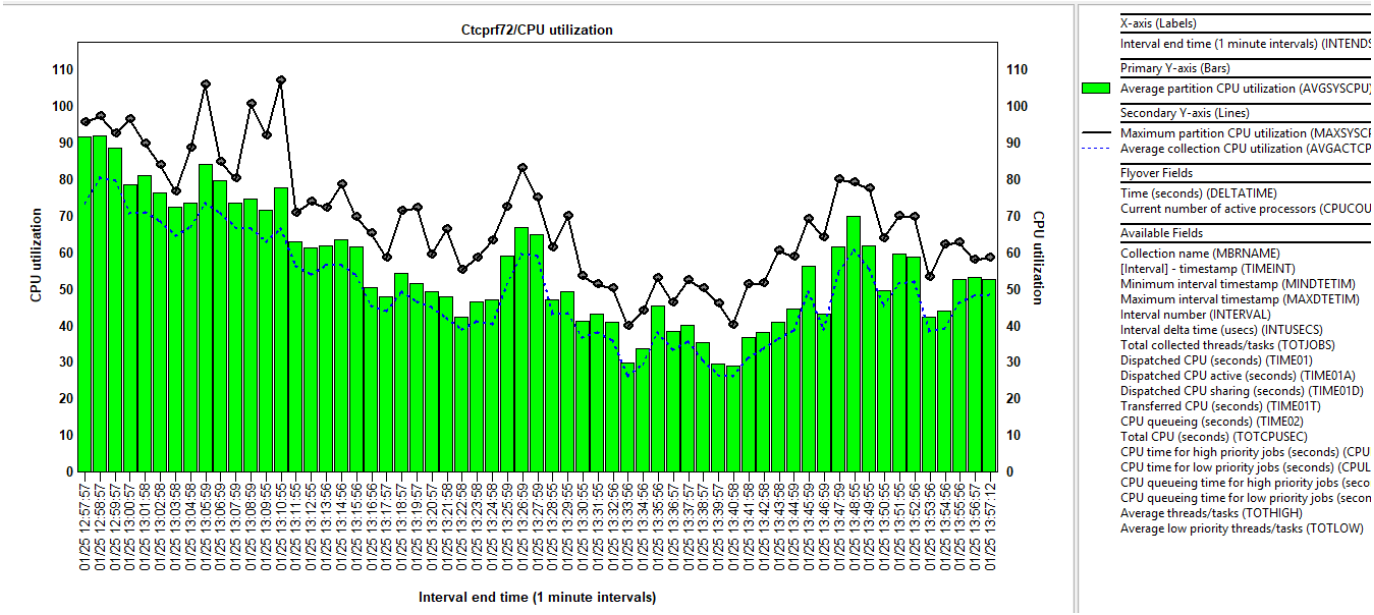
This graph displays the total CPU time consumed over time with the CPU utilization percentages of the Y2-axis.



CPU consumption

8.15.2 CPU utilization

This graph displays the CPU utilization average on the Y1-axis and the Y2-axis displays the maximum and minimum utilizations. The later values are the minimum and maximum **averages** when graphing multiple intervals per bar.



CPU utilization

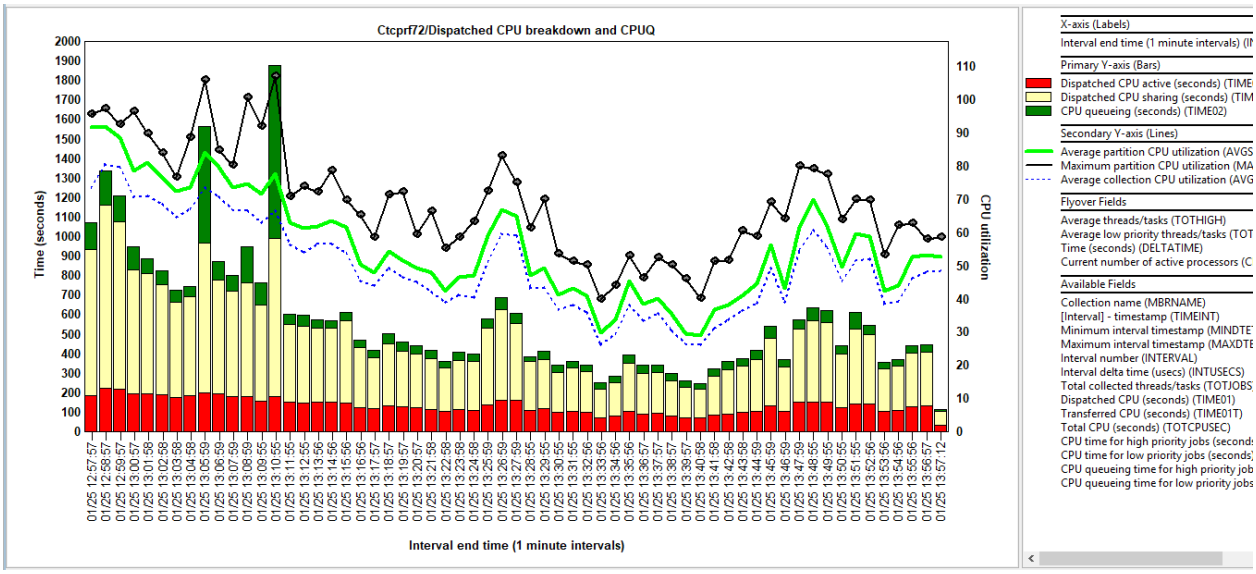
8.15.3 Dispatched CPU breakdown and CPUQ

This graph shows CPU utilization, CPU queuing and CPU dispatched time divided into 2 different buckets:

Dispatched CPU active (red) – This is time spent burning CPU.

Dispatched CPU sharing (light yellow) – This is a type of wait time we can measure where we are dispatched to the processor but NOT actually burning CPU and sharing it with out work. This is normal behavior and many environments and not typically cause for concern.

CPU Queuing – The time indicates the process is not dispatched to the processor and waiting (in line) to use the CPU.

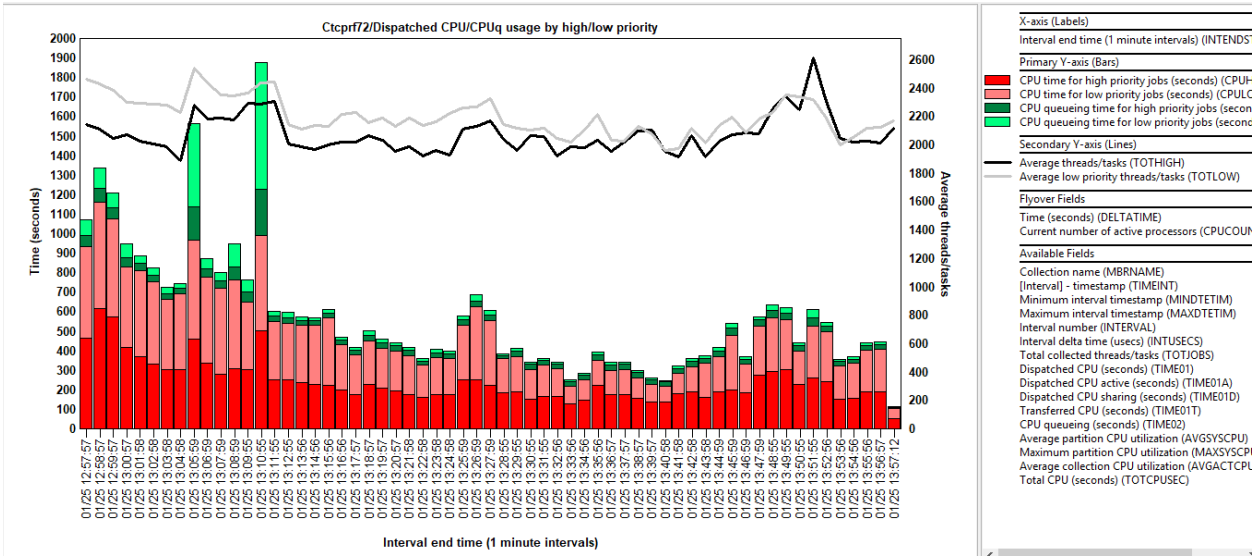


Dispatched CPU breakdown and CPUQ

8.15.4 Dispatched CPU/CPUq usage by high/low priority

This graph shows CPU and CPU queuing times grouped by high or low priority jobs. For the purpose of the graph, high priority is considered 29 or less. Low priority jobs are considered priority 30 or higher.

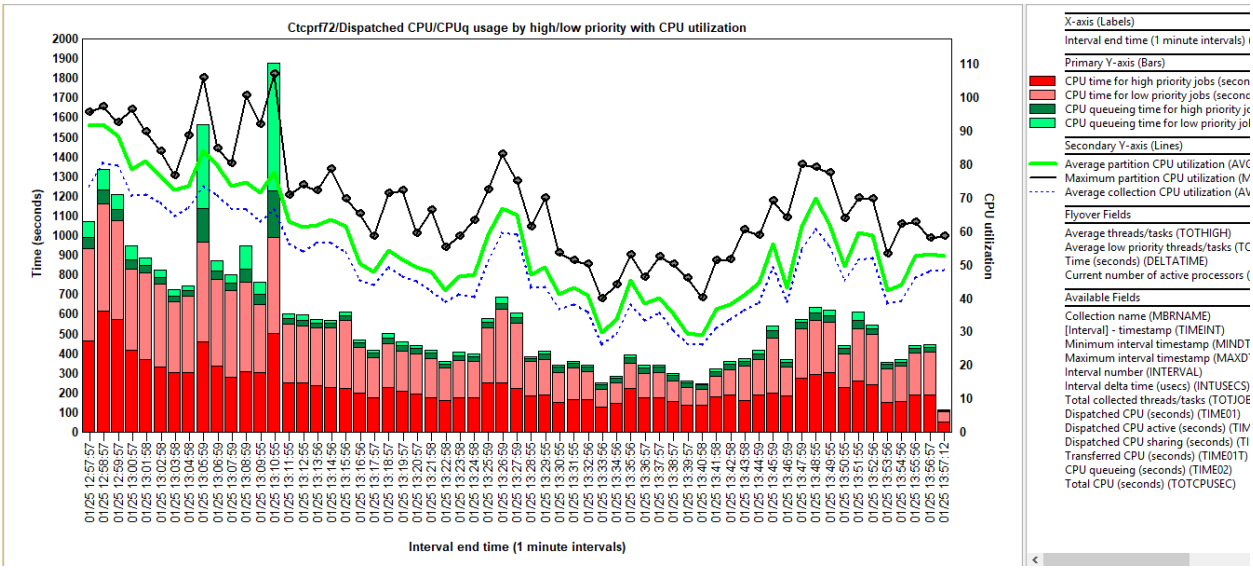
The graph also shows the average number of threads/tasks and the average number of low priority threads/tasks each interval.



Dispatched CPU/CPUq usage by high/low priority

8.15.5 Dispatched CPU/CPUq usage by high/low priority with CPU utilization

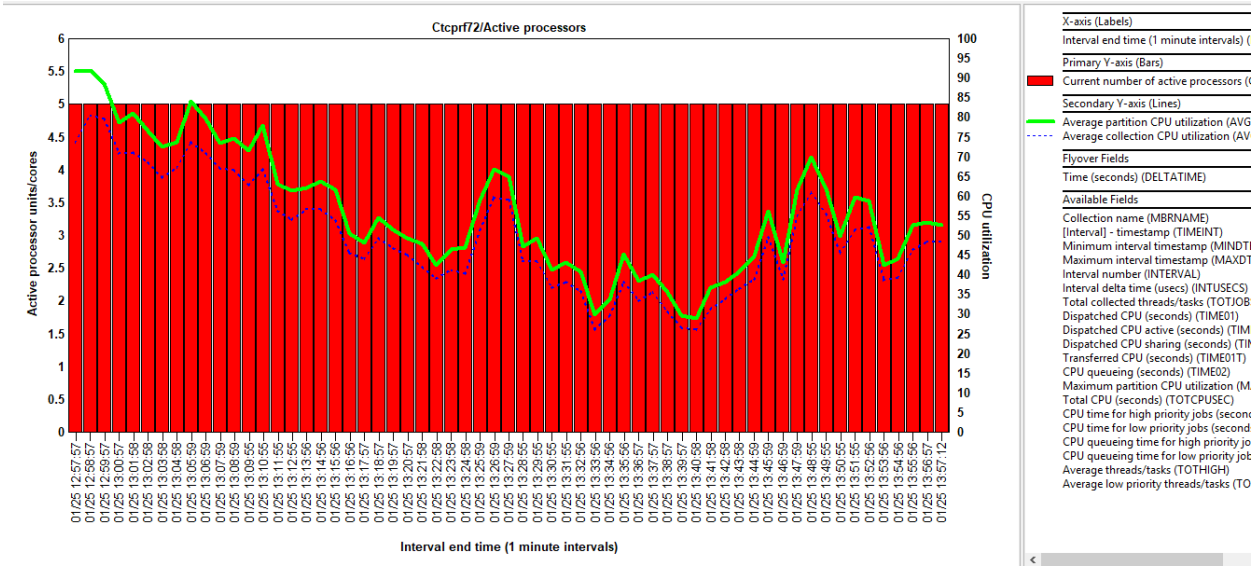
This graph shows the same graph as the previous one but with CPU utilization on the secondary Y-axis.



Dispatched CPU/CPUq usage by high/low priority with CPU utilization

8.15.6 Active processors

This graph displays the active virtual processors on the system over time along with CPU utilization on the Y2-axis.



Active processors

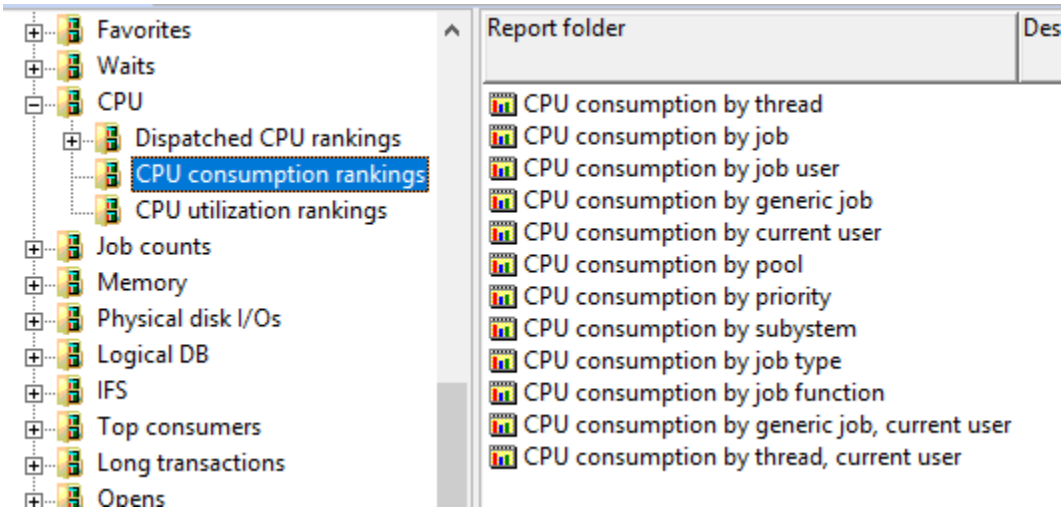
8.15.7 Dispatched CPU rankings

These graphs are identical to the graphs in the Waits -> Dispatched CPU rankings folder.

8.15.8 CPU consumption rankings

These graphs display CPU time consumed ranked by one of many different types of job groupings.

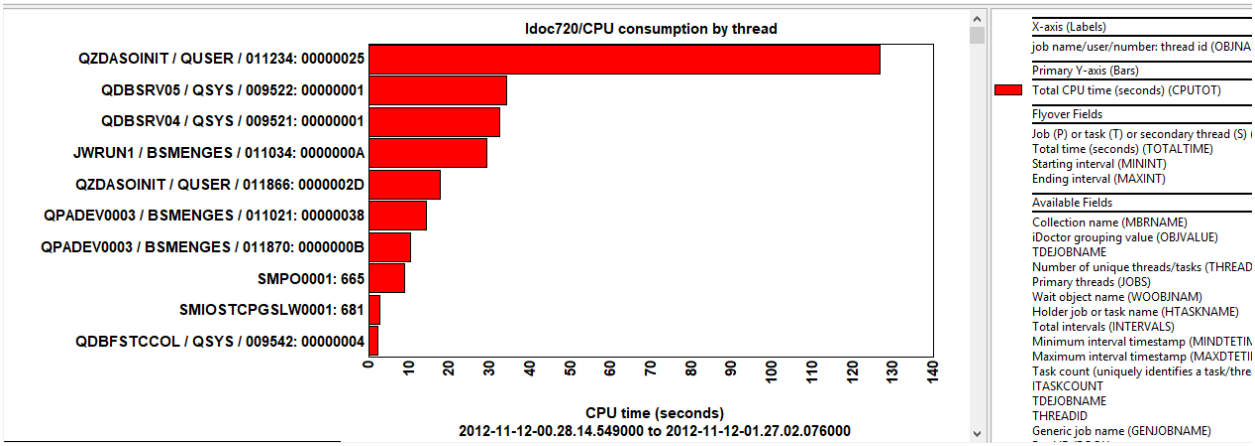
This type of CPU time is different than Dispatched CPU time. These graphs show time spent burning CPU rather than just dispatched to the processor and sharing it with other threads.



CPU consumption rankings

8.15.8.1 CPU consumption by thread

This graph ranks the selected time period's CPU consumption by thread or task.

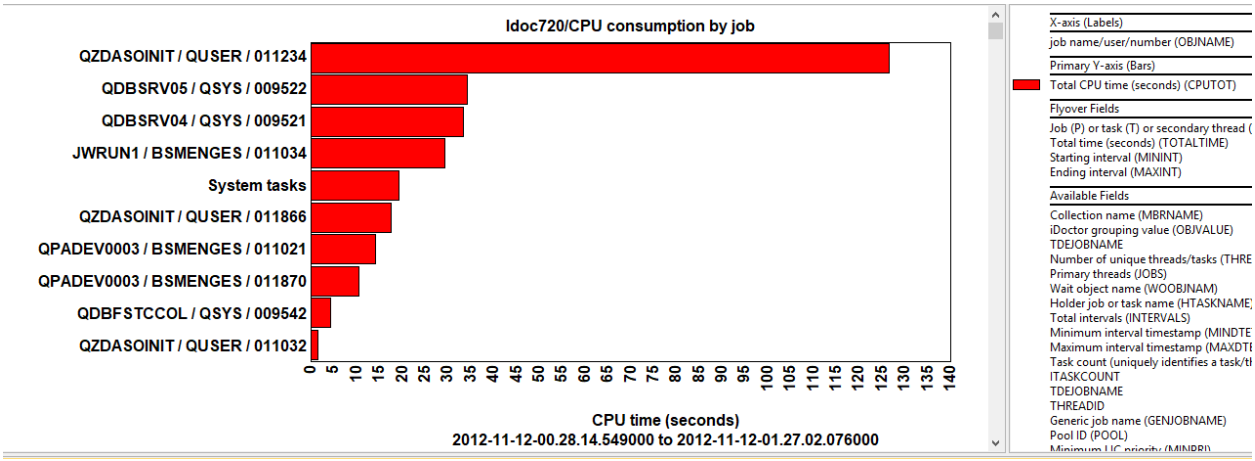


CPU consumption by thread

8.15.8.2 CPU consumption by job

This graph ranks the selected time period's CPU consumption by primary thread. Jobs that have multiple threads are all added up together.

Note: All system tasks are grouped together into one "**System tasks"** record within this report.

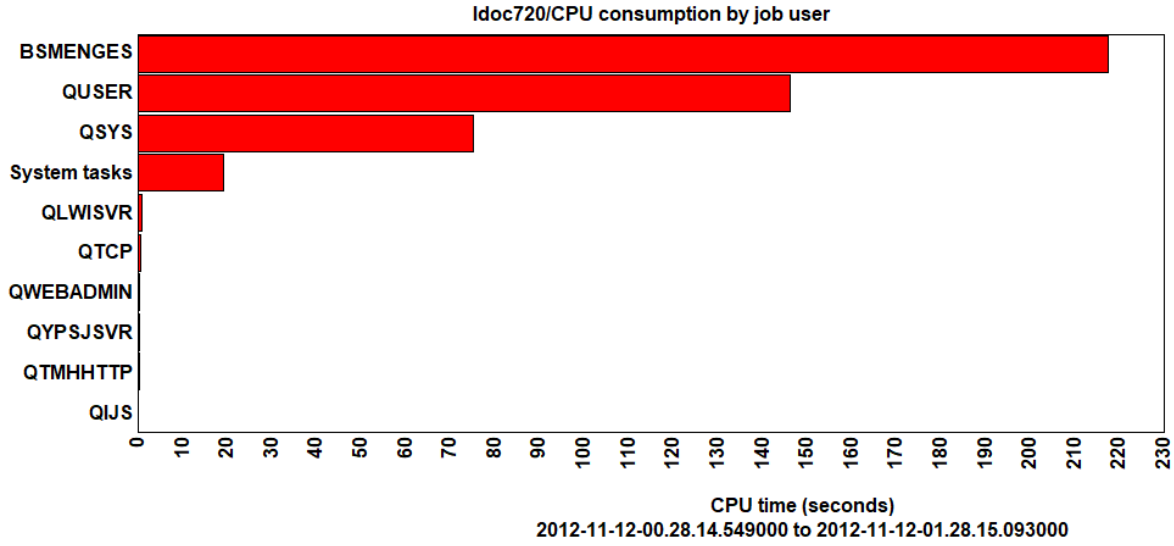


CPU consumption by job

8.15.8.3 CPU consumption by job user

This graph ranks the selected time period's CPU consumption by job user.

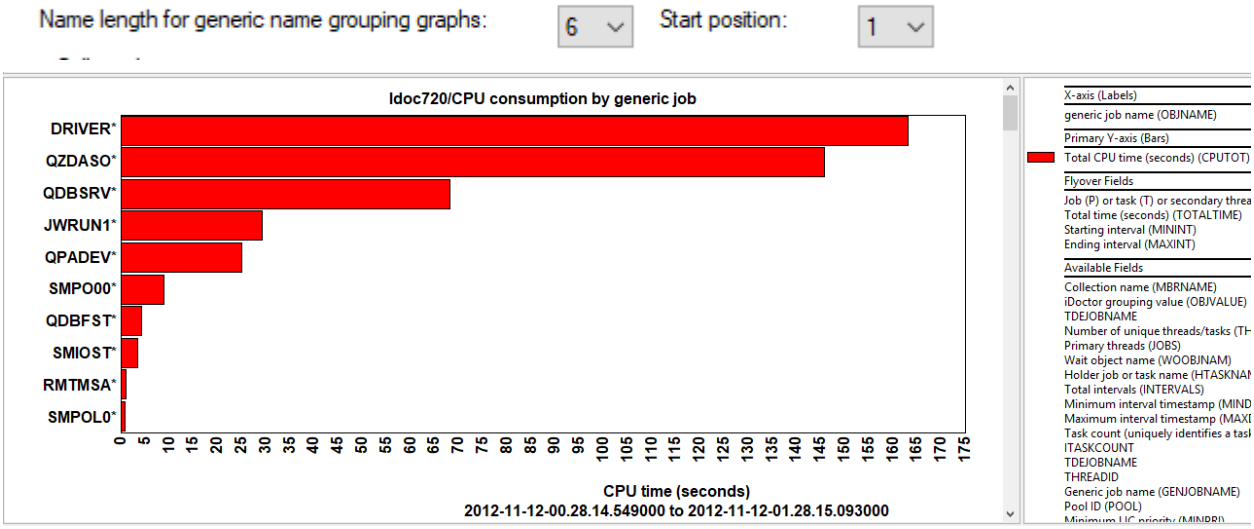
Note: All system tasks are grouped together into one "**System tasks"** record within this report.



CPU consumption by job user

8.15.8.4 CPU consumption by generic job

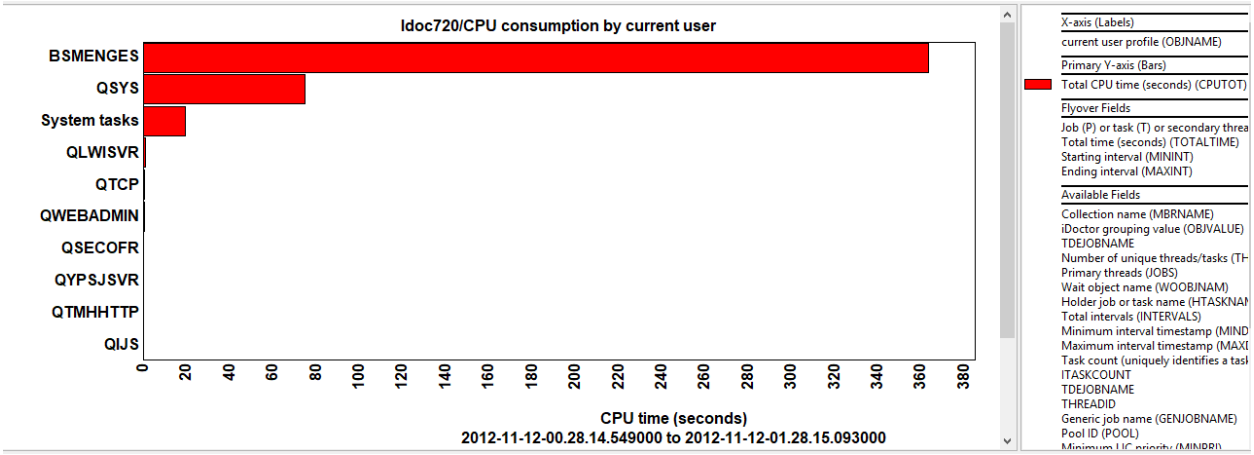
This graph ranks the selected time period's CPU consumption by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:



CPU consumption by generic job

8.15.8.5 CPU consumption by current user

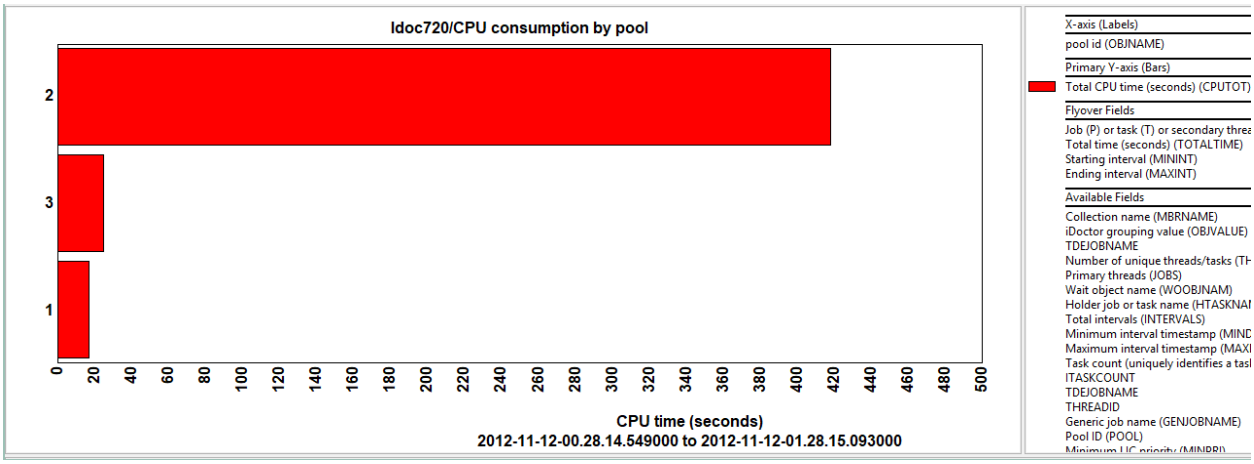
This graph ranks the selected time period's CPU consumption by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



CPU consumption by current user

8.15.8.6 CPU consumption by pool

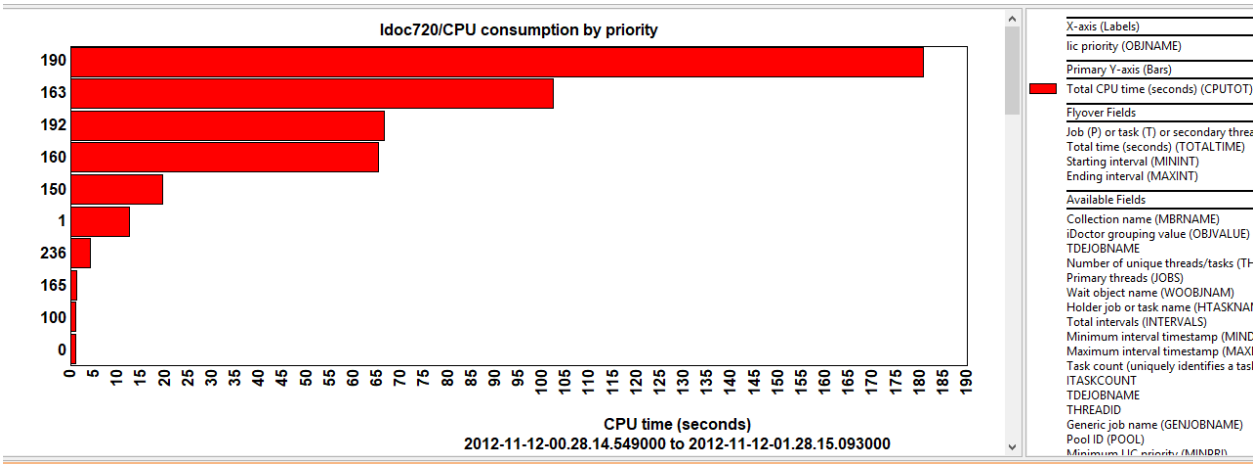
This graph ranks the selected time period's CPU consumption by memory pool.



CPU consumption by pool

8.15.8.7 CPU consumption by priority

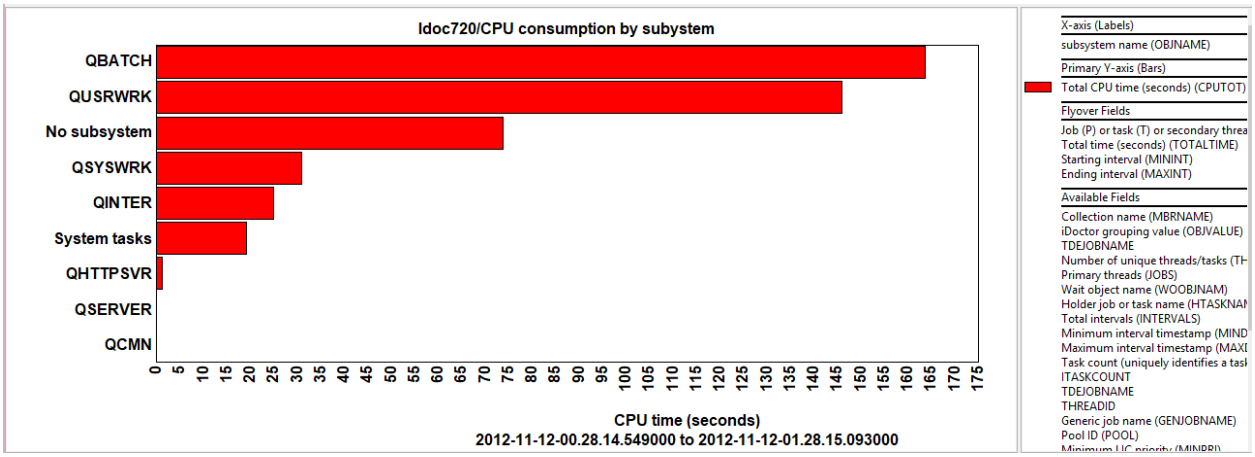
This graph ranks the selected time period's CPU consumption by LIC priority. XPF priority can be calculated by subtracting 140 for those values shown exceeding 140.



CPU consumption by priority

8.15.8.8 CPU consumption by subsystem

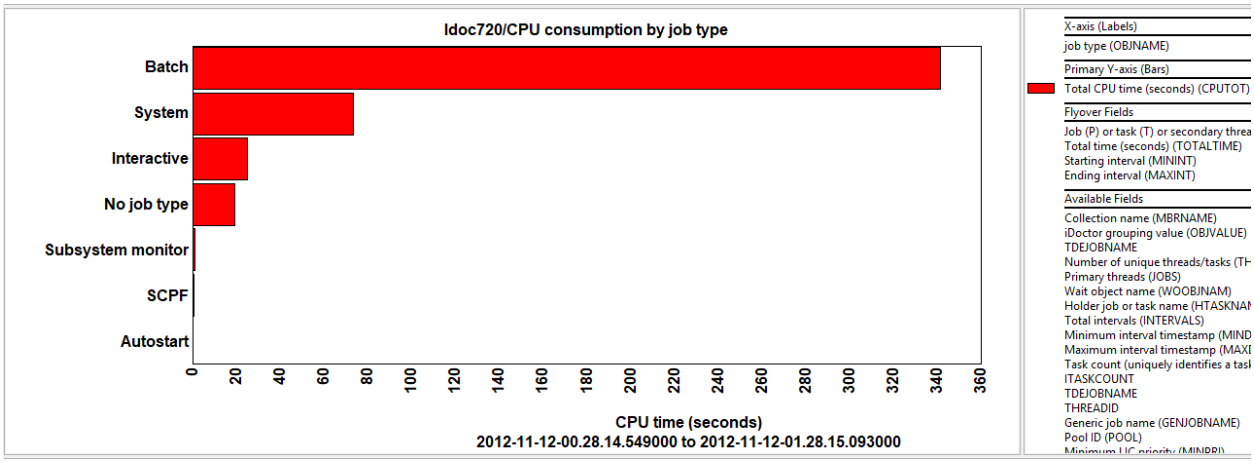
This graph ranks the selected time period's CPU consumption by subsystem. Job times that had no subsystem listed are grouped into 1 bar called "No subsystem". **Note:** All system tasks are grouped together into one "**System tasks"** bar within this report.



CPU consumption by subsystem

8.15.8.9 CPU consumption by job type

This graph ranks the selected time period's CPU consumption by job type (such as Batch, interactive, autostart, etc.) Job wait bucket times that had no job type associated with it are grouped into 1 bar called "No job type".

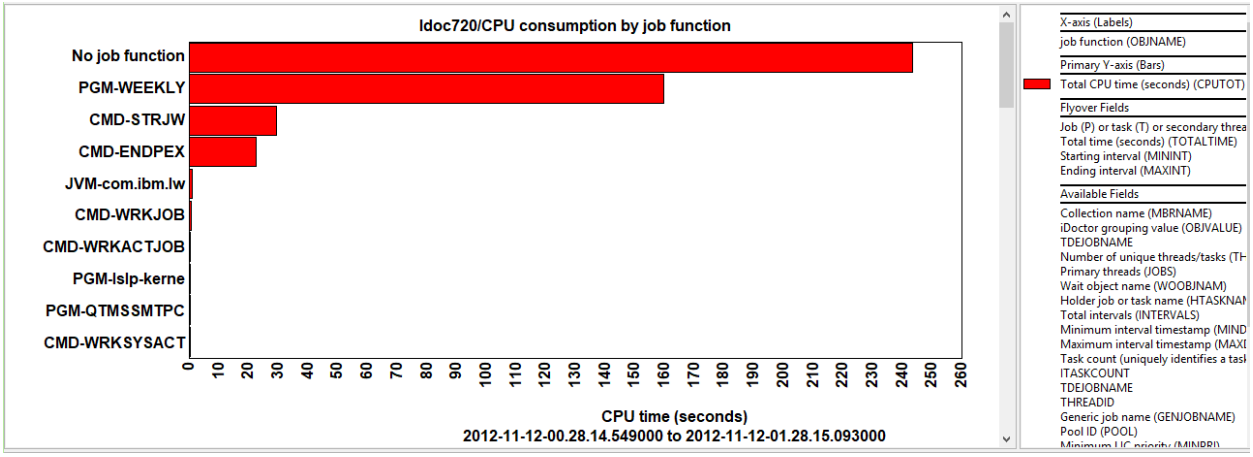


CPU consumption by job type

8.15.8.10 CPU consumption by job function

This graph ranks the selected time period's CPU consumption by job function. Job function is the same as what you would see normally on the WRKACTJOB command's Function column. Any wait bucket times that had no job function associated with it are grouped into 1 bar called "No job function".

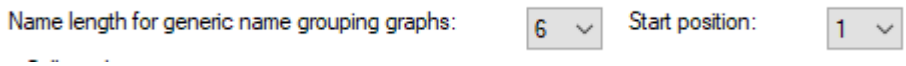
Note: This graph can be misleading in some situations because the current job function for each process is only captured at the end of each Job Watcher interval. The times given are not necessarily 100% from each function listed.



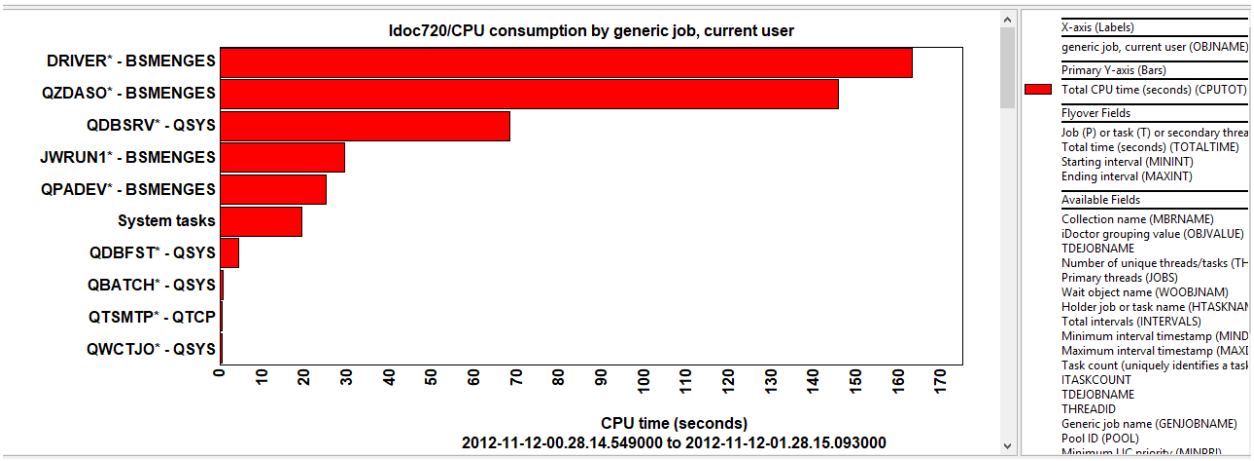
CPU consumption by job function

8.15.8.11 CPU consumption by generic job, current user

This graph ranks the selected time period's CPU consumption by generic job/current user profile combination. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:



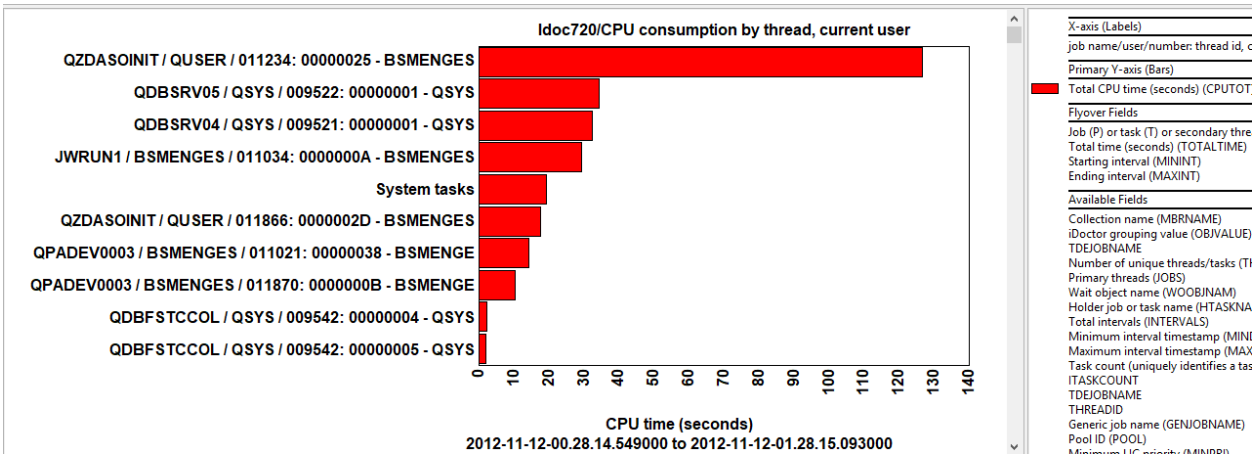
Note: All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



CPU consumption by generic job, current user

8.15.8.12 CPU consumption by thread, current user

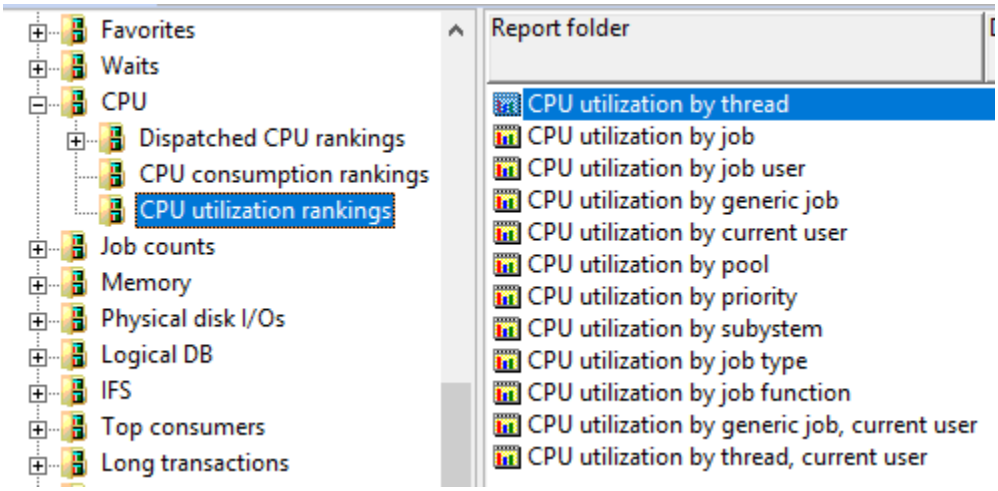
This graph ranks the selected time period's CPU consumption by thread/current user profile combination.
Note: All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



CPU consumption by thread, current user

8.15.9 CPU utilization rankings

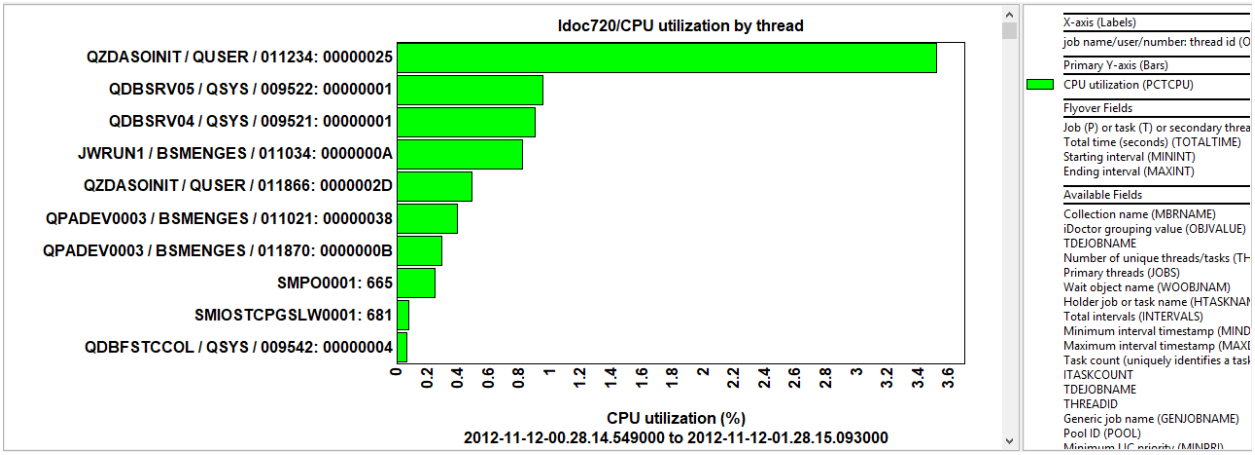
These graphs display CPU utilization ranked by one of many different types of job groupings.



CPU utilization rankings

8.15.9.1 CPU utilization by thread

This graph shows CPU utilization by thread or task.



CPU utilization by thread

The rest of these graphs are very similar to the CPU consumption rankings graphs described in the previous section.

8.16 Job counts

These graphs show the total jobs (primary threads), system tasks and threads (secondary threads) that exist on the system.

Note: These graphs require the Collection summary analysis to be ran in order to appear!

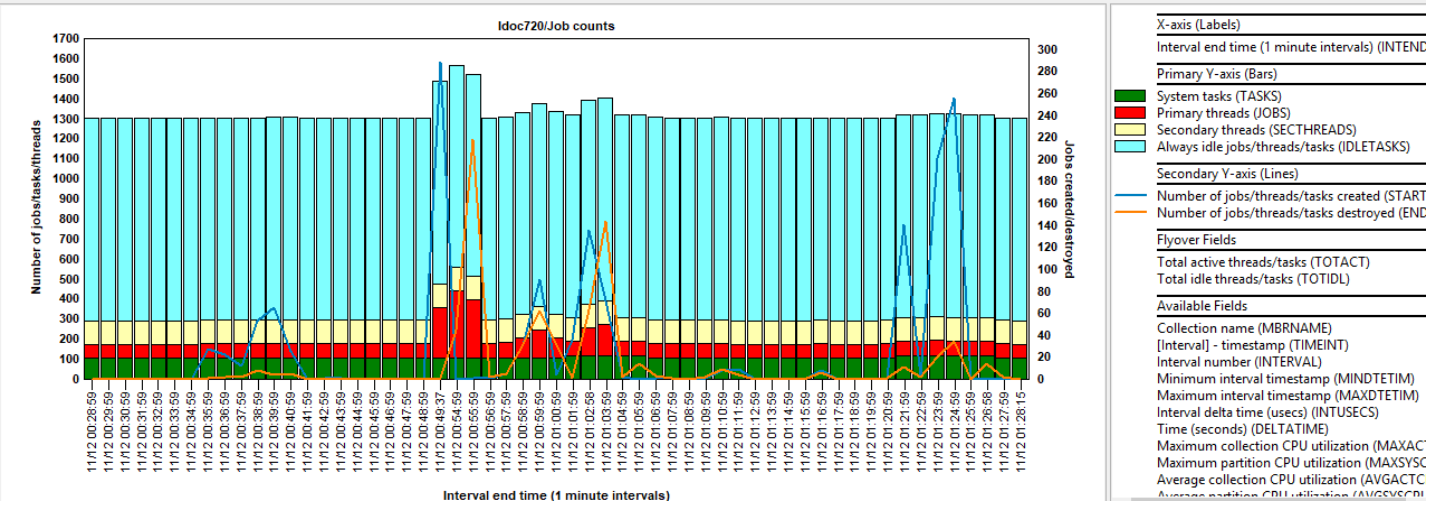
Please be aware that Job Watcher is a snapshot taker and frequently misses short-lived jobs/tasks/threads because they live and die in between snapshots. The numbers shown are always an estimate and are not actual results.

You can also drill down from these graphs to get job counts rankings graphs in various ways.

8.16.1.1 Job counts

This graph shows the total number of active tasks, processes (primary threads) and secondary threads over time. CPU utilization is shown on the 2nd Y-axis.

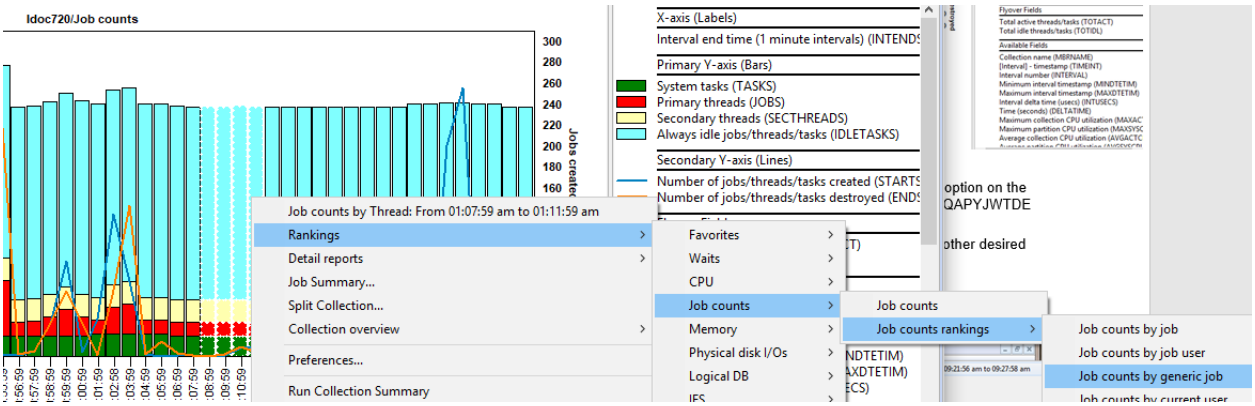
This graph in mosts cases will also display a 4th bucket called “Always idle jobs/threads/tasks”. These are jobs/tasks/threads on the system (we don’t know which kind) that never used CPU. Job Watcher doesn’t know what type of work it is because no CPU was used.



Job counts

The only way to avoid seeing this bucket is to start Job Watcher with the force 1st interval option on the Job Watcher definition. This causes all jobs/tasks/threads to have a record written in the QAPYJWTDE file on interval 1. This in some cases can take a long time.

Tip: Select a time period and right-click and drill down into Job counts by generic job (or other desired groupings) like this:

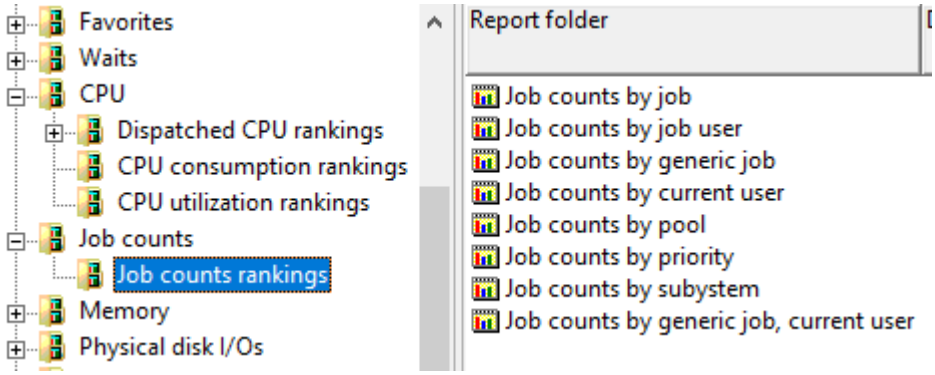


Example of drilling down into job counts by generic job

8.16.2 Job counts rankings

These graphs are used to rank the number of job/tasks/threads that exist on the system for the time period selected (if using a drill-down) or the entire collection.

Note: Not as many job groupings exist in this folder as in the previous ones.



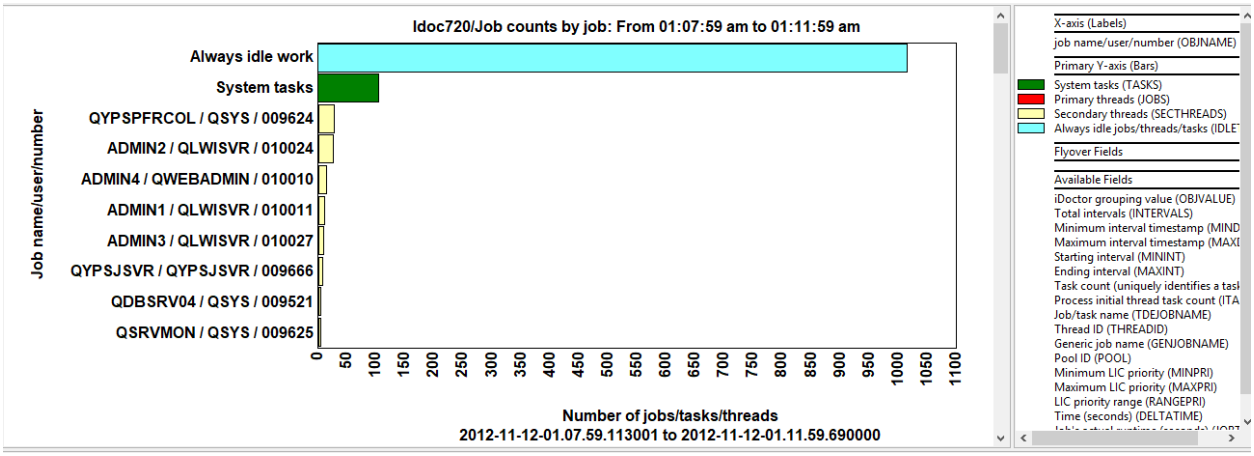
Job counts rankings

8.16.2.1 Job counts by job

This graph ranks the total number of job/tasks/thread by job.

Note: This will typically include the always idle bucket discussed in the previous section.

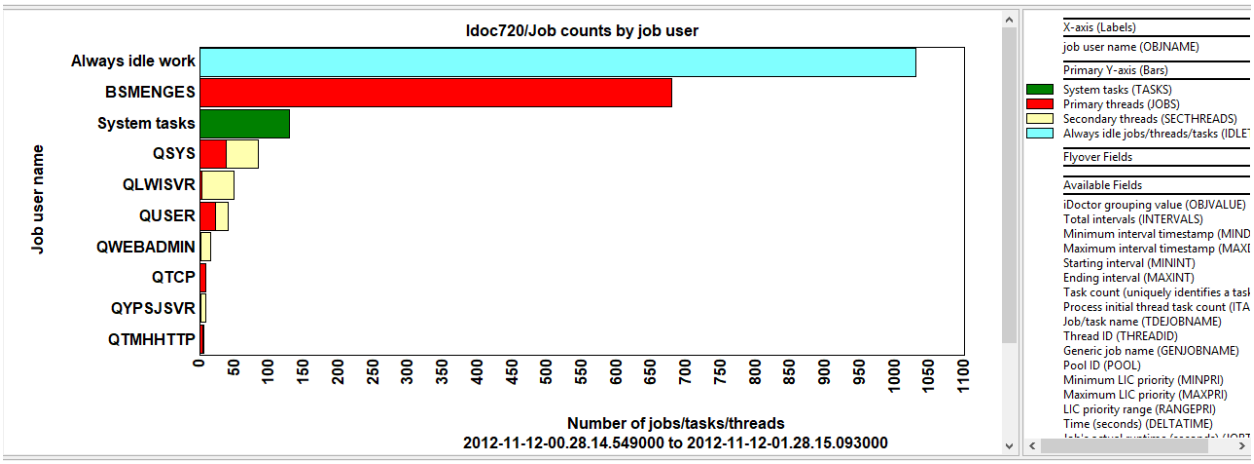
Because the graph groups the data by initial (primary thread) taskcount (ITASKCOUNT field), all system tasks are grouped together into a single System tasks entry in this graph. The rest of these jobs listed on this graph are primary threads along with the total secondary threads they contain.



Job counts by job

8.16.2.2 Job counts by job user

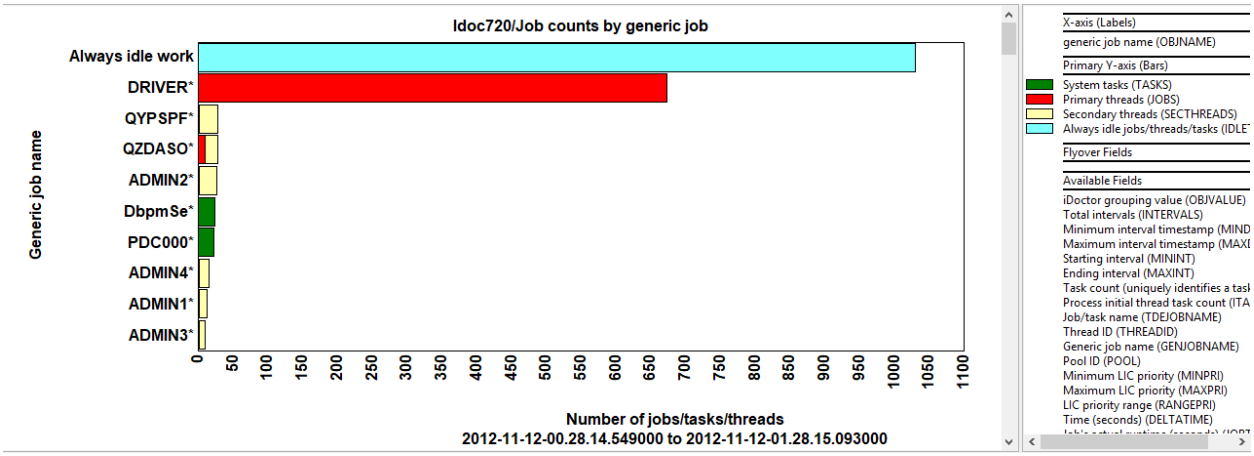
This graph shows the total job counts ranked by job user name.



Job counts by job user

8.16.2.3 Job counts by generic job

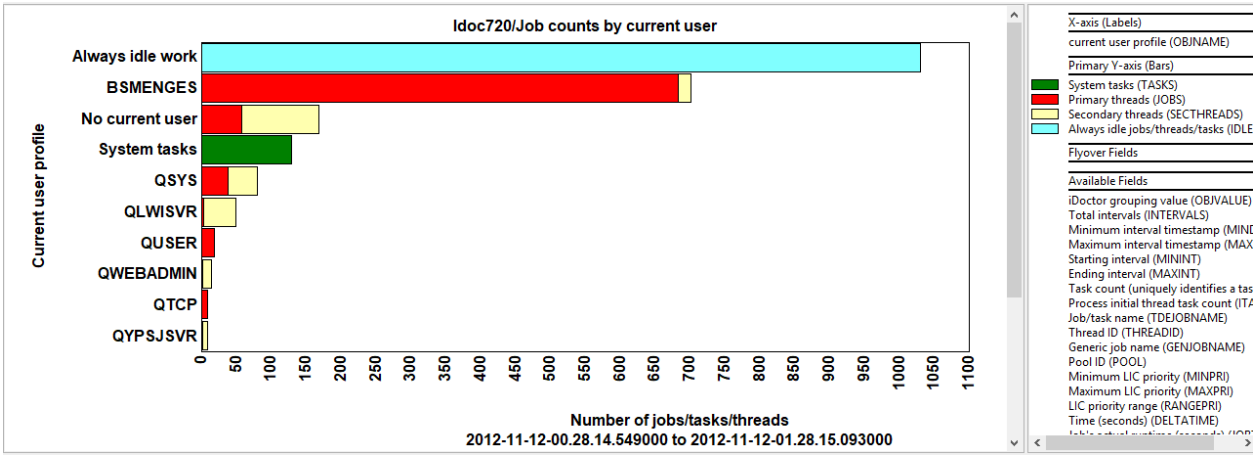
This graph shows the total job counts ranked by generic job name or generic task name.



Job counts by generic job

8.16.2.4 Job counts by current user

This graph shows the total job counts ranked by current user profile. It will include an entry for all system tasks and another row for “No current user.”

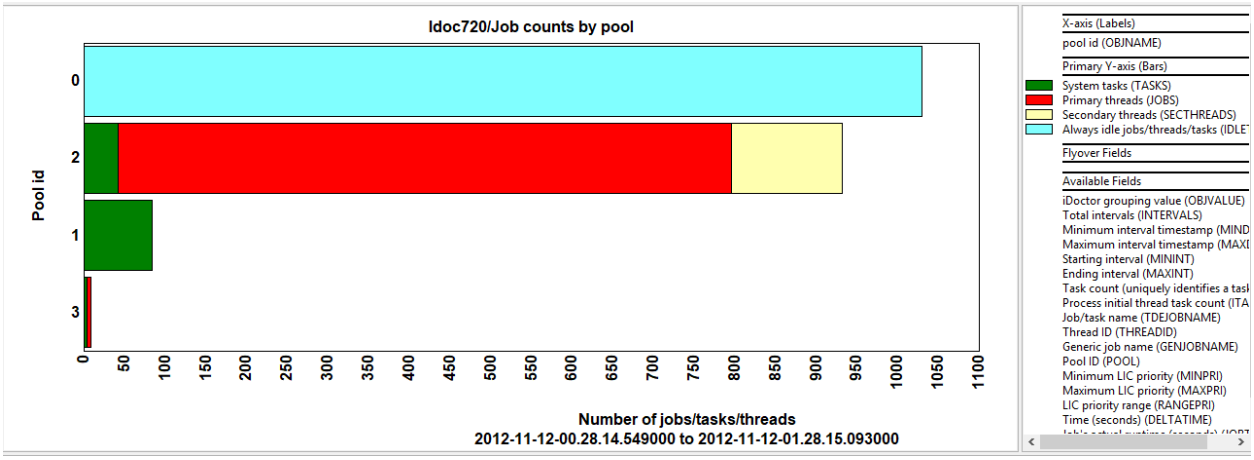


Job counts by current user

8.16.2.5 Job counts by pool

This graph shows the total job counts ranked by memory pool.

Note: Pool 0 is not a real memory pool, but this shows the total of additional jobs on the system

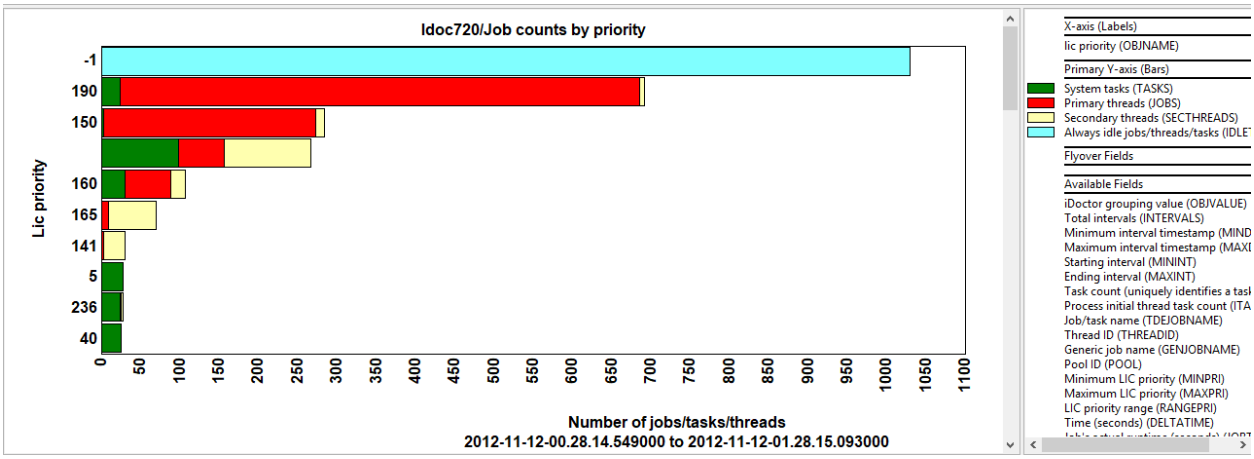


Job counts by pool

8.16.2.6 Job counts by priority

This graph shows the total job counts ranked by LIC priority. If the value is > 140, then subtract 140 to determine XPF priority.

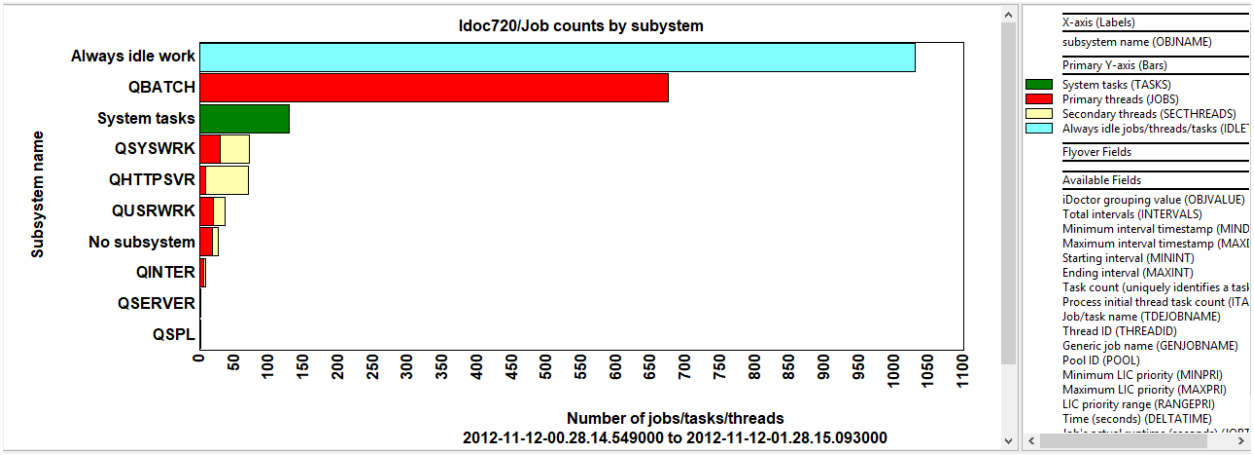
Note: -1 is not a valid LIC priority, but these are from the always idle bucket and the actual priority value is unknown!



Job counts by priority

8.16.2.7 Job counts by subsystem

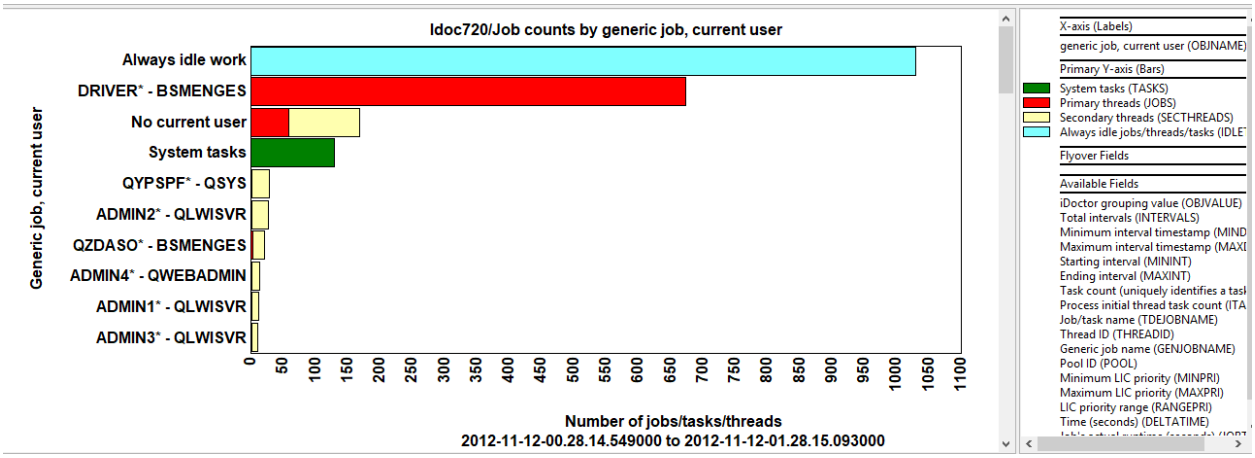
This graph shows the total job counts ranked by subsystem. This graph will include a System tasks bar that groups all the system task counts together. A “No subsystem” entry is also included in this graph.



Job counts by subsystem

8.16.2.8 Job counts by generic job, current user

This graph shows the total job counts ranked by generic job and current user combination. This graph will include a System tasks bar that groups all the system task counts together. A “No current user” entry is also included in this graph.

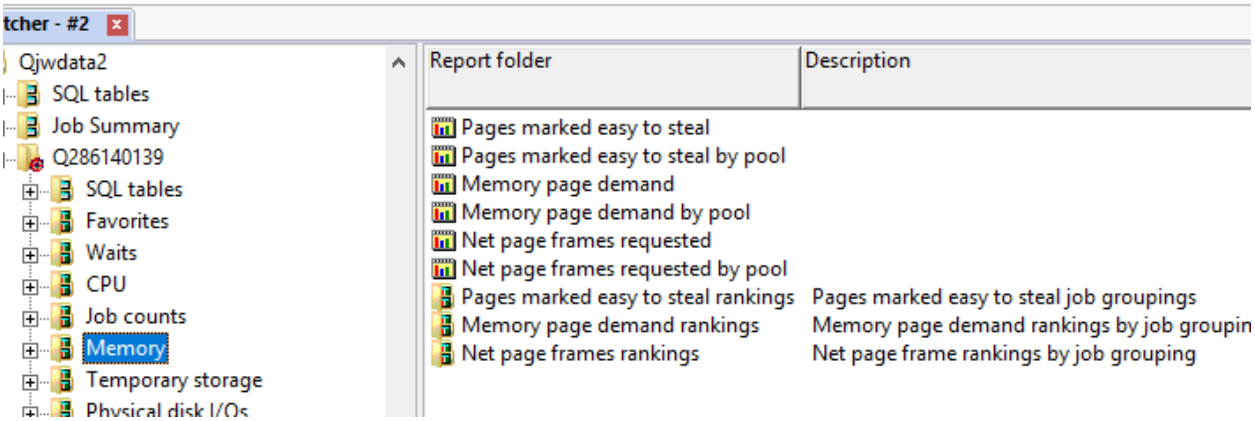


Job counts by generic job, current user

8.17 Memory

This folder contains graphs related to memory page usage in Job Watcher.

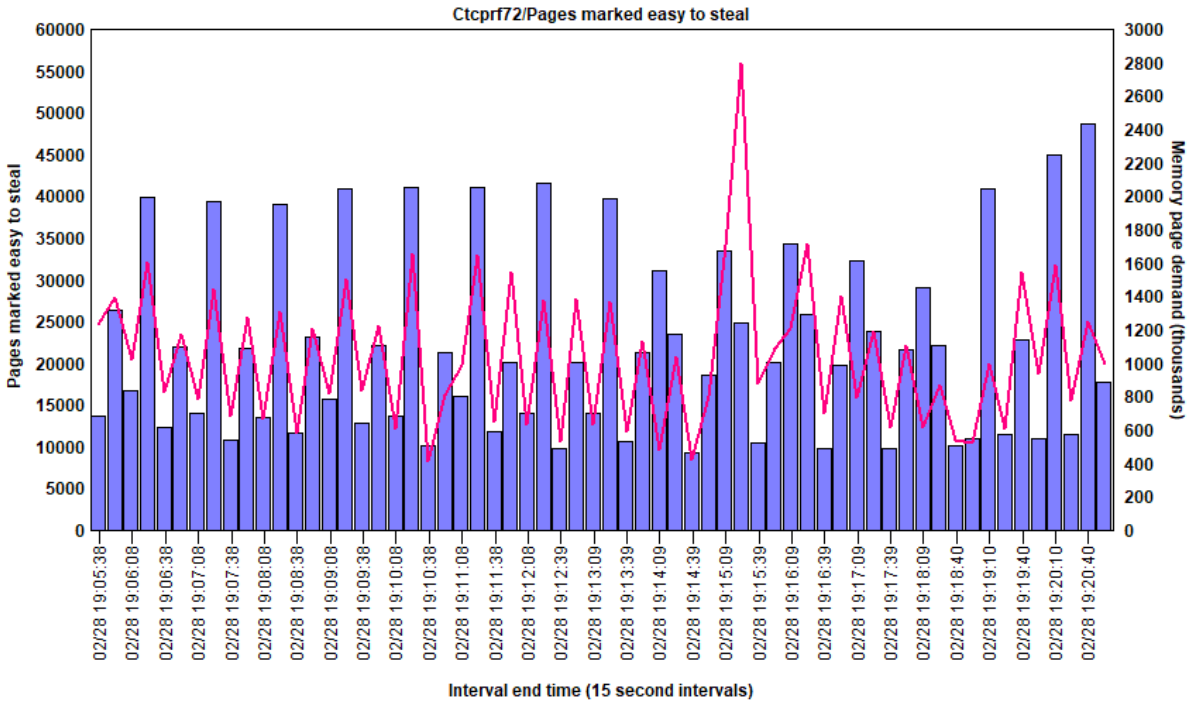
Tip: For graphs related to memory pool sizes and consumption you will need to use Collection Services Investigator.



Memory folder

8.17.1 Pages marked easy to steal (7.2+)

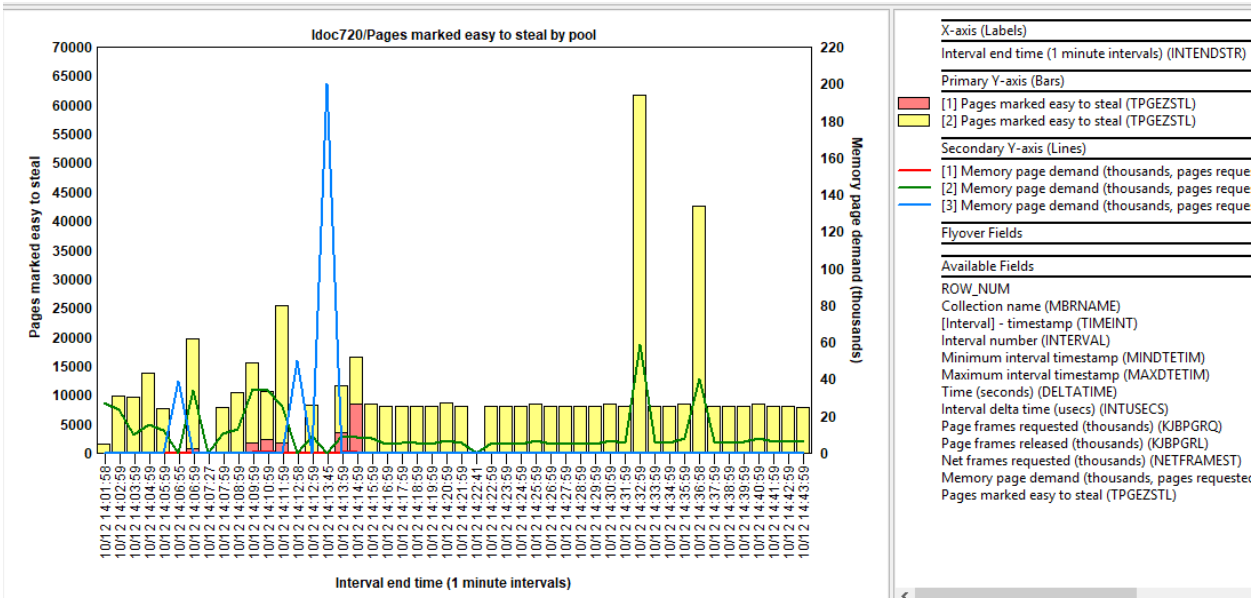
This graph displays the total pages marked easy to steal for all jobs on the system on the Y1-axis with the memory page demand counts on the Y2-axis.



Pages marked easy to steal

8.17.2 Pages marked easy to steal by pool (7.2+)

This graph displays the total pages marked easy to steal for all jobs on the system on the Y1-axis with the memory page demand counts on the Y2-axis. Both values are segregated by memory pool and will use a different color for each metric/pool.

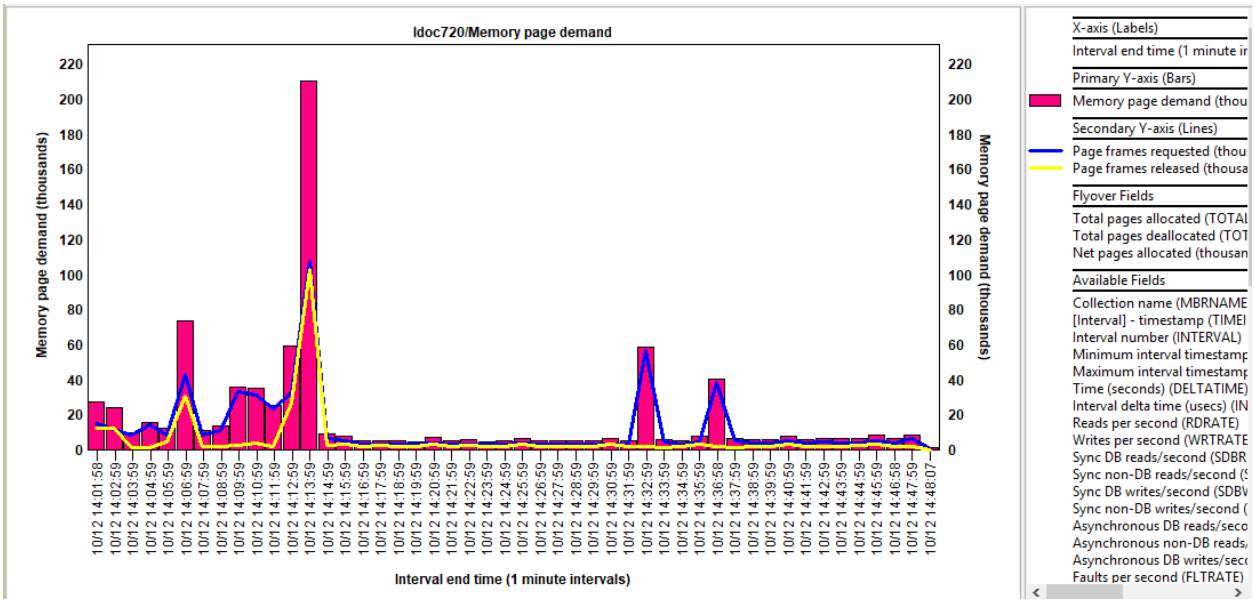


Pages marked easy to steal by pool

8.17.3 Memory page demand

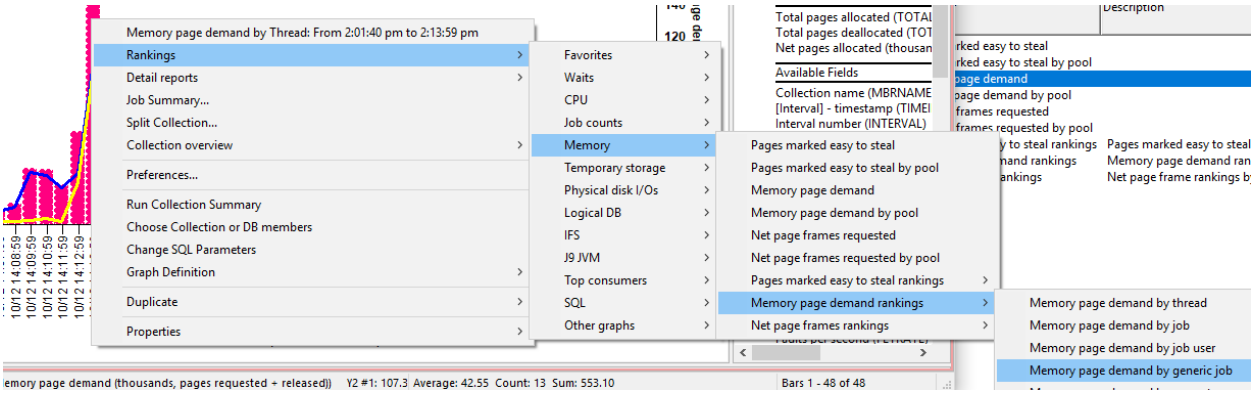
This graph displays the memory pages requested and released which together gives a sense of how much memory is being utilized by the jobs captured by Job Watcher.

The bars display the total of the page frames requested and released. The 2nd Y-axis displays the 2 values as separate lines instead.

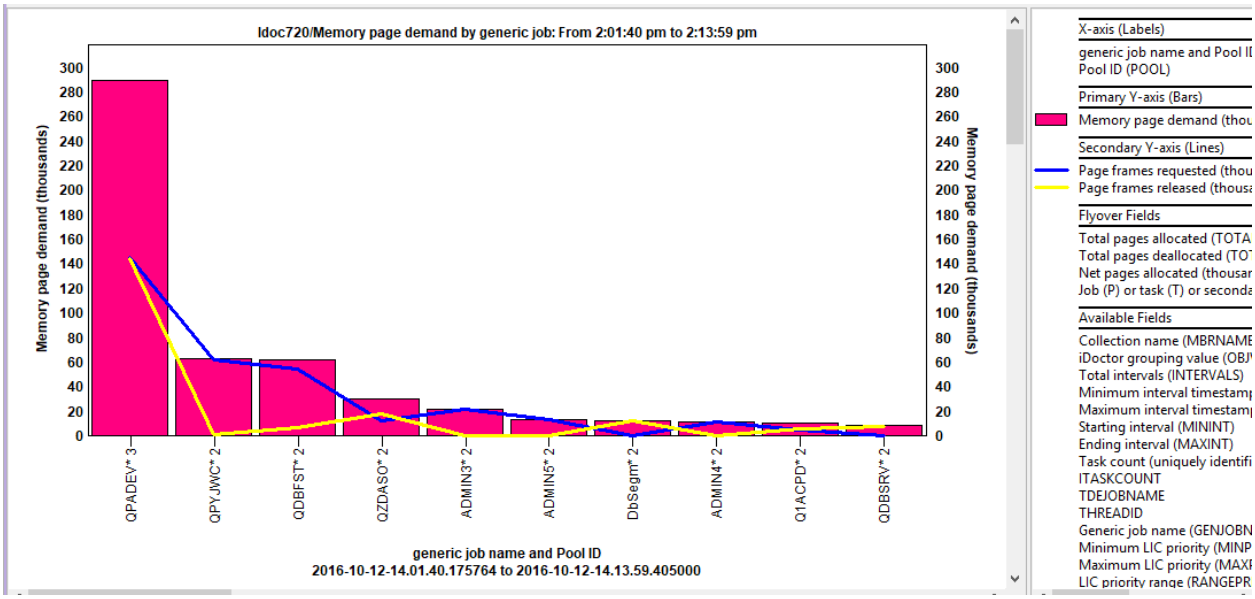


Memory page demand

Below is an example of how to drill down showing the generic jobs having the most memory page demand:



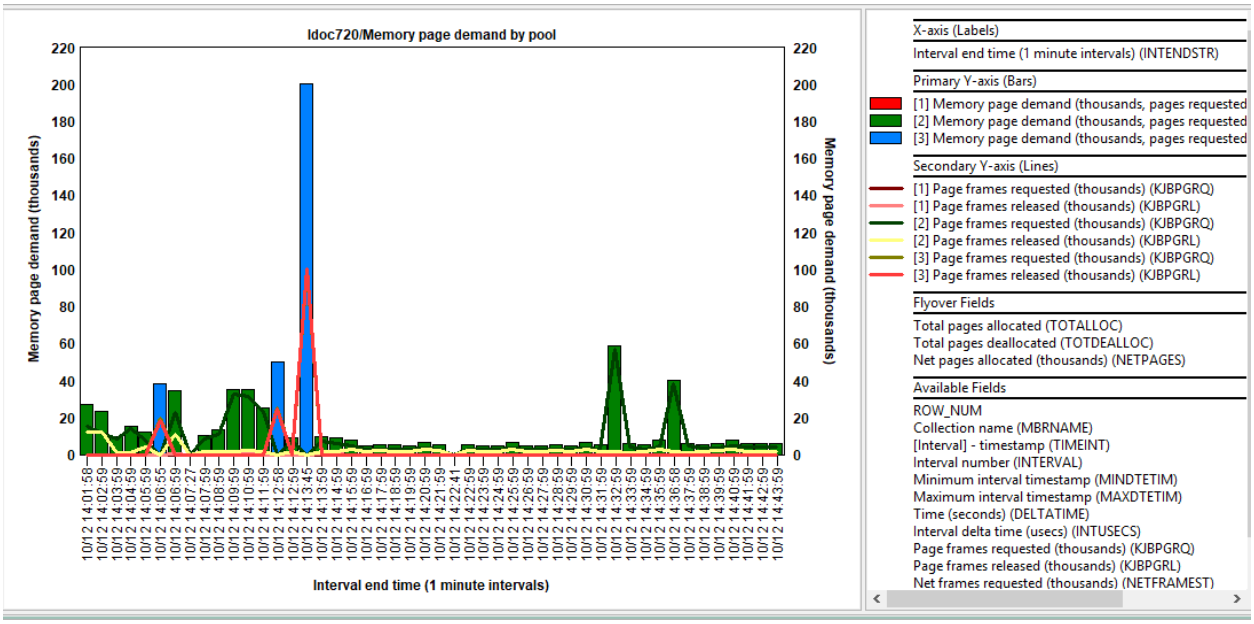
Memory page demand rankings drill down



Memory page demand by generic job

8.17.4 Memory page demand by pool

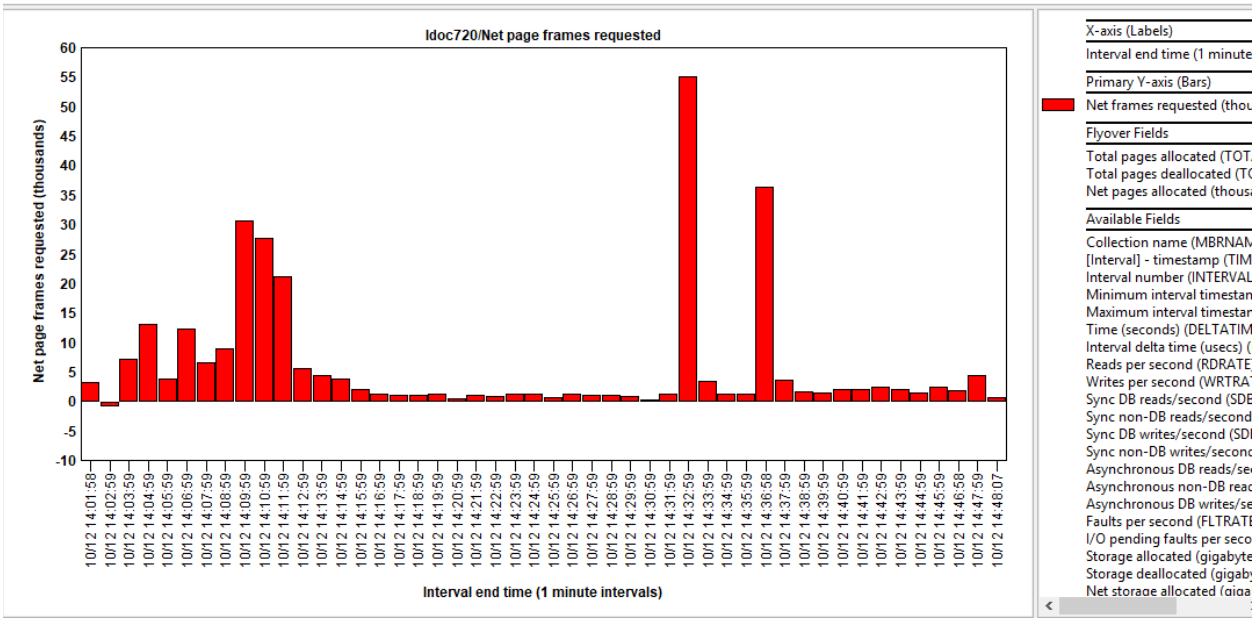
This graph is the same as the previous graph except the data is segregated by memory pool number.



Memory page demand by pool

8.17.5 Net page frames requested (6.1+)

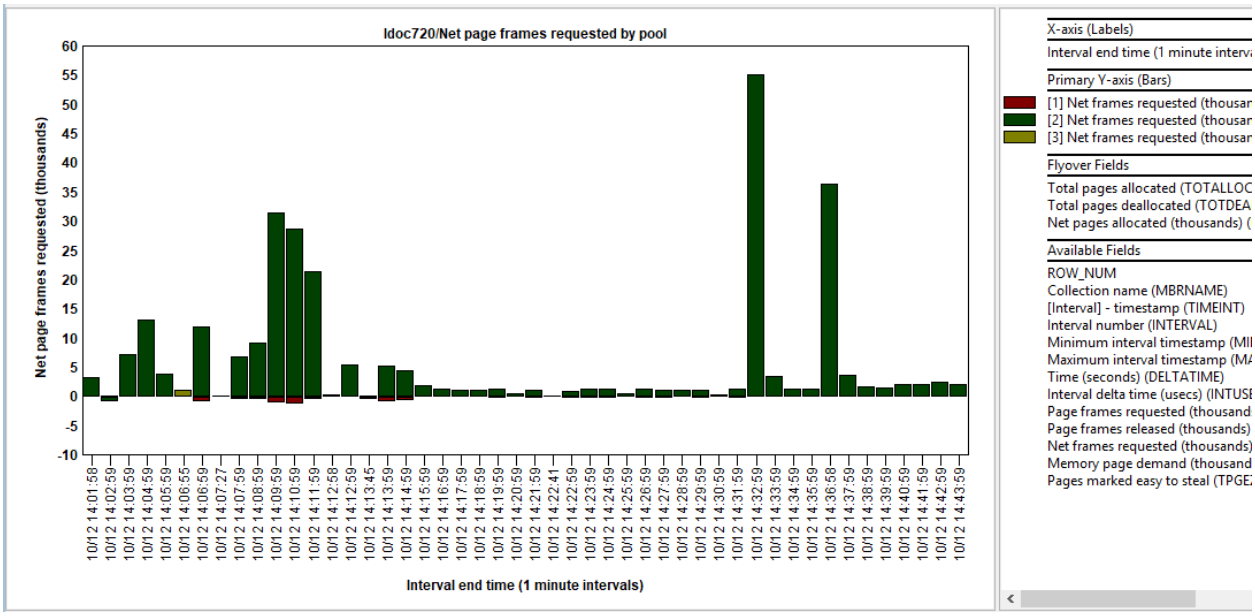
This graph is like the memory page demand graph except it just shows the Net memory pages requested. A negative value indicates that more pages were released than requested in that time interval.



Net page frames requested

8.17.6 Net page frames requested by pool (6.1+)

This graph is the same as the previous one except the data is segregated by memory pool.

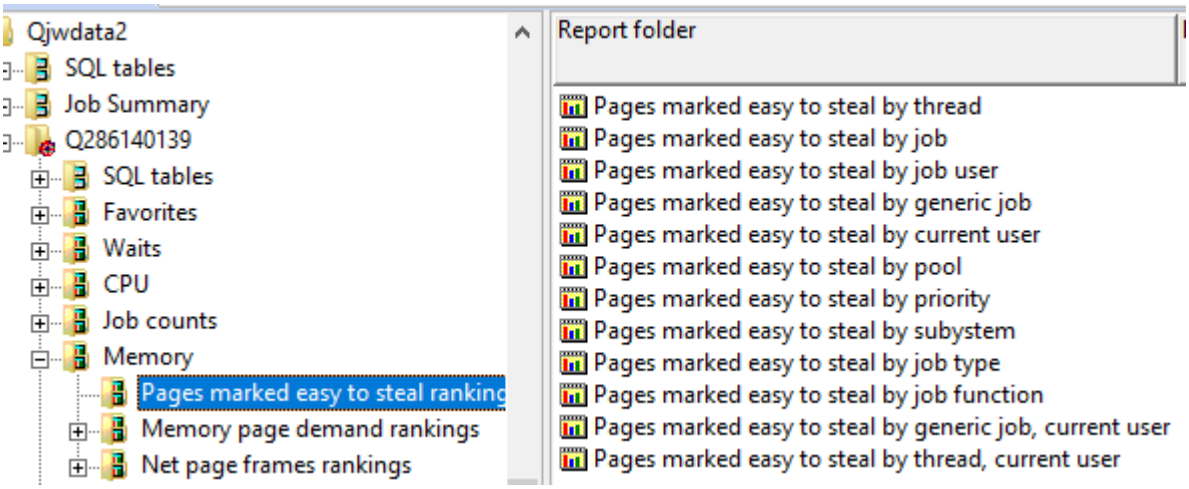


Net page frames requested by pool

8.17.7 Pages marked easy to steal rankings

These graphs rank the pages marked easy to steal metric in various ways. The memory pool on these graphs is listed next to the job grouping value.

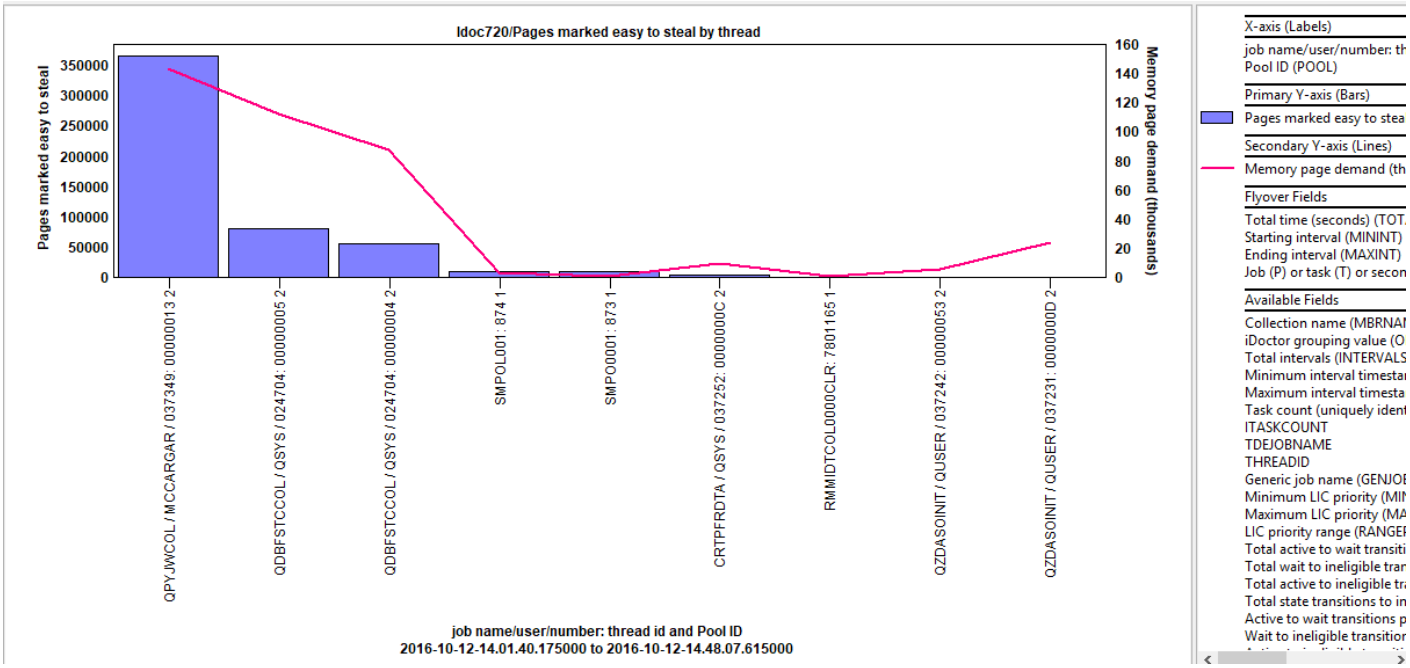
Tip: In the case that multiple pools exist, the min and max pool numbers within the grouping will both be listed after the job grouping value. In some cases, there could be more than 2 pools and those are not listed.



Memory -> Pages marked easy to steal rankings

8.17.7.1 Pages marked easy to steal by thread

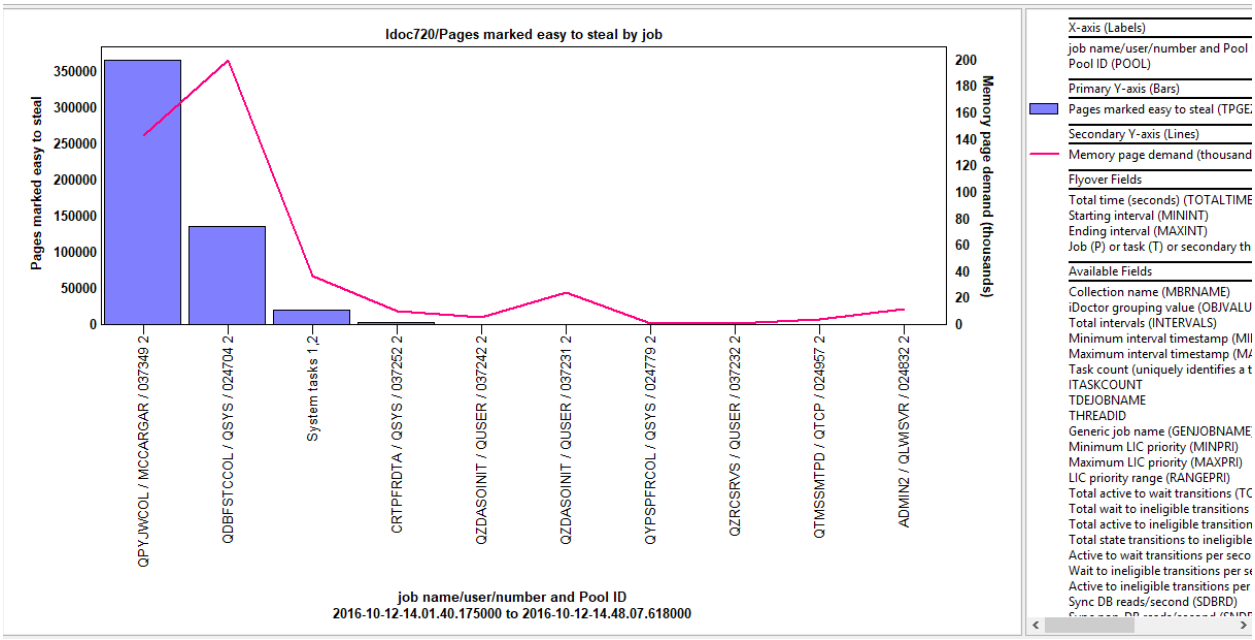
This graph ranks the pages marked easy to steal by thread.



Pages marked easy to steal by thread

8.17.7.2 Pages marked easy to steal by job

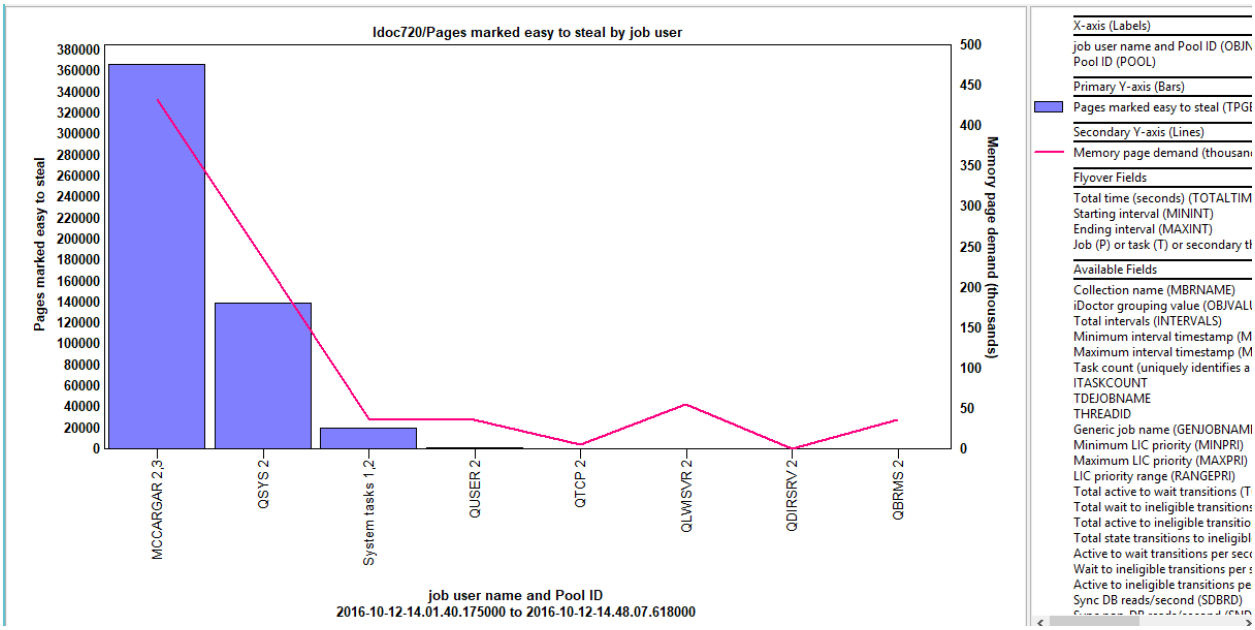
This graph ranks the pages marked easy to steal by job.



Pages marked easy to steal by job

8.17.7.3 Pages marked easy to steal by job user

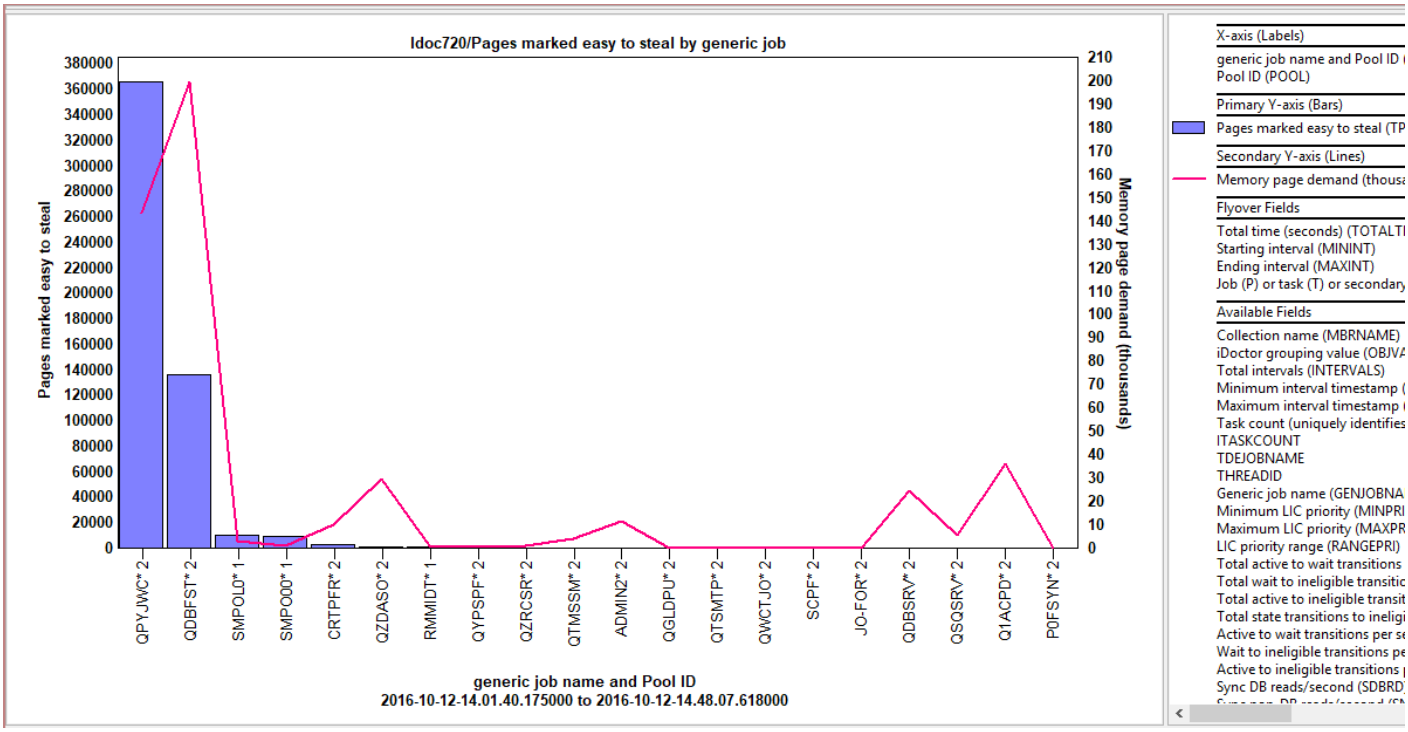
This graph ranks the pages marked easy to steal by job user.



Pages marked easy to steal by job user

8.17.7.4 Pages marked easy to steal by generic job

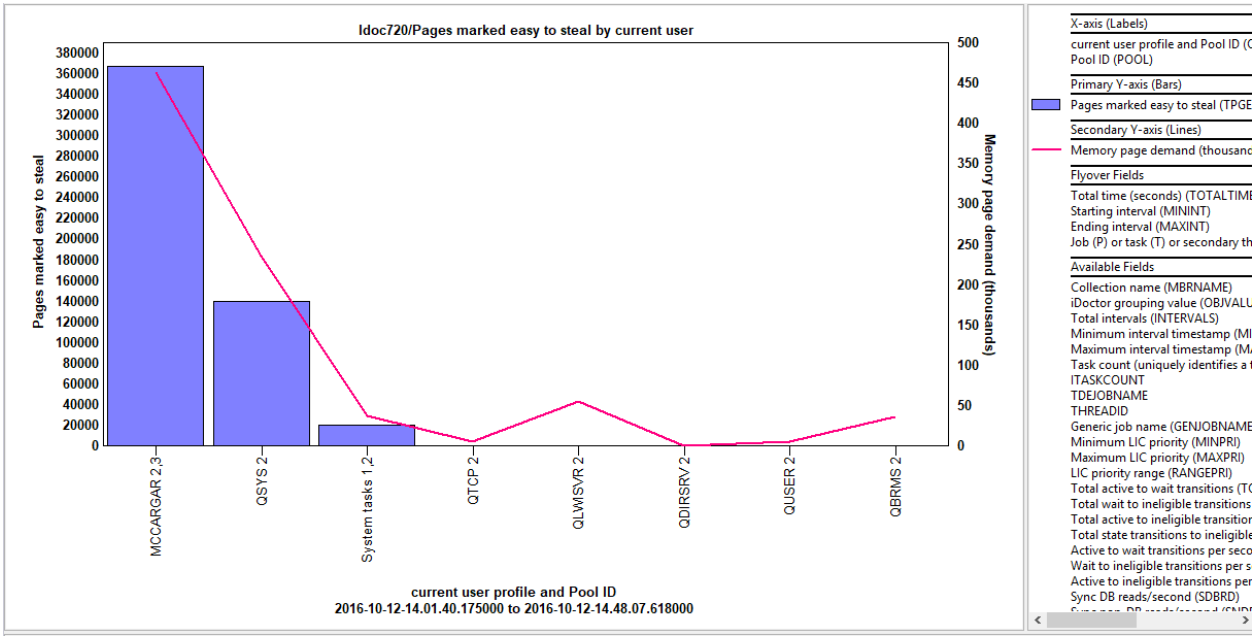
This graph ranks the pages marked easy to steal by generic job.



Pages marked easy to steal by generic job

8.17.7.5 Pages marked easy so steal by current user

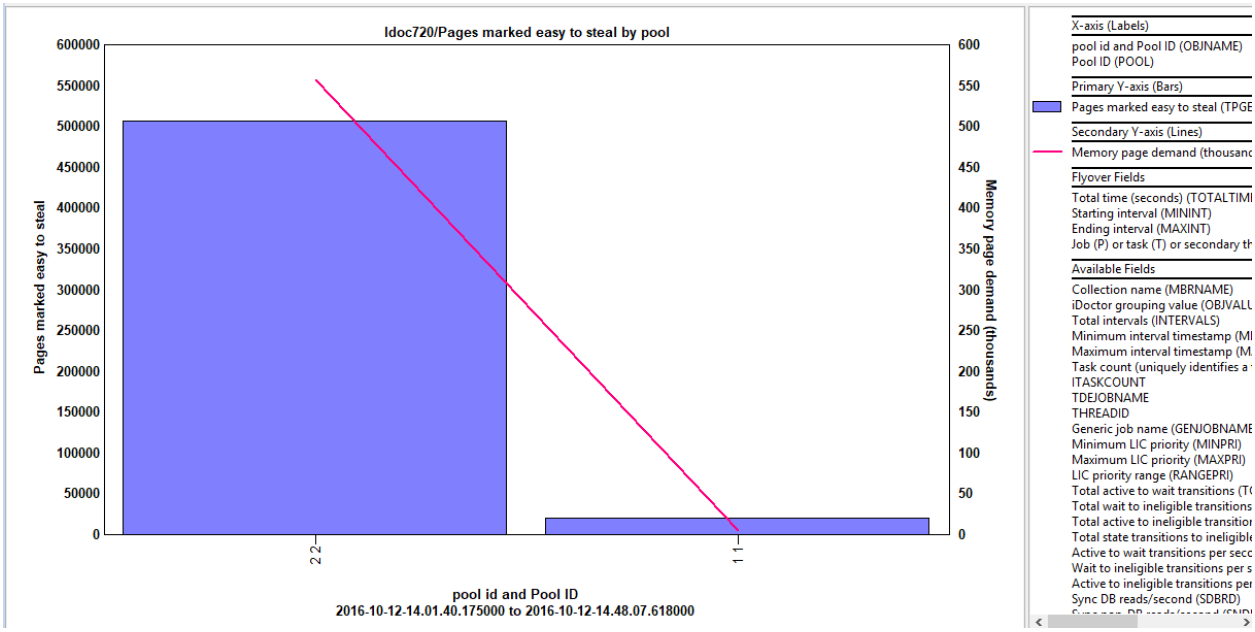
This graph ranks the pages marked easy to steal by current user.



Pages marked easy so steal by current user

8.17.7.6 Pages marked easy so steal by pool

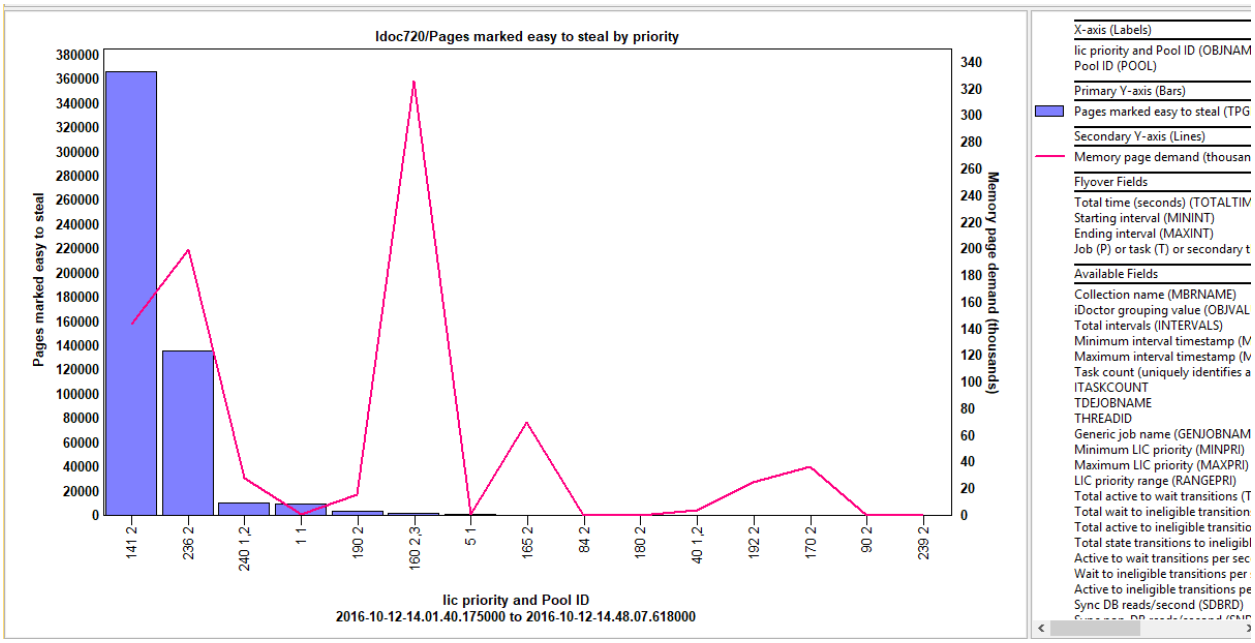
This graph ranks the pages marked easy to steal by pool.



Pages marked easy so steal by pool

8.17.7.7 Pages marked easy so steal by priority

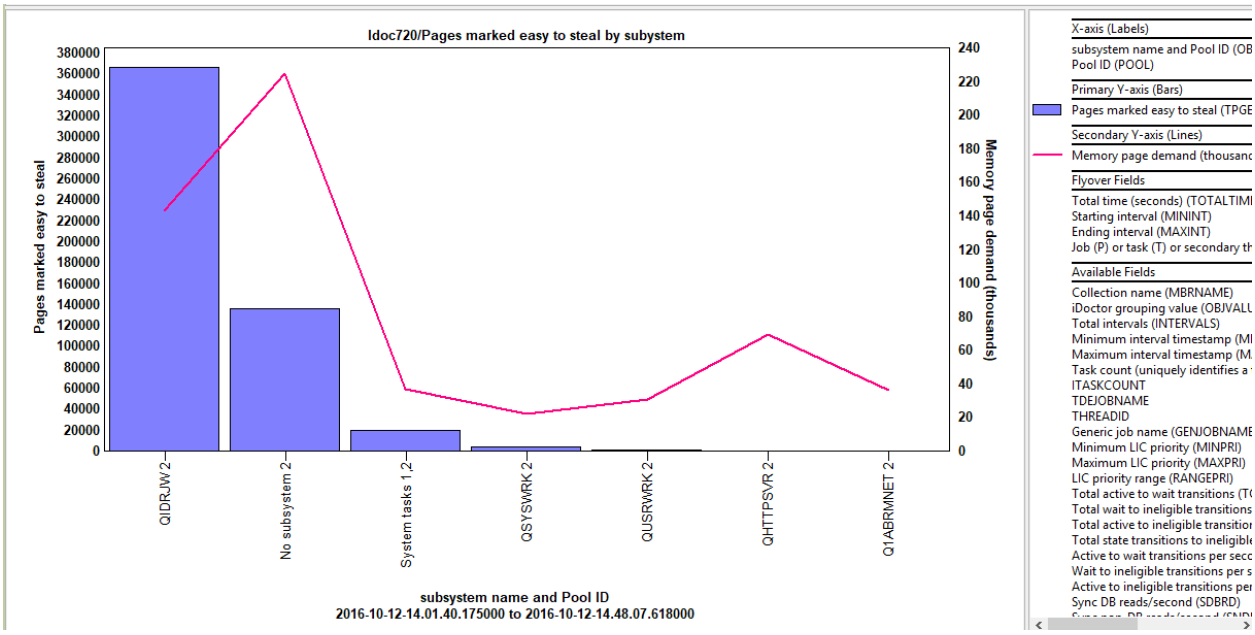
This graph ranks the pages marked easy to steal by priority.



Pages marked easy so steal by priority

8.17.7.8 Pages marked easy so steal by subsystem

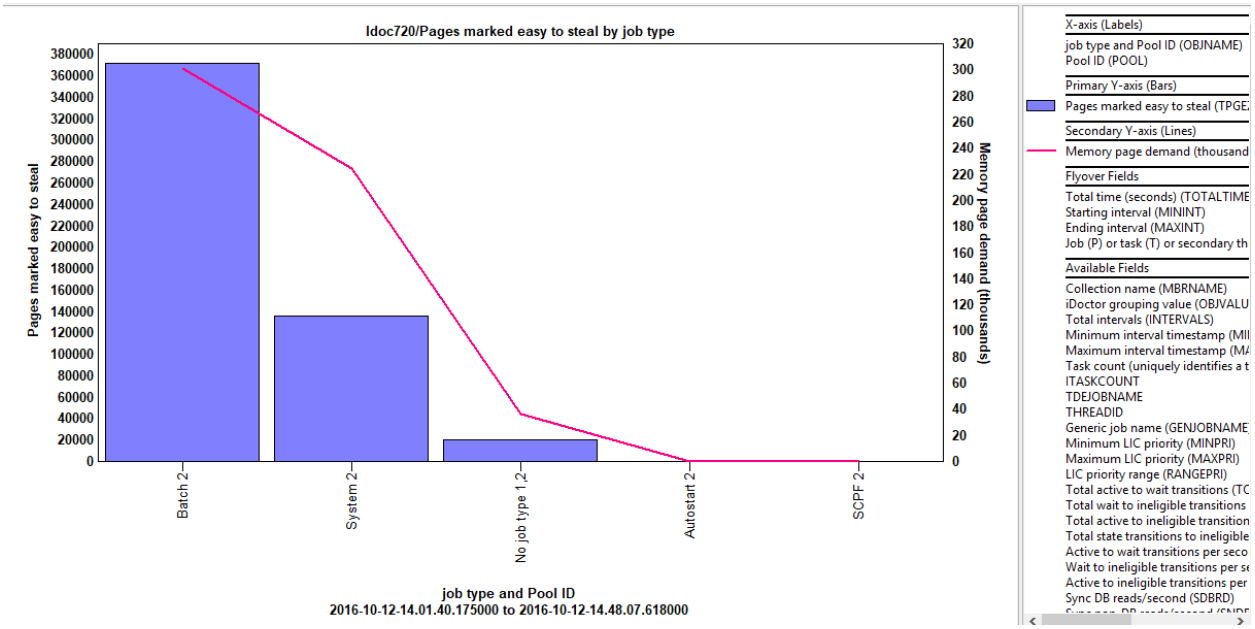
This graph ranks the pages marked easy to steal by subsystem.



Pages marked easy so steal by subsystem

8.17.7.9 Pages marked easy so steal by job type

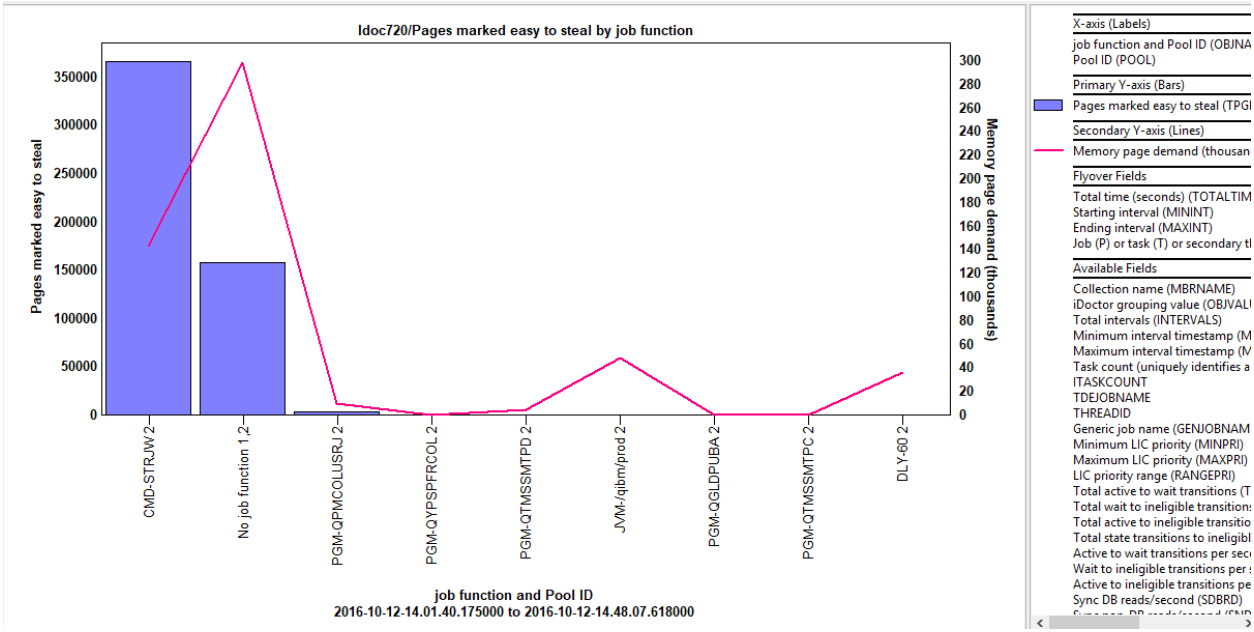
This graph ranks the pages marked easy to steal by job type.



Pages marked easy so steal by job type

8.17.7.10 Pages marked easy so steal by job function

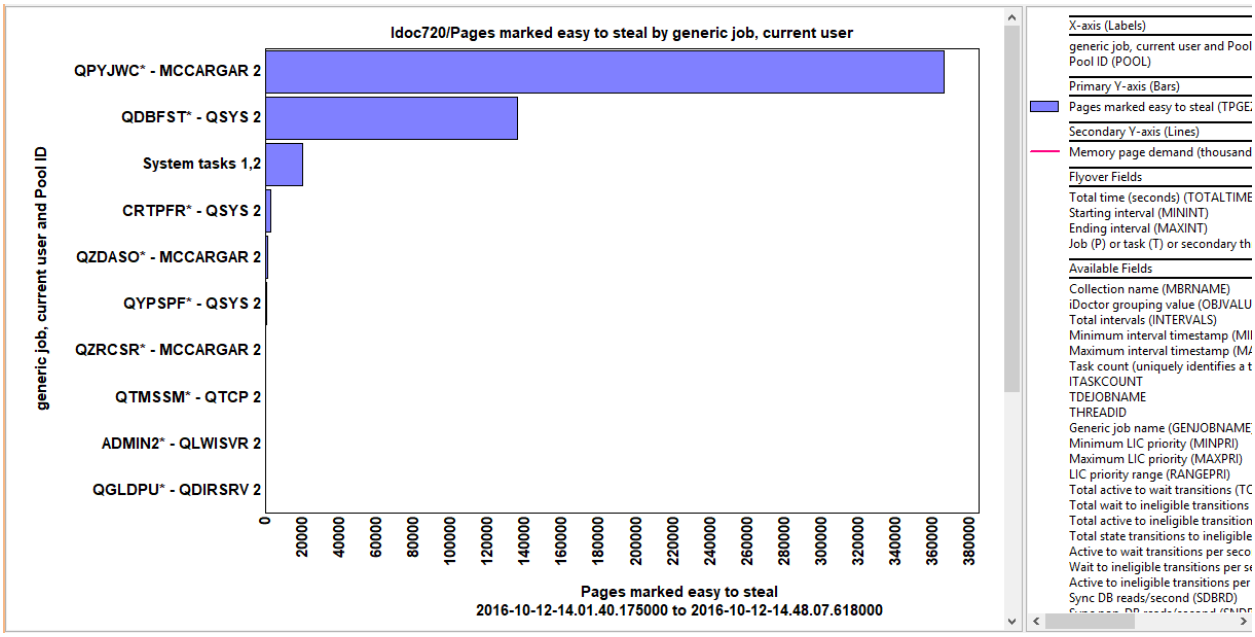
This graph ranks the pages marked easy to steal by job function.



Pages marked easy so steal by job function

8.17.7.11 Pages marked easy so steal by generic job, current user

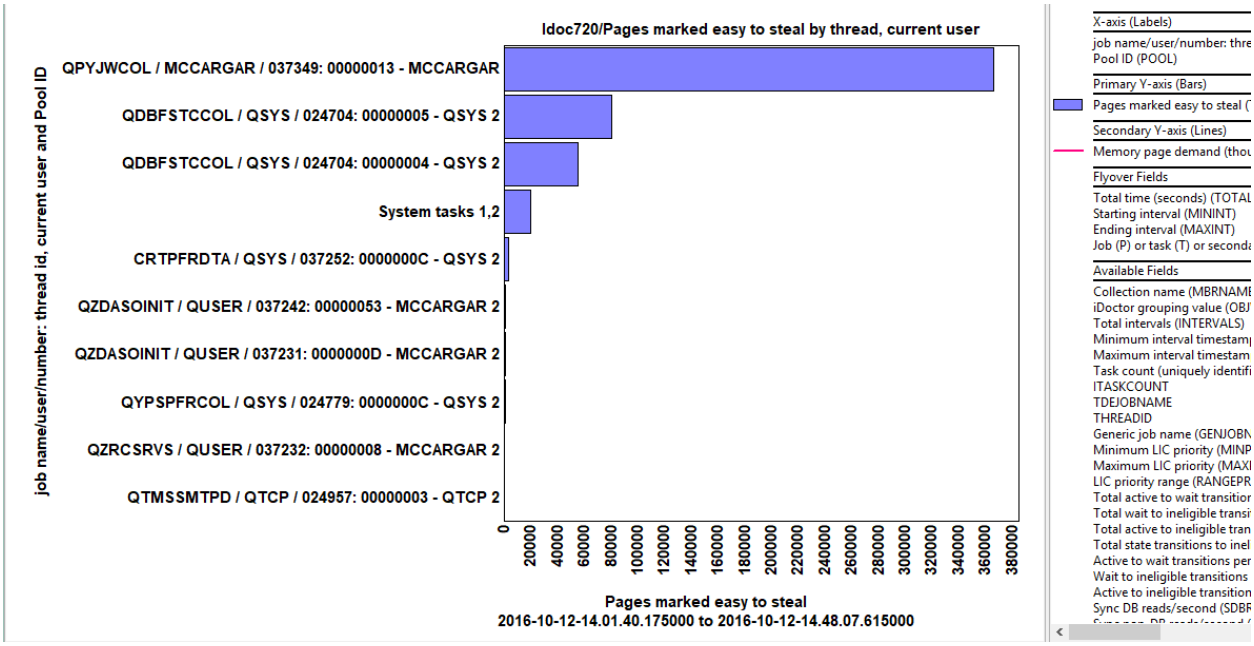
This graph ranks the pages marked easy to steal by generic job, current user.



Pages marked easy so steal by generic job, current user

8.17.7.12 Pages marked easy so steal by thread, current user

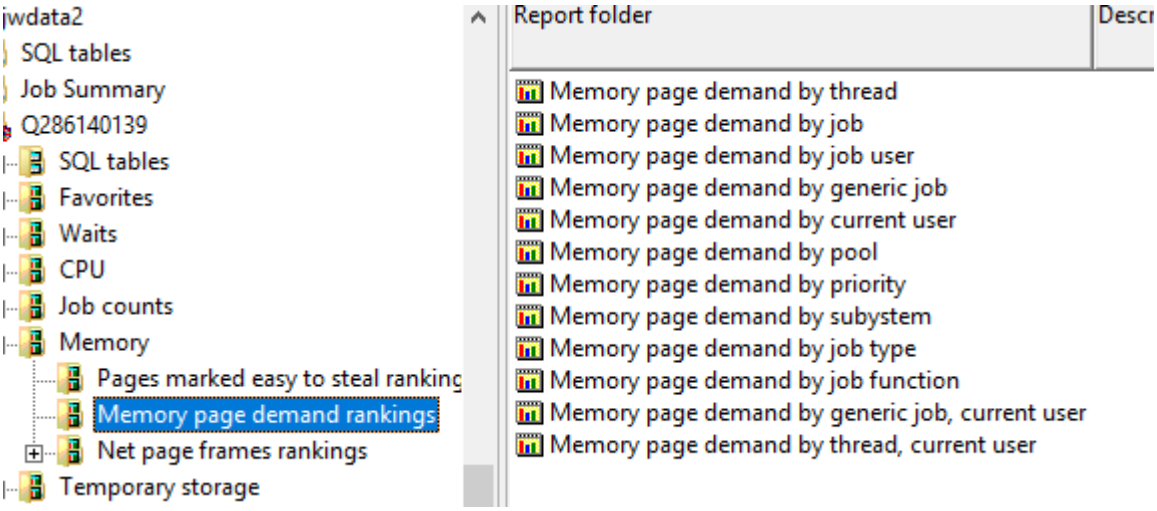
This graph ranks the pages marked easy to steal by thread, current user.



Pages marked easy so steal by thread, current user

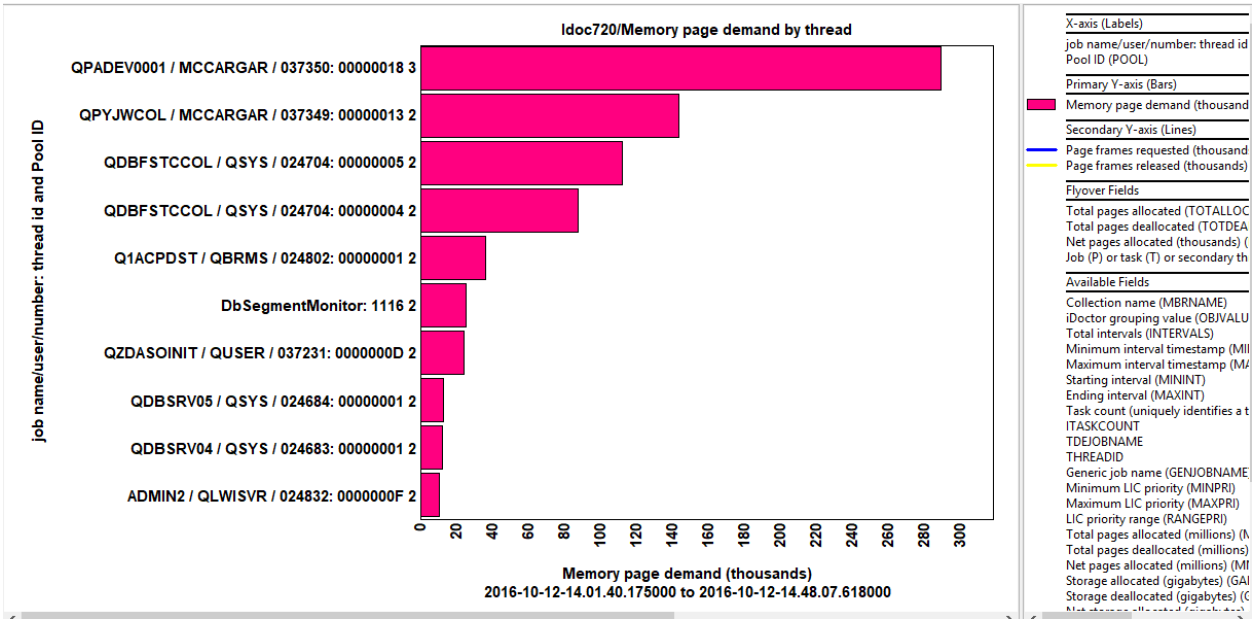
8.17.8 Memory page demand rankings

These graphs rank the memory page demand in various ways. The memory pool on these graphs is listed next to the job grouping value.



Memory -> Memory page demand rankings

These graphs are very similar to the Pages marked easy to steal rankings. See that section for more details.

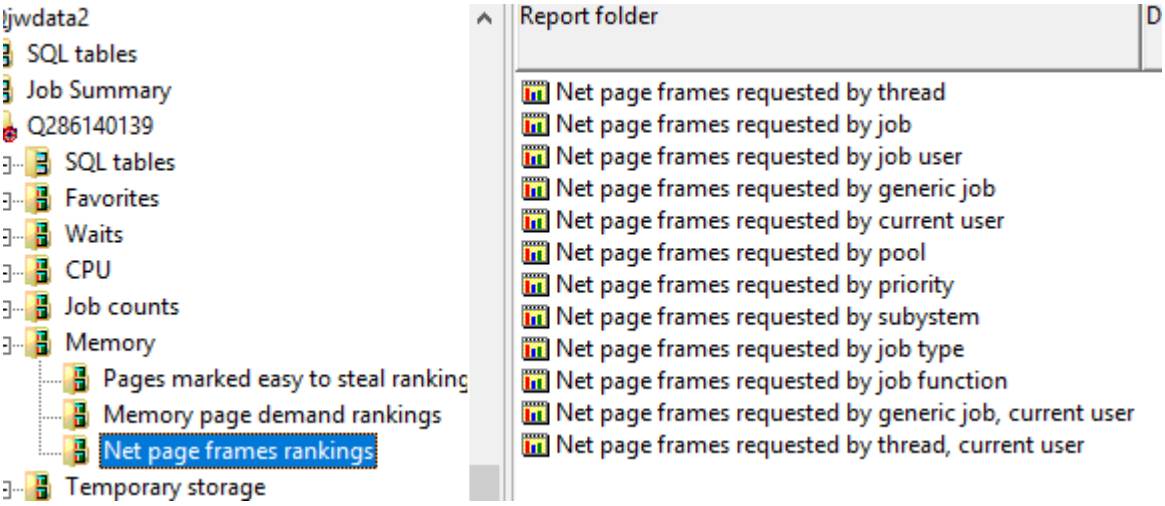


Memory page demand by thread

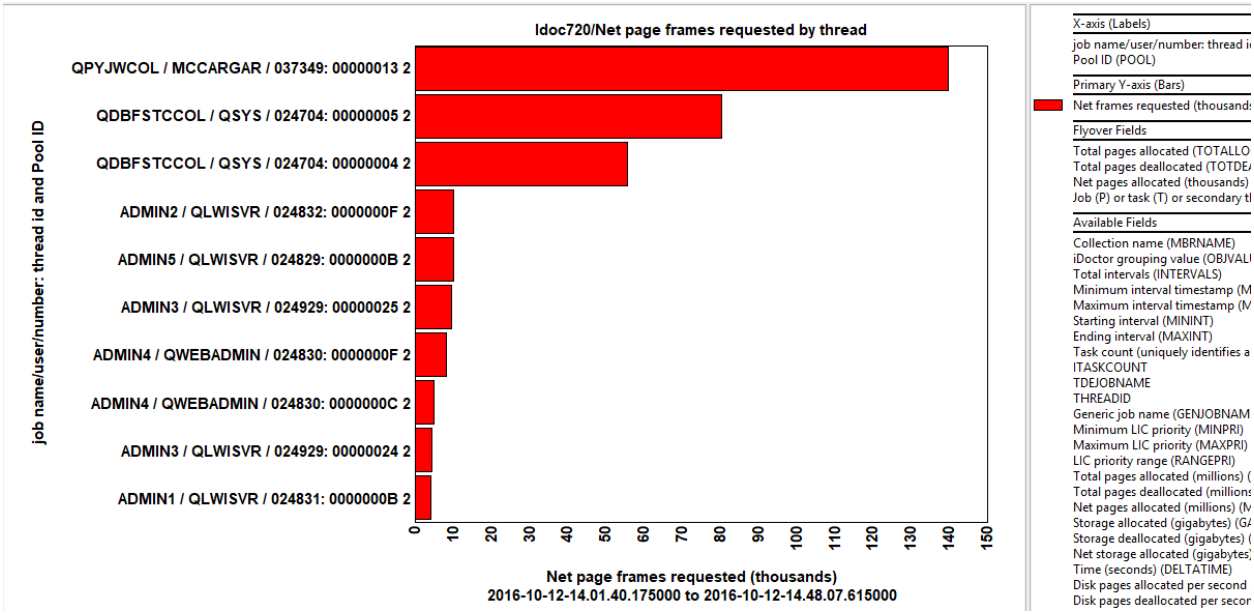
8.17.9 Net page frames requested rankings

These graphs rank the net page frames requested in various ways. The memory pool on these graphs is listed next to the job grouping value.

These graphs are very similar to the Pages marked easy to steal rankings. See that section for more details.



Net page frames requested rankings



Net page frames requested by thread

8.18 Temporary storage (7.2+)

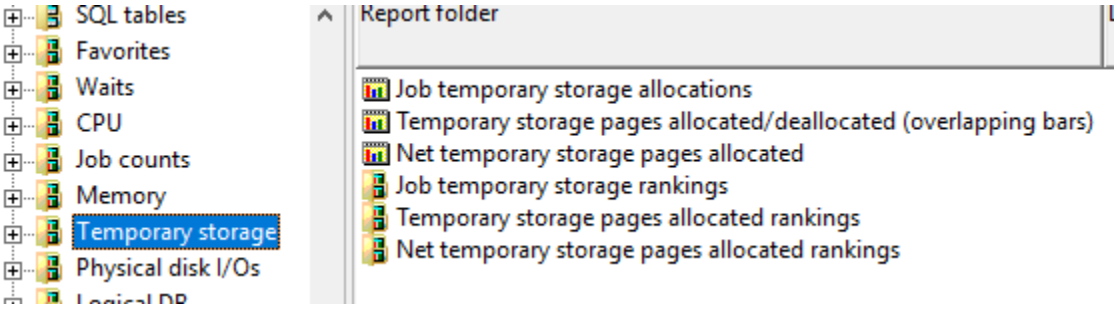
These graphs show the consumption of temporary storage used by jobs captured by Job Watcher.

Note: These graphs require the Collection summary analysis to be ran in order to appear!

Keep in mind that since Job Watcher may miss and not capture very short-lived jobs. Jobs that lived and died within a Job Watcher time interval are not included!

The only way to partially mitigate this is to decrease the Job Watcher interval size to capture more of these short-lived jobs.

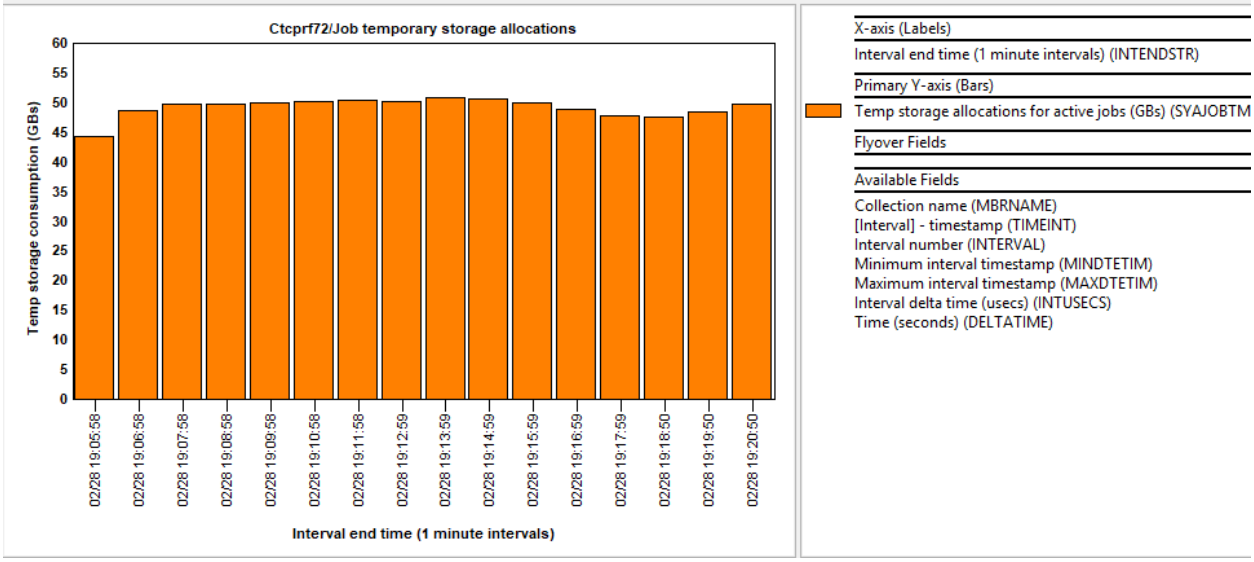
Note: Because the total disk space of the system ASP within the Job Watcher metrics is not known it is not possible to give temporary storage as a percentage of the system ASP.



Temporary storage

8.18.1 Job temporary storage allocations

This graph adds up the current temporary storage allocations across all jobs on the system and displays it over time. Please note that this total may likely not include 100% percent of all jobs because jobs that never used CPU are not included typically in Job Watcher (unless the force 1st interval option is used on the JW definition) and jobs that are short-lived are not included either.



Job temporary storage allocations

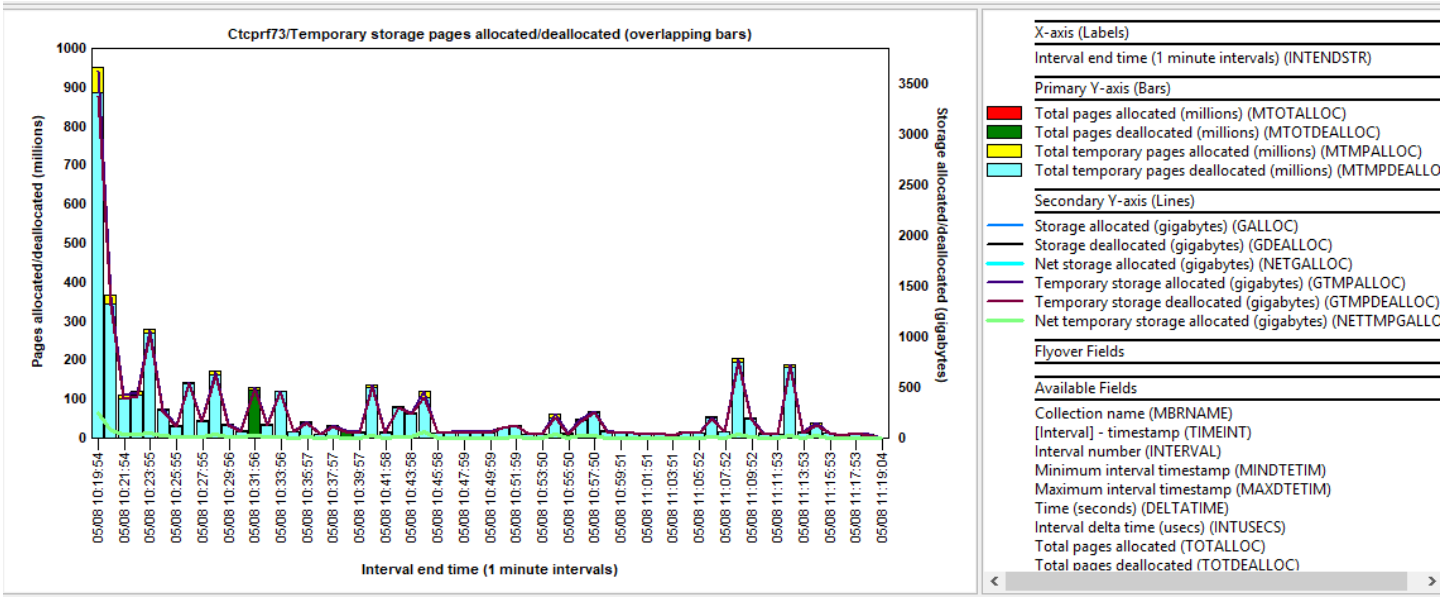
8.18.2 Temporary storage pages allocated/deallocated - overlapping bars

This graph displays 4 metrics, but they are overlapping with the highest values shown at the top of each bar. The smaller values will be displayed at the bottom. The metrics are:

1) Total pages allocated (millions)
2) Total pages deallocated (millions)
3) Total temporary pages allocated (millions)
4) Total temporary pages deallocated (millions)

Note: The metrics shown on these graphs are the total **changed** values within each time interval. They do NOT include totals for any pages left allocated from the past.

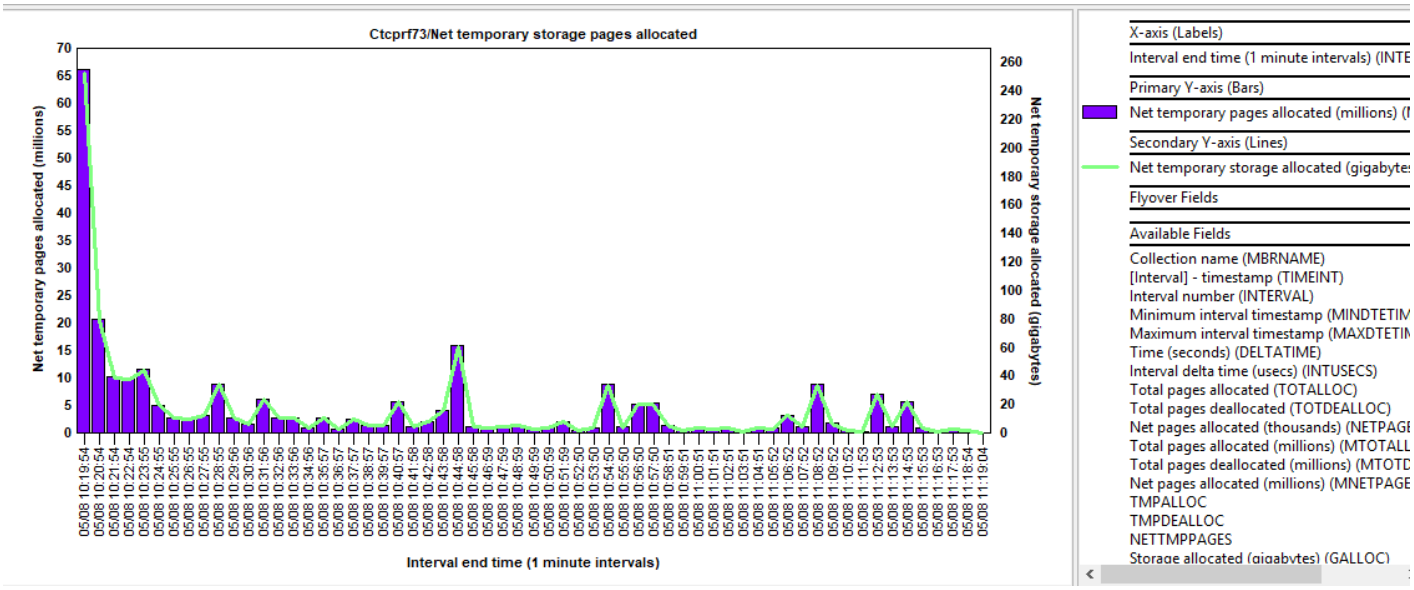
The Y2-axis displays the total storage allocations/deallocations changes for these 4K pages added up over time (in gigabytes). It also displays the net storage allocated for all pages or just the temp storage related pages.



Temporary storage pages allocated/deallocated - overlapping bars

8.18.3 Net temporary storage pages allocated

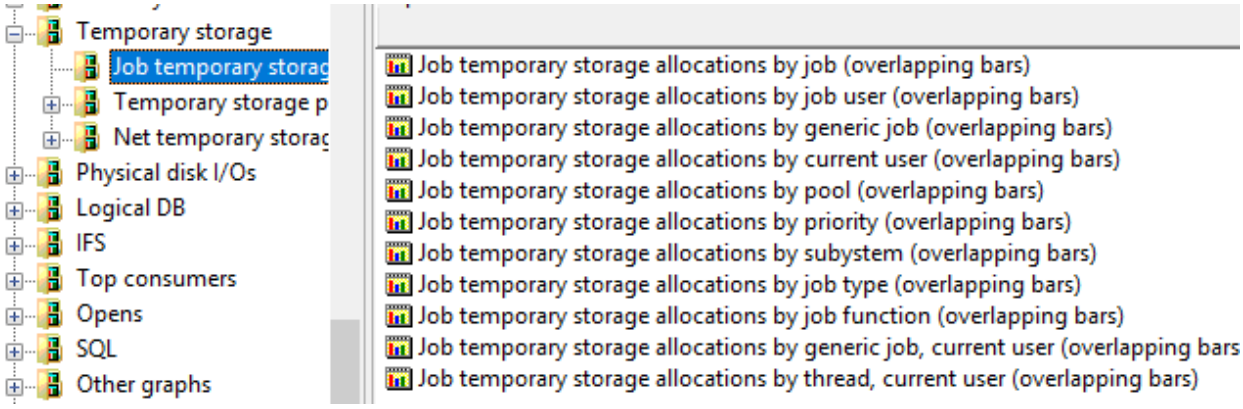
This graph displays the net temporary storage pages allocated over time. It includes the net storage size (in gigabytes) as well on the Y2-axis.



Net temporary storage pages allocated

8.18.4 Job temporary storage pages rankings

These graphs rank the job-based statistics for temporary storage consumption either for the entire collection or as a drill-down from a selection on the overview charts within this folder. These statistics are based on job-based metrics found in QAPYJWPRC and do not include any system tasks.



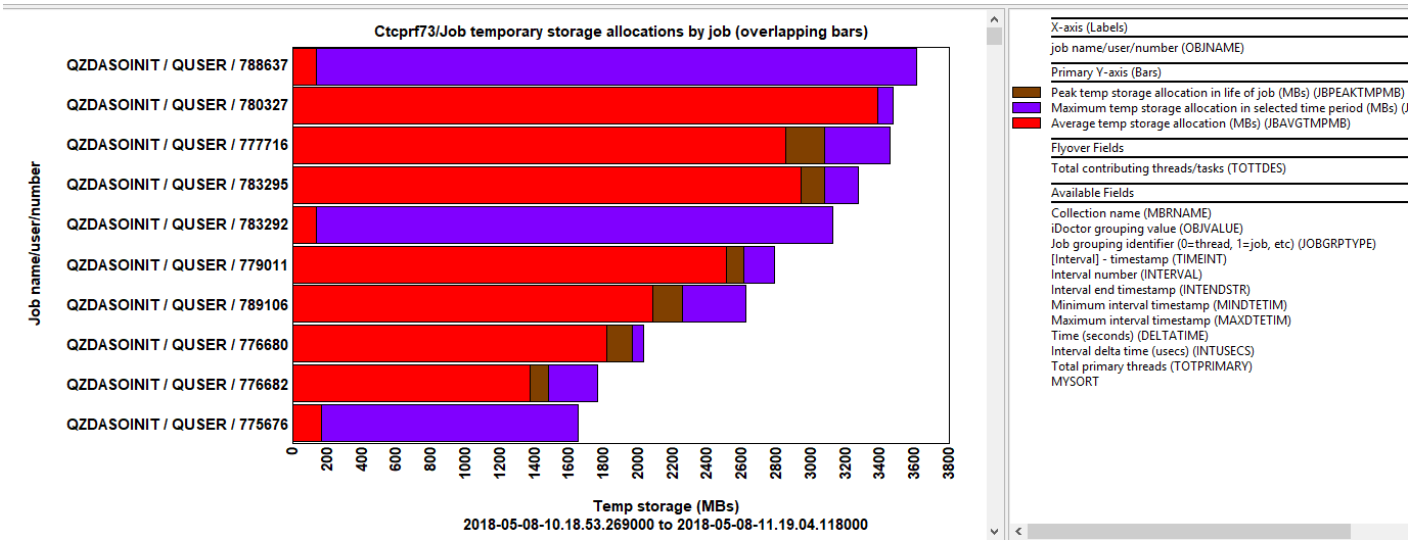
Temporary storage -> Job temporary storage rankings

8.18.4.1 Job temporary storage allocations by job - overlapping bars

This graph ranks the temporary storage allocations on the system by job. The values shown in these charts are:

1) Peak temp storage allocations in life of the job
2) Maximum temp storage allocations in this period
3) Average temp storage allocations

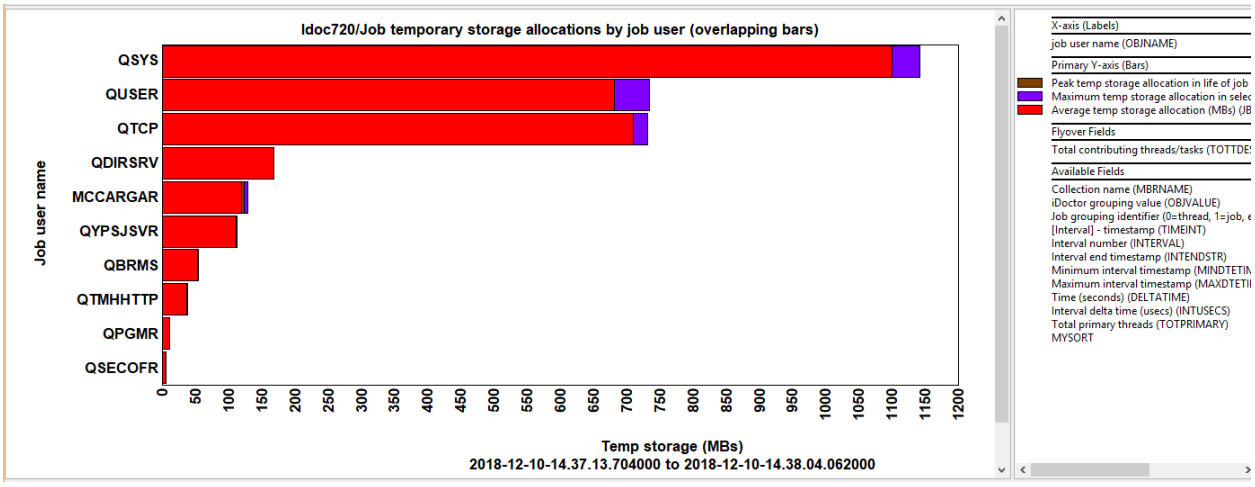
Note: Because the bars overlap each other, the smaller values are visible on the left side of the bar and the larger values are on the right.



Job temporary storage allocations by job - overlapping bars

8.18.4.2 Job temporary storage allocations by job user - overlapping bars

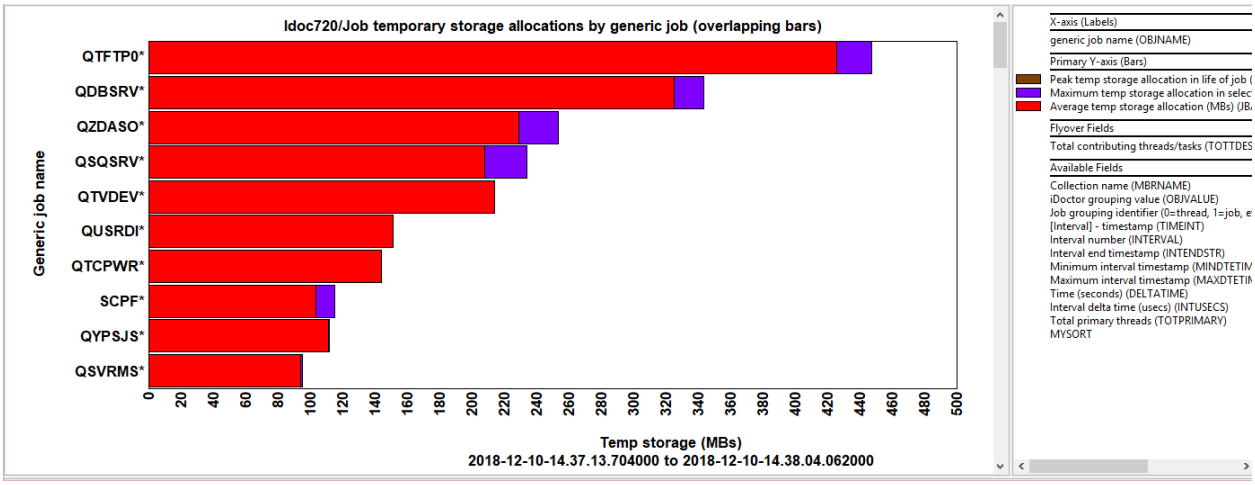
This graph ranks the temporary storage allocations on the system by job user.



Job temporary storage allocations by job user - overlapping bars

8.18.4.3 Job temporary storage allocations by generic job - overlapping bars

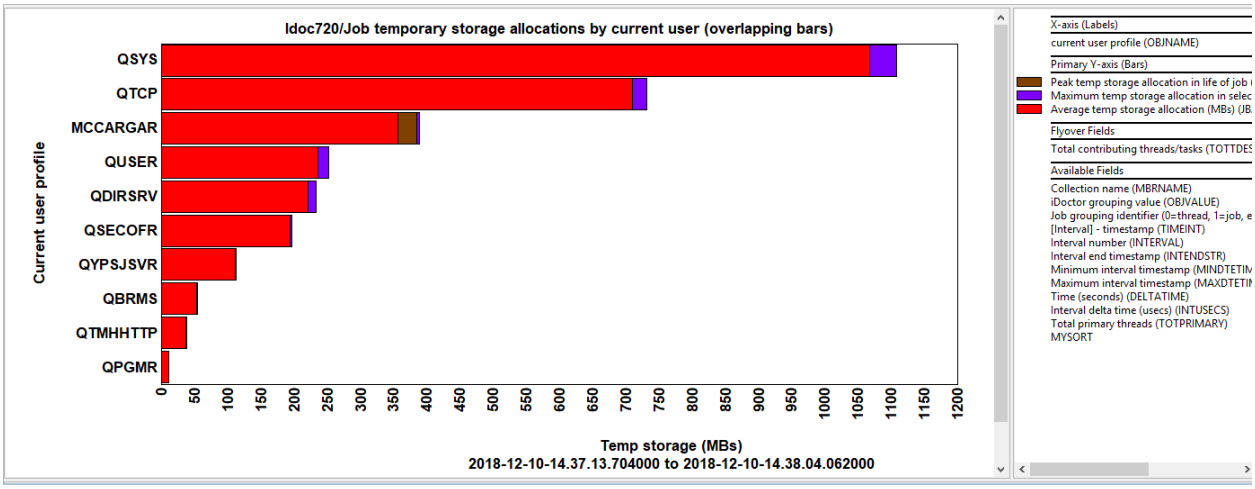
This graph ranks the temporary storage allocations on the system by generic job.



Job temporary storage allocations by generic job - overlapping bars

8.18.4.4 Job temporary storage allocations by current user - overlapping bars

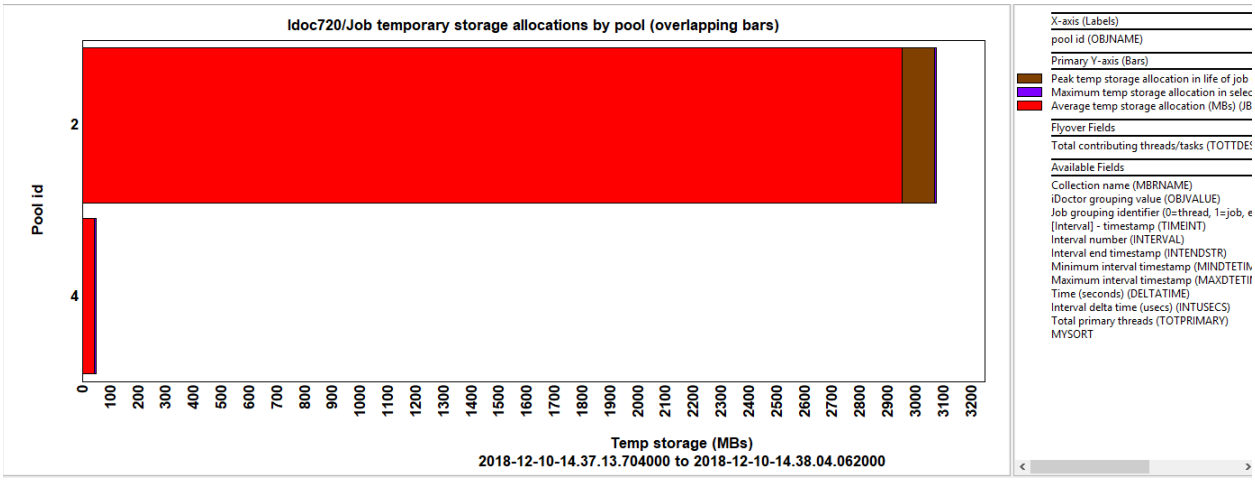
This graph ranks the temporary storage allocations on the system by current user.



Job temporary storage allocations by current user - overlapping bars

8.18.4.5 Job temporary storage allocations by pool - overlapping bars

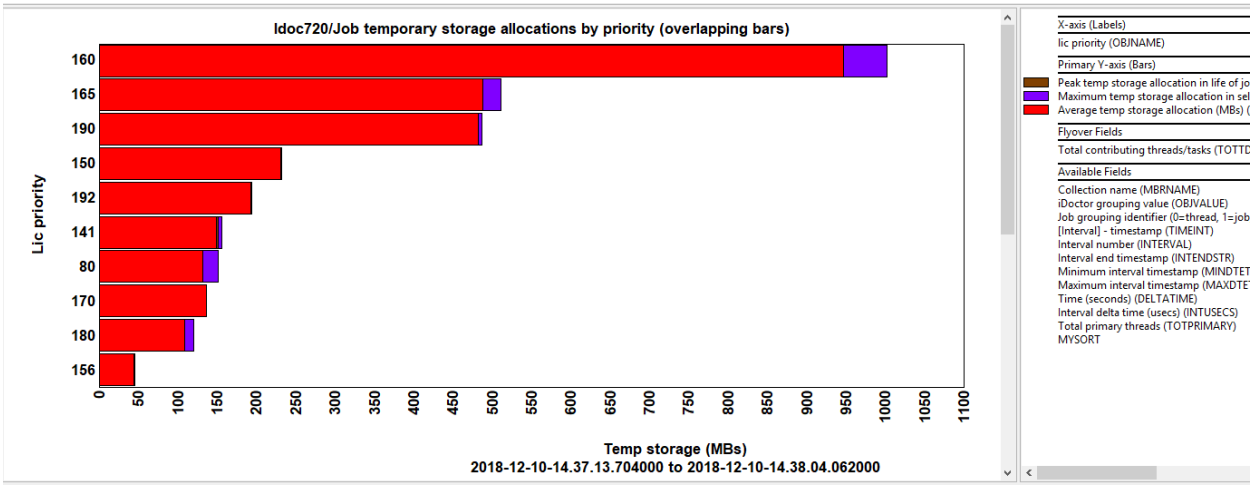
This graph ranks the temporary storage allocations on the system by pool.



Job temporary storage allocations by pool - overlapping bars

8.18.4.6 Job temporary storage allocations by priority - overlapping bars

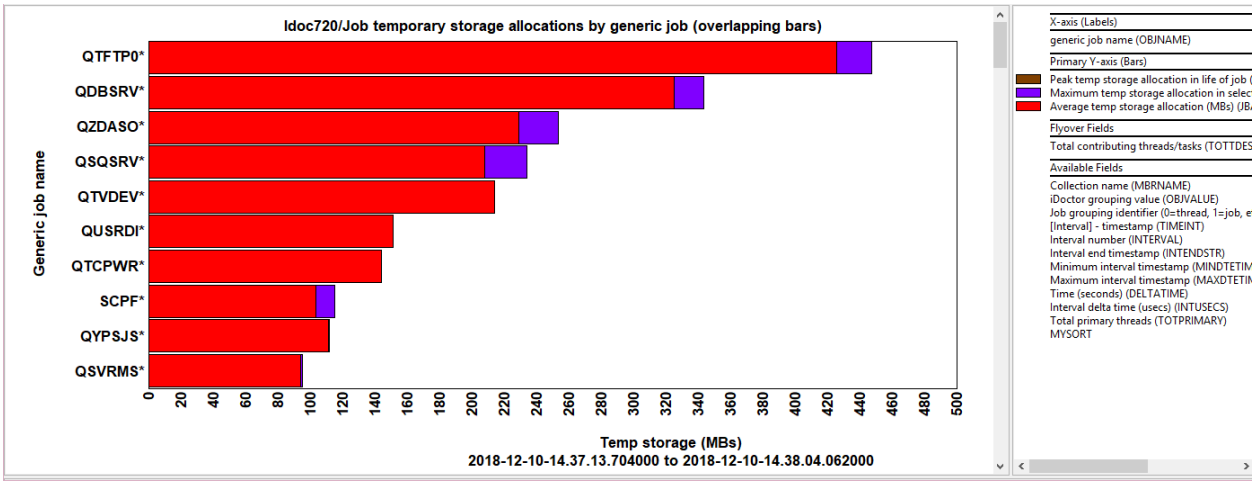
This graph ranks the temporary storage allocations on the system by priority.



Job temporary storage allocations by priority - overlapping bars

8.18.4.7 Job temporary storage allocations by subsystem - overlapping bars

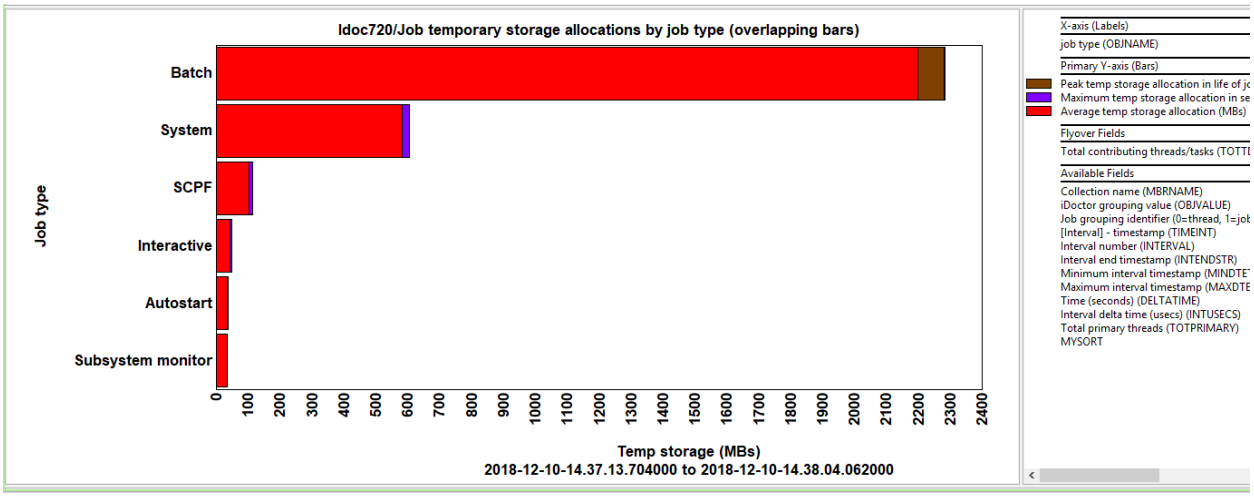
This graph ranks the temporary storage allocations on the system by subsystem.



Job temporary storage allocations by subsystem - overlapping bars

8.18.4.8 Job temporary storage allocations by job type - overlapping bars

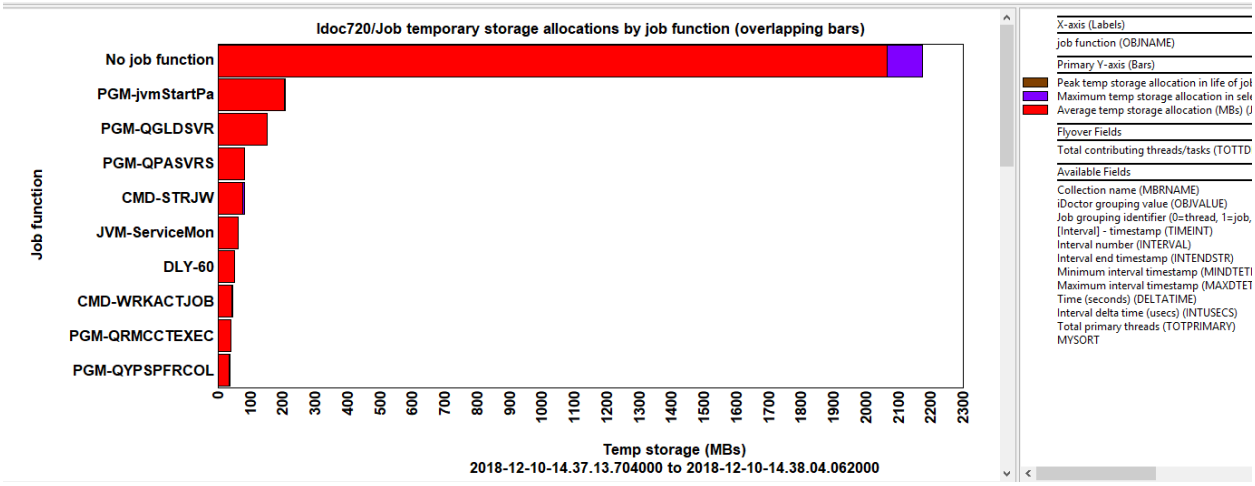
This graph ranks the temporary storage allocations on the system by job type.



Job temporary storage allocations by job type - overlapping bars

8.18.4.9 Job temporary storage allocations by job function - overlapping bars

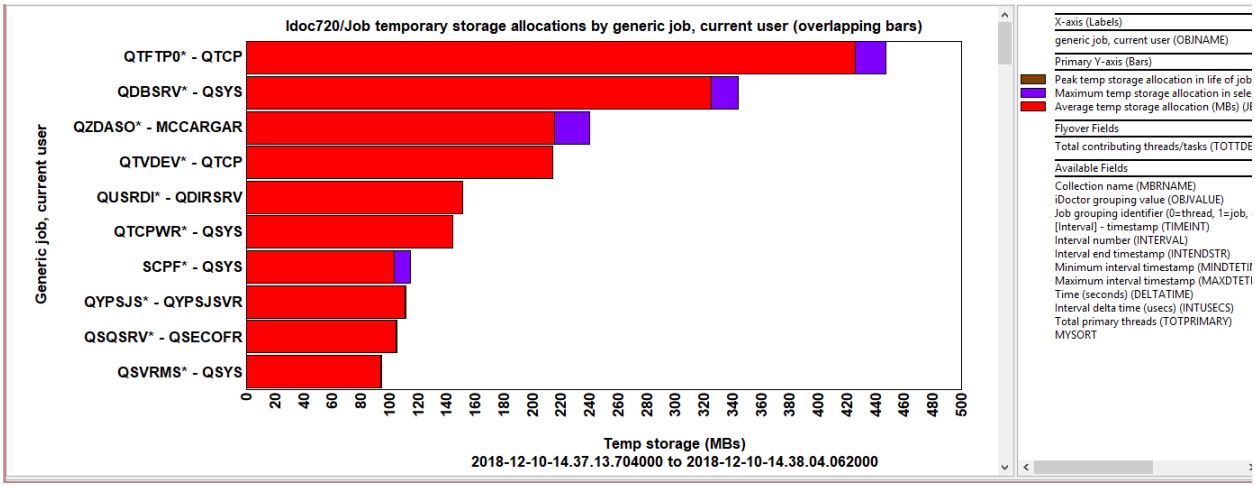
This graph ranks the temporary storage allocations on the system by job function.



Job temporary storage allocations by job function - overlapping bars

8.18.4.10 Job temporary storage allocations by generic job, current user - overlapping bars

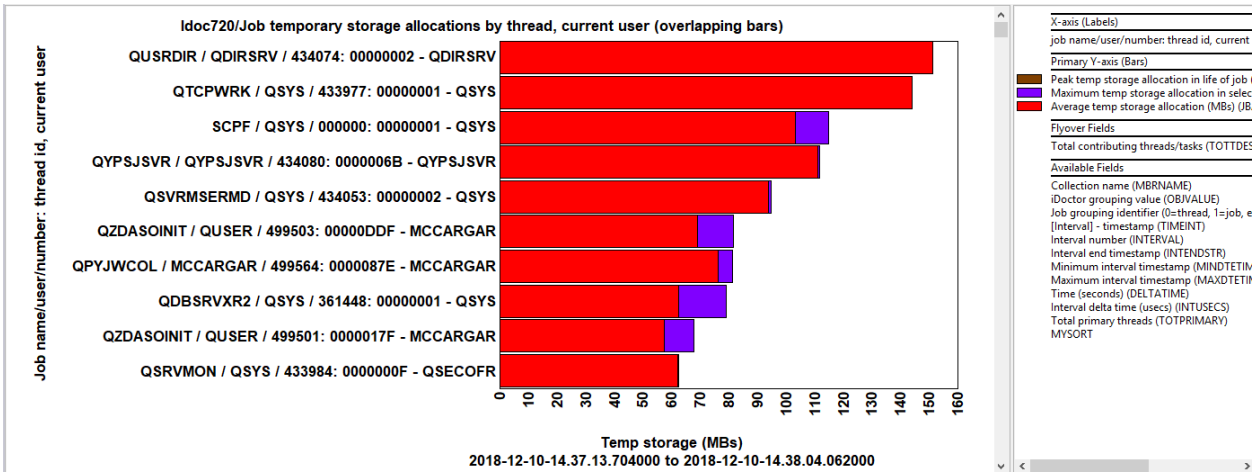
This graph ranks the temporary storage allocations on the system by generic job and current user combination.



Job temporary storage allocations by generic job, current user - overlapping bars

8.18.4.11 Job temporary storage allocations by thread, current user - overlapping bars

This graph ranks the temporary storage allocations on the system by thread, current user.

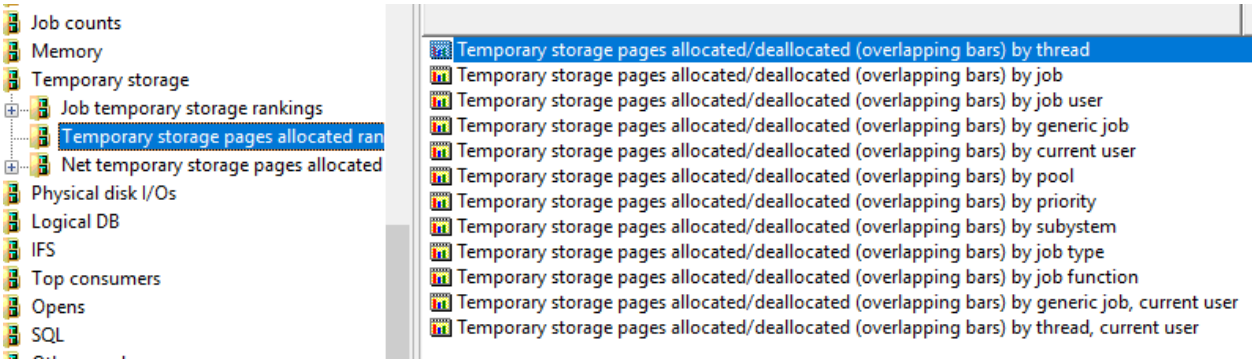


Job temporary storage allocations by thread, current user - overlapping bars

8.18.5 Temporary storage pages allocated/deallocated rankings

These graphs rank 4 metrics related to page allocations either for the entire collection or as a drill-down from a selection on the overview charts within this folder. These statistics are based on thread-based metrics found in QAPYJWTDE. The metrics are:

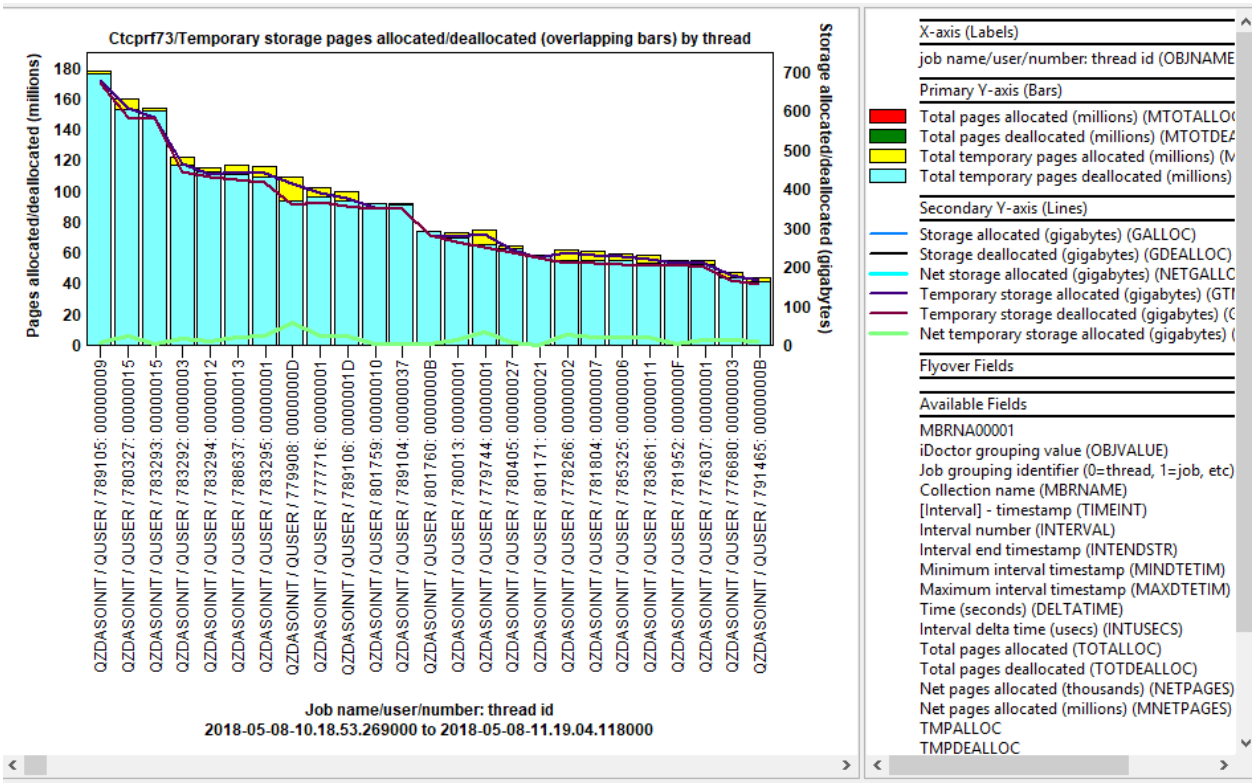
1) Total pages allocated (millions)
2) Total pages deallocated (millions)
3) Total temporary pages allocated (millions)
4) Total temporary pages deallocated (millions)



Temporary storage -> Temporary storage pages allocated/deallocated rankings

8.18.5.1 Temporary storage pages allocated/deallocated - overlapping bars by thread

This graph ranks the pages allocated (both temp and non-temp) on the system by thread.

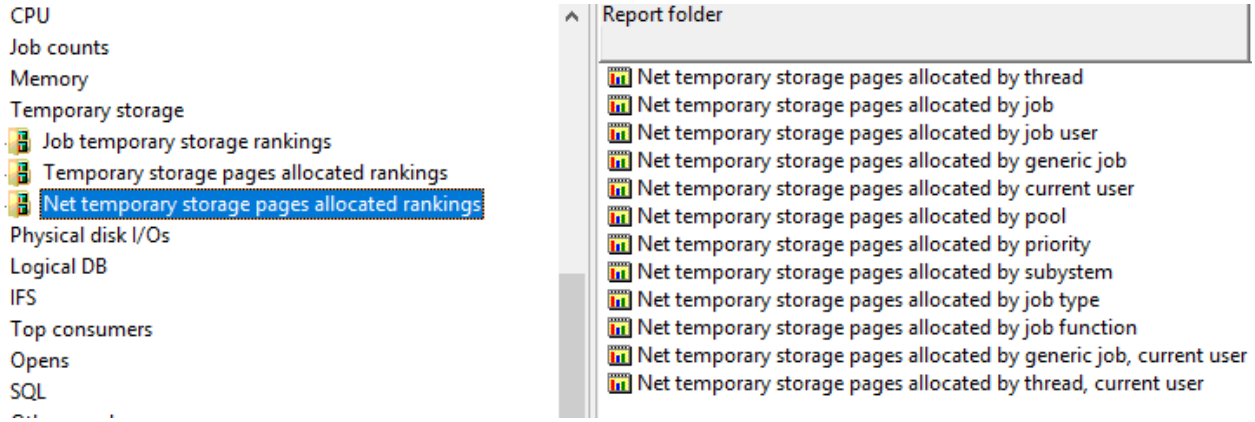


Temporary storage pages allocated/deallocated - overlapping bars by thread

The remaining graphs in this section are not covered but are like the Job temporary storage pages rankings graphs.

8.18.6 Net temporary storage pages allocated rankings

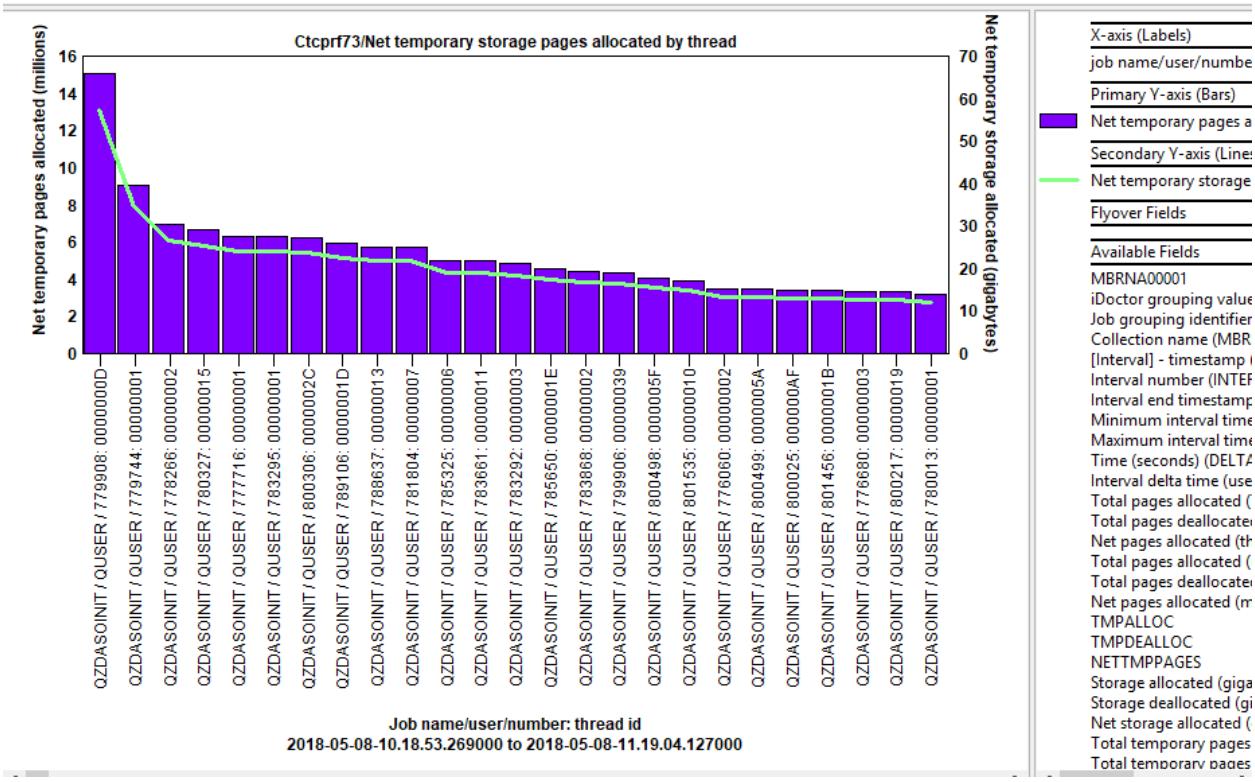
This graph ranks the net temporary storage pages allocated by various job groupings. It includes the net storage size (in gigabytes) as well on the Y2-axis.



Temporary storage -> Net temporary storage pages allocated rankings

8.18.6.1 Net temporary storage pages allocated by thread

This graph ranks the net temporary storage pages allocated by thread.



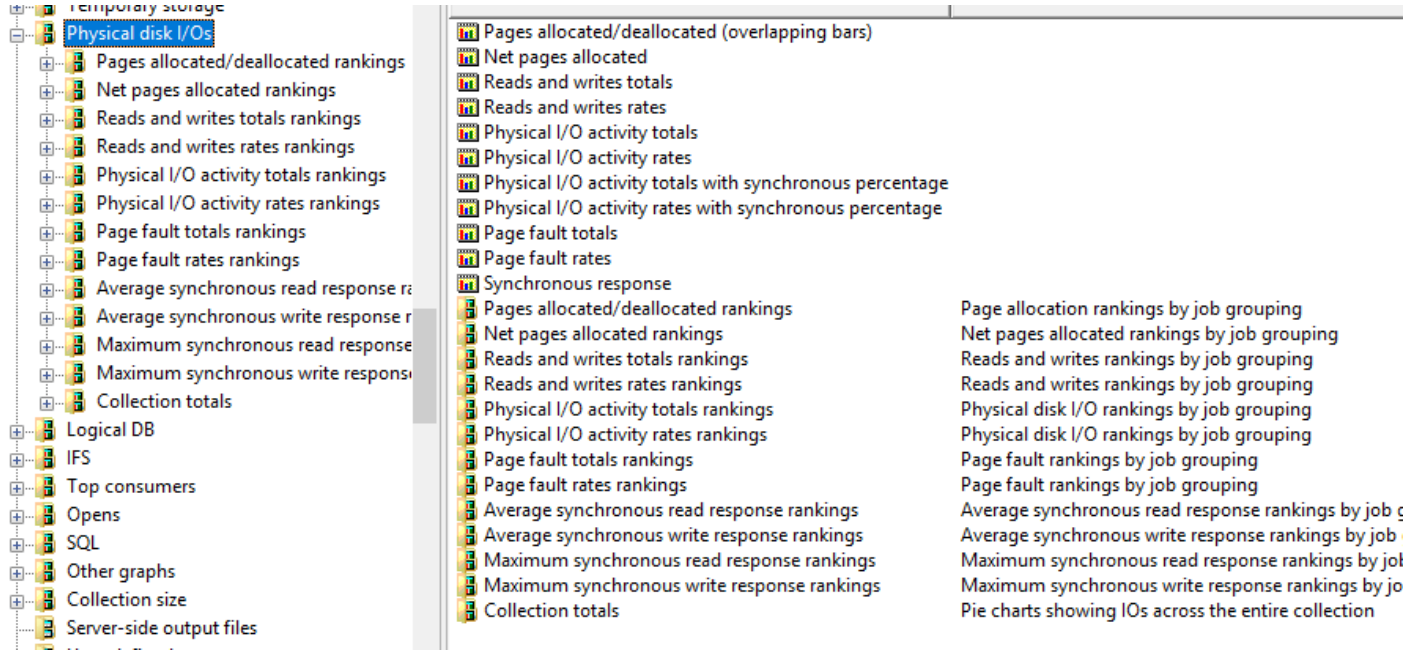
Net temporary storage pages allocated by thread

The remaining graphs in this section are not covered but are like the Job temporary storage pages rankings graphs.

8.19 Physical Disk I/Os

This folder contains graphs related to pages allocated/deallocated, disk reads/writes and synchronous or asynchronous physical disk I/Os. Both overview graphs and ranking graphs (by job) are provided.

Note: Some graphs require the Collection Summary analysis to be ran first in order to appear.



Physical disk I/Os folder

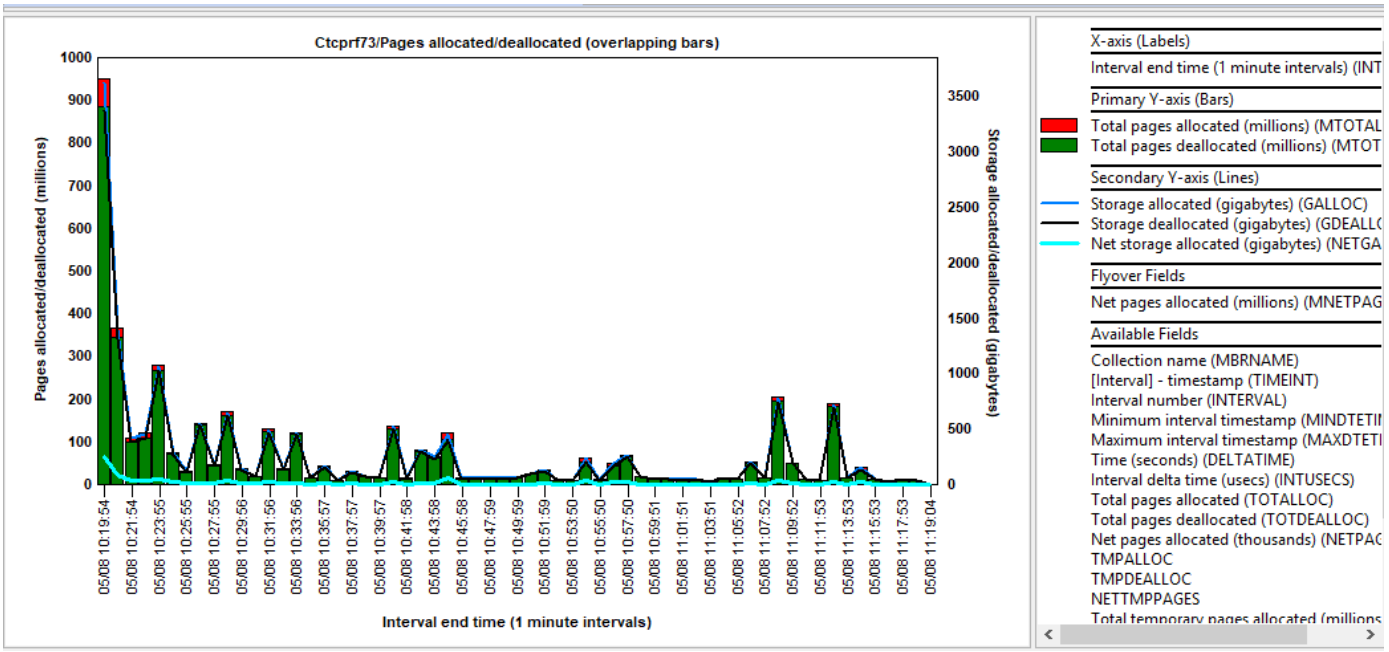
8.19.1 Pages allocated/deallocated - overlapping bars

This graph displays the total 4K pages allocated and deallocated. The bars are overlapping with the highest values shown at the top of each bar. The smaller values will be displayed at the bottom. The metrics are displays in millions.

> **Note:** The metrics shown on these graphs are the total **changed** values within each time interval. They do NOT include totals for any pages left allocated from the past.

This graph requires that the Collection summary be ran first in order to appear.

The Y2-axis displays the total storage allocations/deallocations changes for these 4K pages added up over time (in gigabytes). It also displays the net storage allocated for all pages.

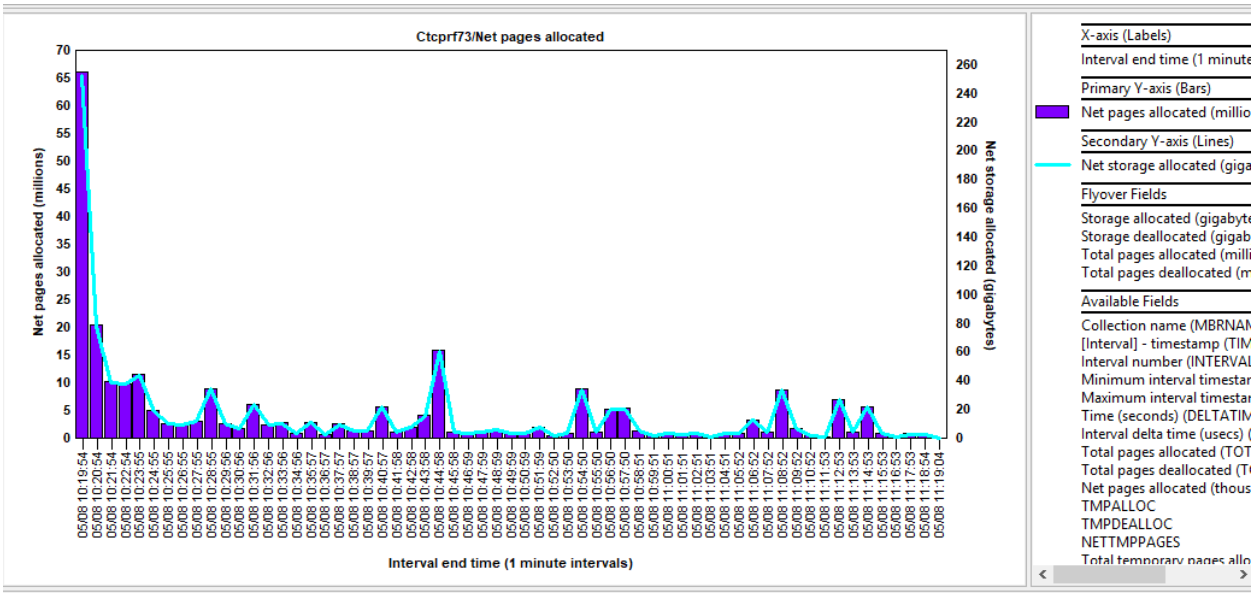


Pages allocated/deallocated - overlapping bars

8.19.2 Net pages allocated

This graph displays the net pages allocated (allocations – deallocations) as well as the net storage allocations on the Y2-axis.

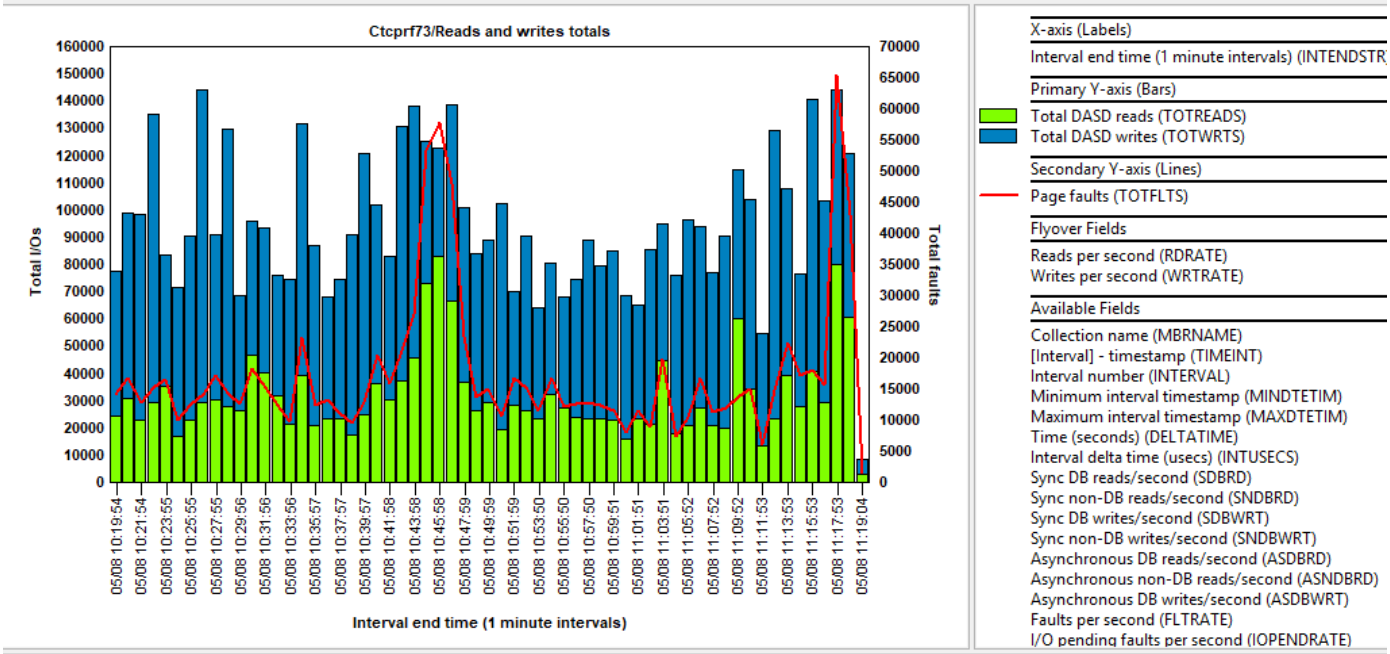
This graph requires that the Collection summary be ran first in order to appear.



Net pages allocated

8.19.3 Read and writes totals

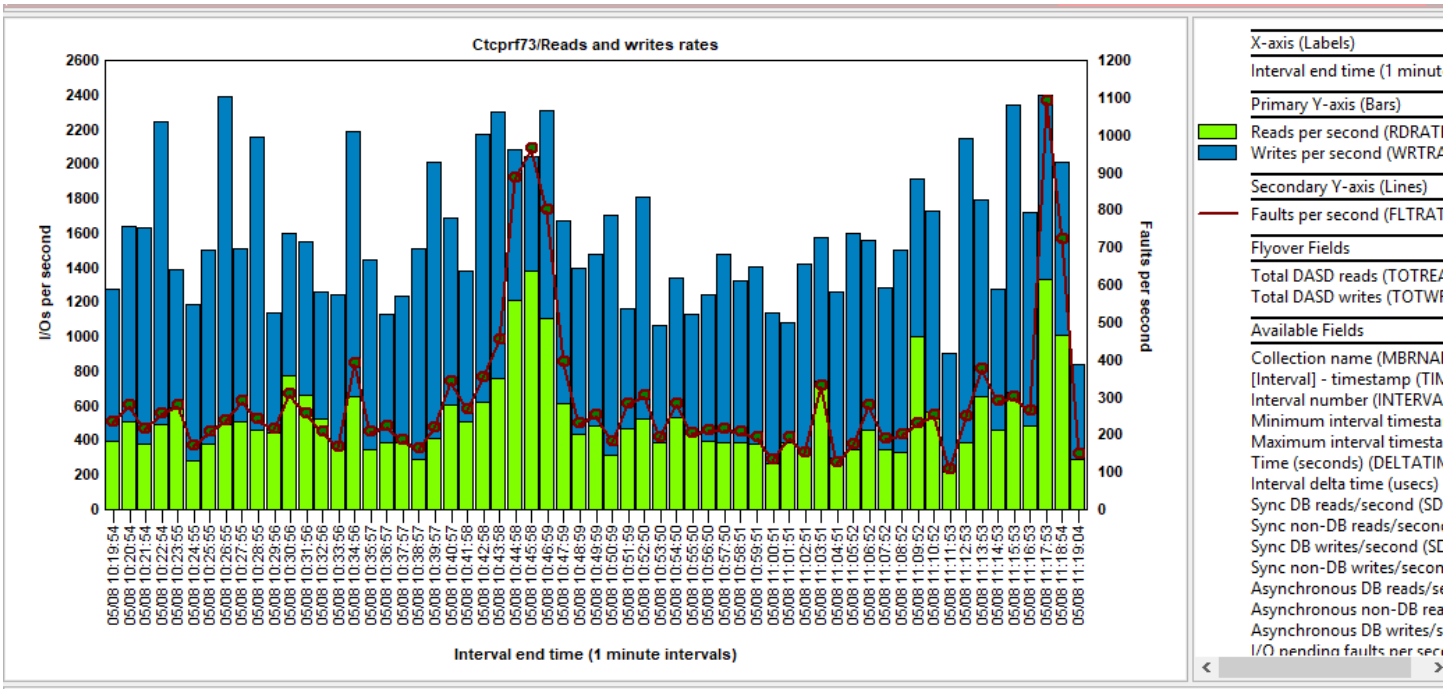
This graph displays the total disk reads and writes for all jobs captured in the collection. The Y2-axis displays the total page faults.



Reads and writes totals

8.19.4 Read and writes rates

This graph displays the disk reads and writes rates per second for all jobs captured in the collection. The Y2-axis displays the page faults per second.

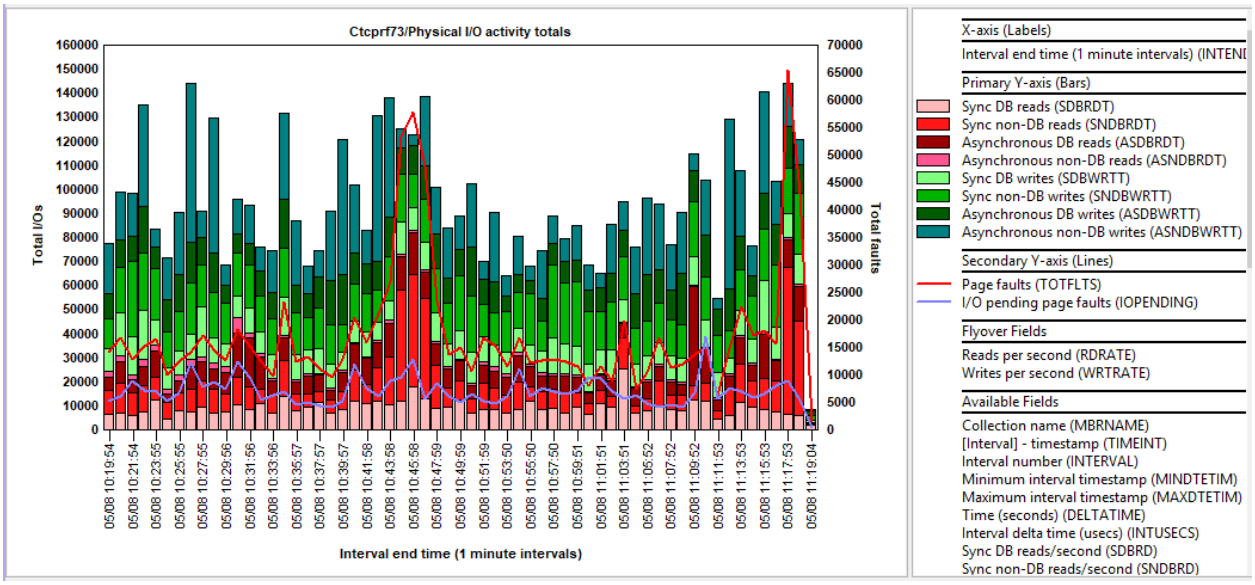


Reads and writes rates

8.19.5 Physical I/O activity totals

This graph shows physical I/O totals for jobs added together per time interval.

These counters include synchronous or asynchronous, database or non-database reads and writes. The Y2-axis displays total page faults as well as IO pending page faults.

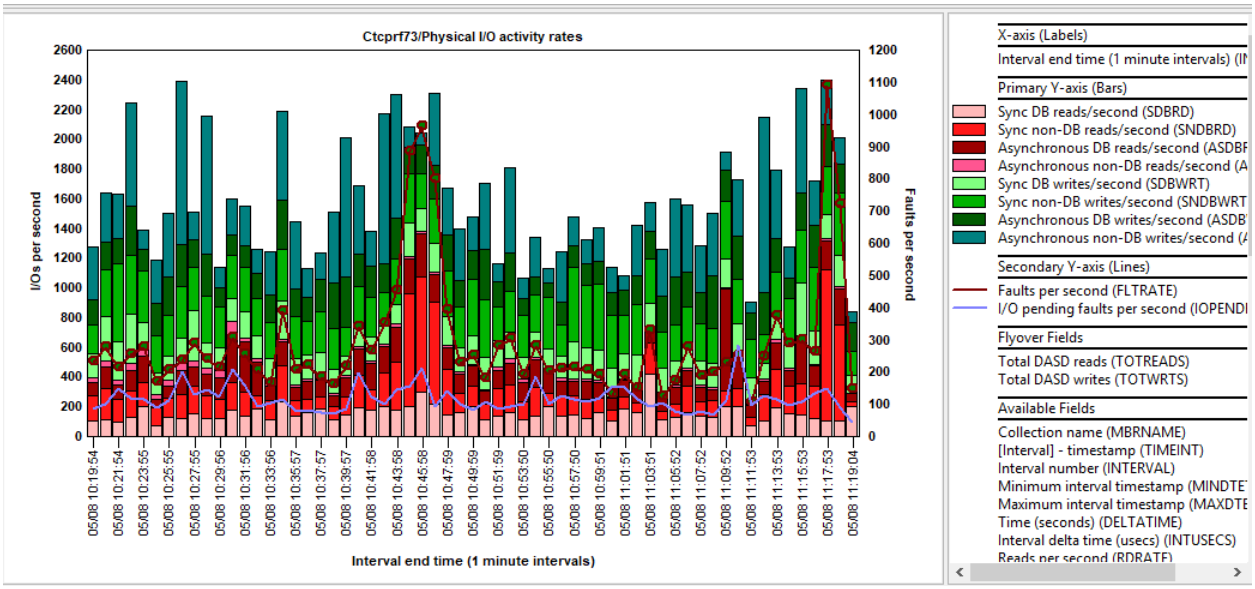


Physical I/O activity totals

8.19.6 Physical I/O activity rates

This graph shows physical I/O rates per second for jobs added together per time interval.

These counters include synchronous or asynchronous, database or non-database reads and writes. The Y2-axis displays page faults per second and IO pending page faults per second.

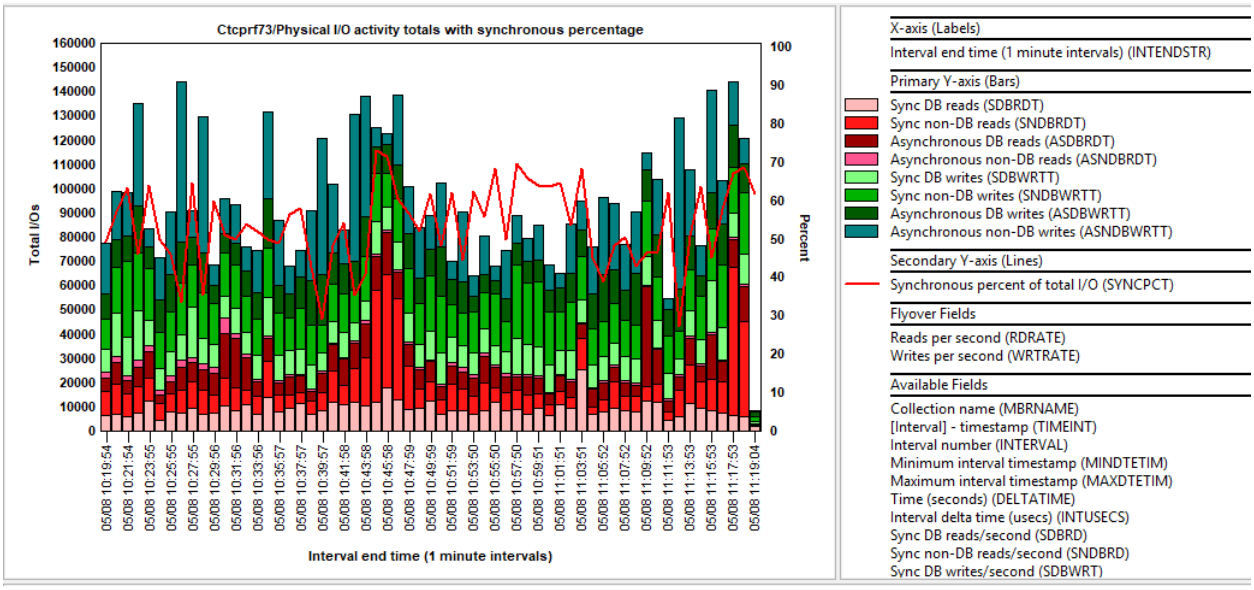


Physical I/O activity rates

8.19.7 Physical I/O activity totals with synchronous percentage

This graph shows physical I/O totals for jobs added together per time interval.

These counters include synchronous or asynchronous, database or non-database reads and writes. The Y2-axis displays the percentage of synchronous I/Os of the total physical disk I/Os.

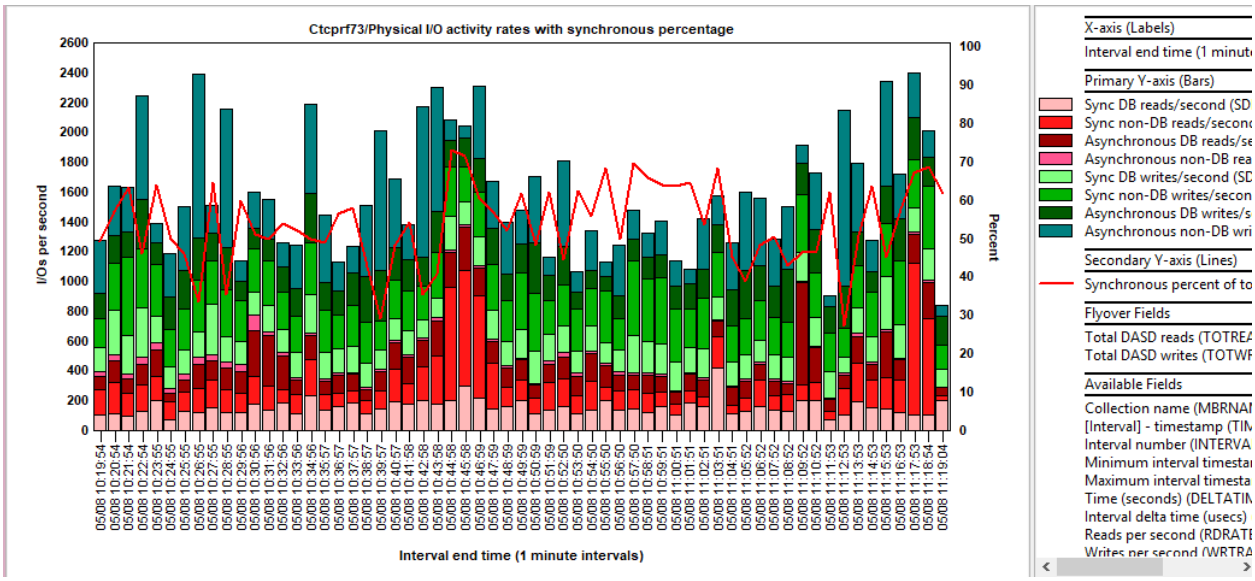


Physical I/O activity totals with synchronous percentage

8.19.8 Physical I/O activity rates with synchronous percentage

This graph shows physical I/O rates per second for jobs added together per time interval.

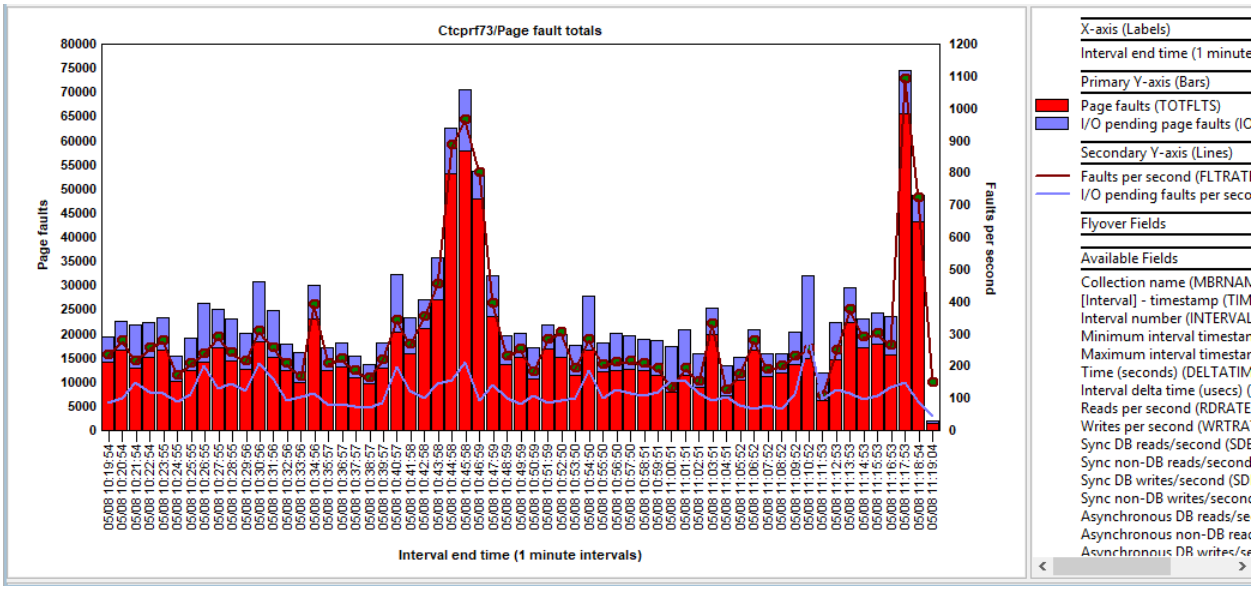
These counters include synchronous or asynchronous, database or non-database reads and writes. The Y2-axis displays the percentage of synchronous I/Os of the total physical disk I/Os.



Physical I/O activity rates with synchronous percentage

8.19.9 Page fault totals

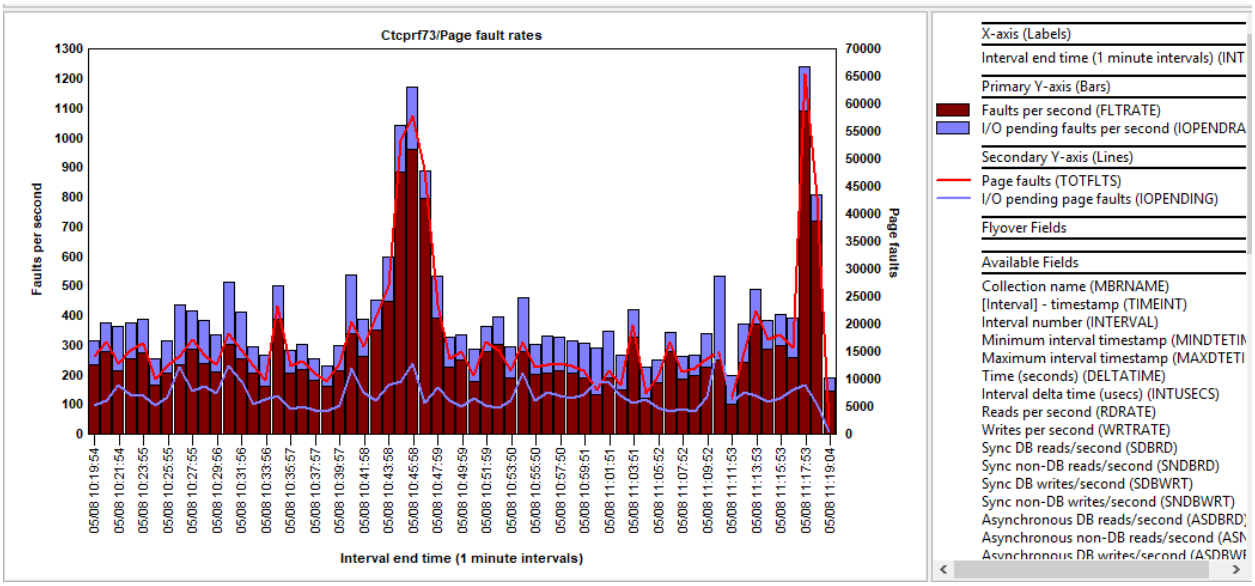
This graph shows total page faults, total I/O pending page faults as well as the rates for each on the secondary Y-axis.



Page fault totals

8.19.10 Page fault rates

This graph shows page faults per second, I/O pending page faults per second on the primary Y-axis with the totals for each on the secondary Y-axis.



Page fault rates

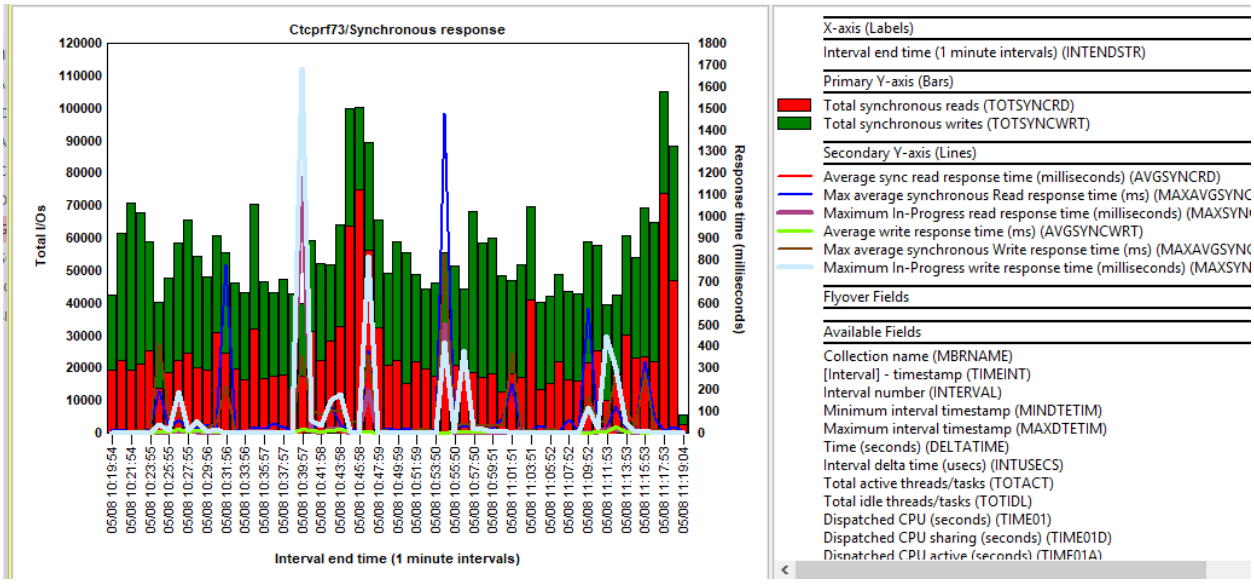
8.19.11 Synchronous response

This graph shows the total number of synchronous reads and writes as well as averages and maximums read and write response time (in milliseconds) for each time interval.

This graph requires that the Collection summary be ran first in order to appear.

Note: The average and maximum times will only differ if the time range is set to something larger than the collected interval size.

4 different sets of average synchronous response graphs exist with different sort metrics are available.

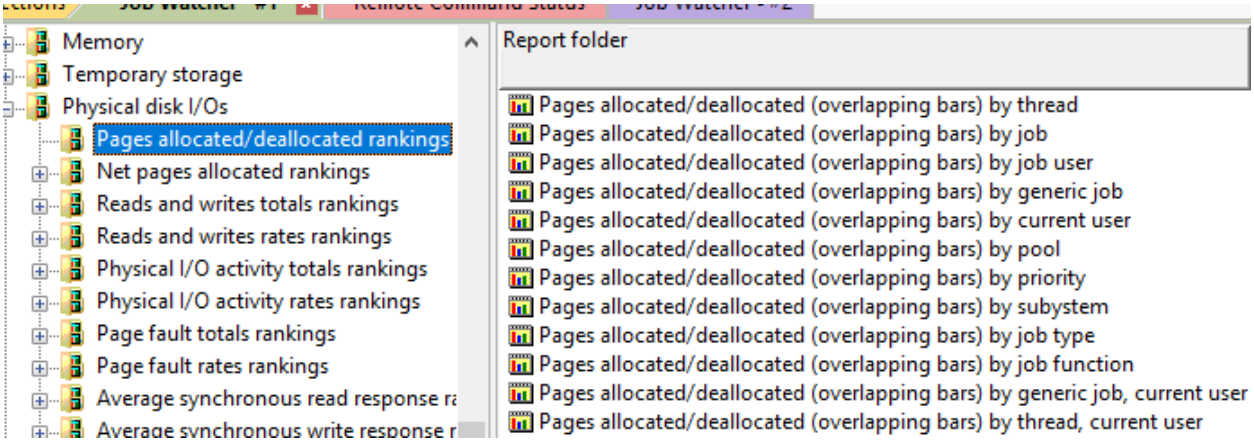


Synchronous response

8.19.12 Pages allocated/deallocated rankings

This folder contains the set of job rankings graphs for the **Pages allocated/deallocated - overlapping bars** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Pages allocated/deallocated rankings menu.

This folder requires that the Collection summary be ran first in order to appear.

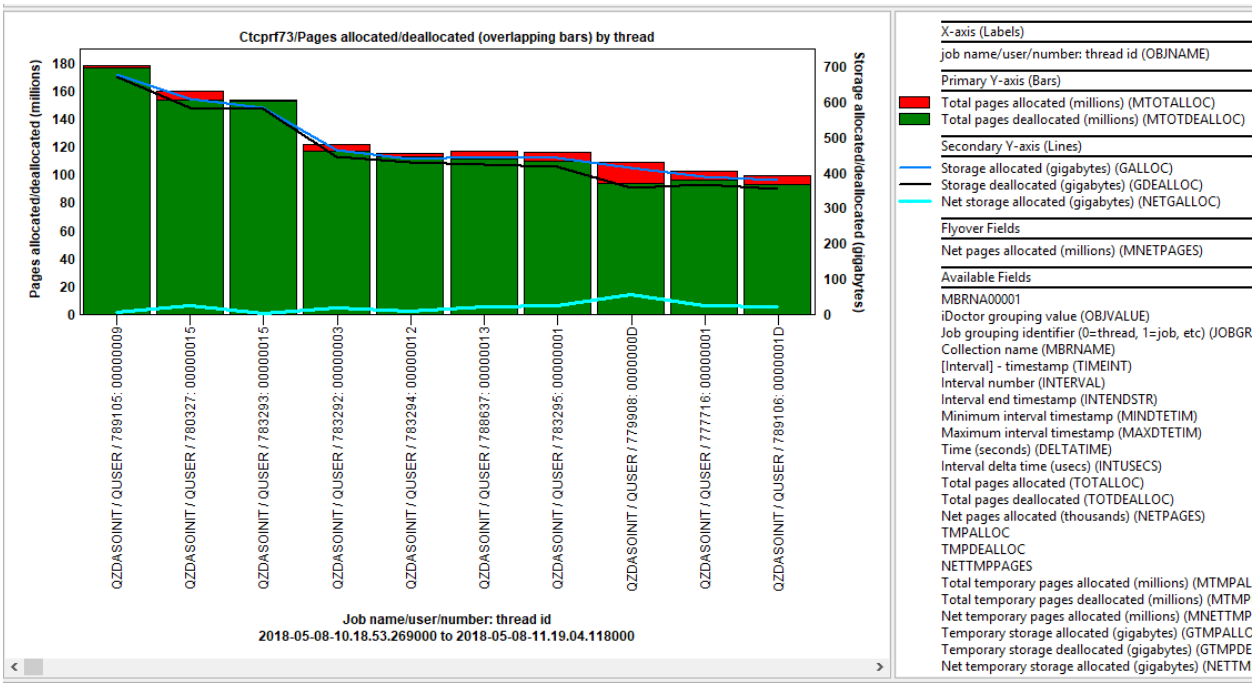


Physical disk I/Os -> Pages allocated/deallocated rankings

8.19.12.1 Pages allocated/deallocated - overlapping bars by thread

This graph ranks the threads/tasks that had the highest total pages allocated or deallocated. These bars are overlapping with the smaller number visible at the bottom and the larger value at the top.

Note: If the 2 metrics are very close in size the pages allocated will only slightly visible.

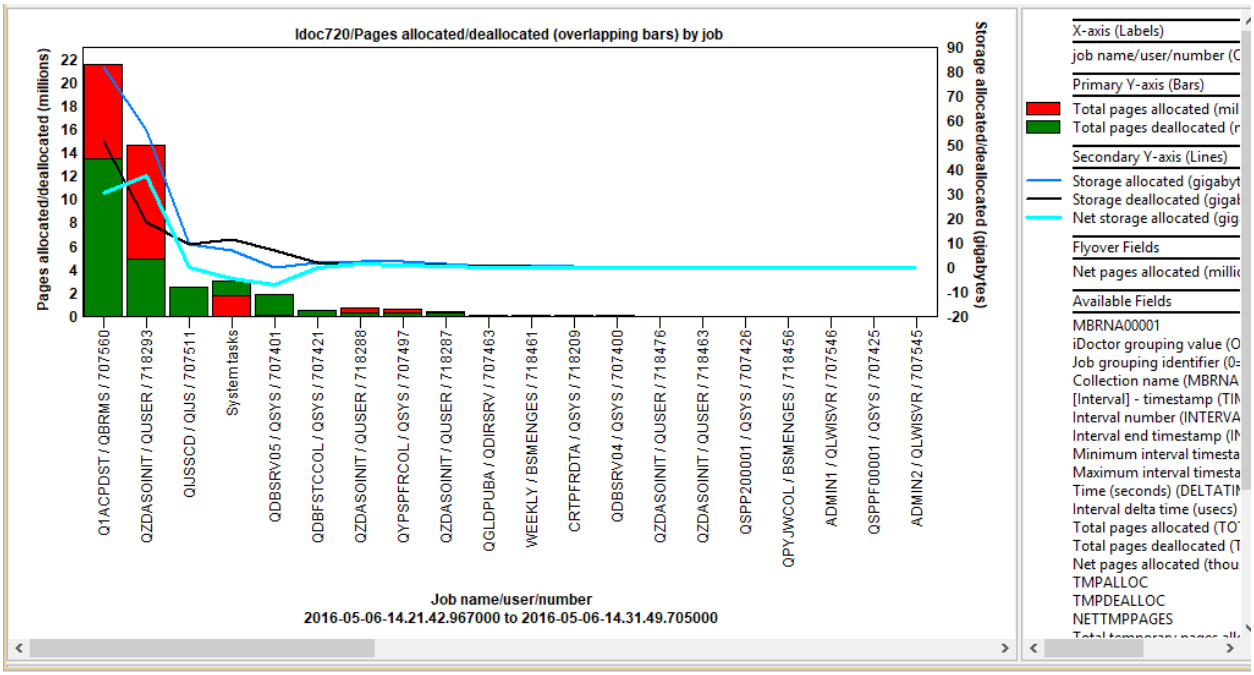


Pages allocated/deallocated - overlapping bars by thread

8.19.12.2 Pages allocated/deallocated - overlapping bars by job

This graph ranks the selected time period's pages allocated/deallocated by primary thread. Jobs that have multiple threads are all added up together.

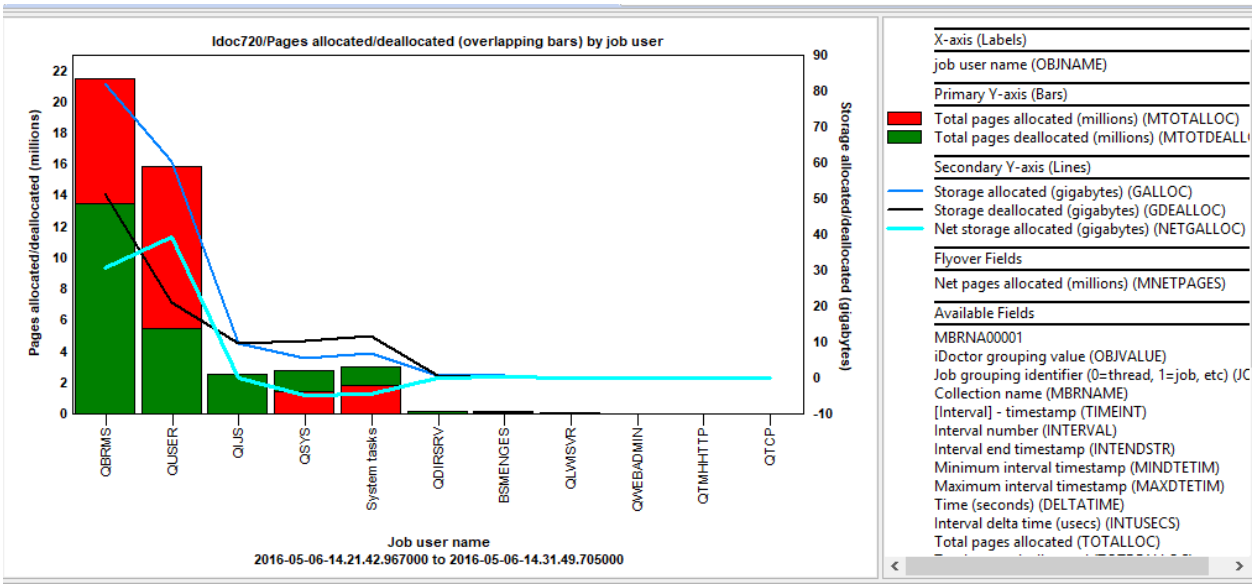
Note: All system tasks are grouped together into one "**System tasks"** record within this report.



Pages allocated/deallocated - overlapping bars by job

8.19.12.3 Pages allocated/deallocated - overlapping bars by job user

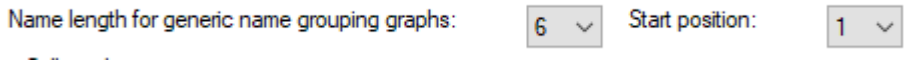
This graph ranks the selected time period's pages allocated/deallocated by job user. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.

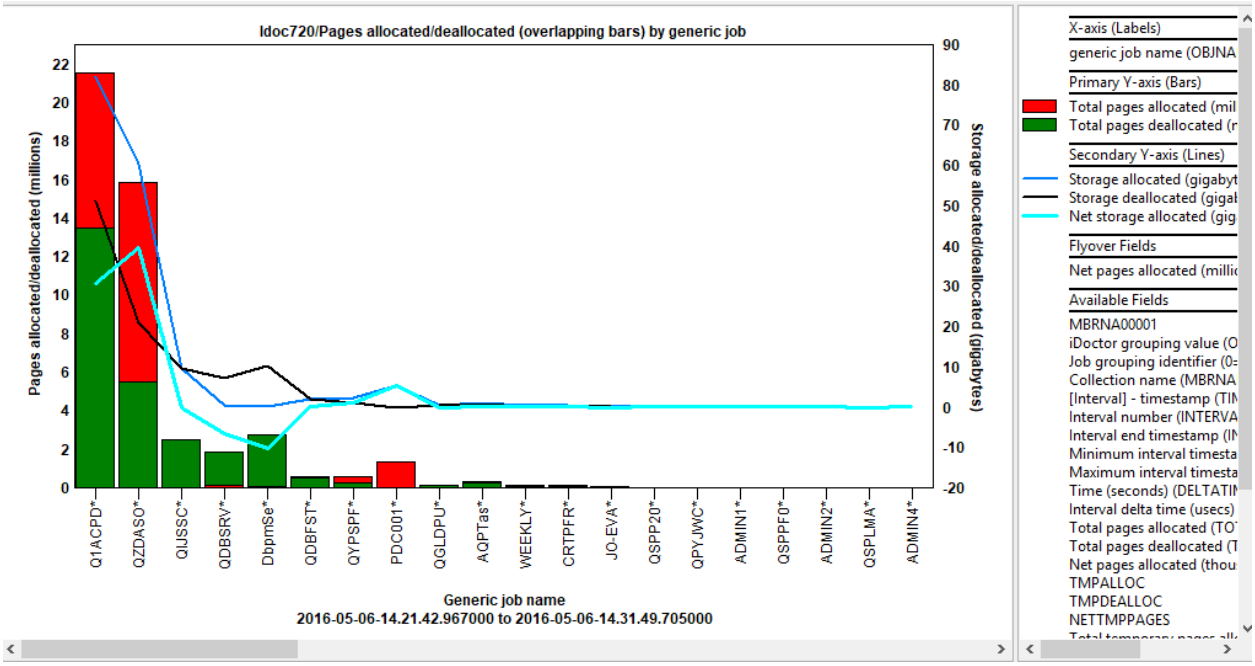


Pages allocated/deallocated - overlapping bars by job user

8.19.12.4 Pages allocated/deallocated - overlapping bars by generic job

This graph ranks the selected time period's pages allocated/deallocated by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

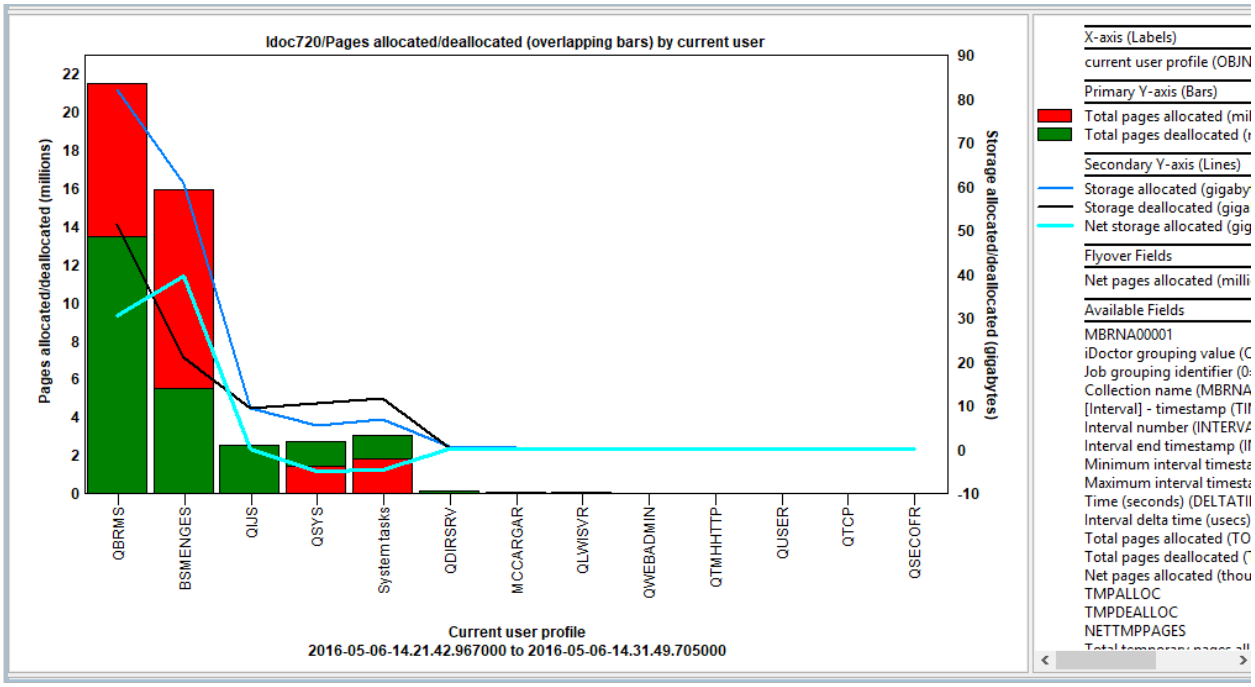




Pages allocated/deallocated - overlapping bars by generic job

8.19.12.5 Pages allocated/deallocated - overlapping bars by current user

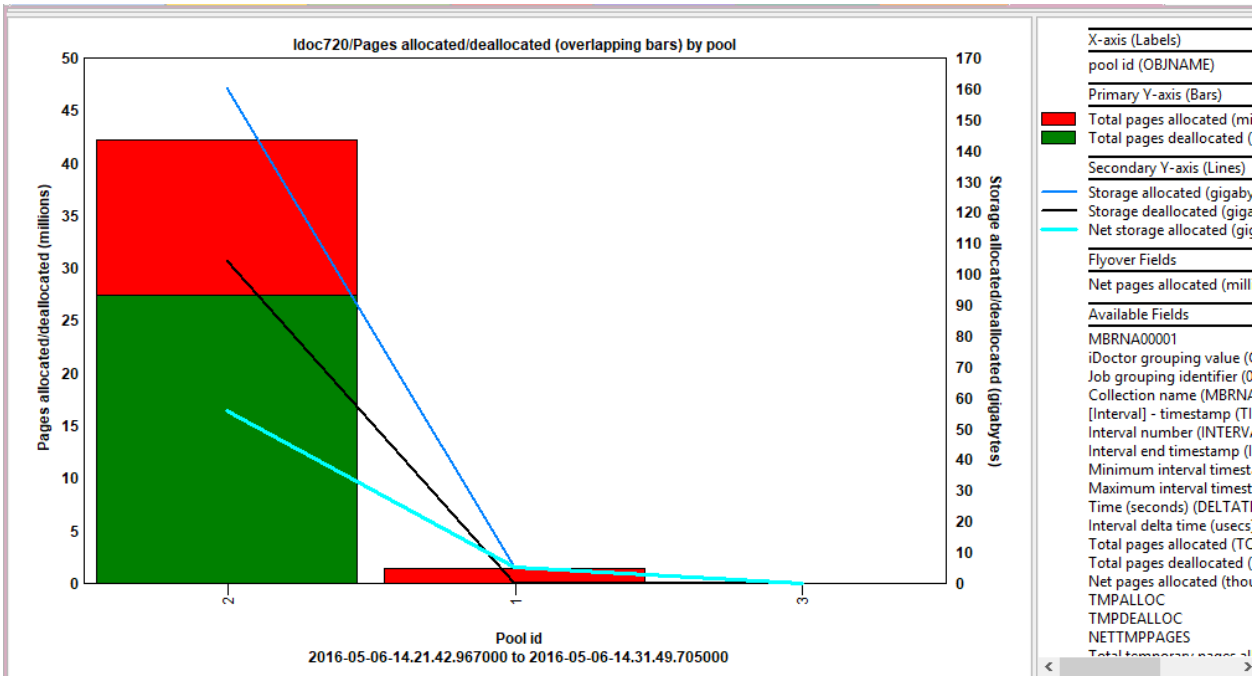
This graph ranks the selected time period's pages allocated/deallocated by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



Pages allocated/deallocated - overlapping bars by current user

8.19.12.6 Pages allocated/deallocated - overlapping bars by pool

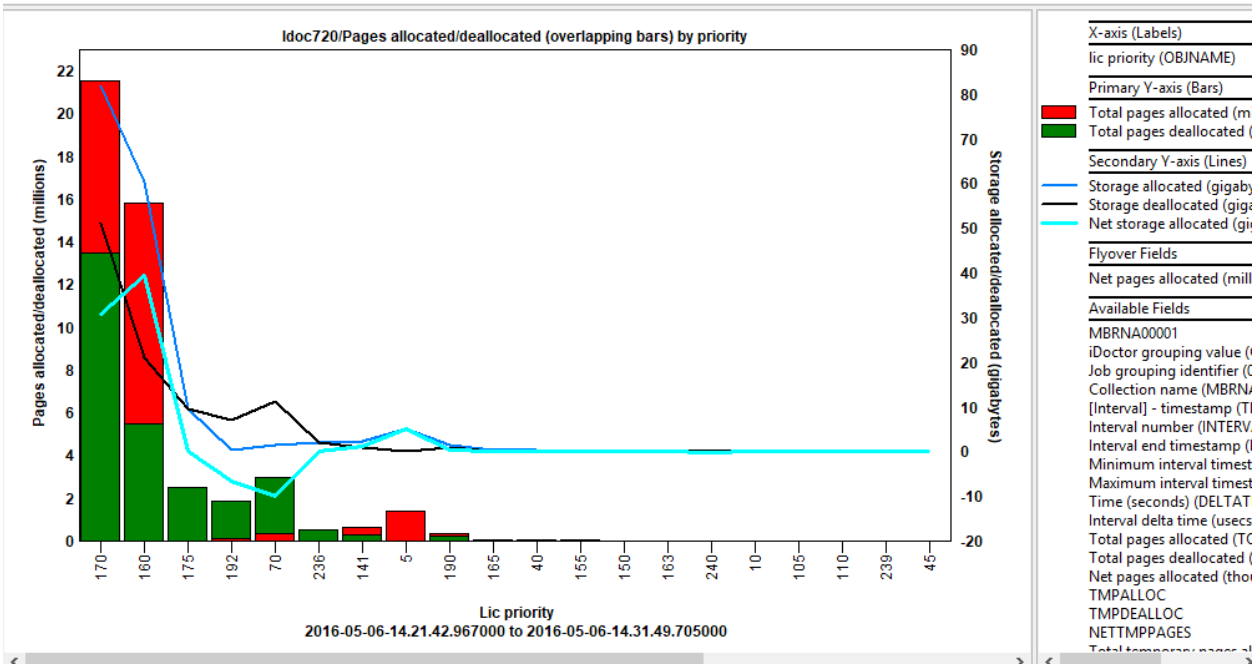
This graph ranks the selected time period's pages allocated/deallocated by memory pool.



Pages allocated/deallocated - overlapping bars by pool

8.19.12.7 Pages allocated/deallocated - overlapping bars by priority

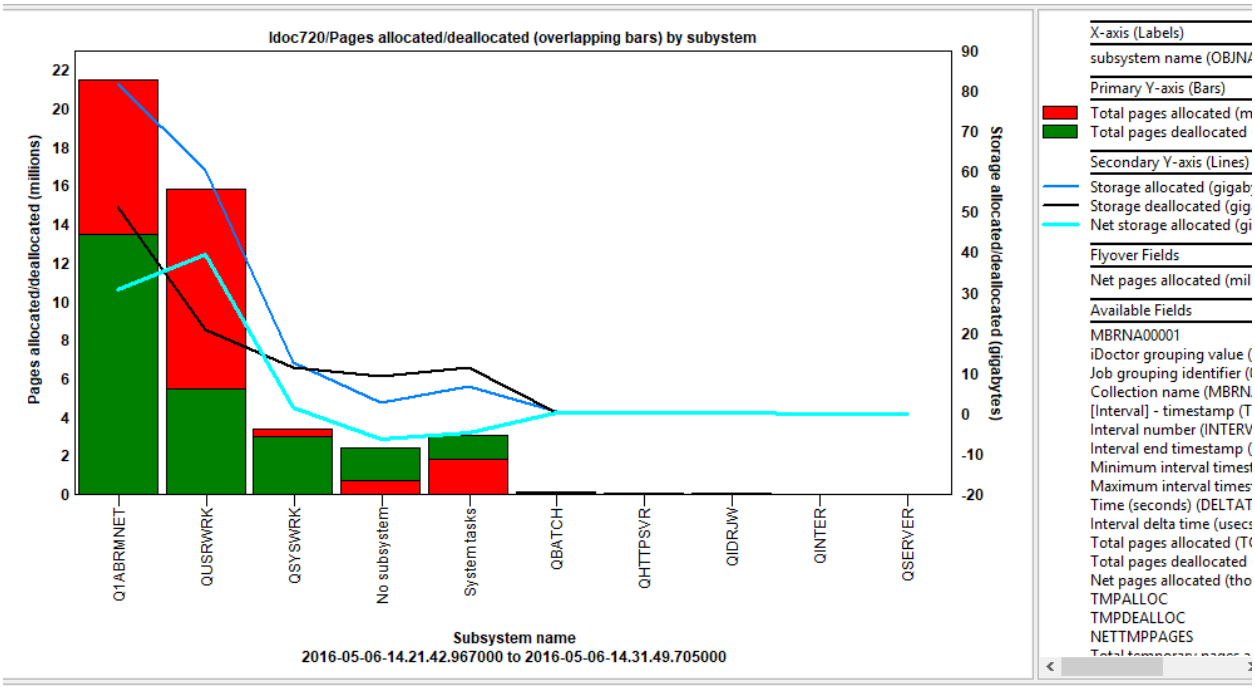
This graph ranks the selected time period's pages allocated/deallocated by LIC priority. XPF priority can be calculated by subtracting 140 for those values shown exceeding 140.



Pages allocated/deallocated - overlapping bars by LIC priority

8.19.12.8 Pages allocated/deallocated - overlapping bars by subsystem

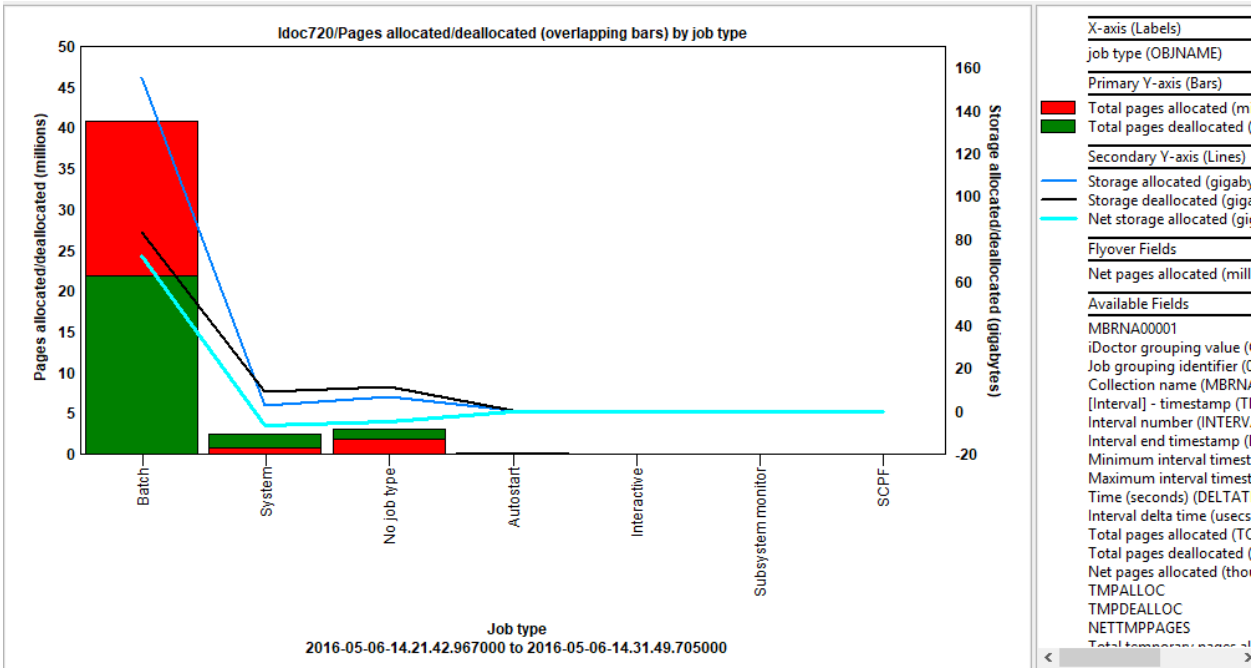
This graph ranks the selected time period's pages allocated/deallocated by subsystem. Job times that had no subsystem listed are grouped into 1 bar called "No subsystem". **Note:** All system tasks are grouped together into one "**System tasks"** bar within this report.



Pages allocated/deallocated - overlapping bars by subsystem

8.19.12.9 Pages allocated/deallocated - overlapping bars by job type

This graph ranks the selected time period's pages allocated/deallocated by job type (such as Batch, interactive, autostart, etc.) Job wait bucket times that had no job type associated with it are grouped into 1 bar called "No job type".

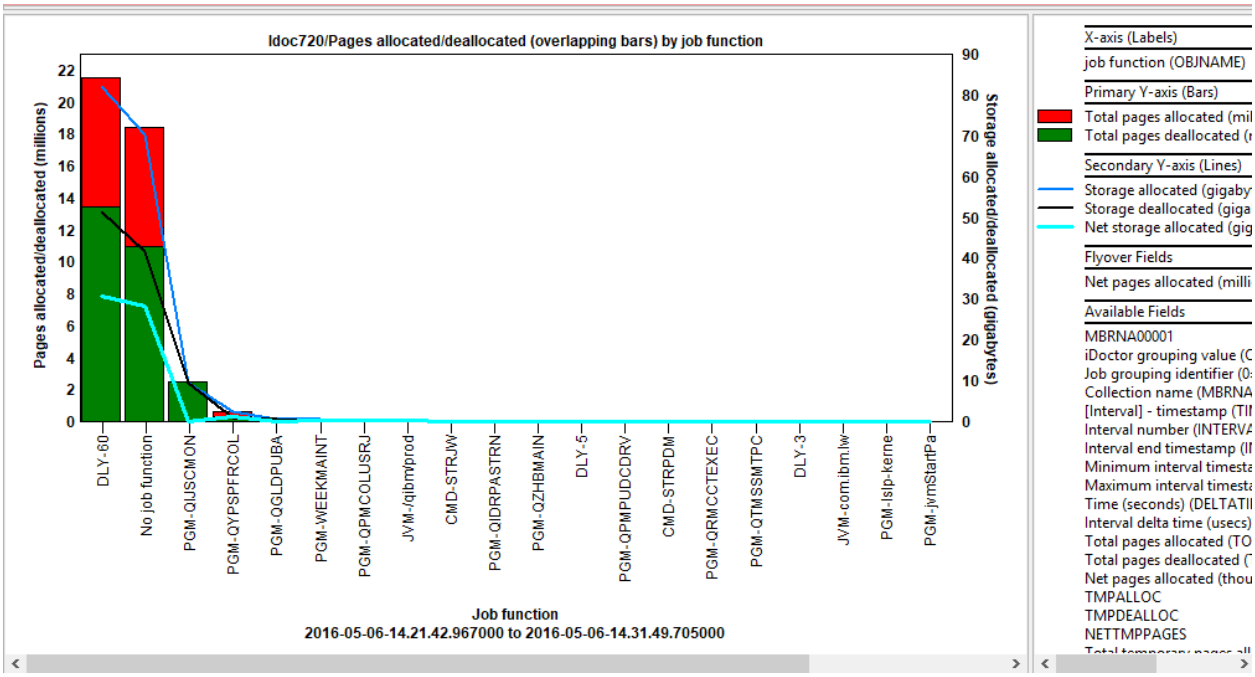


Pages allocated/deallocated - overlapping bars by job type

8.19.12.10 Pages allocated/deallocated - overlapping bars by job function

This graph ranks the selected time period's pages allocated/deallocated by job function. Job function is the same as what you would see normally on the WRKACTJOB command's Function column. Any wait bucket times that had no job function associated with it are grouped into 1 bar called "No job function".

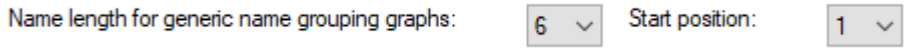
Note: This graph can be misleading in some situations because the current job function for each process is only captured at the end of each Job Watcher interval. The times given are not necessarily 100% from each function listed.



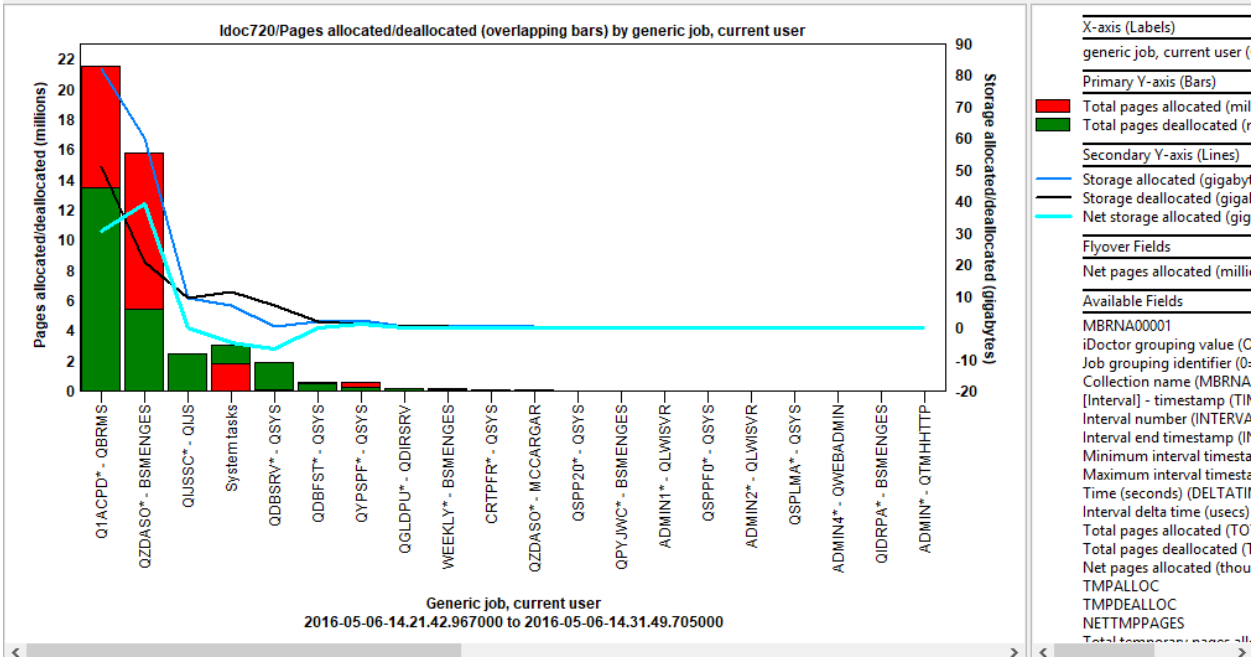
Pages allocated/deallocated - overlapping bars by job function

8.19.12.11 Pages allocated/deallocated - overlapping bars by generic job, current user

This graph ranks the selected time period's pages allocated/deallocated by generic job/current user profile combination. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:



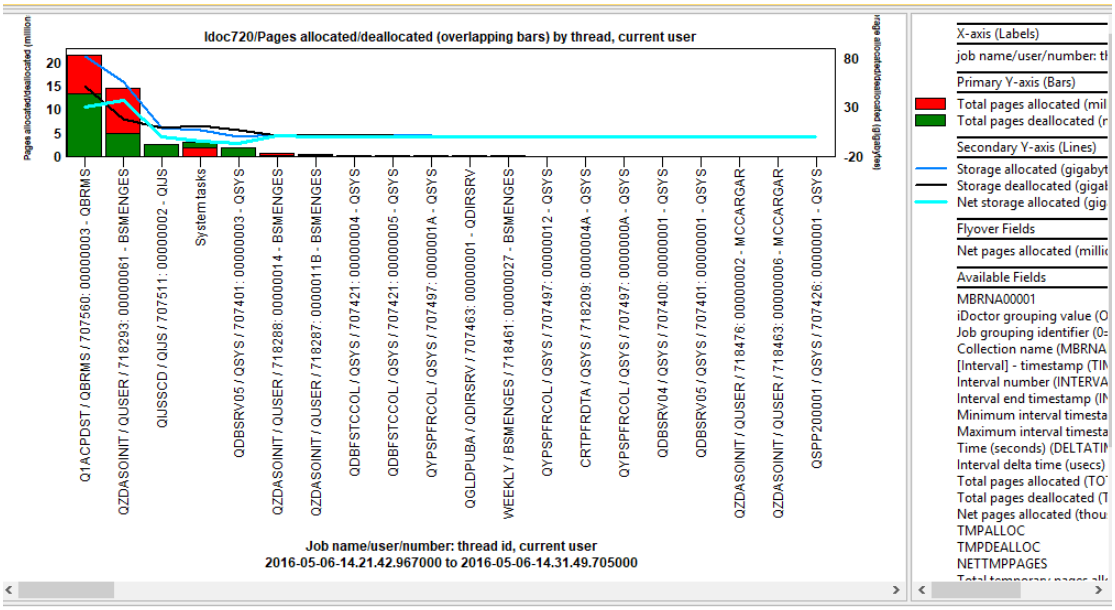
Note: All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



Pages allocated/deallocated - overlapping bars by generic job, current user

8.19.12.12 Pages allocated/deallocated - overlapping bars by thread, current user

This graph ranks the selected time period's pages allocated/deallocated by thread/current user profile combination. **Note:** All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.

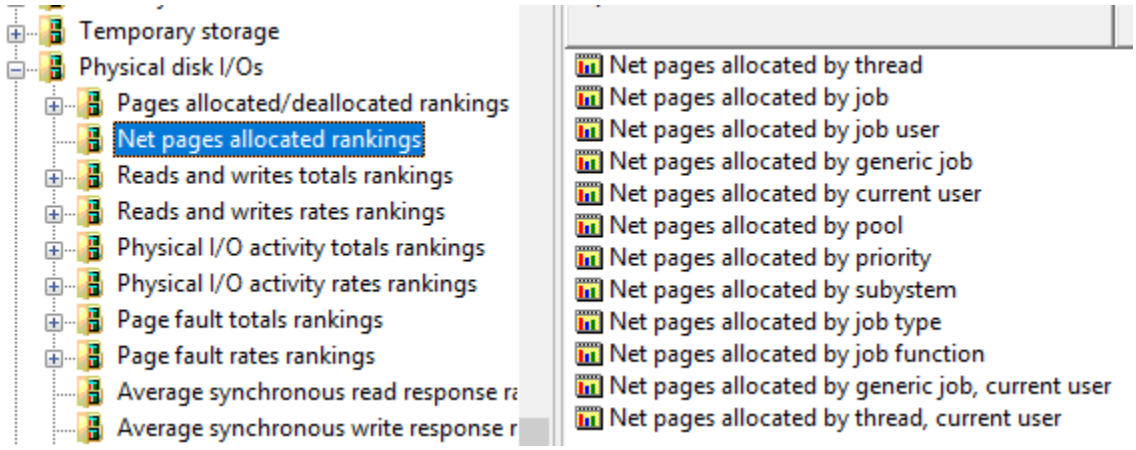


Pages allocated/deallocated - overlapping bars by thread, current user

8.19.13 Net pages allocated rankings

This folder contains the set of job rankings graphs for the **Net pages allocated** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Net pages allocated rankings menu.

This folder requires that the Collection summary be ran first in order to appear.

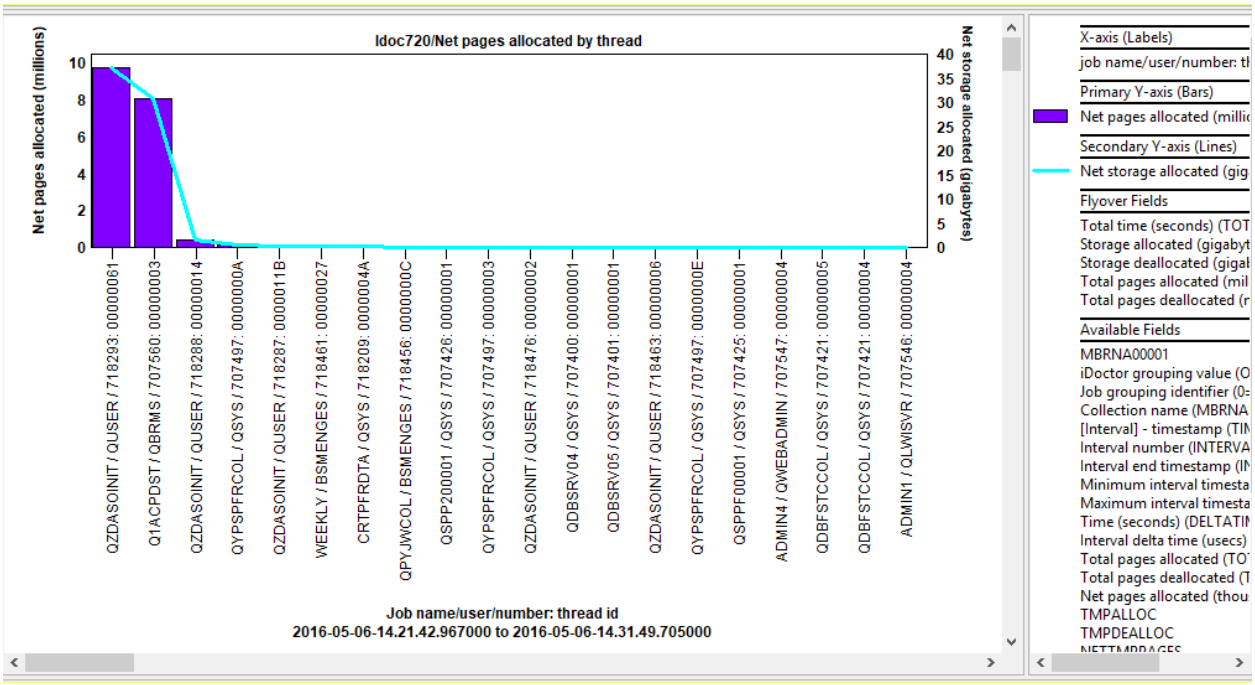


Physical disk I/Os -> Net pages allocated rankings

8.19.13.1 Net pages allocated by thread

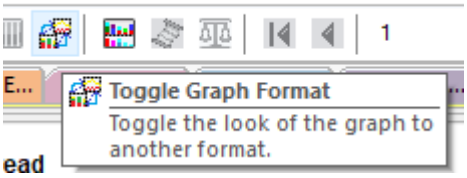
This graph displays the net pages allocated (allocated pages – deallocated pages) ranked by thread/task.

The Y2-axis displays the net storage consumption for these 4K allocations.

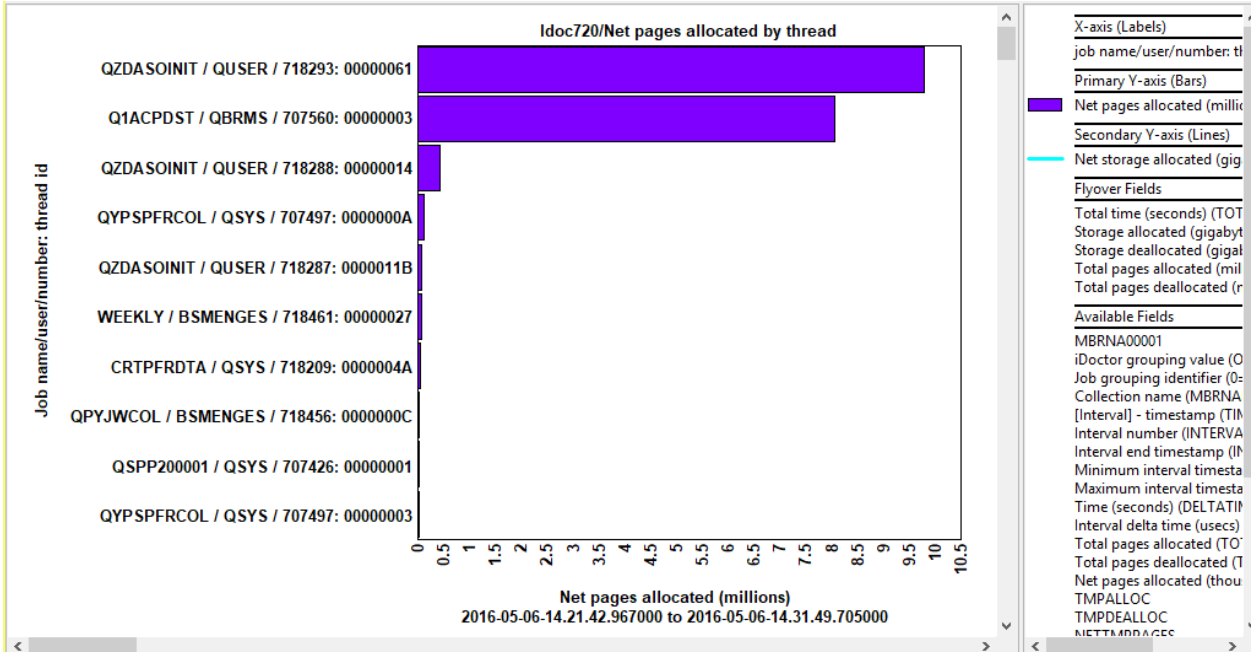


Net pages allocated/deallocated by thread

Tip: Press the Toggle Graph Format toolbar button to switch this graph to a horizontal bar graph if desired.



Toggle graph format button

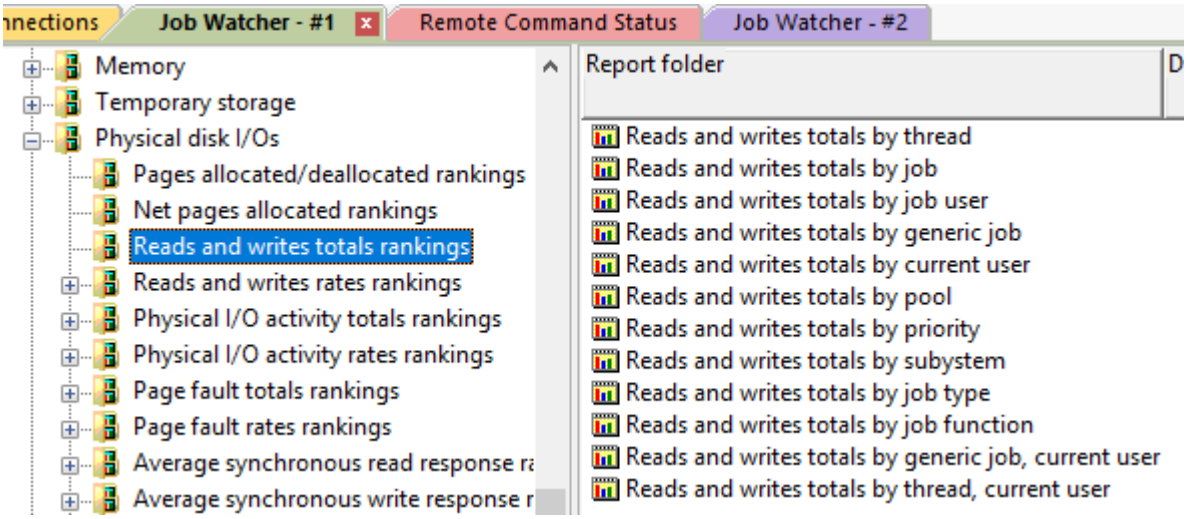


Net pages allocated/deallocated by thread (horizontal bars)

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.14 Reads and writes totals rankings

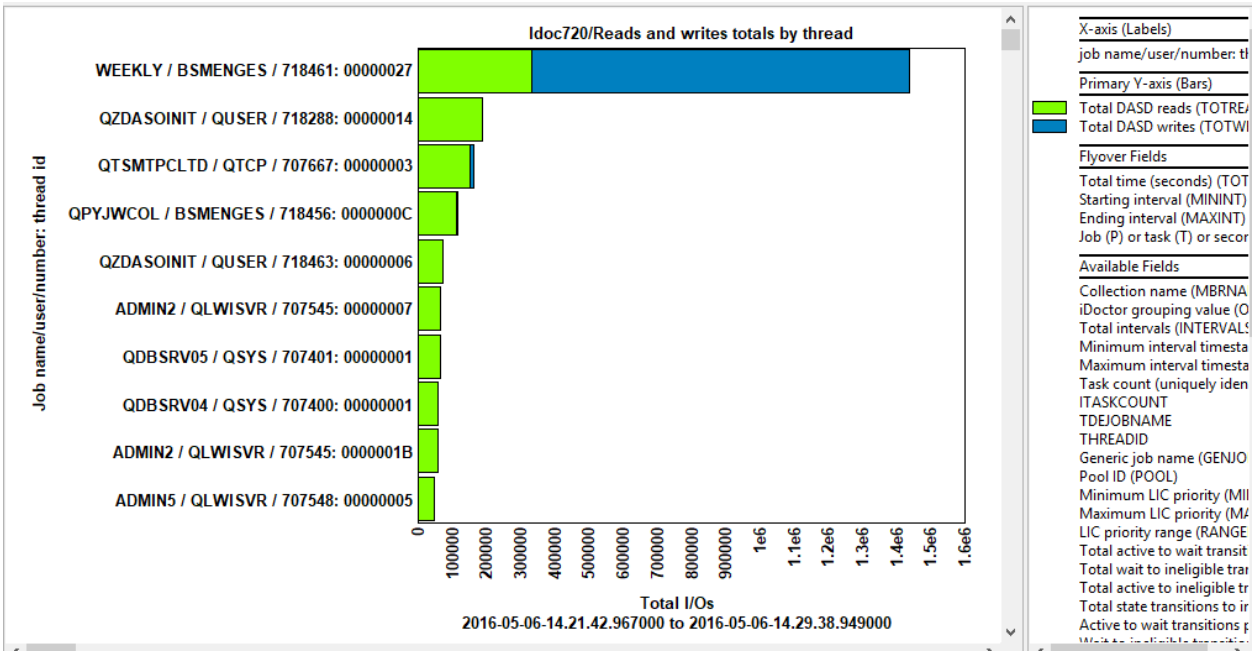
This folder contains the set of job rankings graphs for the **Reads and writes totals** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Reads and writes totals rankings menu.



Physical disk I/Os -> Reads and writes totals rankings

8.19.14.1 Reads and writes totals by thread

This graph displays the total disk reads and writes ranked by thread/task.

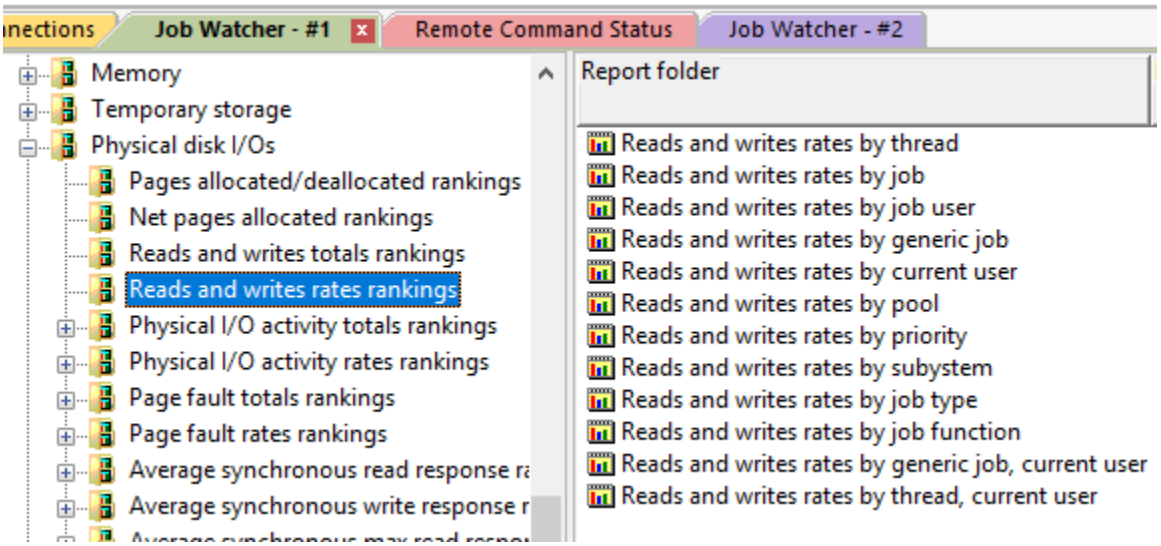


Reads and writes totals by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.15 Reads and writes rates rankings

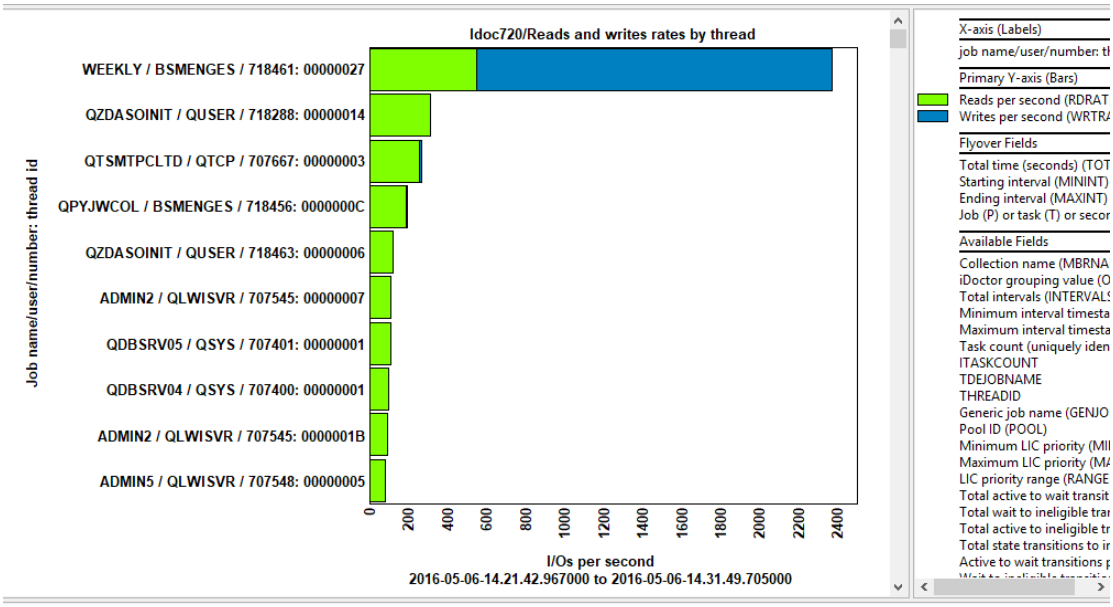
This folder contains the set of job rankings graphs for the **Reads and writes rates** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Reads and writes rates rankings menu.



Physical disk I/Os -> Reads and writes rates rankings

8.19.15.1 Reads and writes rates by thread

This graph displays the disk reads and writes rates per second ranked by thread/task.

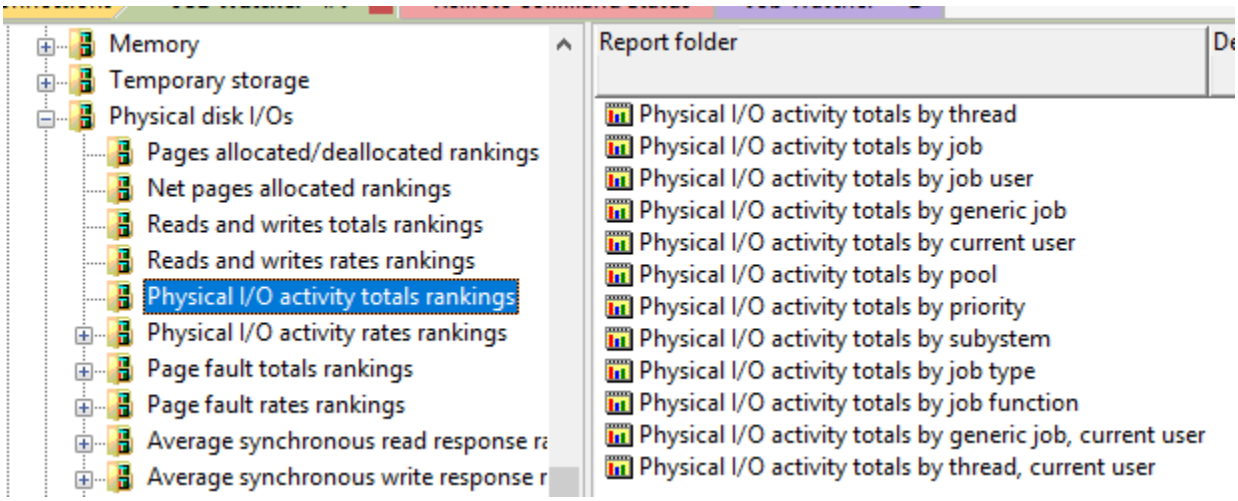


Reads and writes rates by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.16 Physical I/O activity totals rankings

This folder contains the set of job rankings graphs for the **Physical I/O activity totals** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Physical I/O activity totals rankings menu.

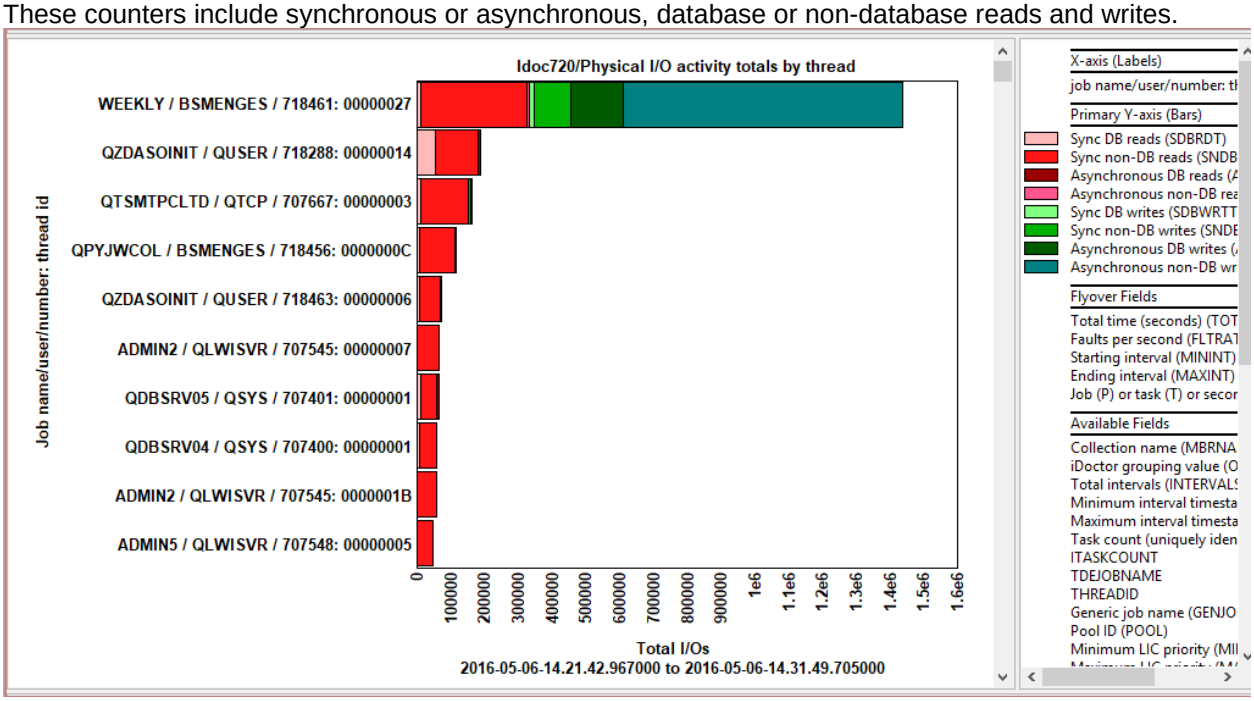


Physical disk I/Os -> Physical I/O activity totals rankings

8.19.16.1 Physical I/O activity totals by thread

This graph shows physical I/O totals ranked by thread.

These counters include synchronous or asynchronous, database or non-database reads and writes.

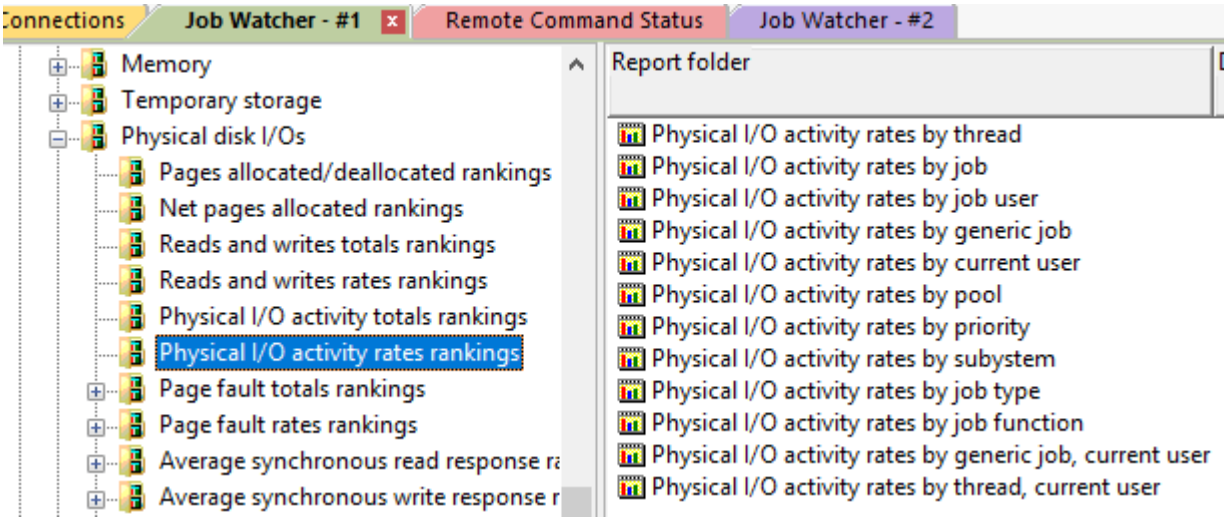


Physical I/O activity totals by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.17 Physical I/O activity rates rankings

This folder contains the set of job rankings graphs for the **Physical I/O activity rates** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Physical I/O activity rates rankings menu.

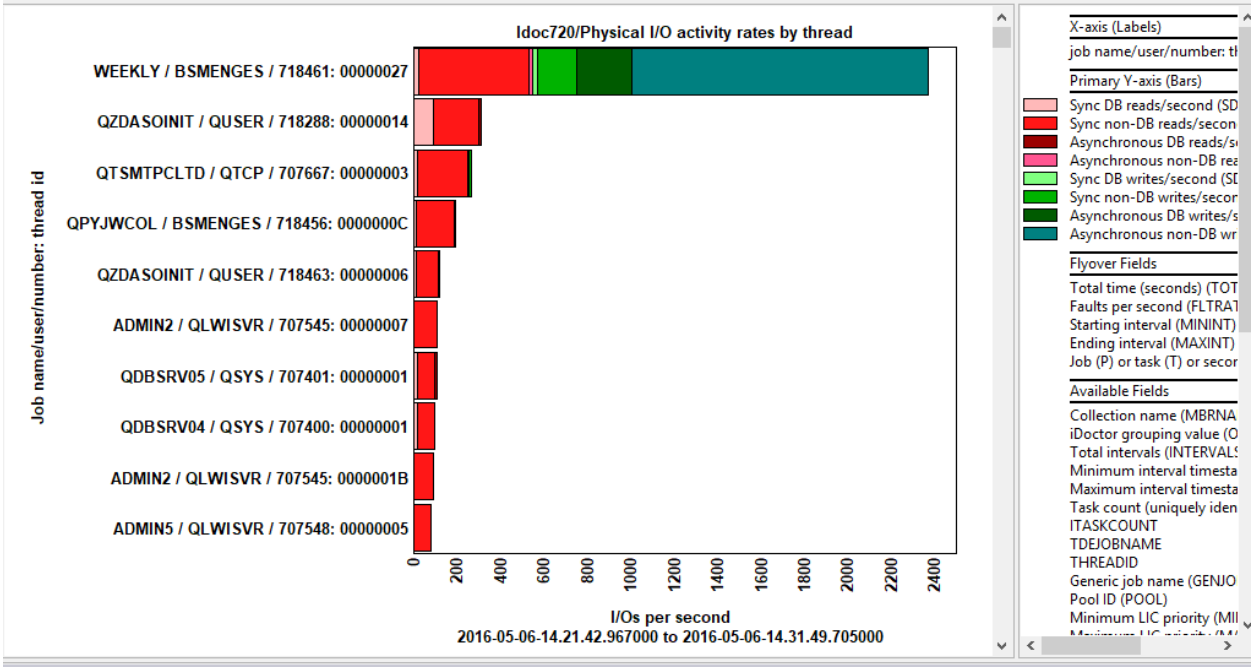


Physical disk I/Os -> Physical I/O activity rates rankings

8.19.17.1 Physical I/O activity rates by thread

This graph shows physical I/O rates per second ranked by thread.

These counters include synchronous or asynchronous, database or non-database reads and writes.

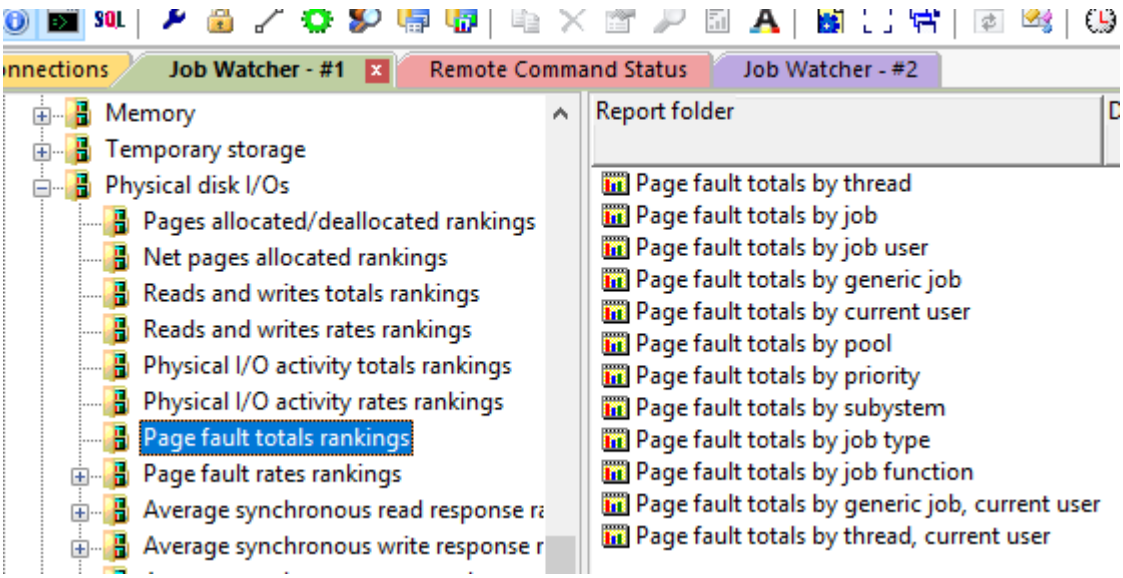


Physical I/O activity rates by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.18 Page fault totals rankings

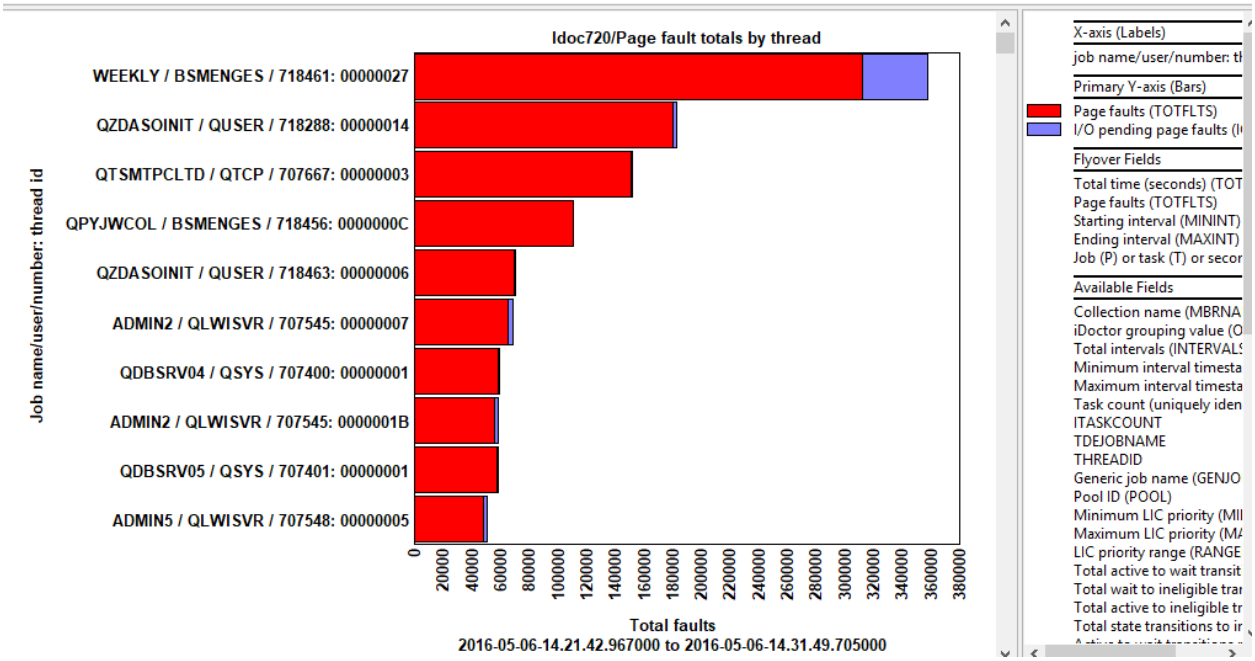
This folder contains the set of job rankings graphs for the **Page fault totals** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Page fault totals rankings menu.



Physical disk I/Os -> Page fault totals rankings

8.19.18.1 Page fault totals by thread

This graph shows page faults and IO pending page faults ranked by thread.

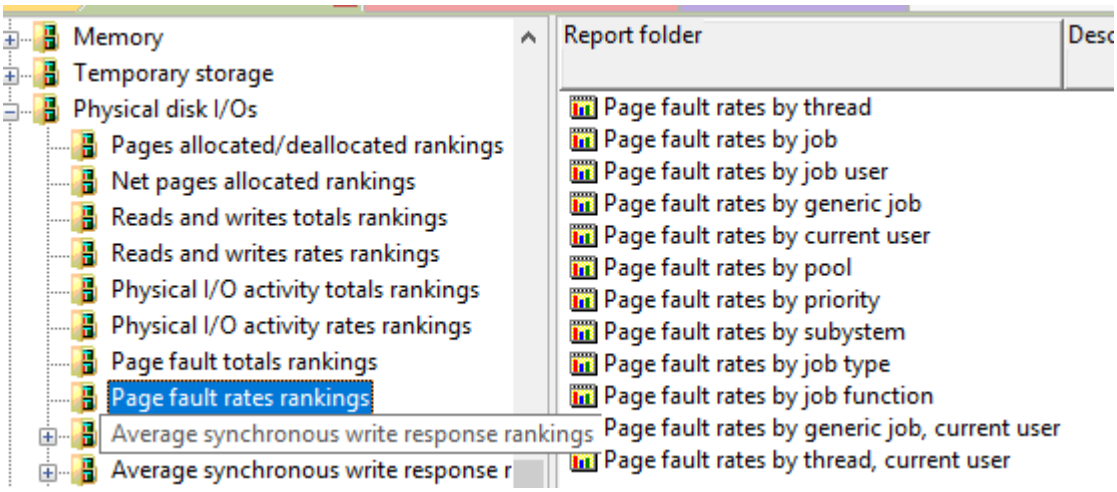


Page fault totals by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.19 Page fault rates rankings

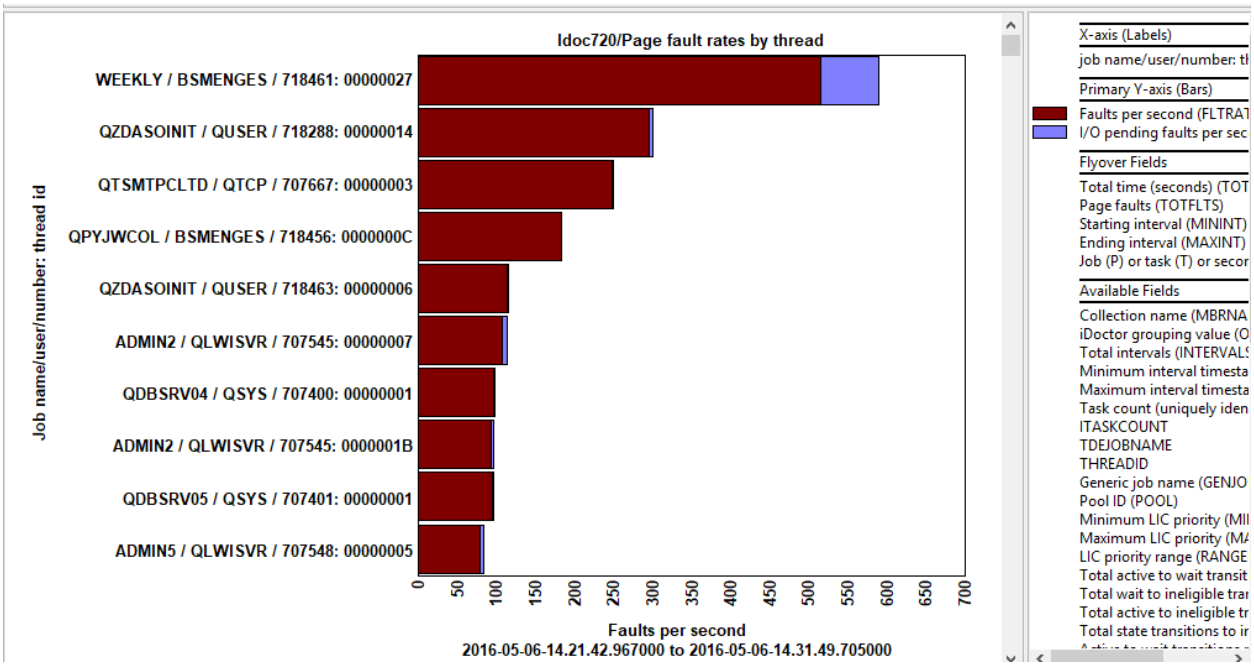
This folder contains the set of job rankings graphs for the **Page fault rates** graph. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Page fault totals rankings menu.



Physical disk I/Os -> Page fault rates rankings

8.19.19.1 Page fault rates by thread

This graph shows page faults per second and IO pending page faults per second ranked by thread.



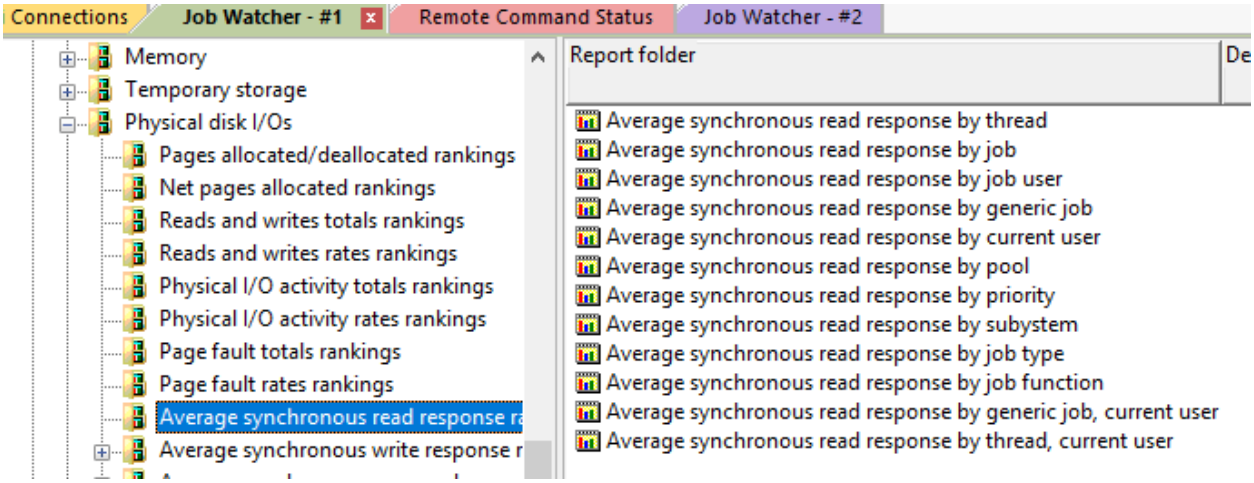
Page fault rates by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.20 Average synchronous read response rankings

This folder contains job rankings graphs based on average synchronous read response times. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Average synchronous read response rankings menu.

This folder requires that the Collection summary be ran first in order to appear.

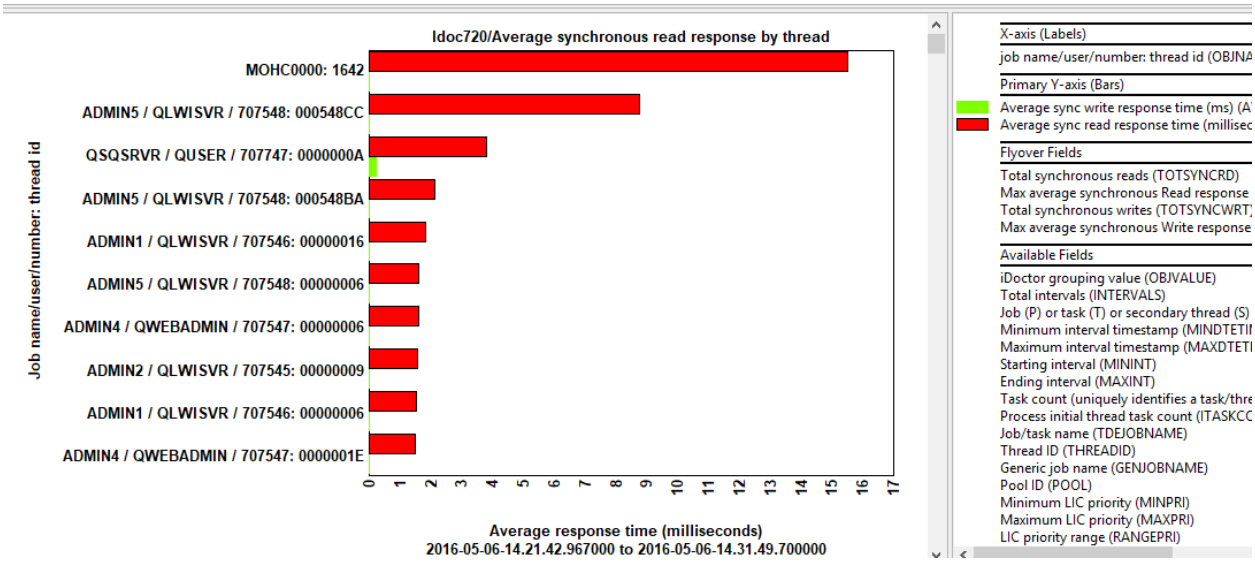


Physical disk I/Os -> Average synchronous read response rankings

8.19.20.1 Average synchronous read response by thread

This graph includes the average synchronous read and write response times ranked by the thread/tasks having the highest average synchronous **read** response time. All times are shown in milliseconds.

The bars in these graphs are shown side-by-side.



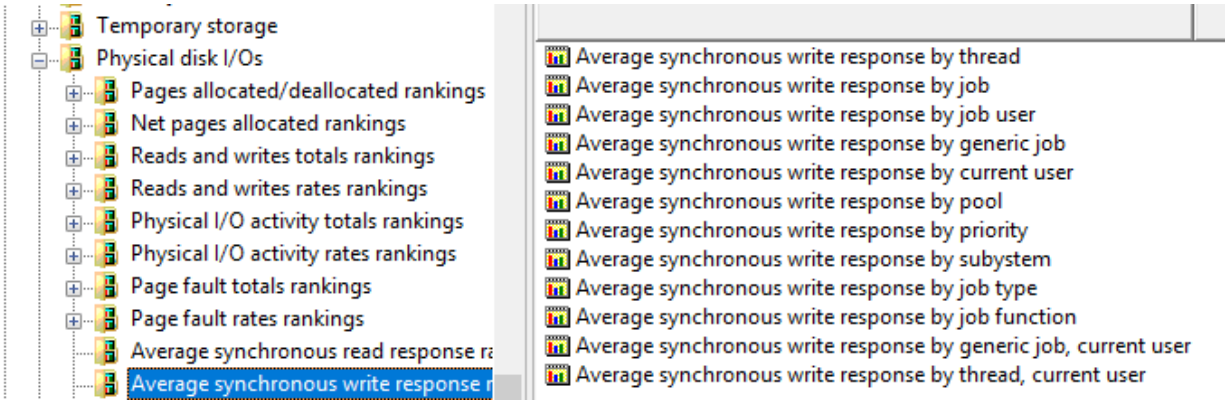
Average synchronous read response by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.21 Average synchronous write response rankings

This folder contains job rankings graphs based on average synchronous write response times. These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Average synchronous write response rankings menu.

This folder requires that the Collection summary be ran first in order to appear.

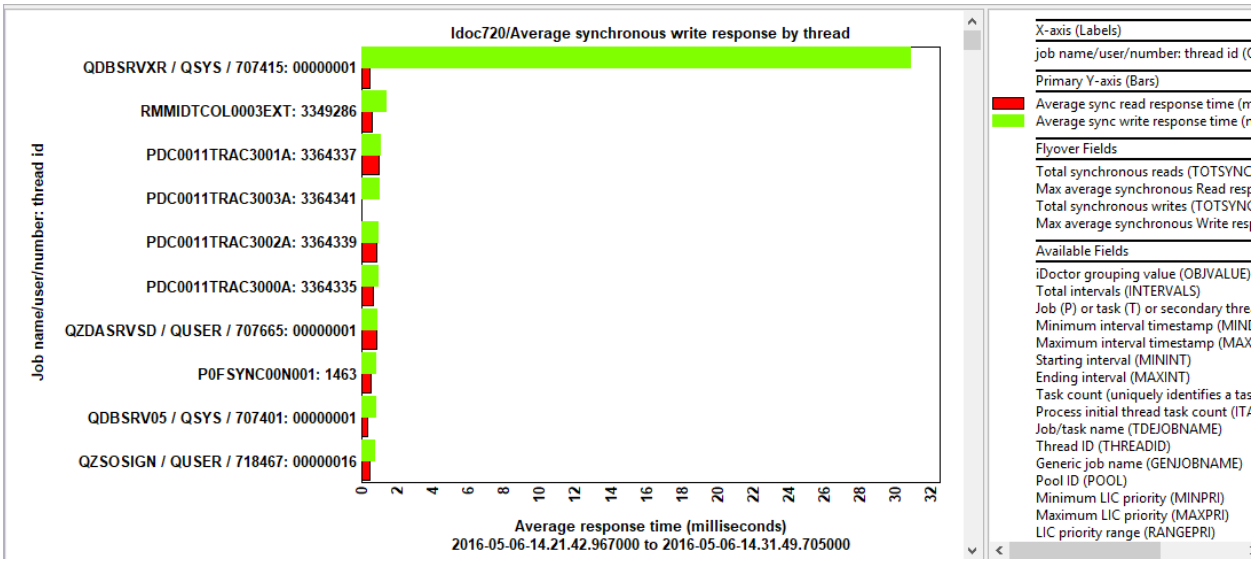


Physical disk I/Os -> Average synchronous write response rankings

8.19.21.1 Average synchronous write response by thread

This graph includes the average synchronous read and write response times ranked by the thread/tasks having the highest average synchronous **write** response time. All times are shown in milliseconds.

The bars in these graphs are shown side-by-side.



Average synchronous write response by thread

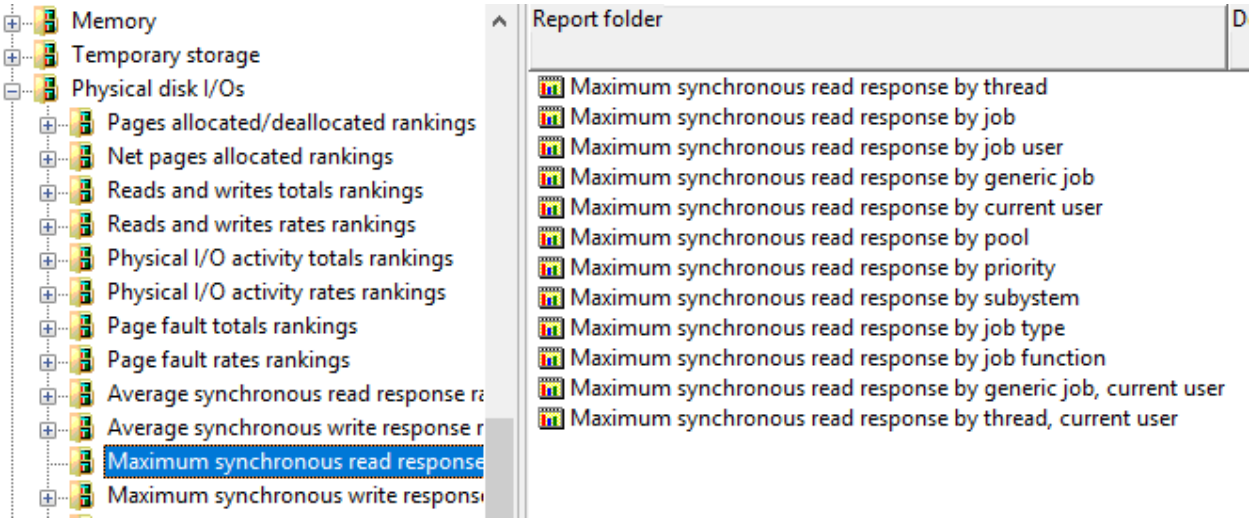
For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.22 Maximum synchronous read response rankings

This folder contains job rankings graphs based on maximum synchronous read response times. This compares the worst read response time for each grouping with the average read response time.

These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Maximum synchronous read response rankings menu.

This folder requires that the Collection summary be ran first in order to appear.

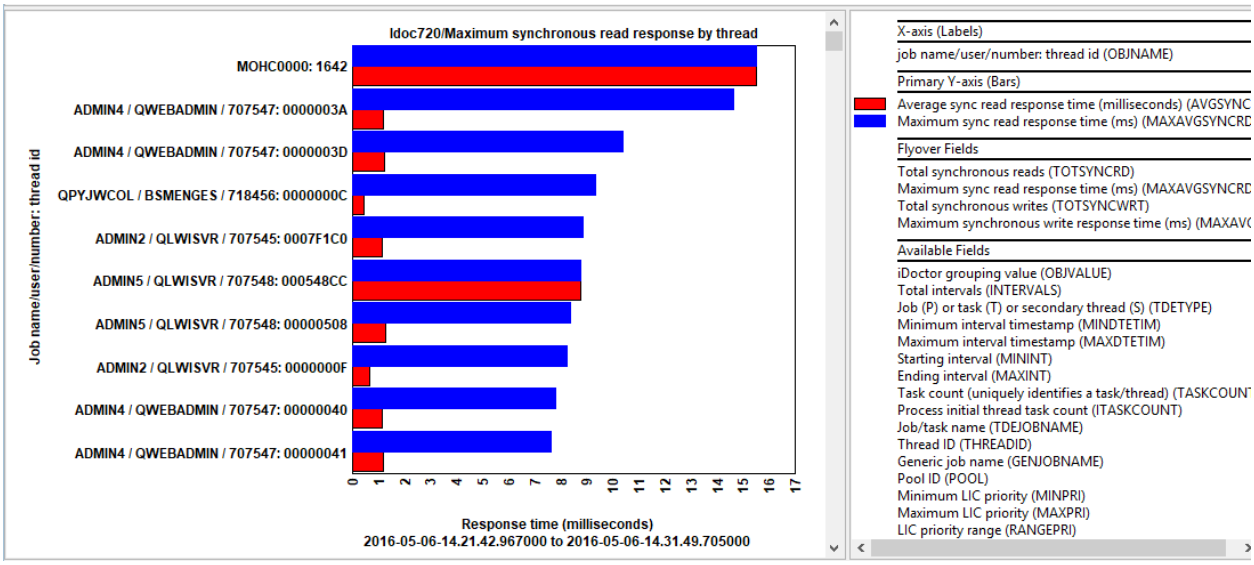


Physical disk I/Os -> Maximum synchronous read response rankings

8.19.22.1 Maximum synchronous read response by thread

This graph includes the average synchronous read response times and the maximum synchronous read response times ranked by the thread/tasks having the worst synchronous read response time. All times are shown in milliseconds.

The bars in these graphs are shown side-by-side.



Maximum synchronous read response by thread

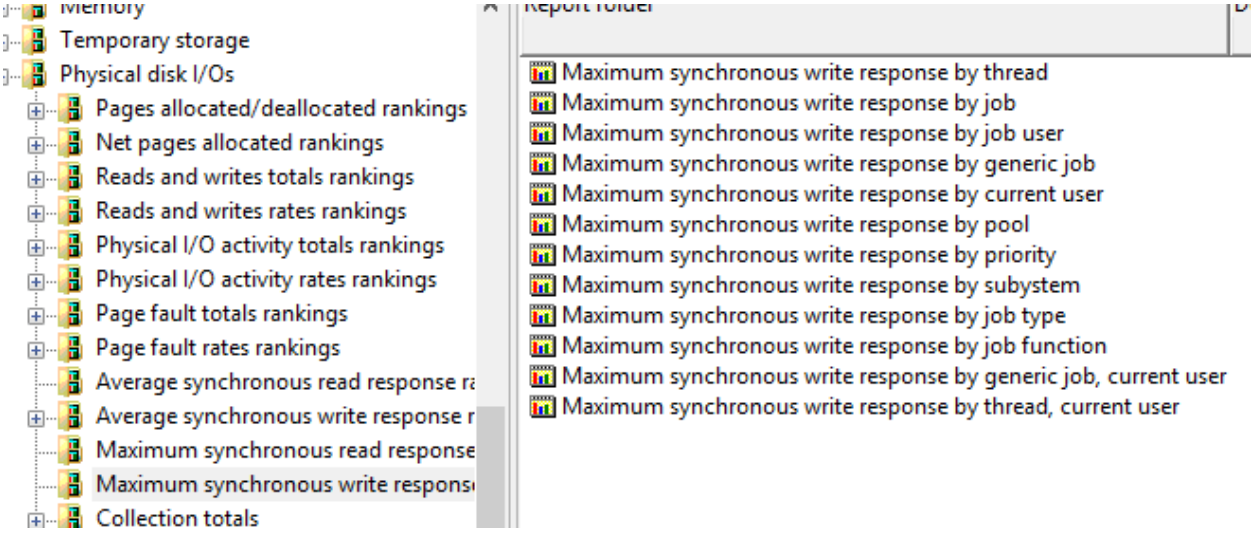
For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.23 Maximum synchronous write response rankings

This folder contains job rankings graphs based on maximum synchronous write response times. This compares the worst write response time for each grouping with the average write response time.

These graphs can either be accessed from this folder or as a drill-down (based on a time selection) using the Rankings -> Physical disk I/Os -> Maximum synchronous write response rankings menu.

This folder requires that the Collection summary be ran first in order to appear.

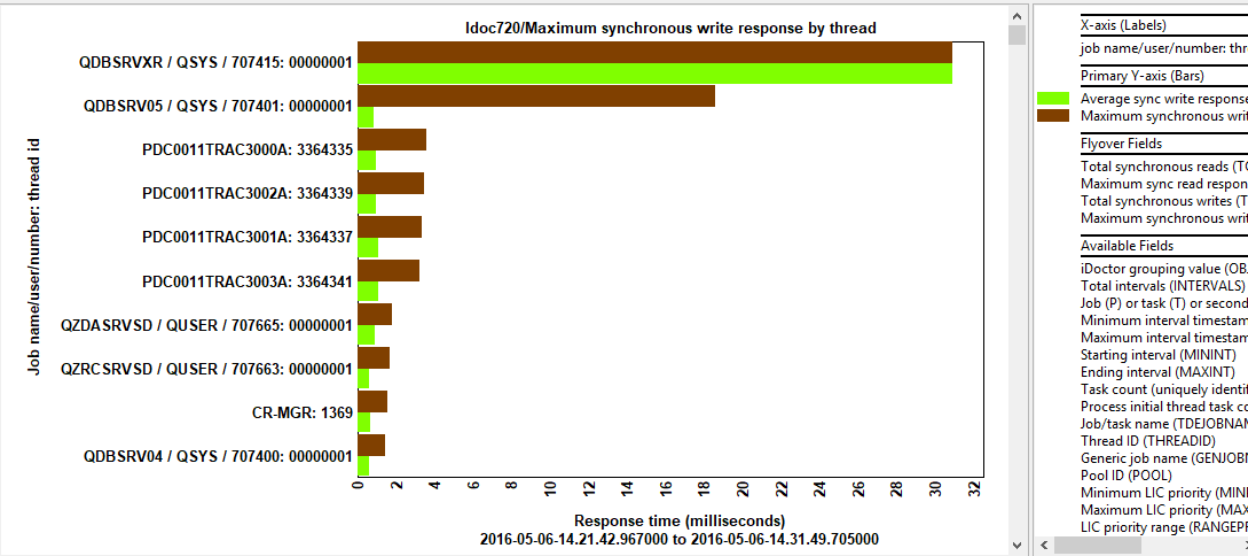


Physical disk I/Os -> Maximum synchronous write response rankings

8.19.23.1 Maximum synchronous write response by thread

This graph includes the average and maximum synchronous write response times ranked by the thread/tasks having the worst maximum synchronous write response time. All times are shown in milliseconds.

The bars in these graphs are shown side-by-side:

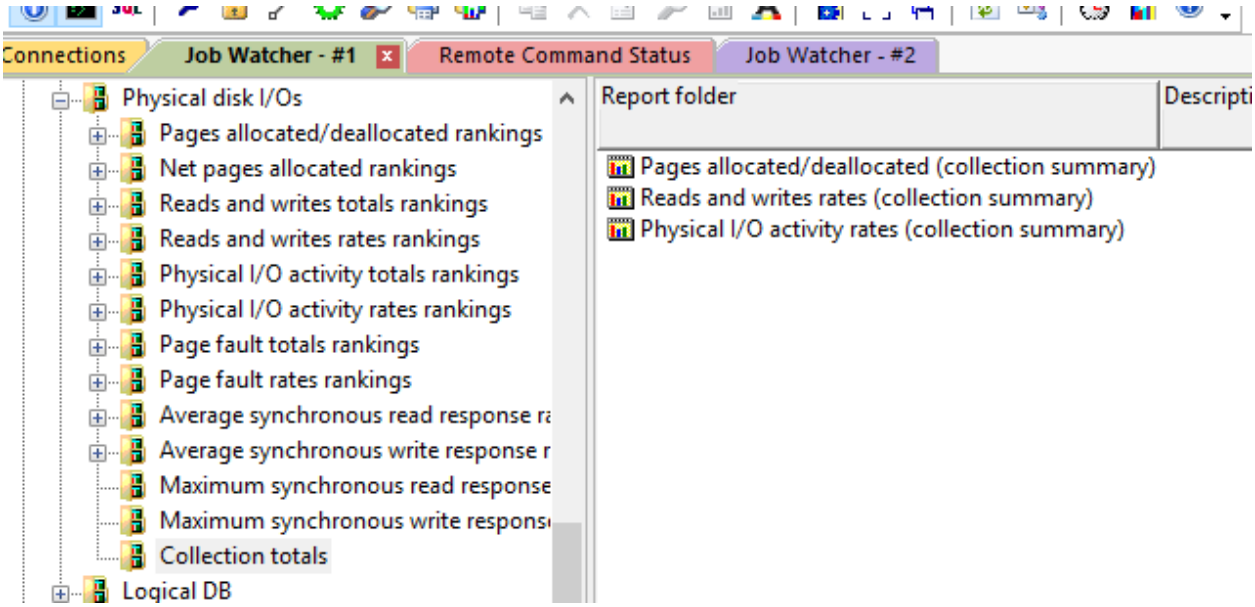


Maximum synchronous write response by thread

For more information on the job groupings provided by these graphs see the Pages allocated/deallocated rankings graphs section.

8.19.24 Collection totals

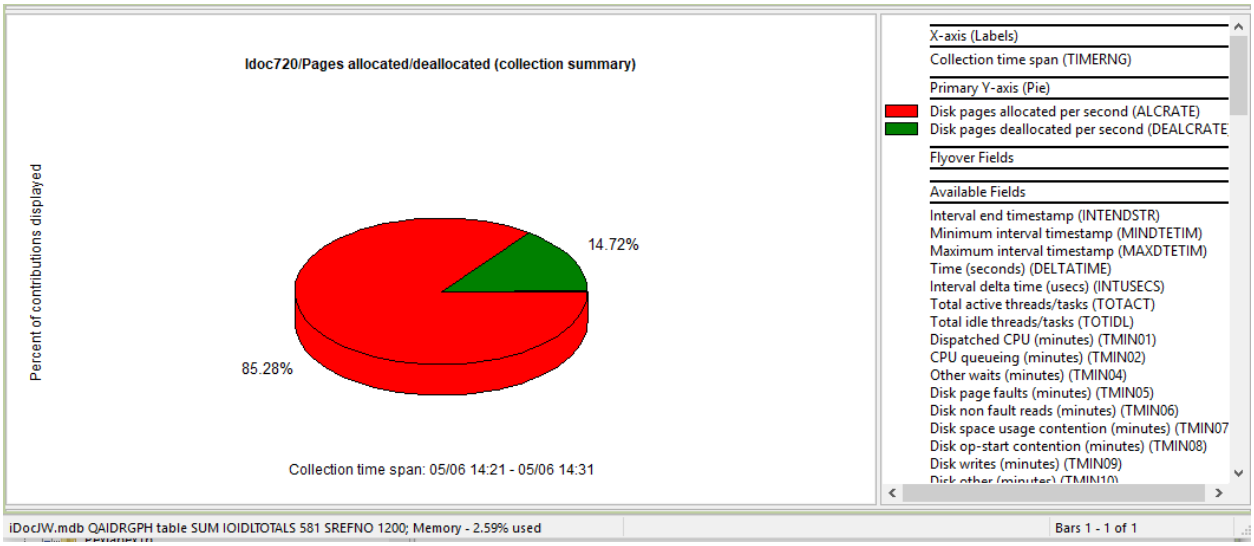
This folder contains a set of pie charts that show high-level statistics about the entire collection related to Physical disk I/Os.



Physical disk I/Os -> Collection totals

8.19.24.1 Pages allocated/deallocated (collection summary)

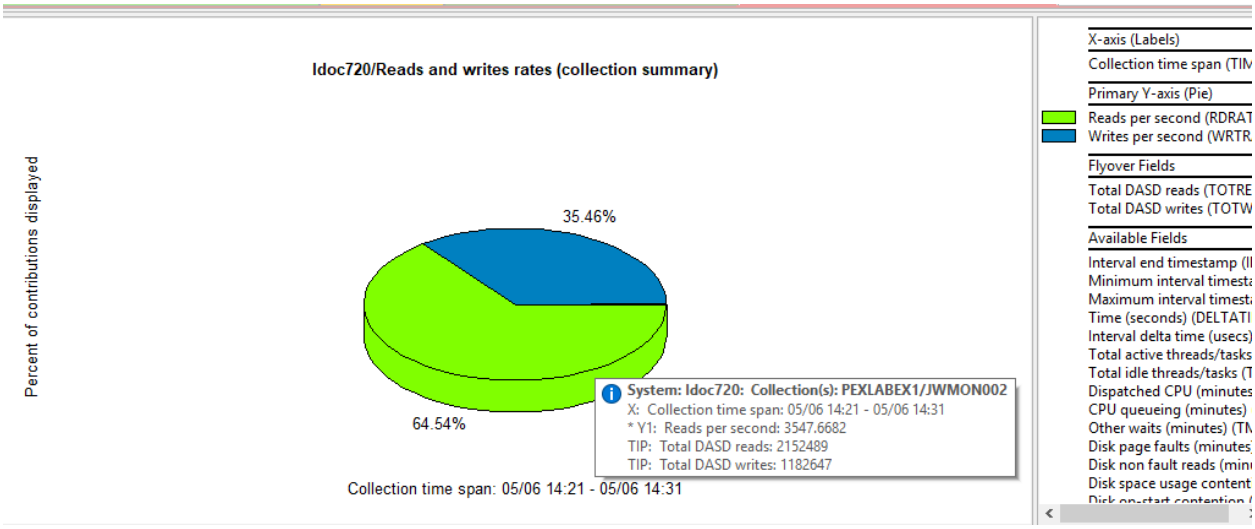
This pie chart simply compares pages allocated vs deallocated in the entire collection.



Pages allocated/deallocated (collection summary)

8.19.24.2 Reads and writes rates (collection summary)

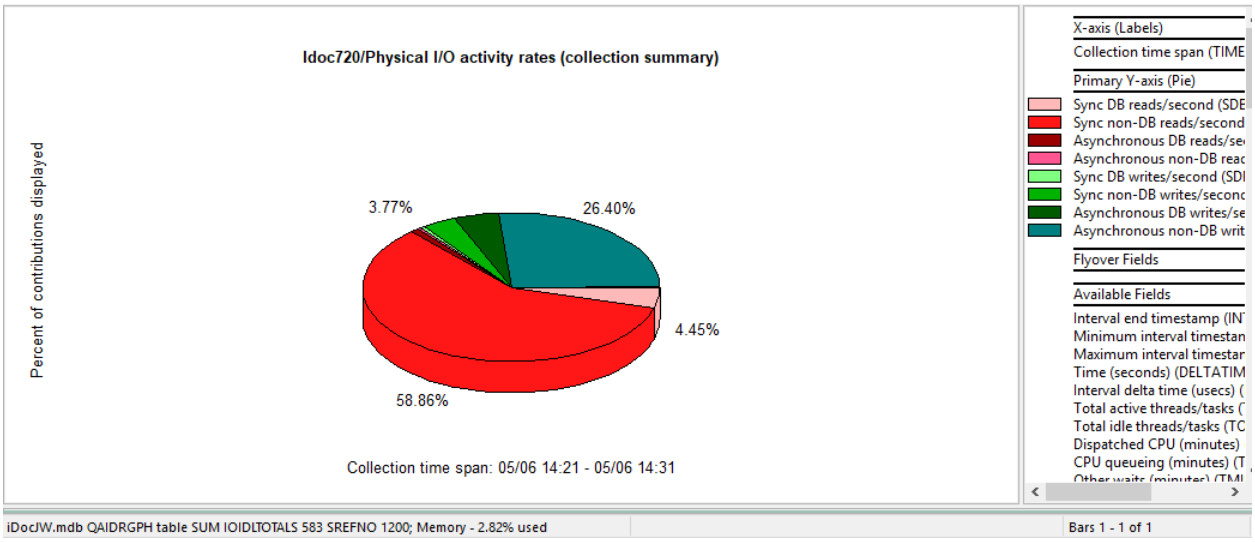
This pie chart compares disk read rates vs write rates for the entire collection.



Reads and writes rates (collection summary)

8.19.24.3 Physical disk I/O rates (collection summary)

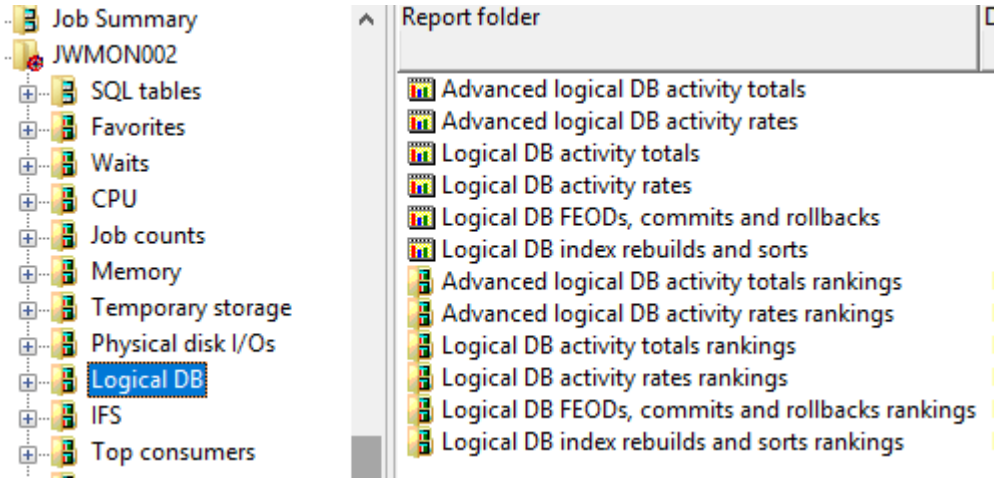
This pie chart compares physical disk I/O metric rates for the entire collection.



Physical I/O activity rates (collection summary)

8.20 Logical DB

This folder contains graphs related to logical database statistics. Both overview graphs and ranking graphs (by job) are provided.



Logical DB

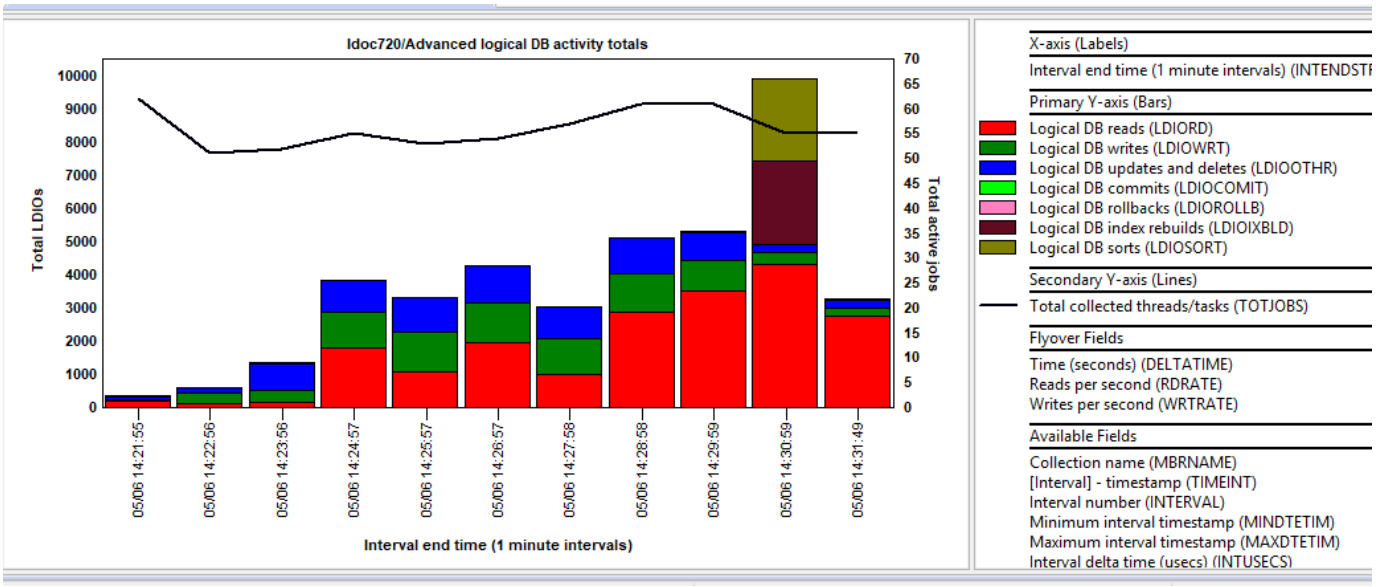
8.20.1 Advanced logical DB activity totals

This graph shows several types of logical DB metrics added up across all jobs in the collection and shown over time.

The metrics shown in this graph include the following:

1) Reads
2) Writes
3) Updates and deletes (combined)
4) Commits

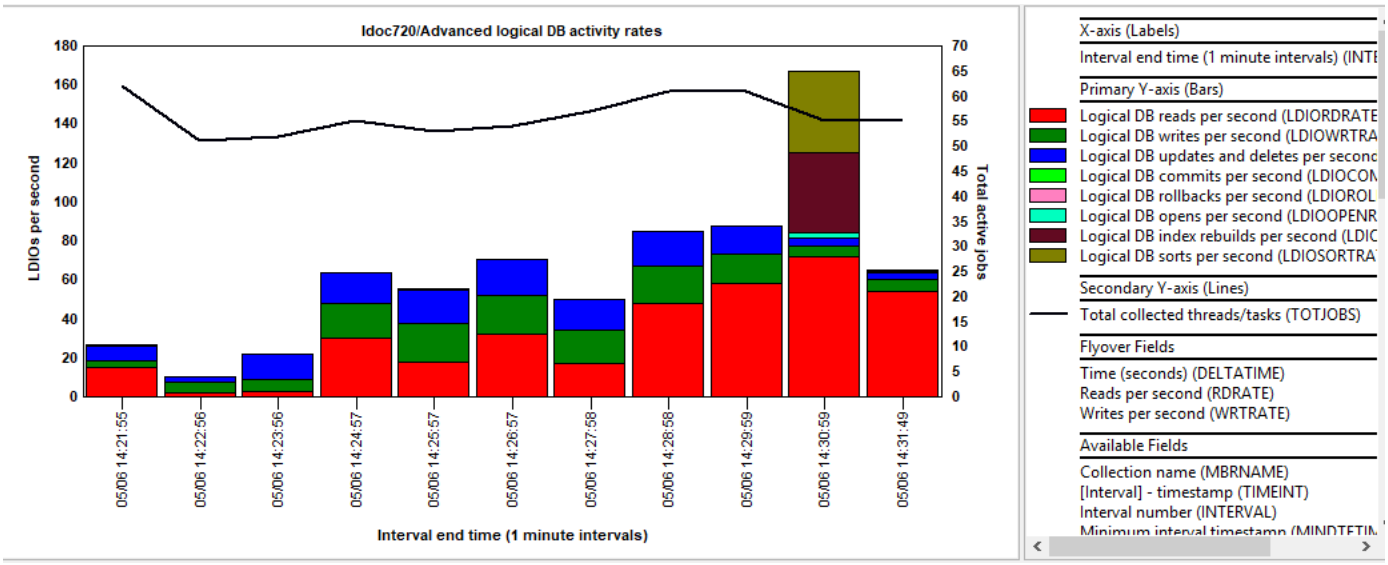
5) Rollbacks
6) Index rebuilds
7) Sorts



Advanced logical DB activity totals

8.20.2 Advanced logical DB activity rates

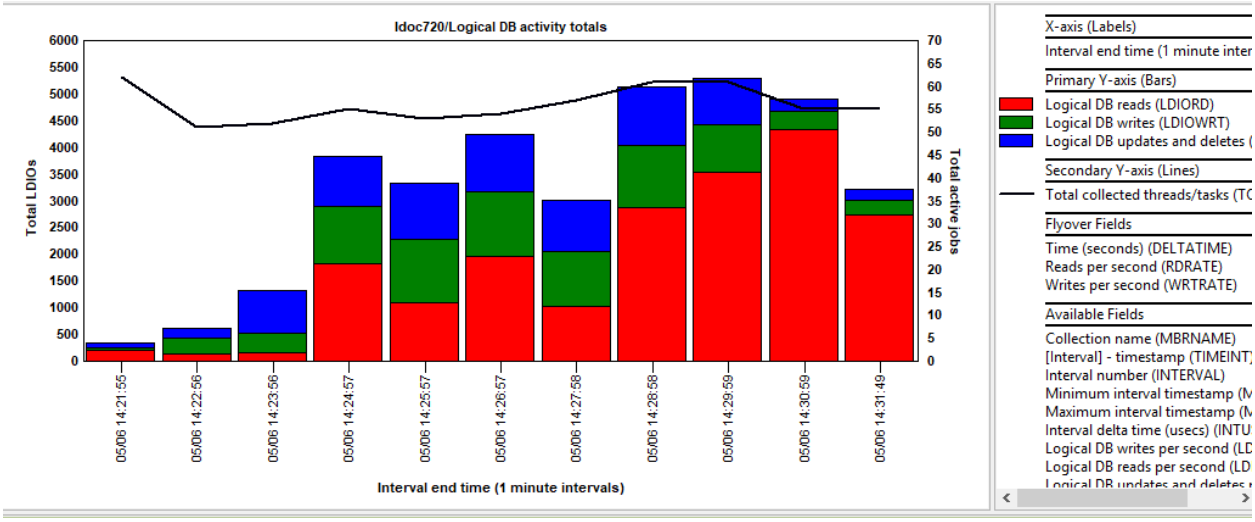
This graph shows several types of logical DB rates per second across all jobs in the collection and shown over time.



Advanced logical DB activity rates

8.20.3 Logical DB activity totals

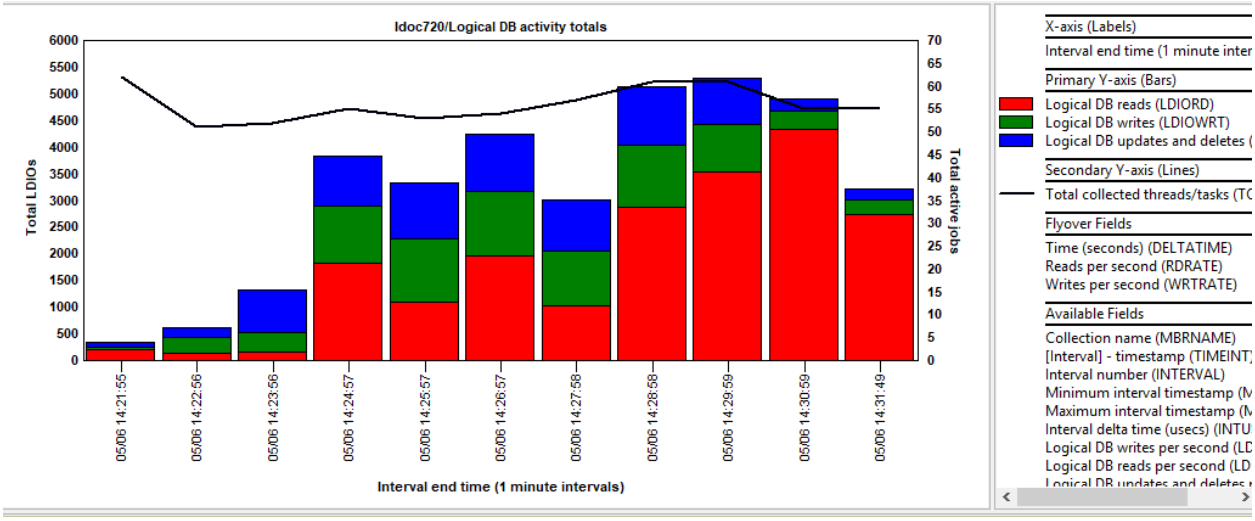
This graph is the same as the Advanced logical DB activity totals graph except it only contains the first 3 metrics shown: Reads, writes and updates + deletes.



Logical DB activity totals

8.20.4 Logical DB activity rates

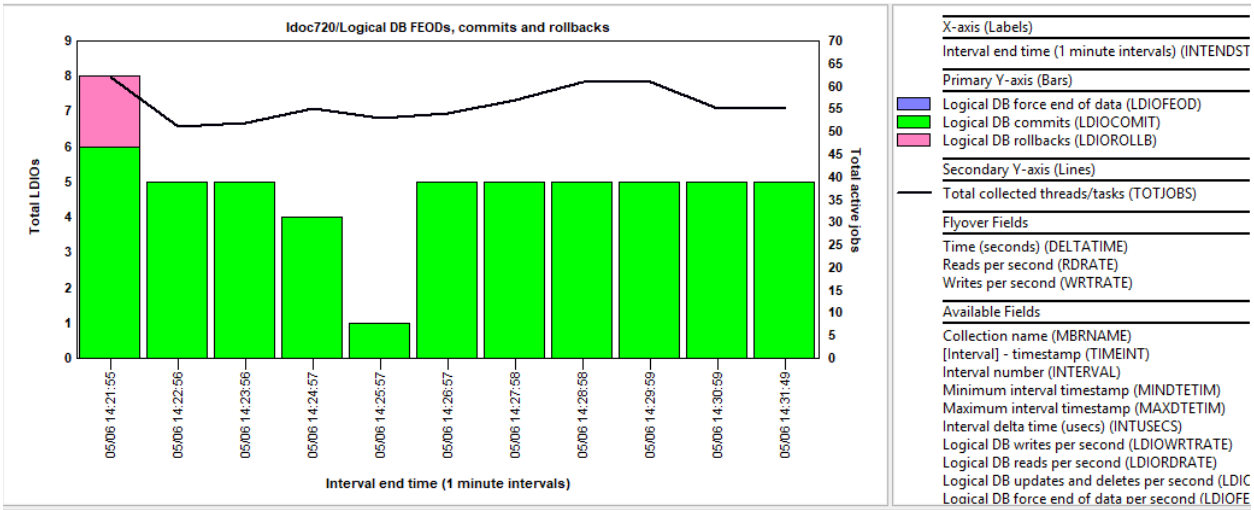
This graph is the same as the Advanced logical DB activity rates graph except it only contains the first 3 metrics shown: Reads, writes and updates + deletes.



Logical DB activity rates

8.20.5 Logical DB FEODs, commit and rollbacks

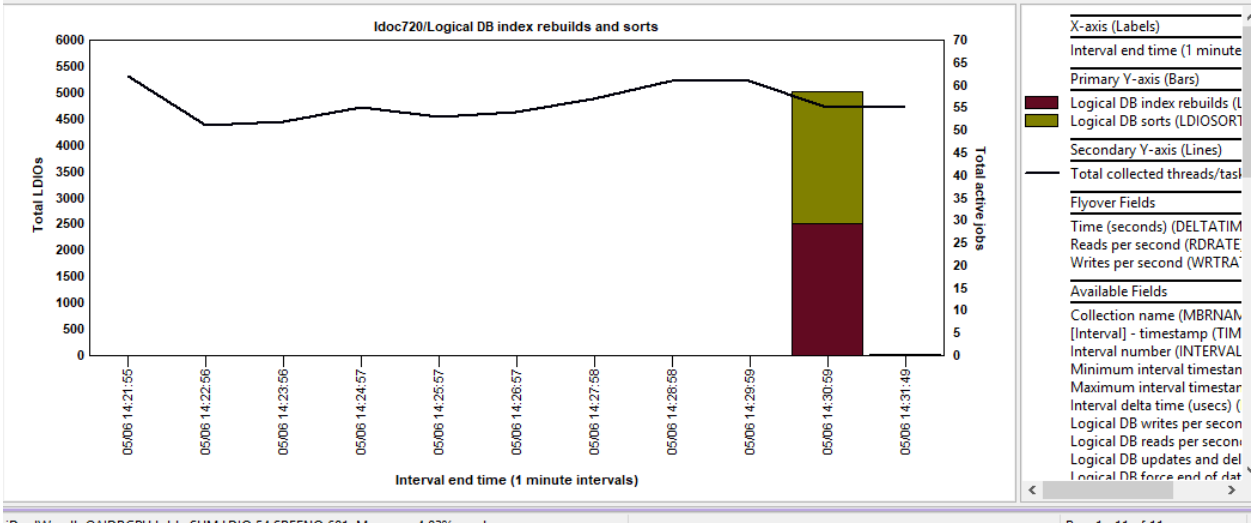
This graph contains totals for just logical DB force end of data, commits and rollbacks over time.



Logical DB FEODs, commit and rollbacks

8.20.6 Logical DB index rebuilds and sorts

This graph contains totals for just logical DB index rebuilds and sorts over time.



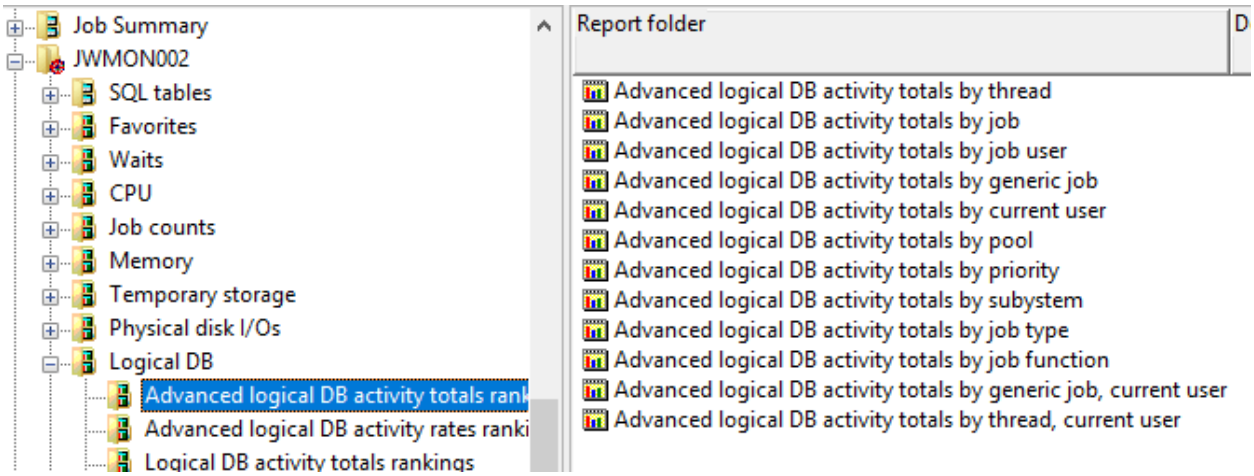
Logical DB index rebuilds and sorts

8.20.7 Advanced Logical DB activity totals rankings

This folder contains a set of job ranking graphs based on logical DB metrics for all jobs in the collection. The metrics shown in these graphs include the following:

1) Reads
2) Writes
3) Updates and deletes (combined)
4) Commits
5) Rollbacks
6) Index rebuilds

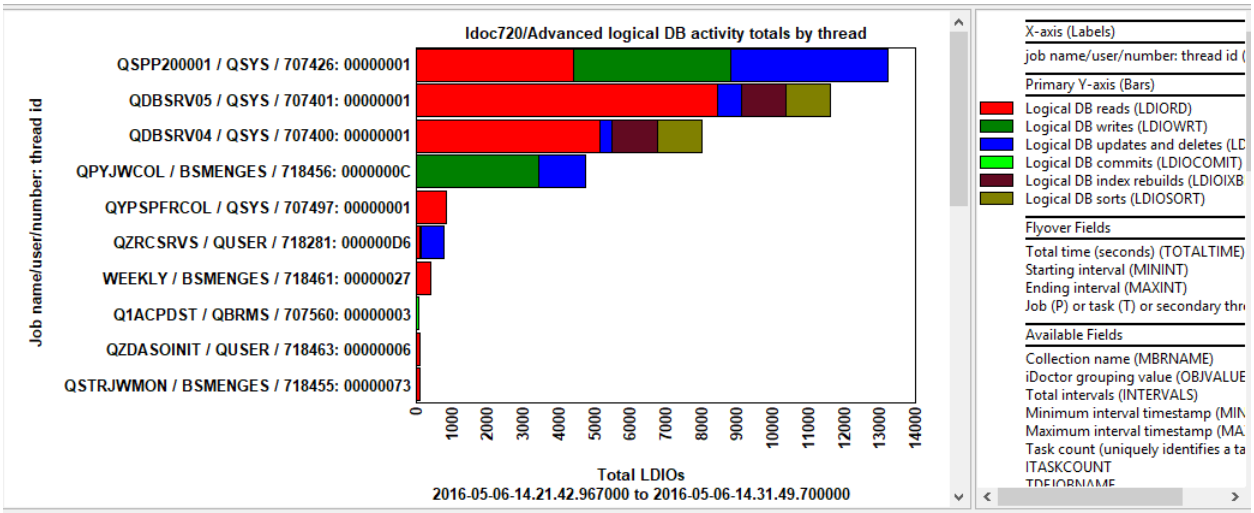
7) Sorts



Logical DB -> Advanced logical DB activity totals rankings

8.20.7.1 Advanced logical DB activity totals by thread

This graph displays several types of logical DB activity totals ranked by thread or system task.

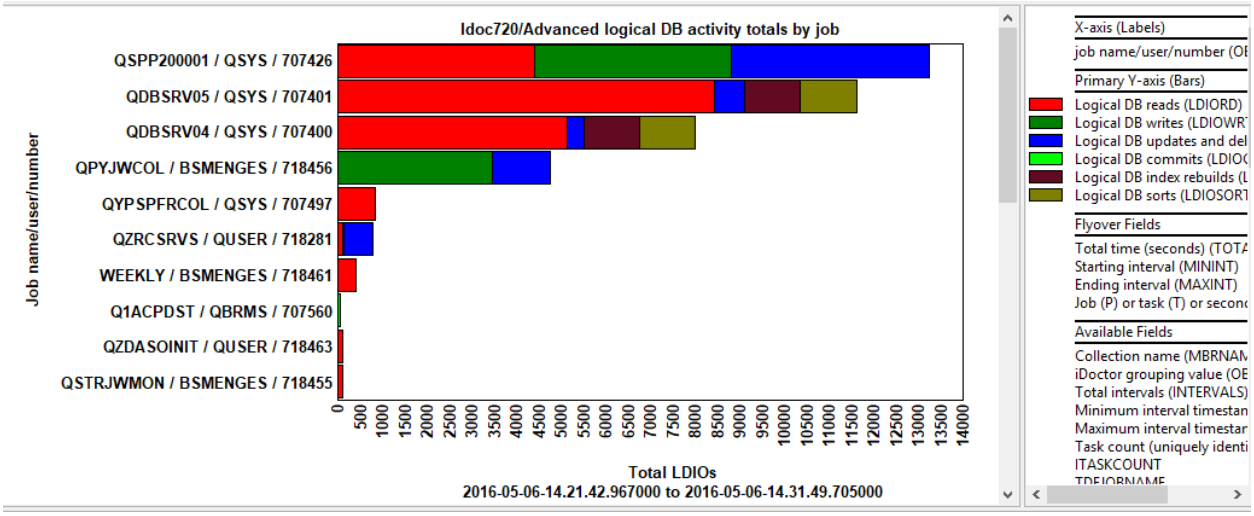


Advanced logical DB activity totals by thread

8.20.7.2 Advanced logical DB activity totals by job

This graph displays several types of logical DB activity totals ranked by primary thread.

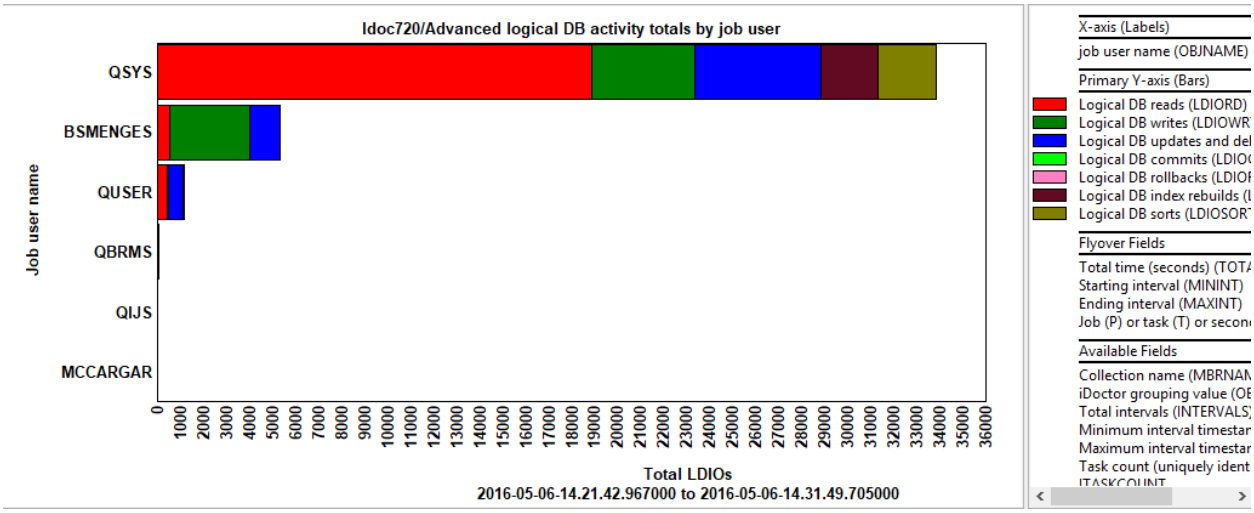
Jobs that have multiple threads are all added up together.



Advanced logical DB activity totals by job

8.20.7.3 Advanced logical DB activity totals by job user

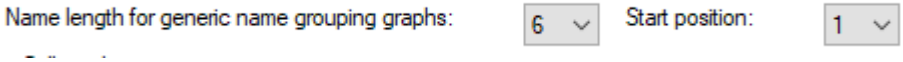
This graph displays several types of logical DB activity totals ranked by job user.

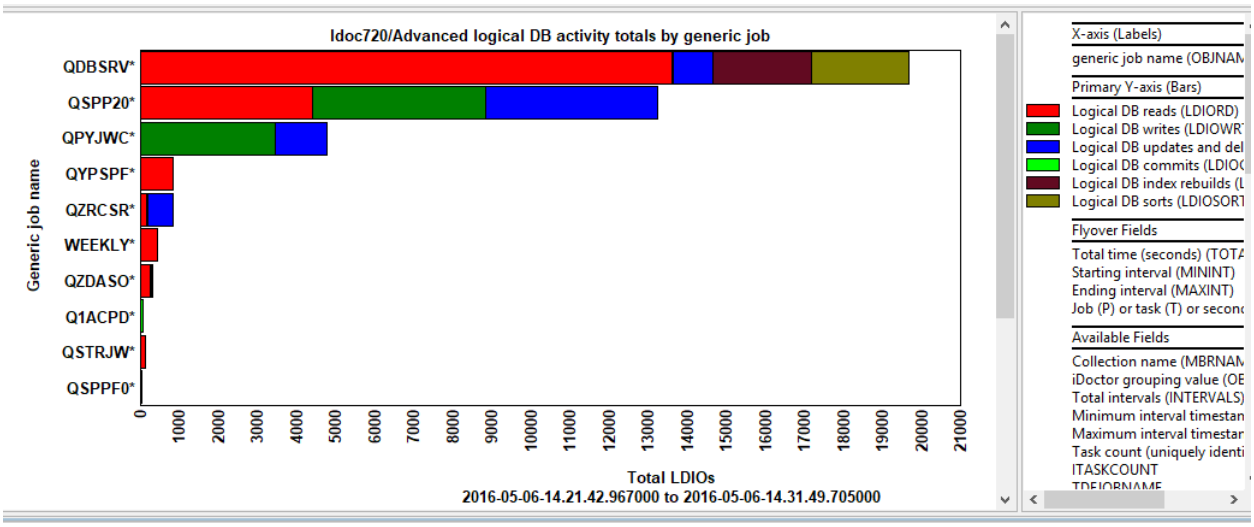


Advanced logical DB activity totals by job user

8.20.7.4 Advanced logical DB activity totals by generic job

This graph displays several types of logical DB activity totals ranked by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

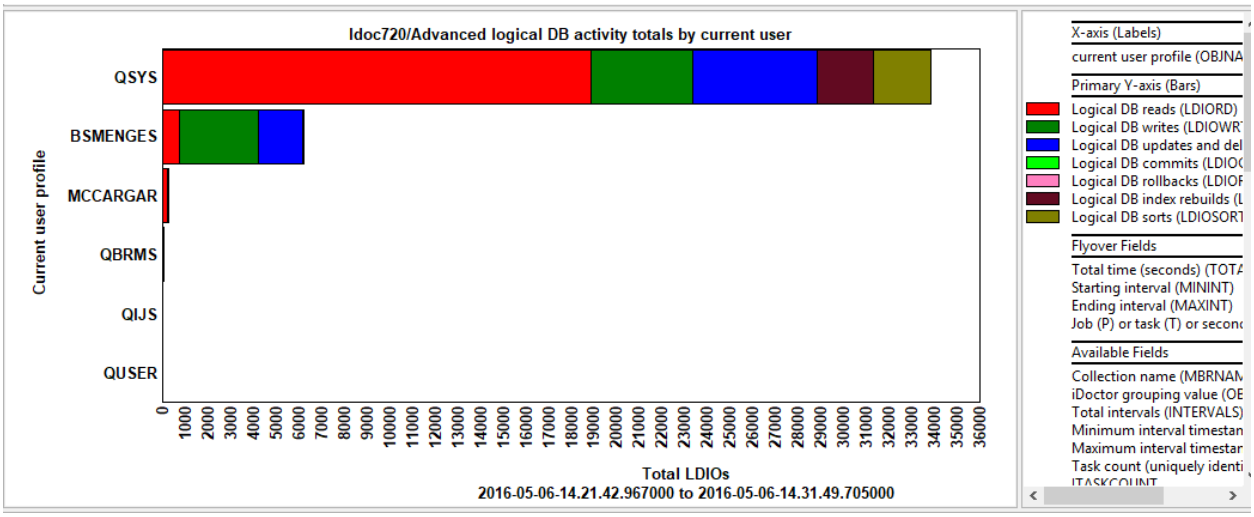




Advanced logical DB activity totals by generic job

8.20.7.5 Advanced logical DB activity totals by current user

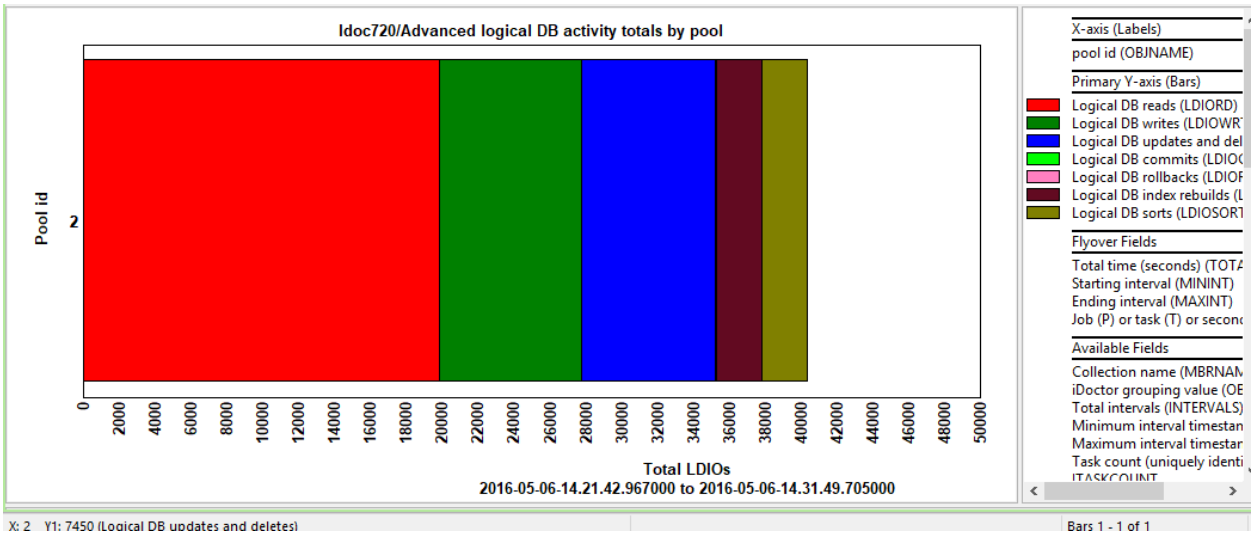
This graph displays several types of logical DB activity totals ranked by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



Advanced logical DB activity totals by current user

8.20.7.6 Advanced logical DB activity totals by pool

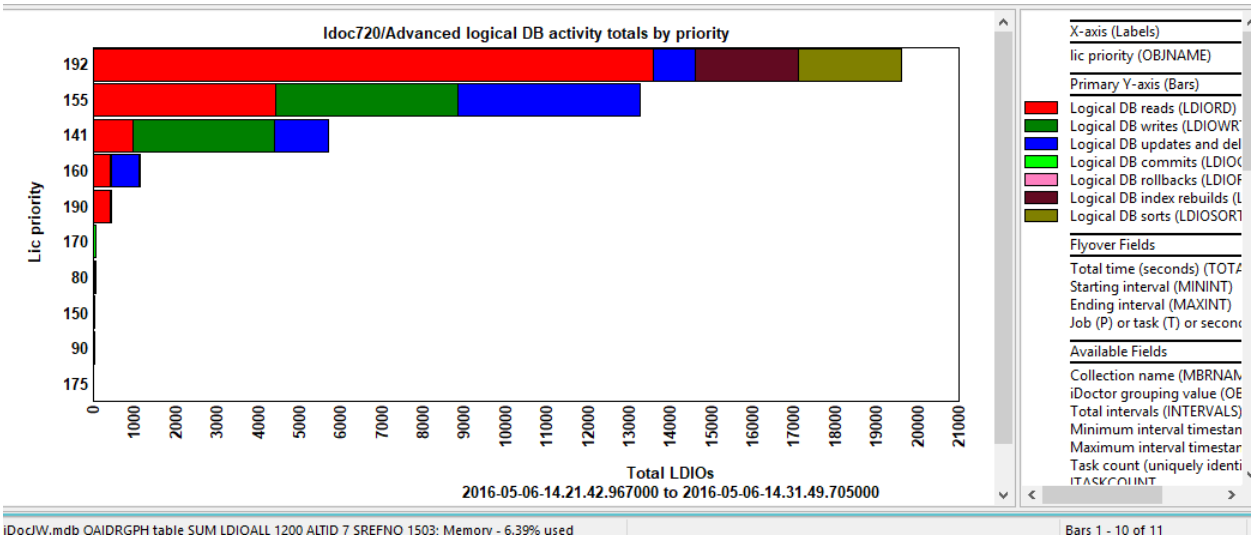
This graph displays several types of logical DB activity totals ranked by memory pool.



Advanced logical DB activity totals by pool

8.20.7.7 Advanced logical DB activity totals by priority

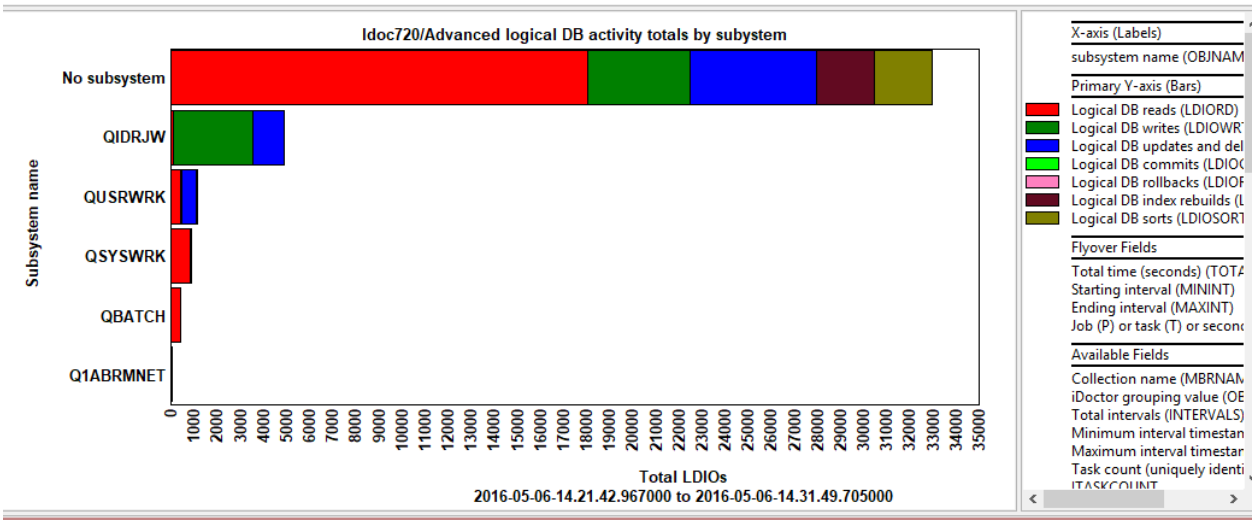
This graph displays several types of logical DB activity totals ranked by LIC priority. XPF priority can be calculated by subtracting 140 for those values shown exceeding 140.



Advanced logical DB activity totals by LIC priority

8.20.7.8 Advanced logical DB activity totals by subsystem

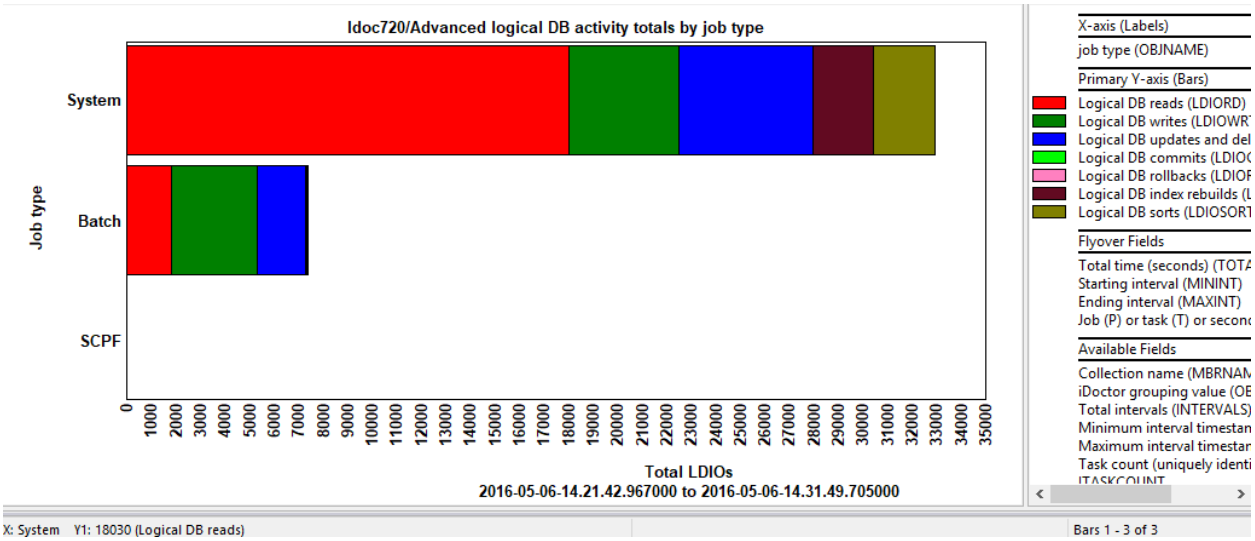
This graph displays several types of logical DB activity totals ranked by subsystem. Job times that had no subsystem listed are grouped into 1 bar called “No subsystem”.



Advanced logical DB activity totals by subsystem

8.20.7.9 Advanced logical DB activity totals by job type

This graph displays several types of logical DB activity totals ranked by job type (such as Batch, interactive, autostart, etc.) Job wait bucket times that had no job type associated with it are grouped into 1 bar called “No job type”.

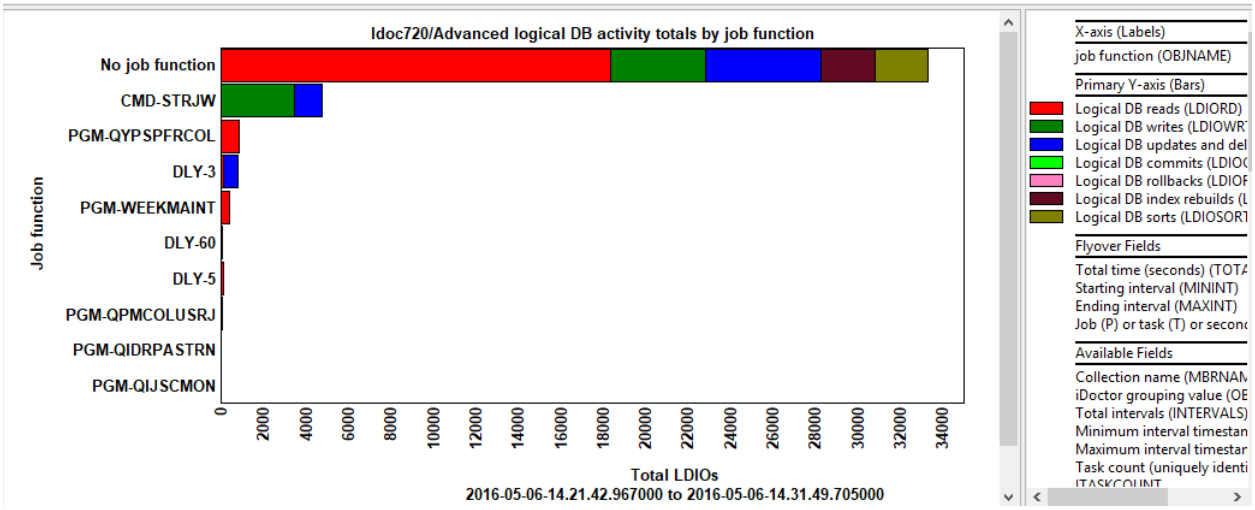


Advanced logical DB activity totals by job type

8.20.7.10 Advanced logical DB activity totals by job function

This graph displays several types of logical DB activity totals ranked by job function. Job function is the same as what you would see normally on the WRKACTJOB command’s Function column. Any wait bucket times that had no job function associated with it are grouped into 1 bar called “No job function”.

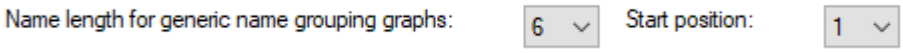
Note: This graph can be misleading in some situations because the current job function for each process is only captured at the end of each Job Watcher interval. The times given are not necessarily 100% from each function listed.



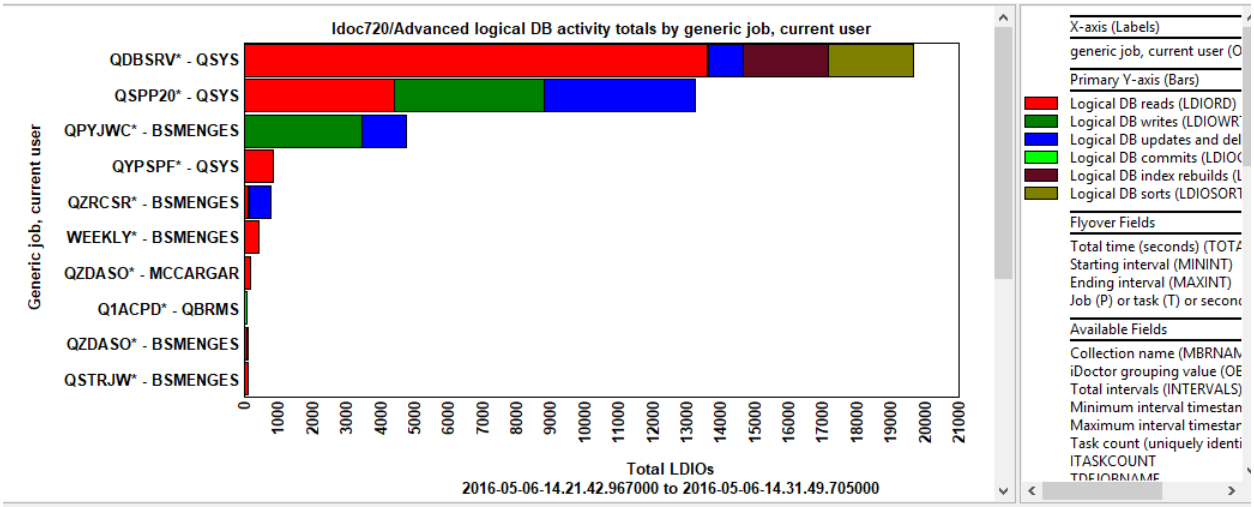
Advanced logical DB activity totals by job function

8.20.7.11 Advanced logical DB activity totals by generic job, current user

This graph displays several types of logical DB activity totals ranked by generic job/current user profile combination. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:



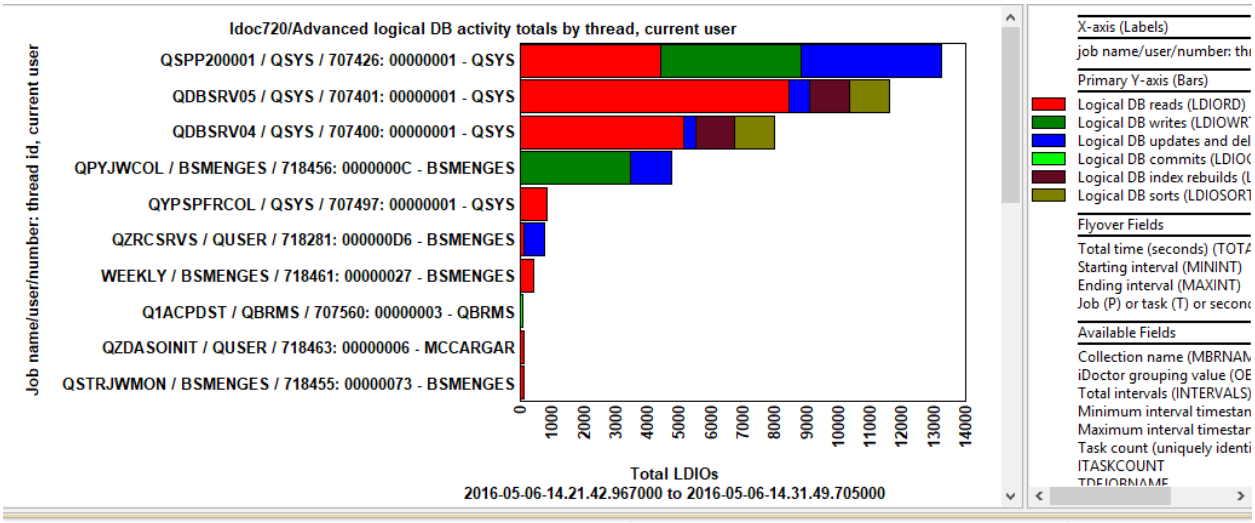
Note: All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



Advanced logical DB activity totals by generic job, current user

8.20.7.12 Advanced logical DB activity totals by thread, current user

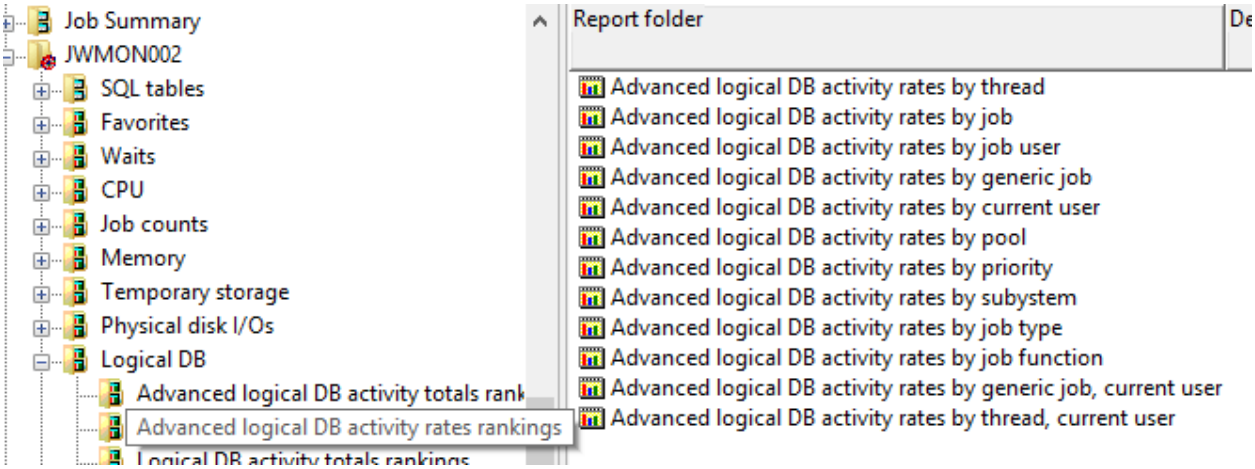
This graph displays several types of logical DB activity totals ranked by thread/current user profile combination. **Note:** All system tasks are grouped together into one "**System tasks"** bar within this report because system tasks do not have current user profiles.



Advanced logical DB activity totals by thread, current user

8.20.8 Advanced Logical DB activity rates rankings

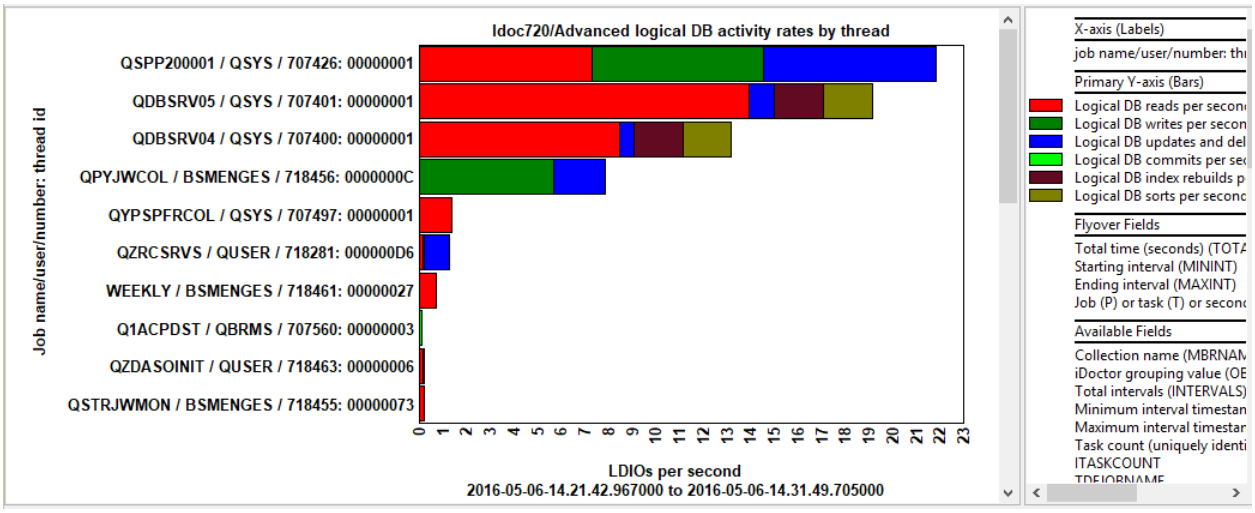
This folder contains a set of job ranking graphs based on logical DB rates per second for all jobs in the collection.



Logical DB -> Advanced logical DB activity rates rankings

8.20.8.1 Advanced logical DB activity rates by thread

This graph displays several types of logical DB activity rates per second ranked by thread or system task.

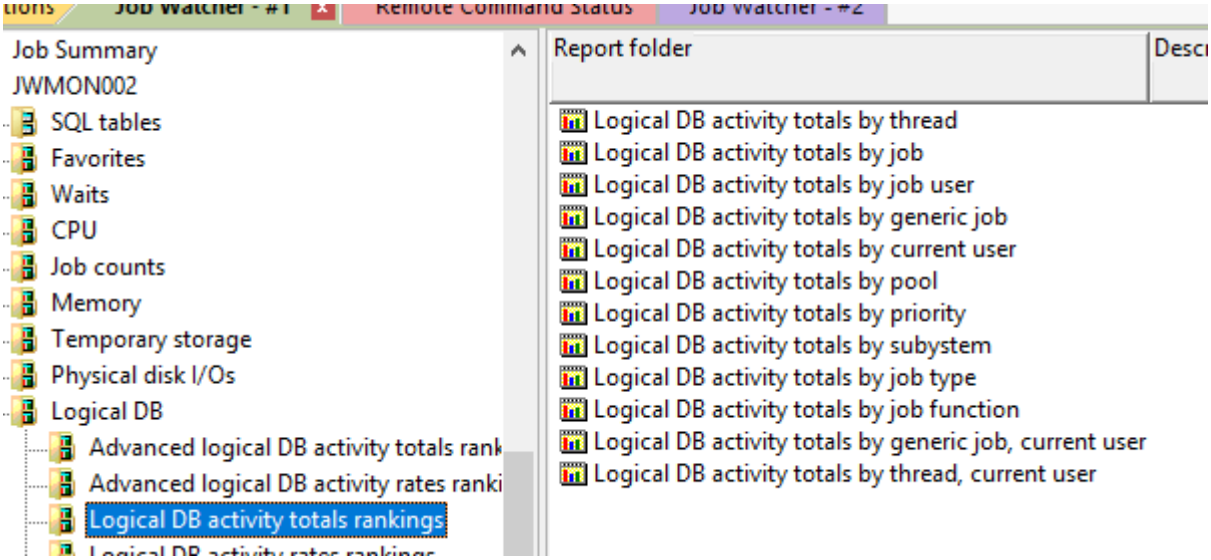


Advanced logical DB activity rates by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more information and examples.

8.20.9 Logical DB activity totals rankings

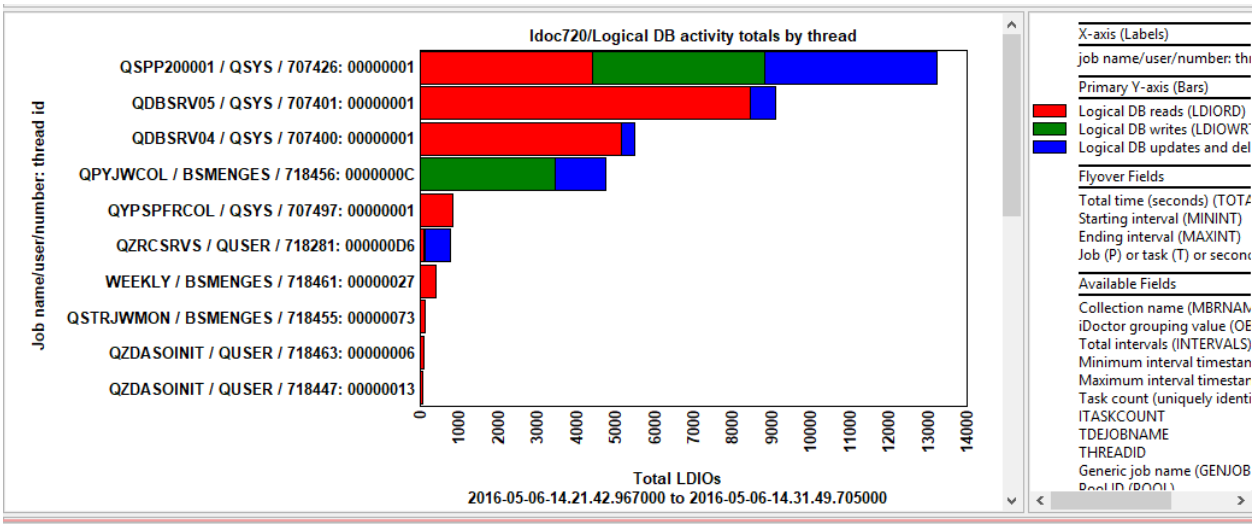
This folder contains a set of job ranking graphs based on logical DB totals for all jobs in the collection.



Logical DB -> Logical DB activity totals rankings

8.20.9.1 Logical DB activity totals by thread

This graph displays several types of logical DB activity totals ranked by thread or system task.

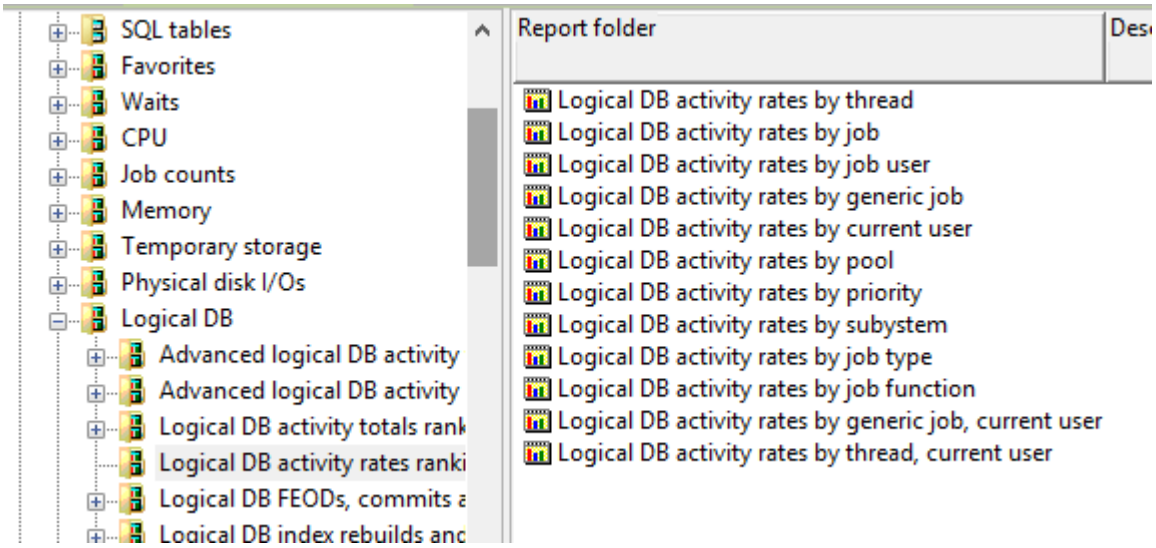


Logical DB activity totals by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more information and examples.

8.20.10 Logical DB activity rates rankings

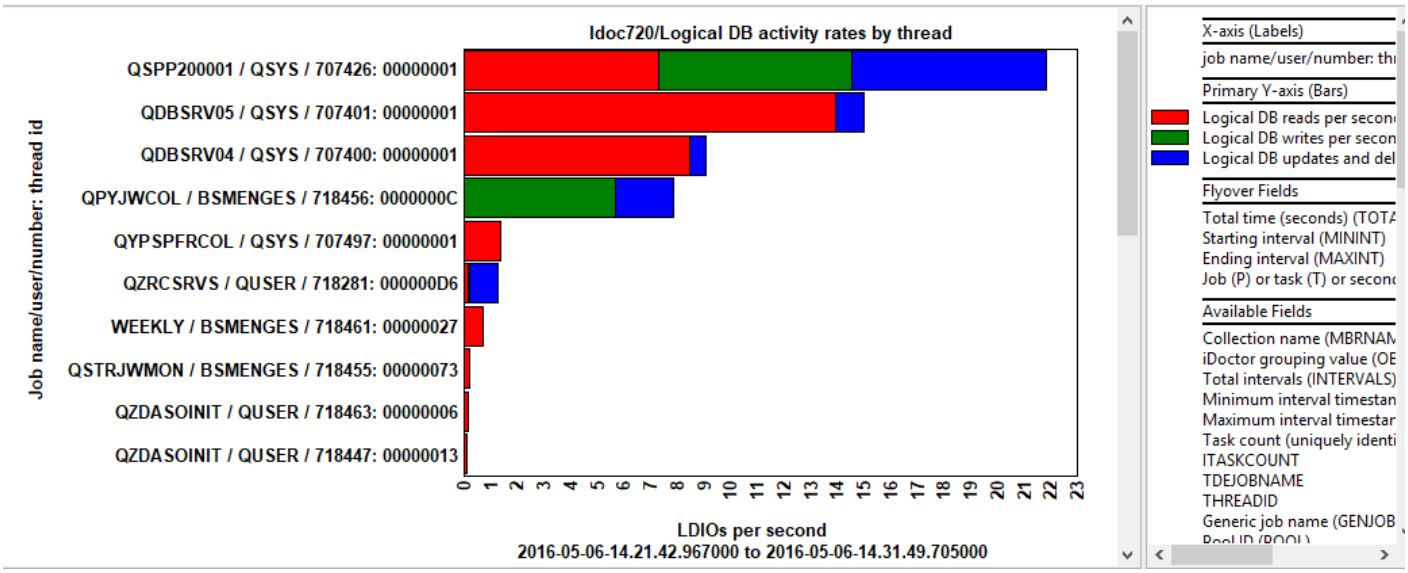
This folder contains a set of job ranking graphs based on logical DB rates for all jobs in the collection.



Logical DB -> Logical DB activity rates rankings

8.20.10.1 Logical DB activity rates by thread

This graph displays several types of logical DB activity rates per second ranked by thread or system task.

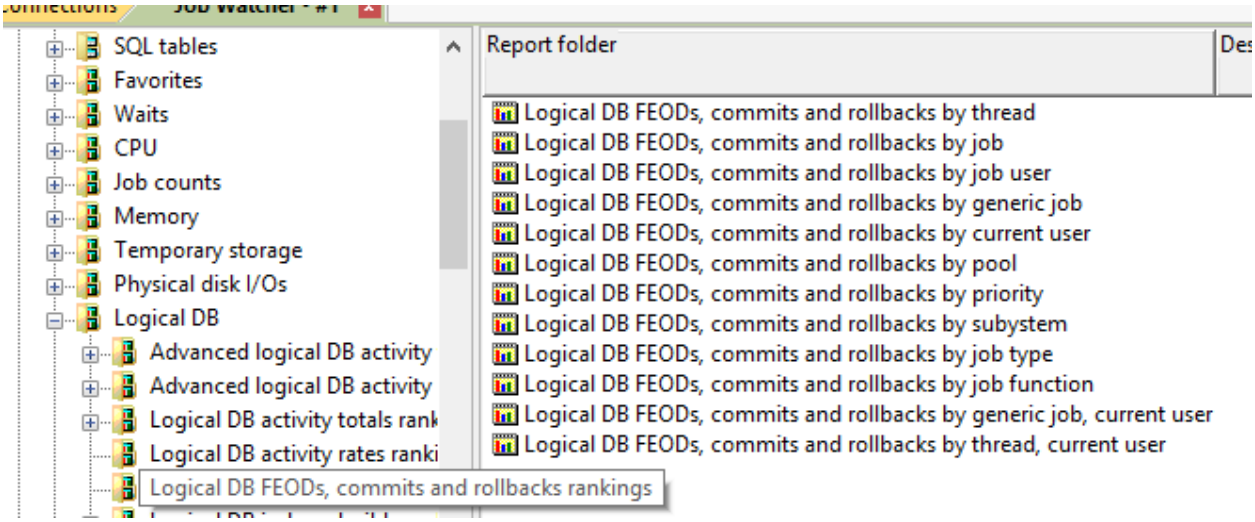


Logical DB activity rates by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more information and examples.

8.20.11 Logical DB FEODS, commits and rollbacks rankings

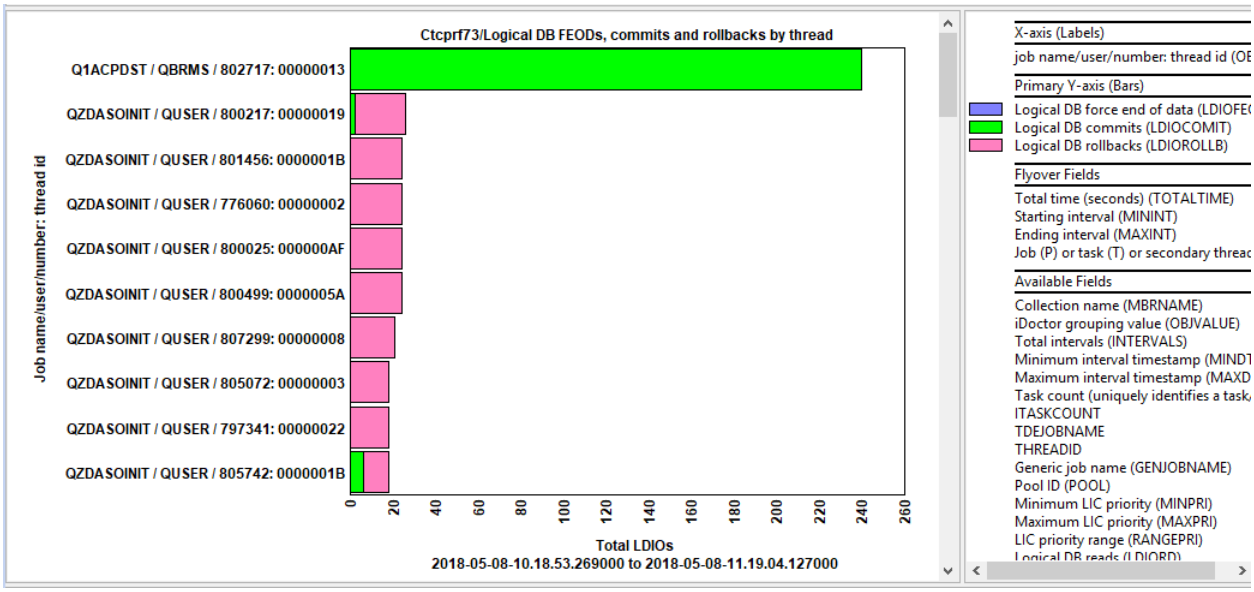
This folder contains a set of job ranking graphs based on logical DB force end of data, commits and rollbacks totals over time.



Logical DB -> Logical DB FEODs, commit and rollbacks rankings

8.20.11.1 Logical DB FEODS, commit and rollbacks by thread

This graph displays ranks logical DB force end of data, commits and rollbacks totals by thread or system task.

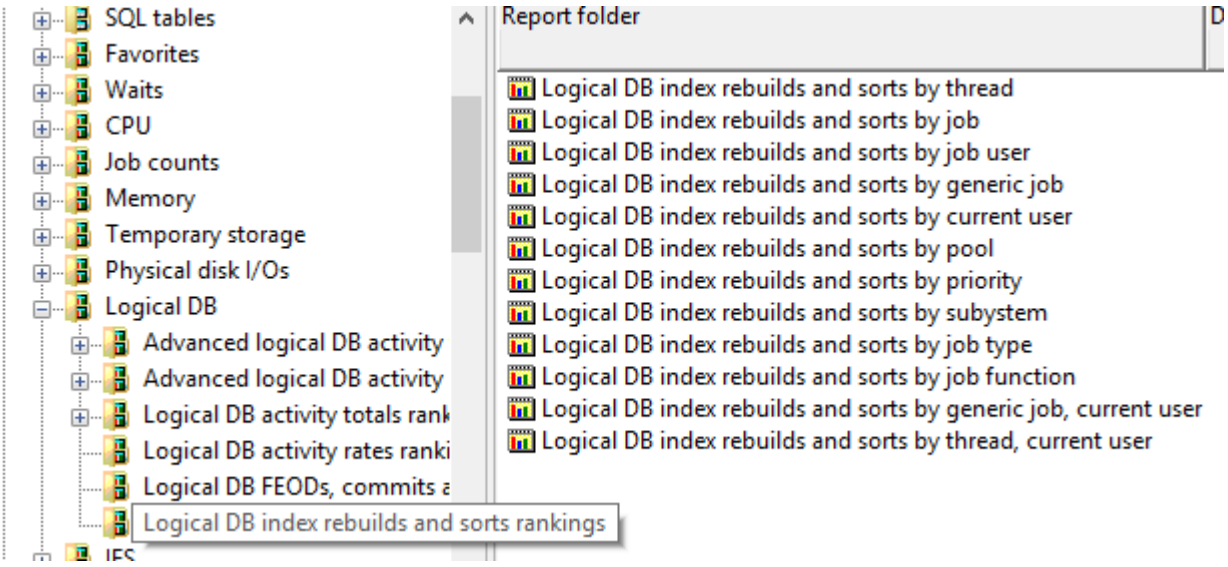


Logical DB FEODs, commit and rollbacks by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more information and examples.

8.20.12 Logical DB index rebuilds and sorts rankings

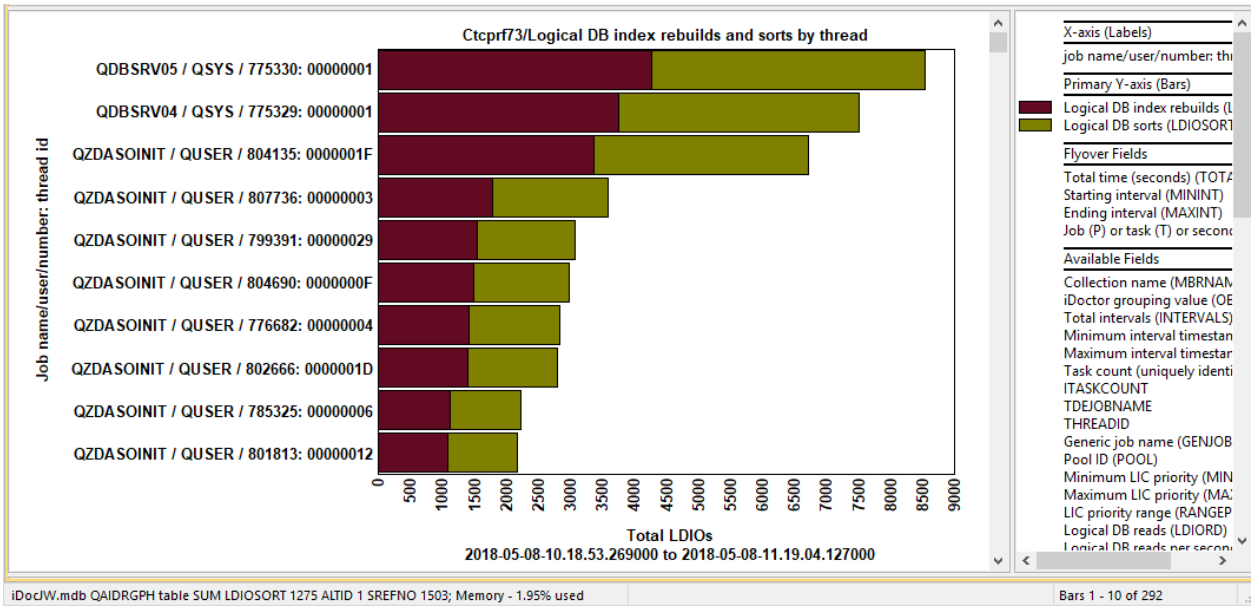
This folder contains a set of job ranking graphs based on logical DB index rebuilds and sorts for all jobs in the collection.



Logical DB -> Logical DB index rebuilds and sorts rankings

8.20.12.1 Logical DB index rebuilds and sorts rates by thread

This graph displays several types of logical DB index rebuilds and sorts ranked by thread or system task.

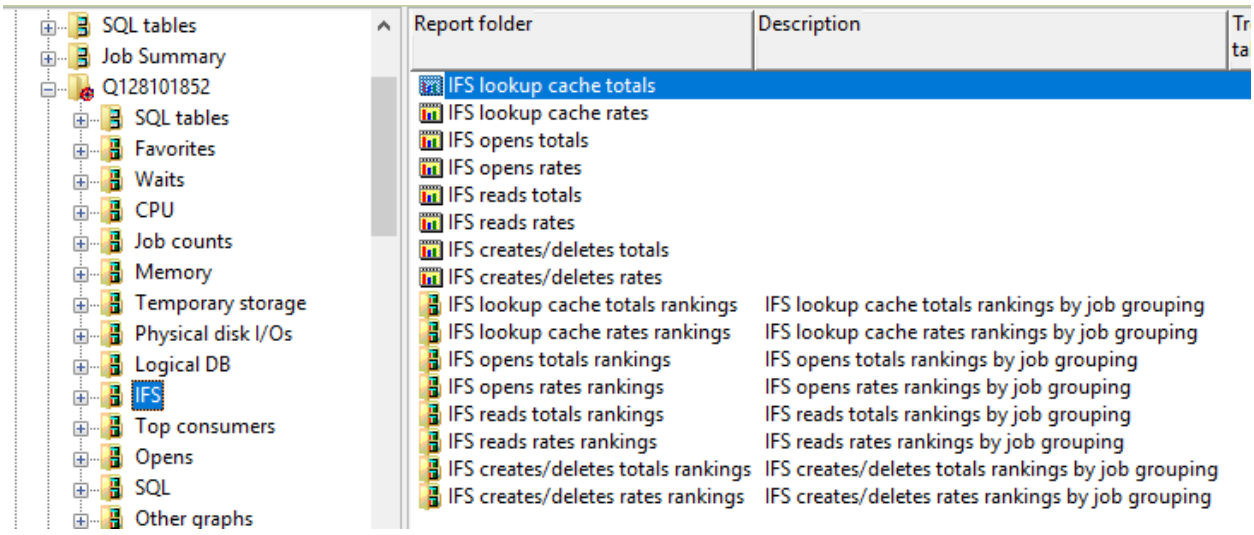


Logical DB index rebuilds and sorts by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more information and examples.

8.21 IFS

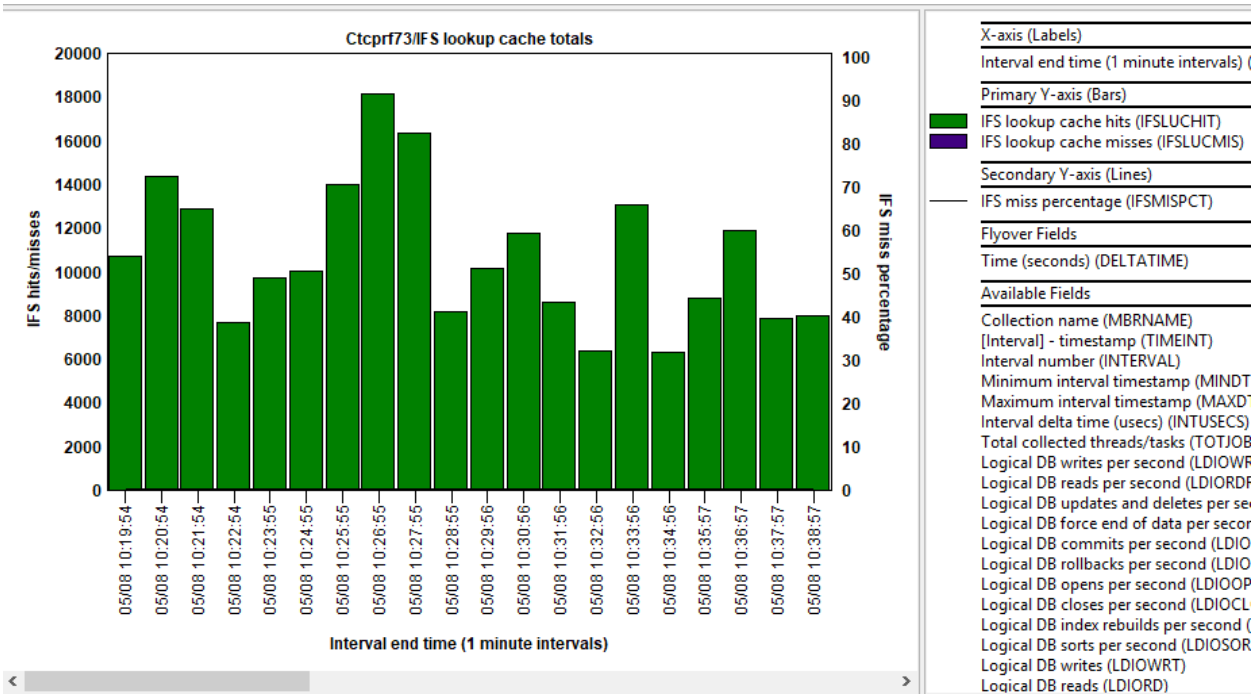
These graphs show IFS statistics for all jobs as either rates or totals over time. These statistics include IFS lookup cache hits/misses, opens, reads (symbolic link reads and directory reads), and creates/deletes.



IFS Folder

8.21.1 IFS lookup cache totals

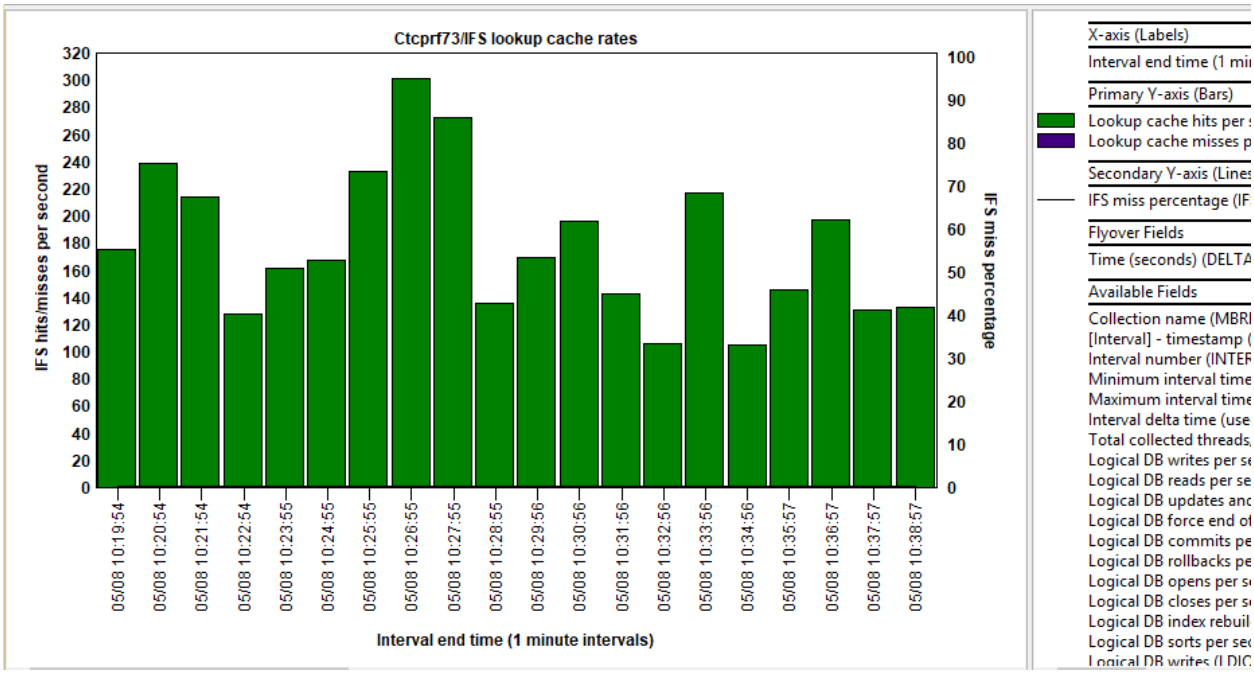
This graph shows the IFS lookup cache hits and misses as well as the percentage missed on the Y2-axis.



IFS lookup cache totals

8.21.2 IFS lookup cache rates

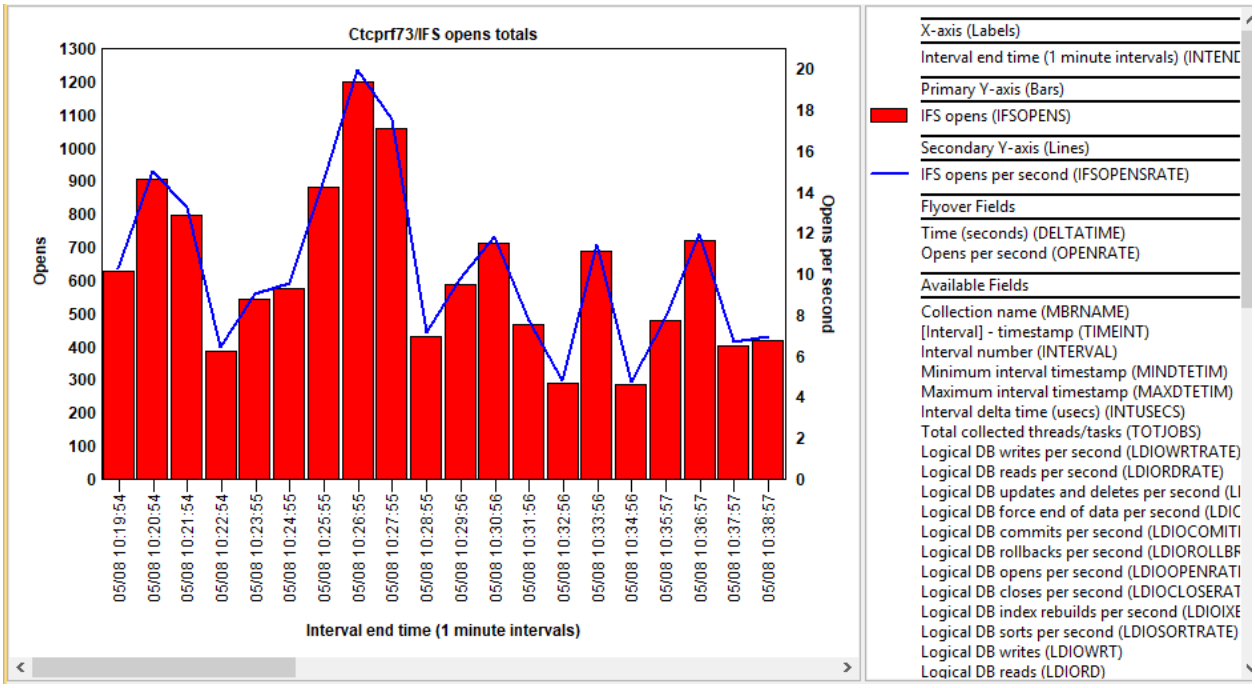
This graph shows the IFS lookup cache hits and misses rates per second on the primary Y-axis as well as the percentage missed on the Y2-axis.



IFS lookup cache rates

8.21.3 IFS opens totals

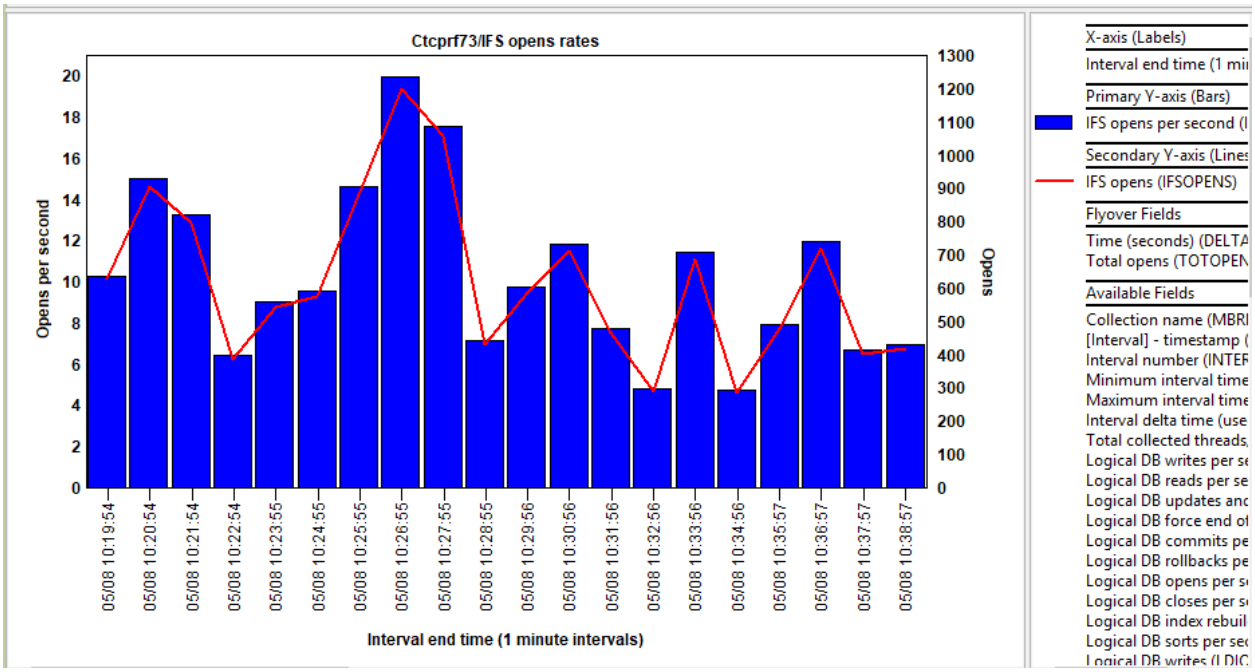
This graph shows the IFS opens totals over time with the IFS opens rate on the second Y-axis.



IFS opens totals

8.21.4 IFS opens rates

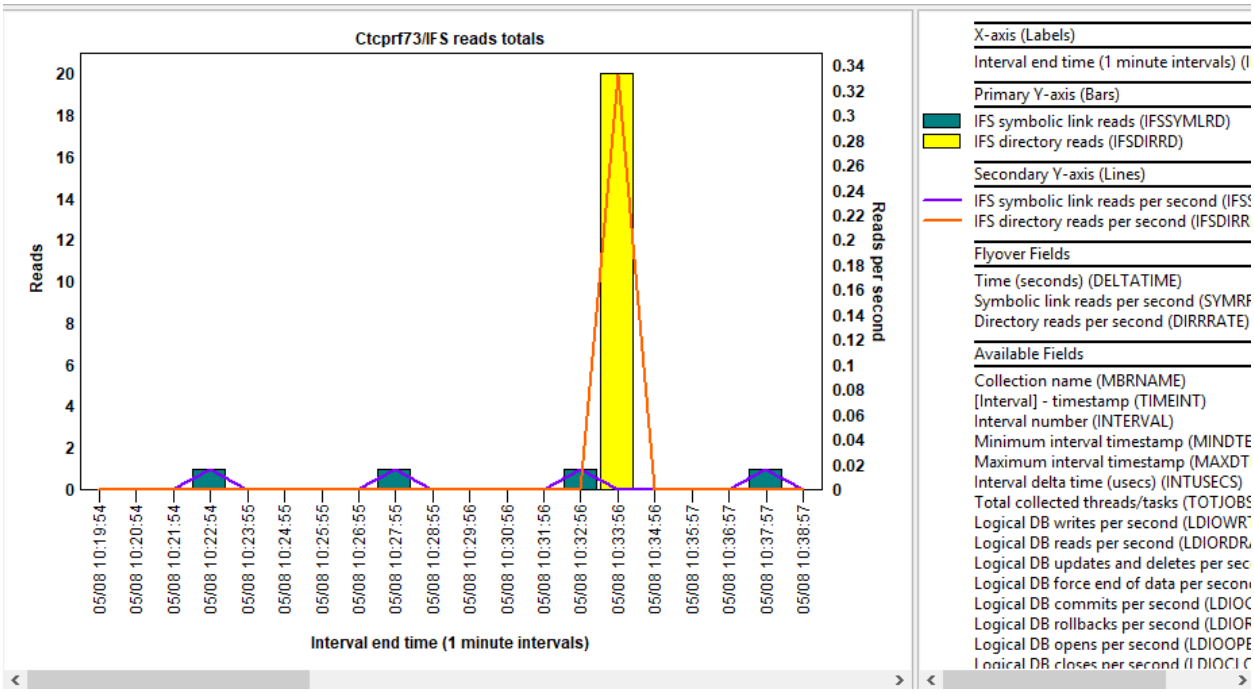
This graph shows the IFS opens rate per second over time with the IFS opens on the second Y-axis.



IFS opens rates

8.21.5 IFS reads totals

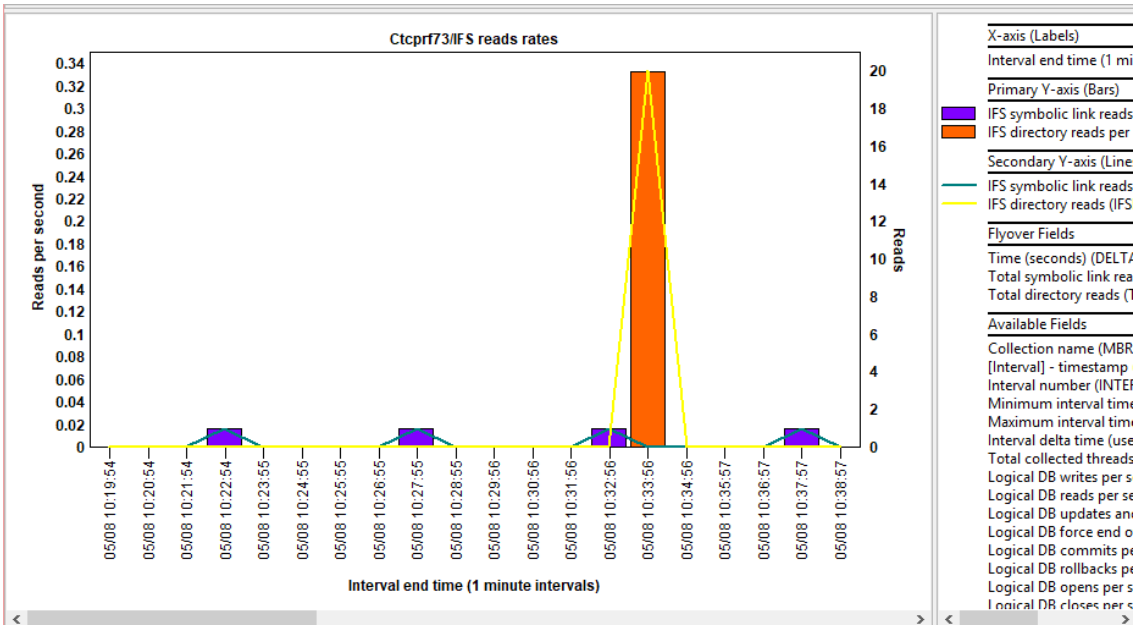
This graph shows the IFS symbolic link reads and directory reads over time. The totals are found on the secondary Y-axis.



IFS reads totals

8.21.6 IFS reads rates

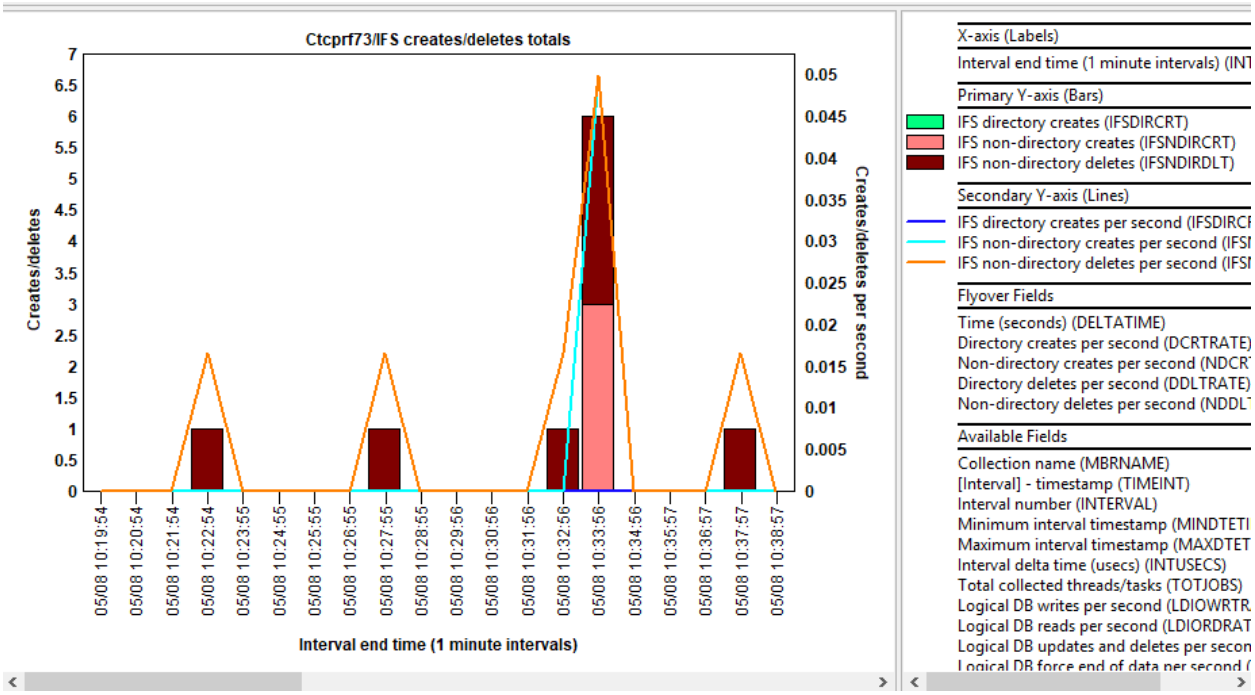
This graph shows the IFS symbolic link reads and directory reads over time. The rates are found on the secondary Y-axis.



IFS read rates

8.21.7 IFS creates/deletes totals

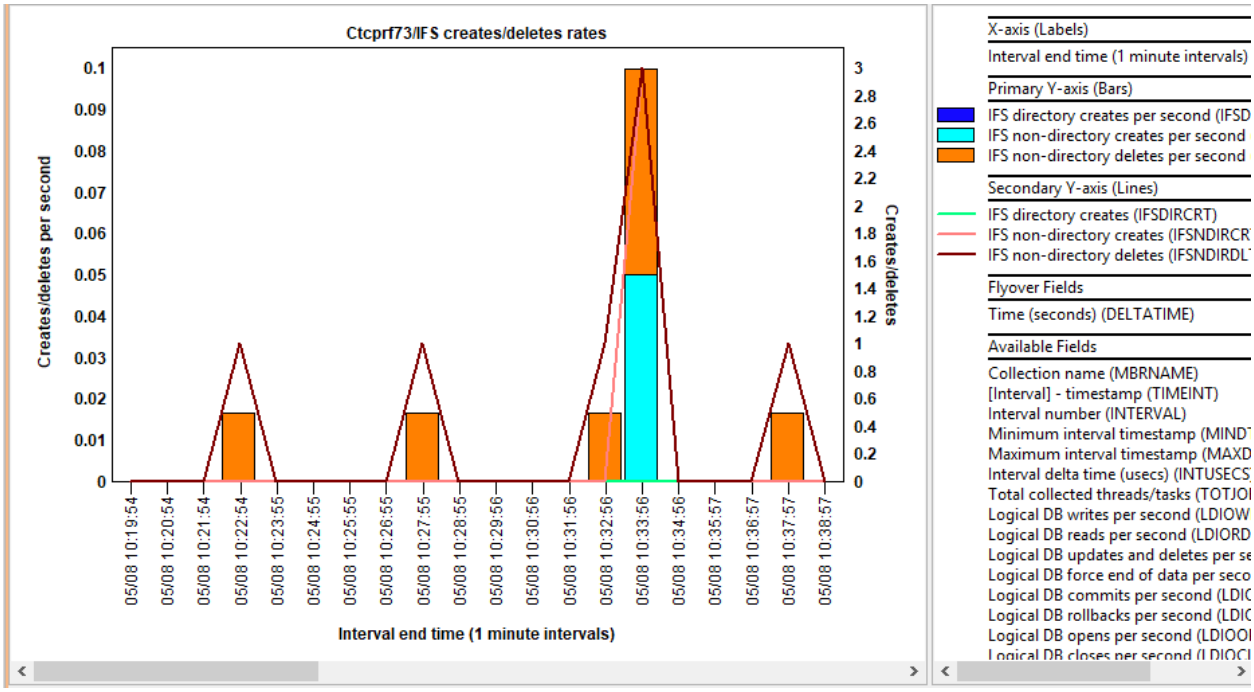
This graph shows the IFS creates/deletes totals as well as the rates for these on the Y2-axis.



IFS creates/delete totals

8.21.8 IFS create/deletes rates

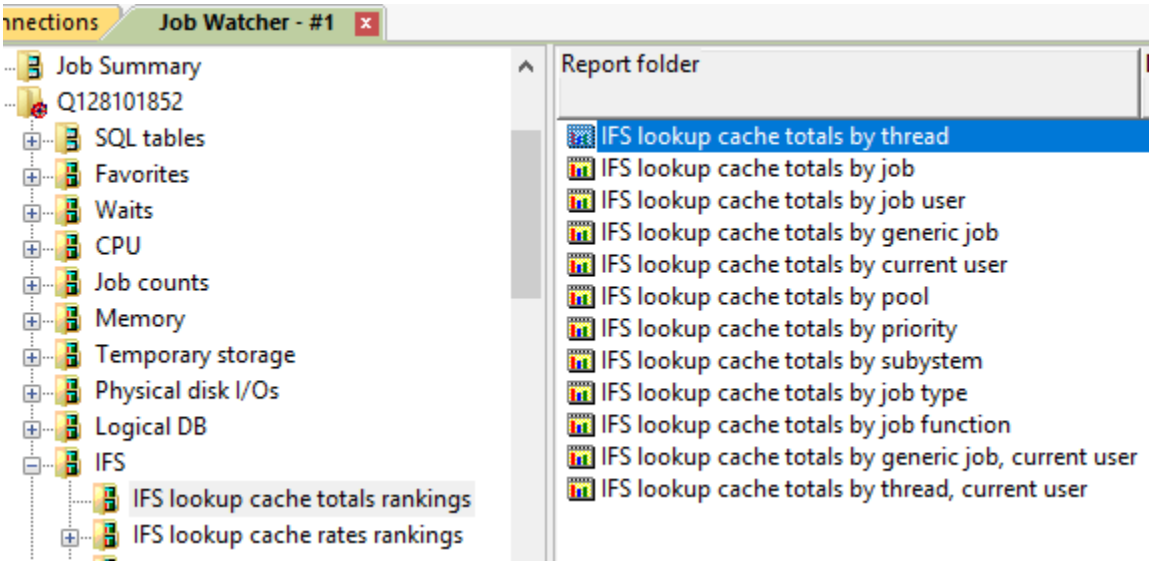
This graph shows the IFS creates/deletes rates per second as well as the totals for these on the Y2-axis.



IFS creates/deletes rates

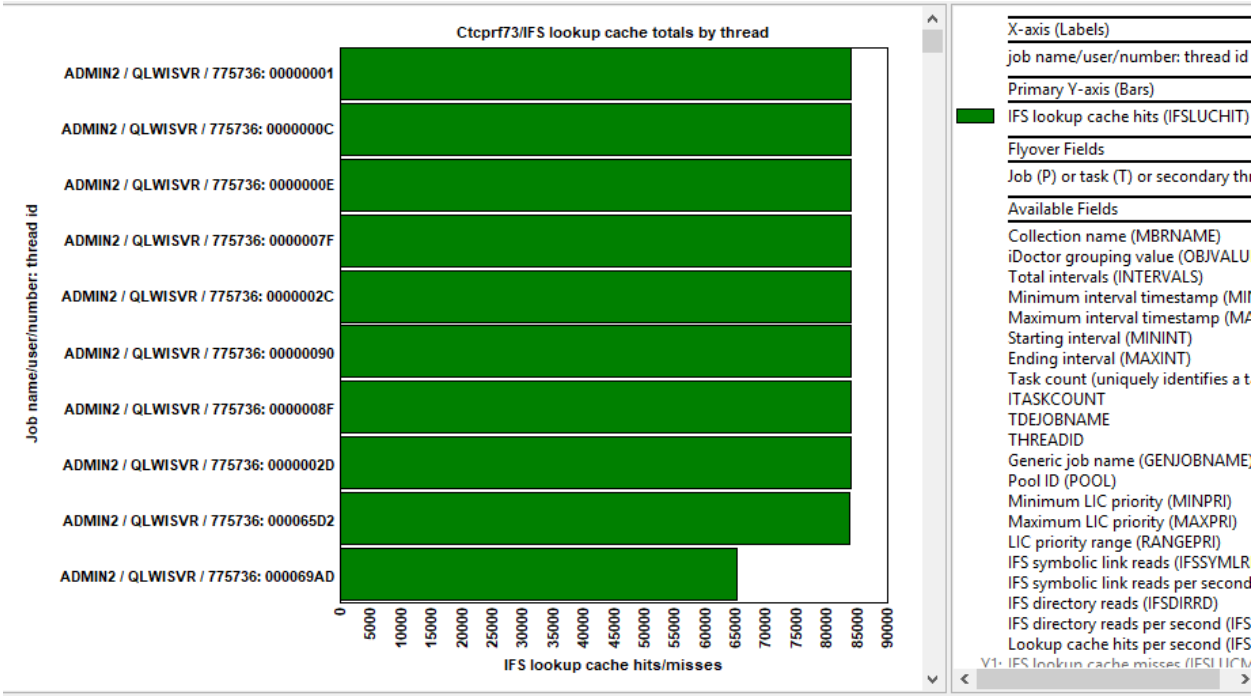
8.21.9 IFS lookup cache rankings

These graphs display the IFS lookup cache hits and misses (either showing total counts or rates) ranked in various ways.



IFS -> IFS lookup cache totals rankings

An example is shown below:

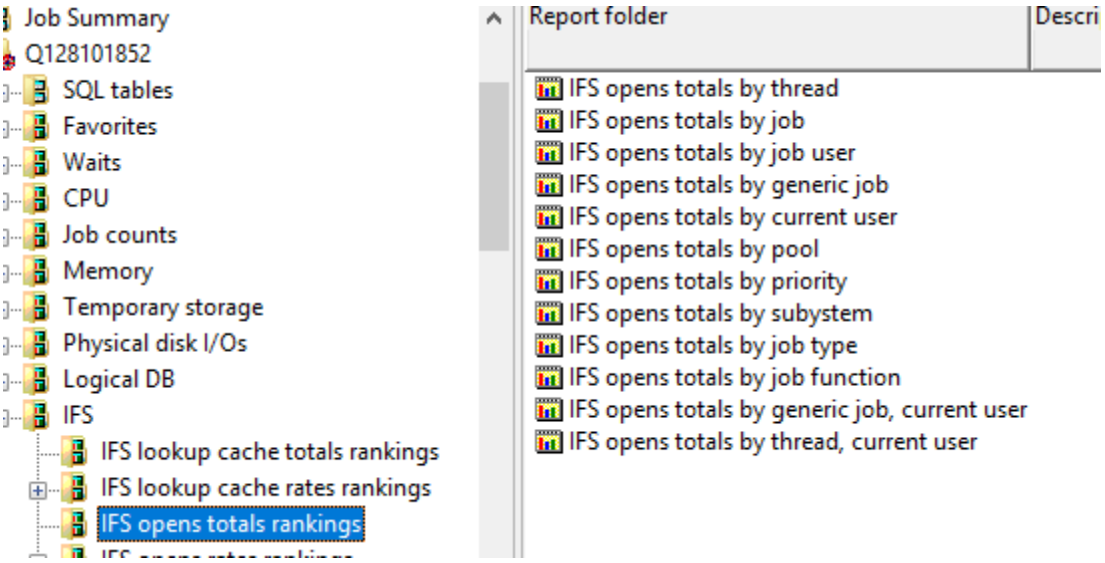


IFS lookup cache totals by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more examples.

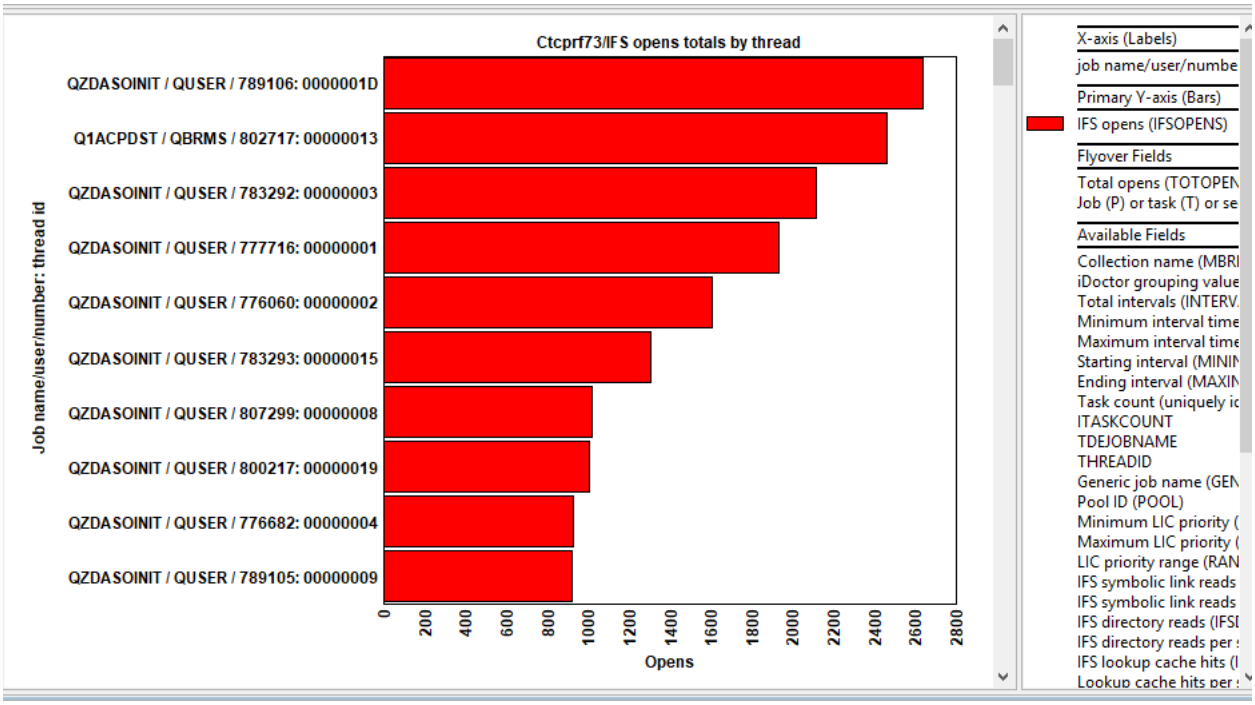
8.21.10 IFS opens rankings

These graphs display the IFS opens (either showing total counts or rates) ranked in various ways.



IFS -> IFS opens totals rankings

An example is shown below:

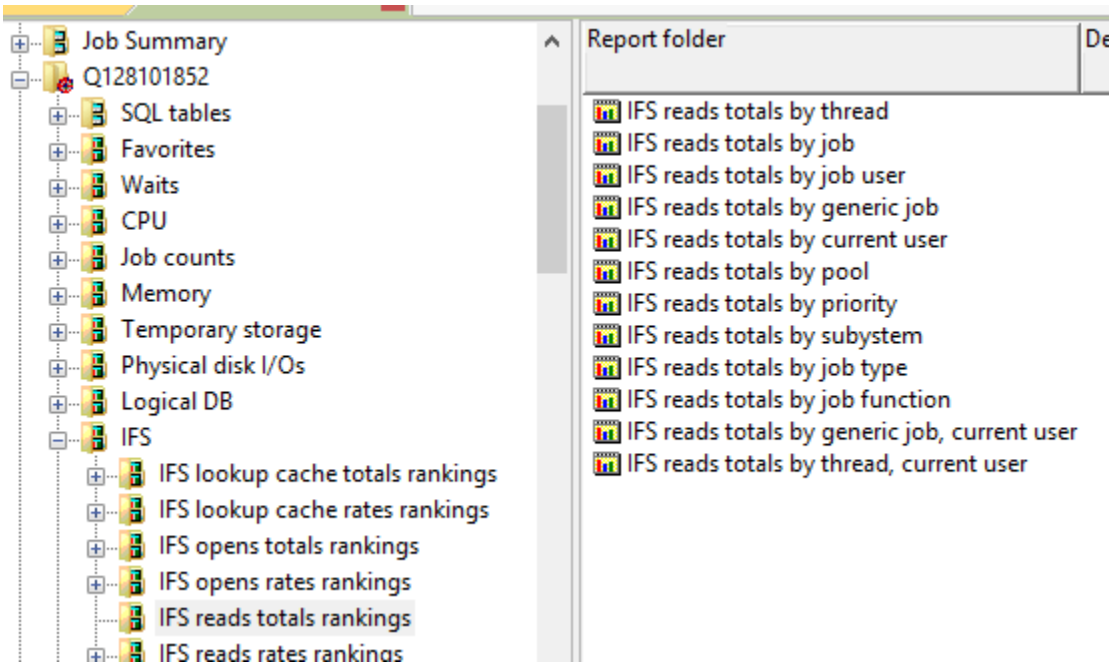


IFS opens totals by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more examples.

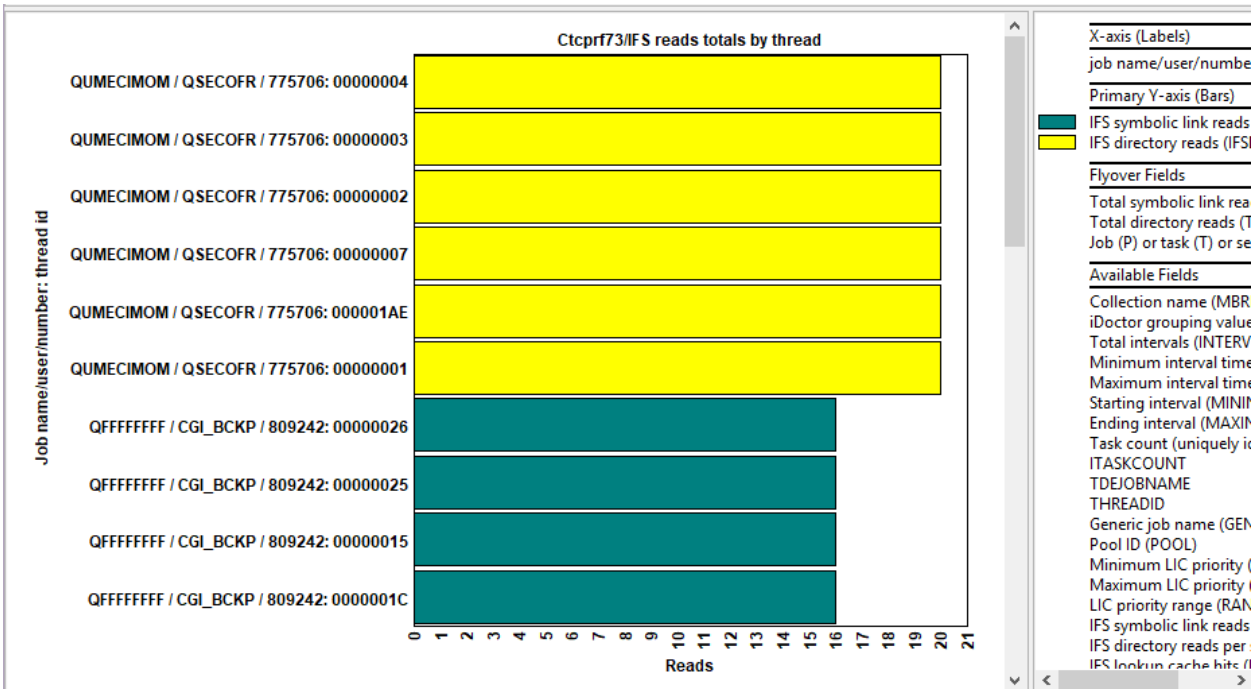
8.21.11 IFS reads rankings

These graphs display the IFS reads (either showing total counts or rates) ranked in various ways.



IFS -> IFS reads totals rankings

An example is shown below:

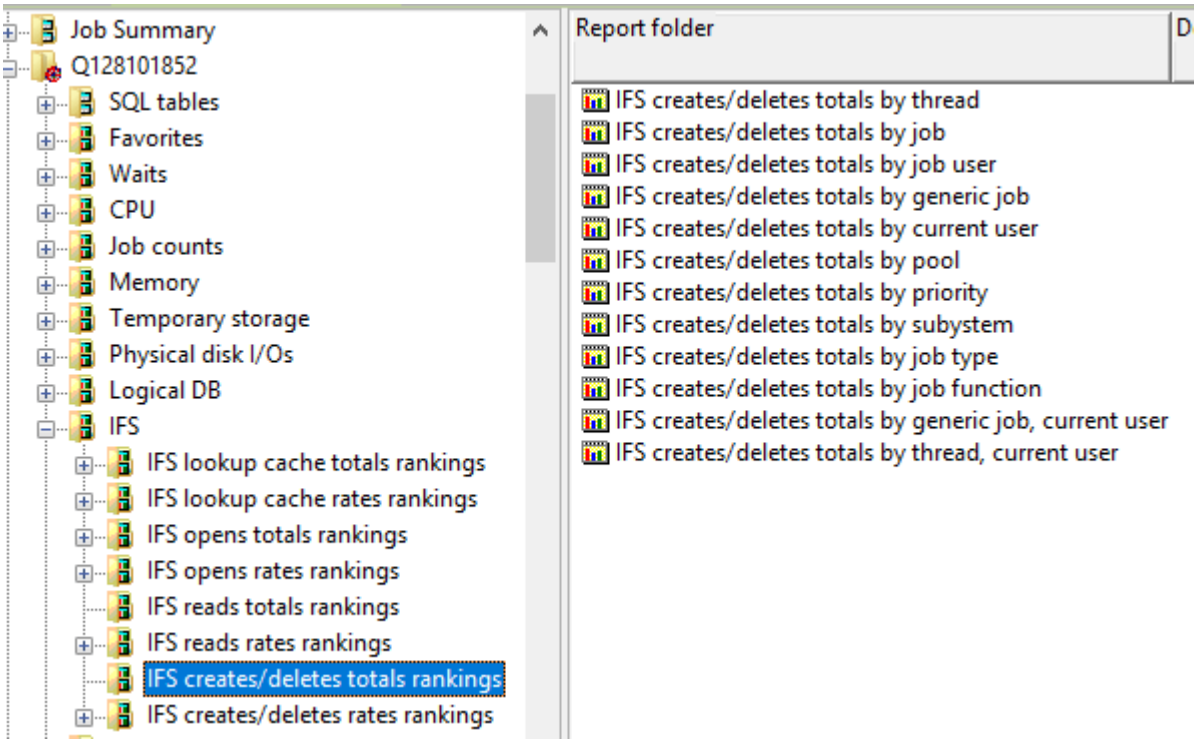


IFS reads totals by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more examples

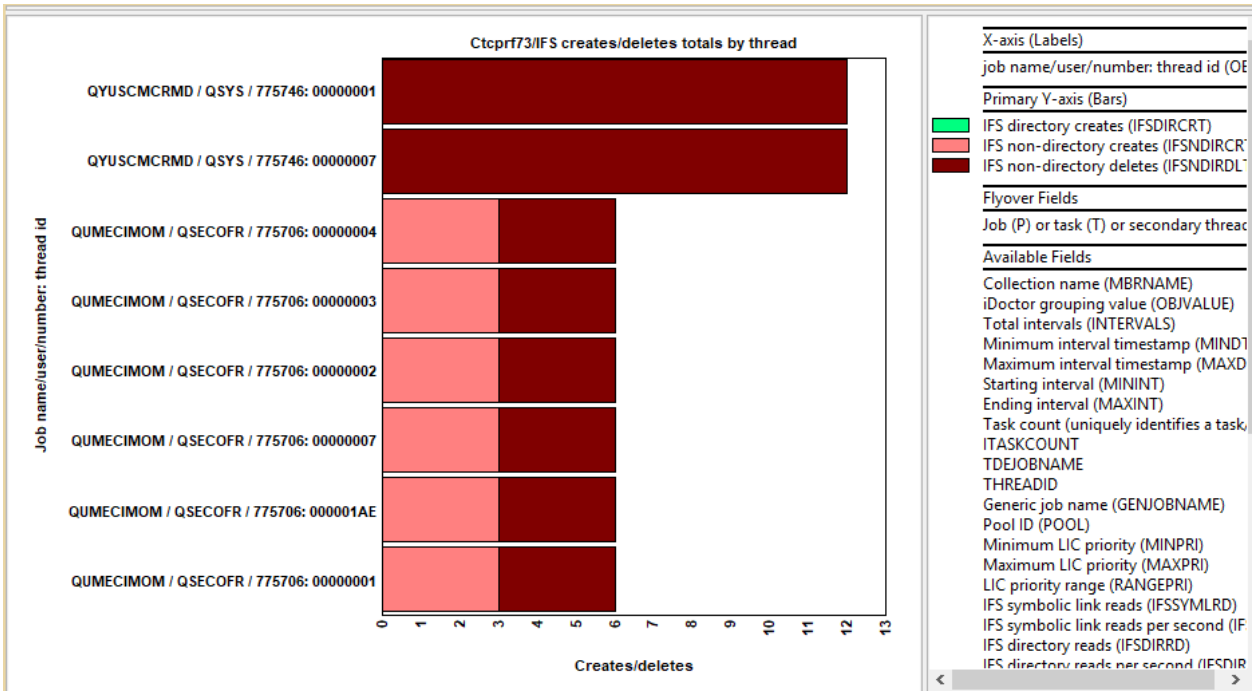
8.21.12 IFS creates/deletes rankings

These graphs display the IFS creates/deletes (either showing total counts or rates) ranked in various ways.



IFS -> IFS creates/deletes totals rankings

An example is shown below:

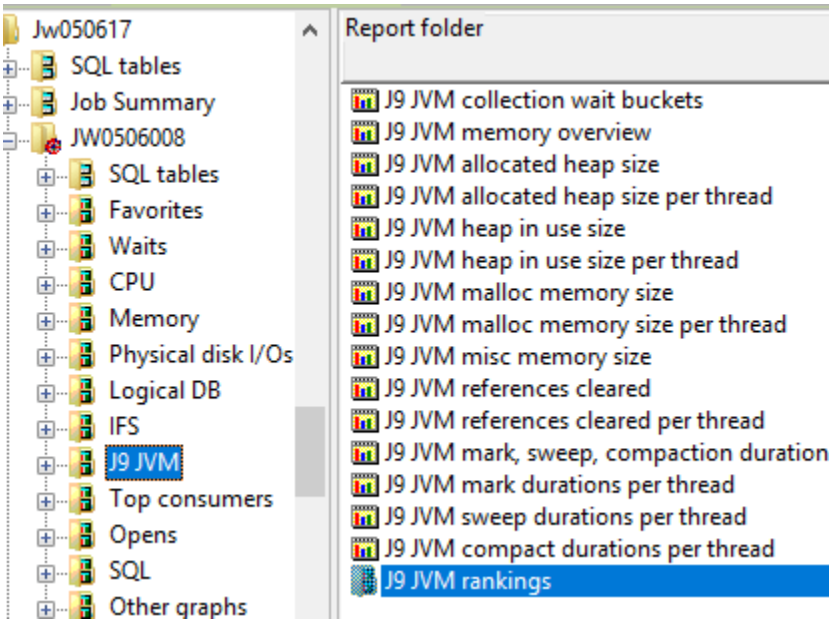


IFS creates/deletes totals by thread

These graphs are very similar to the Advanced logical DB activity totals rankings. See that section for more examples.

8.22 J9 JVM

The J9 JVM folder shows statistics related to the J9 JVMs (IBM Technology for Java) job found in the collection. The **J9 JVM collection wait buckets** (and the J9 JVM rankings folder), shows the wait buckets for all JVMs added up together and the rest of the graphs pertain to garbage collection statistics.

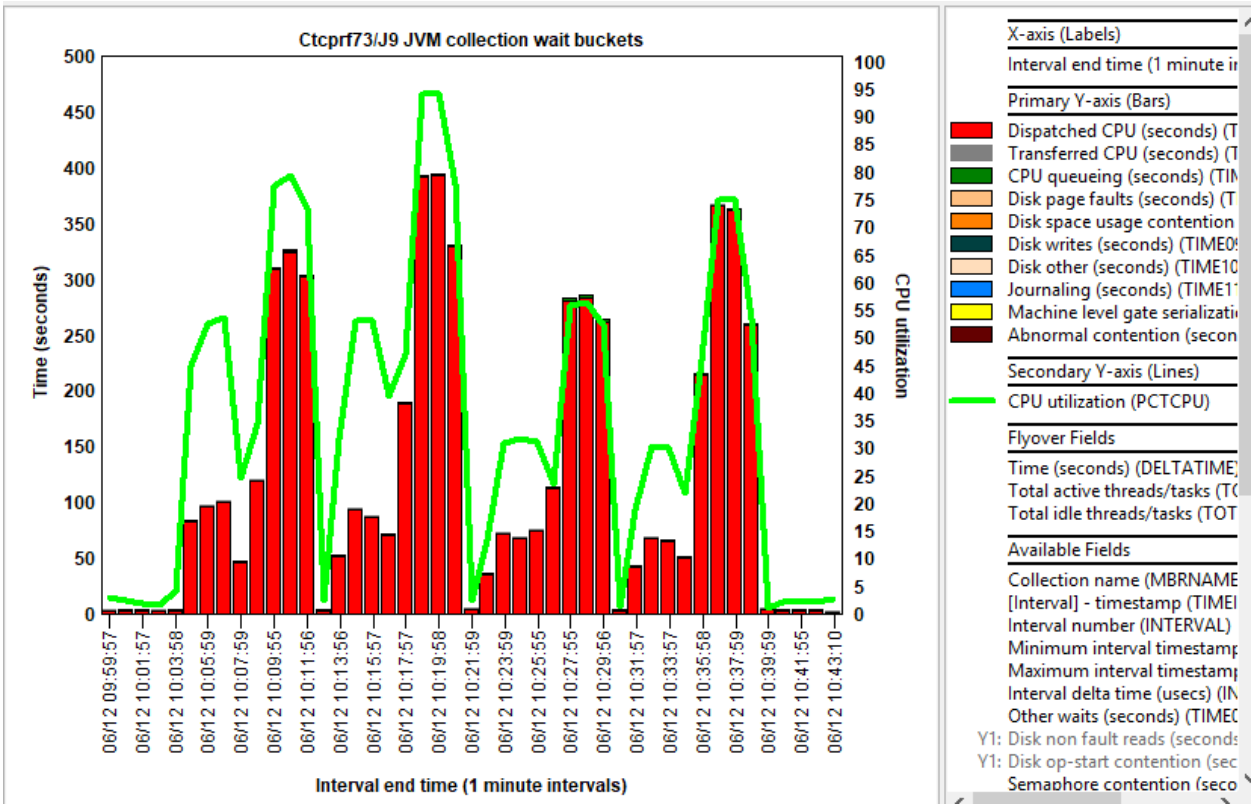


J9 JVM Folder

Note: The data to build these graphs is NOT collected by default. You must define a definition that includes the IBM Technology for Java data for the required files to get created which will then allow these graphs to appear.

8.22.1 J9 JVM collection wait buckets

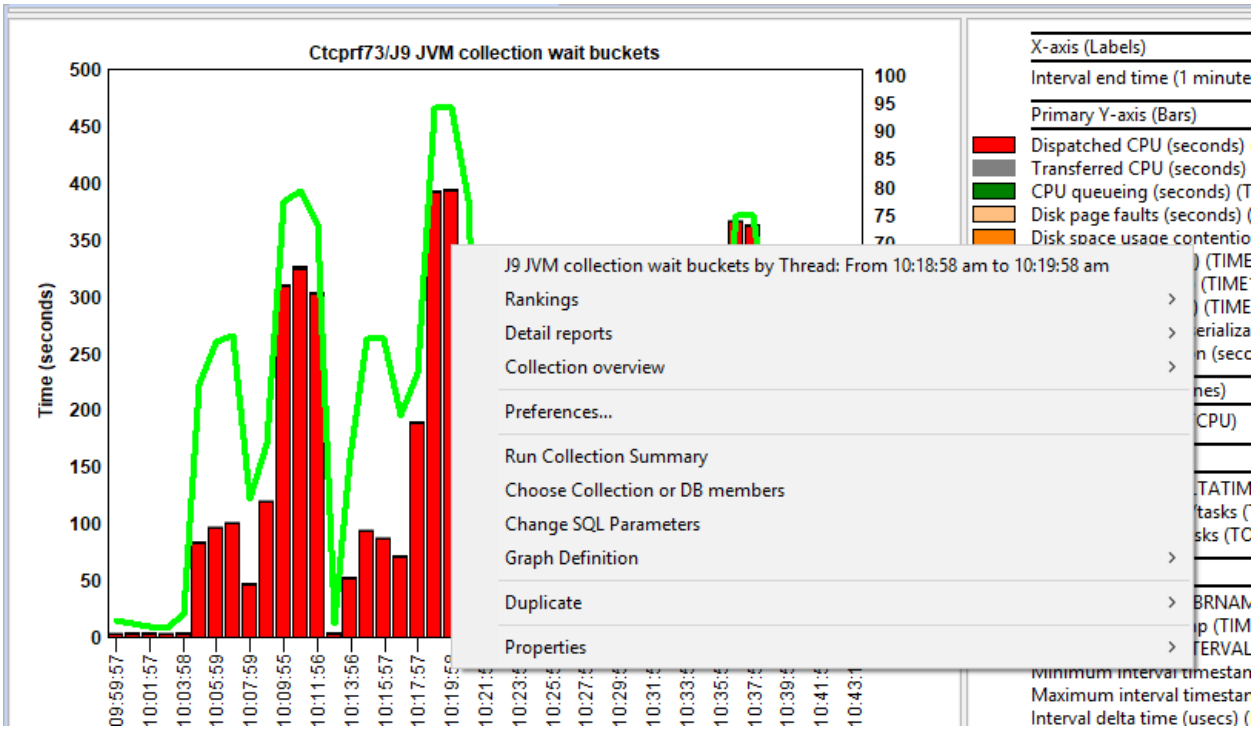
This graph summarizes the wait buckets in a way like the Collection overview time signature but only includes jobs running J9 JVMs. Keep in mind that this graph does not include PASE time which is a common type of wait for J9 JVMs, but it is typically used as an idle wait for jobs and is not an interesting type of wait.



J9 JVM collection wait buckets

You can drill down and see the JVMs behind this graph by doing a right-click on the desired time period and selecting the 1st option.

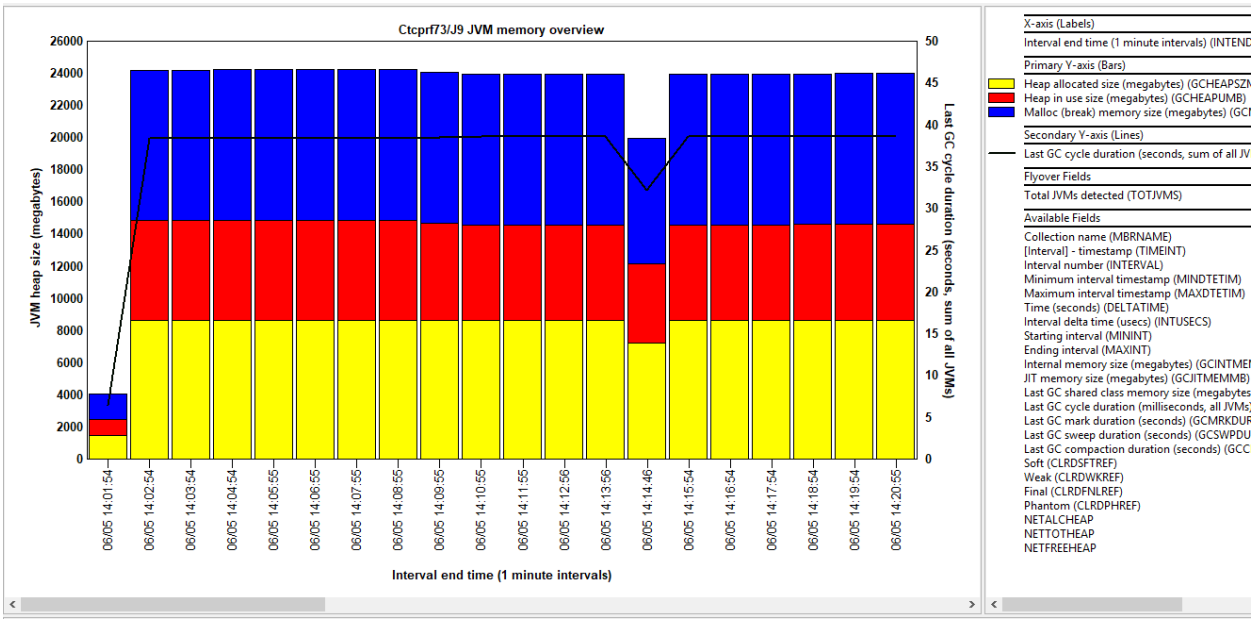
Note: The Collection summary analysis must be ran first in order for this drill-down option to appear.



J9 JVM collection wait buckets drill-down menu

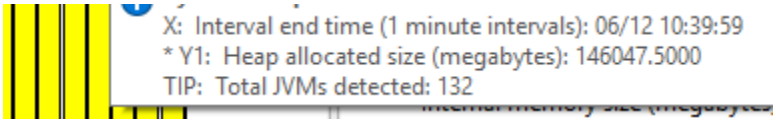
8.22.2 J9 JVM memory overview

This graph adds up and displays heap related metrics for all J9 JVMs. This graph also includes the total last GC cycle duration (in milliseconds) for all JVMs added up on the Y2-axis.



J9 JVM memory overview

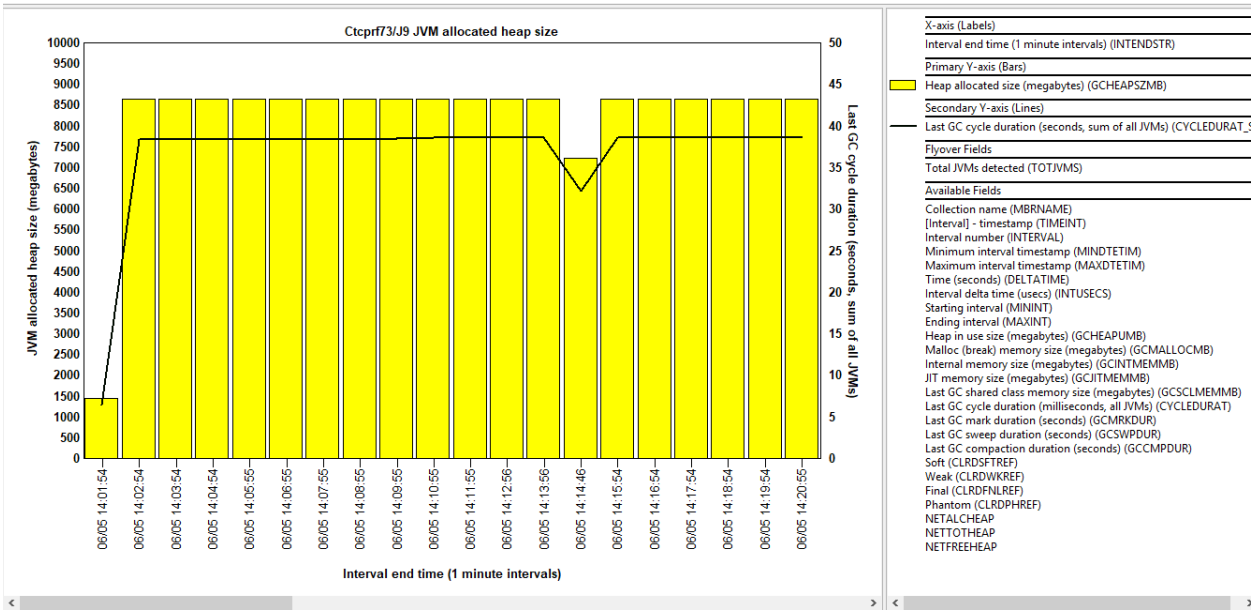
Tip: The **total JVMs detected** can be shown by looking at the graph flyover.



J9 JVM memory overview graph flyover

8.22.3 J9 JVM allocated heap size

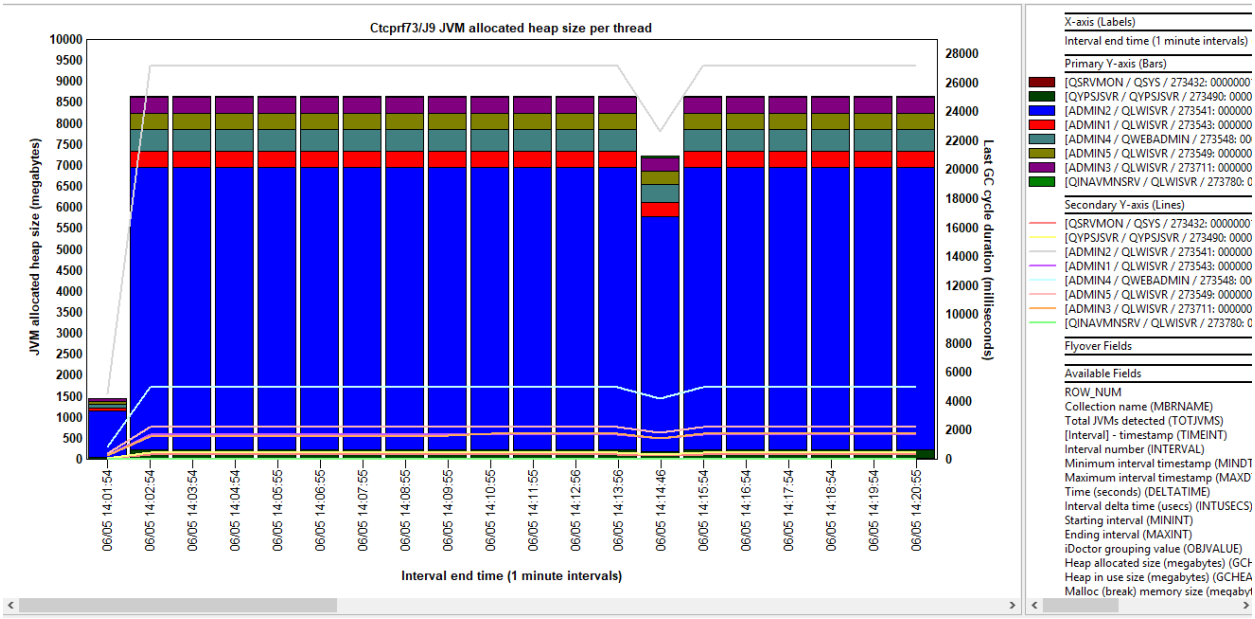
This graph adds up JVM allocated heap size for all J9 JVMs. The graph also includes the total last GC cycle duration (in milliseconds) for all JVMs added up on the Y2-axis.



J9 JVM allocated heap size

8.22.4 J9 JVM allocated heap size per thread

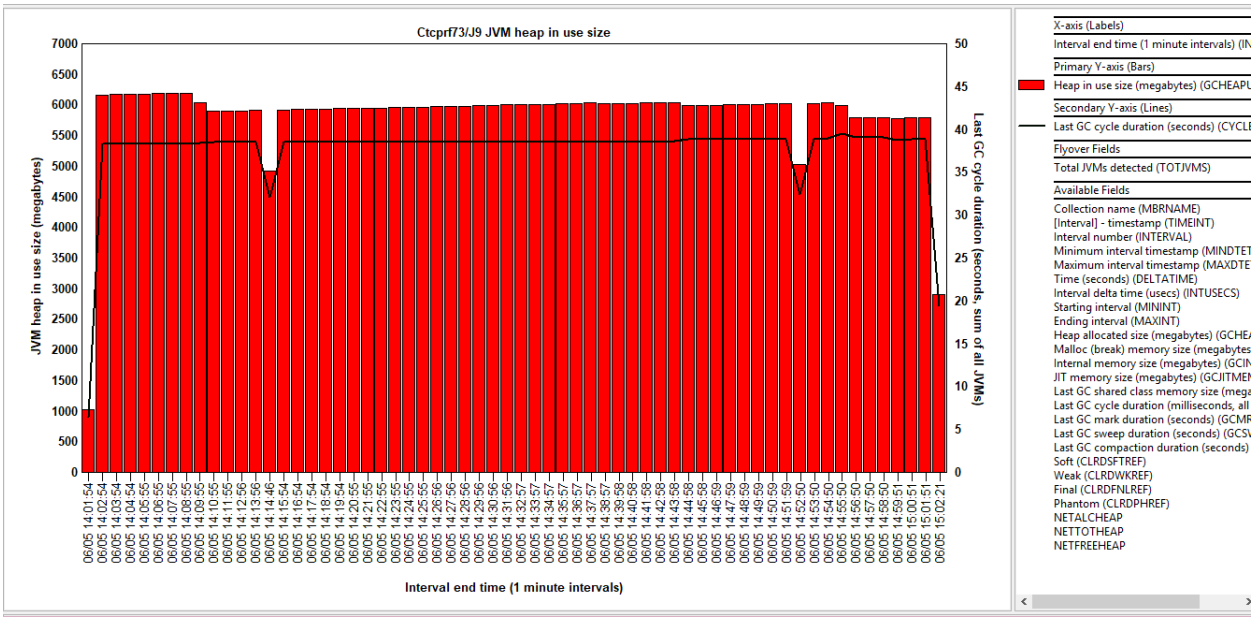
This graph is a “flattened” style graph and shows each J9 JVM’s allocated heap size together over time. The 2nd Y-axis displays each JVM’s cycle duration (in milliseconds.)



J9 JVM allocated heap size per thread

8.22.5 J9 JVM heap in use size

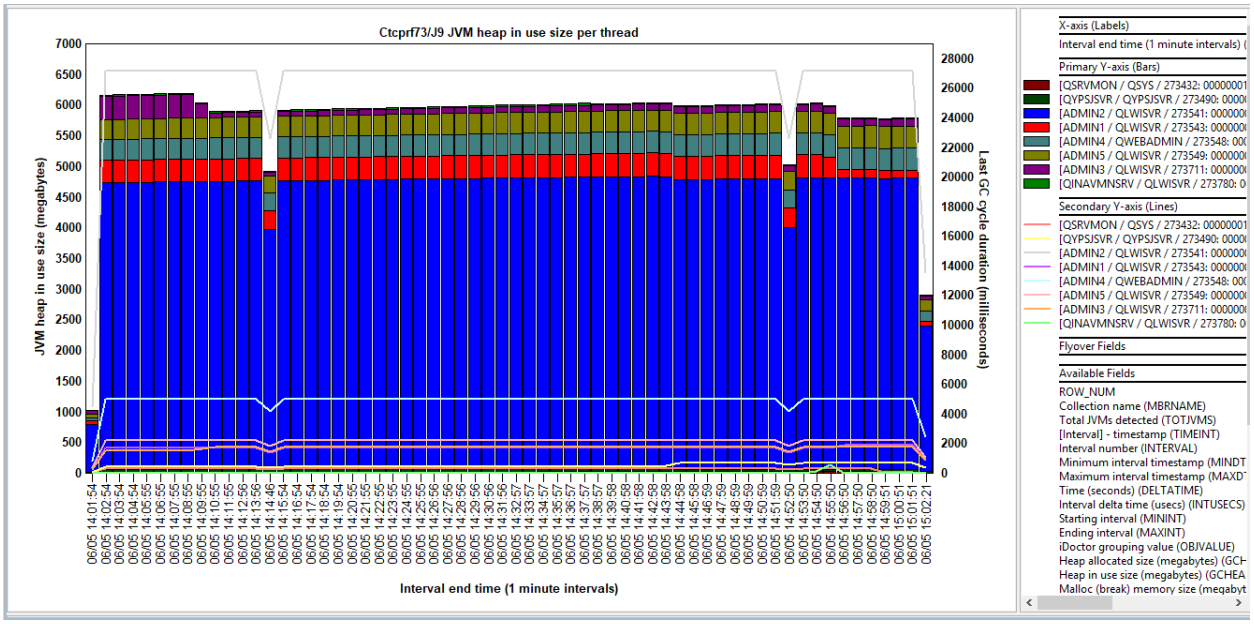
This graph adds up the J9 JVM heap in use total size (in megabytes) for all JVMs. The 2nd Y-axis shows the sum of all J9 JVM cycle durations (in seconds.)



J9 JVM heap in use size

8.22.6 J9 JVM heap in use size per thread

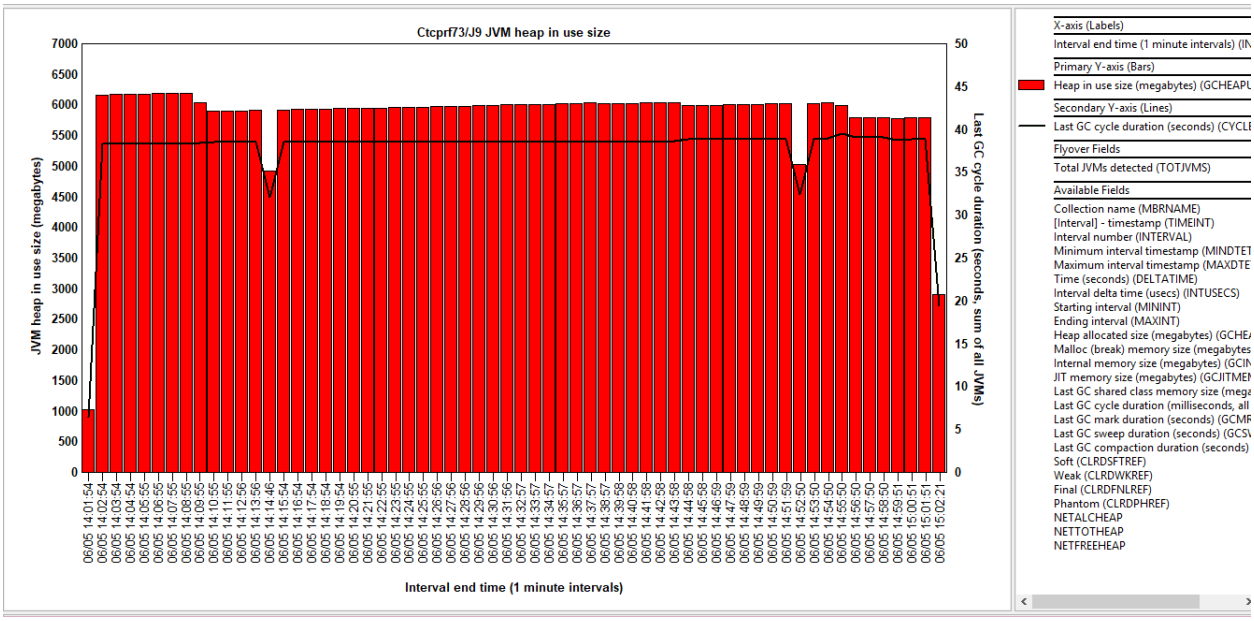
This graph breaks out the heap in use size from the previous graph by thread and displays the results in "flattened" format over time. The 2nd Y-axis shows the individual J9 JVM cycle duration (in milliseconds.)



J9 JVM heap in use size per thread

8.22.7 J9 JVM malloc memory size

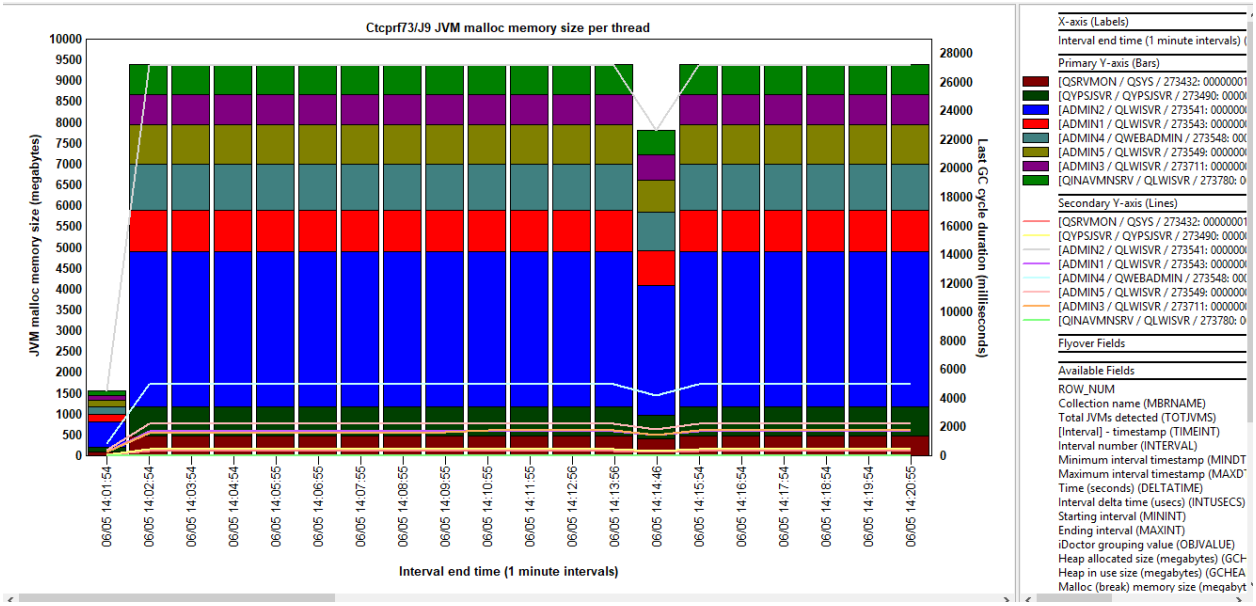
This graph adds up the J9 JVM malloc (memory allocations, in megabytes) for all JVMs. The 2nd Y-axis shows the sum of all J9 JVM cycle durations (in seconds.)



J9 JVM heap in use size

8.22.8 J9 JVM malloc memory size per thread

This graph breaks out the J9 JVM malloc (memory allocations, in megabytes) from the previous graph by thread and displays the results in “flattened” format over time. The 2nd Y-axis shows the individual J9 JVM cycle duration (in milliseconds.)



J9 JVM heap in use size per thread

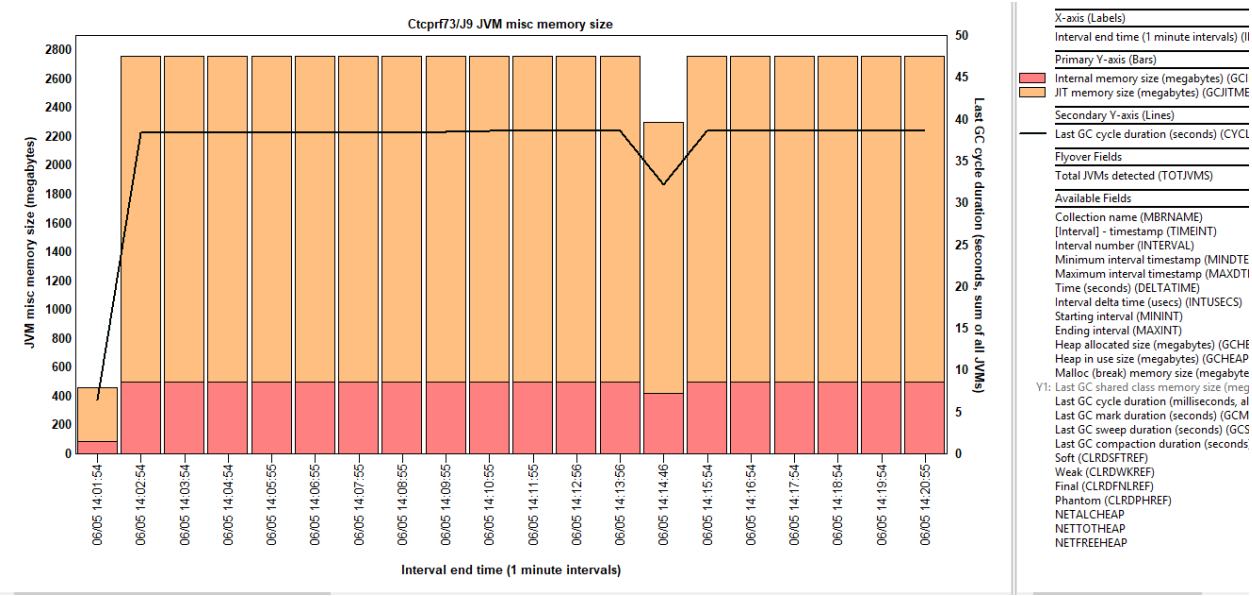
8.22.9 J9 JVM misc memory size

This graph adds up and displays additional memory sizes related to J9 JVMs.

The metrics included (in megabytes) are:

1) Internal memory size
2) JIT memory size
3) Last GC shared class memory size

The 2nd Y-axis shows the sum of all J9 JVM cycle durations (in seconds.)



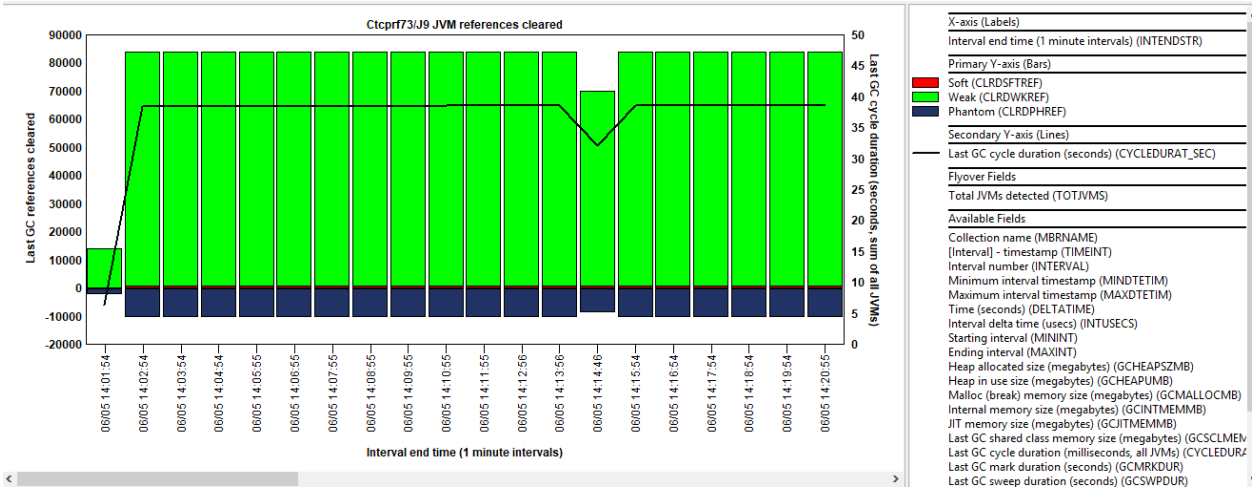
J9 JVM misc memory size

8.22.10 J9 JVM references cleared

This graph adds up and displays the 4 types of references cleared counts for all J9 JVMs.

The values are taken from file QAPYJWIJVM and include the following:

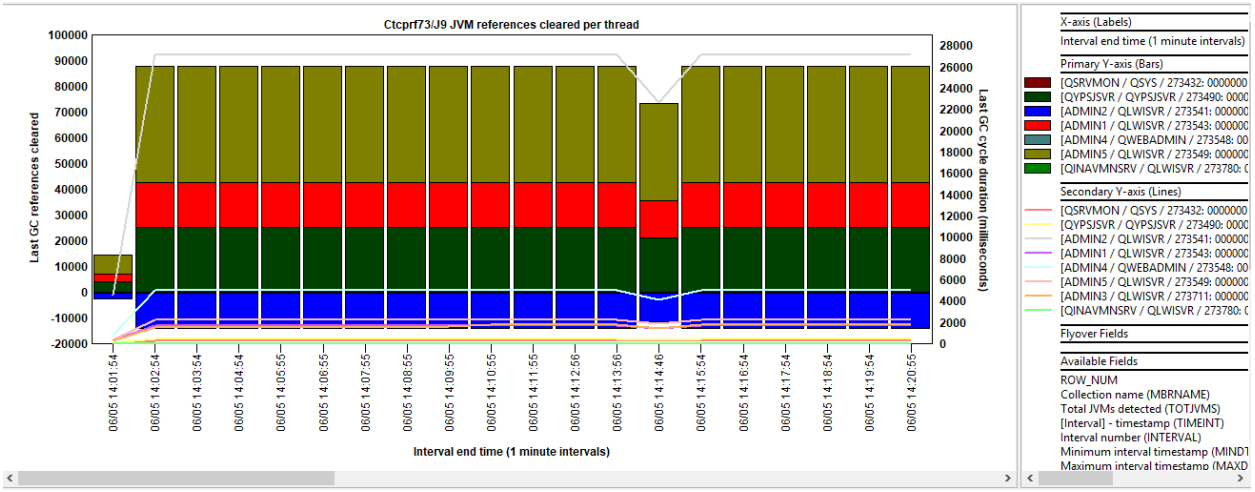
1) JMGCSFRCLR – GC soft references cleared
2) JMGCWKRCLR – GC weak references cleared
3) JMGCFNRCLR – GC finalizer references cleared
4) JMGCPHRCLR – GC phantom references cleared (Note: During testing this contained some negative values which appears to be a bug in the IBM i.)



J9 JVM references cleared

8.22.11 J9 JVM references cleared per thread

This graph adds up all 4 types of references cleared in the previous chart and shows the total but per thread over time.



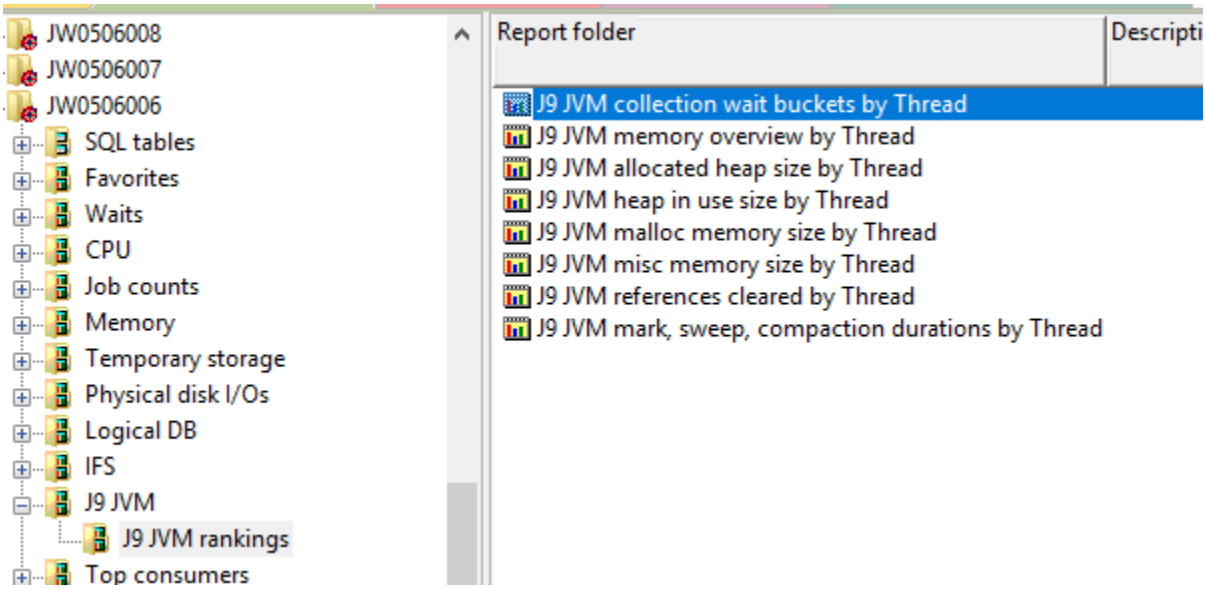
J9 JVM references cleared per thread

8.22.12 J9 JVM mark, sweep, compaction durations

Note: The values in the raw data in QAPYJWIJVM for this graph do not seem correct currently.

8.22.13 J9 JVM rankings

This folder contains ranking graphs by thread/JVM for several of the overview graphs in the parent folder. From these graphs a user-can right-click a single JVM and graph it over time.



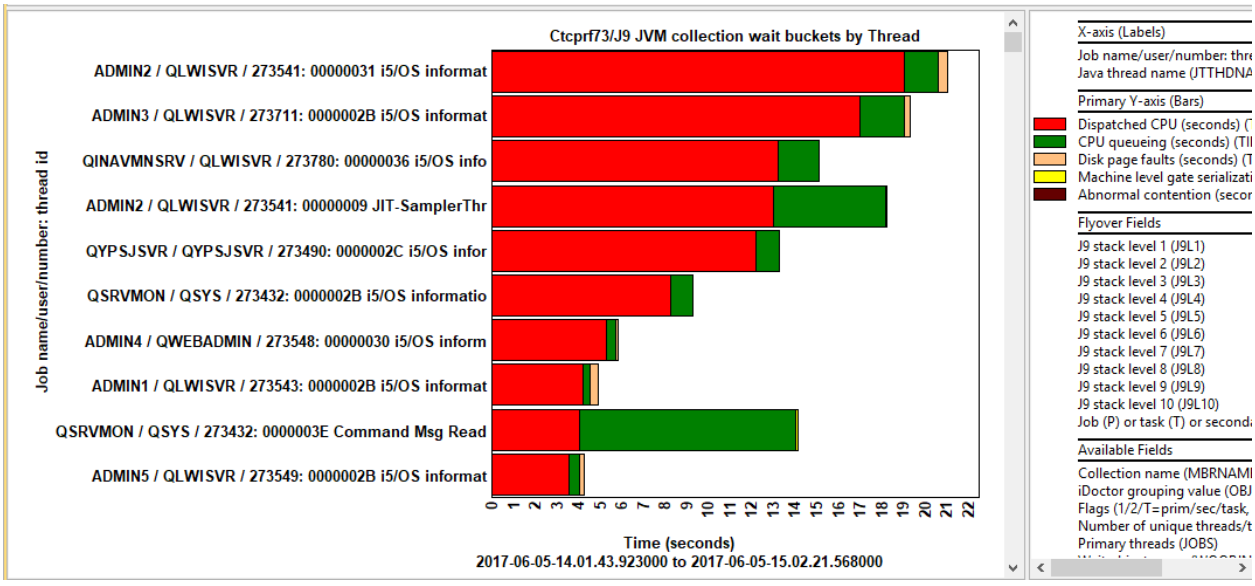
J9 JVM rankings

8.22.13.1 J9 JVM collection wait buckets by Thread

This graph provides a ranking of the J9 JVMs on Dispatched CPU by thread. All interesting wait buckets will appear on this graph like the Dispatched CPU rankings by thread graph.

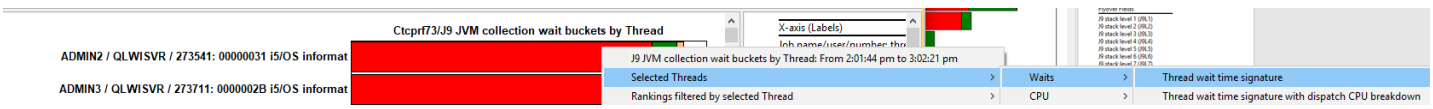
Tip: Next to each thread name the java thread name is also included.

Note: This graph requires the Collection summary analysis to be ran in order to appear!



J9 JVM collection wait buckets by thread

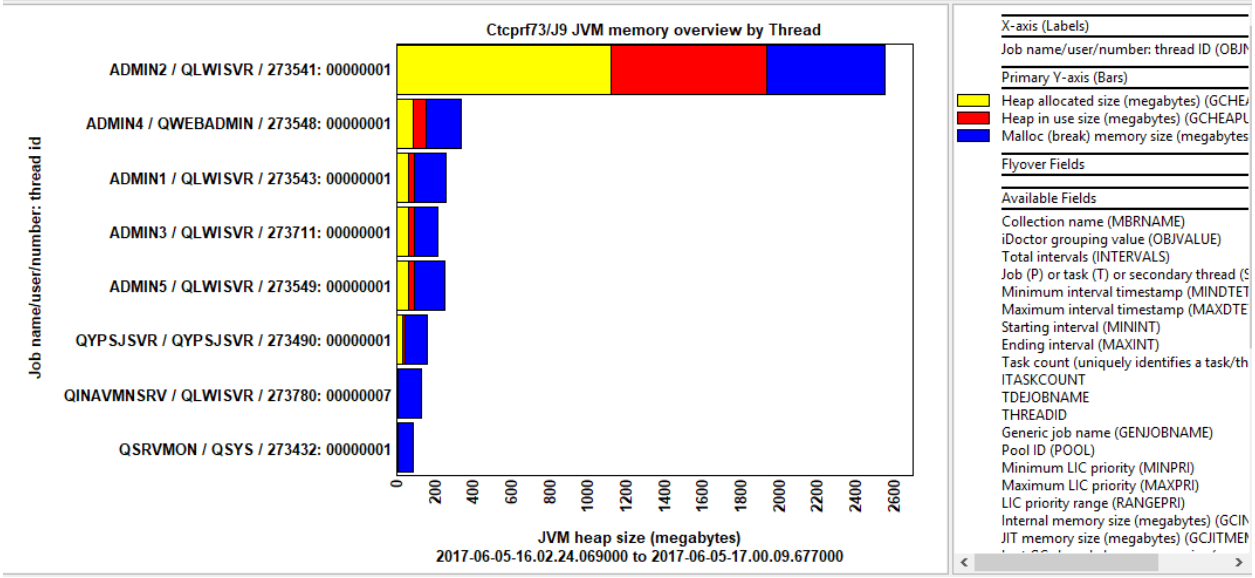
From this graph users can drill down again and graph the thread wait bucket contributions over time.



Selected Threads -> Waits -> Thread wait time signature drill down

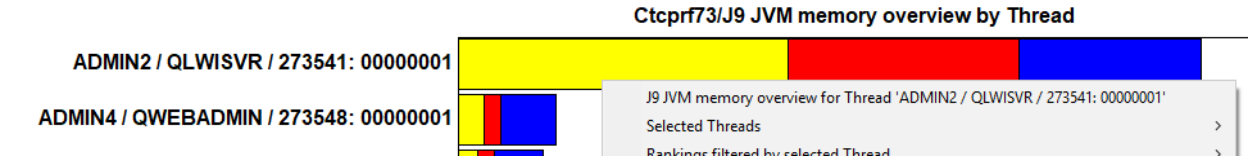
8.22.13.2 J9 JVM memory overview by thread

This graph adds up and ranks heap related metrics for each J9 JVM. You can right-click the desired thread/JVM and take the default drill down to see the same metrics over time.

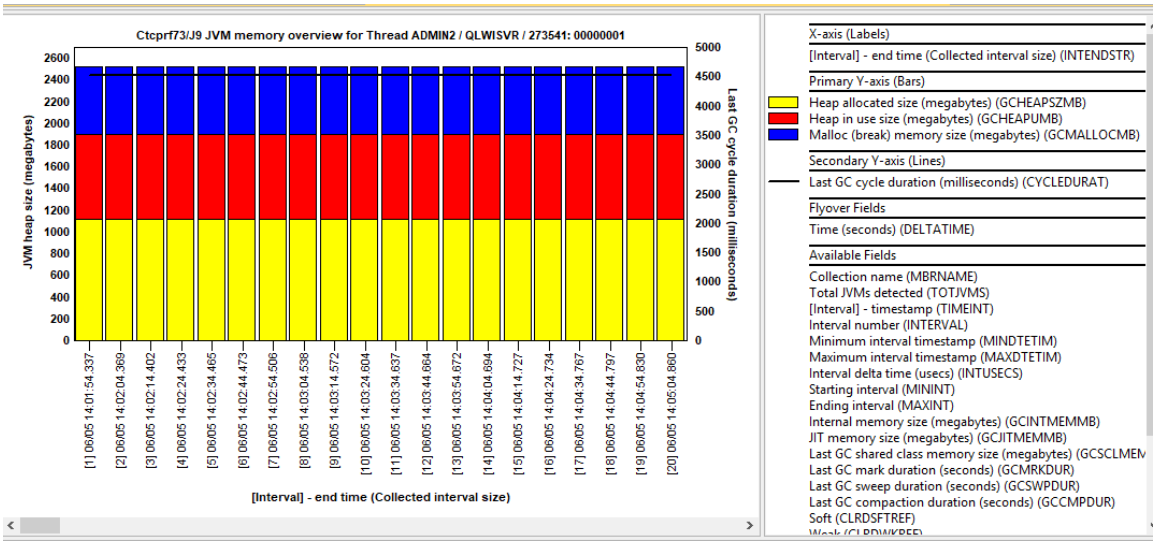


J9 JVM memory overview by thread

After drilling down to the next level…



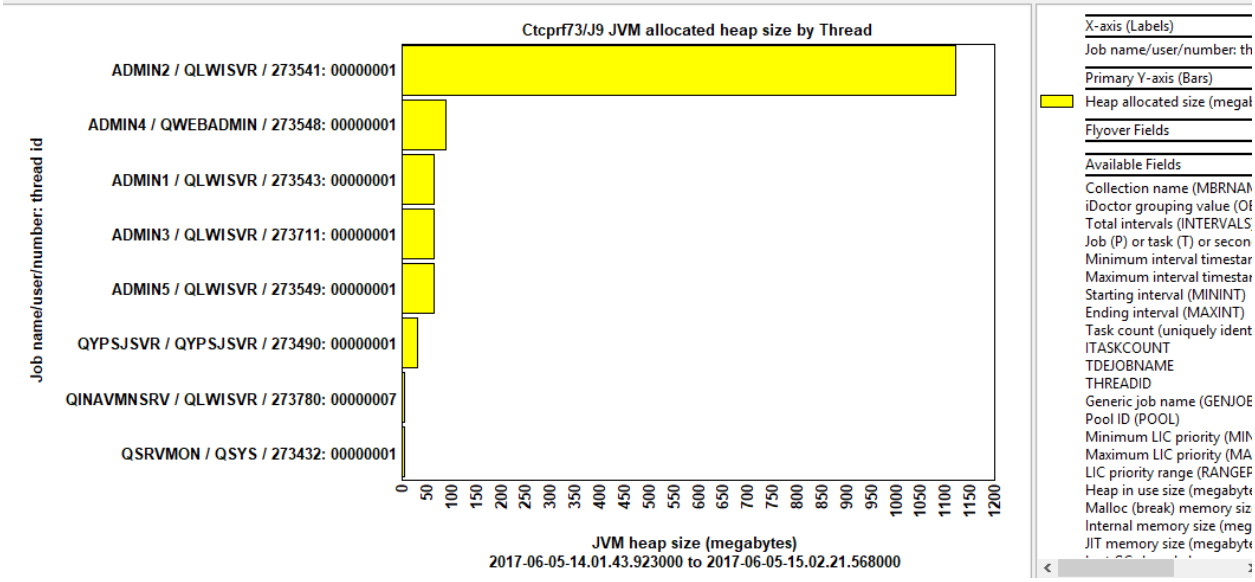
J9 JVM memory overview for Thread ADMIN2 drill-down



J9 JVM memory overview for Thread ADMIN2

8.22.13.3 J9 JVM allocated heap size by Thread

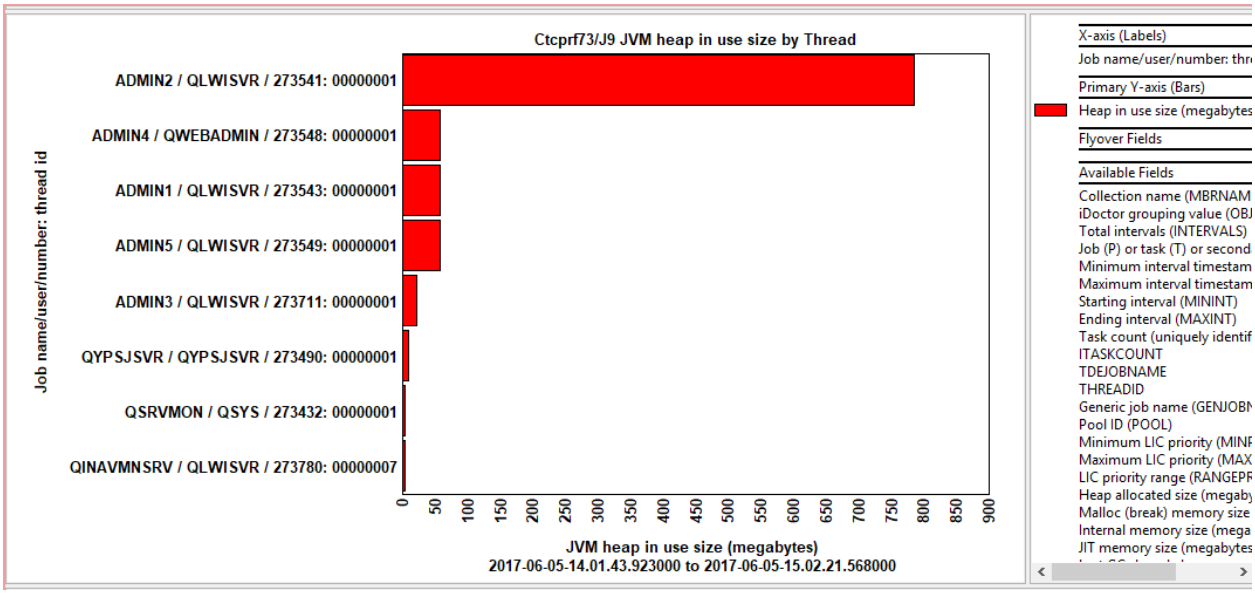
This graph displays just the J9 JVM allocated heap size (in megabytes) ranked by thread.



J9 JVM allocated heap size by thread

8.22.13.4 J9 JVM heap in use size by Thread

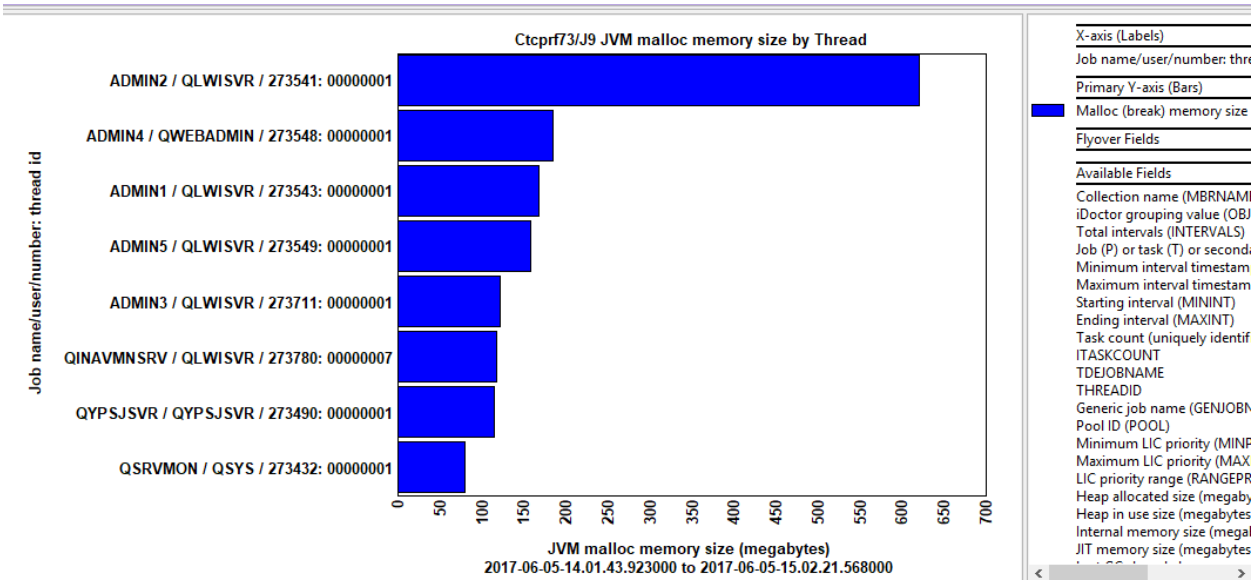
This graph displays the J9 JVM heap in use size (in megabytes) ranked by Thread.



J9 JVM heap in use size by thread

8.22.13.5 J9 JVM malloc memory size by Thread

This graph displays the J9 JVM malloc memory size (in megabytes) ranked by Thread.



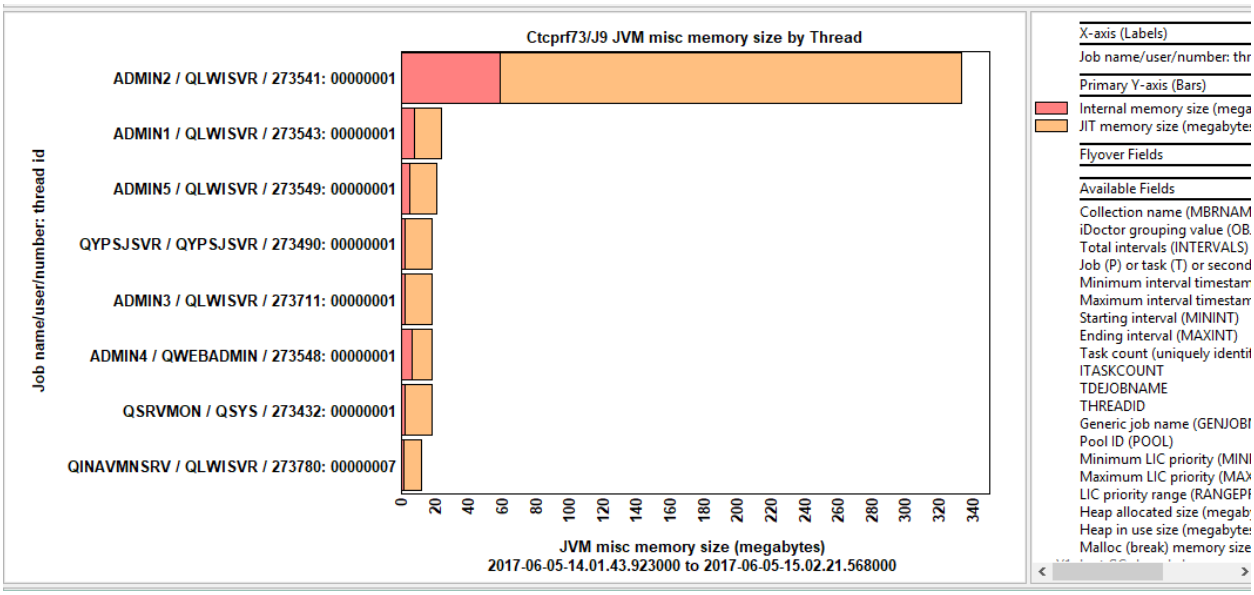
J9 JVM malloc memory size by thread

8.22.13.6 J9 JVM misc memory size by Thread

This graph displays the J9 JVM miscellaneous memory sizes (in megabytes) ranked by Thread.

The metrics included (in megabytes) are:

1) Internal memory size
2) JIT memory size
3) Last GC shared class memory size



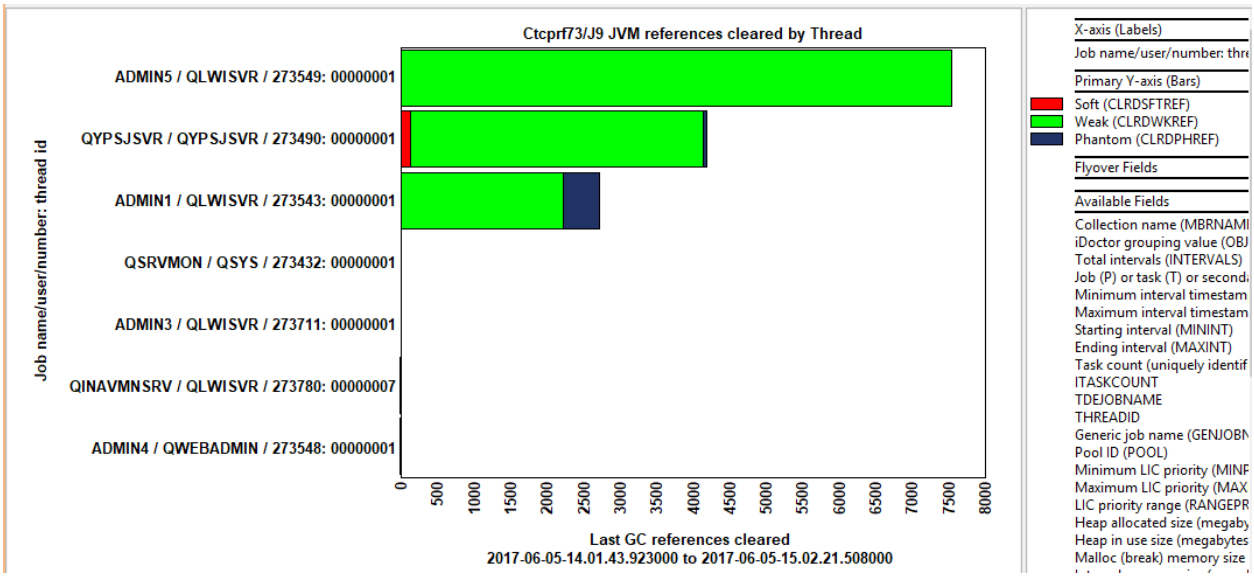
J9 JVM misc memory size by thread

8.22.13.7 J9 JVM references cleared by Thread

This graph displays the 4 types of JVM references cleared and ranks by the highest totals by thread..

The values are taken from file QAPYJWIJVM and include the following:

1) JMGCSFRCLR – GC soft references cleared
2) JMGCWKRCLR – GC weak references cleared
3) JMGCFNRCLR – GC finalizer references cleared
4) JMGCPHRCLR – GC phantom references cleared (Note: During testing this contained some negative values which appears to be a bug in the IBM i. This value may not be reliable until fixed.)



J9 JVM references cleared by thread

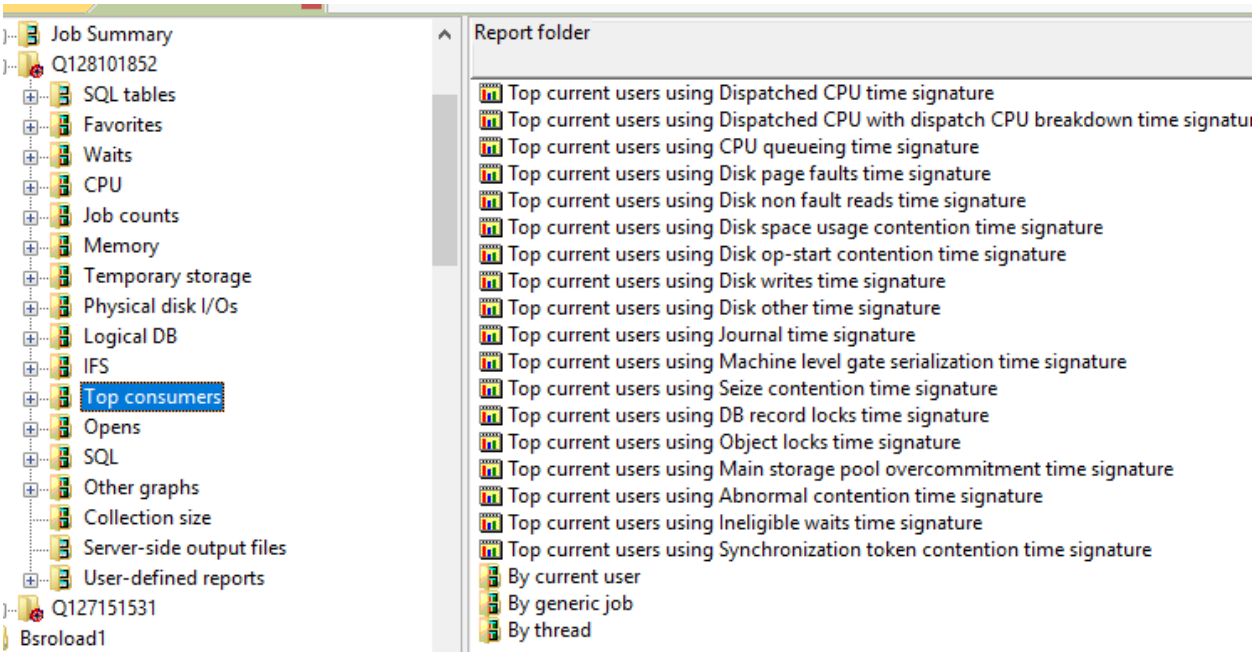
8.22.13.8 J9 JVM mark, sweep, compaction durations by Thread

Note: The values in the raw data in QAPYJWIJVM for this graph do not seem correct currently.

8.23 Top consumers

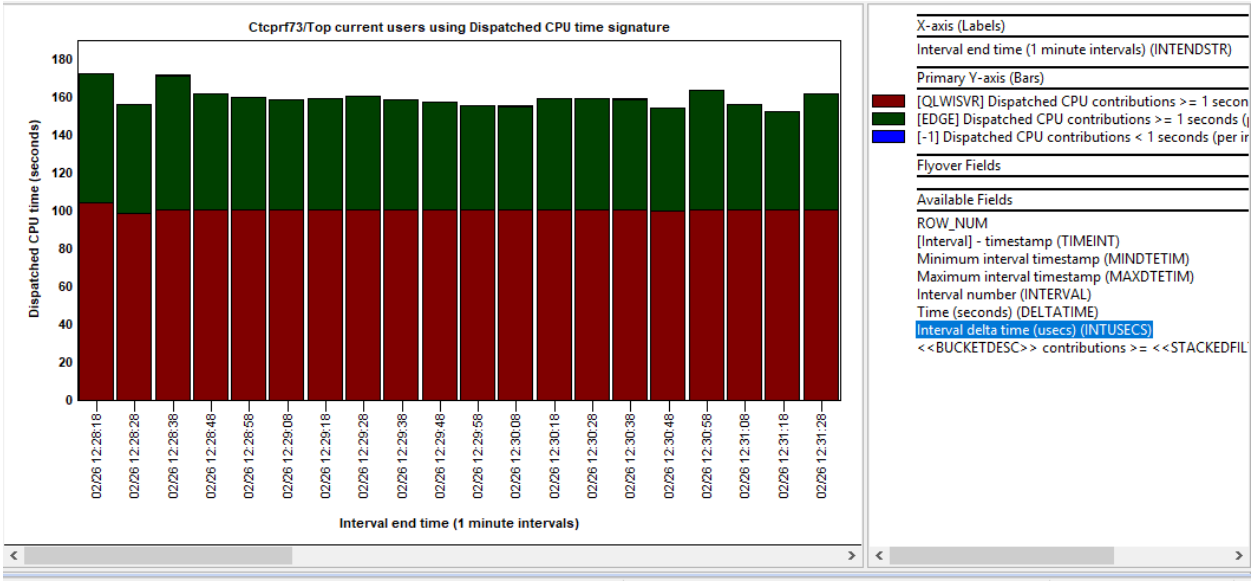
These graphs are a special-type of graph called “flattened graphs” that will show a variable number of top contributors of the desired metric over time. These show the top current users, generic jobs or threads using the desired wait bucket or CPU time.

The graphs work best with smaller number of contributors per bar (< 10 ideally) and will not handle many contributors well as the graphs can be very slow to load and visualize. However, a filter option exists to group many smaller values together. This defaults to 1 second and will group all users, generic jobs or threads having < 1 second of the desired wait bucket or CPU time together.



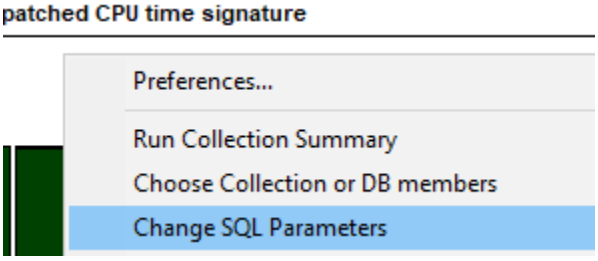
Top consumers folder

An example of this type of graph is:



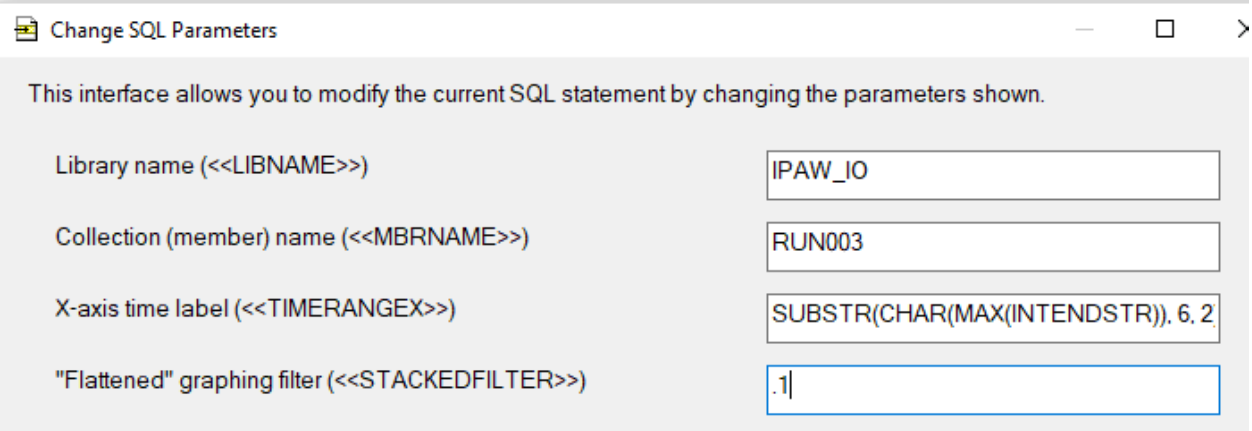
Top current users using Dispatched CPU time signature (1 second filter)

To change the filter, right-click the graph and use the Change SQL Parameter menu:

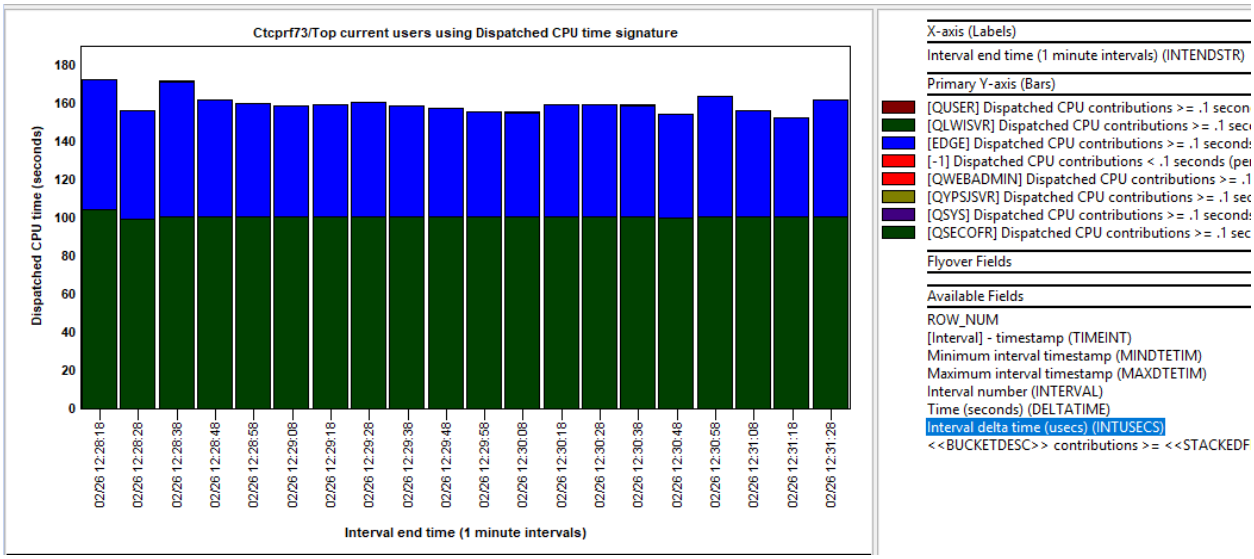


Change SQL Parameters menu

Here you can make the “Flattened” graphing filter value smaller to show more contributors and bigger to show less.



Change SQL Parameters window



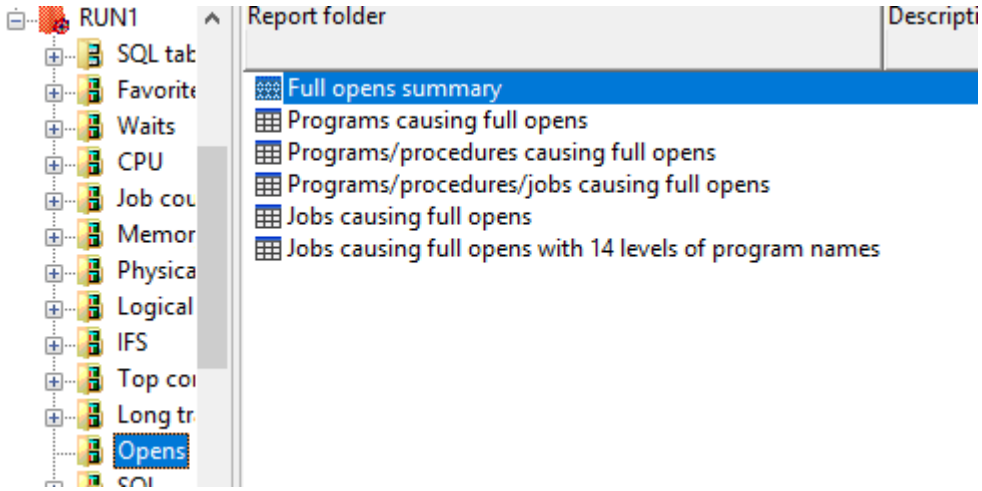
Top current users using Dispatched CPU time signature (.1 second filter)

8.24 Opens

This folder contains a set of reports relating to file opens and helps identify what programs are causing opens to occur. These statistics are based entirely on the Job Watcher call stacks that are captured each interval. It is normal to have no (or very little) data produced by these reports because of how Job Watcher works. Since Job Watcher is a snapshot taker and only captures a call stack at the end of each interval many opens will be missed.

To increase the numbers of call stacks captured where opens occurred, you will need to decrease the collection’s interval duration or collect data “as fast as possible.” By doing so, you can increase the number of opens caught in the call stacks and improve the analysis you can do with these reports.

Another option is to use PEX (DB opens analysis) instead.



Opens folder

8.24.1 Full opens summary

This report identifies the types of opens most commonly found in the current collection.

/BSMENGES/RUN1/Full opens summary - #1

Hit count (PROCCNT)	Program open type (OPENTYPE)	Procedure (PROCNAME)
65	QDBOPEN (SQL or Native)	QDBOPEN
43	C Native	_Ropen
22	CQE	OPENUFCB

Full opens summary

8.24.2 Programs causing full opens

This report shows which programs most commonly caused the full opens found in the Job Watcher call stacks sampled.

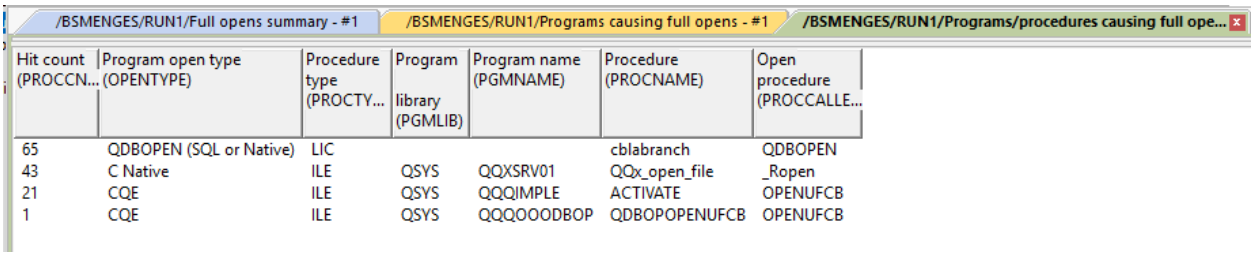
/BSMENGES/RUN1/Full opens summary - #1 | /BSMENGES/RUN1/Programs causing full opens - #1

Hit count (PROCCN...	Program open type (OPENTYPE)	Procedure type (PROCTY...	Program library (PGMLIB)	Program name (PGMNAME)
65	QDBOPEN (SQL or Native)	LIC		
43	C Native	ILE	QSYS	QQXSRV01
21	CQE	ILE	QSYS	QQQIMPLE
1	CQE	ILE	QSYS	QQQOOODBOP

Programs causing full opens

8.24.3 Programs/procedures causing full opens

This report shows which programs and procedures most commonly caused the full opens found in the Job Watcher call stacks sampled.



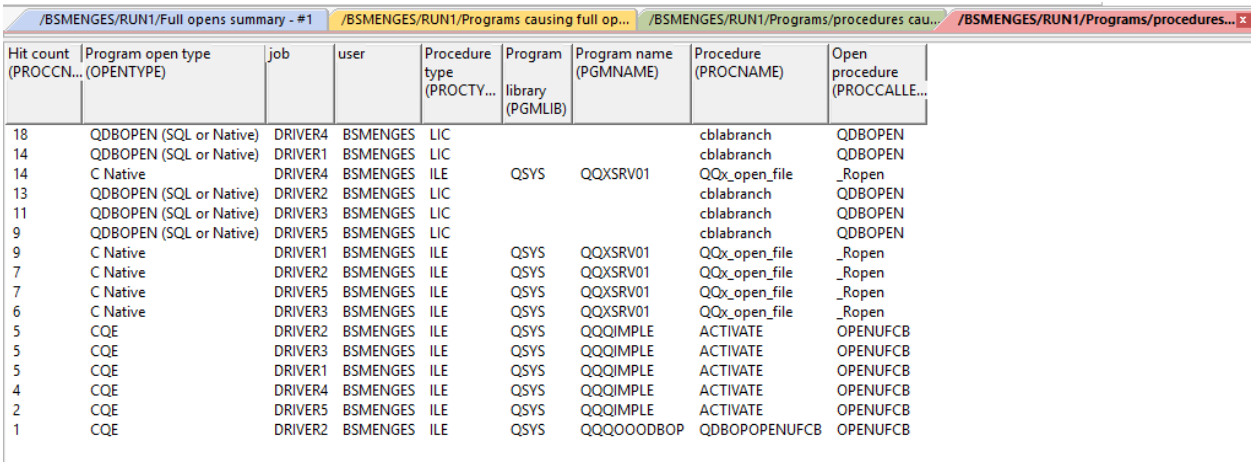
/BSMENGES/RUN1/Full opens summary - #1 | /BSMENGES/RUN1/Programs causing full opens - #1 | /BSMENGES/RUN1/Programs/procedures causing full ope...

Hit count (PROCCN...	Program open type (OPENTYPE)	Procedure type (PROCTY...	Program library (PGMLIB)	Program name (PGMNAME)	Procedure (PROCNAME)	Open procedure (PROCCALLE...
65	QDBOPEN (SQL or Native)	LIC			cblabranch	QDBOPEN
43	C Native	ILE	QSYS	QQXSRV01	QQx_open_file	_Ropen
21	CQE	ILE	QSYS	QQQIMPLE	ACTIVATE	OPENUFCB
1	CQE	ILE	QSYS	QQQOOODBOP	QDBOPOPENUFCB	OPENUFCB

Programs/procedures causing full opens

8.24.4 Programs/procedures/jobs causing full opens

This report is the same as the previous one but also includes the job name and job user to the mix.



/BSMENGES/RUN1/Full opens summary - #1 | /BSMENGES/RUN1/Programs causing full op... | /BSMENGES/RUN1/Programs/procedures cau... | /BSMENGES/RUN1/Programs/procedures...

Hit count (PROCCN...	Program open type (OPENTYPE)	job	user	Procedure type (PROCTY...	Program library (PGMLIB)	Program name (PGMNAME)	Procedure (PROCNAME)	Open procedure (PROCCALLE...
18	QDBOPEN (SQL or Native)	DRIVER4	BSMENGES	LIC			cblabranch	QDBOPEN
14	QDBOPEN (SQL or Native)	DRIVER1	BSMENGES	LIC			cblabranch	QDBOPEN
14	C Native	DRIVER4	BSMENGES	ILE	QSYS	QQXSRV01	QQx_open_file	_Ropen
13	QDBOPEN (SQL or Native)	DRIVER2	BSMENGES	LIC			cblabranch	QDBOPEN
11	QDBOPEN (SQL or Native)	DRIVER3	BSMENGES	LIC			cblabranch	QDBOPEN
9	QDBOPEN (SQL or Native)	DRIVER5	BSMENGES	LIC			cblabranch	QDBOPEN
9	C Native	DRIVER1	BSMENGES	ILE	QSYS	QQXSRV01	QQx_open_file	_Ropen
7	C Native	DRIVER2	BSMENGES	ILE	QSYS	QQXSRV01	QQx_open_file	_Ropen
7	C Native	DRIVER5	BSMENGES	ILE	QSYS	QQXSRV01	QQx_open_file	_Ropen
6	C Native	DRIVER3	BSMENGES	ILE	QSYS	QQXSRV01	QQx_open_file	_Ropen
5	CQE	DRIVER2	BSMENGES	ILE	QSYS	QQQIMPLE	ACTIVATE	OPENUFCB
5	CQE	DRIVER3	BSMENGES	ILE	QSYS	QQQIMPLE	ACTIVATE	OPENUFCB
5	CQE	DRIVER1	BSMENGES	ILE	QSYS	QQQIMPLE	ACTIVATE	OPENUFCB
4	CQE	DRIVER4	BSMENGES	ILE	QSYS	QQQIMPLE	ACTIVATE	OPENUFCB
2	CQE	DRIVER5	BSMENGES	ILE	QSYS	QQQIMPLE	ACTIVATE	OPENUFCB
1	CQE	DRIVER2	BSMENGES	ILE	QSYS	QQQOOODBOP	QDBOPOPENUFCB	OPENUFCB

Programs/procedures/jobs causing full opens

8.24.5 Jobs causing full opens

This report shows the job/thread (and SQL statement) most commonly causing full opens based on the call stacks sampled in the collection.

/BSMENGES/RUN1/Jobs causing full opens - #1

Hit count (PROCCN...	TDEJOBNAME	THREADID	SQL statement (SQLSTMT)	Task count (uniquely identifies a task/thread) (TASKCOUNT)
2	DRIVER4 BSMENGES 011764	0000000000000004		14,333
2	DRIVER4 BSMENGES 011769	0000000000000005		14,339
2	DRIVER4 BSMENGES 011789	0000000000000002		14,359
2	DRIVER1 BSMENGES 011851	0000000000000005		14,422
1	DRIVER5 BSMENGES 011259	0000000000000001		13,780
1	DRIVER3 BSMENGES 011853	0000000000000005		14,424
1	DRIVER1 BSMENGES 011240	00000000000000D1		13,761
1	DRIVER2 BSMENGES 011727	0000000000000003		14,296
1	DRIVER1 BSMENGES 011726	0000000000000005	INSERT INTO "BSMENGES"/"SLOT13" SELECT * FROM QTEMP/QQXTEMPA	14,295
1	DRIVER3 BSMENGES 011627	0000000000000002		14,156
1	DRIVER2 BSMENGES 011737	0000000000000005		14,306
1	DRIVER5 BSMENGES 011254	0000000000000001		13,775
1	DRIVER1 BSMENGES 011826	0000000000000002		14,397
1	DRIVER4 BSMENGES 011578	0000000000000003		14,107
1	DRIVER2 BSMENGES 011697	0000000000000004		14,266
1	DRIVER3 BSMENGES 011773	0000000000000006	INSERT INTO "BSMENGES"/"SLOT12" SELECT * FROM QTEMP/QQXTEMP|	14,343
1	DRIVER2 BSMENGES 011827	0000000000000006	INSERT INTO "BSMENGES"/"SLOT13" SELECT * FROM QTEMP/QQXTEMP(	14,398
1	DRIVER1 BSMENGES 011746	0000000000000007	INSERT INTO "BSMENGES"/"SLOT12" SELECT * FROM QTEMP/QQXTEMP|	14,315
1	DRIVER4 BSMENGES 011633	0000000000000002		14,162
1	DRIVER2 BSMENGES 011692	0000000000000006		14,261
1	DRIVER5 BSMENGES 011639	0000000000000002		14,168

Jobs causing full opens

8.24.6 Jobs causing full opens with 14 levels of program names

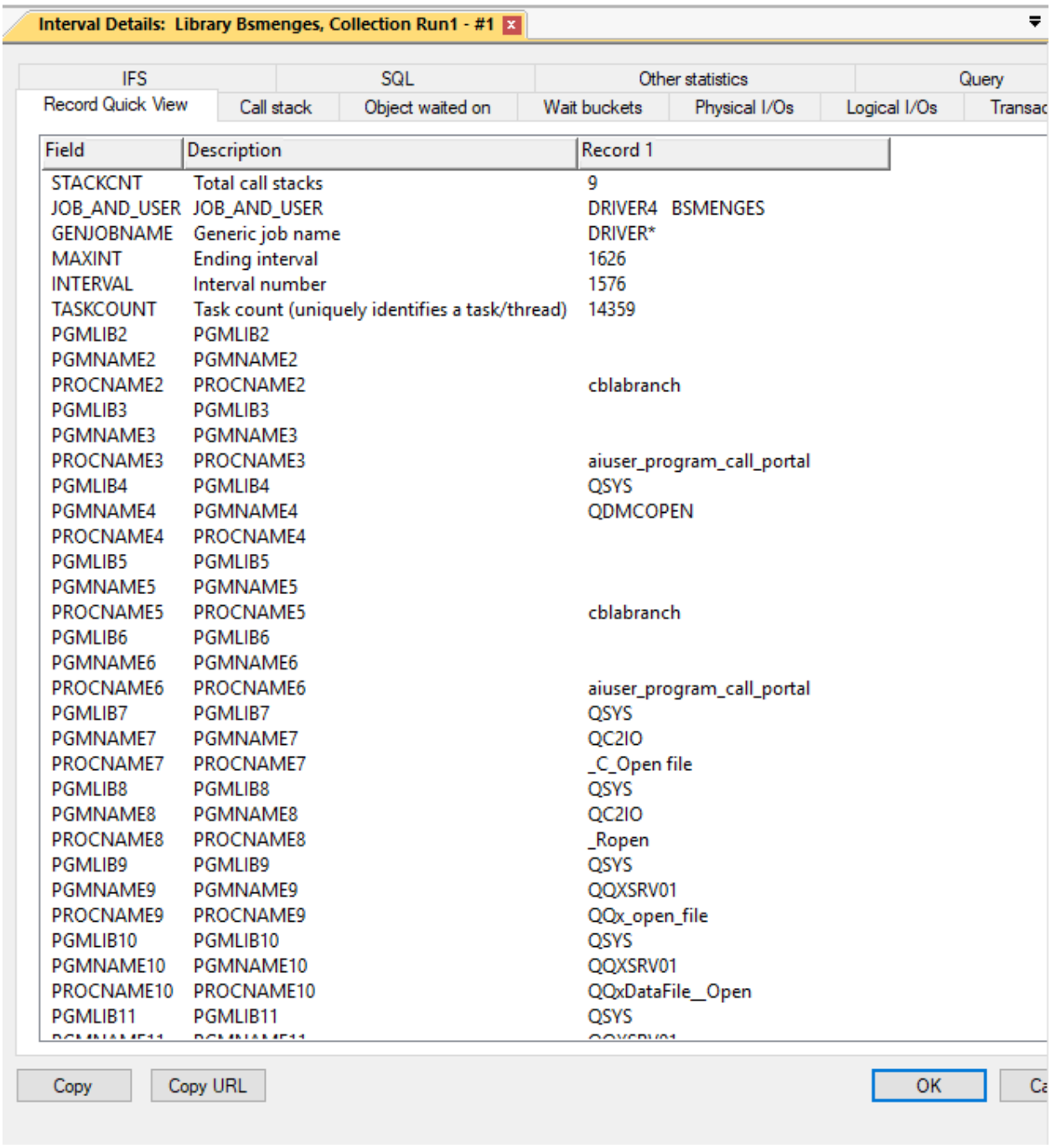
This report summarizes the jobs causing the most full opens with the 14 call levels occurring in the stack before the open.

/BSMENGES/RUN1/Jobs causing full opens with 14 levels of program names - #1

Total call stacks (STACKCNT)	JOB_AND_USER	Generic job name (GENJOBNA...	Ending interval (MAXIN...	Interval number (INTERVA...	Task count (uniquely identifies a task/thread) (TASKCOUNT)	PGMLIB2	PGMNAME2	PROCNAME2	PGMLIB3	PGMNAME3	PROCNAME3
9	DRIVER4 BSMENGES	DRIVER*	1,626	1,576	14,359			cblabranch			aiuser_program_call_portal
6	DRIVER1 BSMENGES	DRIVER*	1,650	1,585	14,422			cblabranch			aiuser_program_call_portal
5	DRIVER2 BSMENGES	DRIVER*	1,636	1,409	14,398			cblabranch			aiuser_program_call_portal
5	DRIVER3 BSMENGES	DRIVER*	1,656	1,187	14,424			cblabranch			aiuser_program_call_portal
5	DRIVER3 BSMENGES	DRIVER*	1,626	1,404	14,343			cblabranch			aiuser_program_call_portal
5	DRIVER1 BSMENGES	DRIVER*	1,649	1,187	14,422			cblabranch			aiuser_program_call_portal
5	DRIVER4 BSMENGES	DRIVER*	1,626	1,404	14,359			cblabranch			aiuser_program_call_portal
4	DRIVER5 BSMENGES	DRIVER*	1,608	1,404	14,360			cblabranch			aiuser_program_call_portal
4	DRIVER4 BSMENGES	DRIVER*	1,196	1,187	13,819			cblabranch			aiuser_program_call_portal
4	DRIVER2 BSMENGES	DRIVER*	1,608	1,431	14,337			cblabranch			aiuser_program_call_portal
3	DRIVER5 BSMENGES	DRIVER*	1,613	1,409	14,365			cblabranch			aiuser_program_call_portal
3	DRIVER2 BSMENGES	DRIVER*	1,613	1,404	14,306			cblabranch			aiuser_program_call_portal
3	DRIVER1 BSMENGES	DRIVER*	1,608	1,419	14,315			cblabranch			aiuser_program_call_portal
2	DRIVER5 BSMENGES	DRIVER*	1,187	1,187	13,780			cblabranch			aiuser_program_call_portal
1	DRIVER3 BSMENGES	DRIVER*	1,576	1,576	14,267			cblabranch			aiuser_program_call_portal

Jobs causing full opens with 14 levels of program names

Double-click a row to get into the Record Quick View to make the display of the call stack information easier.



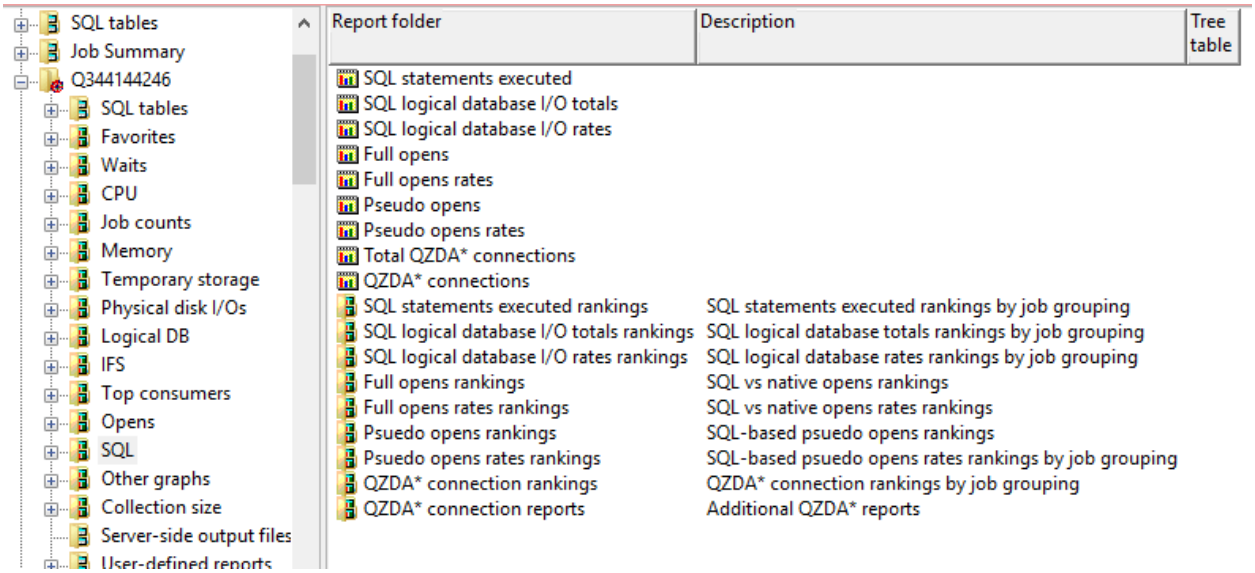
Record Quick View

Note: The call stack tab may not work as a drill down from this report, because the taskcount listed in the report is only a single instance that occurred which may not have had an open at the interval range listed. So you may need to use the arrows to navigate through the intervals to find the call stack with the open (if desired.)

8.25 SQL

This folder contains reports related to SQL usage. It contains both overview graphs and ranking graphs against the desired metrics.

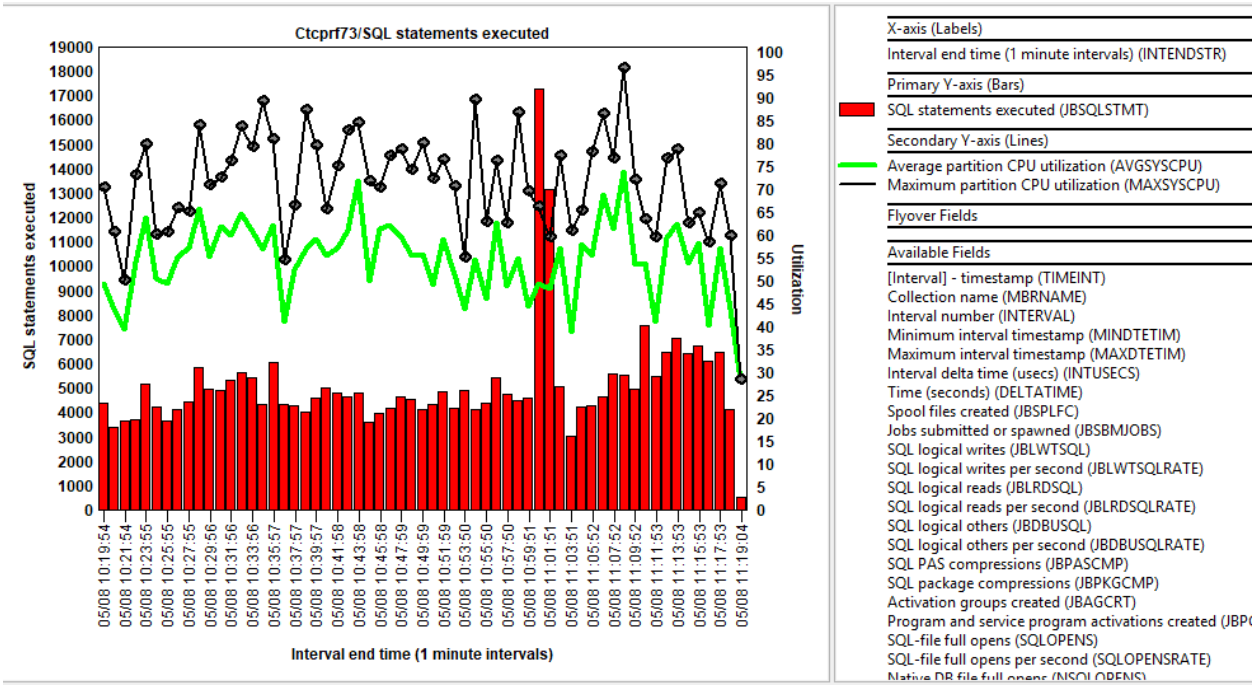
Note: Most of these graphs are only available at 7.2+.



SQL Folder

8.25.1 SQL statements executed

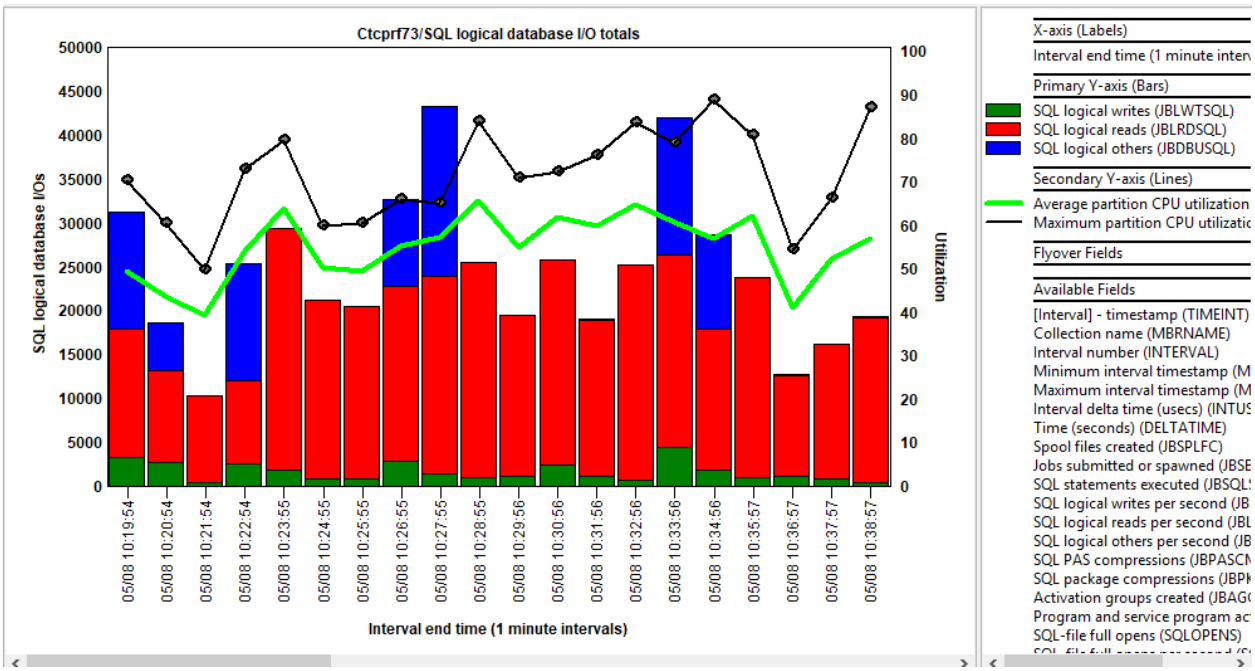
This graph shows a total count of SQL statements executed across all jobs on the system over time.



SQL statements executed

8.25.2 SQL logical database I/O totals

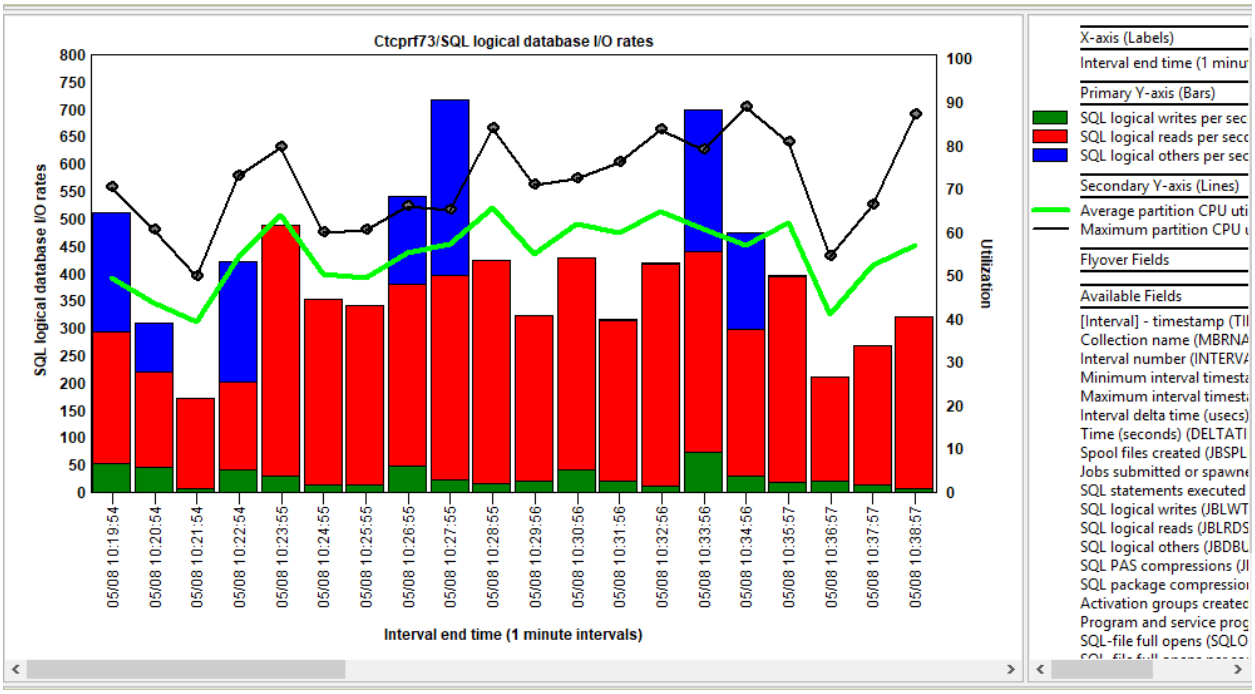
This graph displays SQL-related logical DB operations including: writes, reads and others (updates + deletes.)



SQL logical database I/O totals

8.25.3 SQL logical database I/O rates

This graph displays SQL-related logical DB operation rates including: writes, reads and others (updates + deletes.)

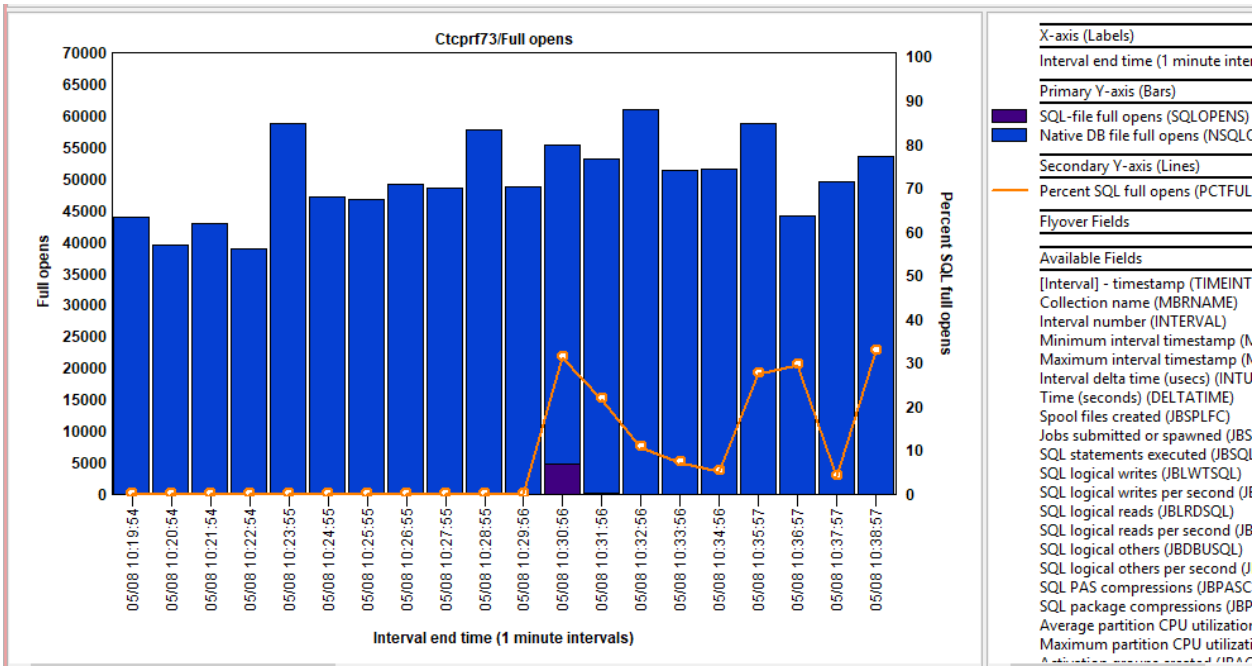


SQL logical database I/O rates

8.25.4 Full opens

This graph displays the total full opens occurring for both native and SQL driven opens over time.

The percentage of total SQL causing opens is shown on the second Y-axis. Note that if no native opens occur in those intervals then the percentage will be 100% even if the number of SQL-driven full opens is very small.

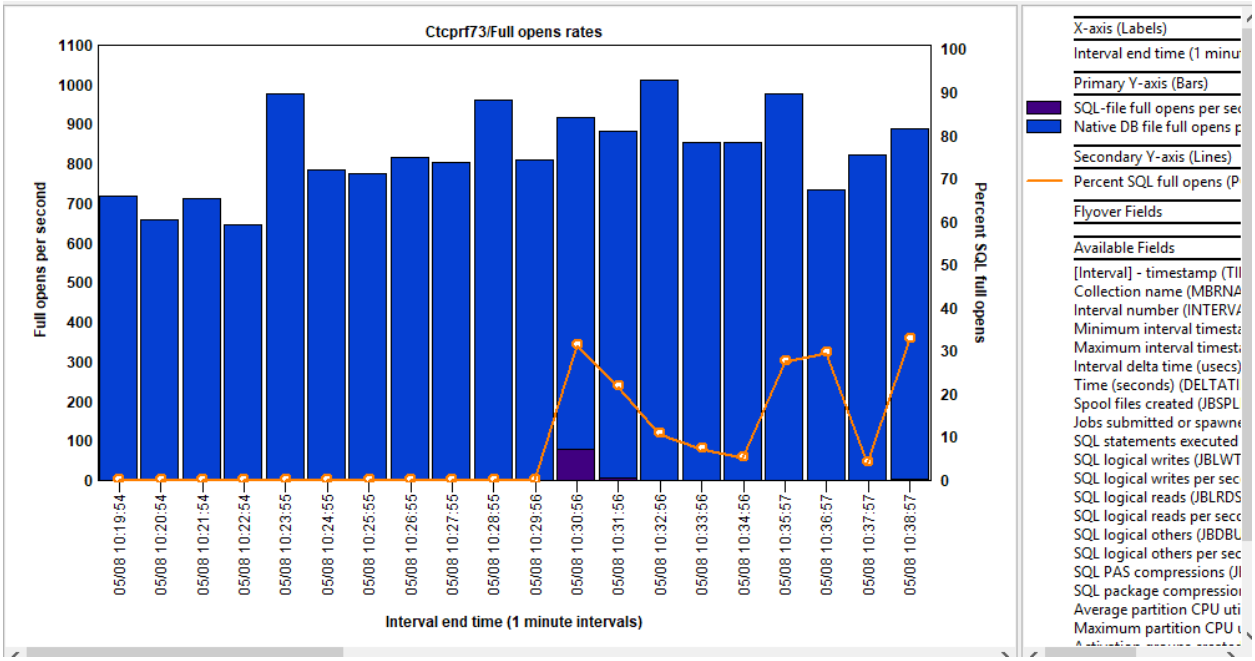


Full opens

8.25.5 Full opens rates

This graph displays the full opens per second occurring for both native and SQL driven opens over time.

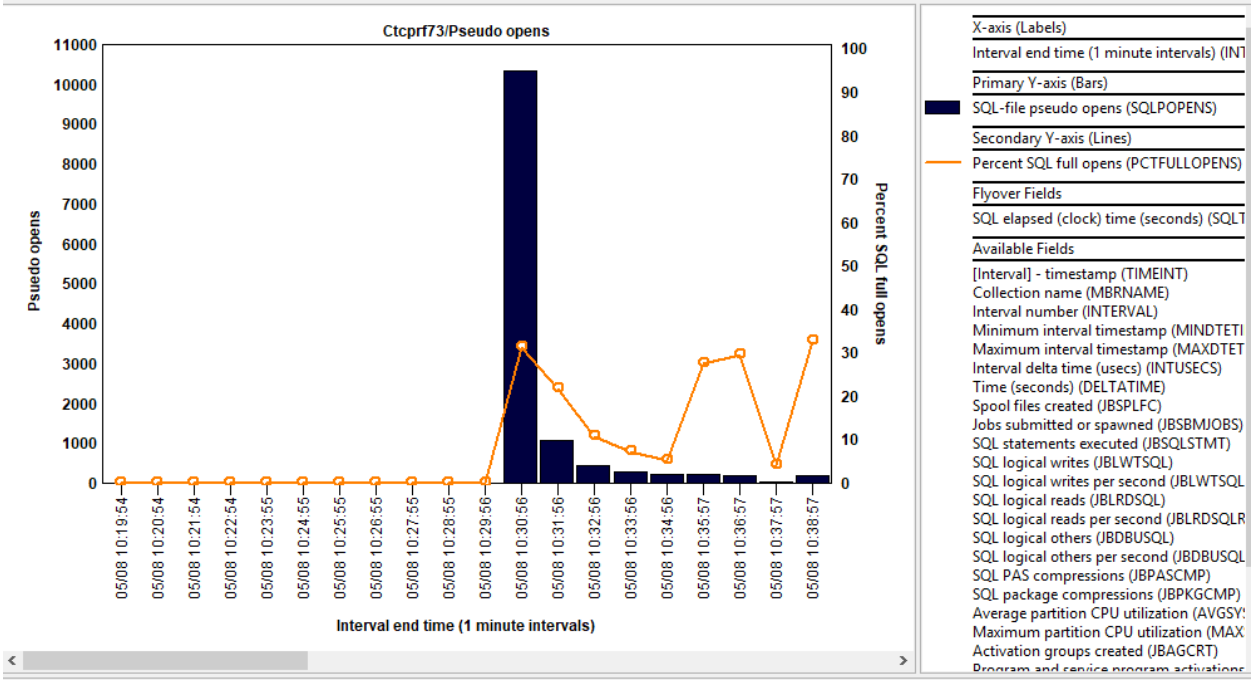
The percentage of SQL full opens of the total SQL opens (pseudo + full) is shown on the second Y-axis. Note that if no or pseudo opens occurred in those intervals then the percentage will be 100% even if the number of SQL full opens is very small.



Full opens rates

8.25.6 Psuedo opens

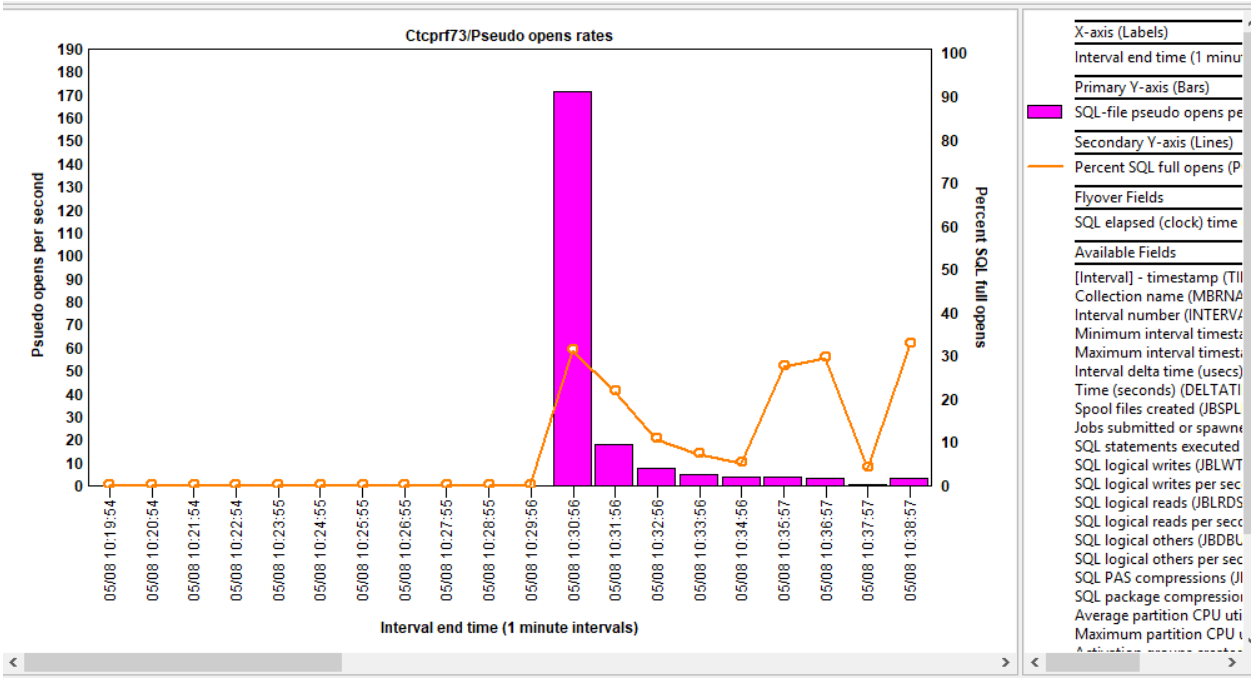
This graph shows the SQL pseudo opens that occurred over time. These are less costly than SQL full opens. The percentage of SQL full opens of the total SQL opens (pseudo + full) is shown on the second Y-axis.



Psuedo opens

8.25.7 Psuedo opens rates

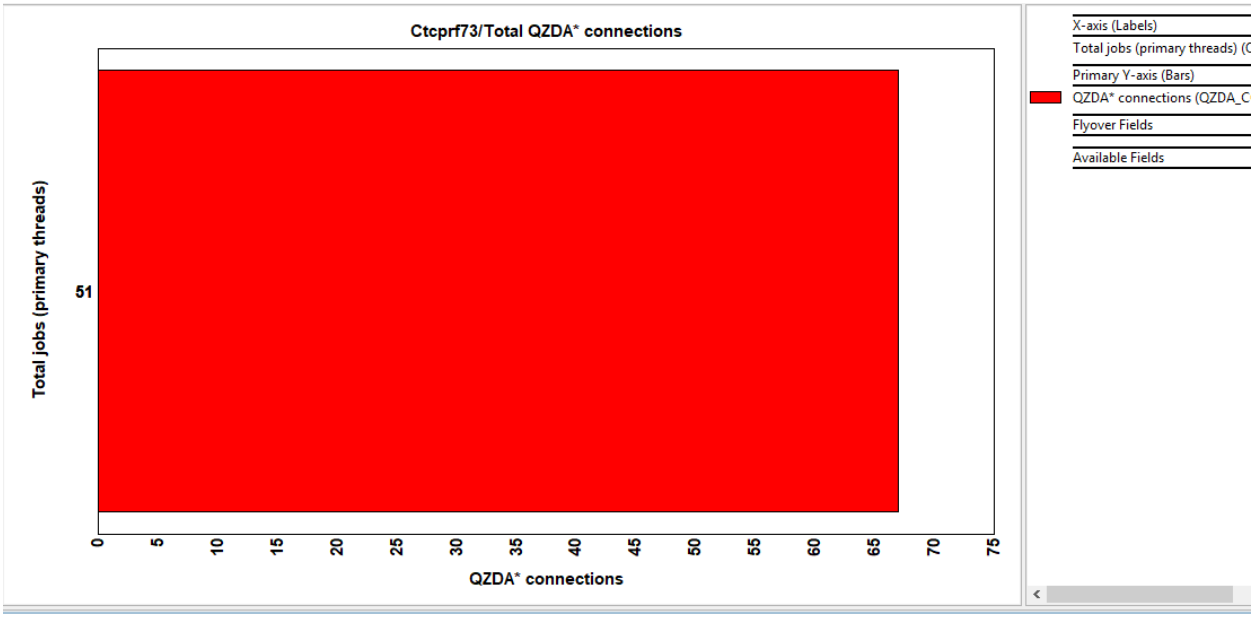
This graph shows the SQL pseudo open per second that occurred over time. These are less costly than SQL full opens. The percentage of SQL full opens of the total SQL opens (pseudo + full) is shown on the second Y-axis.



Psuedo opens rates

8.25.8 Total QZDA* connections

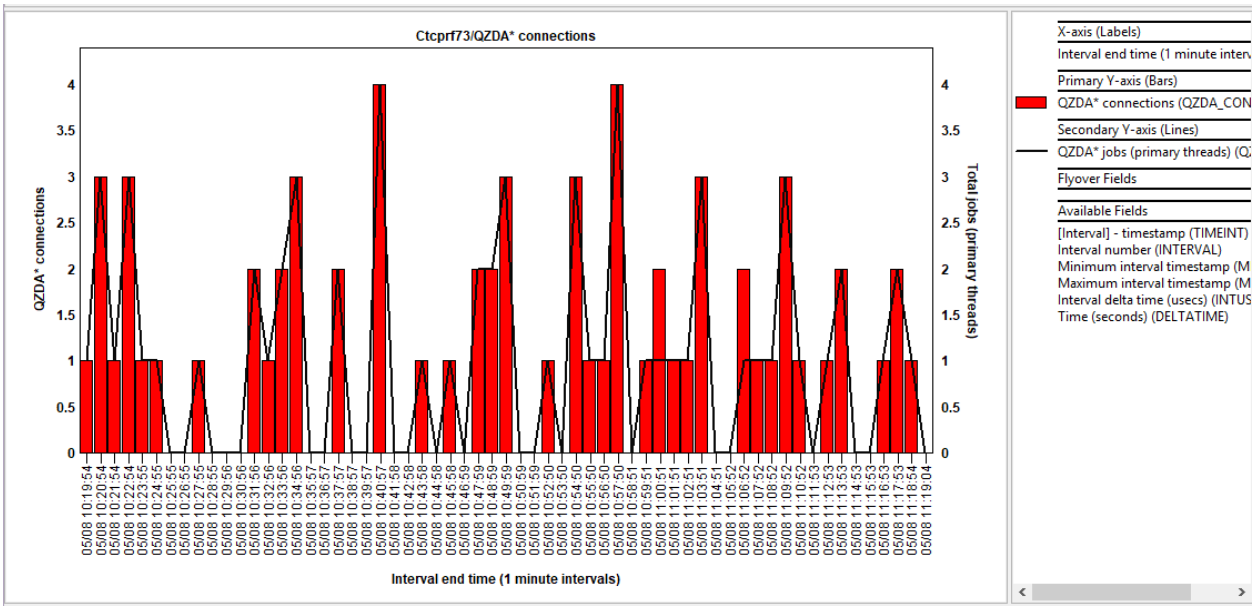
This graph simply displays the total unique QZDA connections (servicing potentially different users) and number of jobs (primary threads) serving them.



Total QZDA connections*

8.25.9 QZDA* connections

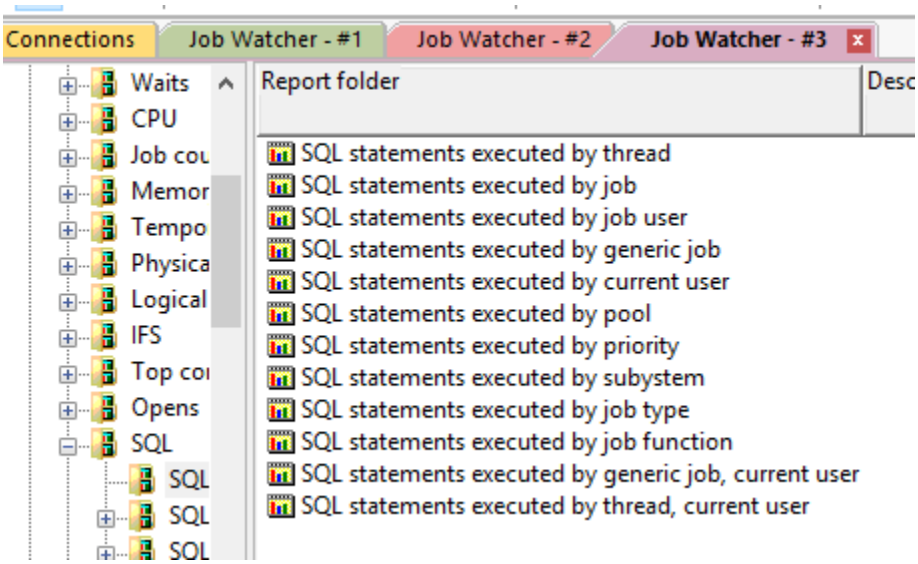
This graph displays the total QZDA* connections detected over time.



QZDA connections*

8.25.10 SQL statements executed rankings

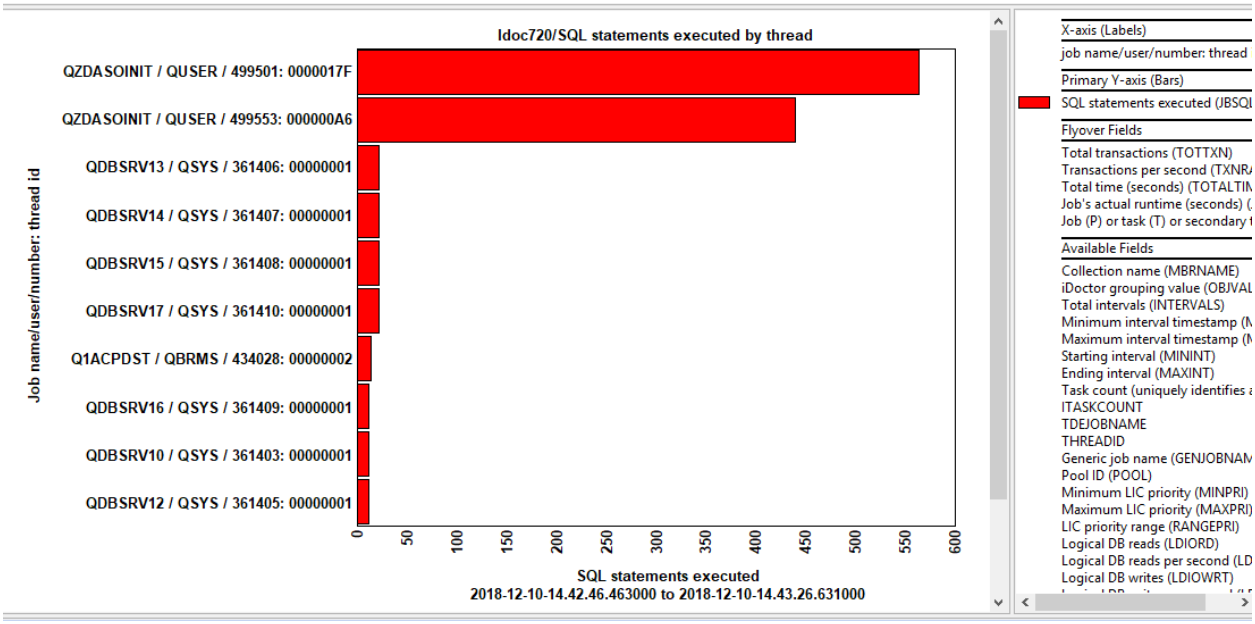
These graphs display the total SQL statements executed ranked by various job groupings.



SQL -> SQL statements executed rankings

8.25.10.1 SQL statements executed by thread

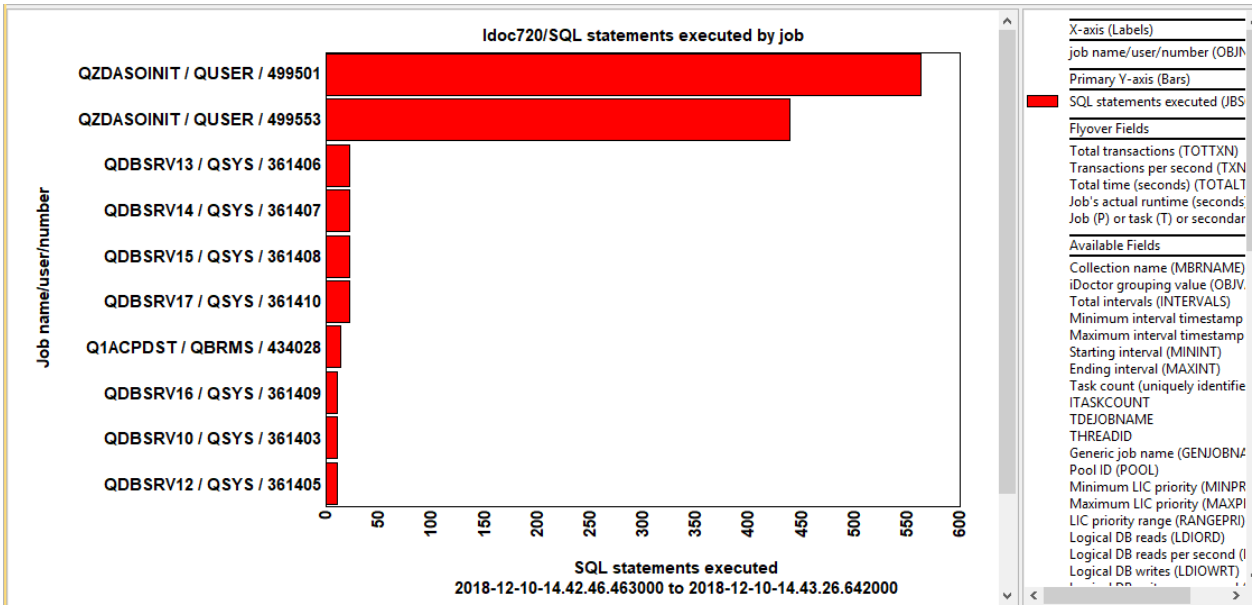
This graph ranks the SQL statements executed in the selected time period by thread or system task.



SQL statements executed by thread

8.25.10.2 SQL statements executed by job

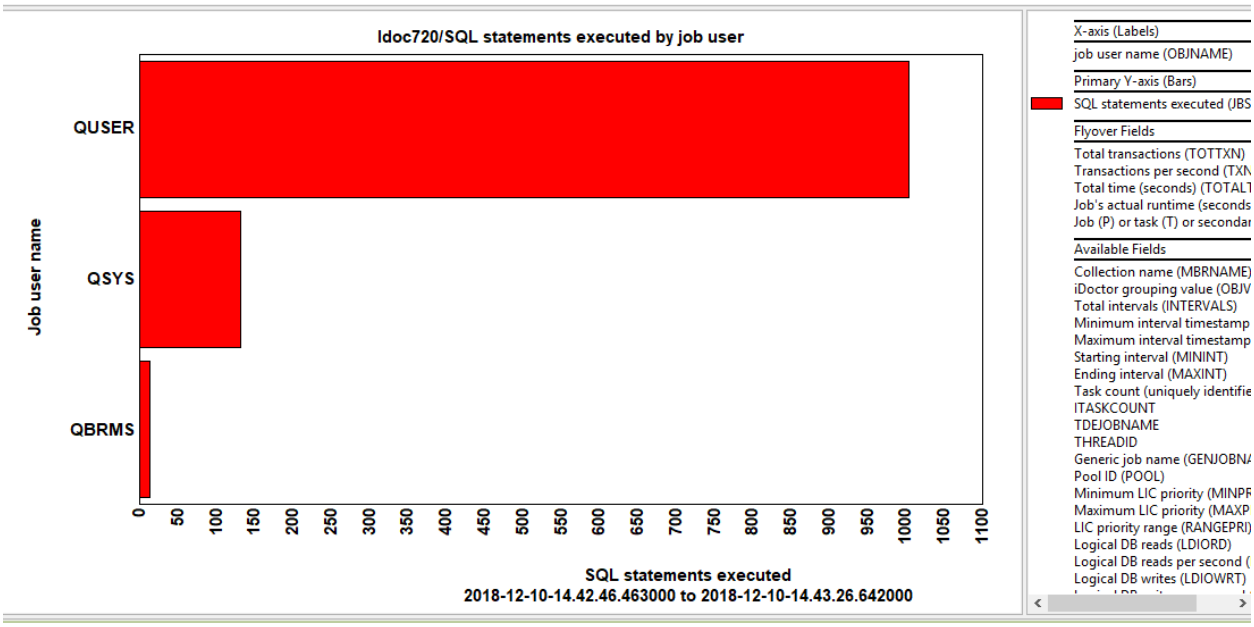
This graph ranks the SQL statements executed in the selected time period by primary thread. Jobs that have multiple threads are all added up together.



SQL statements executed by job

8.25.10.3 SQL statements executed by job user

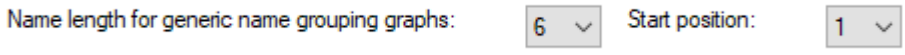
This graph ranks the SQL statements executed in the selected time period by job user.

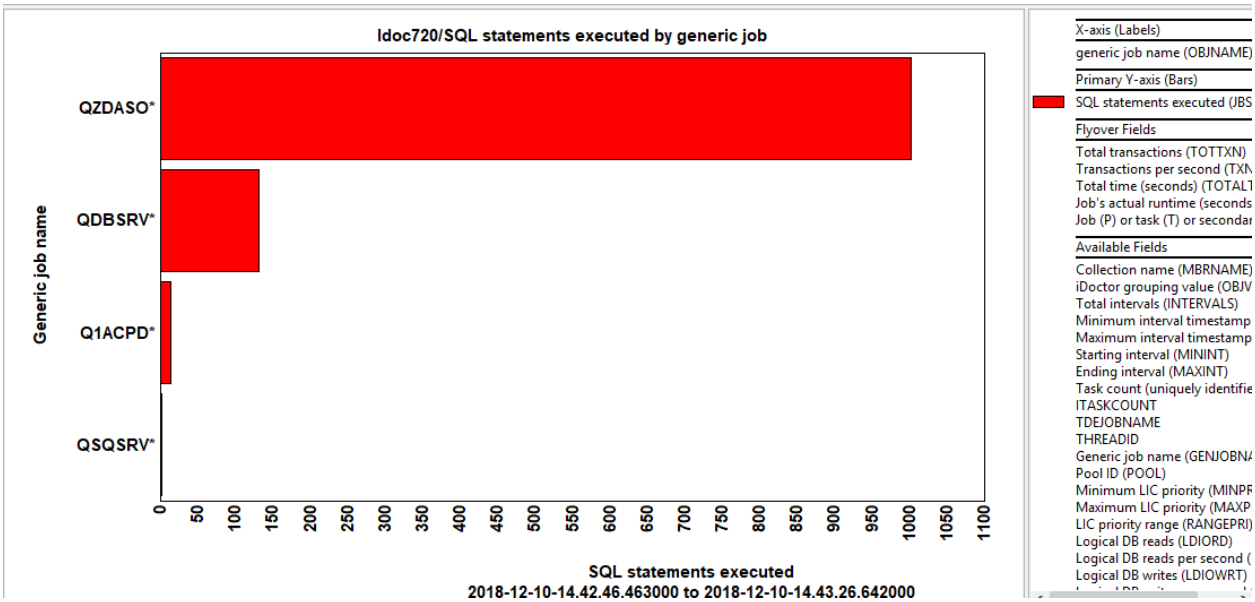


SQL statements executed by job user

8.25.10.4 SQL statements executed by generic job

This graph ranks the SQL statements executed in the selected time period by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

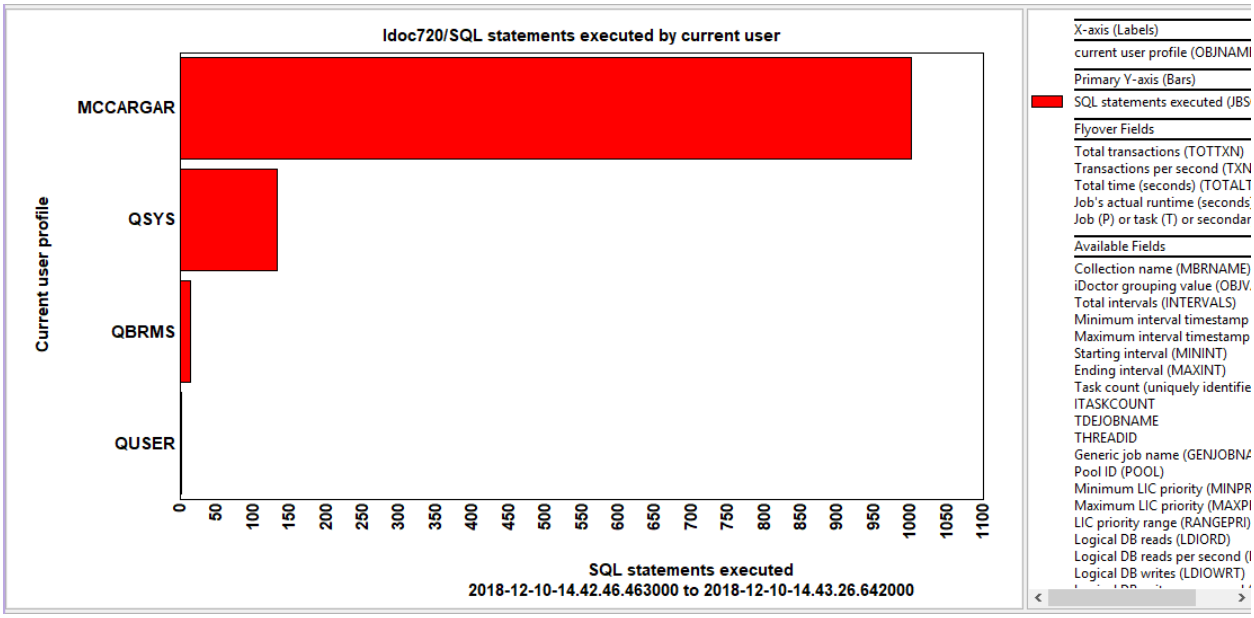




SQL statements executed by generic job

8.25.10.5 SQL statements executed by current user

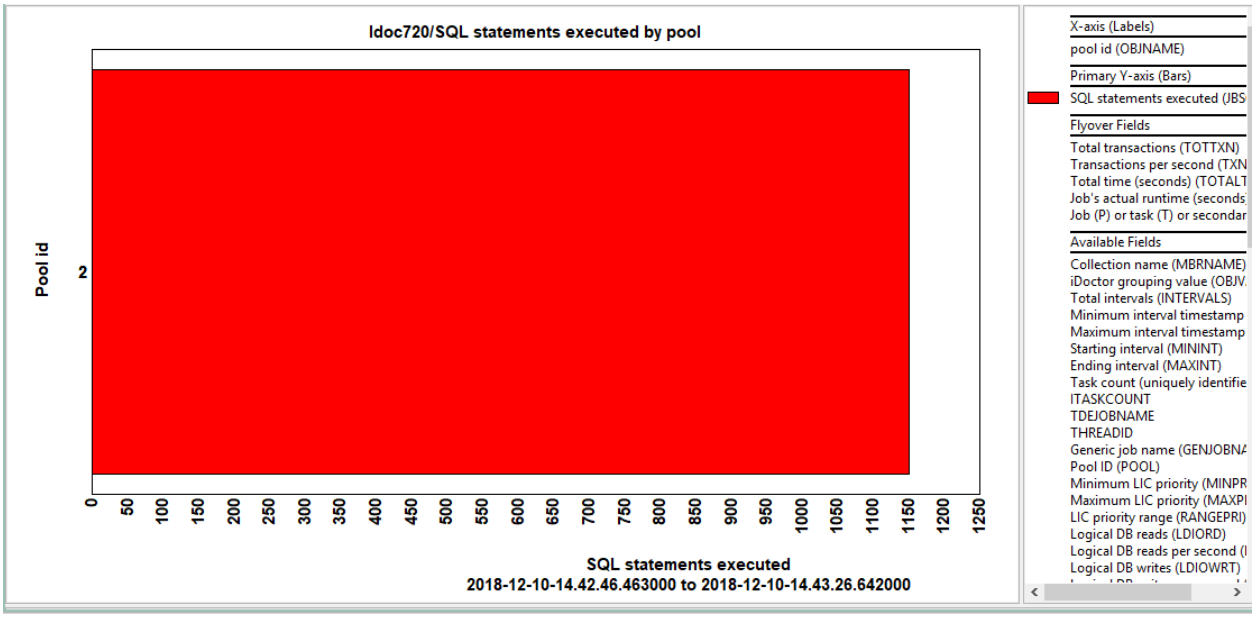
This graph ranks the SQL statements executed in the selected time period by current user profile.



SQL statements executed by current user

8.25.10.6 SQL statements executed by pool

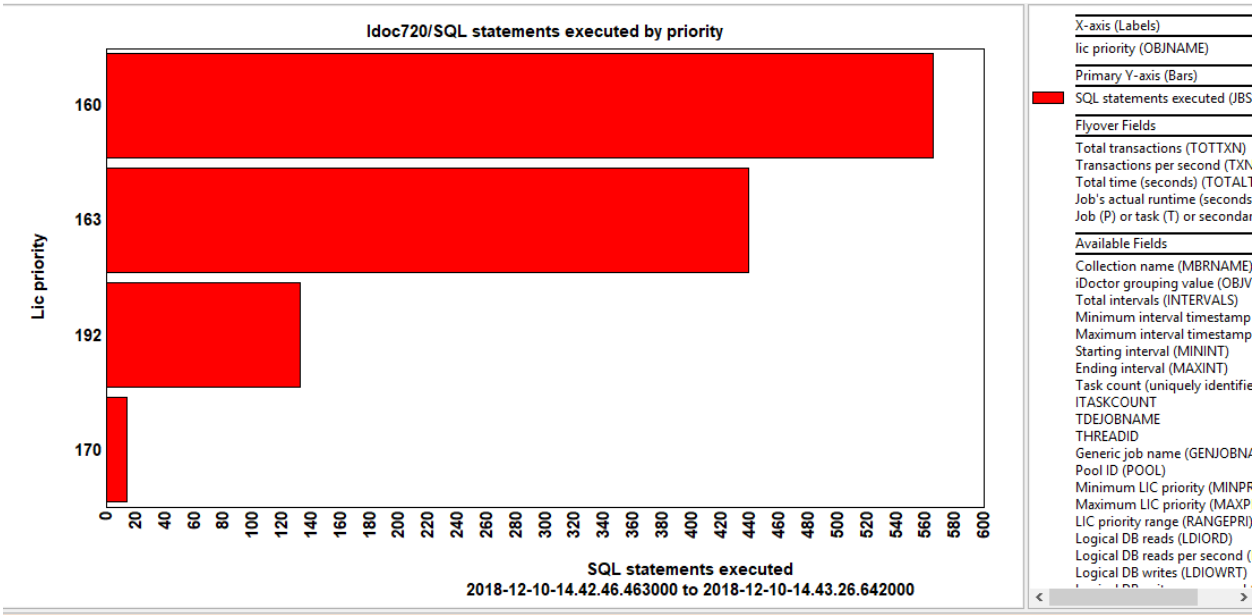
This graph ranks the SQL statements executed in the selected time period by memory pool.



SQL statements executed by pool

8.25.10.7 SQL statements executed by priority

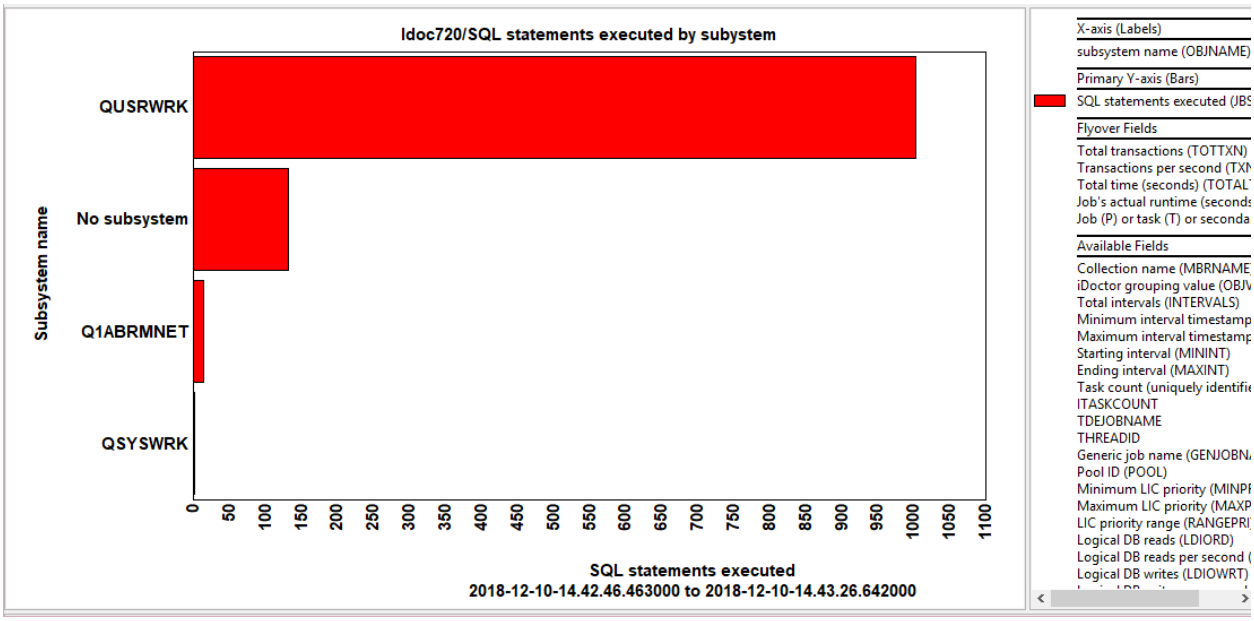
This graph ranks the SQL statements executed in the selected time period by LIC priority. XPF priority can be calculated by subtracting 140 for those values shown exceeding 140.



SQL statements executed by LIC priority

8.25.10.8 SQL statements executed by subsystem

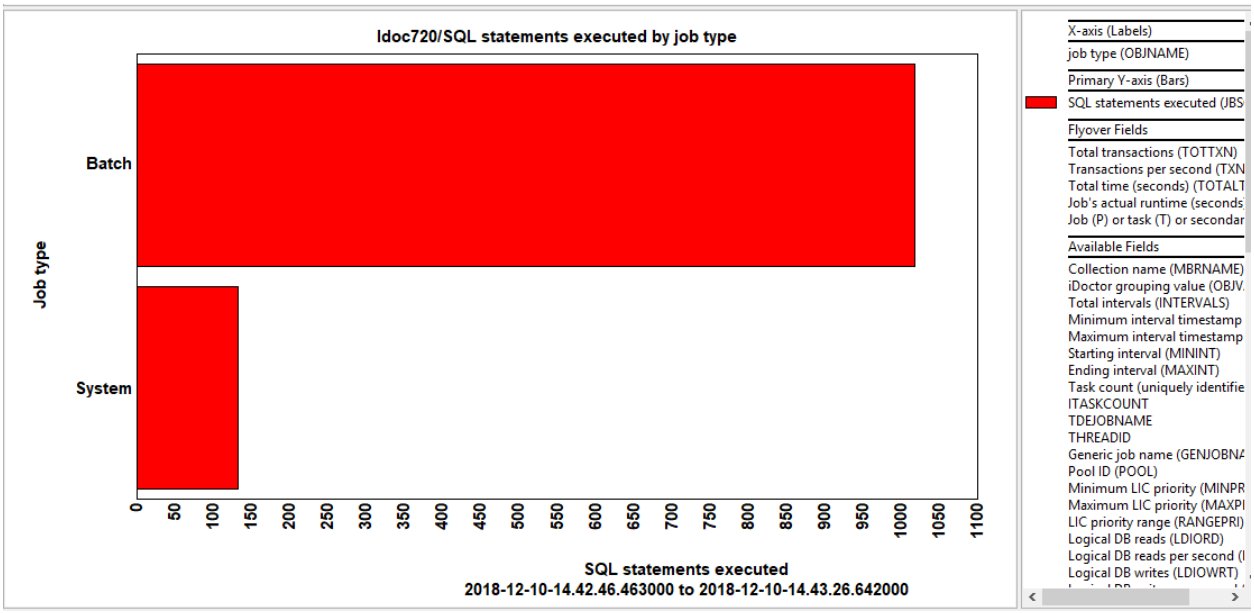
This graph ranks the SQL statements executed in the selected time period by subsystem. Job that had no subsystem listed are grouped into 1 bar called “No subsystem”.



SQL statements executed by subsystem

8.25.10.9 SQL statements executed by job type

This graph ranks the SQL statements executed in the selected time period by job type (such as Batch, interactive, autostart, etc.) Jobs that had no job type associated with it are grouped into 1 bar called “No job type”.

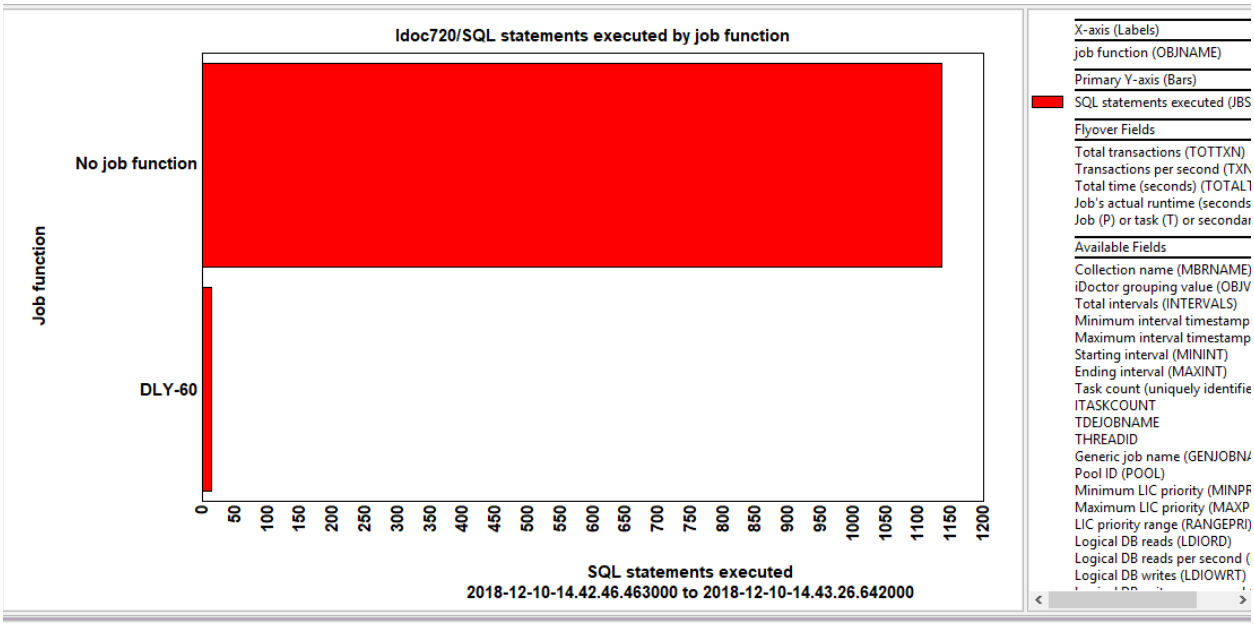


SQL statements executed by job type

8.25.10.10 SQL statements executed by job function

This graph ranks the SQL statements executed in the selected time period by job function. Job function is the same as what you would see normally on the WRKACTJOB command's Function column. Any wait bucket times that had no job function associated with it are grouped into 1 bar called "No job function".

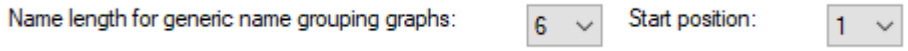
Note: This graph can be misleading in some situations because the current job function for each process is only captured at the end of each Job Watcher interval. The times given are not necessarily 100% from each function listed.

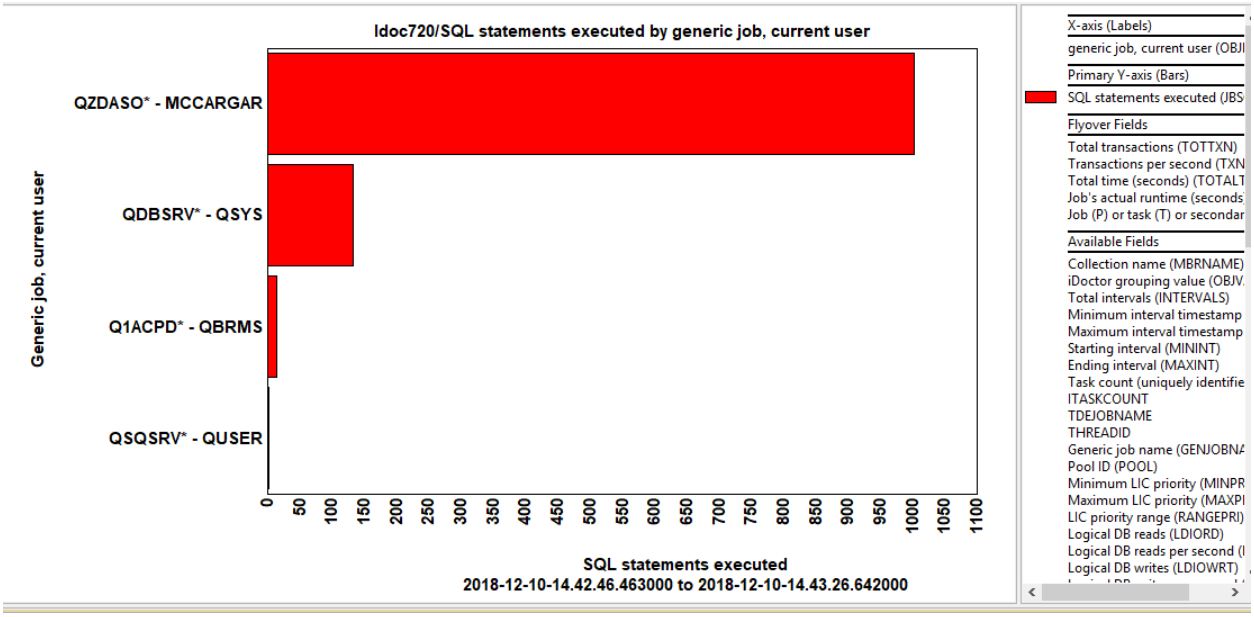


SQL statements executed by job function

8.25.10.11 SQL statements executed by generic job, current user

This graph ranks the SQL statements executed in the selected time period by generic job/current user profile combination. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

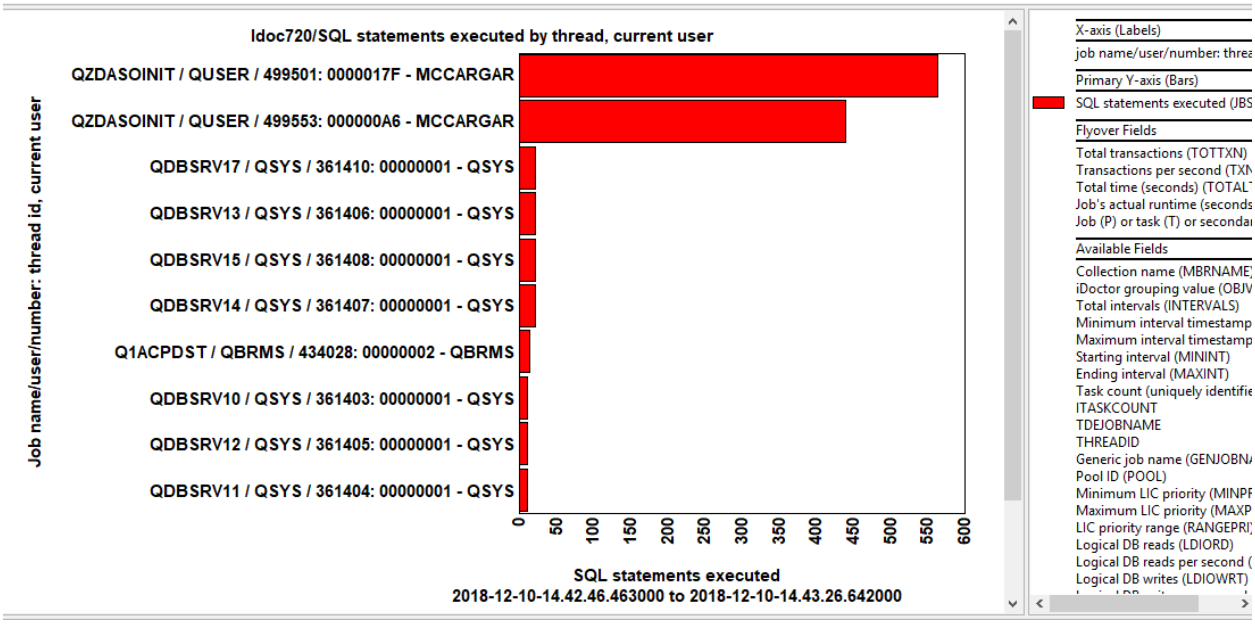




SQL statements executed by generic job, current user

8.25.10.12 SQL statements executed by thread, current user

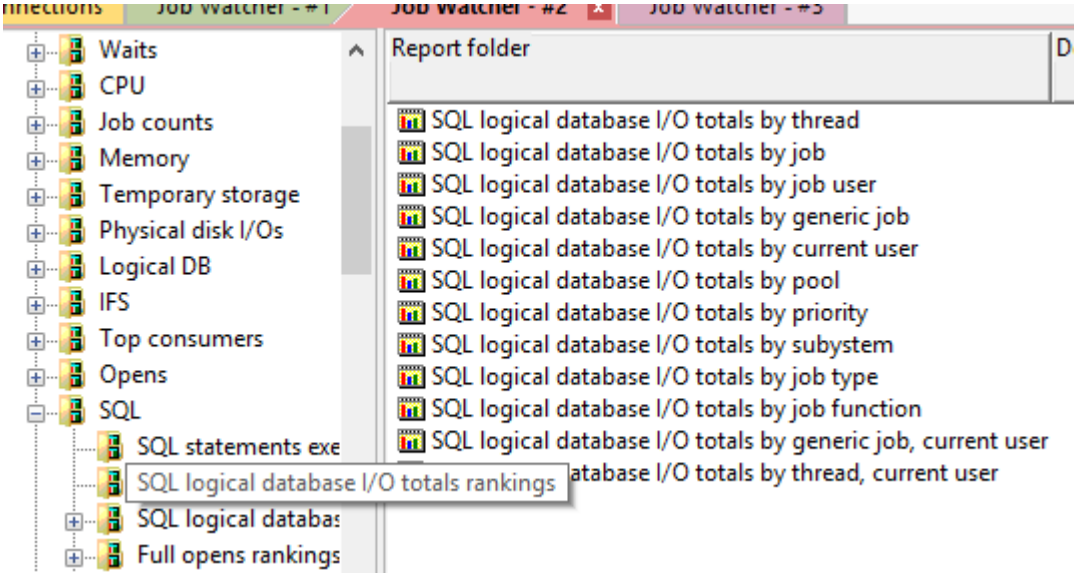
This graph ranks the SQL statements executed in the selected time period by thread/current user profile combination.



SQL statements executed by thread, current user

8.25.11 SQL logical database I/O rankings

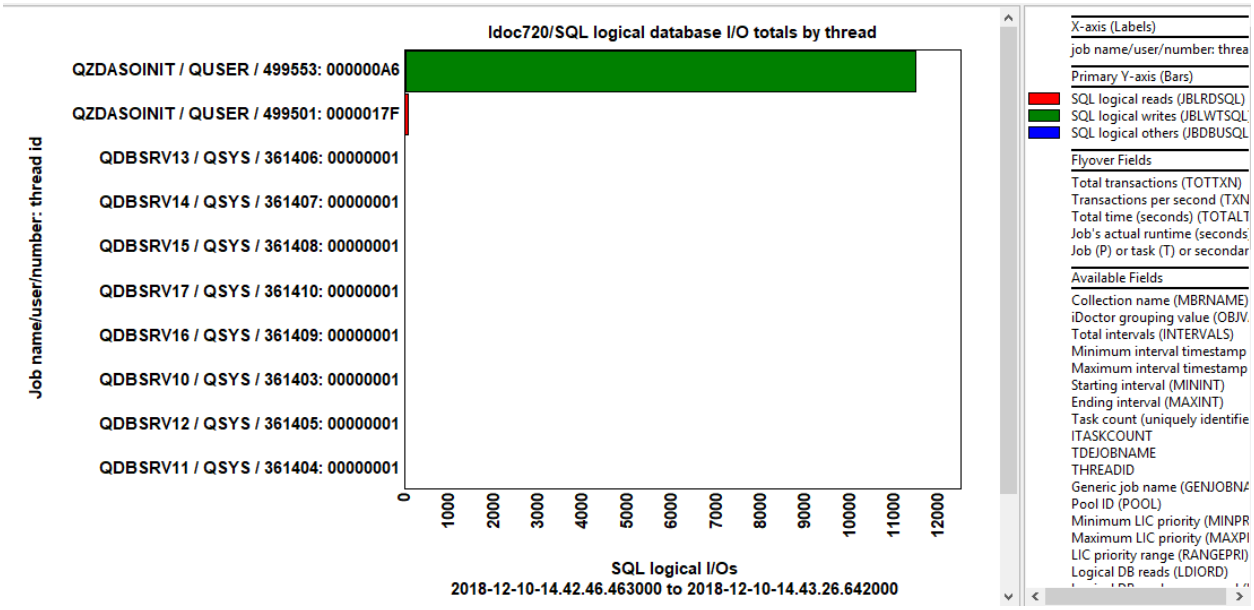
These graphs display the total SQL-related logical DB operations (for either totals or rates) ranked by various job groupings.



SQL -> SQL logical database I/O totals rankings

8.25.11.1 SQL logical database I/O totals by thread

This graph ranks the SQL logical database I/O operations by thread or system task.

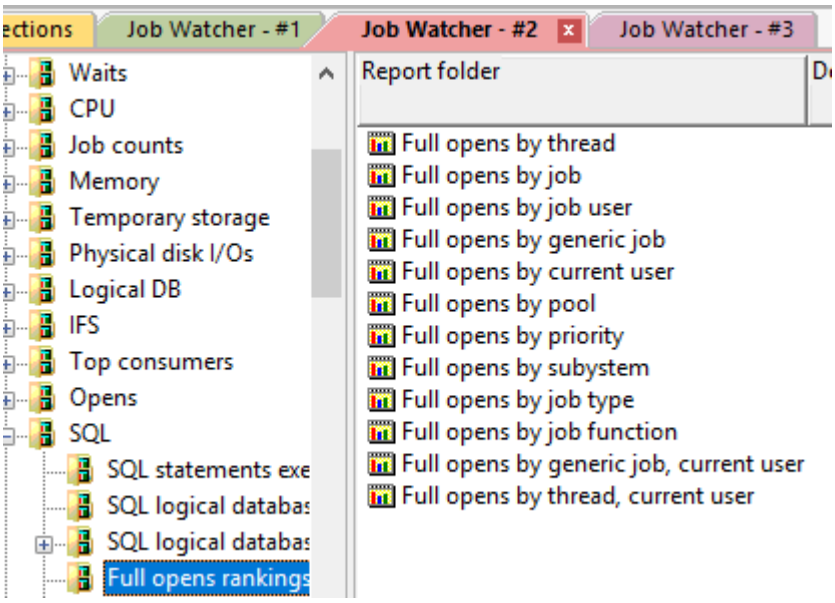


SQL logical database I/O totals by thread

The rest of these graphs are very similar to the SQL statements executed rankings. See that section for more examples.

8.25.12 Full opens rankings

These graphs display the full opens (native and SQL-driven) as either a total or a rate per second ranked by various job groupings.

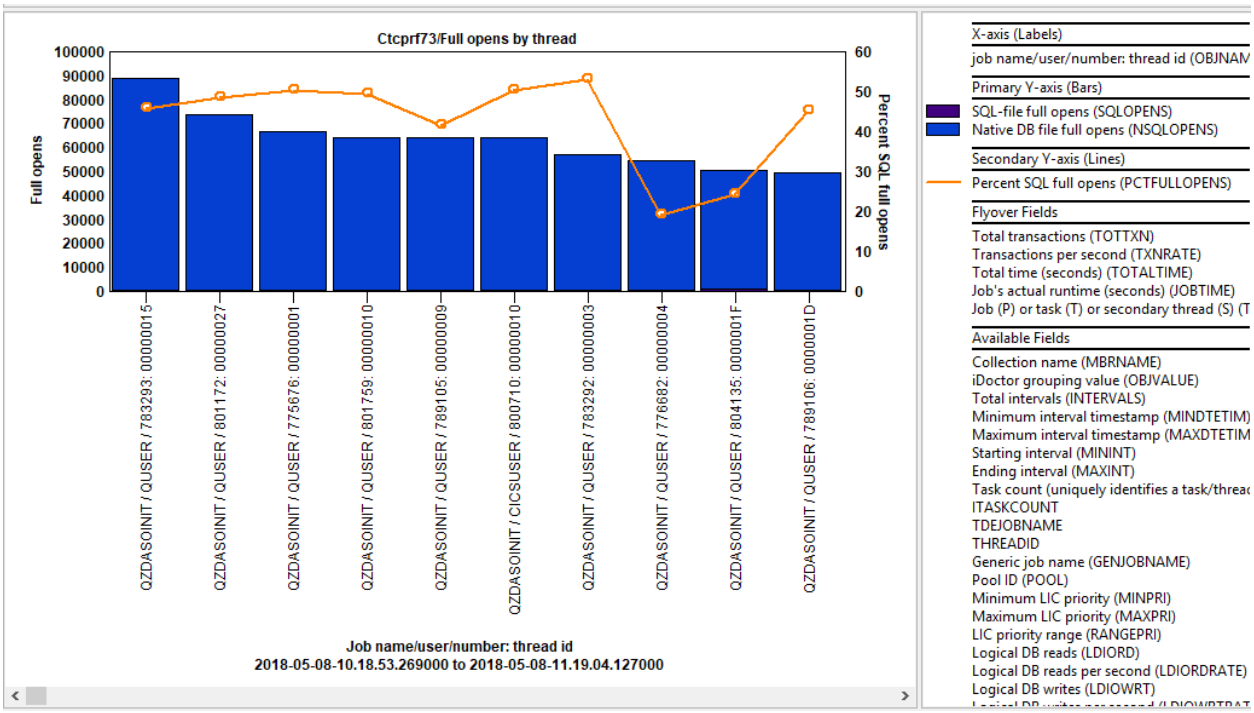


SQL -> Full opens rankings

8.25.12.1 Full opens by thread

This graph ranks the total full opens occurring for both native and SQL driven opens by thread or system task.

The percentage of total SQL causing opens is shown on the second Y-axis. Note that if no native opens occur in those intervals then the percentage will be 100% even if the number of SQL-driven full opens is very small.

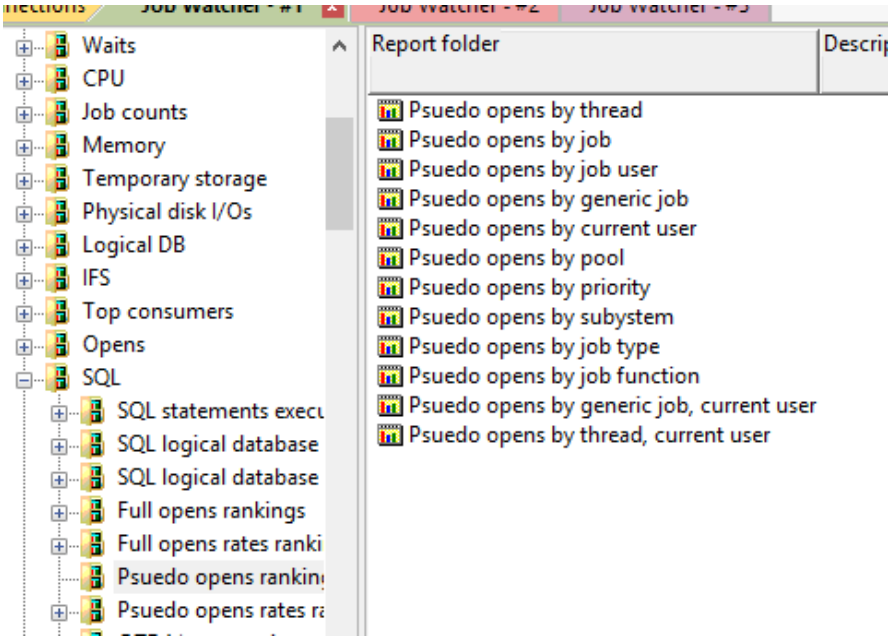


Full opens by thread

The rest of these graphs are very similar to the SQL statements executed rankings. See that section for more examples.

8.25.13 Psuedo opens rankings

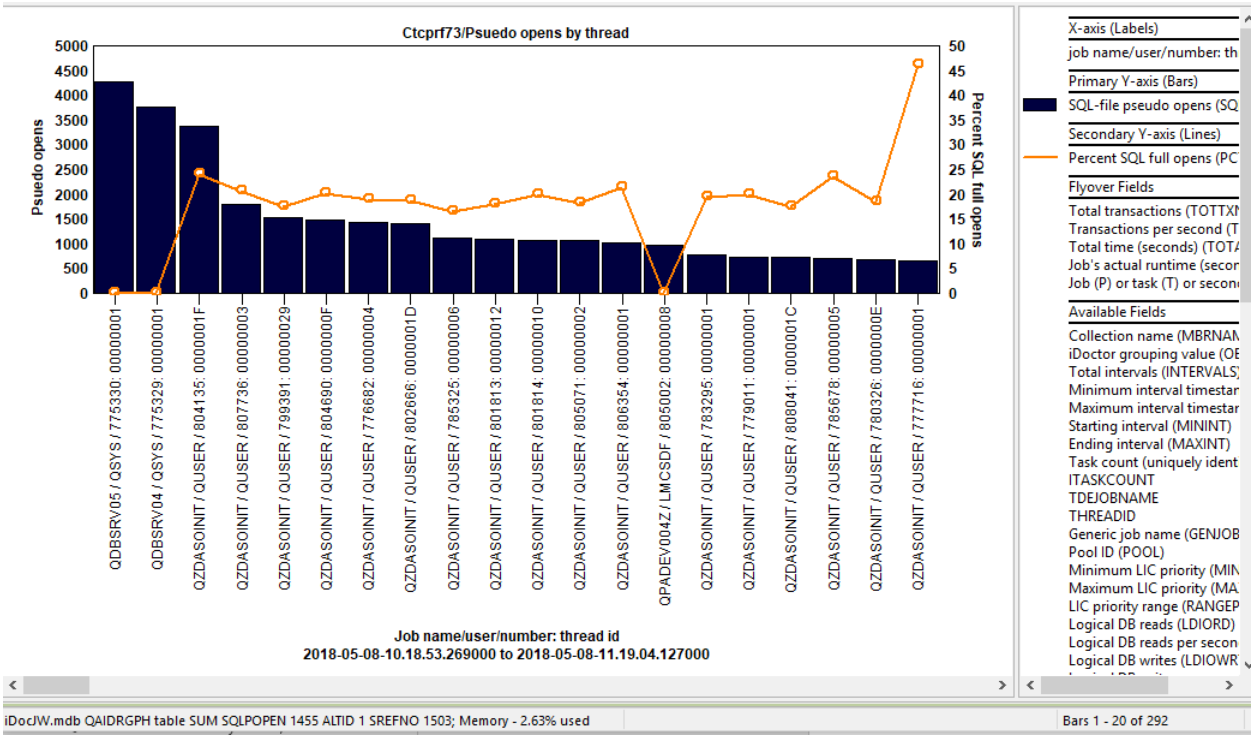
These graphs display the total SQL pseudo opens (for either totals or rates) ranked by various job groupings.



SQL -> Psuedo opens rankings

8.25.13.1 Psuedo opens by thread

This graph ranks the SQL pseudo opens by thread or system task.

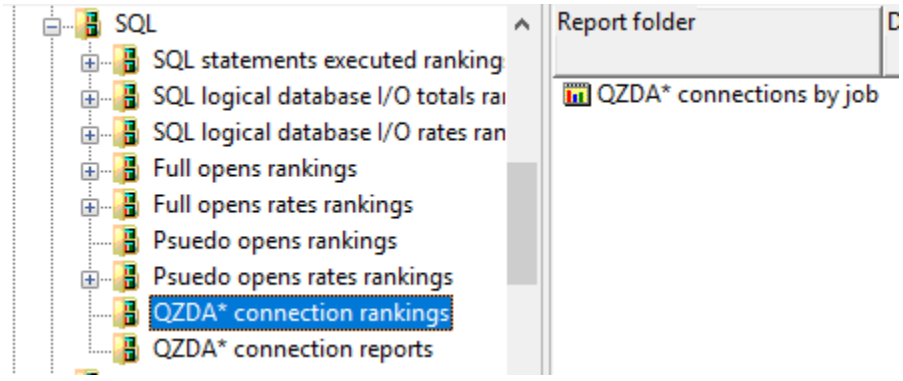


SQL pseudo opens by thread

The rest of these graphs are very similar to the SQL statements executed rankings. See that section for more examples.

8.25.14 QZDA* connections rankings

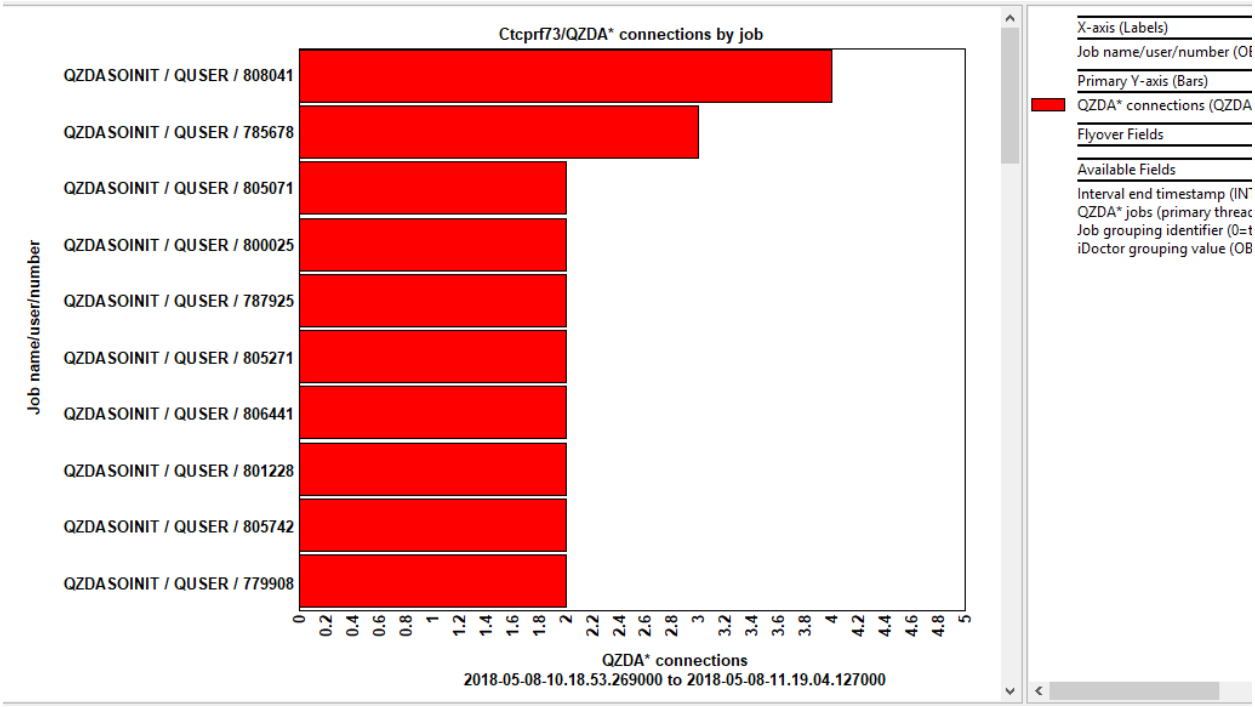
This subfolder contains a single graph that displays the total unique QZDA* connections by job (taskcount.)



QZDA connections rankings*

8.25.14.1 QZDA* connections by job

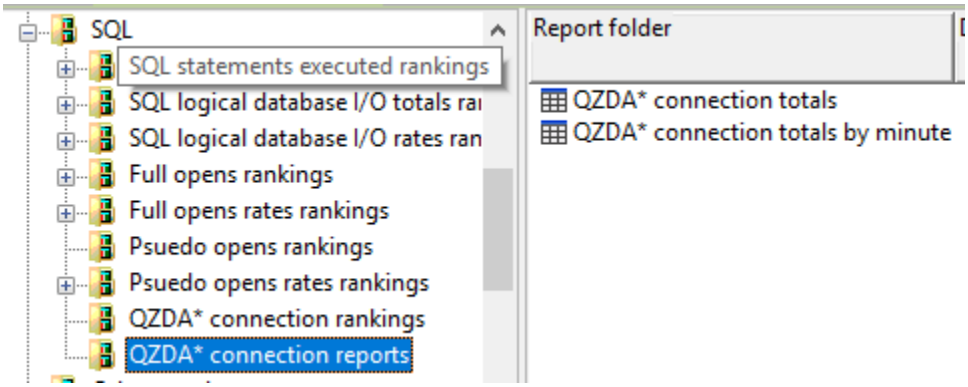
This graph shows the job names with the highest total unique QZDA connections (servicing potentially different users.)



QZDA connections by job*

8.25.15 QZDA* connection reports

This folder contains a couple of reports that count up the number of QZDA* connections in the collection.



QZDA connection reports*

8.25.15.1 QZDA* connection totals

This report lists the job name and taskcount that had the highest number of QZDA* connections/users served.

/@$CGIJW1/Q1281018... | /@$CGIJW1/Q128101852/... | /@$CGIJW1/Q128101852/... | /@$CGIJW1/Q128101852/... | /@$CGIJW1/Q128101852/...

TOTAL	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Job/task name (TDEJOBNAME)		Minimum interval timestamp (MINDTETIM)	Maximum interval timestamp (MAXDTETIM)
4	158,627	QZDASOINITQUSER	808041	2018-05-08-10.32.16.429000	2018-05-08-11.06.32.211000
3	31,119	QZDASOINITQUSER	785678	2018-05-08-10.27.05.584000	2018-05-08-10.33.56.756000
2	145,995	QZDASOINITQUSER	801228	2018-05-08-10.34.06.799000	2018-05-08-10.37.27.377000
2	21,837	QZDASOINITQUSER	779908	2018-05-08-10.57.40.938000	2018-05-08-11.09.52.703000
2	62,437	QZDASOINITQUSER	800025	2018-05-08-10.22.54.868000	2018-05-08-11.08.22.451000
2	153,809	QZDASOINITQUSER	805271	2018-05-08-11.03.31.746000	2018-05-08-11.09.42.676000
2	142,242	QZDASOINITCICSUSER	800713	2018-05-08-10.21.54.697000	2018-05-08-10.22.24.773000
2	34,818	QZDASOINITQUSER	787925	2018-05-08-11.16.13.745000	2018-05-08-11.17.33.894000
2	142,234	QZDASOINITCICSUSER	800710	2018-05-08-11.00.21.409000	2018-05-08-11.00.31.442000
2	155,748	QZDASOINITQUSER	806441	2018-05-08-10.34.06.793000	2018-05-08-11.03.21.711000
2	154,648	QZDASOINITQUSER	805742	2018-05-08-10.47.09.073000	2018-05-08-10.48.19.276000
2	153,530	QZDASOINITQUSER	805071	2018-05-08-10.40.07.852000	2018-05-08-10.59.01.173000
2	30,547	QZDASOINITQUSER	785325	2018-05-08-10.55.30.524000	2018-05-08-10.57.30.901000
1	153,231	QZDASOINITQUSER	804859	2018-05-08-10.24.55.239000	2018-05-08-10.24.55.239000
1	159,330	QZDASOINITQUSER	808469	2018-05-08-10.33.16.622000	2018-05-08-10.33.16.622000
1	155,478	QZDASOINITQUSER	806260	2018-05-08-10.54.10.300000	2018-05-08-10.54.10.300000

QZDA connection totals*

8.25.15.2 QZDA* connection totals by minute

This report lists the number of QZDA* connections/users served and number of jobs per minute of the collection.

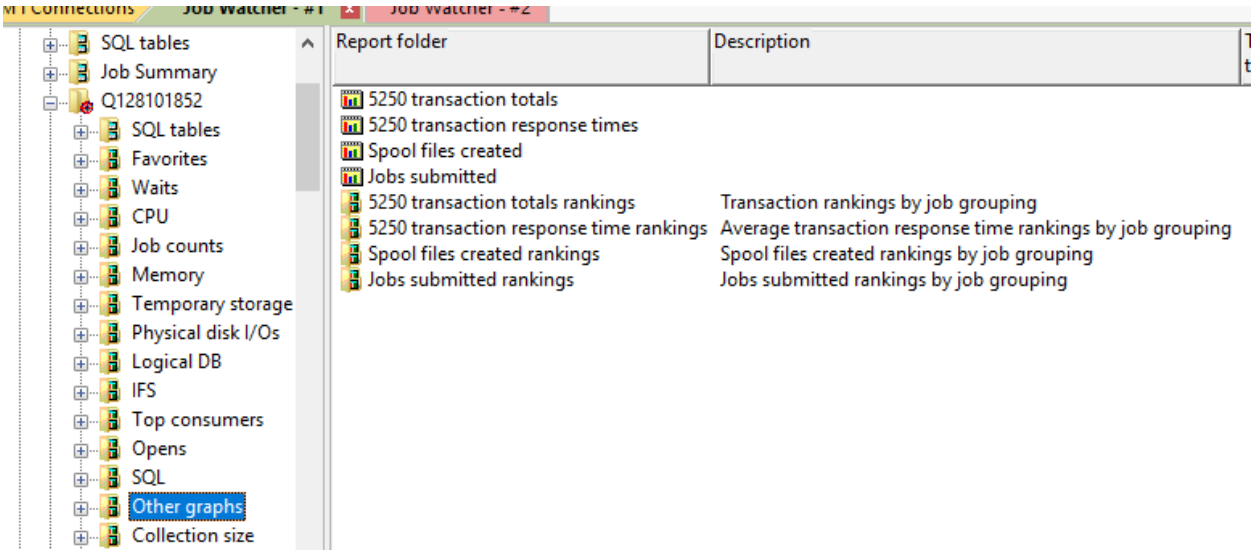
QZDA* connections (QZDA_CON...	HOUR_	MINUTE_	QZDA* jobs (primary threads) (QZDA_JO...
1	10	19	1
3	10	20	3
1	10	21	1
3	10	22	3
1	10	23	1
1	10	24	1
1	10	27	1
2	10	31	2
1	10	32	1
2	10	33	2
3	10	34	3
2	10	37	2
4	10	40	4
1	10	43	1
1	10	45	1

QZDA connections totals by minute*

8.26 Other graphs

This folder contains some additional graphs covering other statistics not found in the previous graphs.

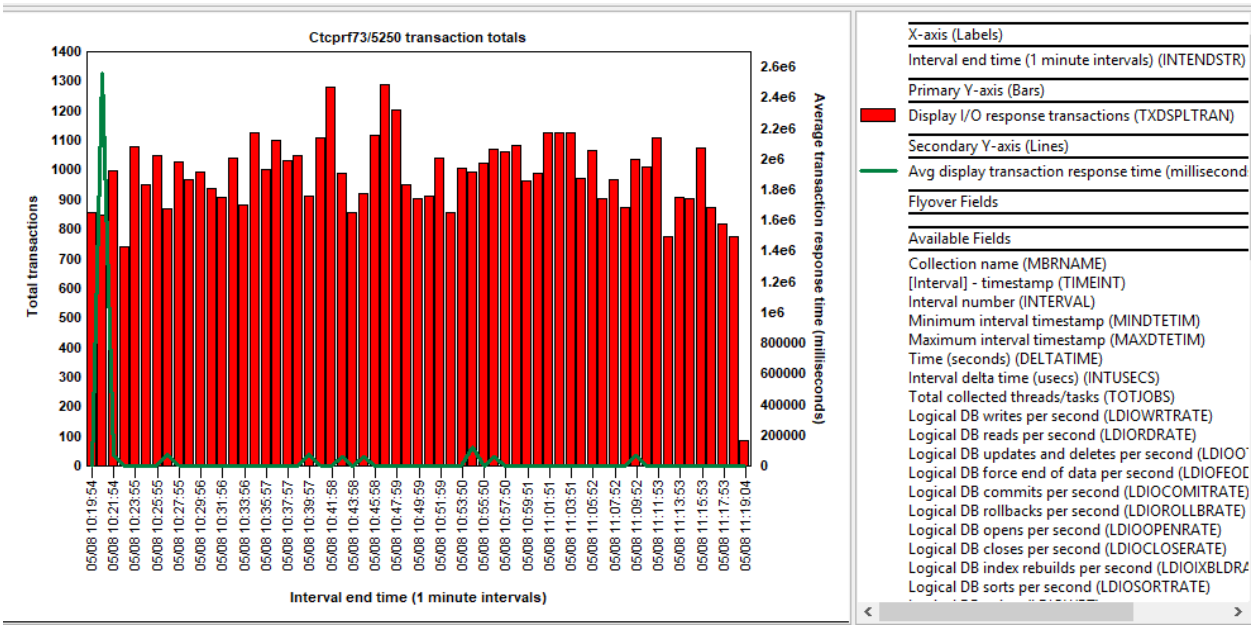
These statistics include: Spool file creations, jobs submitted and transactions.



Other graphs folder

8.26.1 5250 transaction totals

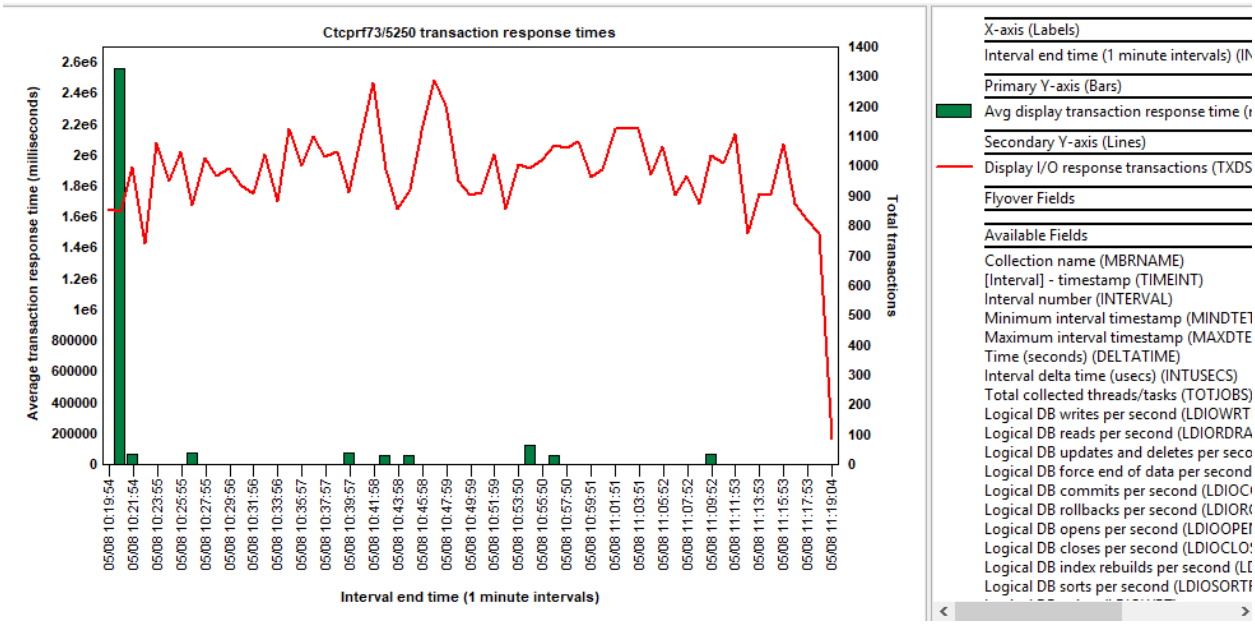
This graph displays the total 5250 display transactions that occurred during the collection along with their average response times on the Y2-axis. These are recorded for interactive jobs only. The transaction starts on detection of enter from the workstation; the transaction ends when the keyboard is unlocked.



5250 transaction totals

8.26.2 5250 transaction response times

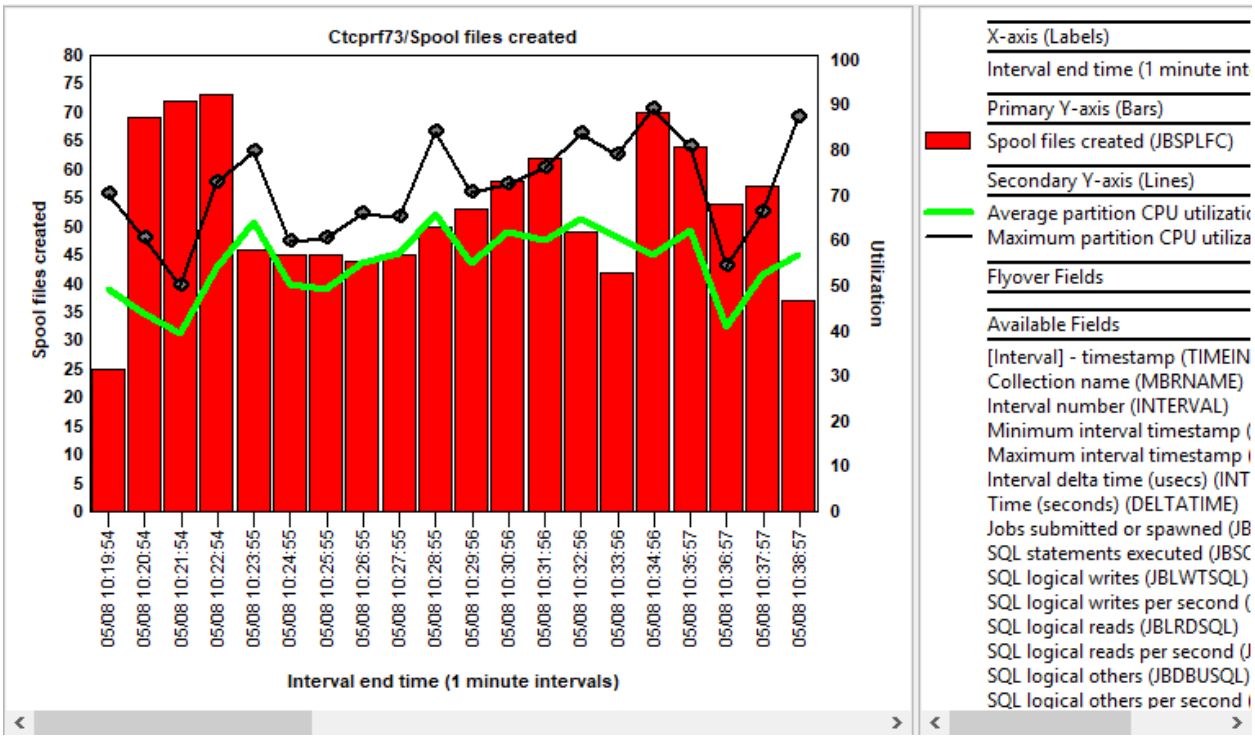
This graph displays the total 5250 display transactions average response times that occurred during the collection along with the total transactions on the Y2-axis. These are recorded for interactive jobs only. The transaction starts on detection of enter from the workstation; the transaction ends when the keyboard is unlocked.



5250 transaction response times

8.26.3 Spool files created

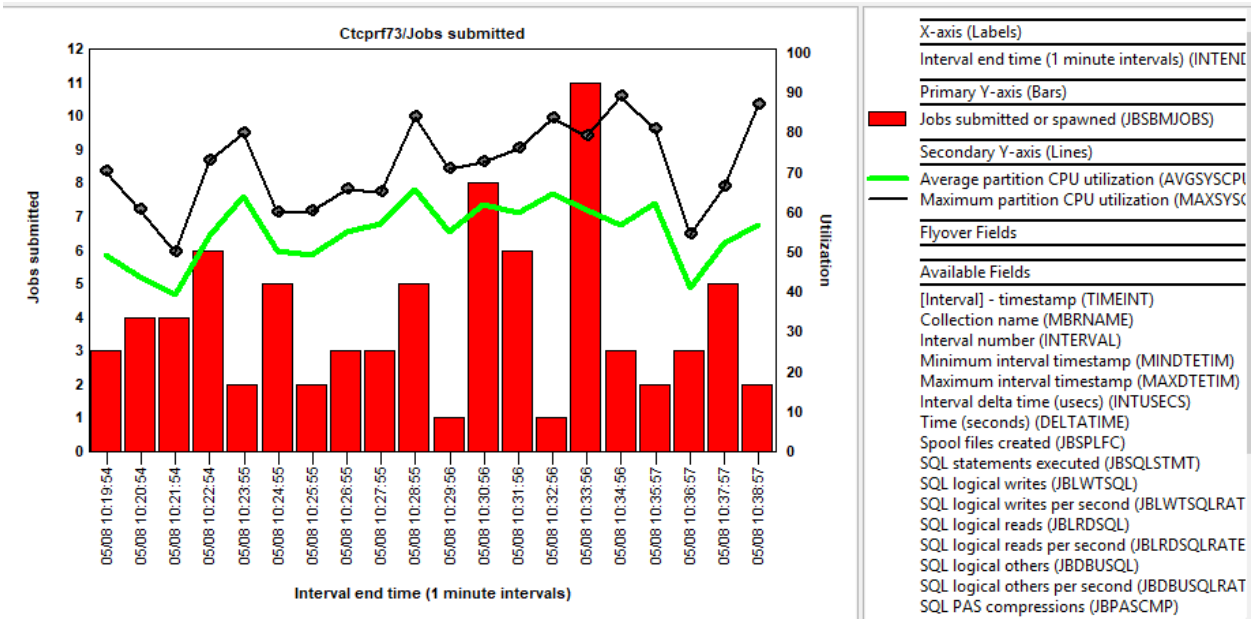
This graph displays the number of spool files created over time during the collection. The Y2-axis contains CPU utilization metrics.



Spool files created

8.26.4 Jobs submitted

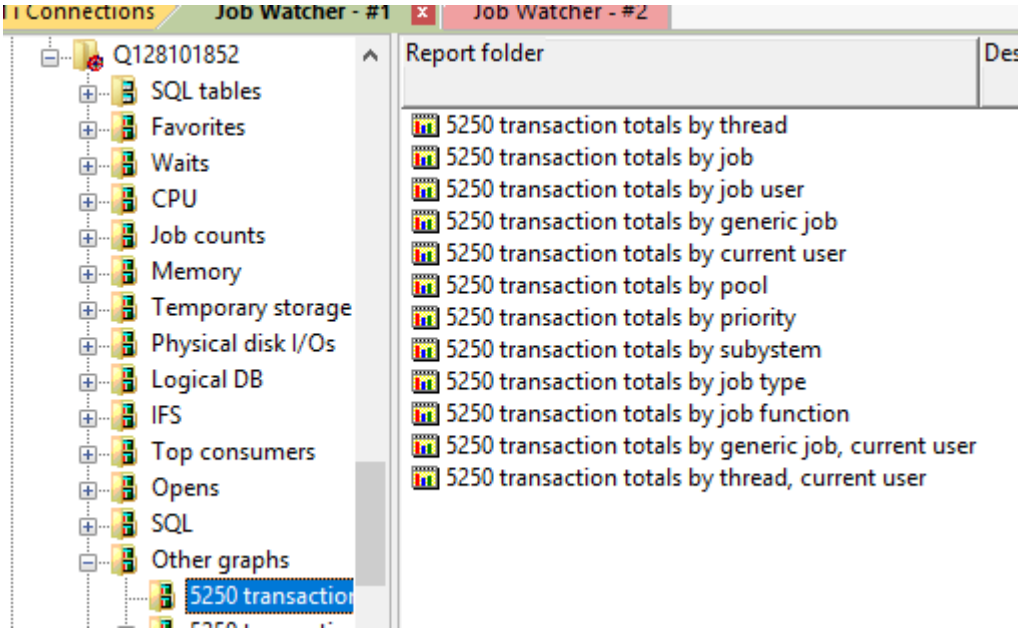
This graph displays the number of submitted jobs over time during the collection. The Y2-axis contains CPU utilization metrics.



Jobs submitted

8.26.5 5250 transaction totals rankings

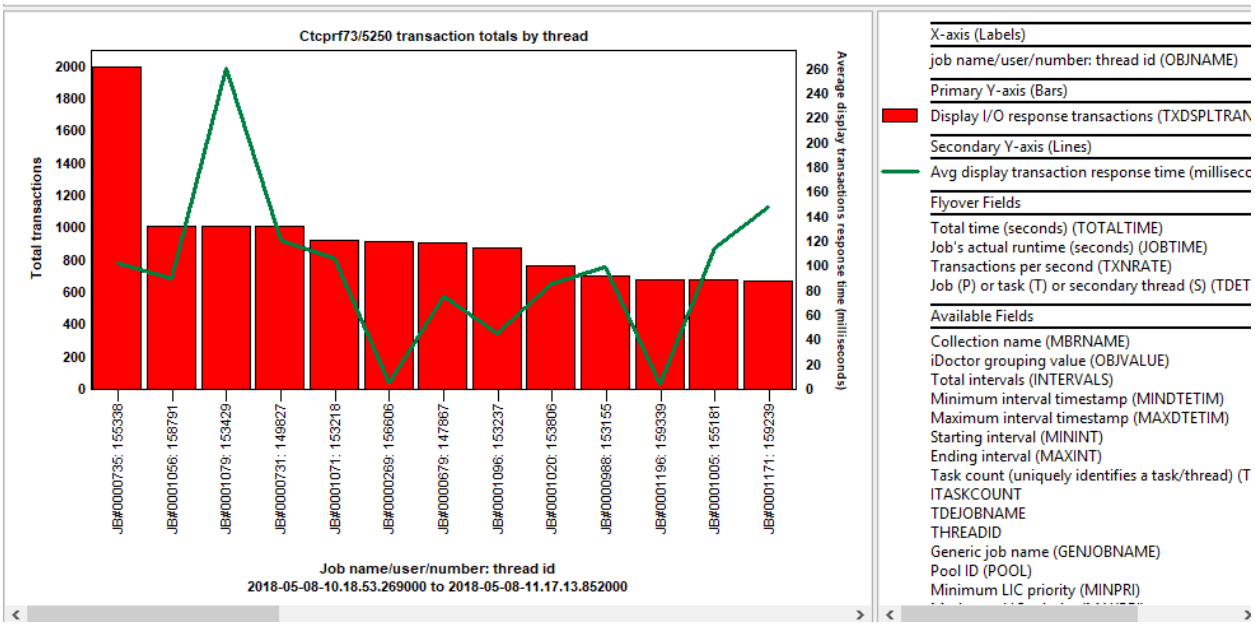
This folder contains a set of ranking graphs for 5250 display transactions totals. These allow you to rank the number of transactions that occurred by several different types of job groupings.



Other graphs -> 5250 transaction totals rankings

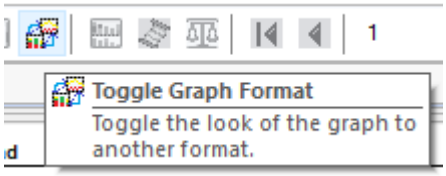
8.26.5.1 5250 transactions totals by thread

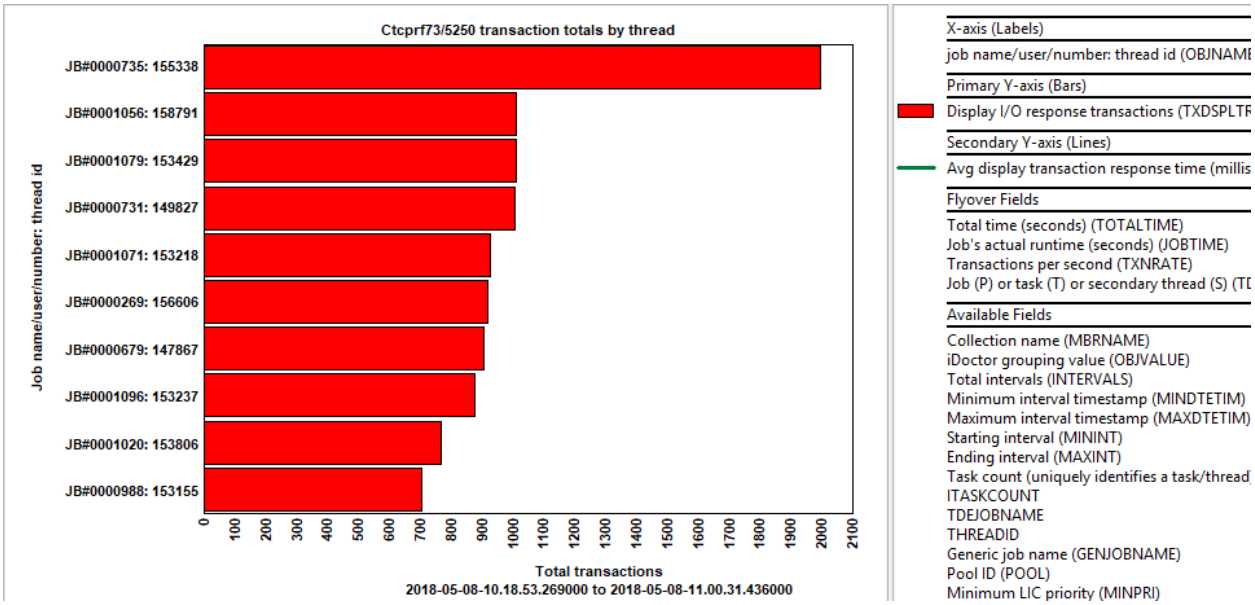
This graph ranks the number of 5250 display transactions by thread or system task.



5250 transaction totals by thread (vertical bars)

Tip: If you don't care to see the average response times on the Y2-axis and wish to read the job names horizontally then press the toggle graph format button on the Data Viewer toolbar to convert it to horizontal bars.

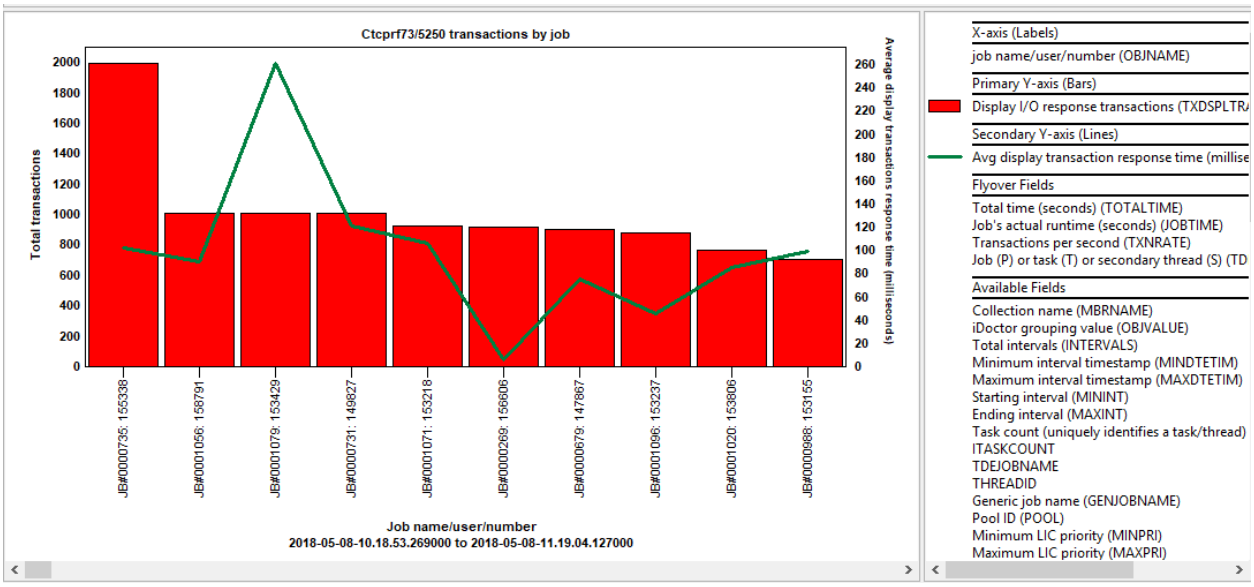




5250 transactions totals by thread

8.26.5.2 5250 transaction totals by job

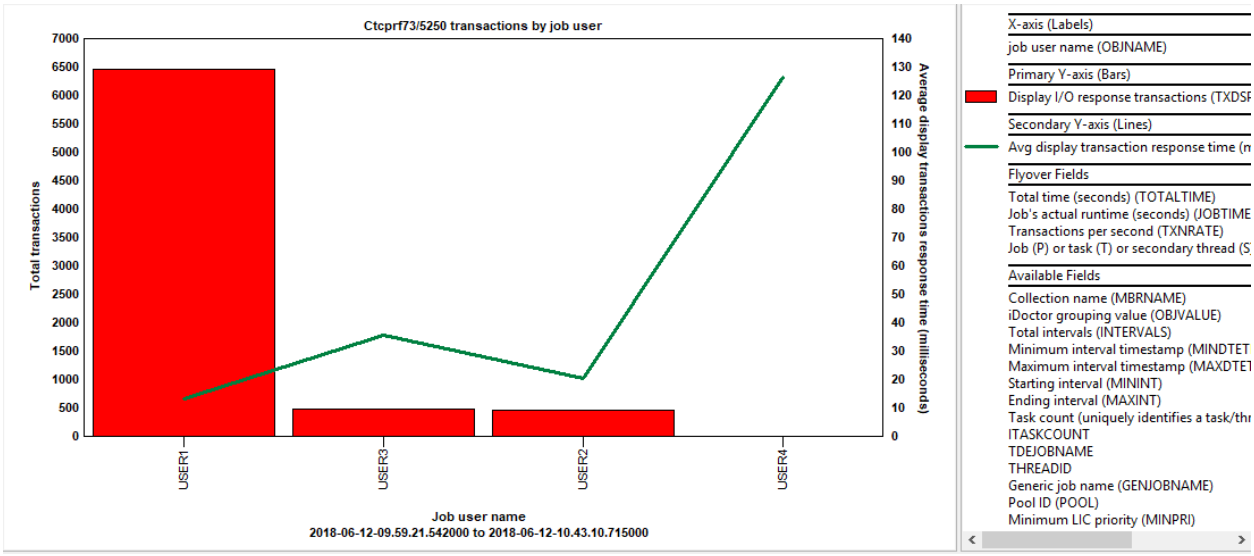
This graph ranks the number of 5250 display transactions by primary thread.



5250 transaction totals by job

8.26.5.3 5250 transaction totals by job user

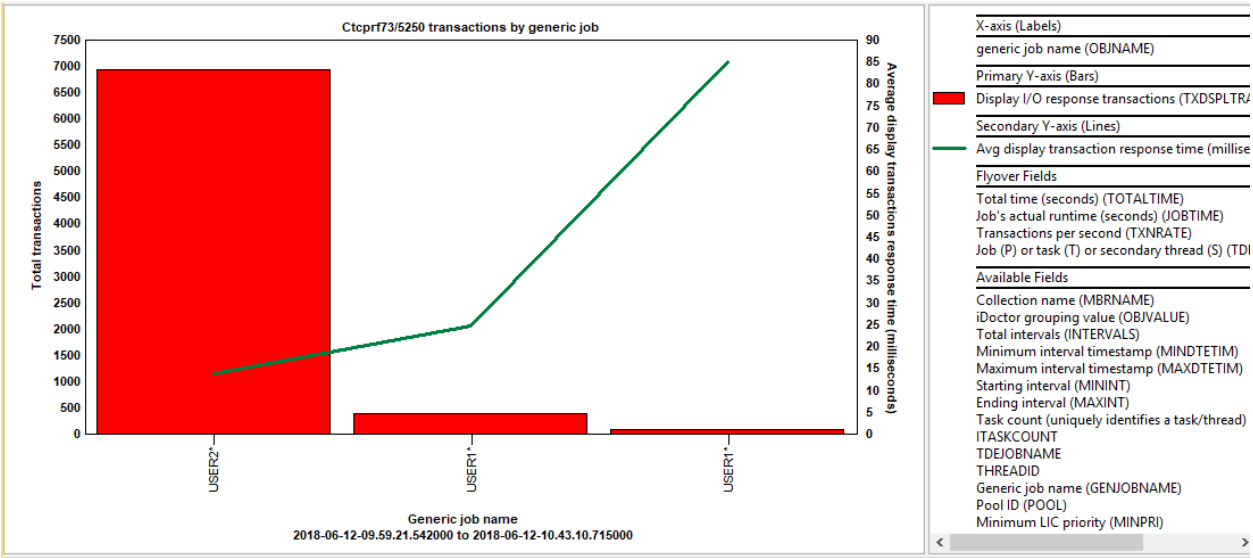
This graph ranks the number of 5250 display transactions by job user.



5250 transaction totals by job user

8.26.5.4 5250 transaction totals by generic job

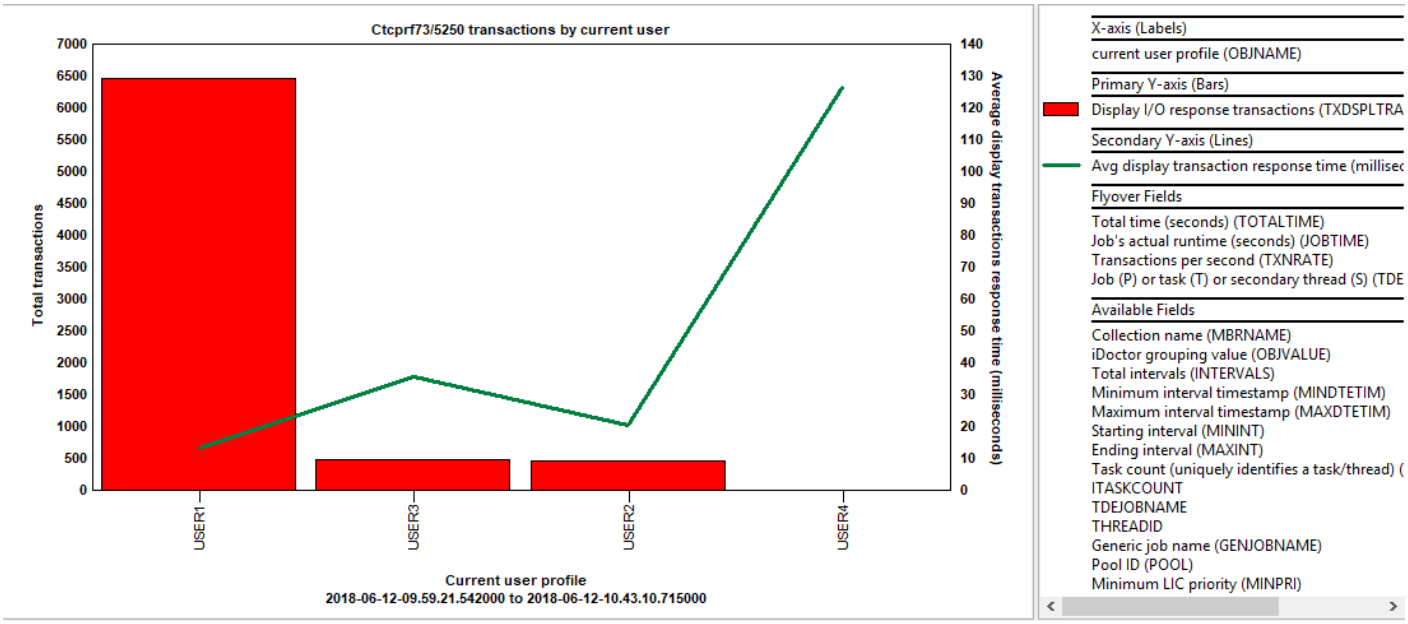
This graph ranks the number of 5250 display transactions by generic job.



5250 transaction totals by generic job

8.26.5.5 5250 transaction totals by current user

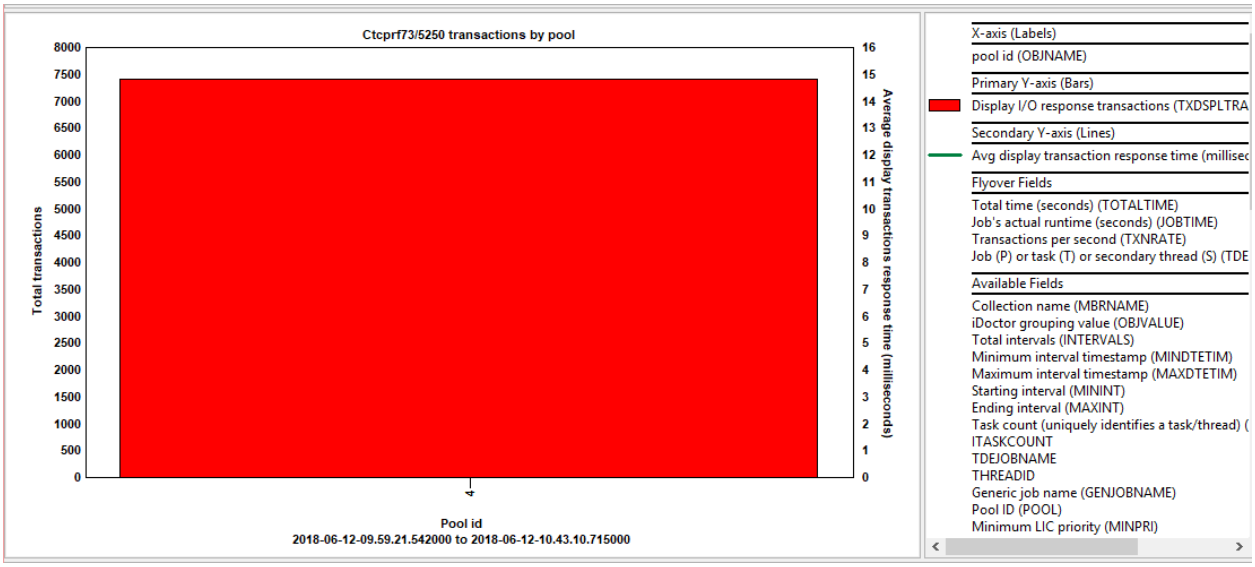
This graph ranks the number of 5250 display transactions by current user profile.



5250 transaction totals by current user

8.26.5.6 5250 transaction totals by pool

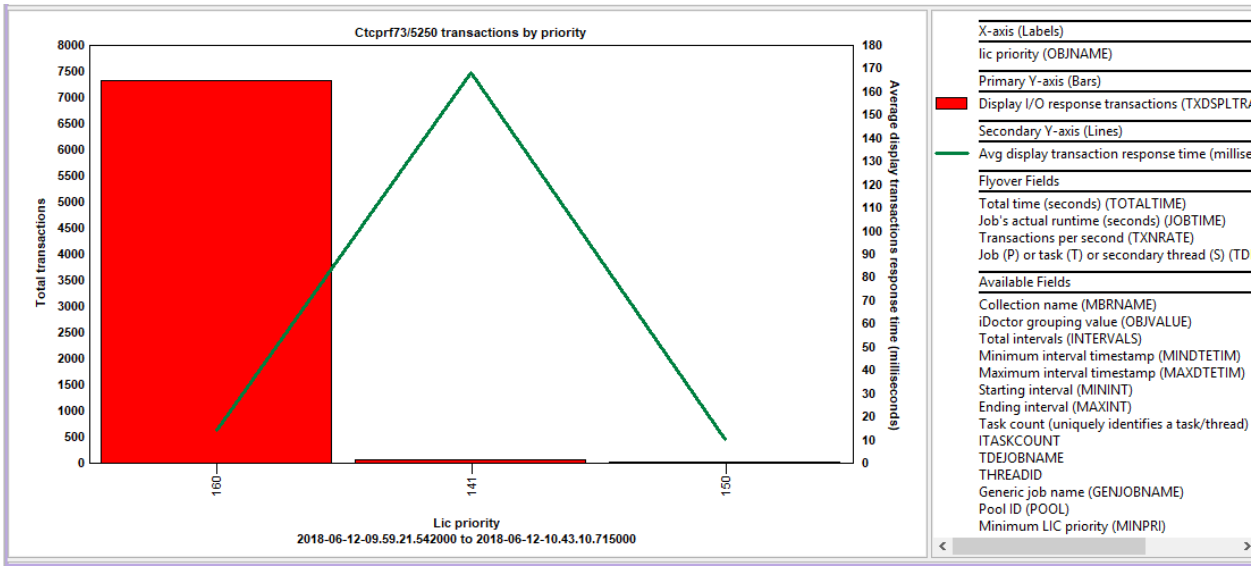
This graph ranks the number of 5250 display transactions by memory pool.



5250 transaction totals by pool

8.26.5.7 5250 transaction totals by priority

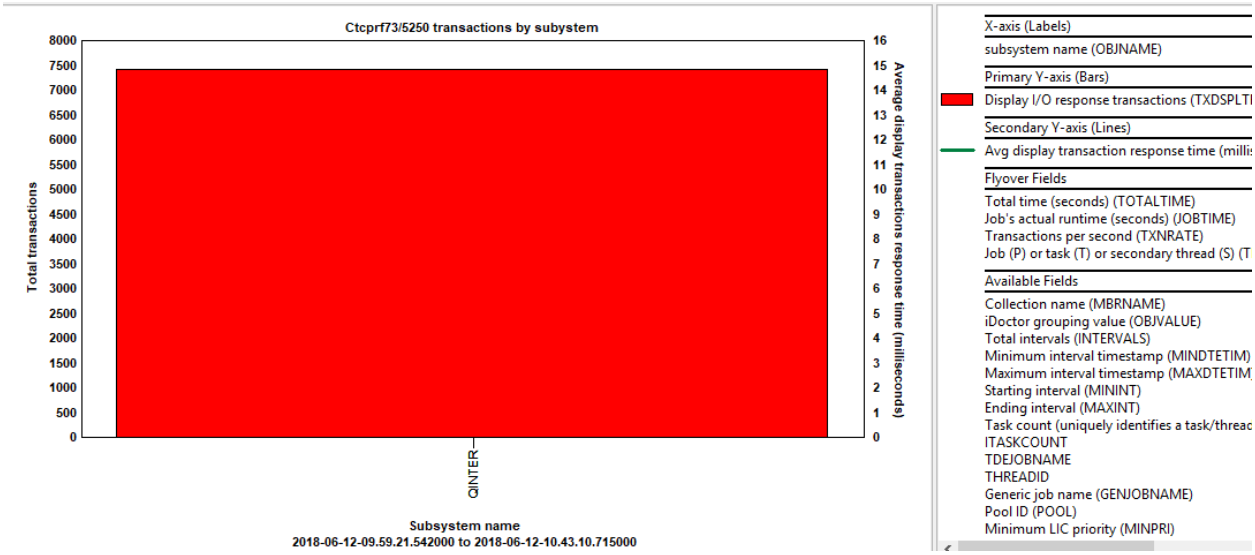
This graph ranks the number of 5250 display transactions by LIC priority.



5250 transaction totals by priority

8.26.5.8 5250 transaction totals by subsystem

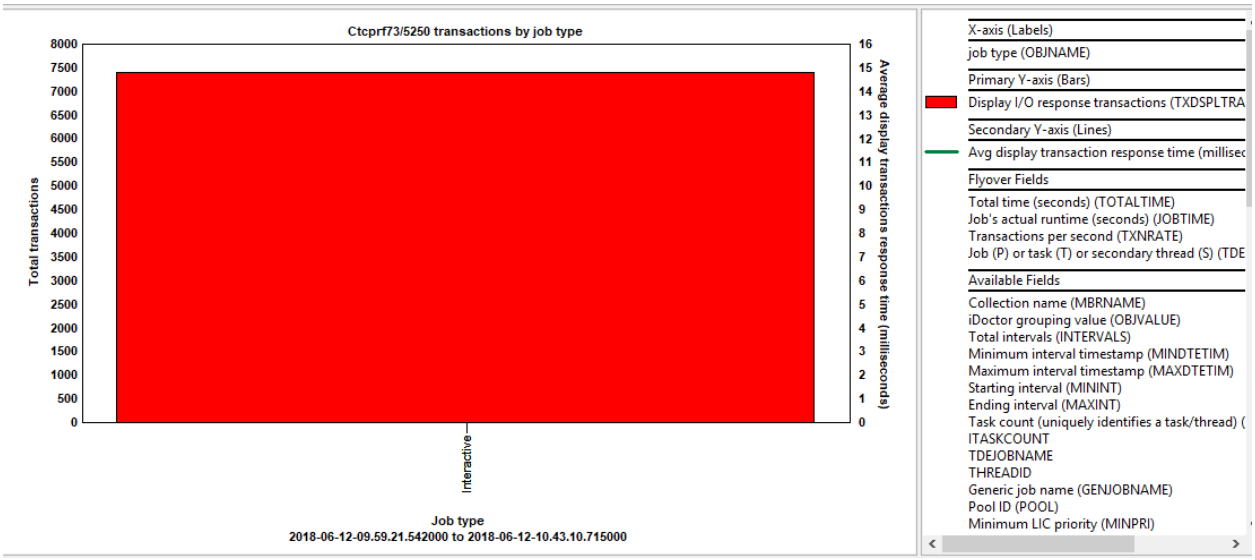
This graph ranks the number of 5250 display transactions by subsystem.



5250 transaction totals by subsystem

8.26.5.9 5250 transaction totals by job type

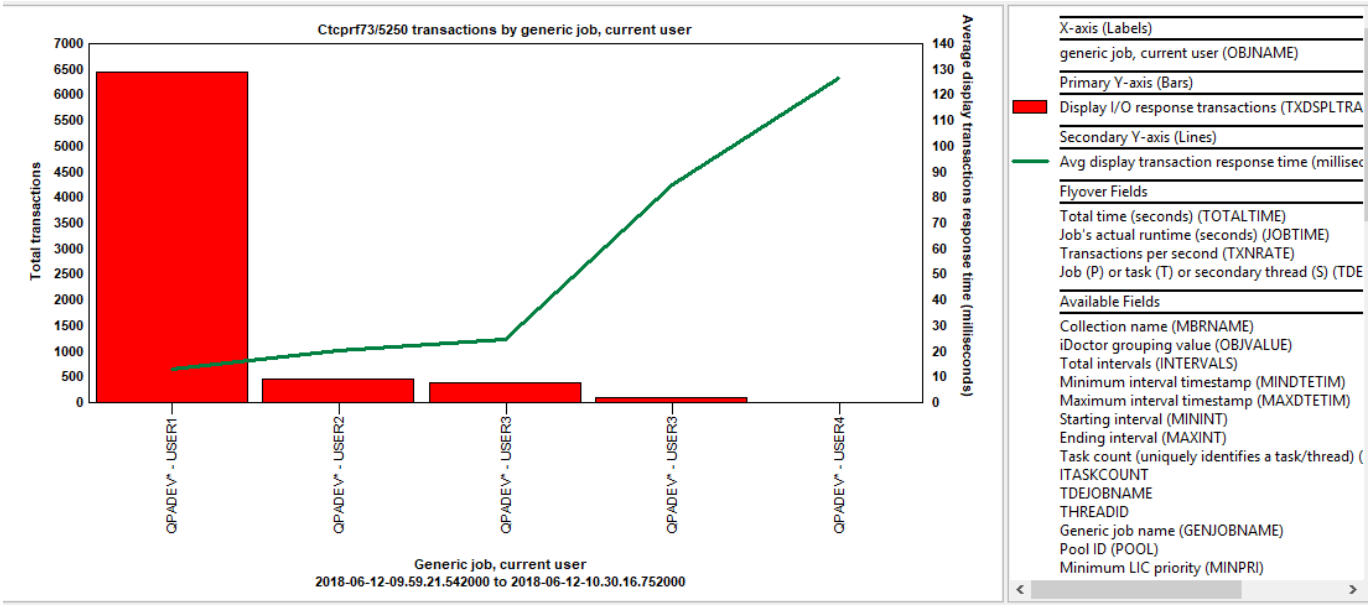
This graph ranks the number of 5250 display transactions by job type. Display transactions are only for interactive jobs so this should always contain only “Interactive” in the graph.



5250 transaction totals by job type

8.26.5.10 5250 transaction totals by generic job, current user

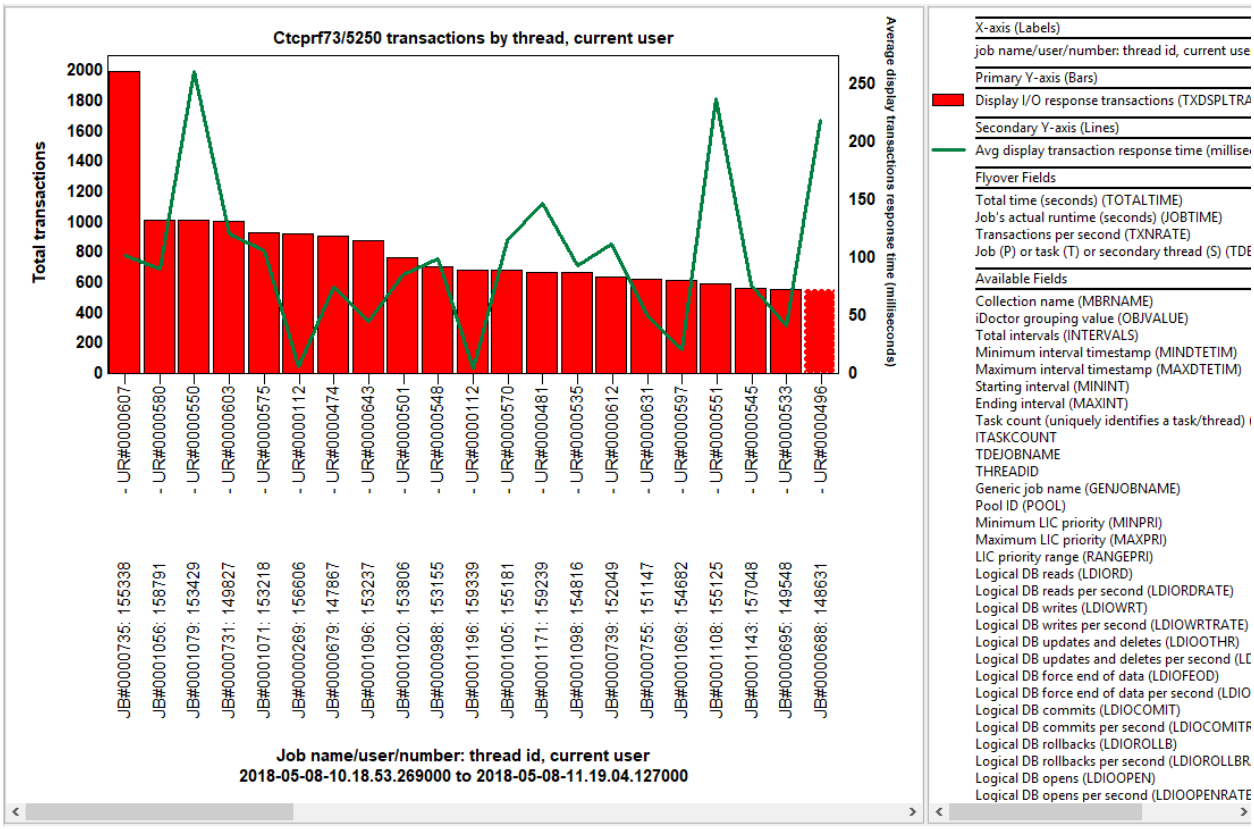
This graph ranks the number of 5250 display transactions by generic job, current user.



5250 transaction totals by generic job, current user

8.26.5.11 5250 transaction totals by thread, current user

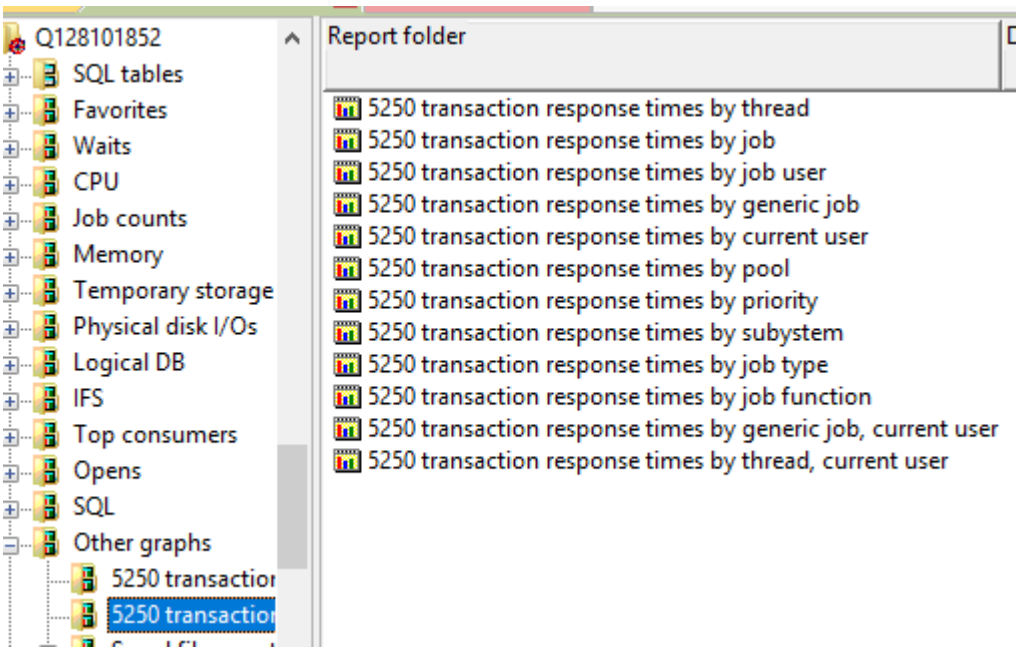
This graph ranks the number of 5250 display transactions by thread and current user combination.



5250 transaction totals by thread, current user

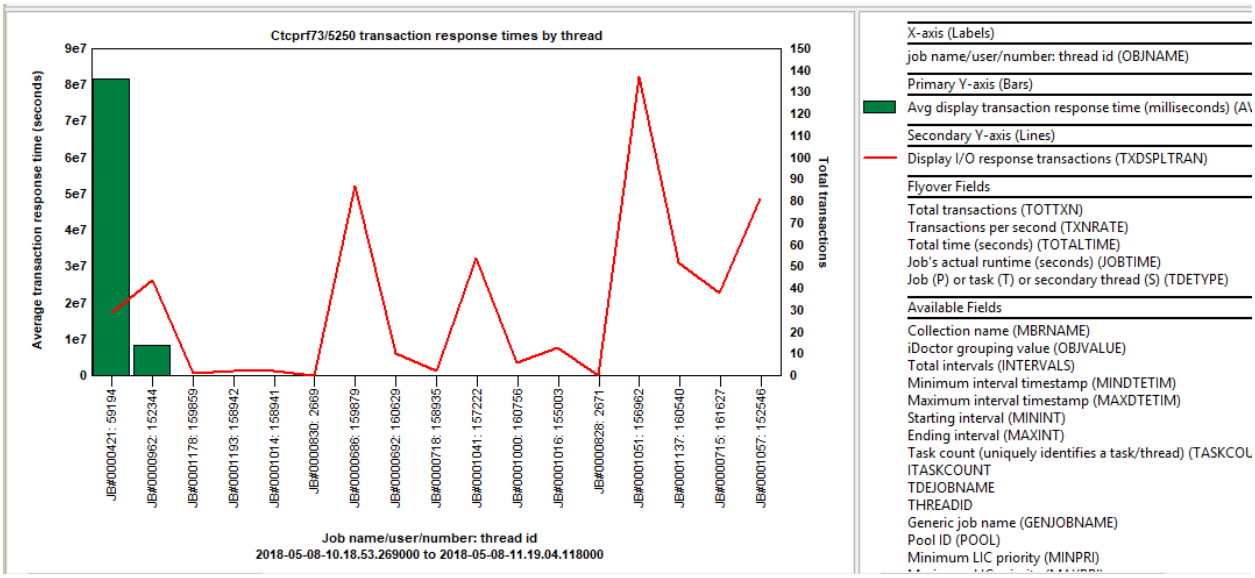
8.26.6 5250 transaction response times rankings

This folder contains a set of ranking graphs for 5250 displays transactions average response times. These allow you to rank the display transaction average response times that occurred by several different types of job groupings.



Other graphs -> 5250 transaction response time rankings

An example is shown below:

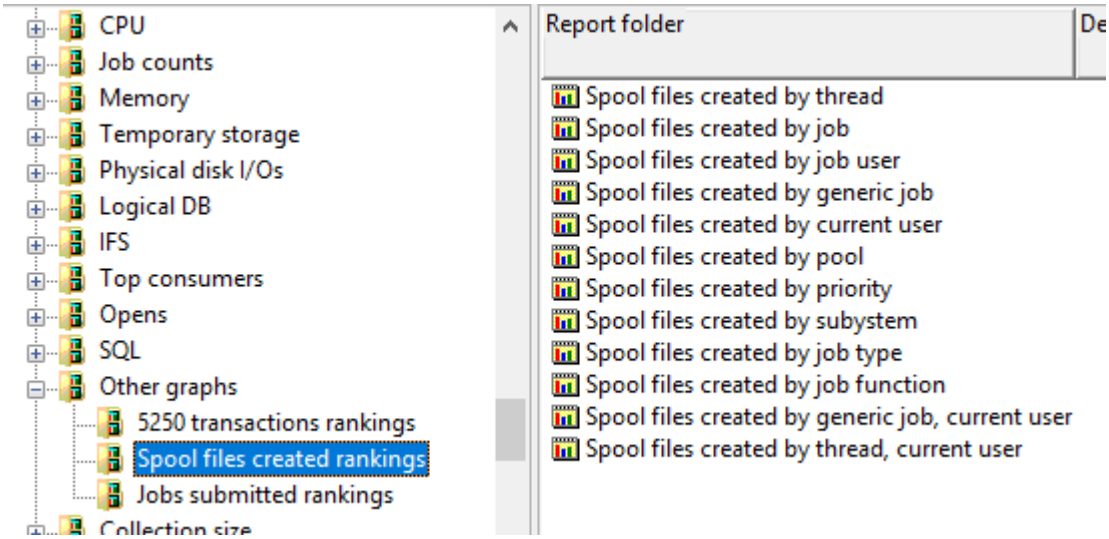


5250 transaction response times by thread

For more examples of the various types of job groupings, see the 5250 transactions rankings section.

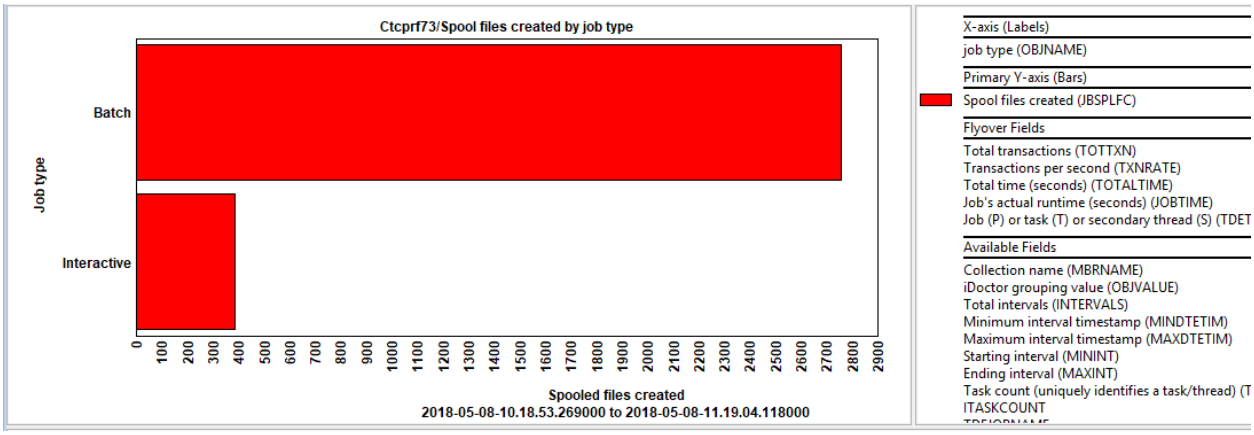
8.26.7 Spool files created rankings

These graphs rank the number of spool files created using various job groupings.



Spool files created rankings

An example is shown below:

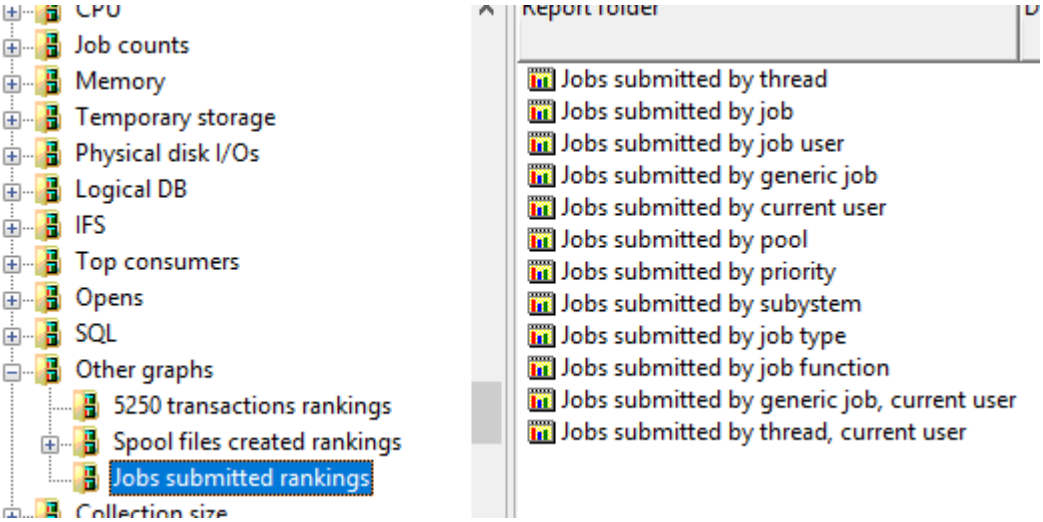


Spool files created by job type

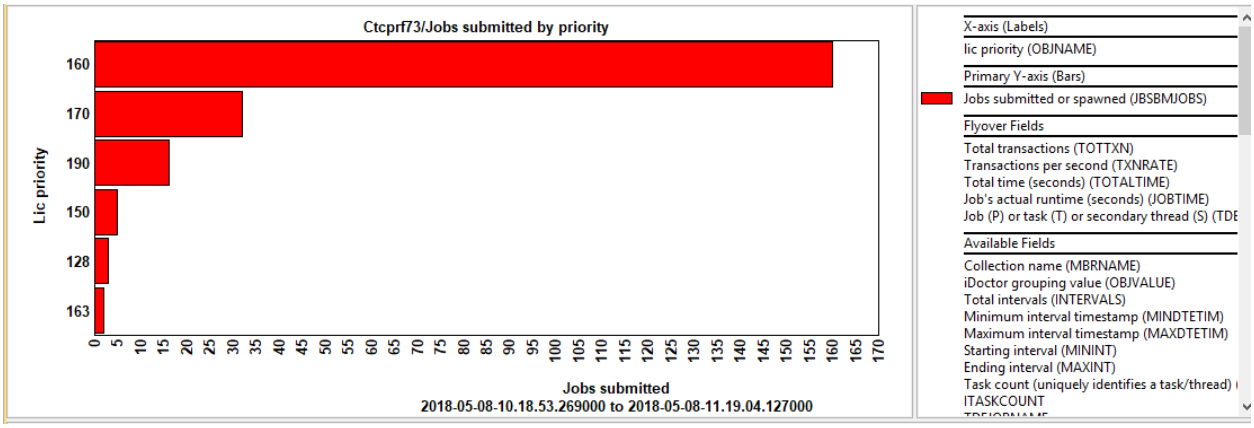
For more examples of the various types of job groupings, see the 5250 transactions rankings section.

8.26.8 Jobs submitted rankings

These graphs rank the number of jobs submitted using various job groupings.



Other graphs -> Jobs submitted rankings

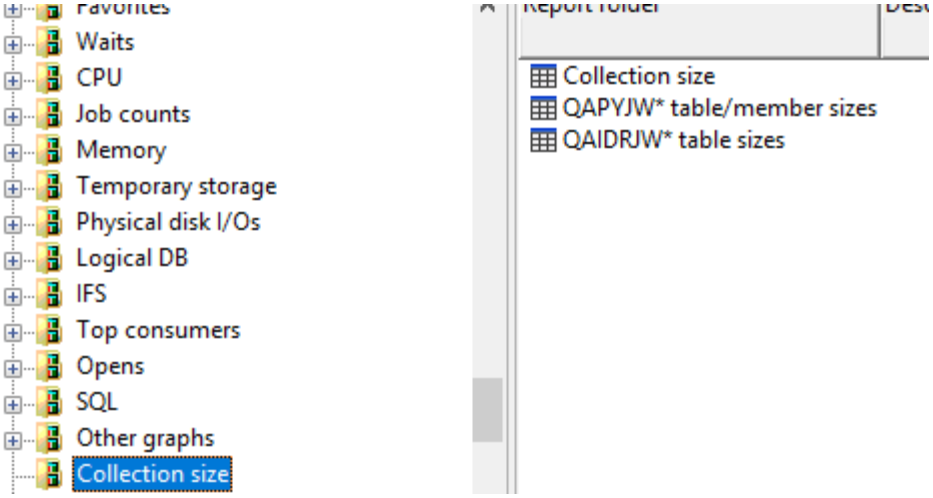


Jobs submitted by priority

For more examples of the various types of job groupings, see the 5250 transactions rankings section.

8.27 Collection size

These reports are used to displays details about the space consumed by the current collection.



Collection size

8.27.1 Collection size

This report calculates the total size of the Job Watcher collection files, but not including the iDoctor analyses SQL tables.

Library name (LIBNAME)	Collection name (MBRNAME)	Size (megabytes) (JW_COLSIZE)
MCCARGAR	Q128101852	522.1211

Collection size

8.27.2 QAPYJW* table/member sizes

This report displays each QAPYJW* file/member and displays the size of each one in megabytes.

Library name (LIBNAME)	Collection name (MBRNAME)	LONG FILE NAME (TABLE_NAME)	Size (megabytes) (JW_COLSIZE)
MCCARGAR	Q128101852	QAPYJWSTK	189.4297
MCCARGAR	Q128101852	QAPYJWTDE	151.6992
MCCARGAR	Q128101852	QAPYJWPRC	126.9141
MCCARGAR	Q128101852	QAPYJWSTS	52.0664
MCCARGAR	Q128101852	QAPYJWPROC	.6836
MCCARGAR	Q128101852	QAPYJWSQL	.5156
MCCARGAR	Q128101852	QAPYJWSQLH	.4453
MCCARGAR	Q128101852	QAPYJWBKT	.1289
MCCARGAR	Q128101852	QAPYJWINTI	.1211
MCCARGAR	Q128101852	QAPYJWSYS	.0547
MCCARGAR	Q128101852	QAPYJWRUNI	.0430
MCCARGAR	Q128101852	QAPYJWJVM	.0117
MCCARGAR	Q128101852	QAPYJWJVTH	.0078

QAPYJW table/member sizes*

8.27.3 QAIDRJW* table sizes

This report displays the sizes of the iDoctor created SQL tables for this collection.

Library name (LIBNAME)	Collection name (MBRNAME)	LONG FILE NAME (TABLE_NAME)	Size (megabytes) (JW_COLSIZE)
MCCARGAR	Q128101852	QAIDRJWGAP_Q128101852	546.0156
MCCARGAR	Q128101852	QAIDRJWSTS_Q128101852	38.0195
MCCARGAR	Q128101852	QAIDRJWXRFPROC_Q128101852	.6758
MCCARGAR	Q128101852	QAIDRJWTL_Q128101852	.5195
MCCARGAR	Q128101852	QAIDRJWSUM_Q128101852	.5078
MCCARGAR	Q128101852	QAIDRJWXRFTDE_Q128101852	.3477
MCCARGAR	Q128101852	QAIDRJWCLTSUM_Q128101852	.0313
MCCARGAR	Q128101852	QAIDRJWXRFRUNI_Q128101852	.0195
MCCARGAR	Q128101852	QAIDRJWANL_DTL_Q128101852	.0117
MCCARGAR	Q128101852	QAIDRJWCLT_Q128101852	.0117

QAIDRJW table sizes*

8.28 Interval Summary Property Pages

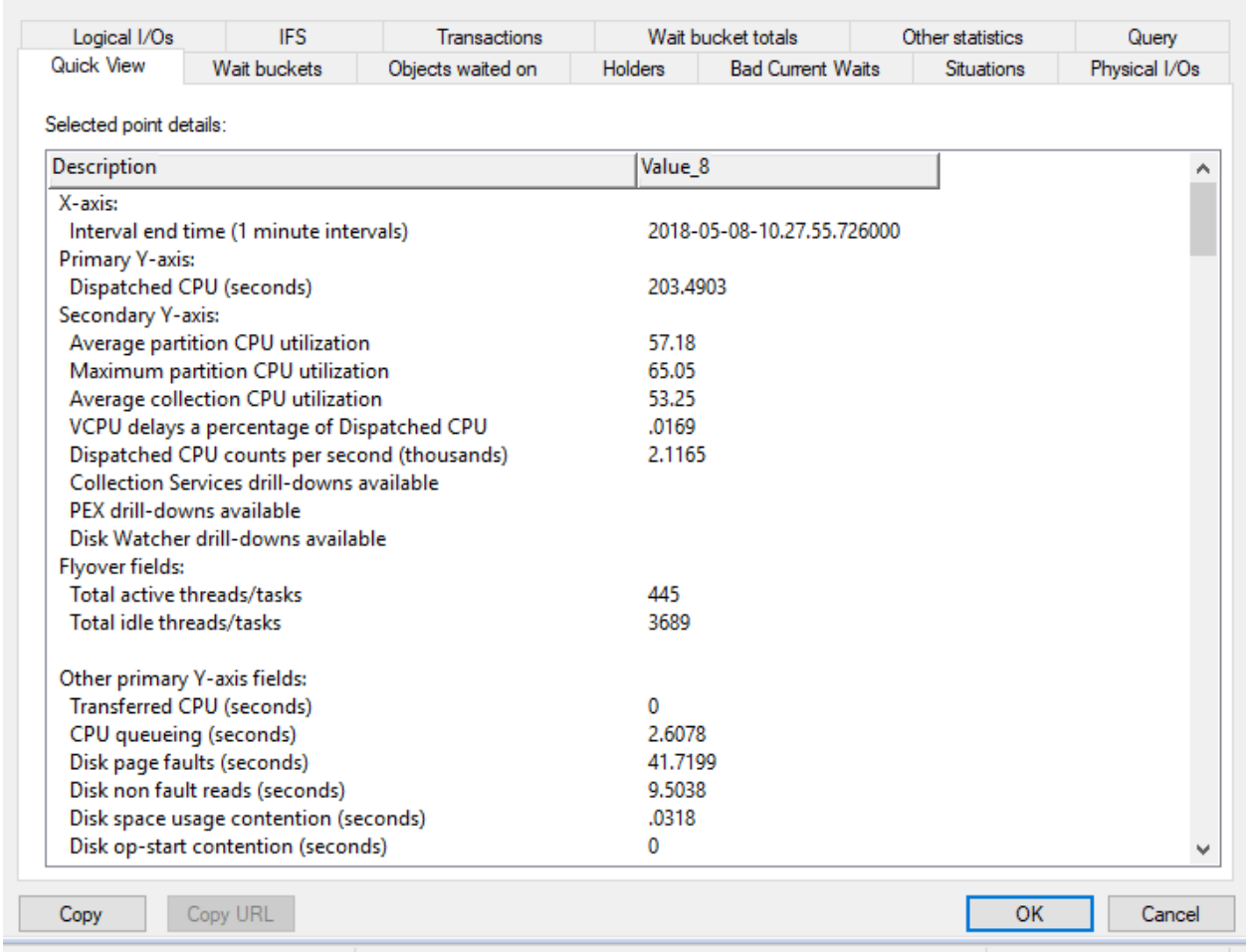
The interval summary property pages are a series of panels that provide more detailed information about the desired interval for a collection.

To access this interface simply double-click the desired interval from any overview graph.

Several tabs are shown, each covering a specific set of metrics or purpose.

8.28.1.1 Quick View

The Quick View tab displays the data from the desired bar/row in a vertical list for easier readability. This shows a complete list of all field descriptions and values from the SQL statement used to build the graph. These fields are grouped together and shown on the other tabs of this interface for easier usability.

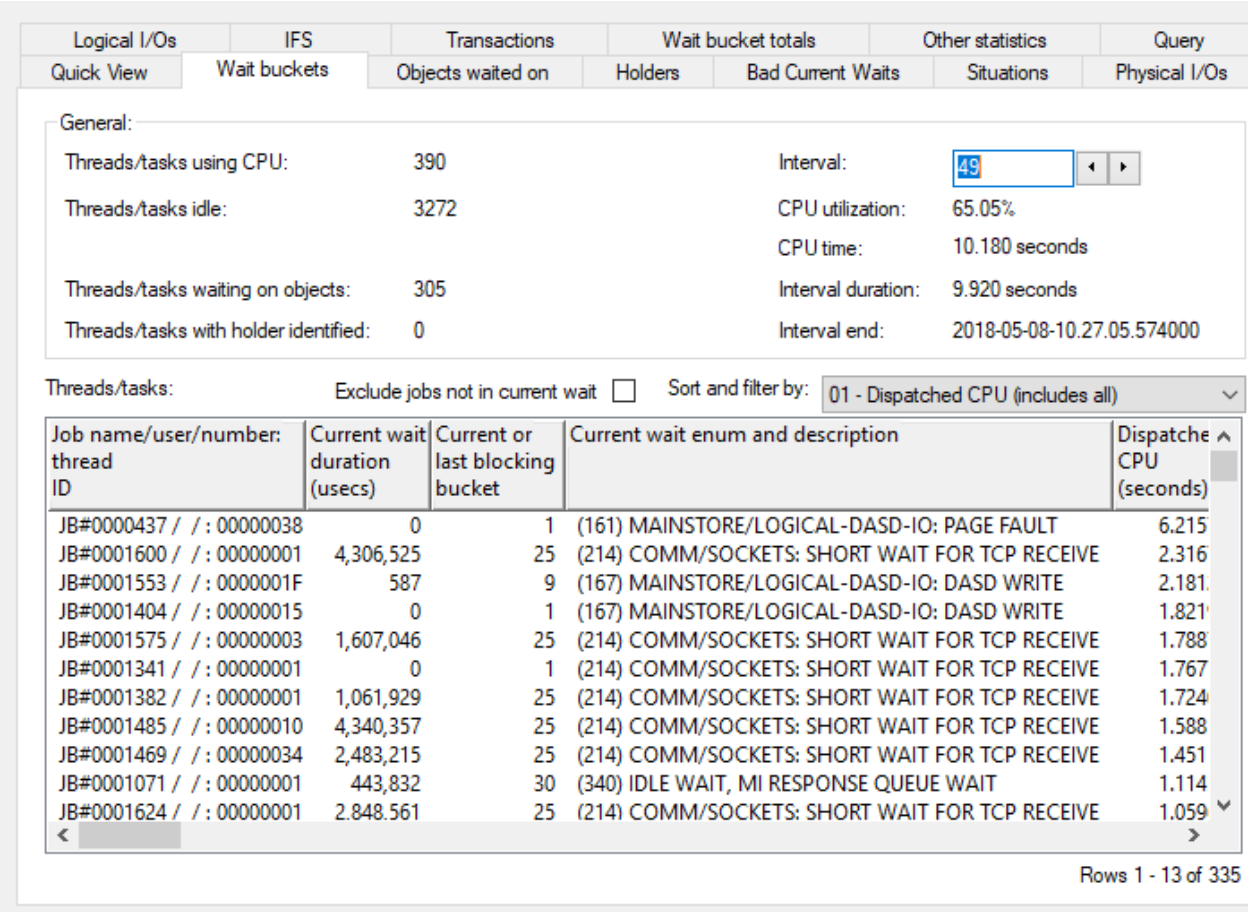


Interval Summary – Quick View

8.28.1.2 Wait buckets

The Wait buckets tab shows all jobs in the interval that experienced wait time for the desired wait bucket (specified using the Sort and Filter by drop down list).

In the example below only jobs that had some CPU time are included. The data is also sorted by Dispatched CPU time in descending sequence.



Interval Summary - Wait buckets

The General section at the top of this interface is the same on all other pages (except Query.) Therefore, the General section fields will be described in the following table but apples to the other tabs too:

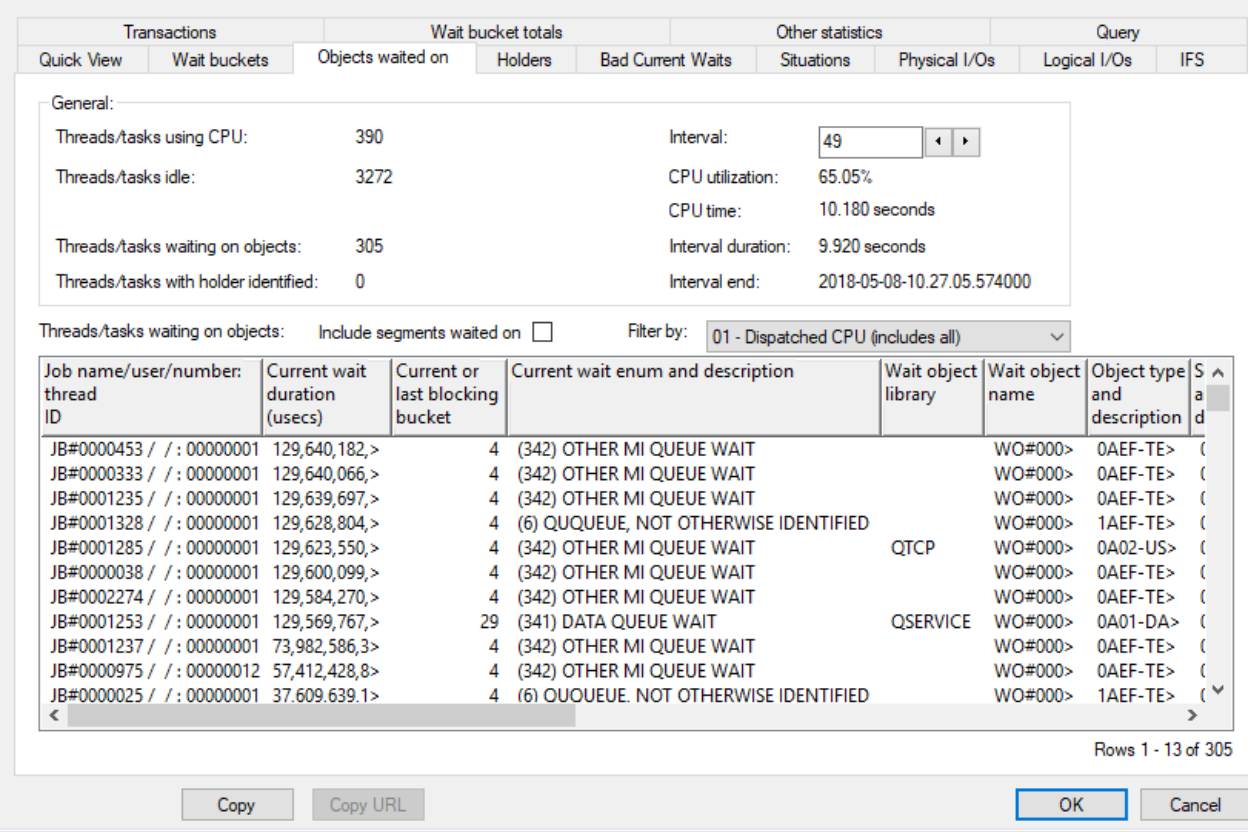
GUI Element	Description
Threads/tasks using CPU	This is the total number of threads/tasks in the interval where CPU usage was > 0.
Threads/tasks idle	The total number of threads/tasks where CPU usage was 0.
Threads/tasks waiting on objects	The total number of threads/tasks that were waiting on an object. **Note:** If the collection summary analysis has not been ran, this value may be lower than it really is.
Threads with holder identified	The total number of threads/tasks that had a holder (another thread/task preventing work being done.) **Note:** If the collection summary analysis has not been ran, this value may be lower than it really is.
Interval	The current interval number being analyzed. **Tip:** Use the arrows to move to the next or previous interval number.
CPU utilization	The average partition CPU utilization for the interval.
CPU time	The total CPU time used by all jobs in the interval.
Interval duration	The delta time between the end of this interval and the end of the previous interval.
Interval end	The timestamp that marked the end of the current interval.

Exclude jobs not in current wait	If this option is unchecked, then all jobs that contain data in the selected wait bucket are shown for the interval. If this option is checked, then only jobs that were in the selected wait bucket during the current wait (the wait that occurred at the end of the interval when the JW snapshot was taken) will be shown.
Sort and filter by	This contains a list of all wait buckets available. Picking one of these wait buckets will sort and filter by the desired wait bucket and only show jobs in the list that experienced the selected wait bucket (or CPU) time.

8.28.1.3 Objects waited on

The Objects waited on tab displays information about the wait objects that were detected by Job Watcher for the jobs/threads running on the system in a single interval. A wait object is the object that the current job wants to use but can't. Sometimes the current job may be waiting for another job to release its lock on the object.

An example of this interface is:



Interval Summary - Objects waited on

Tip: If you want to include segments waited on in the output, then check the "Include segments waited on" checkbox.

On the Objects waited on page, the list contains all waiting jobs with a wait object identified where the type of wait occurring matches the one shown in the drop-down list.

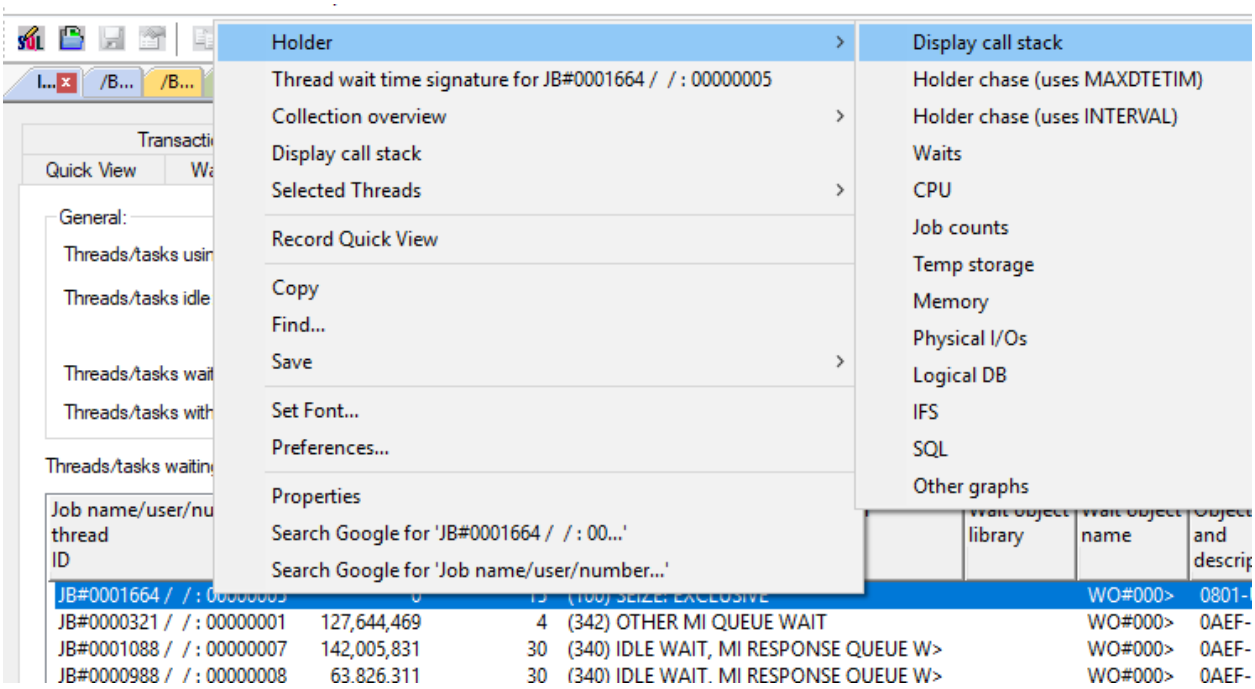
Note: If the drop-down list is set to Dispatched CPU or CPU queuing then all jobs that had a wait object are shown.

The list of jobs waiting on objects contains the following fields:

Column	Description
Waiting job or task information	This is the complete job name/user/number: thread ID or task name that is waiting on the object.
Current wait time	This value is the current wait duration (in microseconds). This is how long the job has been waiting on the object to become available. The type of wait is shown in the next column
Current wait (enum/eye catcher) description	This shows the wait enum (number identifying a specific type of wait), and eye catcher (a SLIC code used to identify different types of waits) and a description of the enum.
Wait object	The name of the wait object. If the wait object is a file this will contain the library and filename.
Object type and description	This field contains the wait object type and description.
Segment type and description	This is the segment type code and description.
Record number if DB record lock conflict	If the wait type happens to be a record lock, then this field shows the record number where the record lock occurred.
Holding job or task information	This is the job name/user/number (without thread ID) of the holder job. This is the job that is holding/locking the object the current job is waiting on. A holder job will not always be present.

8.28.1.3.1 Drilldown for a job waiting on an object

To perform a drill down on a specific job, simply right-click the row for the waiting job and pick one of the "Selected thread" menu options. If the job had a holder on the wait object, you can use the "Holder" menu to drill down into the holder job instead. Often the holder job (or its call stack) shows the cause of the problem whereas the waiter job is usually just one of the "victims" of the holder.



Drilldown example

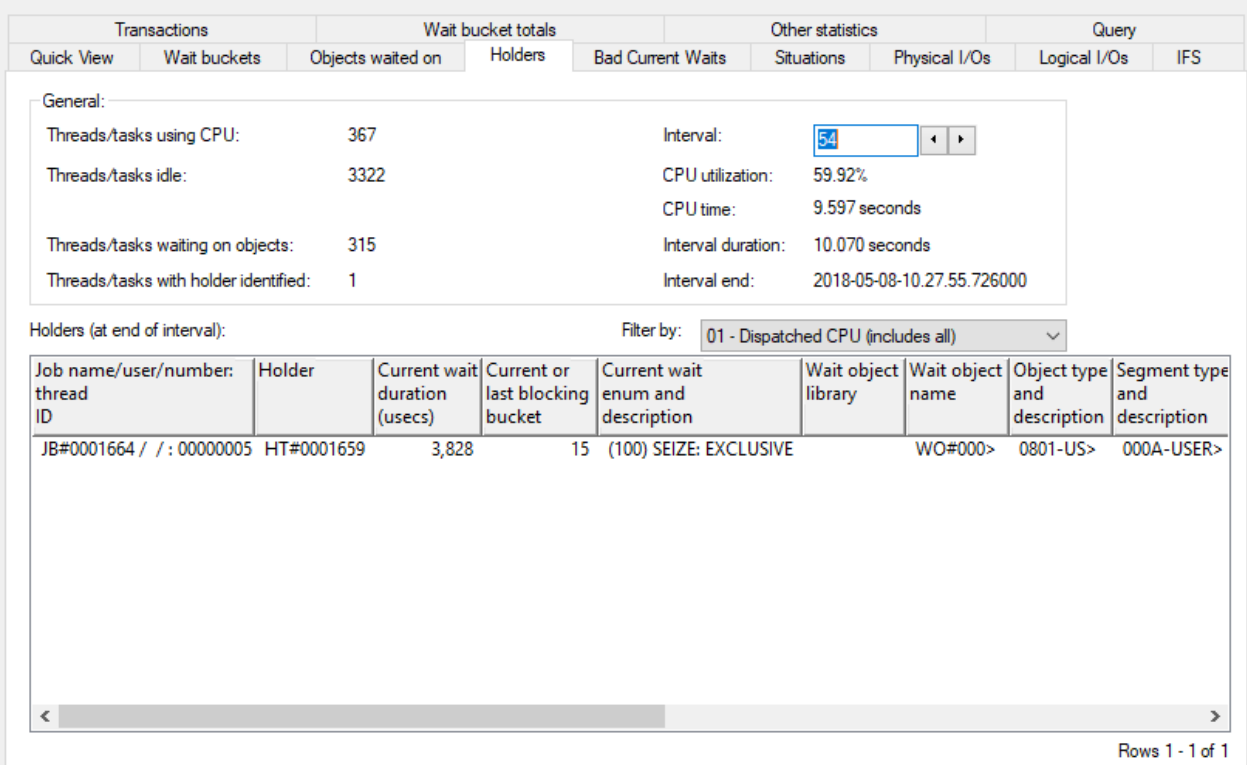
In the example, there is a "Holder chase" option. Sometimes holder jobs can be held up by other holders. This report traverses the holder "chain" until no more holders are found.

Collection name (MBRNAME)	Call level (LEV...	Interval number (INTERVA...	iDoctor grouping name (OBJNAME)	Generic job name (GENJOBNA...	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Holder taskcount (HTASKCNT)	Current or last blocking bucket (BLOCKBCK...	Current or last blocking enum (BLOCKENUM)	Current wait duration (usecs) (CURRWTDU...	Cu us pr (C
Q128101852	1	54	JB#0001664: 156919	JB#000*	156,919	156,803	15	100	3,828	U
Q128101852	2	54	JB#0001659: 156803	JB#000*	156,803	0	10	72	2,646	U

Holder chase example (The waiter job is at level 1 and the holder job is at level 2 with no other holders)

8.28.1.4 Holders

The Holders page is very similar to the Objects waited on page except it only shows jobs in the list that had a holder job. The drilldown options are like the ones described previously under the Objects waited on section.



Interval Summary - Holders

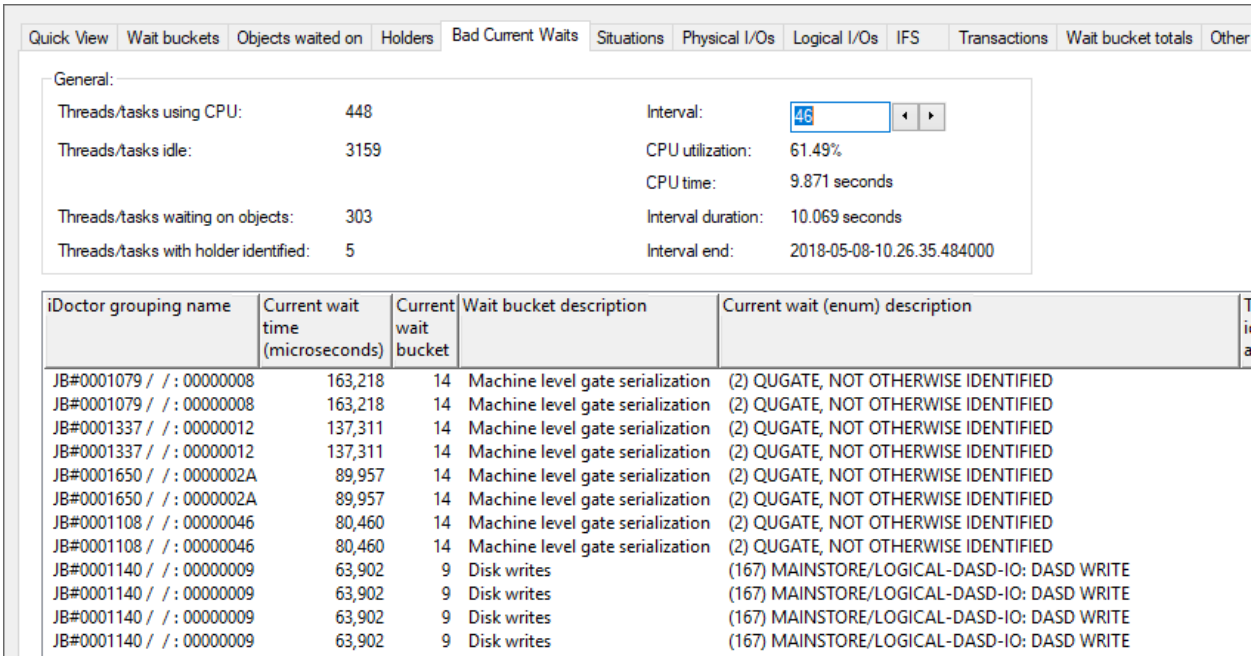
8.28.1.5 Bad Current Waits

The Bad Current waits tab shows the jobs that were experiencing a known “bad” or “interesting” type of wait at the end of the interval.

Note: This tab will only appear if the Collection summary analysis has been ran on this collection.

For 6.1 and 7.1 the wait bucket numbers included are: 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19, 31, 32.

For 7.2 and higher the wait buckets included are: 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 31, 32.



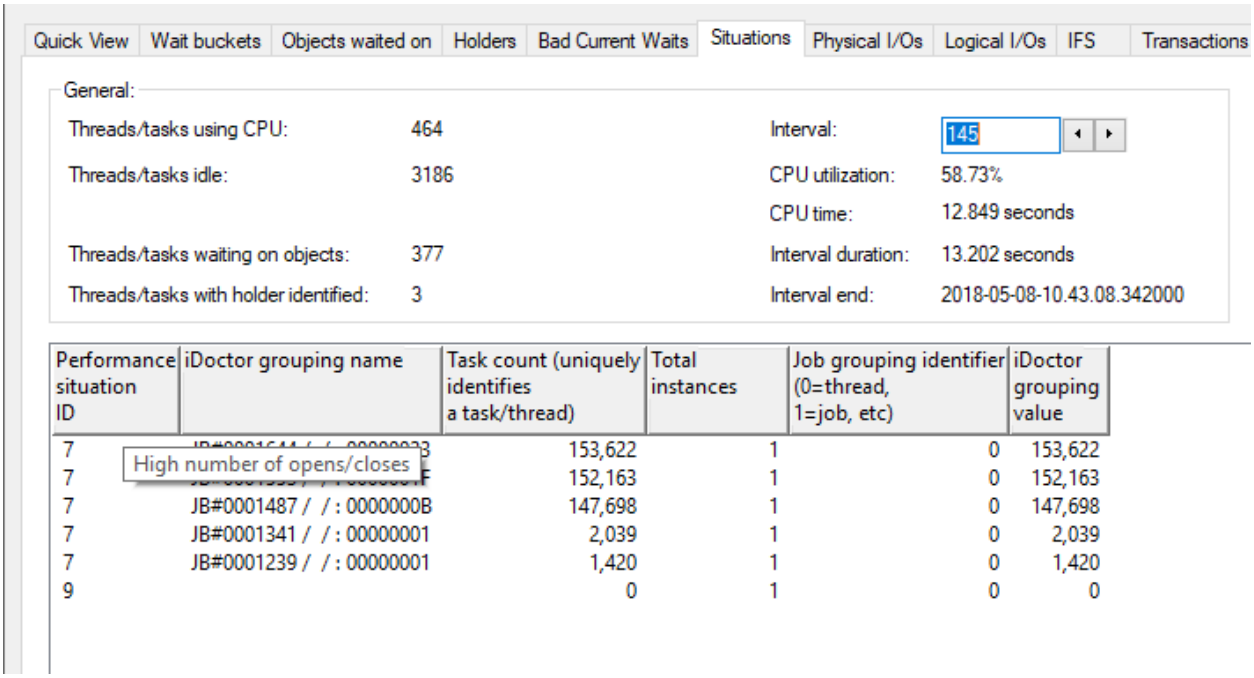
Interval Summary - Bad Current Waits

From the list the user can right-click the desired job and pick one of the “Selected Thread” drill down graphs in order to graph the job’s data over time.

8.28.1.6 Situations

The Situations tab displays jobs that were detected by the Summary process as having experienced one or more situations defined in the Job Watcher Situational Analysis feature.

Note: This tab will only appear if the Collection summary analysis has been ran on this collection.



Interval Summary - Situations

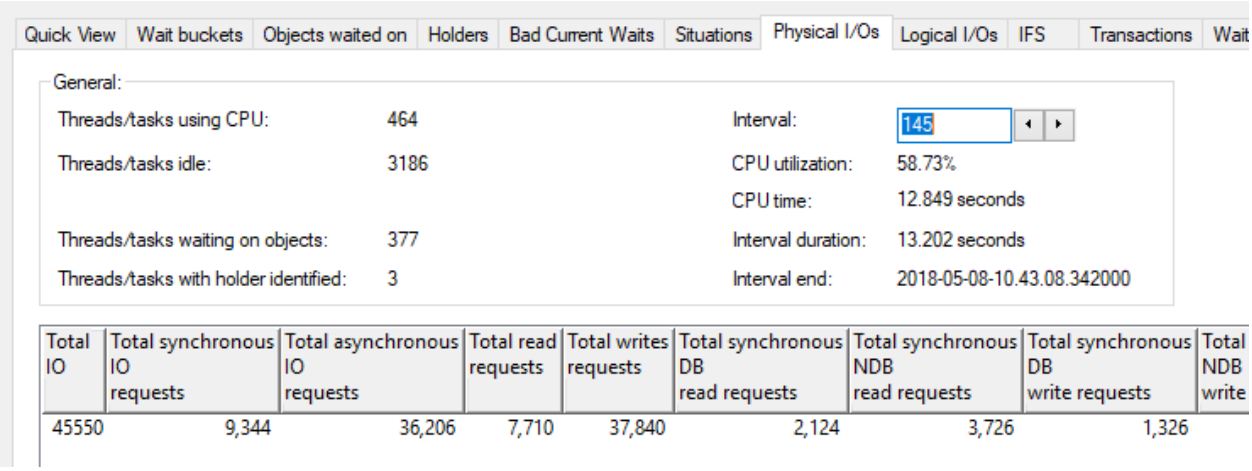
Tip: You can put your mouse over the situation ID column to get a description of the situation.

From the list the user can right-click the desired job and pick one of the “Selected Thread” drill down graphs in order to graph the job’s data over time.

8.28.1.7 Physical I/Os

This panel displays physical I/O metrics for the desired interval.

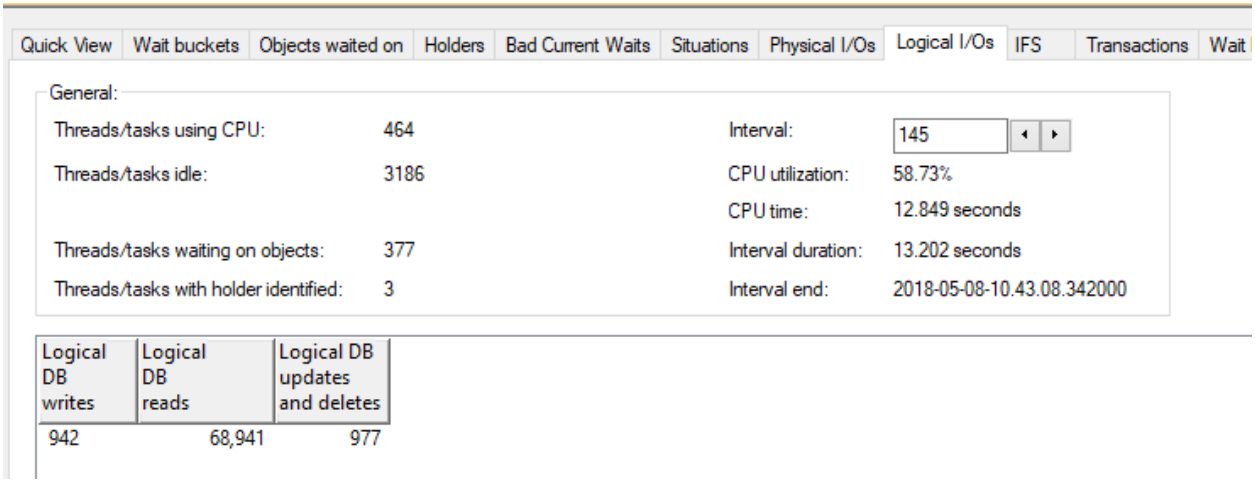
Tip: Right-click the record and use the Record Quick View menu to see all the fields at once more easily.



Interval Summary – Physical I/Os

8.28.1.8 Logical I/Os

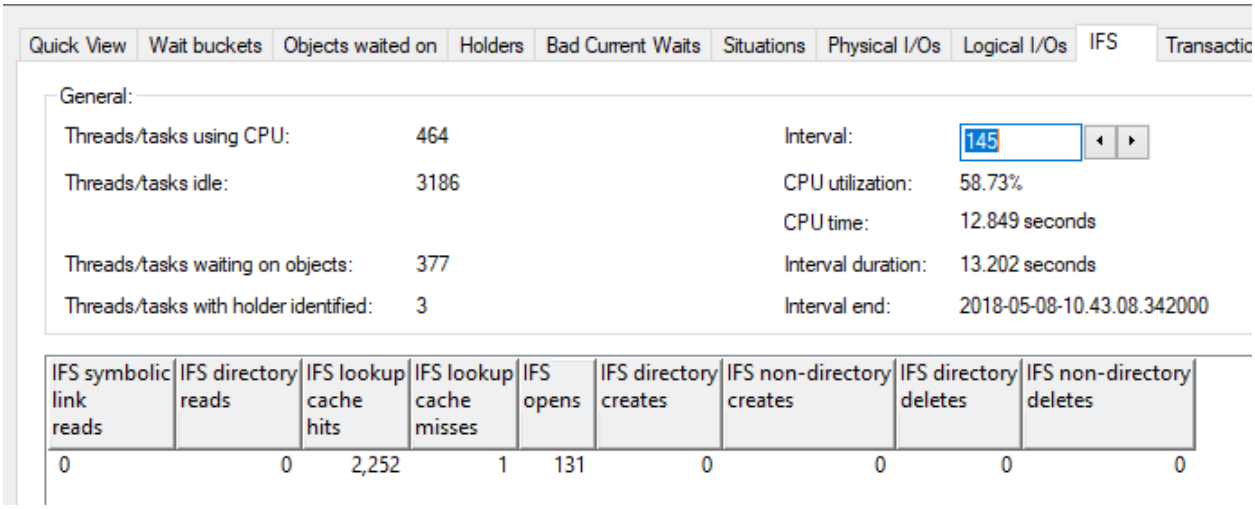
This panel displays logical I/O metrics for the desired interval.



Interval Summary – Logical I/Os

8.28.1.9 IFS

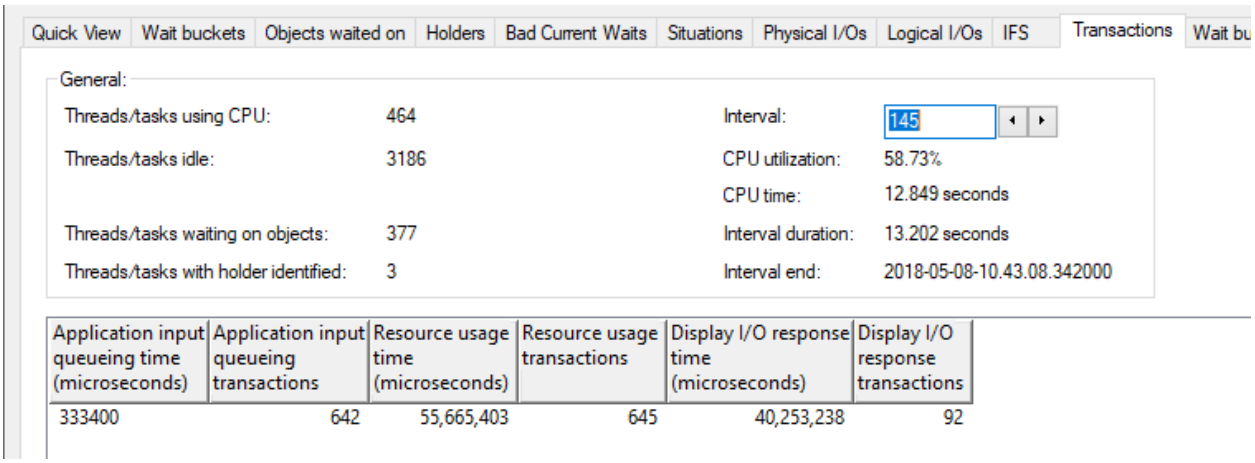
This panel displays IFS metrics for the desired interval.



Interval Summary - IFS

8.28.1.10 Transactions

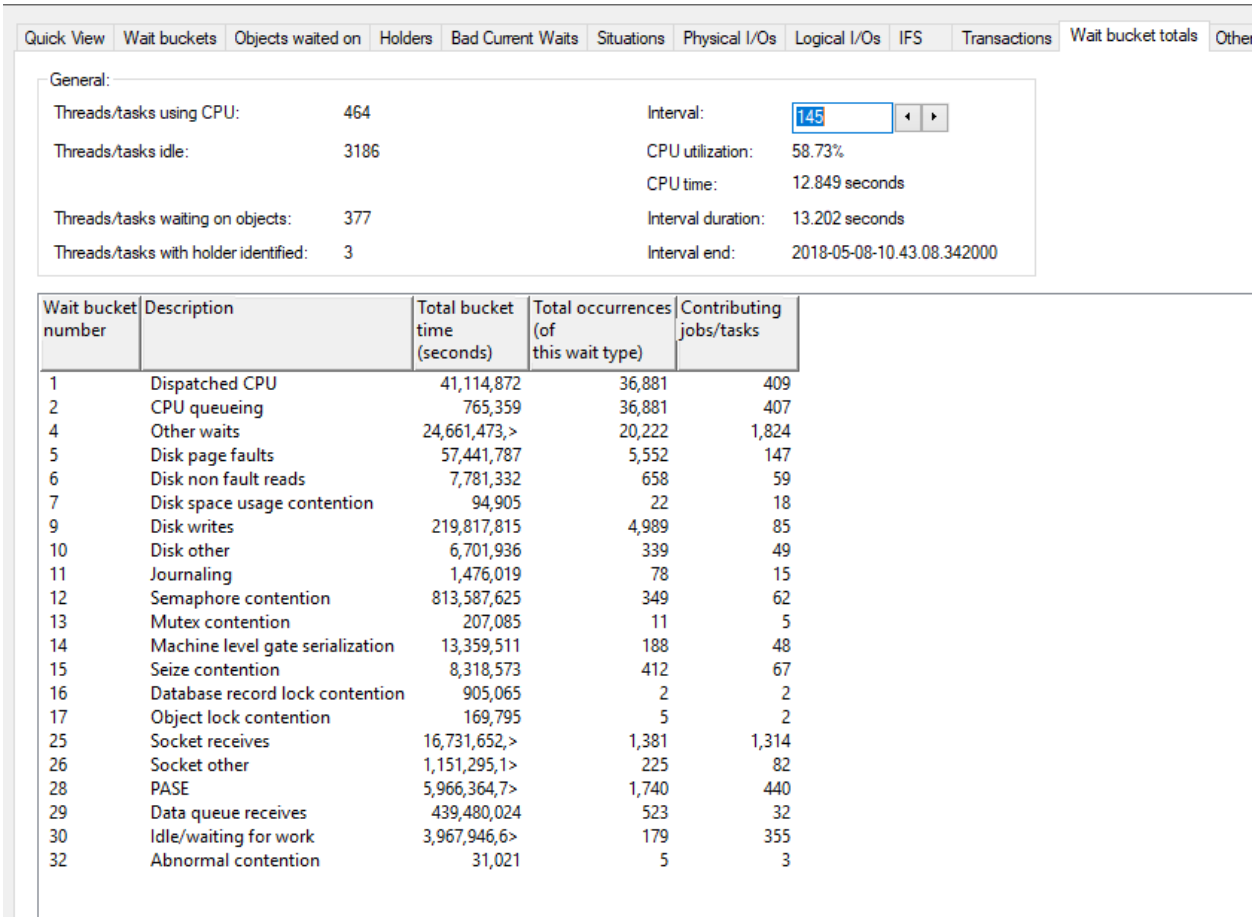
The transactions panel displays information about green screen transactions that occurred in the specified interval.



Interval Summary – Transactions

8.28.1.11 Wait bucket totals

The wait bucket totals panel displays the total amount of time all the jobs in the interval spent time in. Wait buckets that no jobs spent any time in are excluded from the list.

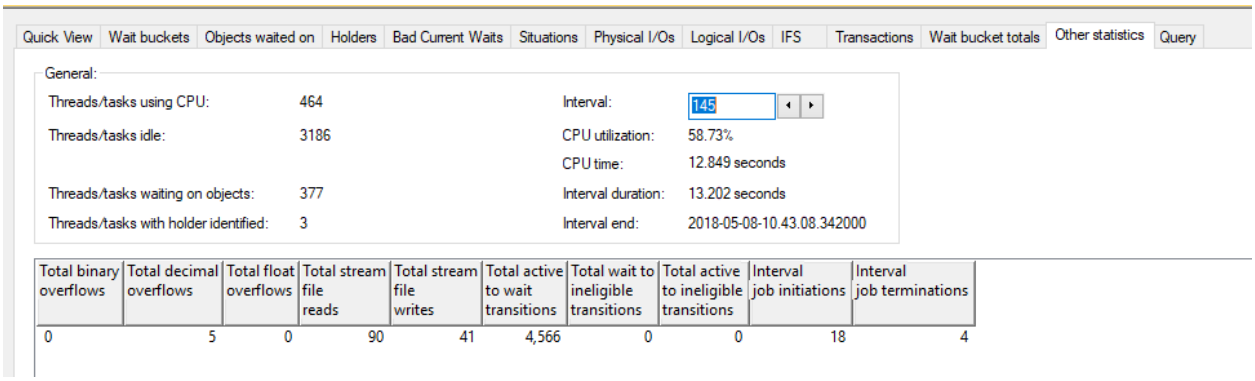


Wait bucket number	Description	Total bucket time (seconds)	Total occurrences (of this wait type)	Contributing jobs/tasks
1	Dispatched CPU	41,114,872	36,881	409
2	CPU queueing	765,359	36,881	407
4	Other waits	24,661,473,>	20,222	1,824
5	Disk page faults	57,441,787	5,552	147
6	Disk non fault reads	7,781,332	658	59
7	Disk space usage contention	94,905	22	18
9	Disk writes	219,817,815	4,989	85
10	Disk other	6,701,936	339	49
11	Journaling	1,476,019	78	15
12	Semaphore contention	813,587,625	349	62
13	Mutex contention	207,085	11	5
14	Machine level gate serialization	13,359,511	188	48
15	Seize contention	8,318,573	412	67
16	Database record lock contention	905,065	2	2
17	Object lock contention	169,795	5	2
25	Socket receives	16,731,652,>	1,381	1,314
26	Socket other	1,151,295,1>	225	82
28	PASE	5,966,364,7>	1,740	440
29	Data queue receives	439,480,024	523	32
30	Idle/waiting for work	3,967,946,6>	179	355
32	Abnormal contention	31,021	5	3

Interval Summary – Wait bucket totals

8.28.1.12 Other statistics

The other statistics panel displays numeric overflows, stream file statistics and job state transitions.



Total binary overflows	Total decimal overflows	Total float overflows	Total stream file reads	Total stream file writes	Total active to wait transitions	Total wait to ineligible transitions	Total active to ineligible transitions	Interval job initiations	Interval job terminations
0	5	0	90	41	4,566	0	0	18	4

Interval Summary – Other statistics

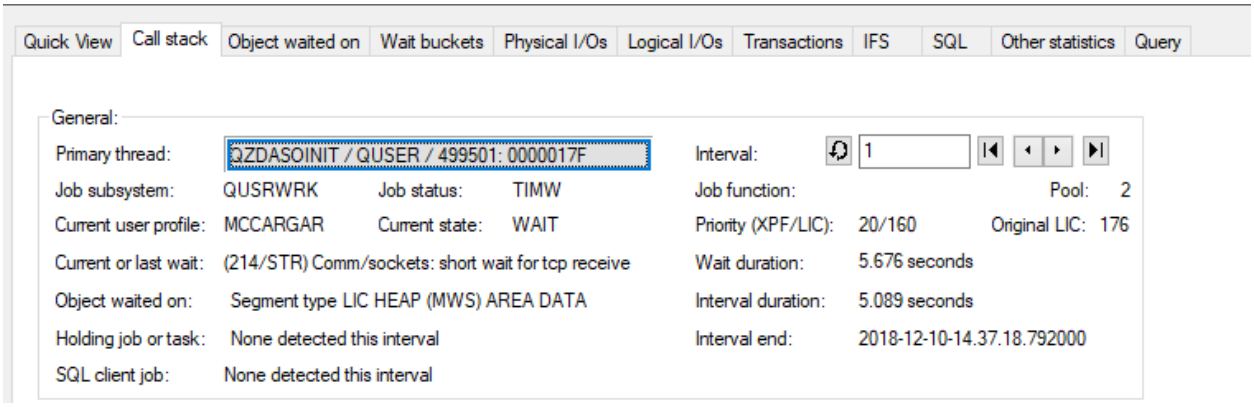
8.29 Interval Details Property Pages

These pages show information about a thread or task during an interval. The information provided includes the call stack, wait object, holder job, wait buckets, physical disk I/Os, IFS statistics, SQL statements, logical I/Os, display transactions and more.

8.29.1 General Section

The interval details property pages contain a section at the top that is consistent for all tabs except Quick View and Query. This section allows the user to consistently see required data about the thread or task when viewing any of the property pages.

An example of this section is:



Option	Description
Primary thread, secondary thread or system task	This label of this first field varies depending on the type of thread or task. This field simply shows the job name/user/number and thread ID or the system task name and taskcount. **Tip:** You can also left-click this field in order to drill down into one of the Selected Thread graphs.
Interval	These buttons allow you to navigate through the intervals for the current job.  Refresh the data for the interval given in the text box.  or  Move to the previous or next interval where a QAPYJWTDE record exists. Since Job Watcher only collects these records when CPU was used, gaps may exist in the data and these buttons allow you to quickly jump over these gaps. **Tip:** If on the Call stack tab these buttons will take you to the previous or next valid call stack. **Tip 2:** If on the SQL tab these buttons will take you to the prior or next collected SQL statement.  These buttons increase or decrease the interval number to the next or prior one.
Job subsystem	The subsystem the job is running in.
Job status	The status indicates what the job/thread was doing at the moment the call stack was taken. The possible field values are the same as those for the Status field in WRKACTJOB's help text.
Job function	The function the primary thread of the job was in when the call stack was taken (if any).
Pool	The memory pool number the job is running in.
Current user profile	The current user profile identifies the profile under which the thread is currently running on behalf of as of the end of the time interval. For jobs that swap user profiles such as QZDA* jobs, this user profile name and the user portion of the job name will be different.
Current state	Indicates the running or waiting state for the thread. The possible values are: RUN, CPUQ or WAIT
Priority (XPF/LIC)	This field displays the XPF and LIC priorities.
Original LIC	This is the original LIC priority when the job or task started.
Current or last wait	Contains the ENUM and eye catcher as well as the enum description. The enum uniquely identifies the type of wait.
Wait duration	The duration of the current wait.
Object waited on	The name of the object waited on. Several drill down reports are available based on the wait object by clicking this field. These reports indicate how many times the wait object was detected and by which jobs.
Interval duration	The elapsed duration of the current interval.
Holding job or task	If provided, this field contains the holder job or job that is most likely preventing the current job from using CPU. A holder job could have another job holding it. You can click this field to drill down into the holder job via the Selected Thread menu options.
Interval end	The timestamp marking the end of the interval. This is the time (or very close to) when the call stack was taken.
SQL client job	This is primarily used to indicate the job that is causing the current QSQSRVR job to perform work. Typically, you will see no value in this field unless you are viewing a QSQSRVR job.

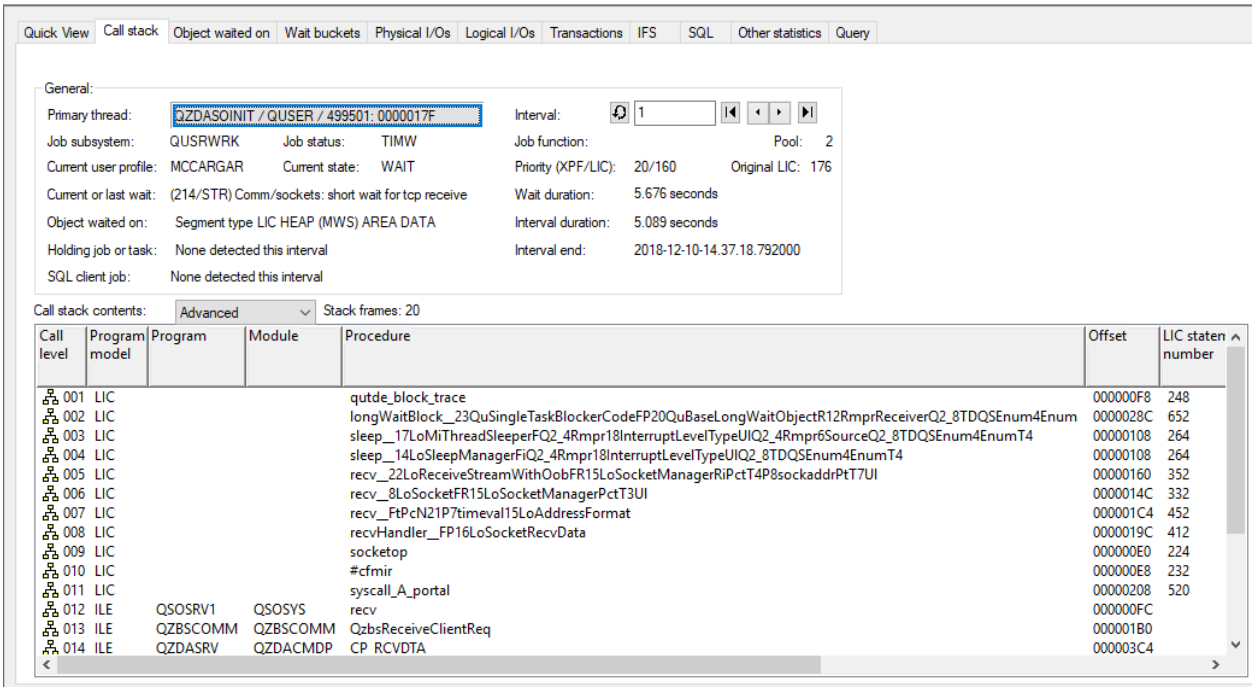
8.29.2 Call Stack

The Call Stack panel helps tell you what the job/application was doing at the end of the Job Watcher interval.

The stack can be up to 1000 levels deep and provides complete information about the program/module /procedure for each level. Call stacks are also provided for J9 (Pase) jobs. These have a different look to them as the data must be retrieved from a separate file for J9 call stacks.

Above the call stack you may notice a label that indicates how many holder call stacks were collected. This is an unintended feature of Job Watcher where call stacks are collected for holder jobs. If a holder job is holding up many waiter jobs a call stack can be collected of the holder job for every waiter. These are of the holder job for the same interval but at different very slightly different instances in time (perhaps less than a microsecond apart). Most of these stacks are going to be identical and there is currently no way to view these (you just see the 1st one). But the fact that this many holder stacks were collected gives you a strong indicator of how many jobs this job was 'holding up".

An example of this interface is:



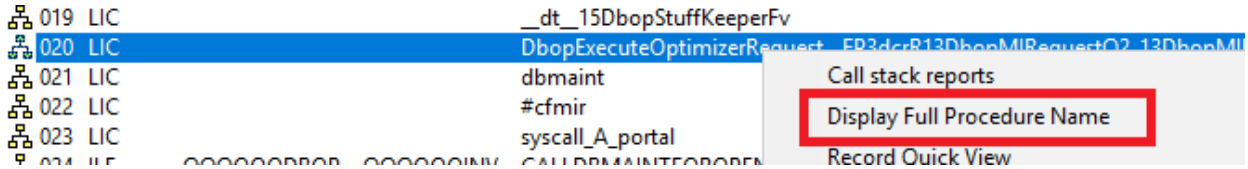
Interval Details – Call Stack

A table that describes the options on this page (under the General section) is shown below:

Option	Description
(Call stack mode) drop-down box	This option controls the number of columns displayed for the call stack with varying levels of complexity. It will contain 3 options and an optional 4th option if the Retrieve program/module details analysis has been ran against the collection. **Basic** - Shows the call level, model, program, module and procedure **Detailed** – Basic + Offset, LIC statement number **Advanced** – Detailed + 4 addresses **Program Details –** Basic + additional fields related to the program (type, object attribute, description, LICPGM name, LICPGM level, PTF ID, APAR ID, created by user, signed by trusted source, creation date, change date)
Label	A label after the drop-down box indicates some additional information about the call stack. The reason collected labels all indicate the call stack was captured for a job that did not use CPU in the current interval, but it was captured anyway for one of the reasons listed. 1. Stack frames (number of call levels) 2. Reason collected: Holder (call stack was collected because it is a holding another job) 3. Reason collected: Waiter (call stack was collected because it is held by another job) 4. Reason collected: Bad wait (call stack was collected because the job was stuck in a bad wait)

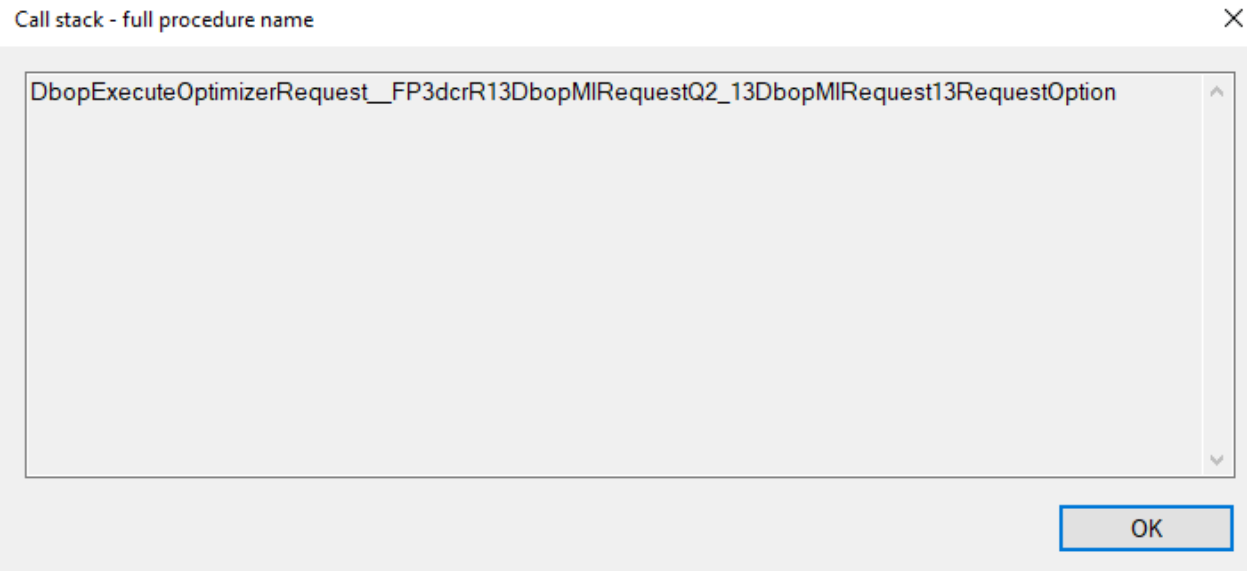
8.29.2.1 Display Full Procedure Name

If there is a particularly long procedure name in the stack (they can be many hundreds of characters long) and you need to see the entire thing you can right-click the row where the procedure is found and choose the Display Full Procedure Name menu.



Display Full Procedure Name menu

This option will display the full procedure name in a window like this:

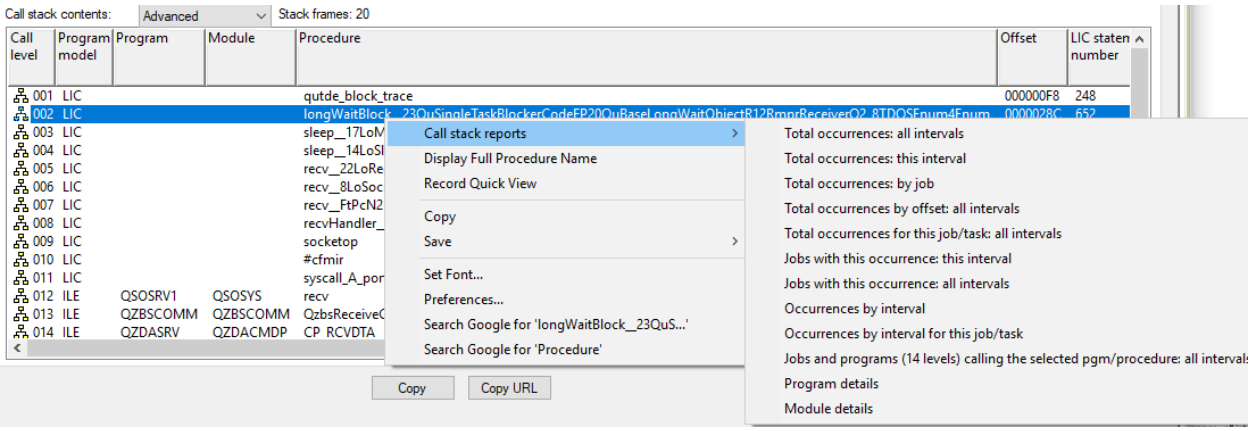


Call stack – full procedure name

8.29.2.2 Call Stack reports

If you wish to know how frequently a program/procedure in the stack was found in other call stacks in the collection, you can right-click (1 or more) selected rows from the call stack and a menu option is displayed giving you several ways to look for that same stack information in the other jobs in the collection.

If you select one 1 row of the call stack, then the query will look for only call stacks that contain the traceback table address of the entry selected. If you select multiple rows then all of those traceback table address entries for those call levels selected must exist in order to be returned in the resulting reports.

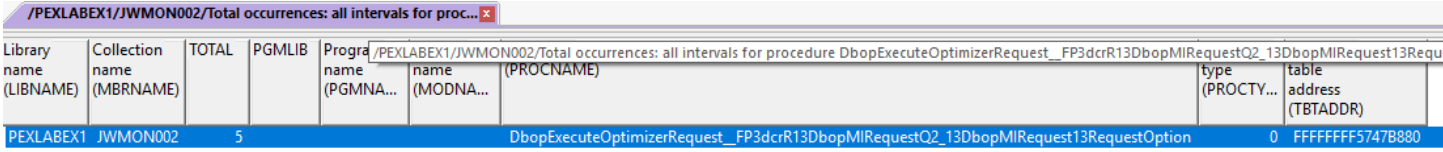


Call Stack Reports Menu from Interval Details – Call Stack

PLEASE NOTE: None of these call stack reports return any data for J9 JVM call stack levels.

8.29.2.2.1 Total occurrences: all intervals

This report looks for the desired call stack entries in the entire collection and counts the total occurrences.



Total occurrences: all intervals

8.29.2.2.2 Total occurrences: this interval

This report is the same as the previous one except it only looks for matches for the current interval.

/PEXLABEX1/JWMON002/Total occurrences: this interval for procedure QDBCRTME...

/PEXLABEX1/JWMON002/Total occurrences: this interval for procedure QDBCRTME offset 00006A60 - #1

Library name (LIBNAME)	Collection name (MBRNAME)	TOTAL		name (PGMNAME)	name (MODNAME)	(PROCNAME)	type (PROCTY...	table address (TBTADDR)
PEXLABEX1	JWMON002	1	QSYS	QDBCRTME	QDBCRTME	QDBCRTME	1	3924483DB703EAC0

Total occurrences: this interval

8.29.2.2.3 Total occurrences: by job

This report looks for the desired call stack entries in the entire collection and counts the total occurrences on a per job basis.

Library name (LIBNAME)	Collection name (MBRNAME)	TOTAL	Task count (uniquely identifies a task/thread) (TASKCOUNT)	TDEJOBNAME	THREADID	Ger job nar (GE
PEXLABEX1	JWMON002	1	3,364,365	WEEKLY BSMENGES 718461	0000000000000027	WE
PEXLABEX1	JWMON002	1	1,725	QGLDPUBA QDIRSRV 707463	0000000000000001	QG
PEXLABEX1	JWMON002	1	1,761	QIJSSCD QIJS 707511	0000000000000002	QI.
PEXLABEX1	JWMON002	1	694,590	QIDRDIRMONMCCARGAR 712953	0000000000000018	QI
PEXLABEX1	JWMON002	1	3,361,280	ADMIN2 QLWISVR 707545	0000000000081158	AD
PEXLABEX1	JWMON002	1	2,306	QTSMTPSRVDQTCP 707659	0000000000000001	QT
PEXLABEX1	JWMON002	1	2,315	QTSMTPCLTDQTCP 707667	0000000000000002	QT
PEXLABEX1	JWMON002	1	2,298	QSLPSVR QSYS 707658	0000000000000001	QS
PEXLABEX1	JWMON002	1	3,349,297	CRTPFRDTA QSYS 718209	000000000000004A	CR
PEXLABEX1	JWMON002	1	3,364,332	QIDRPACOL BSMENGES 718460	00000000000000C6	QI
PEXLABEX1	JWMON002	1	1,854	ADMIN QTMHHTTP 707528	0000000000000001	AD
PEXLABEX1	JWMON002	1	1,737	QPASUTIL QSYS 707494	0000000000000002	QP
PEXLABEX1	JWMON002	1	2,220	ADMIN4 QWEBADMIN 707547	0000000000000020	AD
PEXLABEX1	JWMON002	1	3,364,306	QSTRJWMON BSMENGES 718455	0000000000000073	QS
PEXLABEX1	JWMON002	1	1,740	QYPSPFRCOLQSYS 707497	0000000000000001	QY
PEXLABEX1	JWMON002	1	2,286	ADMIN2 QLWISVR 707545	000000000000001A	AD
PEXLABEX1	JWMON002	1	2,297	QTOOROUTE QTCP 707657	0000000000000001	QT
PEXLABEX1	JWMON002	1	1,842	QUSRDIR QDIRSRV 707525	0000000000000001	QU
PEXLABEX1	JWMON002	1	3,349,309	QHTTP QTMHHTTP 718211	0000000000000014	QH
PEXLABEX1	JWMON002	1	2,287	ADMIN2 QLWISVR 707545	000000000000001B	AD

Total occurrences: by job

8.29.2.2.4 Total occurrences: by offset, all intervals

This report checks for the desired trace back table entries as well as the offset listed in the call stack (in detailed or advanced modes) and adds up the total occurrences for each offset.

Library name (LIBNAME)	Collection name (MBRNAME)	PGMLIB	Program name (PGMNA...	Module name (MODNA...	Procedure (PROCNA...	Offset time (seconds) (OFFSET)	Traceback table address (TBTADDR)	TOTAL
PEXLABEX1	JWMON002	QSYS	QCMD			00003914	2E8EAEE703013C10	979
PEXLABEX1	JWMON002	QSYS	QCMD			00008D78	2E8EAEE703013C10	8

Total occurrences: by offset, all intervals

8.29.2.2.5 Total occurrences for this job/task: all intervals

This reports adds up the total occurrences of the selected call stack entries but only for the current job.

Library name (LIBNAME)	Collection name (MBRNAME)	TOTAL	PGMLIB	Program name (PGMNA...	Module name (MODNA...	Procedure (PROCNAME)	Procedure type (PROCTY...	Traceback table address (TBTADDR)
PEXLABEX1	JWMON002	1				flushMe__17SmMainStoragePoolFv	0	FFFFFFFF4889C2D8

Total occurrences for this job/task: all intervals

8.29.2.2.6 Jobs with this occurrence: this interval

This report displays the list of jobs matching the current selection in the current time interval.

Library name (LIBNAME)	Collection name (MBRNAME)	Interval number (INTERVA...	Task count (uniquely identifies a task/thread) (TASKCOUNT)	TDEJOBNAME	THREADID	Gen job nam (GEI
PEXLABEX1	JWMON002	42	1,737	QPASUTIL QSYS 707494		QP
PEXLABEX1	JWMON002	42	1,740	QYPSPFRCOLQSYS 707497		QY
PEXLABEX1	JWMON002	42	1,842	QUSRDIR QDIRSRV 707525		QU
PEXLABEX1	JWMON002	42	2,297	QTOOROUTE QTCP 707657		QT
PEXLABEX1	JWMON002	42	2,298	QSLPSVR QSYS 707658		QS
PEXLABEX1	JWMON002	42	2,306	QTSMTPSRVDQTCP 707659		QT
PEXLABEX1	JWMON002	42	2,315	QTSMTPCLTDQTCP 707667		QT
PEXLABEX1	JWMON002	42	694,590	QIDRDIRMONMCCARGAR 712953		QI
PEXLABEX1	JWMON002	42	3,364,306	QSTRJWMON BSMENGES 718455		QS
PEXLABEX1	JWMON002	42	3,364,365	WEEKLY BSMENGES 718461		WE

Jobs with this occurrence this interval

8.29.2.2.7 Jobs with this occurrence: all intervals

This report displays the list of jobs matching the current selection in the entire collection for each interval that matched.

Library name (LIBNAME)	Collection name (MBRNAME)	Interval number (INTERVA...	Task count (uniquely identifies a task/thread) (TASKCOUNT)	TDEJOBNAME	THREADID	Gener job name (GENJ
PEXLABEX1	JWMON002	1	1,842	QUSRDIR QDIRSRV 707525		QUSR
PEXLABEX1	JWMON002	1	2,298	QSLPSVR QSYS 707658		QSLP
PEXLABEX1	JWMON002	1	694,590	QIDRDIRMONMCCARGAR 712953		QIDR
PEXLABEX1	JWMON002	1	2,297	QTOOROUTE QTCP 707657		QTOC
PEXLABEX1	JWMON002	1	2,315	QTSMTPCLTDQTCP 707667		QTSM
PEXLABEX1	JWMON002	1	1,740	QYPSPFRCOLQSYS 707497		QYPS
PEXLABEX1	JWMON002	1	1,737	QPASUTIL QSYS 707494		QPAS
PEXLABEX1	JWMON002	1	2,306	QTSMTPSRVDQTCP 707659		QTSM
PEXLABEX1	JWMON002	2	1,842	QUSRDIR QDIRSRV 707525		QUSR
PEXLABEX1	JWMON002	2	1,740	QYPSPFRCOLQSYS 707497		QYPS
PEXLABEX1	JWMON002	2	2,315	QTSMTPCLTDQTCP 707667		QTSM
PEXLABEX1	JWMON002	2	2,298	QSLPSVR QSYS 707658		QSLP
PEXLABEX1	JWMON002	2	3,364,306	QSTRJWMON BSMENGES 718455		QSTR
PEXLABEX1	JWMON002	2	2,297	QTOOROUTE QTCP 707657		QTOC
PEXLABEX1	JWMON002	2	2,306	QTSMTPSRVDQTCP 707659		QTSM
PEXLABEX1	JWMON002	3	2,298	QSLPSVR QSYS 707658		QSLP
PEXLABEX1	JWMON002	3	2,306	QTSMTPSRVDQTCP 707659		QTSM
PEXLABEX1	JWMON002	3	2,315	QTSMTPCLTDQTCP 707667		QTSM
PEXLABEX1	JWMON002	3	2,297	QTOOROUTE QTCP 707657		QTOC
PEXLABEX1	JWMON002	3	1,842	QUSRDIR QDIRSRV 707525		QUSR
PEXLABEX1	JWMON002	3	1,740	QYPSPFRCOLQSYS 707497		QYPS

Jobs with this occurrence: all intervals

8.29.2.2.8 Occurrences by interval

This report displays the total hits per interval for the call stack entries selected.

Library name (LIBNAME)	Collection name (MBRNAME)	Interval number (INTERVA...	TOTAL	PGMLIB	Program name (PGMNAME)	Module name (MODNA...	Procedure (PROCNA...	Procedure type (PROCTY...	Traceback table address (TBTADDR)	Offset time (seconds) (OFFSET)
PEXLABEX1	JWMON002	41	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4
PEXLABEX1	JWMON002	42	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4
PEXLABEX1	JWMON002	43	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4
PEXLABEX1	JWMON002	44	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4

Occurrences by interval

8.29.2.2.9 Occurrences by interval for this job/task

This report displays the total hits per interval but only for the current job/task for the call stack entries selected.

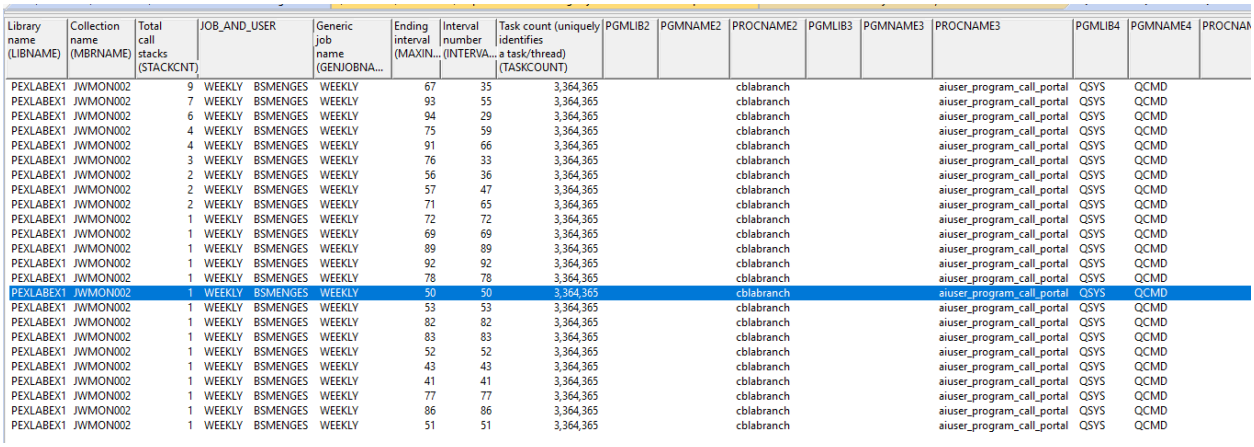
Library name (LIBNAME)	Collection name (MBRNAME)	Interval number (INTERVA...	TOTAL	PGMLIB	Program name (PGMNAME)	Module name (MODNA...	Procedure (PROCNA...	Procedure type (PROCTY...	Traceback table address (TBTADDR)	Offset time (seconds) (OFFSET)	Task count (uniquely identifies a task/thread) (TASKCOUNT)
PEXLABEX1	JWMON002	41	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4	3,364,365
PEXLABEX1	JWMON002	42	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4	3,364,365
PEXLABEX1	JWMON002	43	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4	3,364,365
PEXLABEX1	JWMON002	44	1	BSMENGES	WEEKMAINT			2	305FF49CD30029A0	000007F4	3,364,365

Occurrences by interval for this job/task

8.29.2.2.10 Jobs and programs (14 levels) calling the selected pgm/procedure

This report displays the possible call stacks found (14 levels) that called the selection made. Each 14 level section of the call stack is shown horizontally and the total hits/call stacks is shown in the 3rd column.

You can also double-click a row from this report to view that row vertically instead.

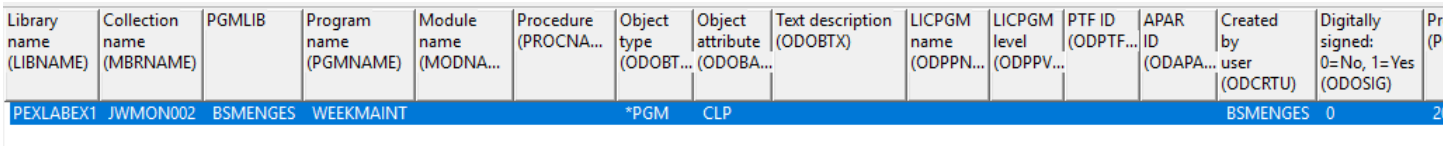


Library name (LIBNAME)	Collection name (MBRNAME)	Total call stacks (STACKCNT)	JOB_AND_USER	Generic job name (GENJOBNA...	Ending interval (MAXIN...	Interval number (INTERVA...	Task count (uniquely identifies a task/thread) (TASKCOUNT)	PGMLIB2	PGMNAME2	PROCNAME2	PGMLIB3	PGMNAME3	PROCNAME3	PGMLIB4	PGMNAME4	PROCNAM
PEXLABEX1	JWMON002	9	WEEKLY BSMENGES	WEEKLY	67	35	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	7	WEEKLY BSMENGES	WEEKLY	93	55	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	6	WEEKLY BSMENGES	WEEKLY	94	29	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	4	WEEKLY BSMENGES	WEEKLY	75	59	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	4	WEEKLY BSMENGES	WEEKLY	91	66	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	3	WEEKLY BSMENGES	WEEKLY	76	33	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	2	WEEKLY BSMENGES	WEEKLY	56	36	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	2	WEEKLY BSMENGES	WEEKLY	57	47	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	2	WEEKLY BSMENGES	WEEKLY	71	65	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	72	72	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	69	69	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	89	89	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	92	92	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	78	78	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	50	50	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	53	53	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	82	82	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	83	83	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	52	52	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	43	43	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	41	41	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	77	77	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	86	86	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	
PEXLABEX1	JWMON002	1	WEEKLY BSMENGES	WEEKLY	51	51	3,364,365			cblabranch			aiuser_program_call_portal	QSYS	QCMD	

Jobs and programs (14 levels) calling the selected pgm/procedure

8.29.2.2.11 Program details

This report displays additional details about the selected program from the call stack. It includes DSPOBJD output that is only available if the Retrieve program/module information analysis has been ran.



Library name (LIBNAME)	Collection name (MBRNAME)	PGMLIB	Program name (PGMNAME)	Module name (MODNA...	Procedure (PROCNA...	Object type (ODOBT...	Object attribute (ODOBA...	Text description (ODOBTX)	LICPGM name (ODPPN...	LICPGM level (ODPPV...	PTF ID (ODPTF...	APAR ID (ODAPA...	Created by user (ODCRTU)	Digitally signed: 0=No, 1=Yes (ODOSIG)	Pr (P
PEXLABEX1	JWMON002	BSMENGES	WEEKMAINT			*PGM	CLP						BSMENGES	0	2

Program details

This report is another example where it is more easily viewed vertically in the Record Quick View interface (double-click a row):

Field	Description	Record 1
LIBNAME	LIBNAME	PEXLABEX1
MBRNAME	MBRNAME	JWMON002
PGMLIB	PGMLIB	BSMENGES
PGMNAME	PGMNAME	WEEKMAINT
MODNAME	MODNAME	
PROCNAME	PROCNAME	
ODOBTP	Object type	*PGM
ODOBAT	Object attribute	CLP
ODOBTX	Text description	
ODPPNM	LICPGM name	
ODPPVR	LICPGM level	
ODPTFN	PTF ID	
ODAPAR	APAR ID	
ODCRTU	Created by user	BSMENGES
ODOSIG	Digitally signed: 0=No, 1=Yes	0
PGMCRTDATE	Program creation date	2016-05-06-14:10:38.000000
PGMCHGDATE	Program changed date	2017-09-22-11:12:45.000000
ODDCEN	Display century: 0=19xx, 1=20xx	1
ODDDAT	Display date (Job date format)	120618
ODDTIM	Display time (HHMMSS)	115228
ODLBNM	Library	BSMENGES
ODOBNM	Object	WEEKMAINT
ODOBTP	Object type	*PGM
ODOBAT	Object attribute	CLP
ODOBFR	Storage freed: 0=Not freed, 1=Freed	0
ODOBSZ	Object size: 9,999,999,999=Use ODSIZU*ODBPUN	45056
ODOBTX	Text description	
ODOBLK	Object locked: 0=Not locked, 1=Locked	0
ODOBDM	Object damaged: 0=Not damaged, 1=Full, 2=Partial	0
ODCCEN	Creation century: 0=19xx, 1=20xx	1
ODCDAT	Creation date (MMDDYY)	050616
ODCTIM	Creation time (HHMMSS)	141038
ODOBOW	Object owner	BSMENGES
ODSCEN	Save century: 0=19xx, 1=20xx	1
ODSDAT	Save date (MMDDYY)	082418
ODSTIM	Save time (HHMMSS)	092428
ODSCMD	Save command	SAVLIB
ODSSZE	Saved size	45056
ODSSLT	Starting slot	0
ODSDEV	Save device	Tape
ODSV01	Saved volume	000326
ODSV02	Saved volume	
ODSV03	Saved volume	

Program details report -> Record Quick View

8.29.2.2.12 Module details

This report displays additional details about the selected module from the call stack.

It only will provide data if you have selected a call level that contains a non-blank module and the Retrieve program/module information analysis has been ran on this collection.

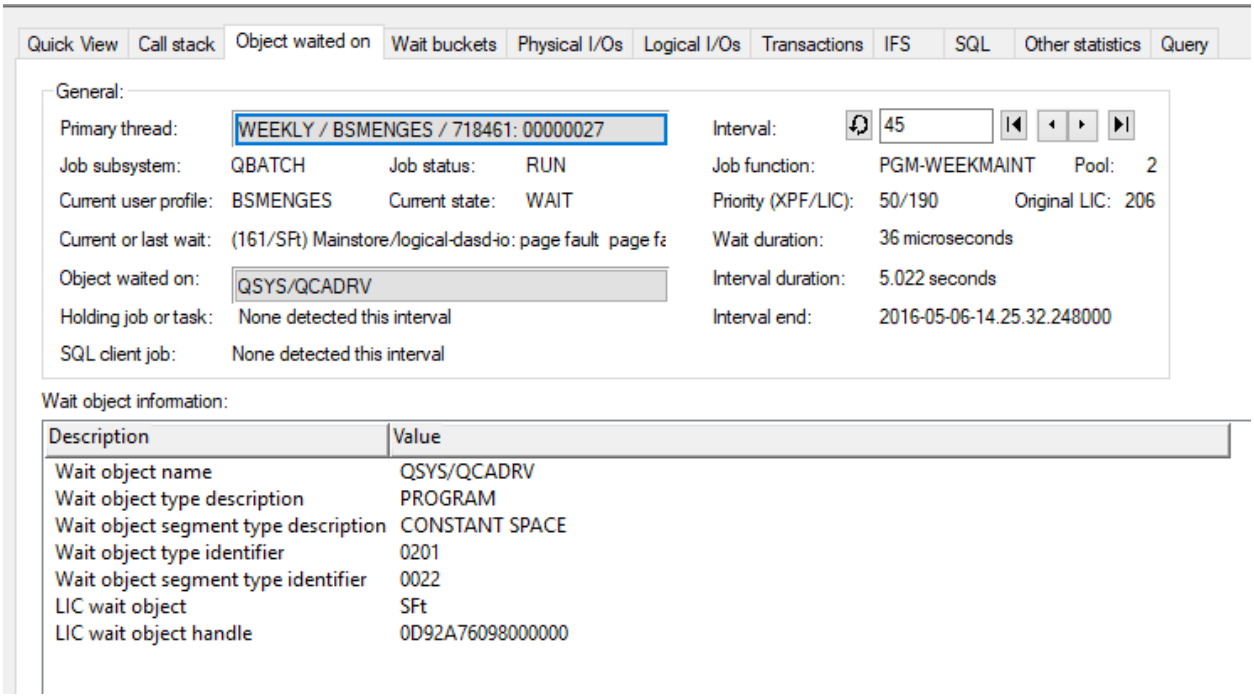
Library name (LIBNA...	Collection name (MBRNAME)	PGMLIB	Program name (PGMNAME)	MODLIB	Module name (MODNAME)	Procedure (PROCNA...	MODSRC	MODSRCLIB	MODSRCMBR	MODATR	MODCRT	OPTLVL	DEBUGDTA	LPP	PTF
QSYS	JWMON002	QSYS	QDBCRTME	QBUILDSS1	QDBCRTME	AOAP					1170628130120	30	*NO	5770SS1V7R2M0	

Module details

8.29.3 Object Waited on

This page provides extra details about the object waited on not shown in the general section.

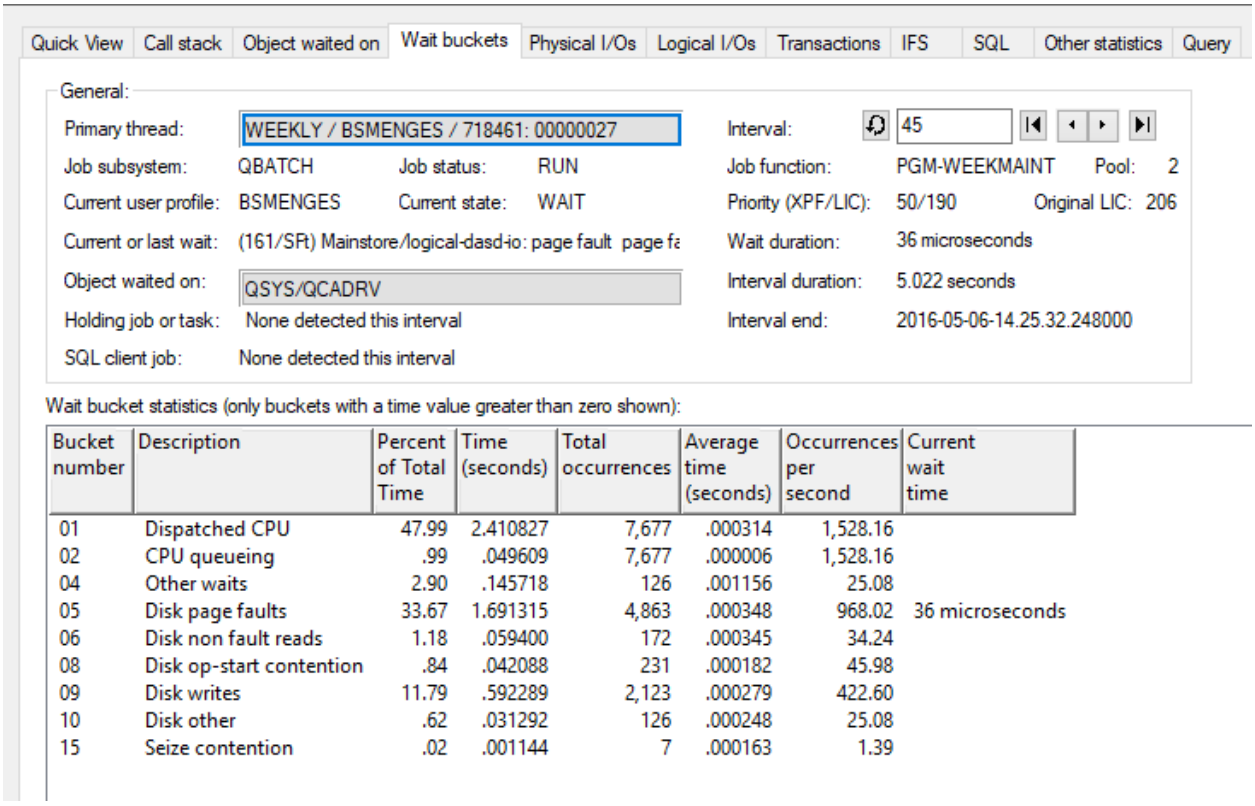
This information includes the Object (and segment) type descriptions, type identifiers and LIC wait object handle.



Interval Details – Object Waited on

8.29.4 Wait Buckets

The wait buckets tab displays a breakdown of all the wait types that occurred during the thread's interval. The number of occurrences for each wait bucket and the avg duration is also provided.

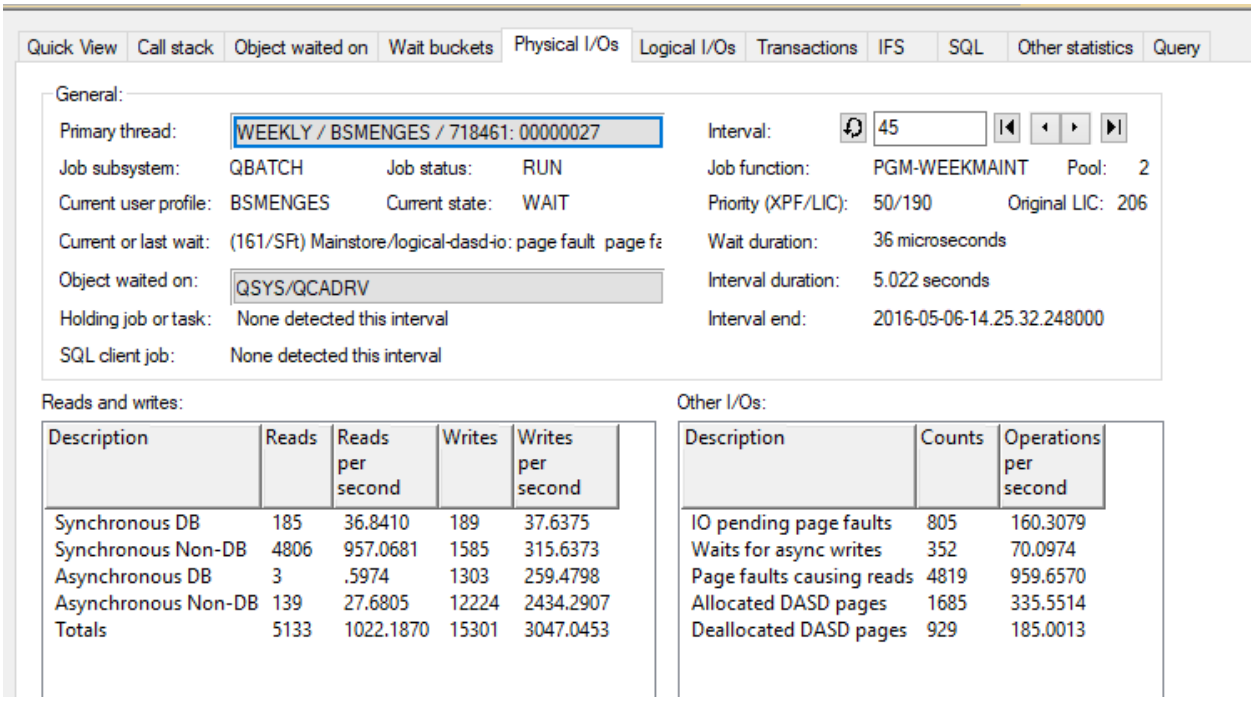


Interval Details – Wait buckets

The last column in the example above shows the current wait duration in the disk page faults. The value of 36 microseconds means it has been that long since the job used.

8.29.5 Physical I/Os

This tab contains additional metrics relating to physical I/Os, pages allocated and page faults.



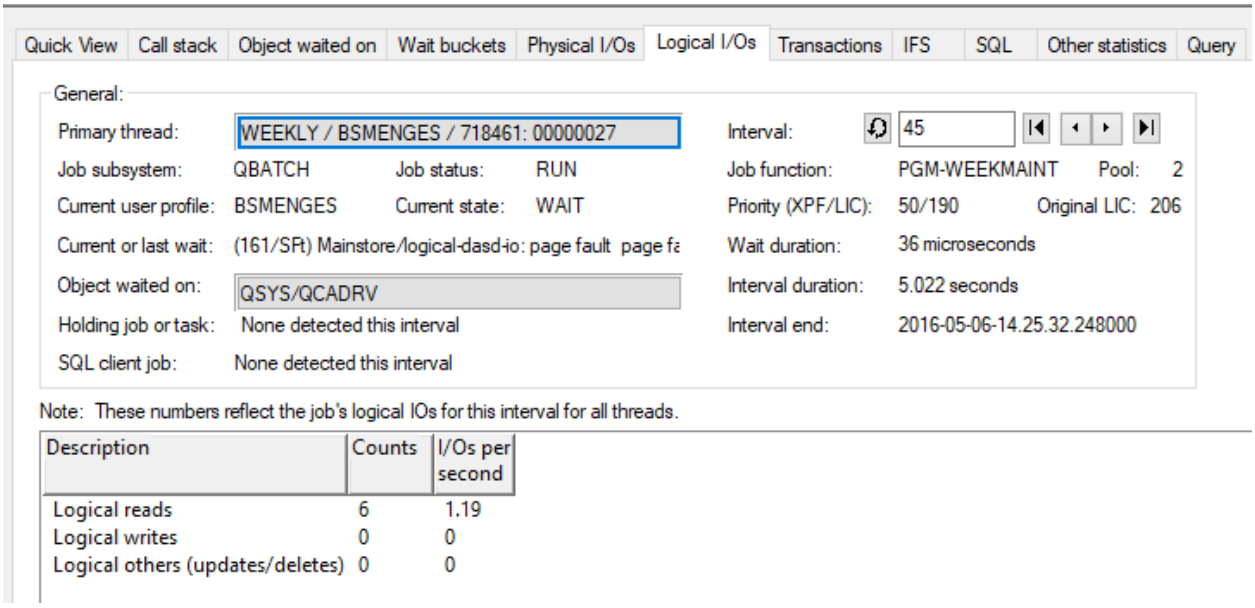
Description	Reads	Reads per second	Writes	Writes per second
Synchronous DB	185	36.8410	189	37.6375
Synchronous Non-DB	4806	957.0681	1585	315.6373
Asynchronous DB	3	.5974	1303	259.4798
Asynchronous Non-DB	139	27.6805	12224	2434.2907
Totals	5133	1022.1870	15301	3047.0453

Description	Counts	Operations per second
IO pending page faults	805	160.3079
Waits for async writes	352	70.0974
Page faults causing reads	4819	959.6570
Allocated DASD pages	1685	335.5514
Deallocated DASD pages	929	185.0013

Interval Details -> Physical I/Os

8.29.6 Logical I/Os

This tab contains metrics relating to Logical database operations for the current job in the interval.

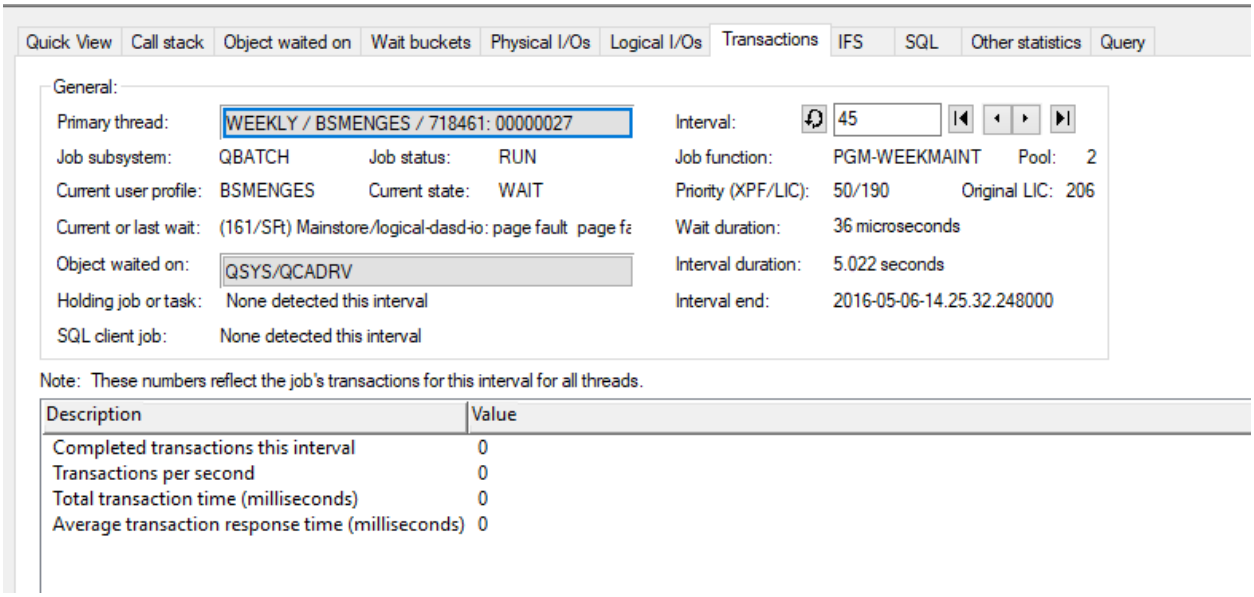


Description	Counts	I/Os per second
Logical reads	6	1.19
Logical writes	0	0
Logical others (updates/deletes)	0	0

Interval Details -> Logical I/Os

8.29.7 Transactions

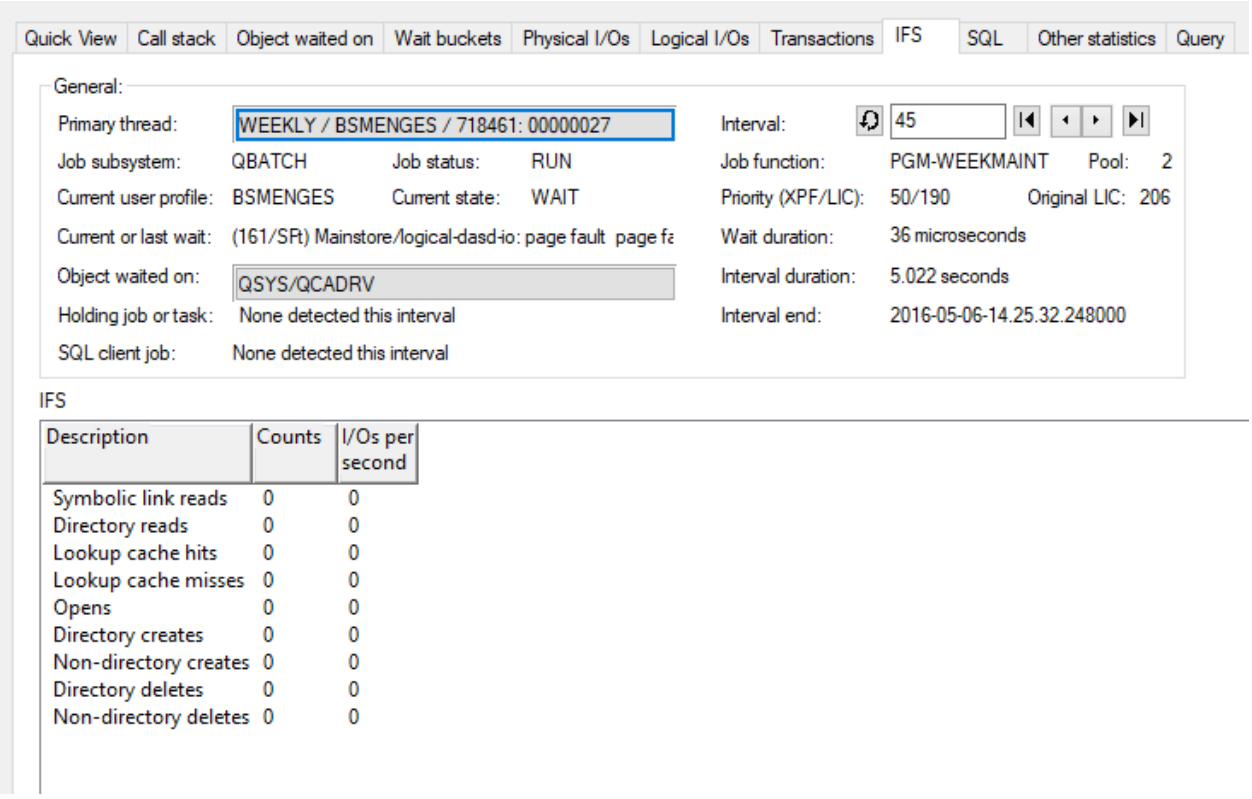
This panel show 5250 display transaction statistics for the current job in the current interval. These metrics apply to all threads in the current job shown.



Interval Details -> Transactions

8.29.8 IFS

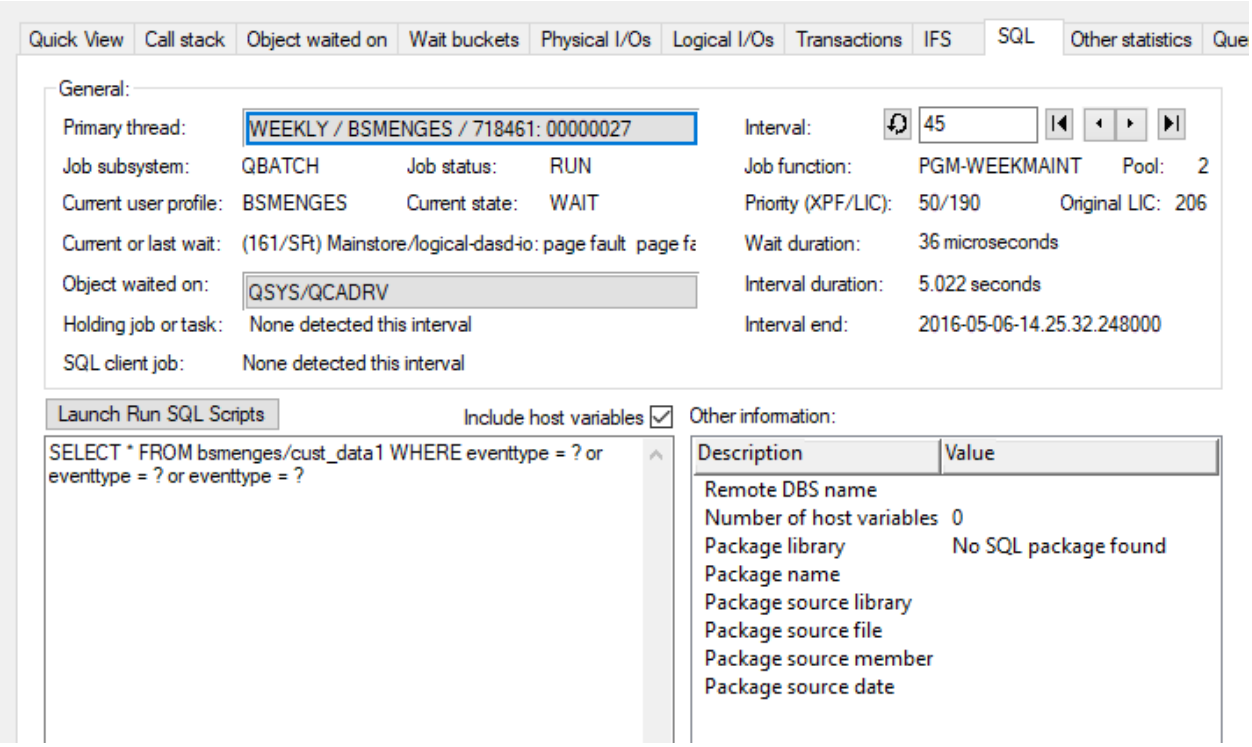
This panel shows the IFS statistics for the current job in the current interval.



Interval Details -> IFS

8.29.9 SQL

The SQL tab displays information about any SQL statements that were running in the job at the end of the interval.



Interval Details – SQL

The SQL statement(s) if any are found are shown within the textbox in the bottom left side of the window.

Host variables will often (but not always) be collected separately in the SQL data and iDoctor will attempt to parse them back into the SQL statement where they belong. However, this is not always possible and sometimes these host variable values are not even given or in a format that is readable. Check or uncheck the **include host variables** option to enable or disable this option.

Also information about the SQL package is also provided on this panel if it is available.

The **Launch Run SQL Scripts** button can be used to open the **Run SQL Scripts** interface within **IBM i Navigator -or- IBM i Access Client Solutions** depending on which you have installed on your PC. From there you could use Visual Explain to analyze the performance of the query. Of course, if the host variables have not been parsed into the SQL statement, some tweaking of the SQL would be required.

9 Collection Services Investigator

This chapter provides an overview of the interfaces within the IBM iDoctor for IBM i - Collection Services Investigator component.

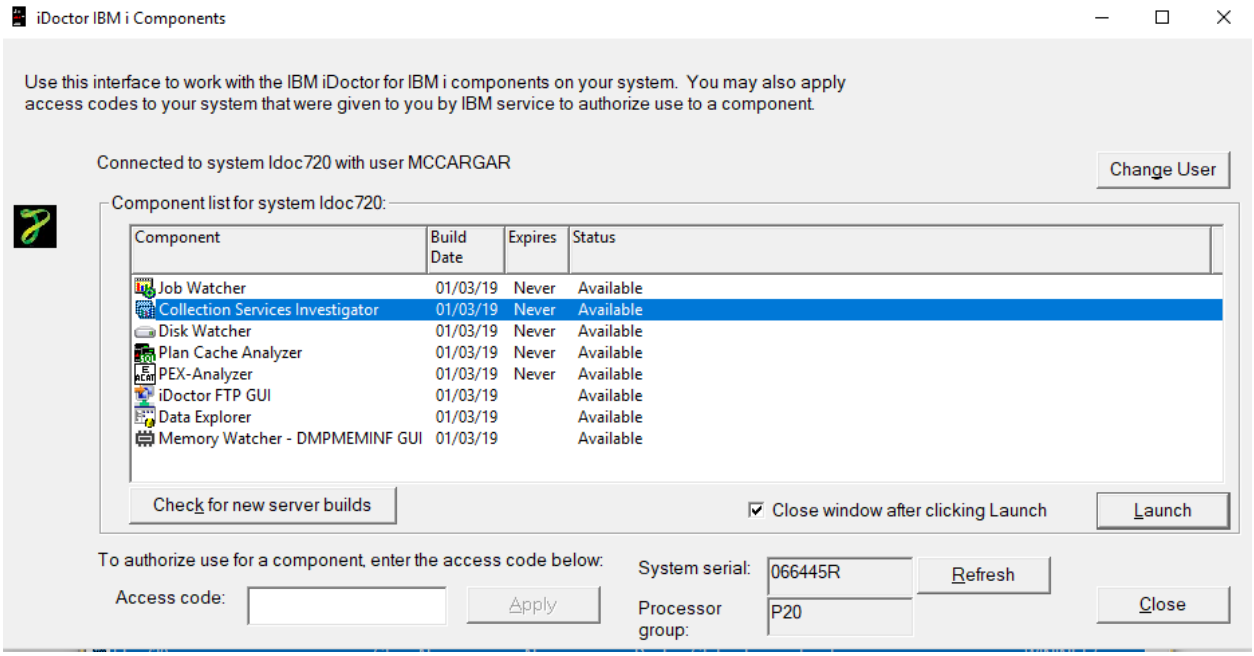
The Collection Services Investigator component provides interfaces designed to help the user analyze performance problems on IBM i using data collected by Collection Services.

Collection Services is shipped with IBM i and is typically ran 24x7.

9.1 Starting Collection Services Investigator

Collection Services Investigator is a component of the iDoctor suite of tools. iDoctor can be started using the Start menu: Start->IBM iDoctor for IBM i. Once the IBM iDoctor for IBM i application appears, the Collection Services Investigator component is started from the IBM i Connections View by double-clicking on the desired system. If there are none you will need to add one to the list first.

A list of available components will appear on the next window. Double-click on the Collection Services Investigator component or select Collection Services Investigator and click the Launch button in order to continue

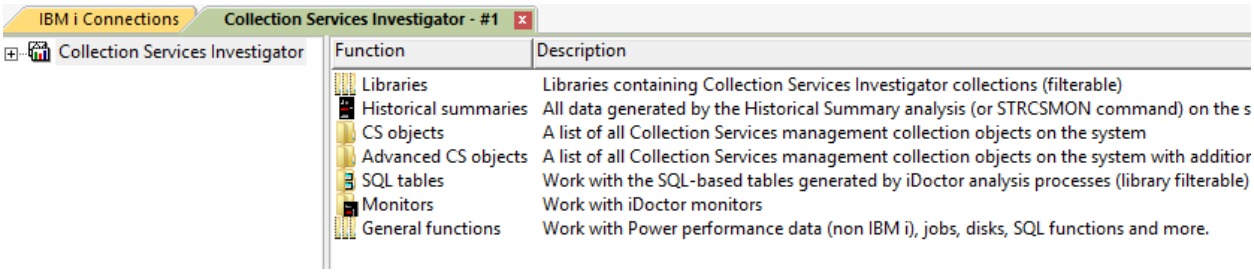


iDoctor IBM i Components Window

Note: Collection Services Investigator, Plan Cache Analyzer and Disk Watcher will only be available if Job Watcher is installed correctly and a valid access code for Job Watcher has been applied. These components are included with the Job Watcher license.

9.2 Collection Services Investigator Component View

The Collection Services Investigator view is the interface used to configure collection options, start new collections or monitors, or work with existing data.



Collection Services Investigator Component View

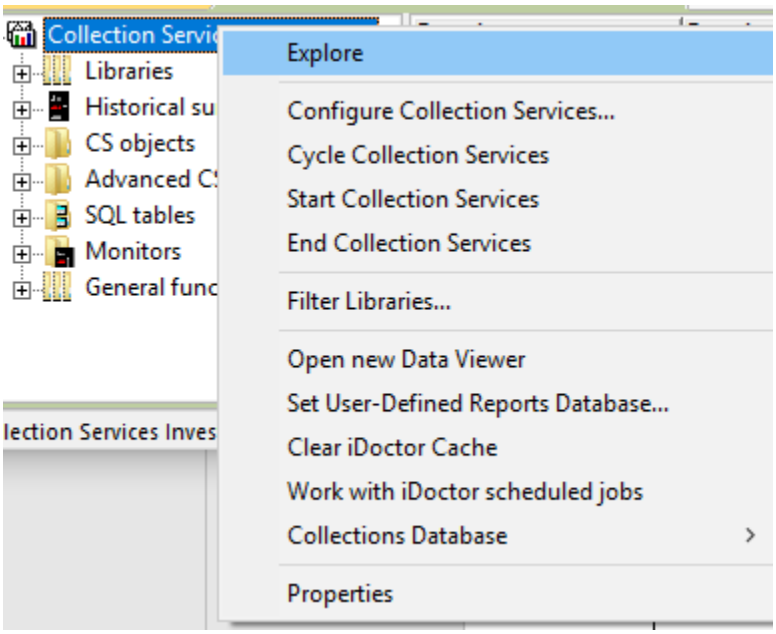
The ' Collection Services Investigator' folder contains a list of folders, each providing different features available. Collections can be displayed in various ways, either under the Libraries folder on a per library basis or as management collection objects *MGTCOL in the CS Objects or Advanced CS Objects folders.

A special type of summarized analysis over Collection Services data is provided within the Historical Summaries folder. This provides the ability to graph data over weeks/months.

The General functions folder contains several additional options for working with the IFS, browsing objects, working with the disk units and ASPs or working with any non-IBM i data (VIOS/HMC/AIX) that has been collected and moved to this system. The options for working with non-IBM i data are stored in the General functions -> Power folder.

9.2.1 Menu Options

The Collection Services Investigator Icon (when right-clicked) provides following menu options:



Collection Services Investigator folder menu options

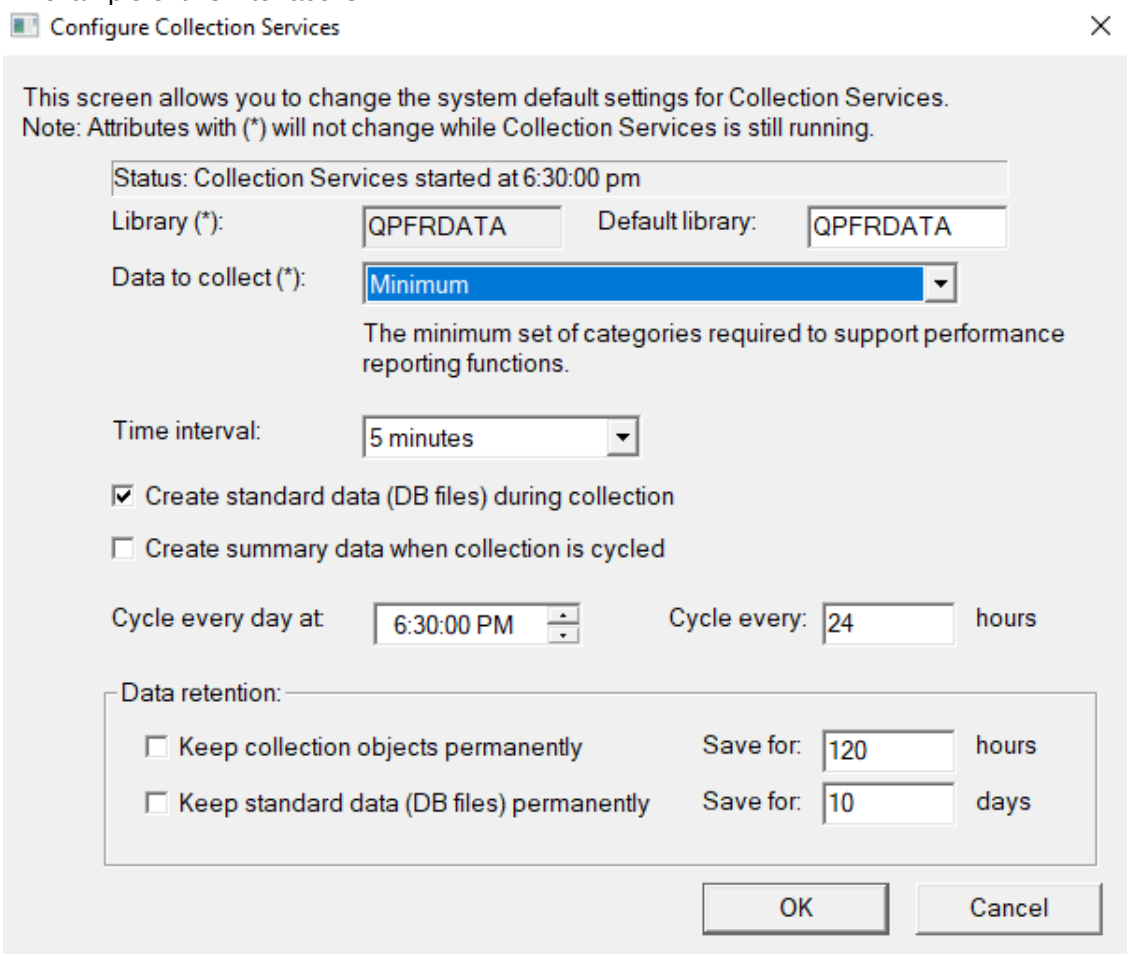
Menu Item	Description
Configure Collection Services	Displays the Configure Collection Services window where you may change settings for collecting data with Collection Services.
Cycle Collection Services	This option will end the current Collection Services collection and start another one.
Start Collection Services	This option will start a new Collection Services collection if it is not already running.
End Collection Services	This option will end the currently active Collection Services collection.
Filter libraries…	This option allows you to filter the libraries shown in the Libraries, CS Objects, Advanced CS Objects and SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of these folders if the system contains many libraries containing Collection Services data.

Additional menu options that are provided with all components are also included.

9.2.2 Configure Collection Services

This interface allows the user to change the system default settings for Collection Services data capturing. Some of the settings cannot change while Collection Services is running. It must be stopped and restarted for the changes to take effect and be visible on this window.

An example of this interface is:

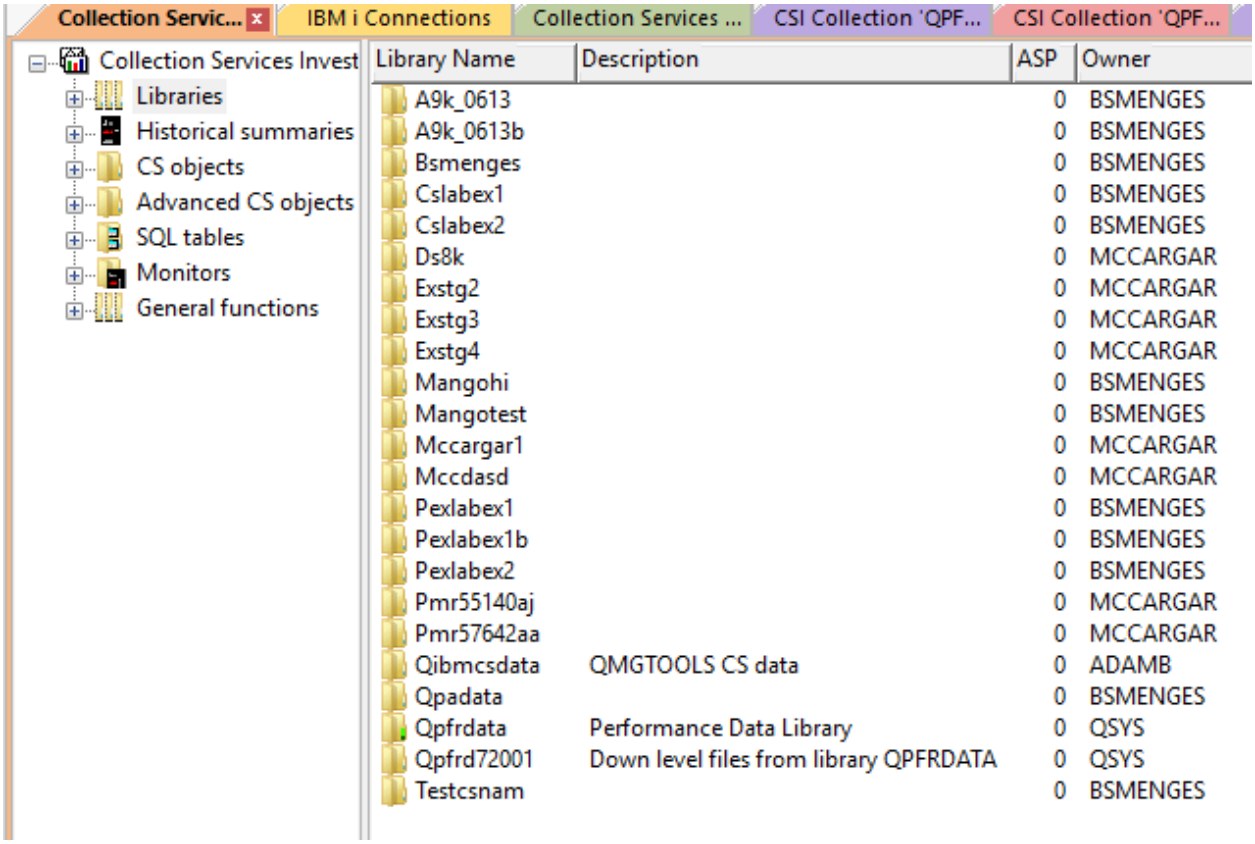


Configure Collection Services

Option	Description
Status	Displays if Collection Services is active or not and when it was last started.
Library	The library where Collection Services data is currently being collected in (if active). This setting cannot be changed by the user.
Default library	The library where Collection Services data will reside when it is next restarted.
Data to collect	These are data collection categories for Collection Services. The default setting will be pre-selected. The options available are: **Minimum:** The minimum set of categories required. **Standard:** All categories that are typically used for performance reporting except for communications protocol related data. **Standard plus communications:** All categories that are typically used for performance reporting. **Enhanced capacity planning:** All categories that are typically used for performance reporting plus PEX processor efficiency data. **Custom:** Customized by the user. This option must be defined using the Configure Collection Services task within IBM Navigator for i.
Time interval	The time interval indicates the duration of each element captured by Collection Services.
Create standard data (DB files) during collection	If checked, the Collection Services DB files (named QAPM*) will be created along with the *MGTCOL object.
Create summary data when collection is cycled	If checked, the Collection Services summary data files will be created.
Cycle every day at	Indicates the time of day when a new collection will start each day.
Cycle hours	The number of hours between cycles (a new collection starts.)
Keep collection objects permanently	Check this box if you don't want the system to delete your Collection Services *MGTCOL objects. You can also specify how many hours they should be retained.
Keep standard data (DB files) permanently	Check the box if you don't want the system to automatically delete your Collection Services DB files. You may also specify how many days the data should be retained.

9.3 Libraries

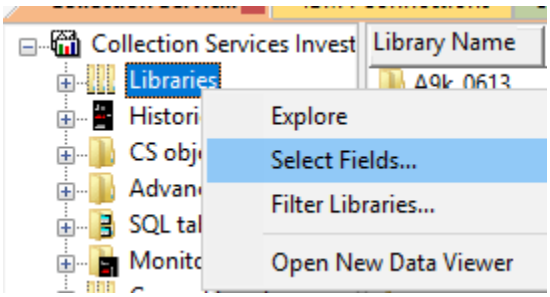
This folder contains the libraries on the system that contain Collection Services DB files (specifically the libraries containing file QAPMCONF). The list displays each library's name and description. By expanding on a library in the tree you will see the collection(s) that exist in the library.



Collection Services Investigator – Libraries Folder

9.3.1 Menu Options

Right-clicking on the Libraries folder provides these options:

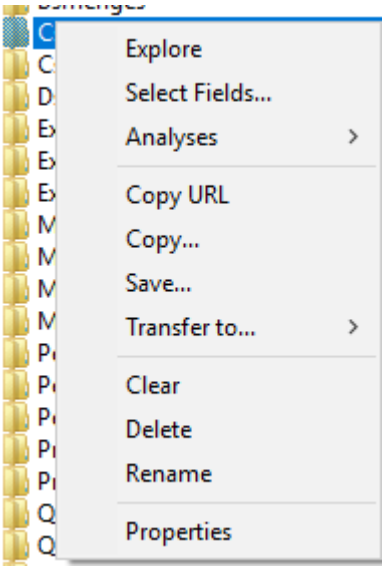


Libraries folder menu options

Menu	Description
Explore	Shows the list of libraries on the system that contain Collection Services data.
Select fields...	Displays the Field Selection Window. This allows you to configure and reorder the fields that are displayed when showing the list of libraries within the **Libraries** folder.
Filter libraries…	This option allows you to filter the libraries shown in the Libraries, CS Objects, Advanced CS Objects and SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of these folders if the system contains many libraries containing Collection Services data.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.

9.3.2 Library Menu Options

The following Collection Services Investigator specific menu options are available by right-clicking on a library in the component view.



Library CSlabex1 menu options

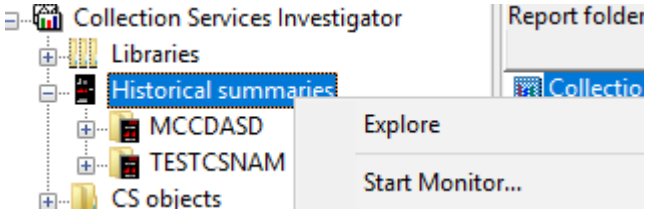
The menu options that are common to all library folders in iDoctor (including all the above shown) are discussed here.

9.4 Historical Summaries

The Historical Summaries folder displays Collection Services data that has been summarized into time periods larger than a single day (weeks/months.) This data is stored in a library where only one Historical Summary collection can exist in a library. Therefore, each entry in the Historical Summaries folder is a library on the system that contains this type of data.

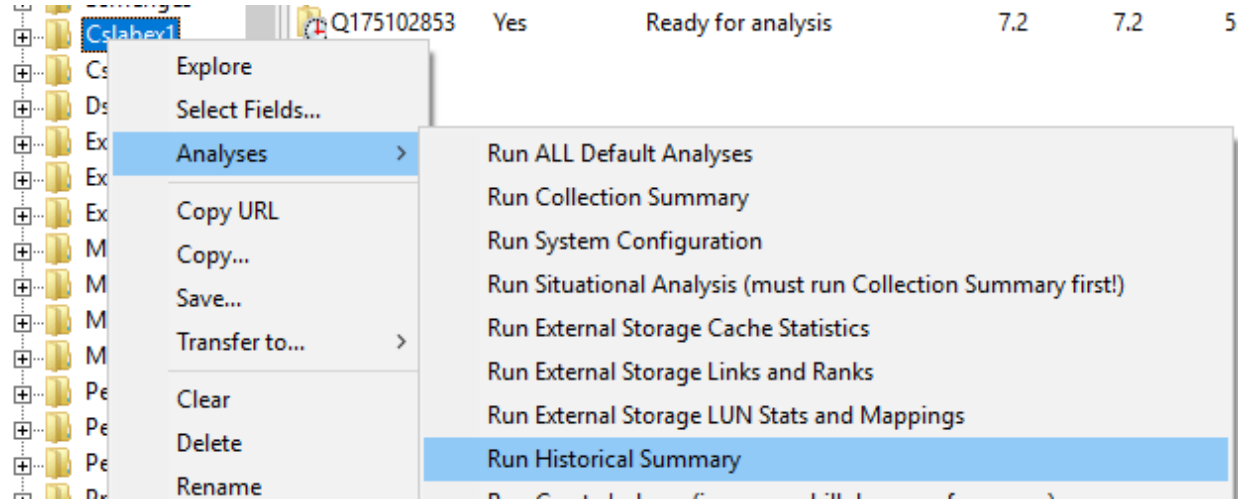
Data can be created in two possible ways:

1. Right-click the Historical summaries folder and click the Start Monitor… menu option and submit the job. The data will be summarized every day at the specified time until the job ends. You will not be able to graph any data until after the 1st summarization completes (probably the next day.) **Note:** If you wish for this to continue after an IPL you will need to configure your system to resubmit the command.



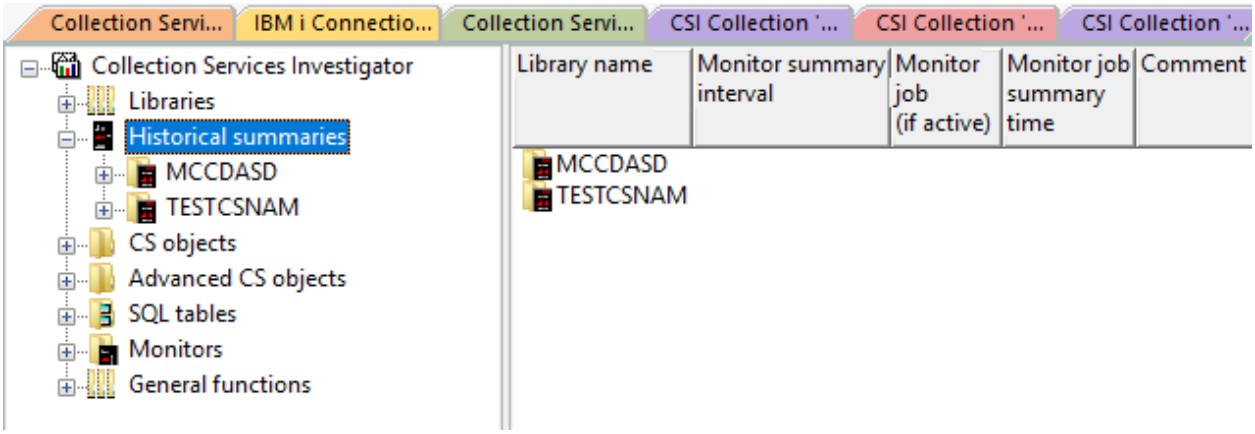
Historical Summaries – Start Monitor menu

2. Right-click a library and use the Analyses -> Run Historical Summary menu. This will summarize all Collection Services data within the specified library. Once the analysis has completed the graphs will be available.



Library menu – Analyses -> Run Historical Summary

After data has been created the Historical Summaries folder will contain an entry for each library on the current system that contains this type of data. Expanding the library will lead to graphing options very similar to those available when analyzing a single Collection Services collection.

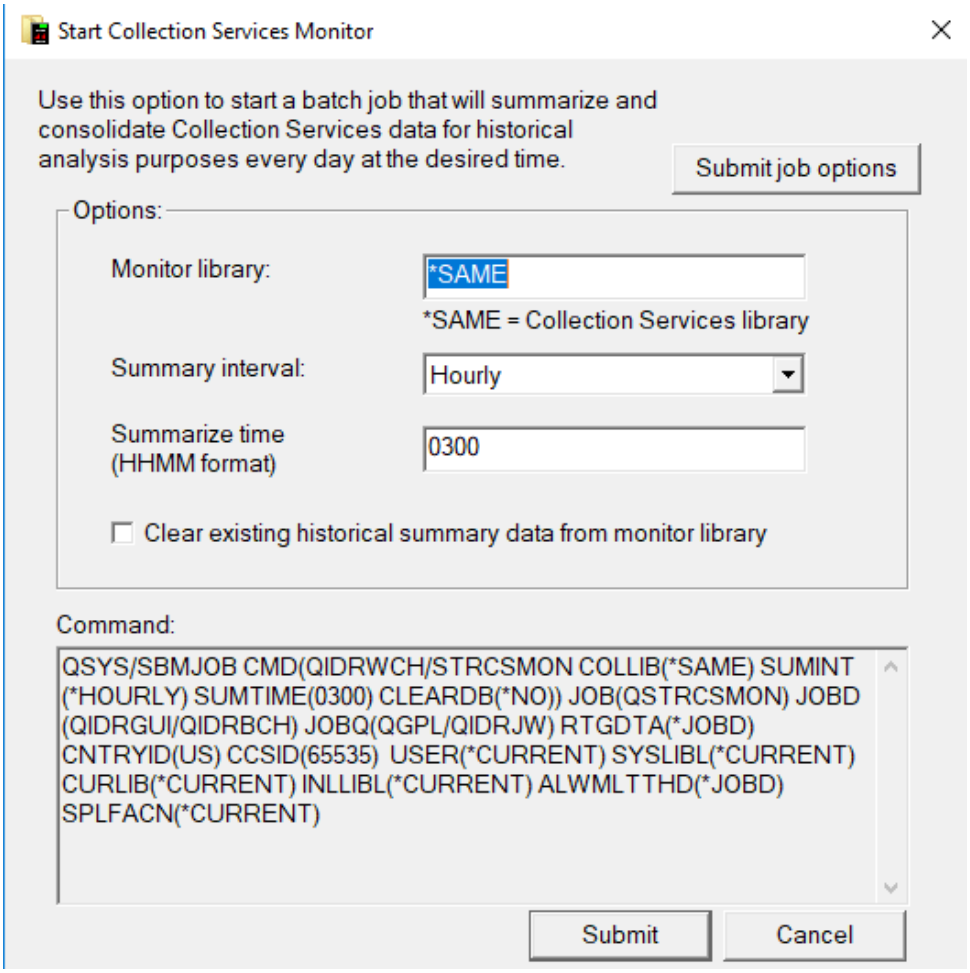


Collection Services Investigator – Historical Summaries Folder

9.4.1 Start Collection Services Monitor

Use this interface to start a batch job that will summarize and consolidate Collection Services data. The data will be created each day at the specified time and graphable from the Historical Summaries folder.

Tip: If the iDoctor Historical Summary database files have changed (see the update history notes), then it will be necessary to use the clear existing historical summary data option. Otherwise the Historical Summary analysis will fail to add new data for all files.

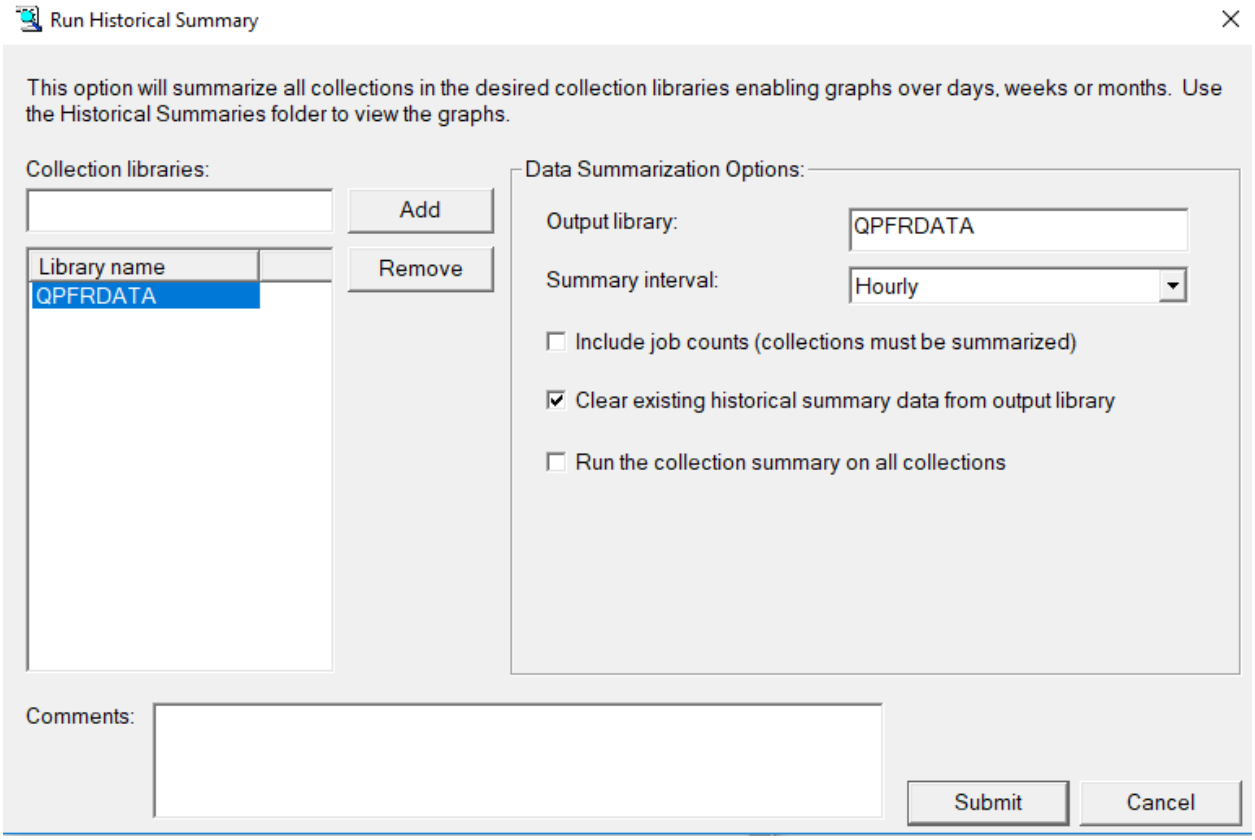


Start Collection Services Monitor

Option	Description
Monitor library	The library where Historical Summary data will be stored. It is recommended to use the *SAME value where *SAME is the default Collection Services data library. (i.e. QPFRDATA)
Summary interval	This value indicates the lowest level of detail to provide after the data is summarized. The possible values are 15 minutes, hourly, every 4 hours, every 8 hours, every 12 hours or daily.
Summarize time	This is the time of when the data will be summarized. It must be entered in HHMM format. (example: 1300 for 1 pm.)
Clear existing data	If checked any existing Historical Summary data that exists in the Monitor library specified will be removed.
Command	The command string shown is what the GUI will use to start the Collection Services Monitor.

9.4.2 Run Historical Summary

This is the interface shown when a user decides to summarize all existing Collection Services data in a library and build a Historical Summary collection. The window has similar options as those discussed in the previous section but the difference here is this process will only run once and not again over time.



Run Historical Summary

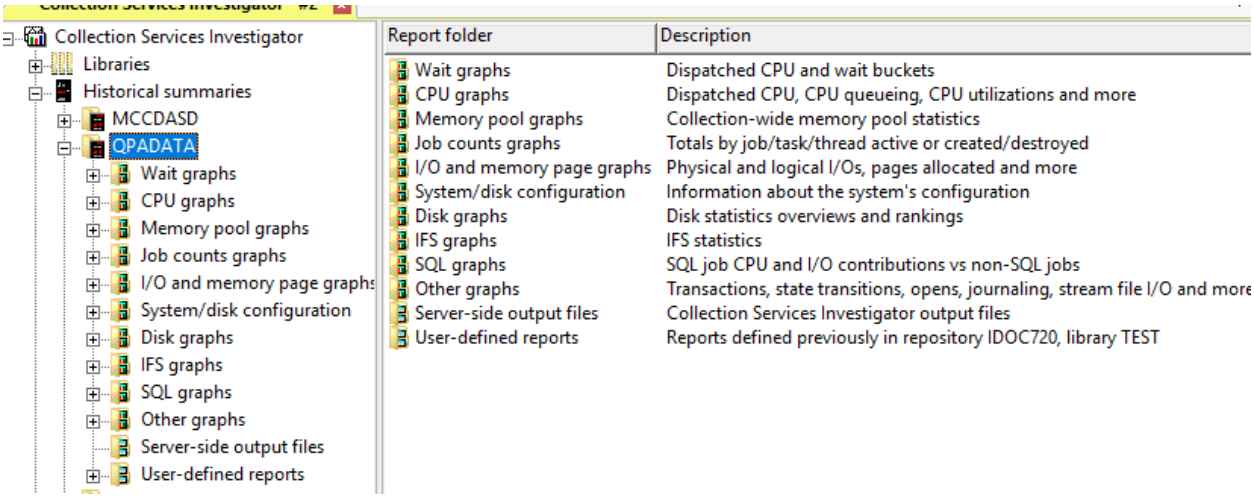
Option	Description
Collection libraries list	This list is used to indicate the library or libraries to include Collection Services data from. All collections in these libraries will be included and there is no option to filter this. If you wish to add a second library to the list, type the library name into the text box above the list and press the Add button.
Output library	The library where the Historical Summary data will be stored.
Summary interval	This value indicates the lowest level of detail to provide after the data is summarized. The possible values are 15 minutes, hourly, every 4 hours, every 8 hours, every 12 hours or daily.
Include job counts	Indicates if statistics for the job counts will be included. If you wish to use this option, then you must also check the “Run the collection summary on all collections” option as well unless this has already been done.
Clear existing data	If checked any existing Historical Summary data that exists in the output library specified will be removed.
Run the collection summary on all collections	This option will run the collection summary analysis first on all collections. The job counts statistics produced by the historical summary analysis requires this to be done or the analysis will fail!
Comments	Add a comment if desired and it will appear within the Historical Summaries folder.

9.4.3 Collections

A library in the historical summary folder acts as a collection of this type. By clicking the collection/library name you will see several analysis options.

These options are a subset of the graphs available under a normal Collection Services Investigator collection and are very similar. These graphs show high level statistics over time primarily and do not show thread/job rankings.

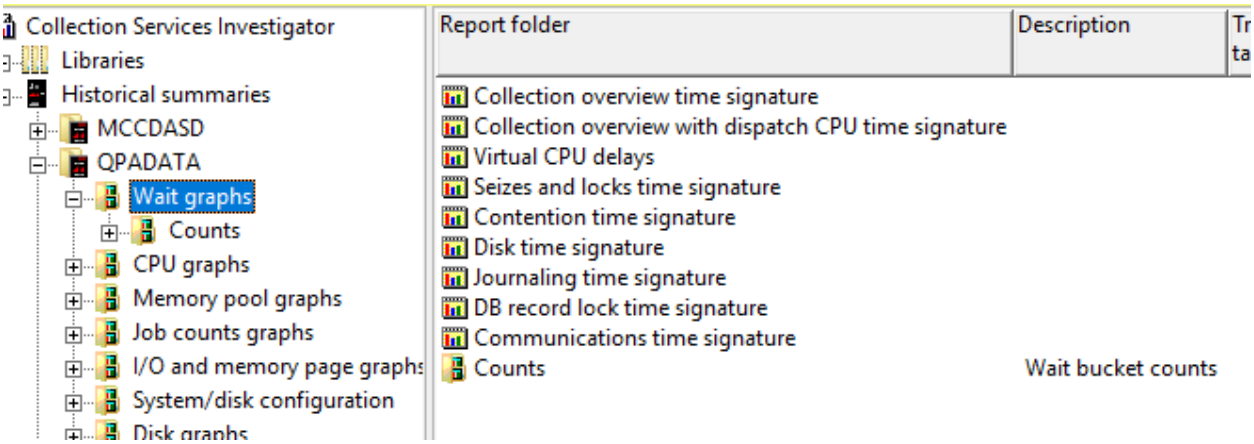
An example of the list of graphs available is:



Historical Summary collection QPADATA’s analysis options

9.4.4 Wait graphs

The Wait graphs folder for a Historical Summary collection contains the same set of overview graphs found in a regular Collection Services Investigator collection’s wait graphs folder.

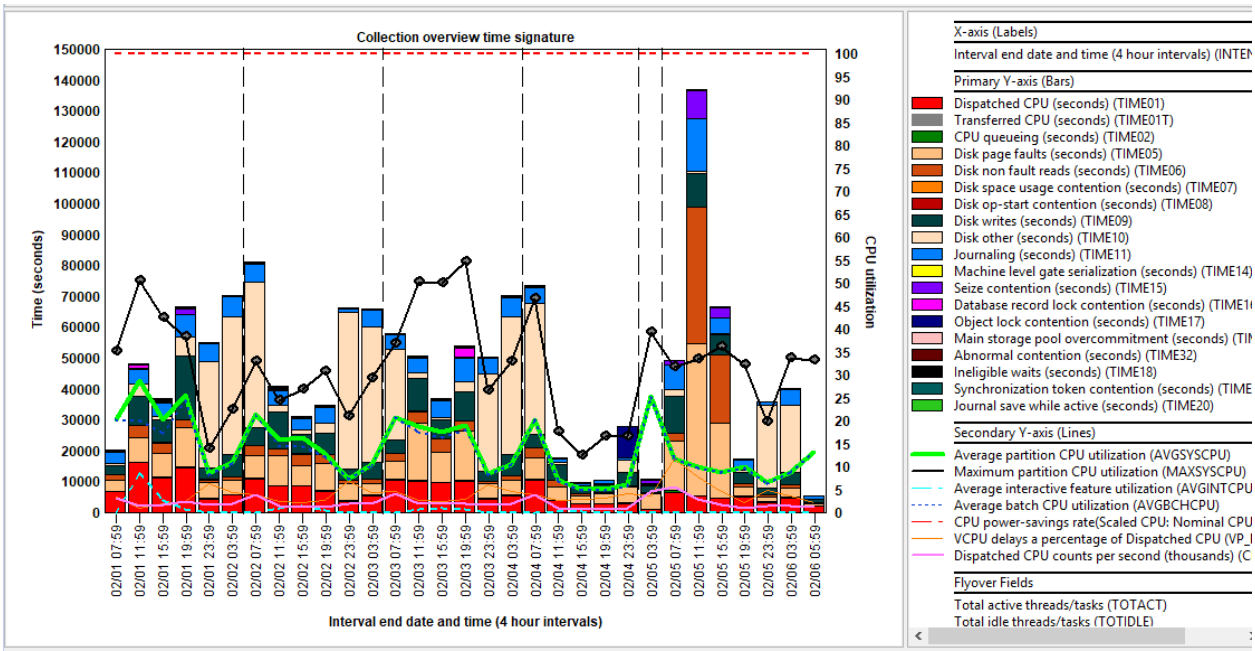


Wait graphs folder

9.4.4.1 Collection overview time signature

This overview graph shows the “interesting” wait bucket times and CPU utilization for all collections included in the analysis.

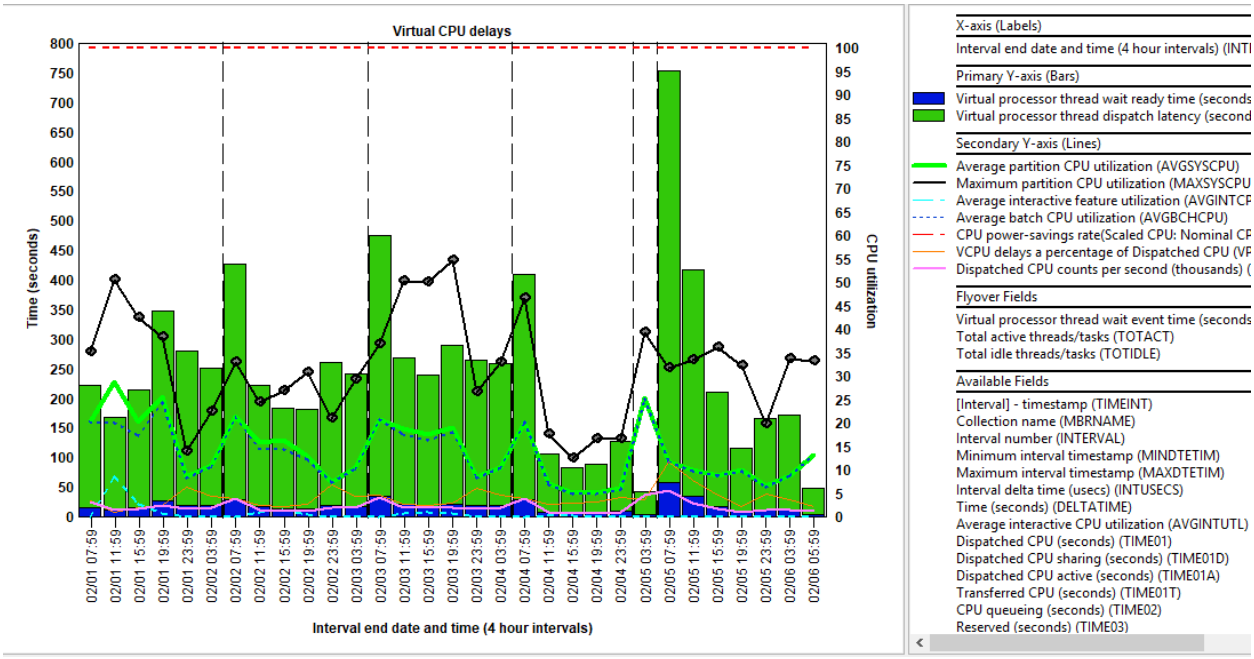
Note: Each vertical dashed line indicates the start of a new collection (this is usually but not always the start of a new day.)



Collection overview time signature

9.4.4.2 Virtual CPU delays

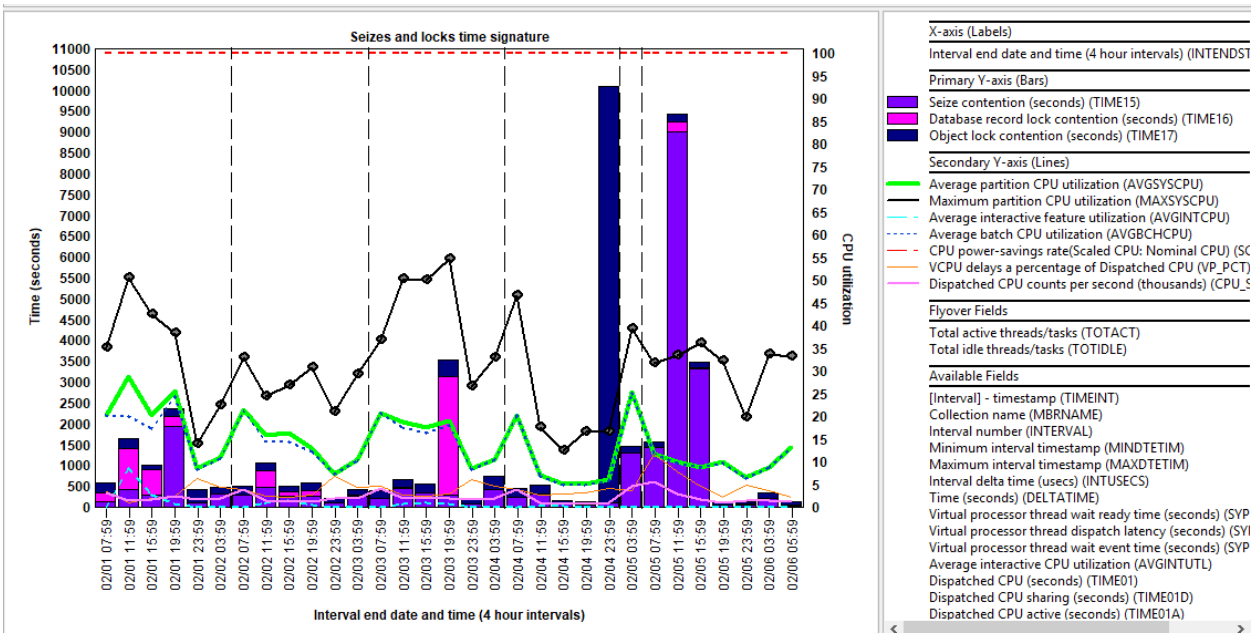
This graph provides a summarized look at the virtual processor thread wait ready and thread dispatch latency times across the entire LPAR.



Virtual CPU delays

9.4.4.3 Seizes and locks time signature

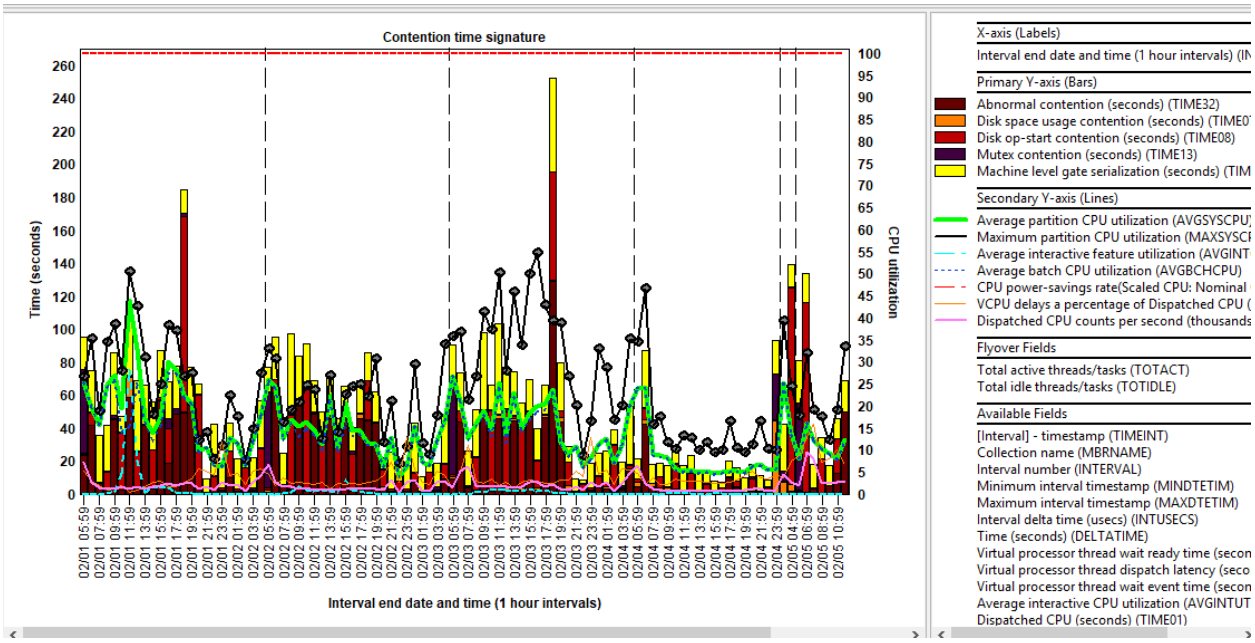
This graph displays seize contention, database record lock contention and object lock times.



Seizes and locks times signature

9.4.4.4 Contention time signature

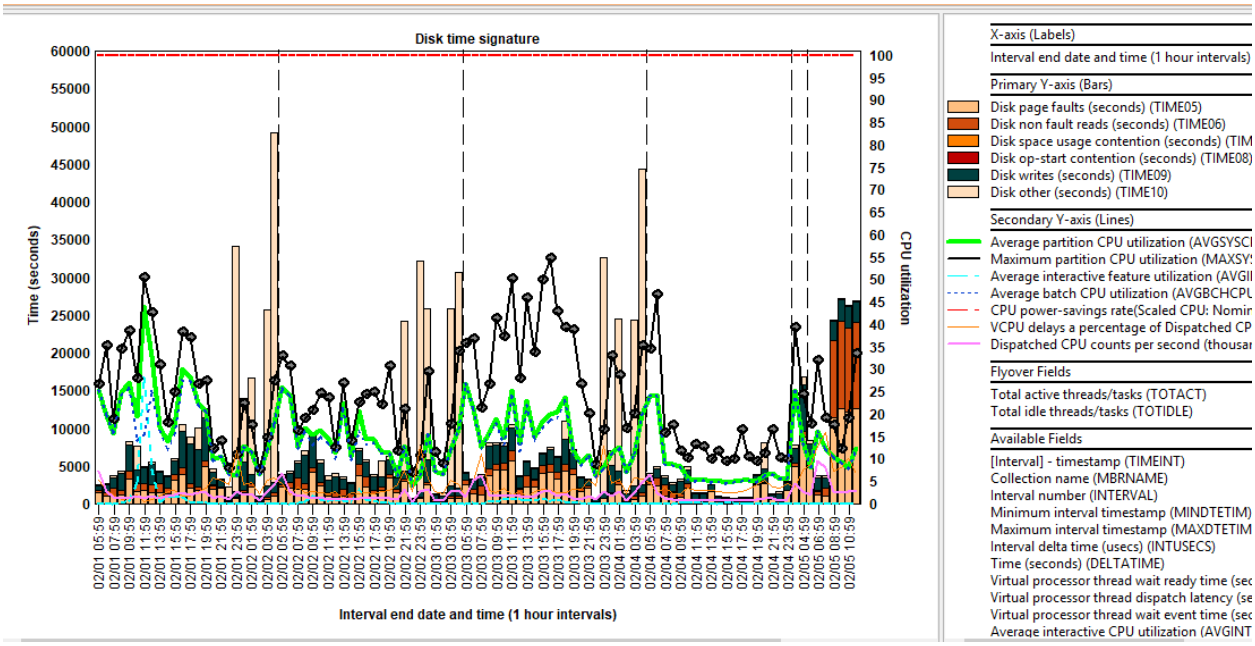
This graph displays abnormal contention, disk contentions, mutex contention as well as machine level gate serializations.



Contention time signature

9.4.4.5 Disk time signature

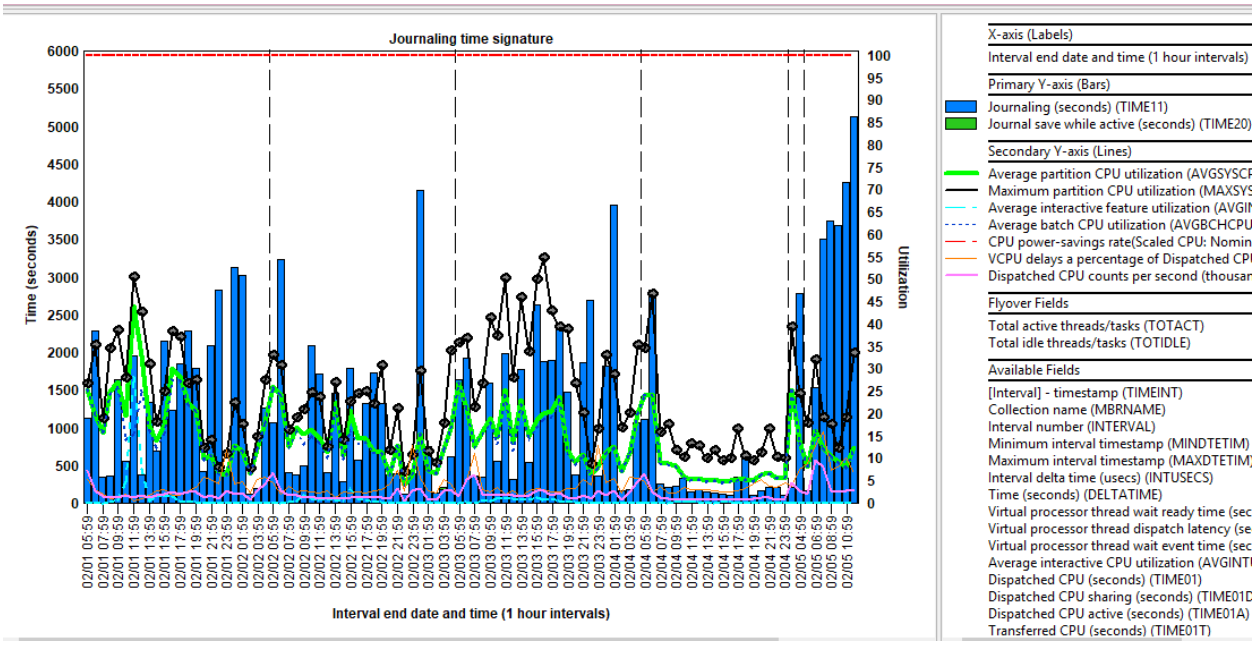
This graph displays the disk wait buckets over time.



Disk time signature

9.4.4.6 Journaling time signature

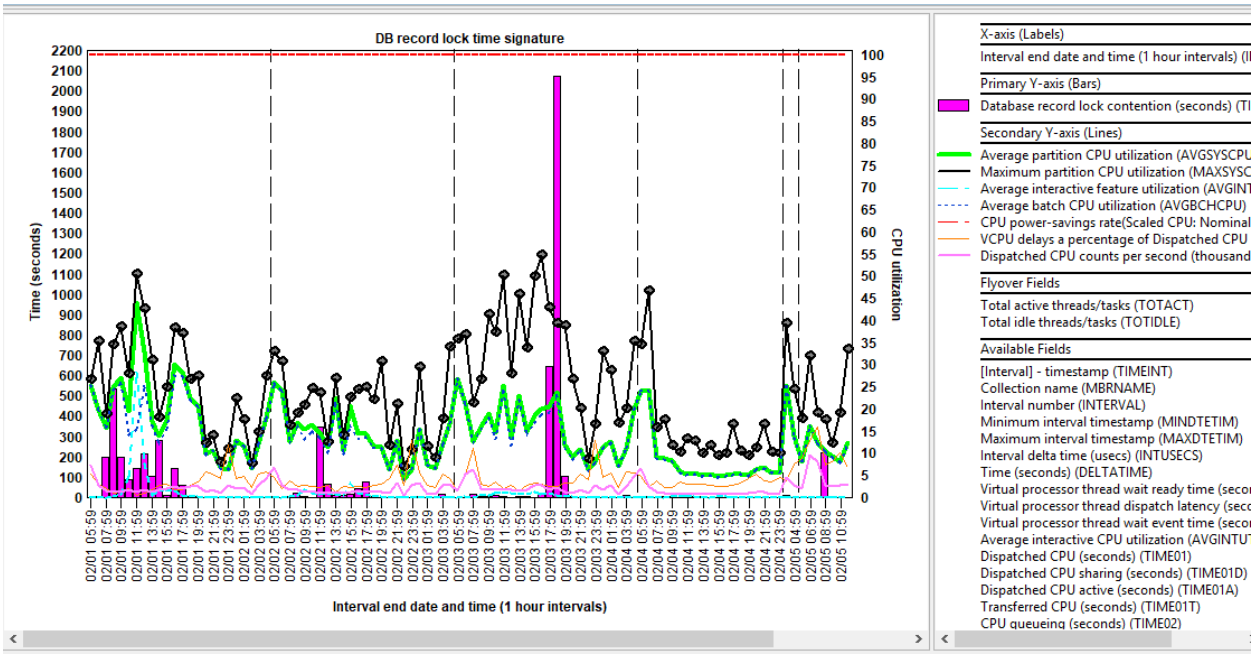
The journaling time signature graph just shows wait times related to journaling.



Journaling time signature

9.4.4.7 DB record lock time signature

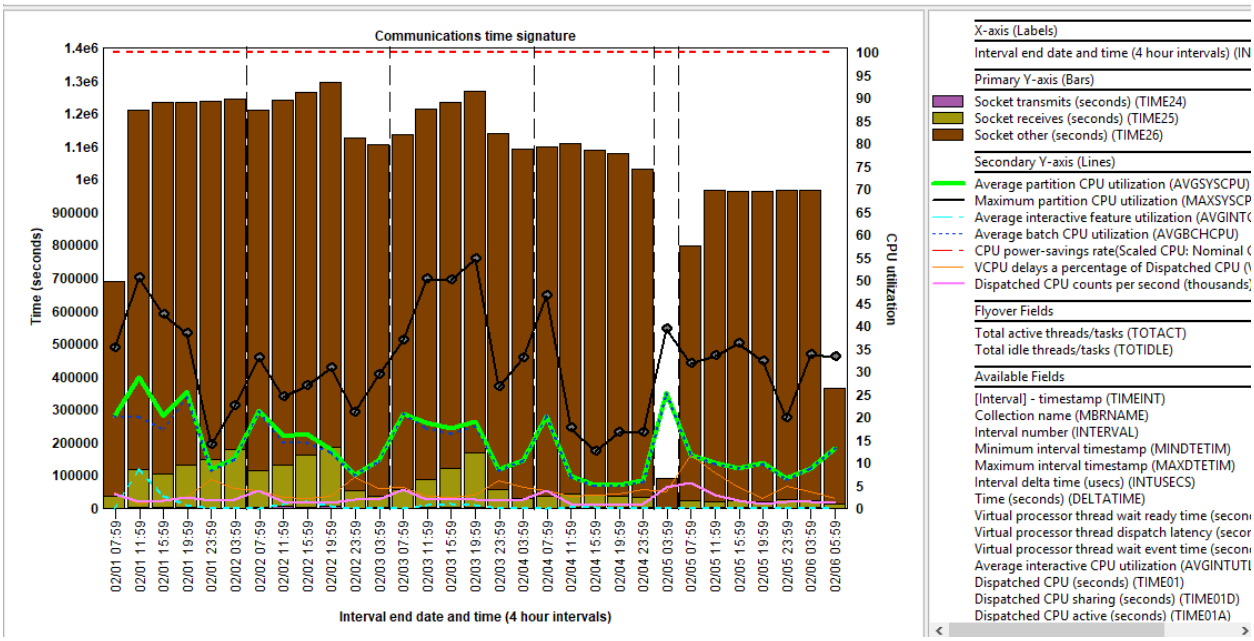
This graph just shows time relating to record locks.



DB record lock time signature

9.4.4.8 Communications time signature

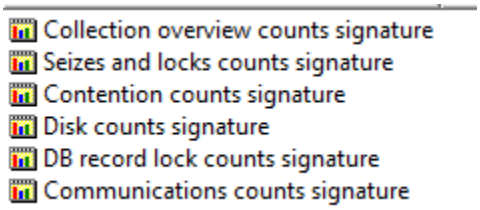
This graph shows the total time spent by jobs in socket communications types of waits.



Communications time signature

9.4.4.9 Counts

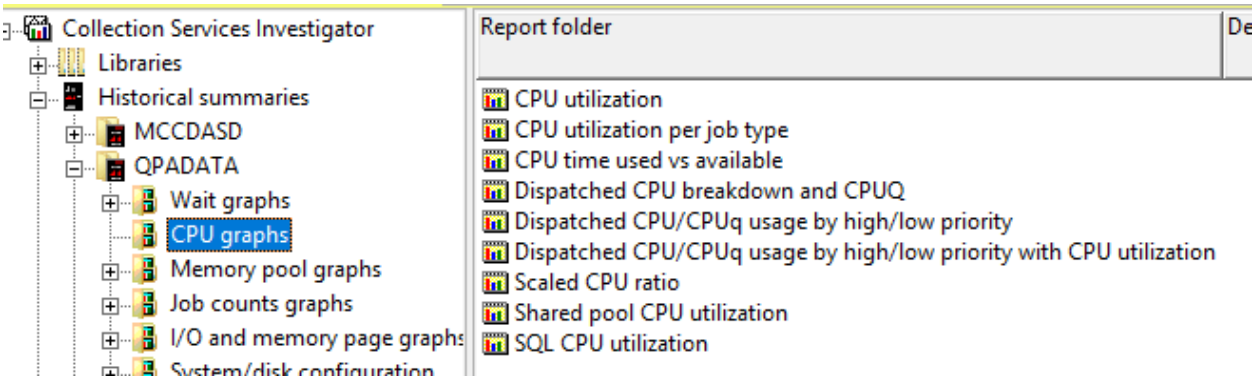
This folder contains graphs that show how many times each type of wait occurred instead of their time contributions.



Wait graphs -> Counts folder

9.4.5 CPU graphs

The CPU graphs folder for a Historical Summary collection contains a reduced set of overview graphs found in a regular Collection Services Investigator collection's CPU graphs folder. None of the Power 8 or higher or rankings graphs are currently available.

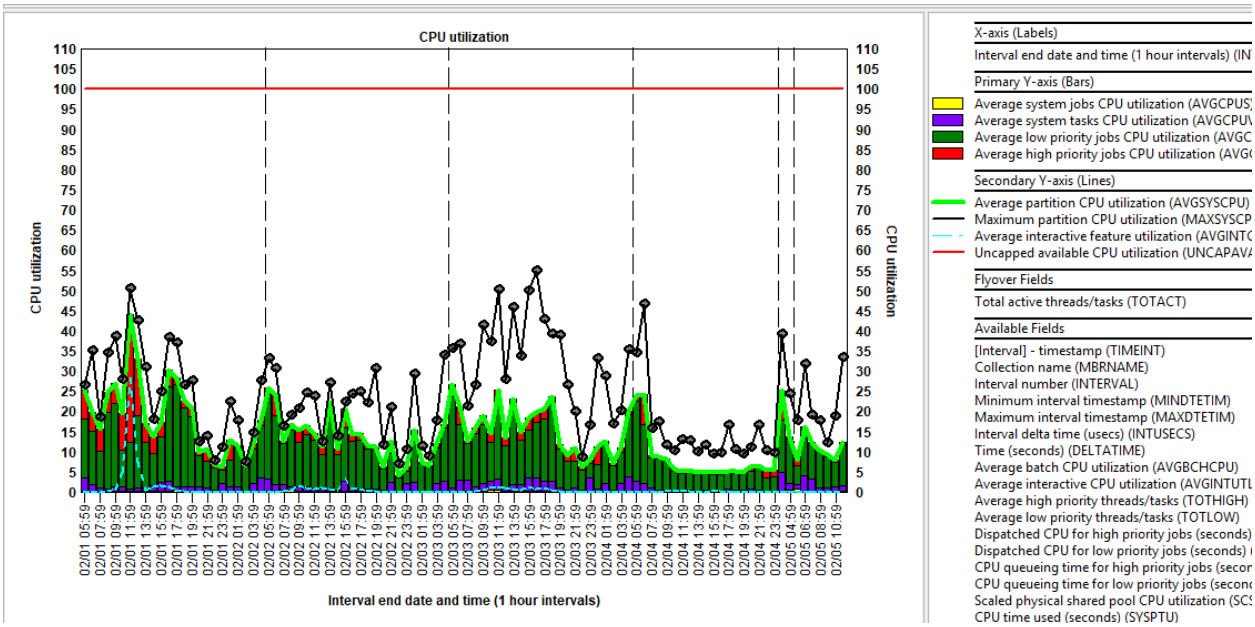


CPU graphs folder

9.4.5.1 CPU utilization

This graph breaks down the CPU utilization contributions into 4 different types of contributors:

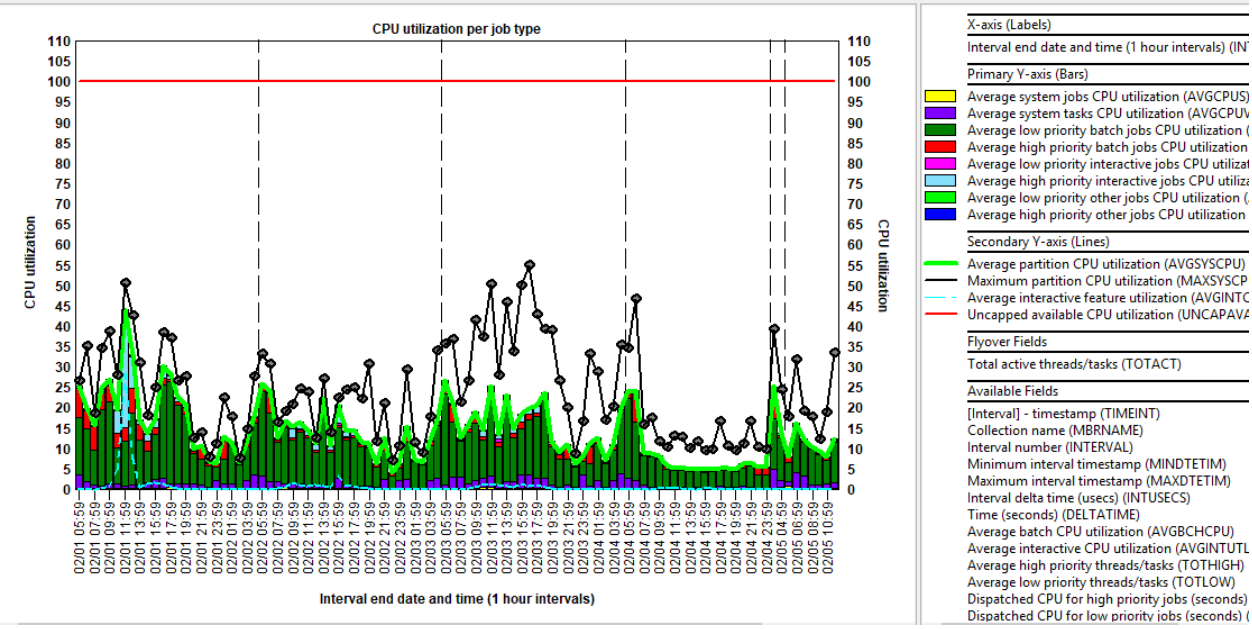
1. System jobs
2. System tasks
3. Low priority jobs (< 29)
4. High priority jobs (>= 29)



CPU utilization

9.4.5.2 CPU utilization per job type

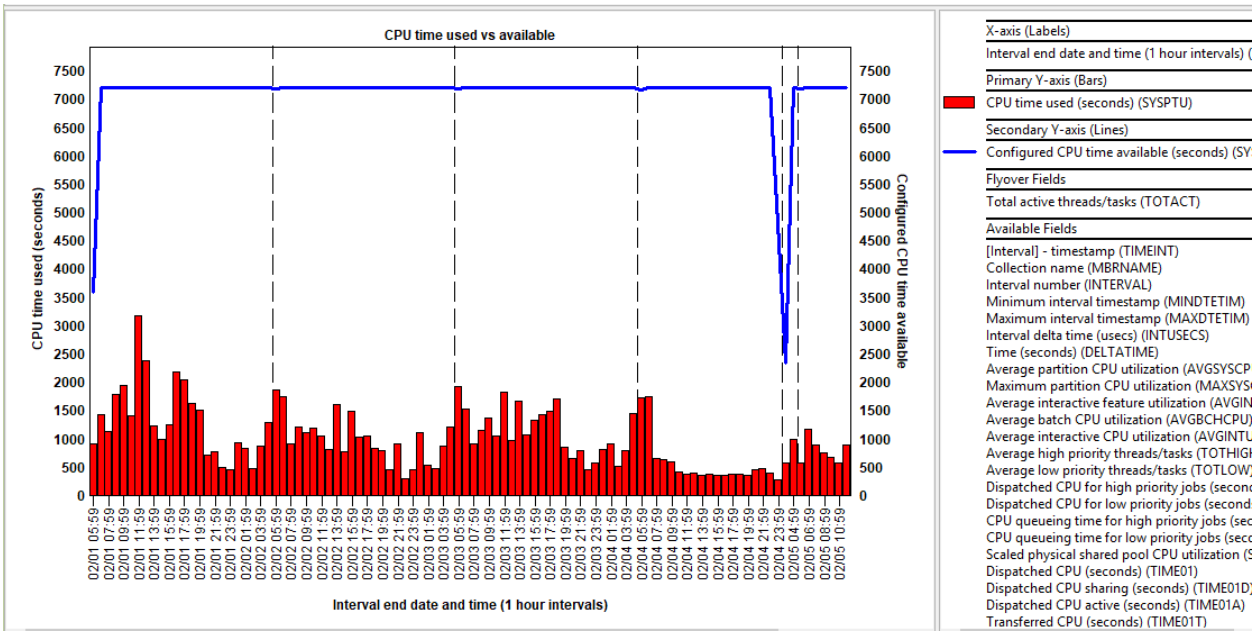
This graph is like the previous one by breaks it down into 8 categories.



CPU utilization per job type

9.4.5.3 CPU time used vs available

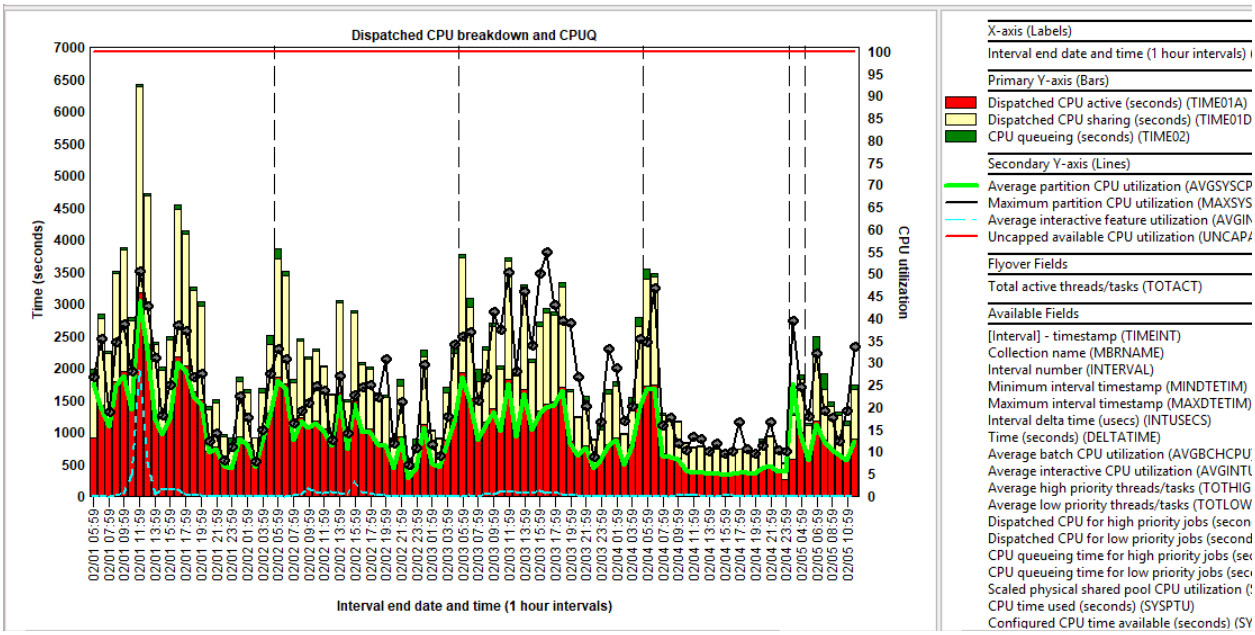
This graph compares CPU time used vs the total CPU time available on the Y2-axis.



CPU time used vs available

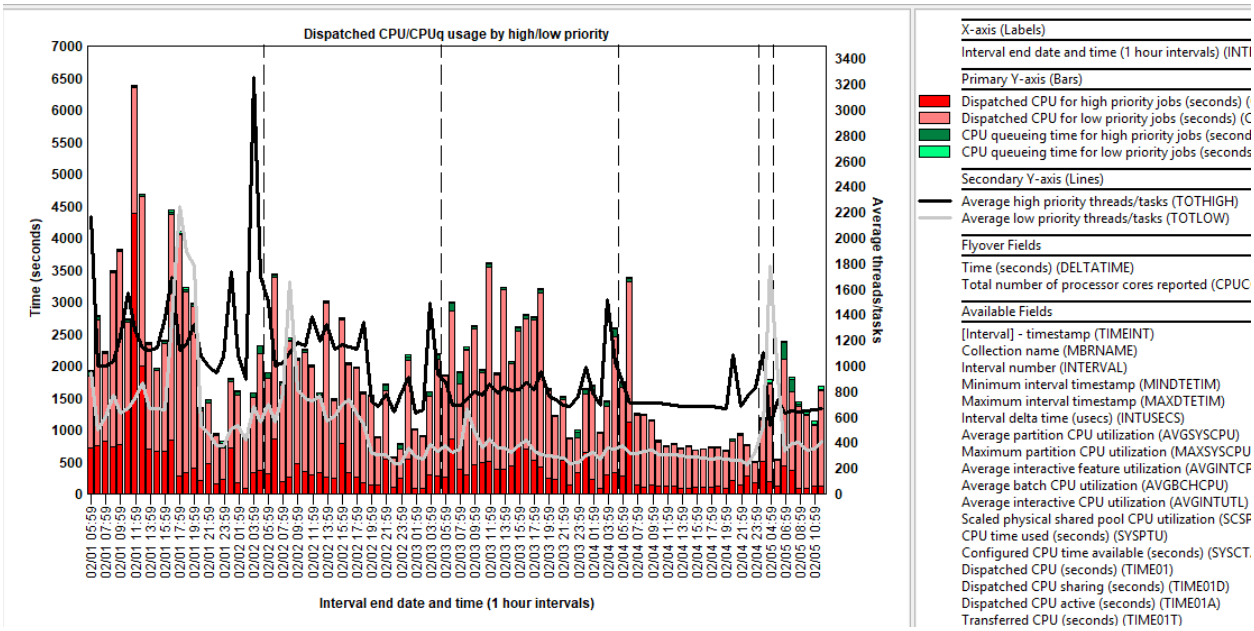
9.4.5.4 Dispatched CPU breakdown and CPUQ

This graph shows the contributions of wait bucket 1 (Dispatch CPU time) broken down into its 2 components as well as CPU queueing.



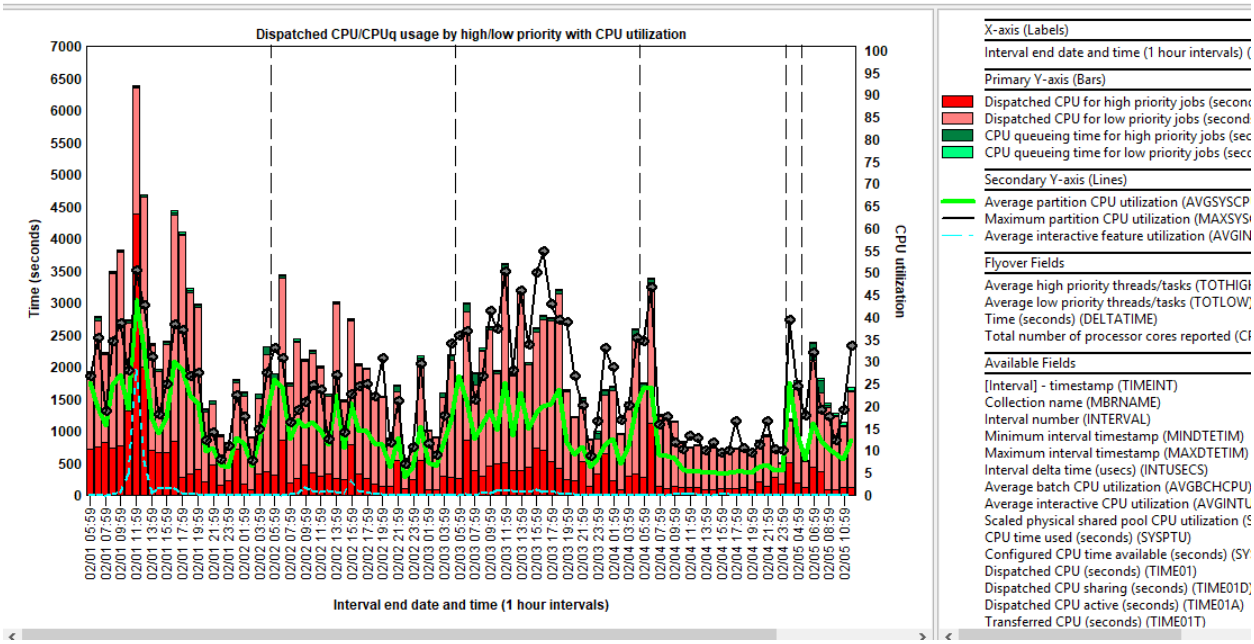
Dispatched CPU breakdown and CPUQ

9.4.5.5 Dispatched CPU/CPUq usage by high/low priority



Dispatched CPU/CPUQ usage by high/low priority

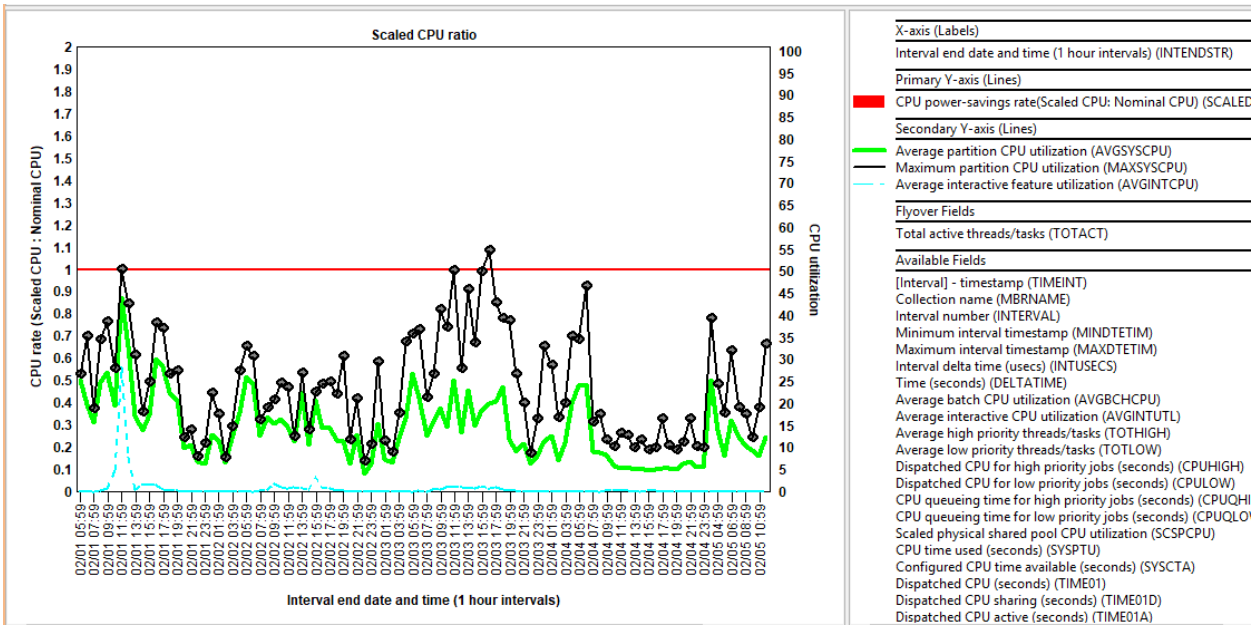
9.4.5.6 Dispatched CPU/CPUq usage by high/low priority with CPU utilization



Dispatched CPU/CPUq usage by high/low priority with CPU utilization

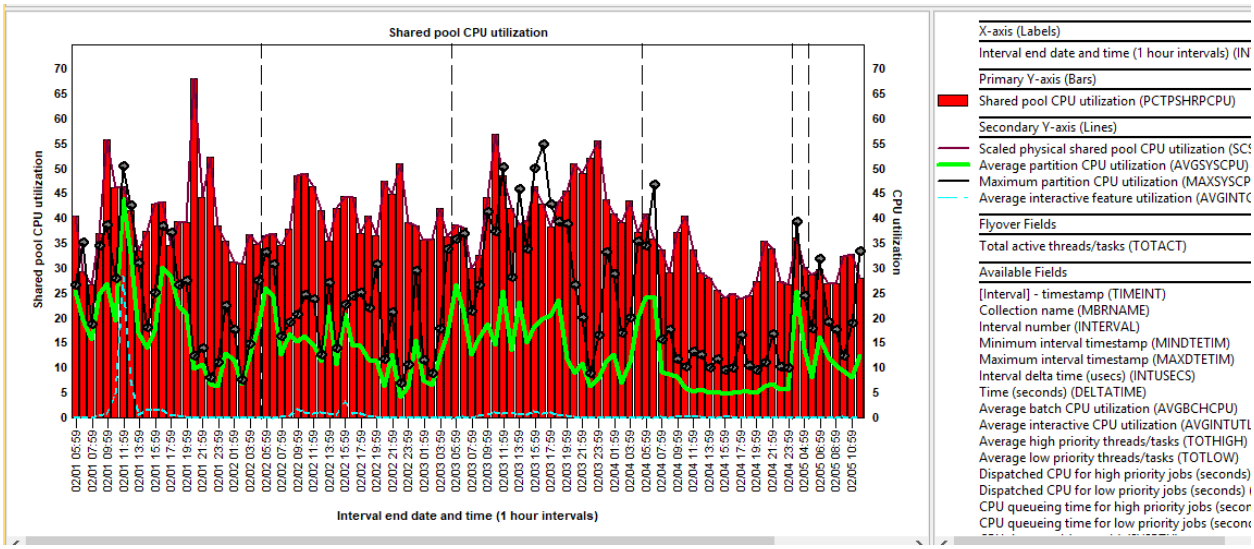
9.4.5.7 Scaled CPU ratio

This graph displays the CPU power-saving rate along with the CPU utilization on the Y2-axis.



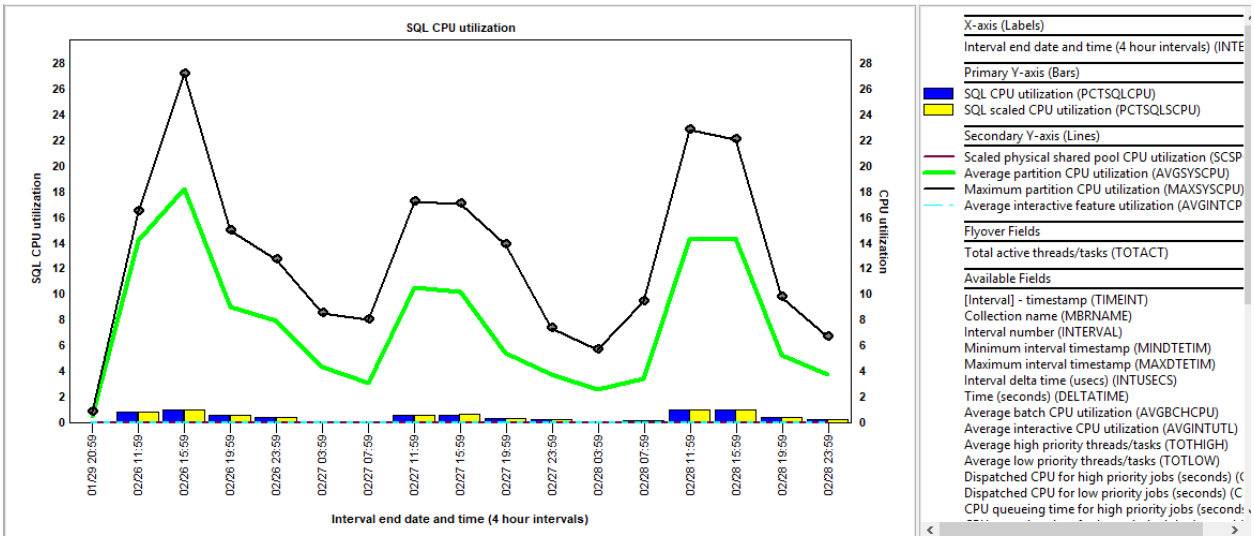
Scaled CPU ratio

9.4.5.8 Shared pool CPU utilization



Shared pool CPU utilization

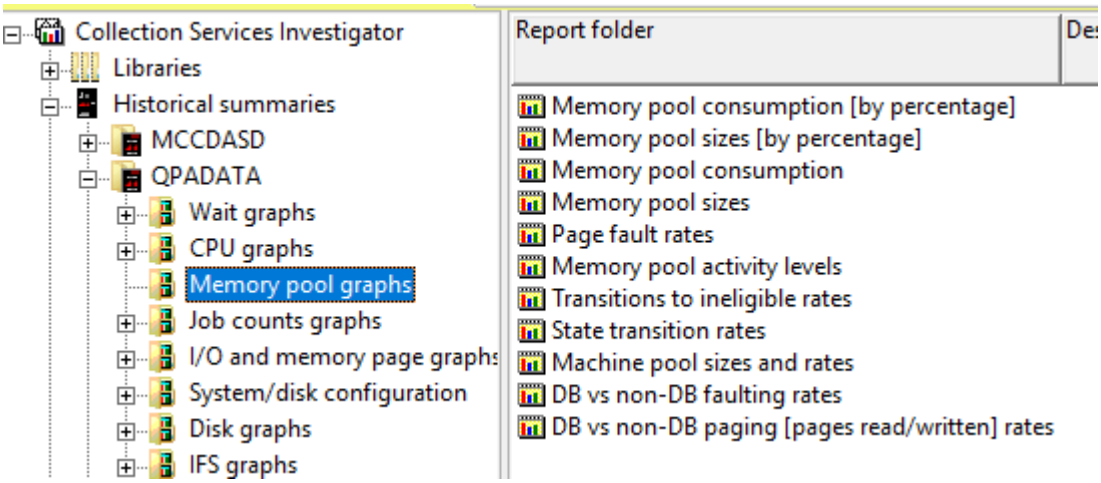
9.4.5.9 SQL CPU utilization



SQL CPU utilization

9.4.6 Memory pool graphs

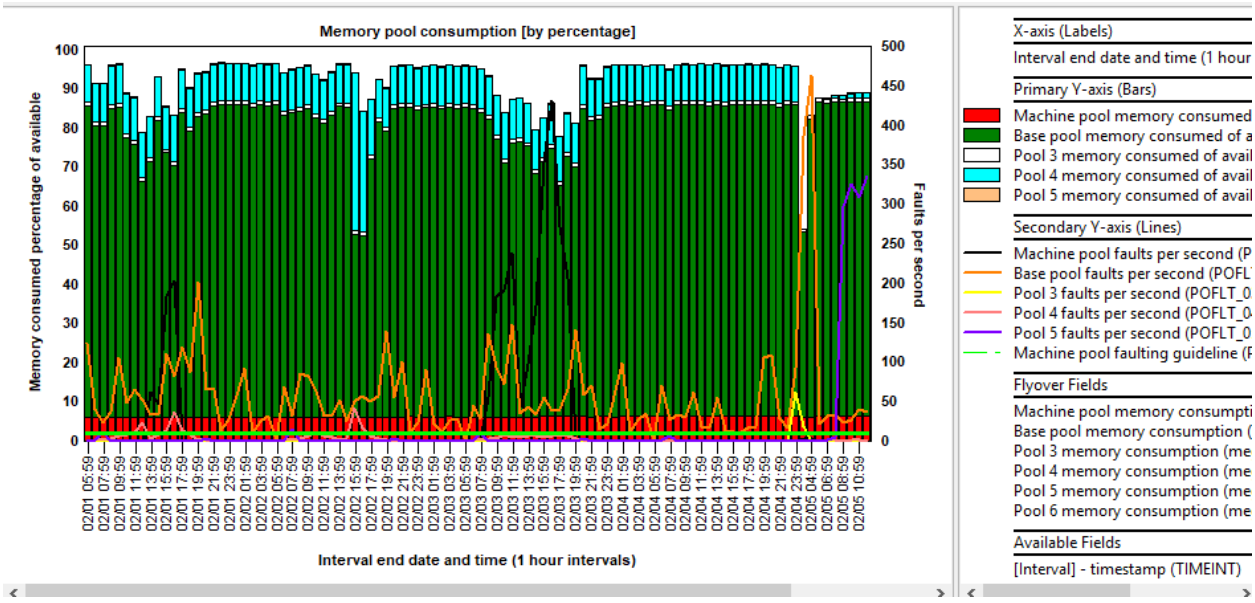
These graphs are like the ones found in a regular Collection Services Investigator collection's Memory pool graphs -> Flattened style folder. Since the graphs are flattened each pool will use a different color.



Memory pool graphs folder

9.4.6.1 Memory pool consumption [by percentage]

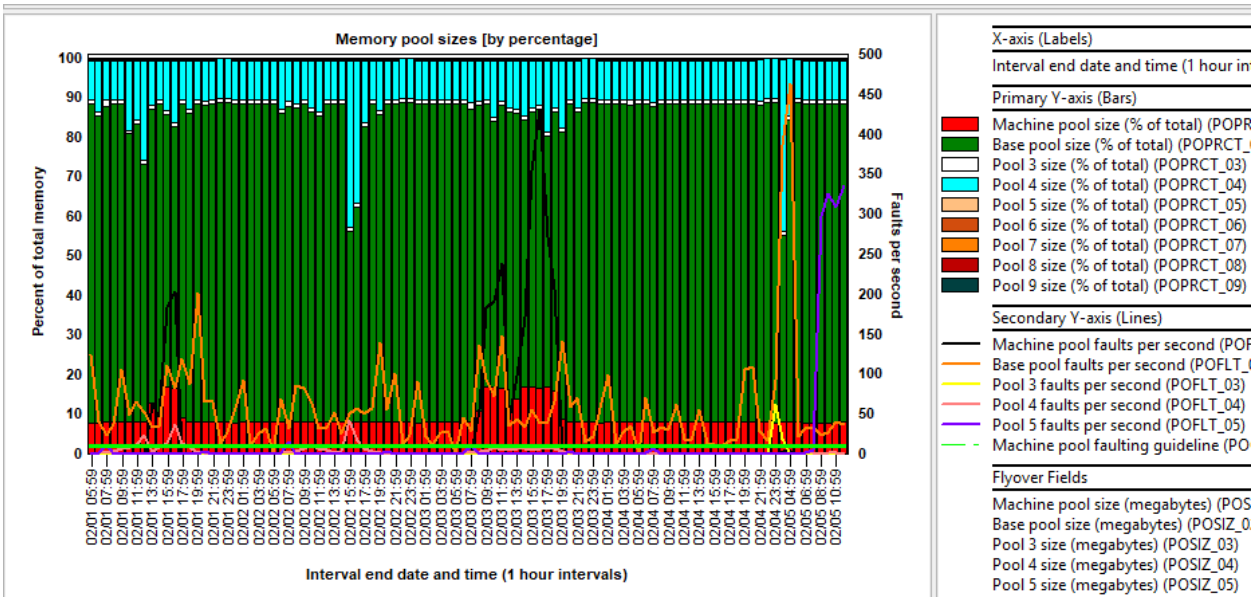
This graph displays the memory pool consumption of available as a percentage.



Memory pool consumption [by percentage]

9.4.6.2 Memory pool sizes [by percentage]

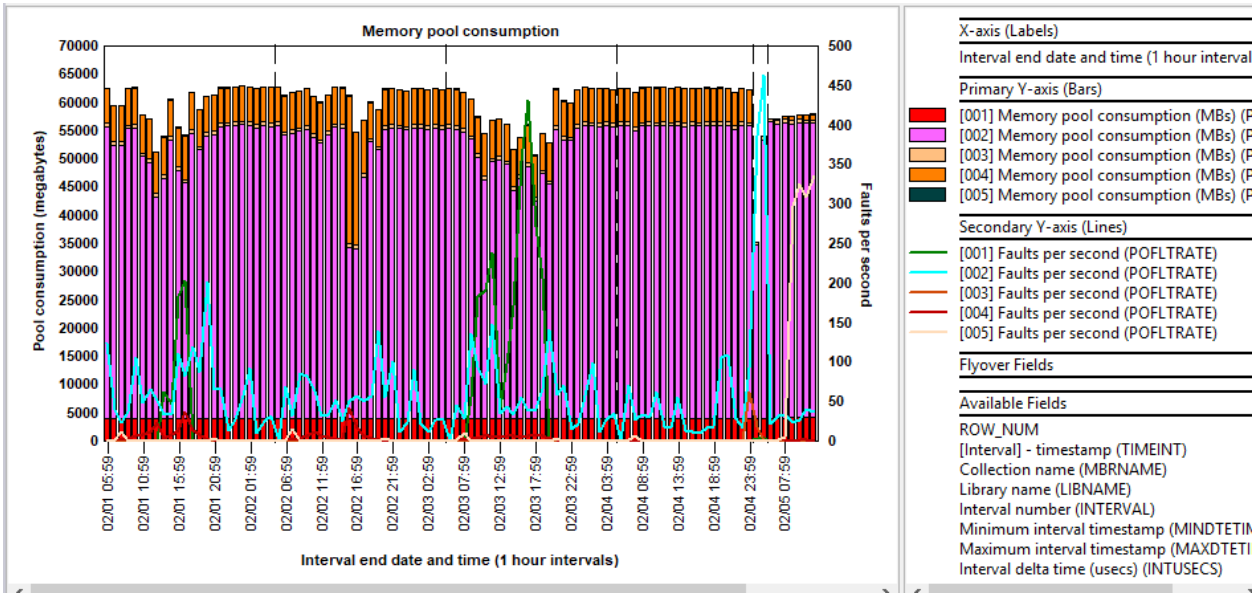
This graph displays memory pool sizes as a percentage of total memory.



Memory pool sizes [by percentage]

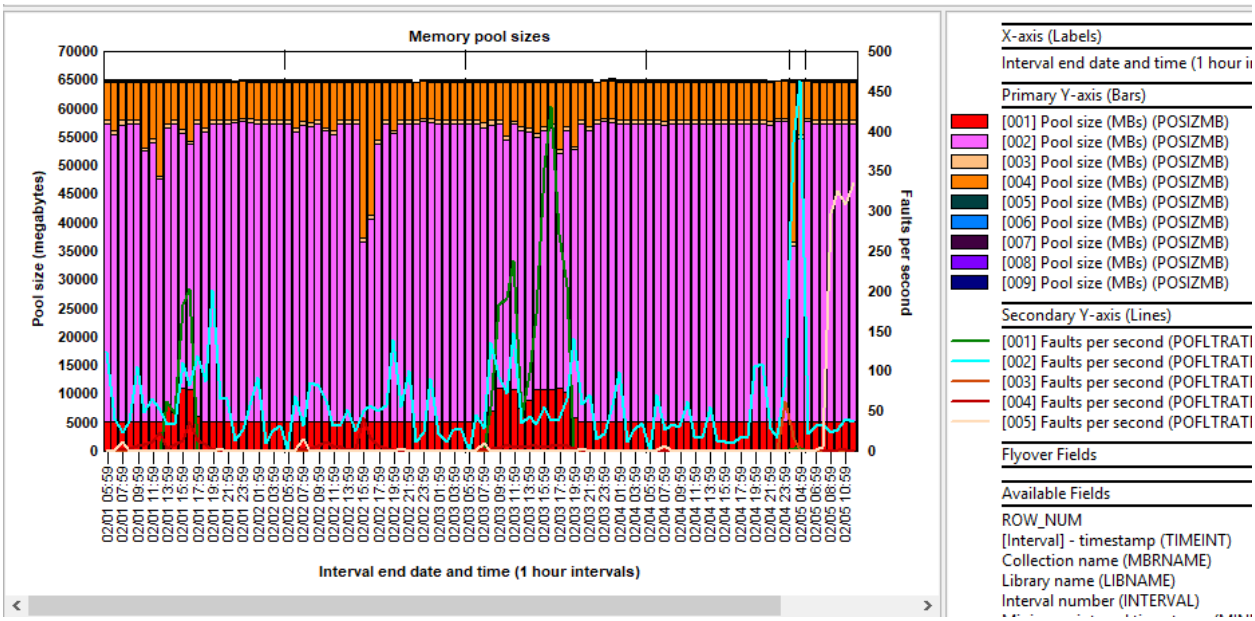
9.4.6.3 Memory pool consumption

This graph displays the consumption of memory (in megabytes) on the primary Y-axis with the faulting rate per pool on the Y2-axis.



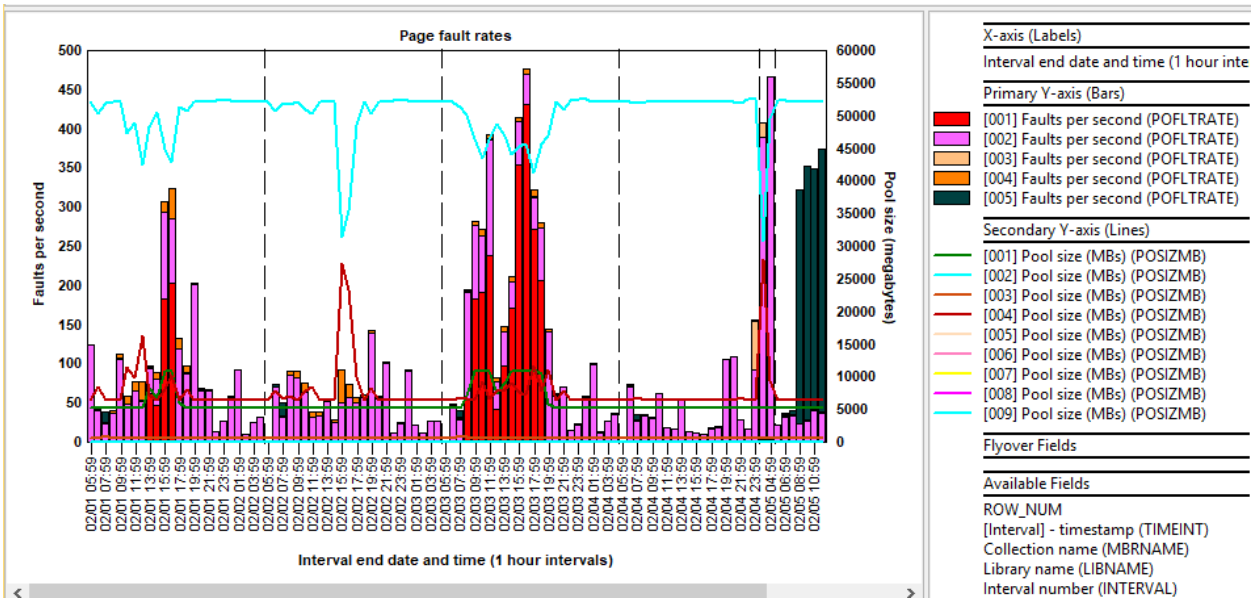
Memory pool consumption

9.4.6.4 Memory pool sizes



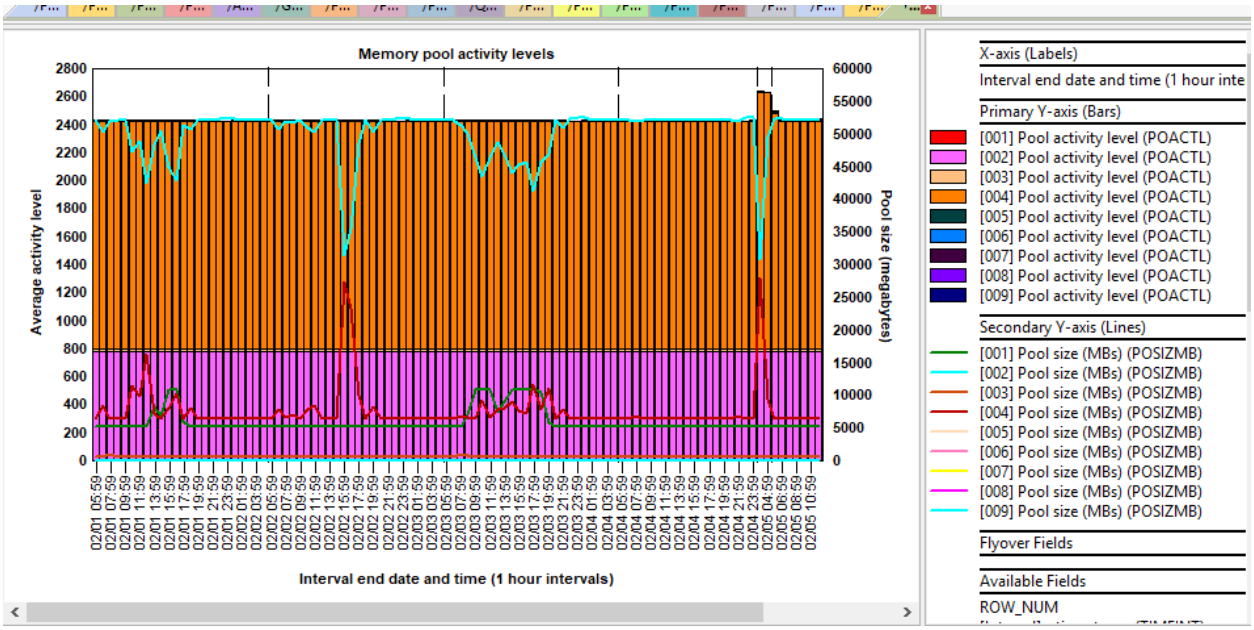
Memory pool sizes

9.4.6.5 Page fault rates



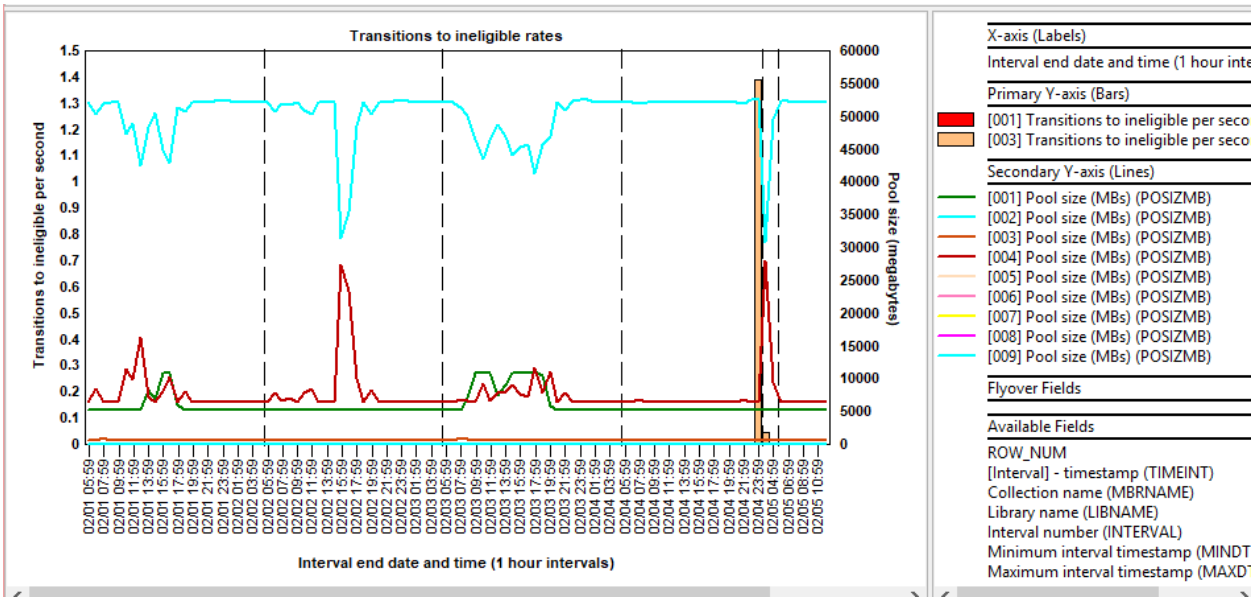
Page fault rates

9.4.6.6 Memory pool activity levels



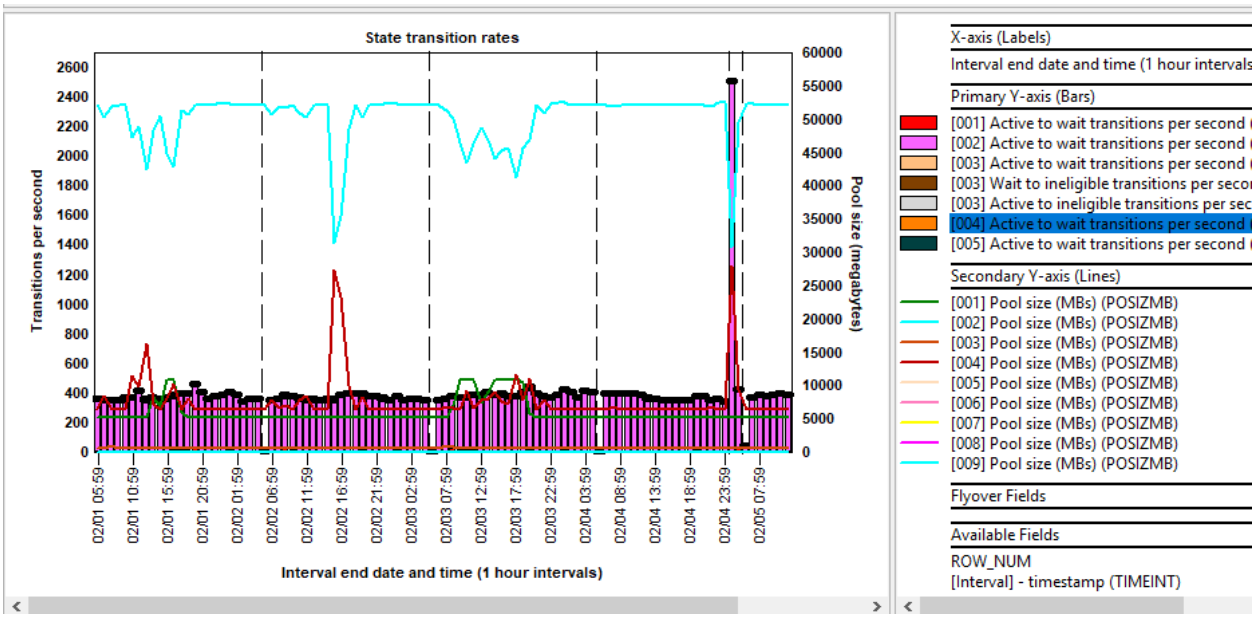
Memory pool activity levels

9.4.6.7 Transitions to ineligible rates



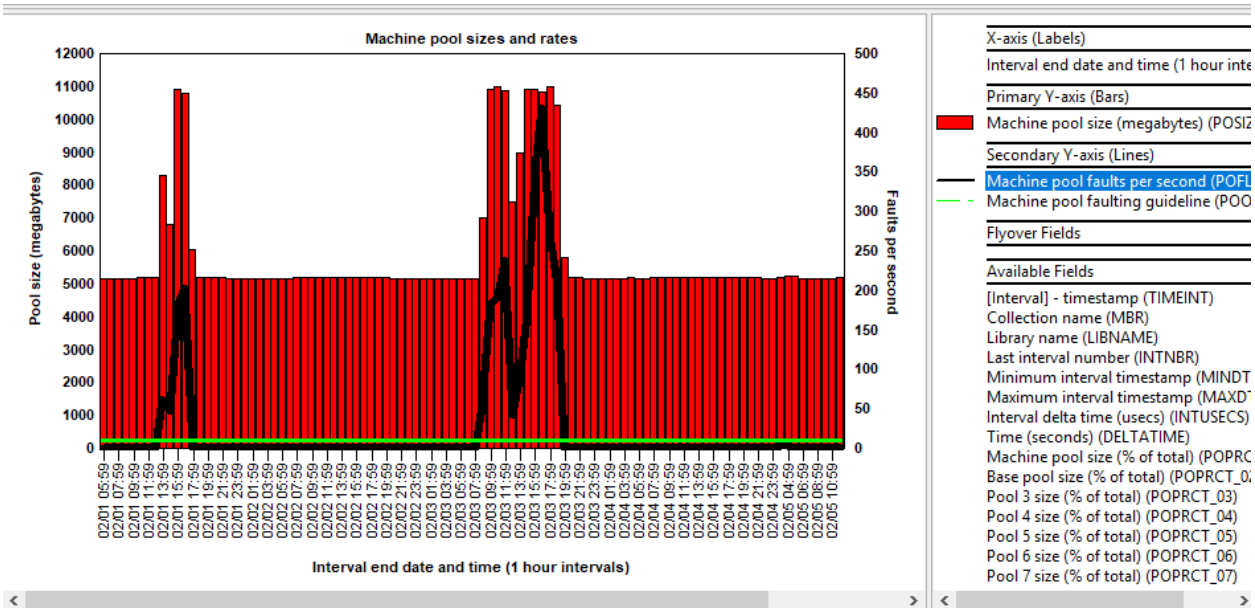
Transitions to ineligible rates

9.4.6.8 State transition rates



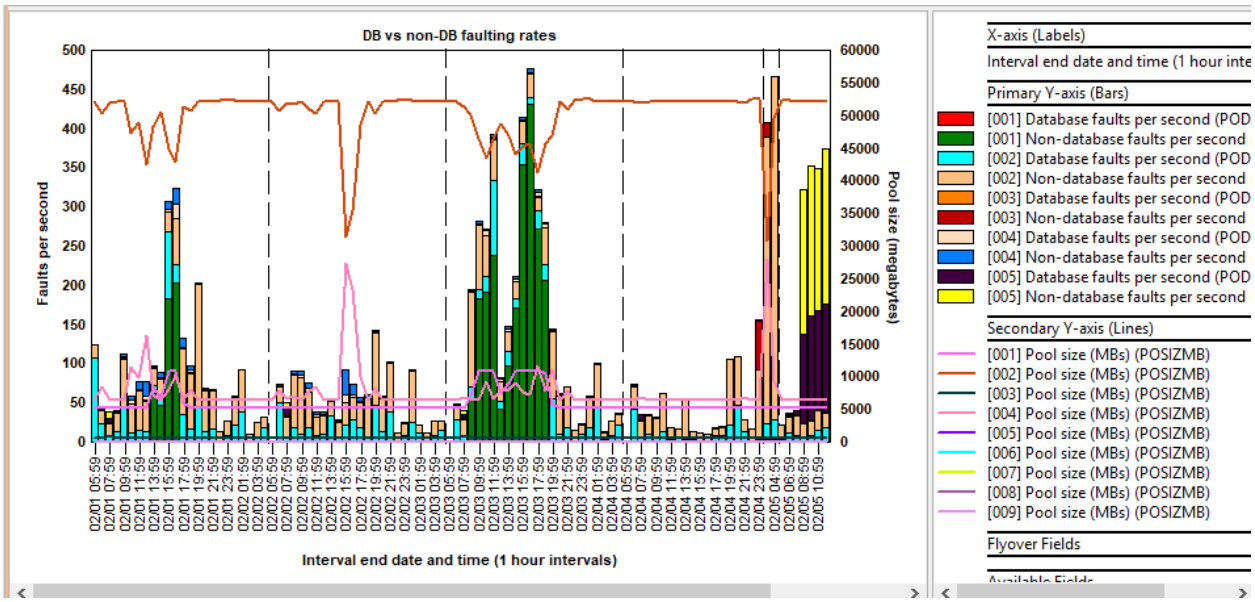
State transition rates

9.4.6.9 Machine pool sizes and rates



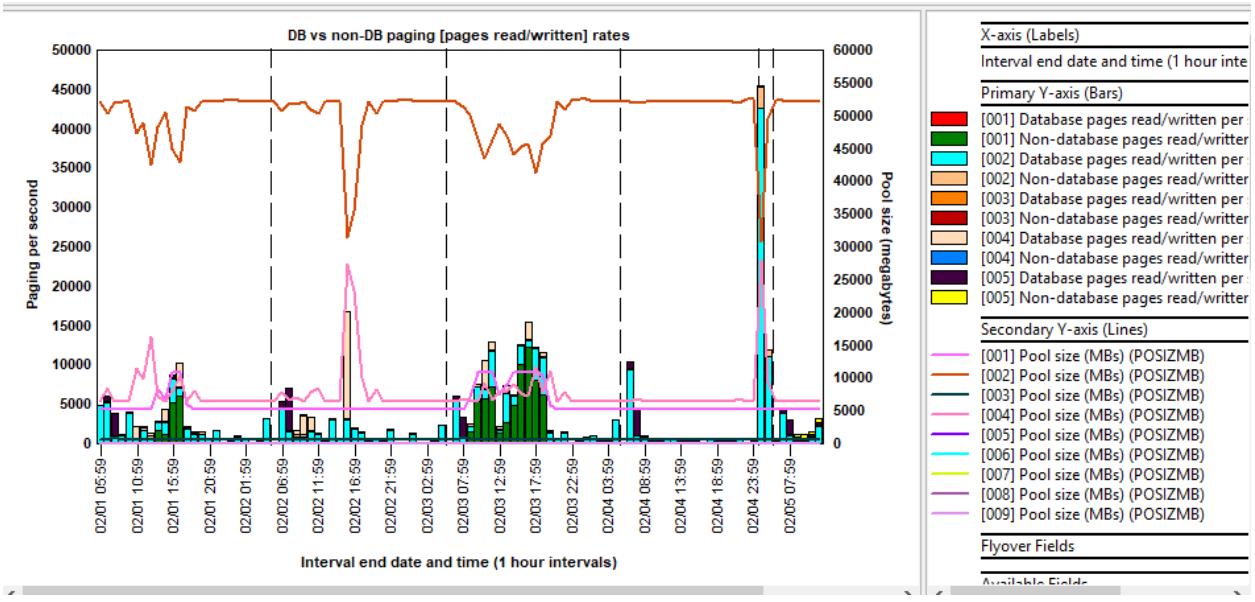
Machine pool sizes and rates

9.4.6.10 DB vs non-DB faulting rates



DB vs non-DB faulting rates

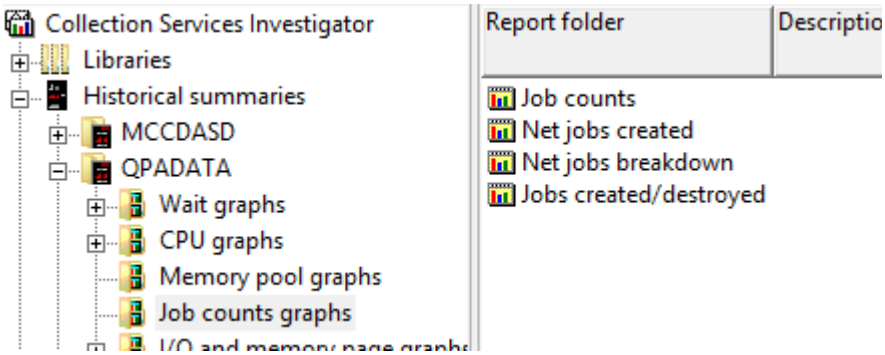
9.4.6.11 DB vs non-DB paging [pages read/written] rates



DB vs non-DB paging [pages read/written] rates

9.4.7 Job counts graphs

This folder contains graphs that show job counts and totals for jobs created/destroyed.

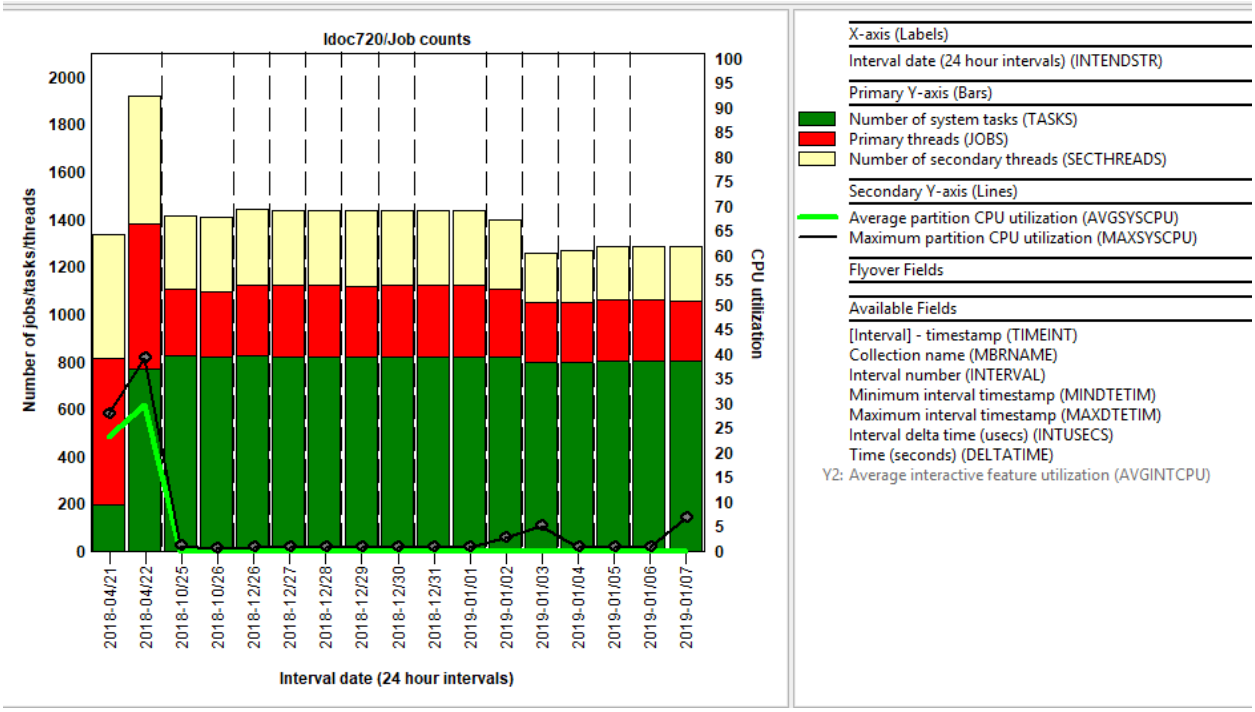


Job counts graphs folder

9.4.7.1 Job counts

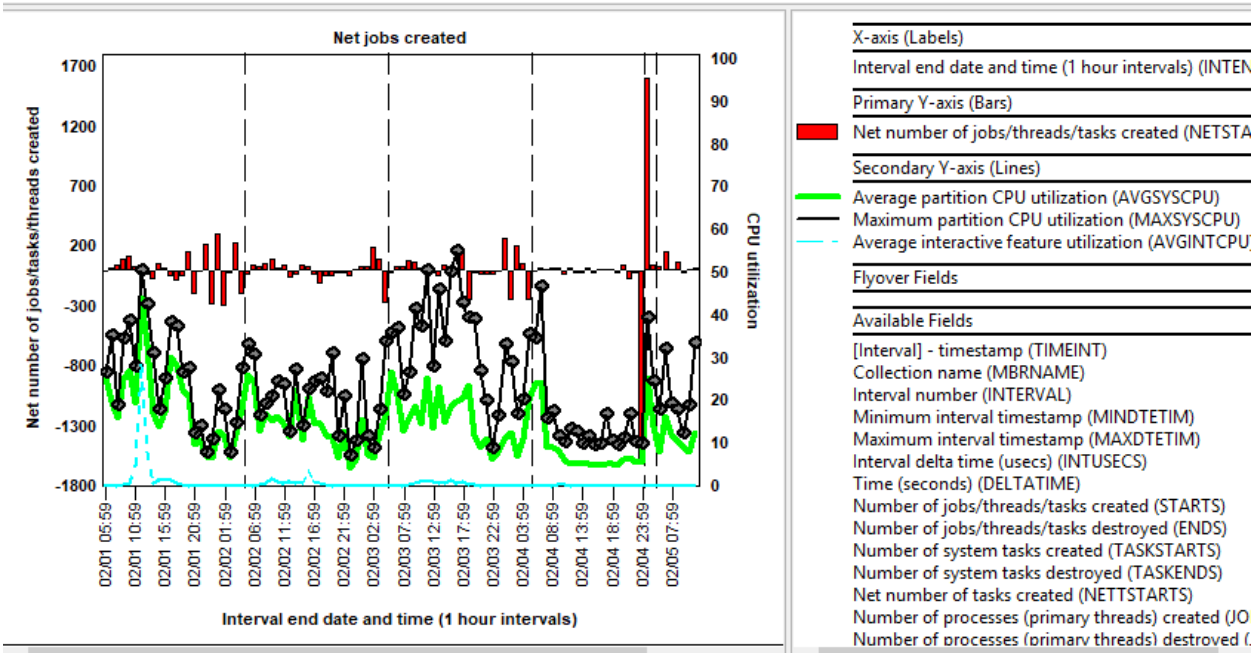
This graph shows number of tasks, primary threads and secondary threads over several days.

Note: Data will only be provided on this graph, if the option to include the job counts was specified when running the historical summary analysis.



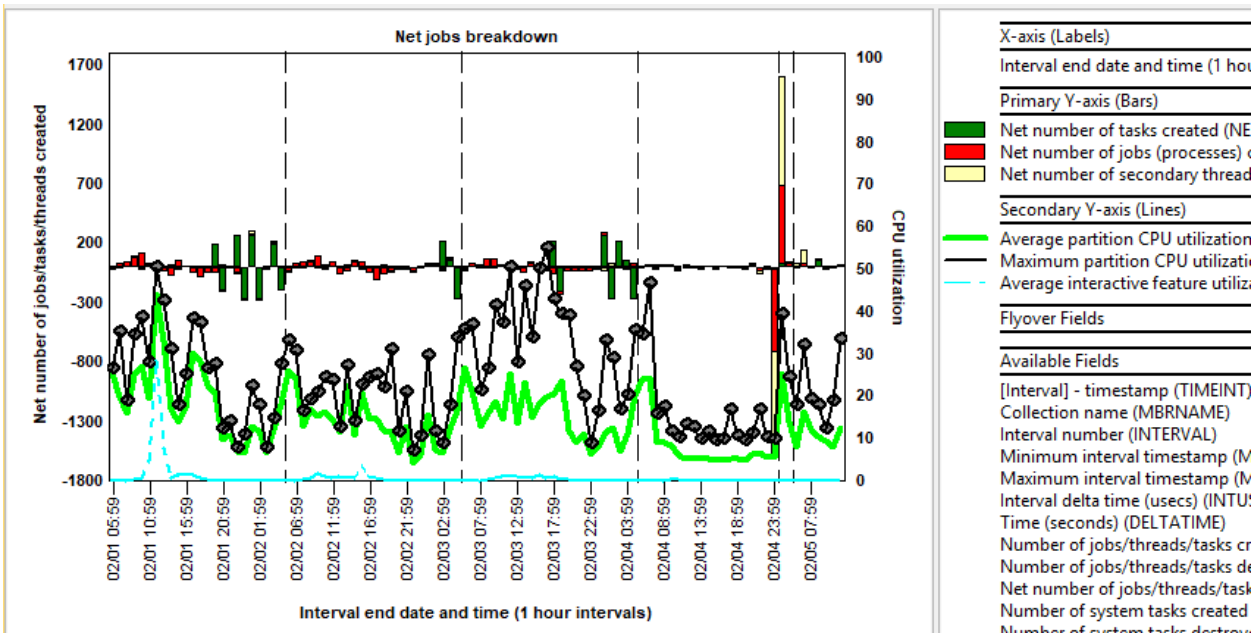
Job counts

9.4.7.2 Net jobs created



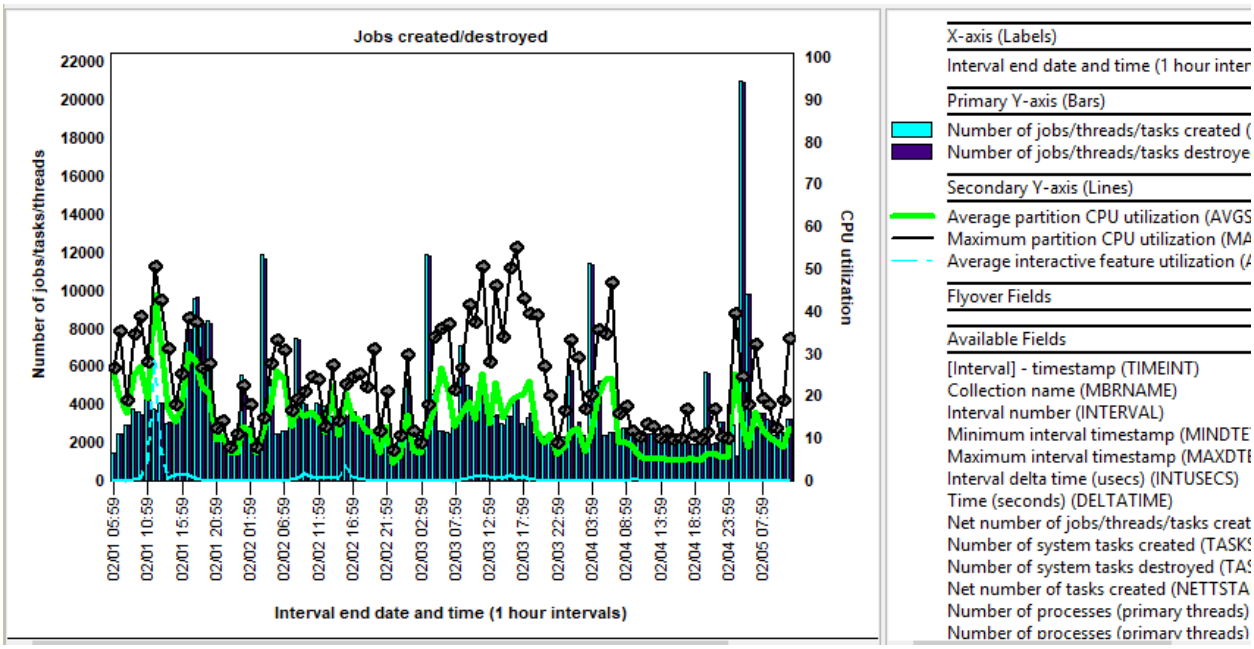
Net jobs created

9.4.7.3 Net jobs breakdown



Net jobs breakdown

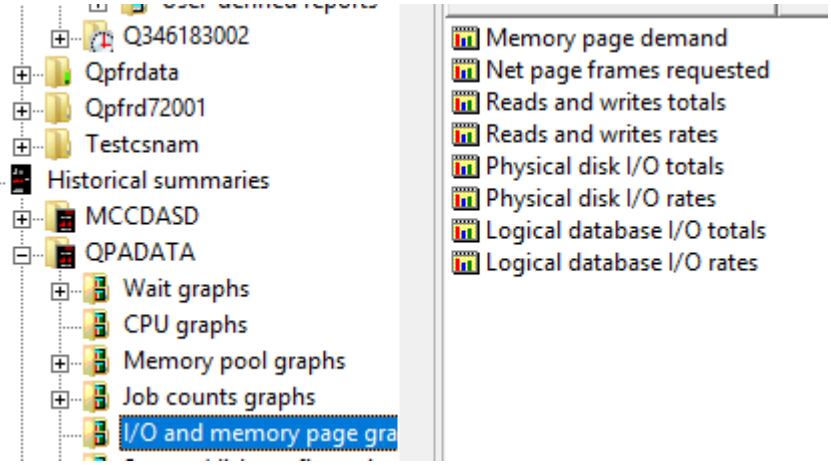
9.4.7.4 Jobs created/destroyed



Jobs created/destroyed

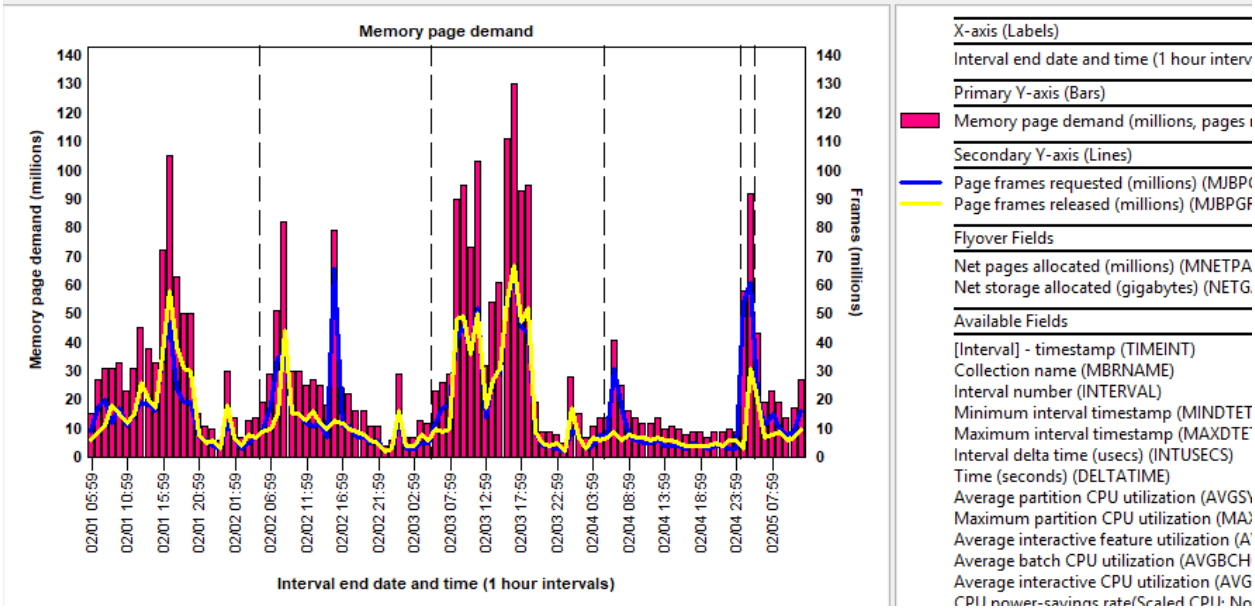
9.4.8 I/O and memory page graphs

These graphs show memory page demands, page frames as well as physical and logical I/O statistics.



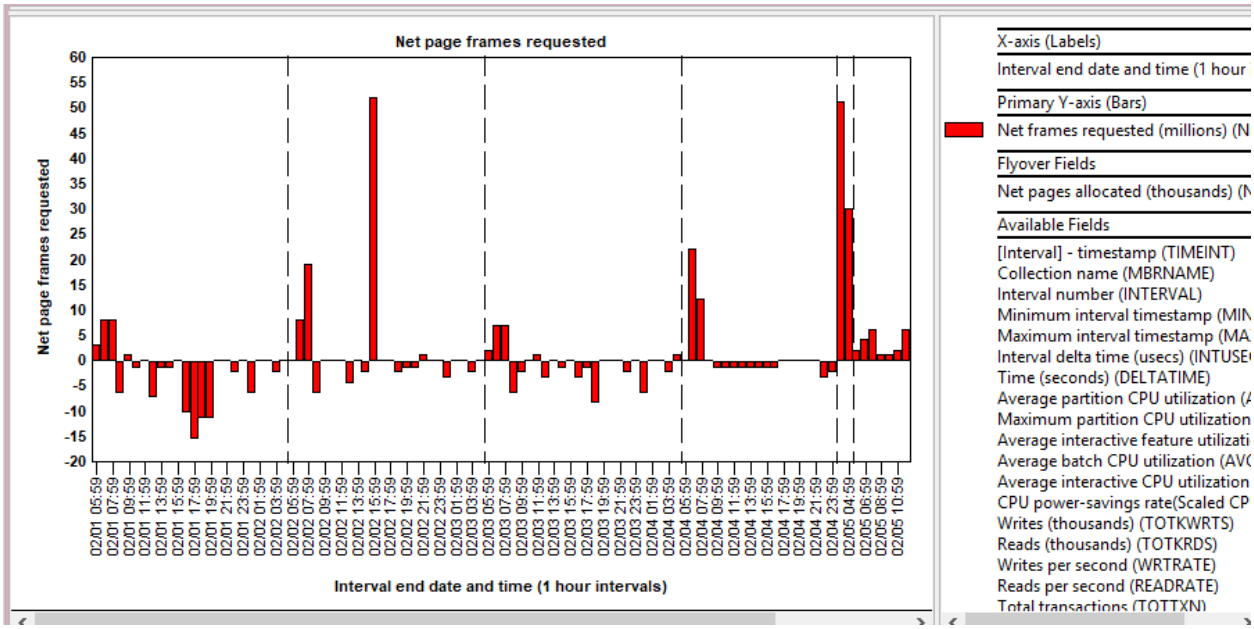
I/O and memory page graphs folder

9.4.8.1 Memory page demand



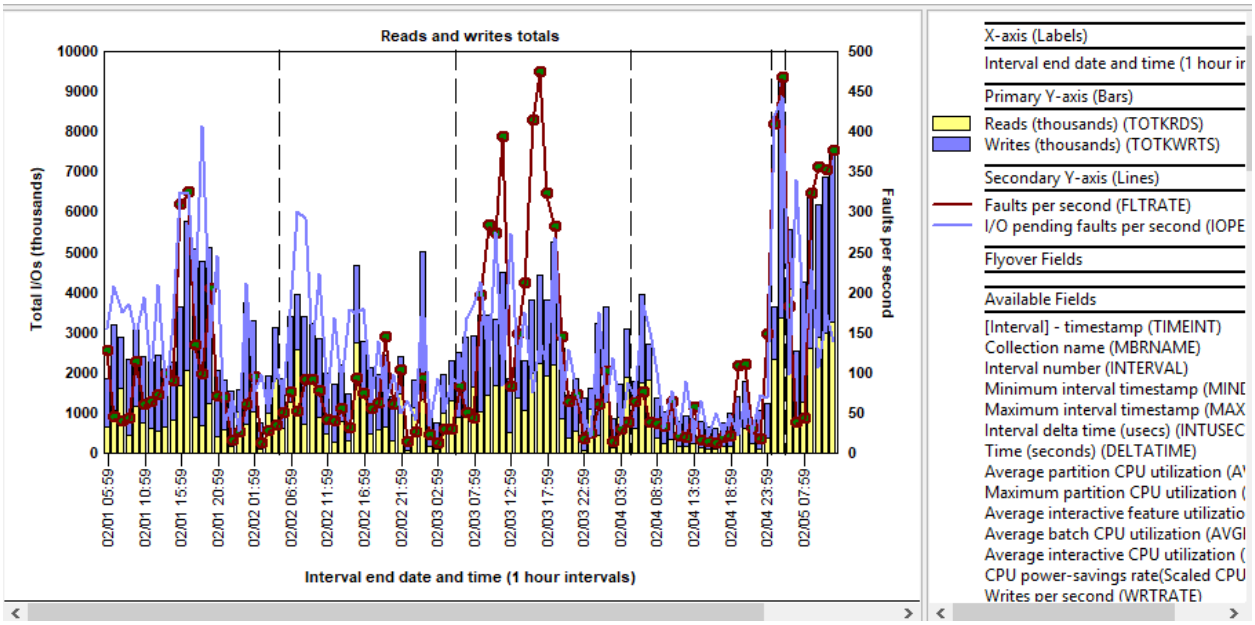
Memory page demand

9.4.8.2 Net page frames requested



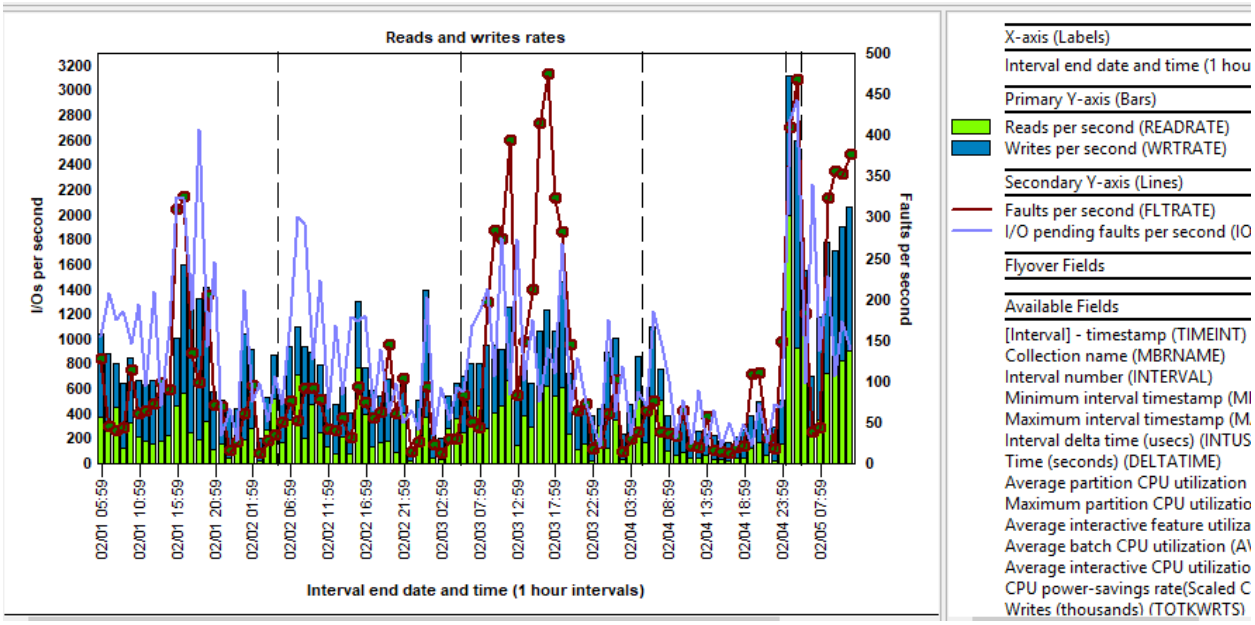
Net page frames requested

9.4.8.3 Reads and writes totals



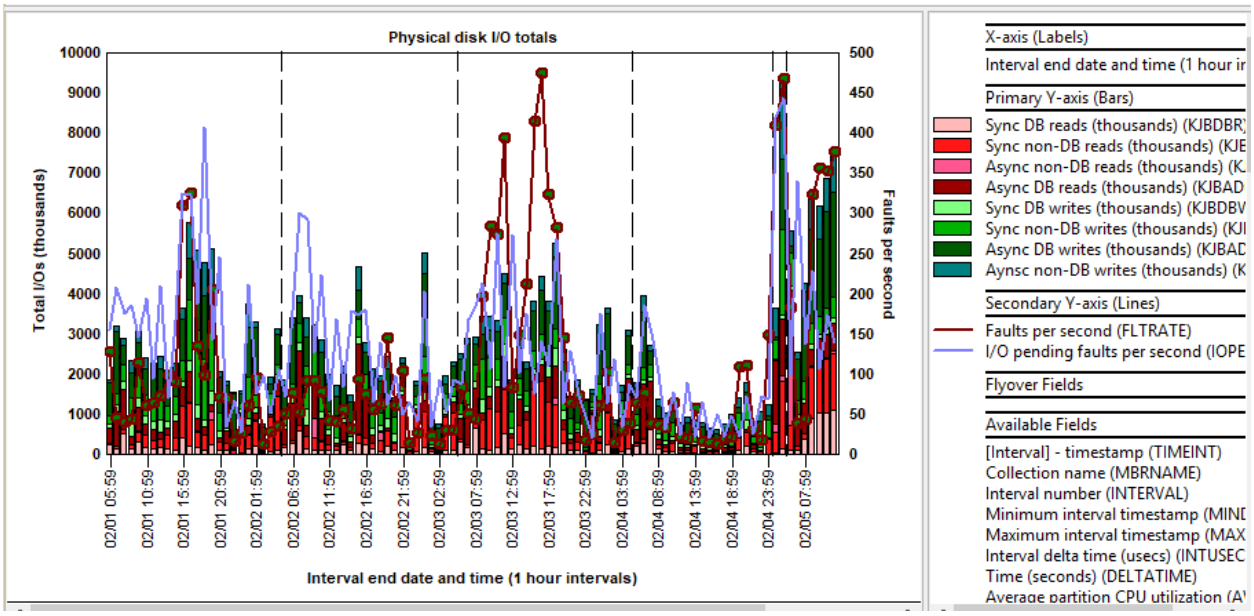
Reads and writes totals

9.4.8.4 Read and writes rates



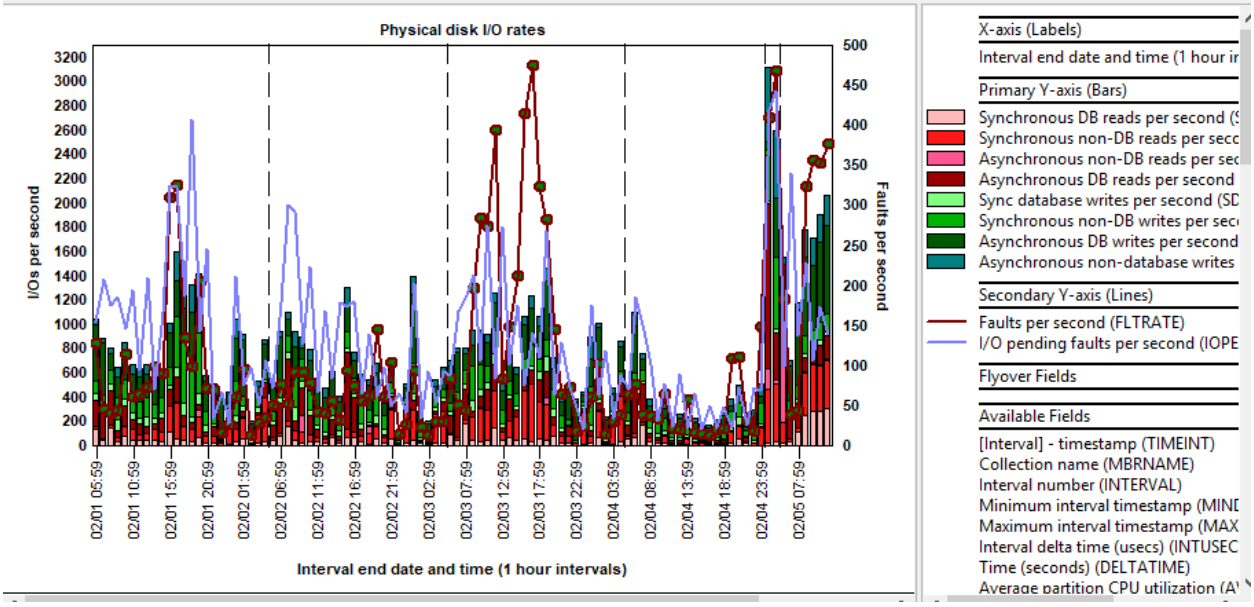
Read and writes rates

Physical disk I/O totals



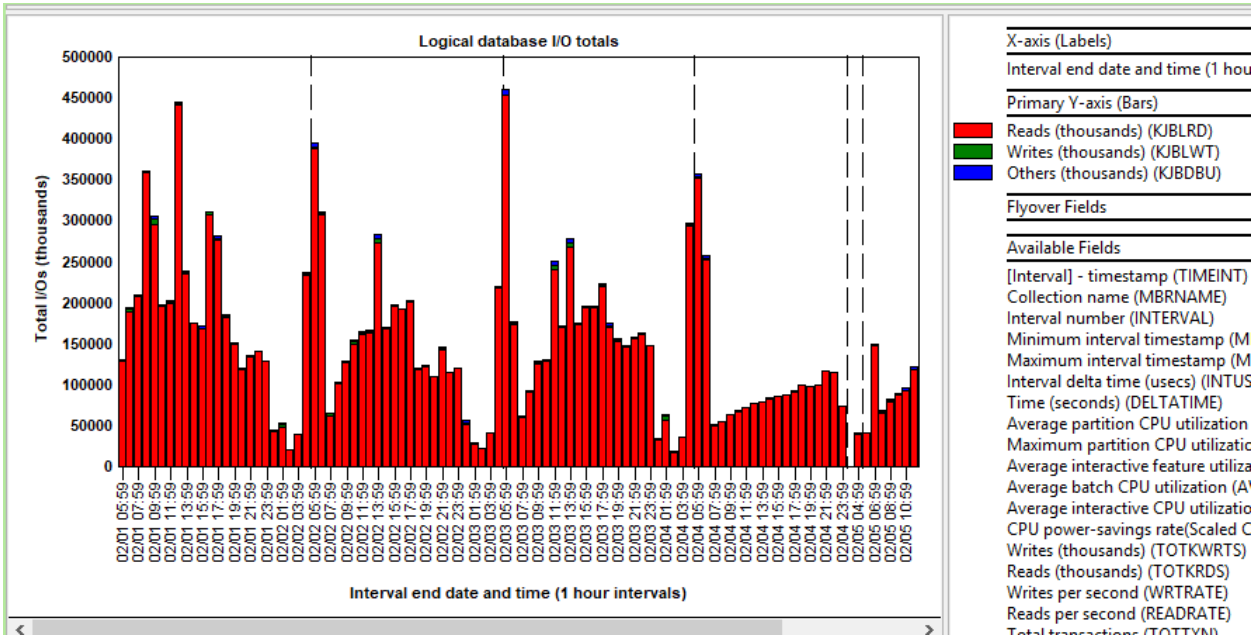
Physical disk I/O totals

9.4.8.5 Physical disk I/O rates



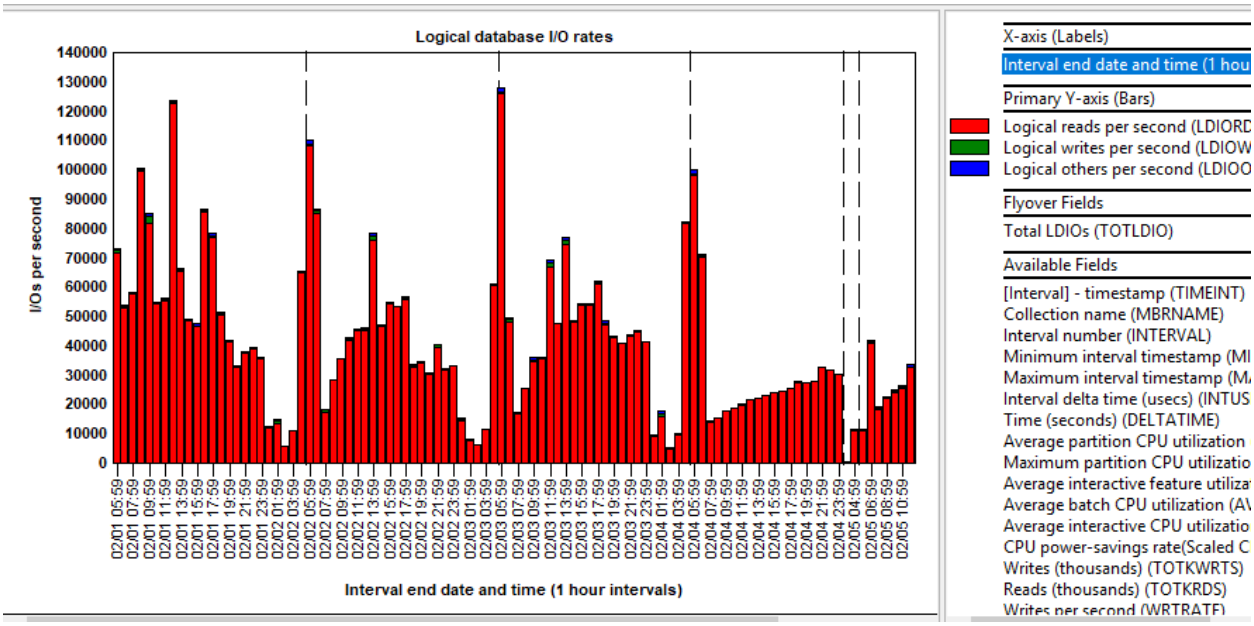
Physical disk I/O rates

9.4.8.6 Logical database I/O totals



Logical database I/O totals

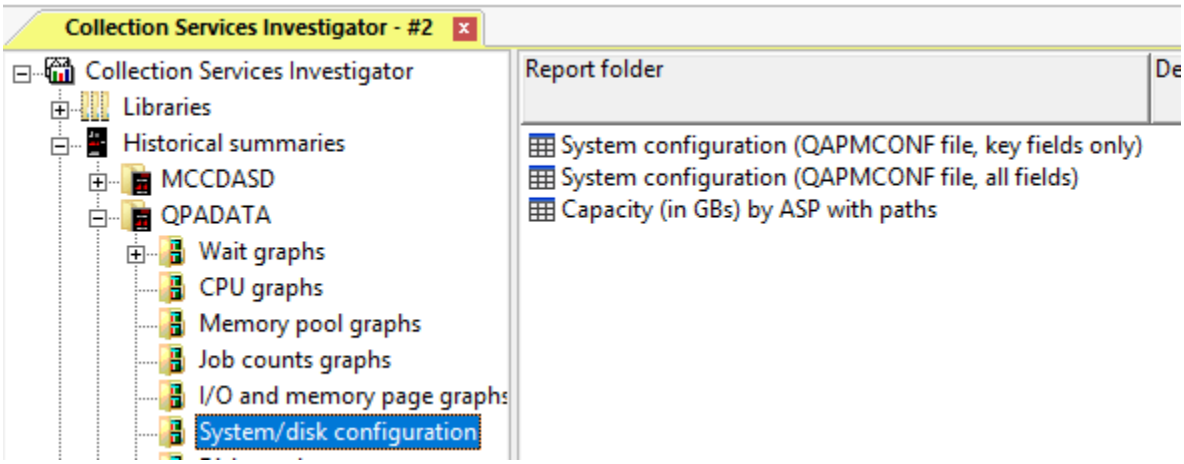
9.4.8.7 Logical database I/O rates



Logical database I/O rates

9.4.9 System/disk configuration

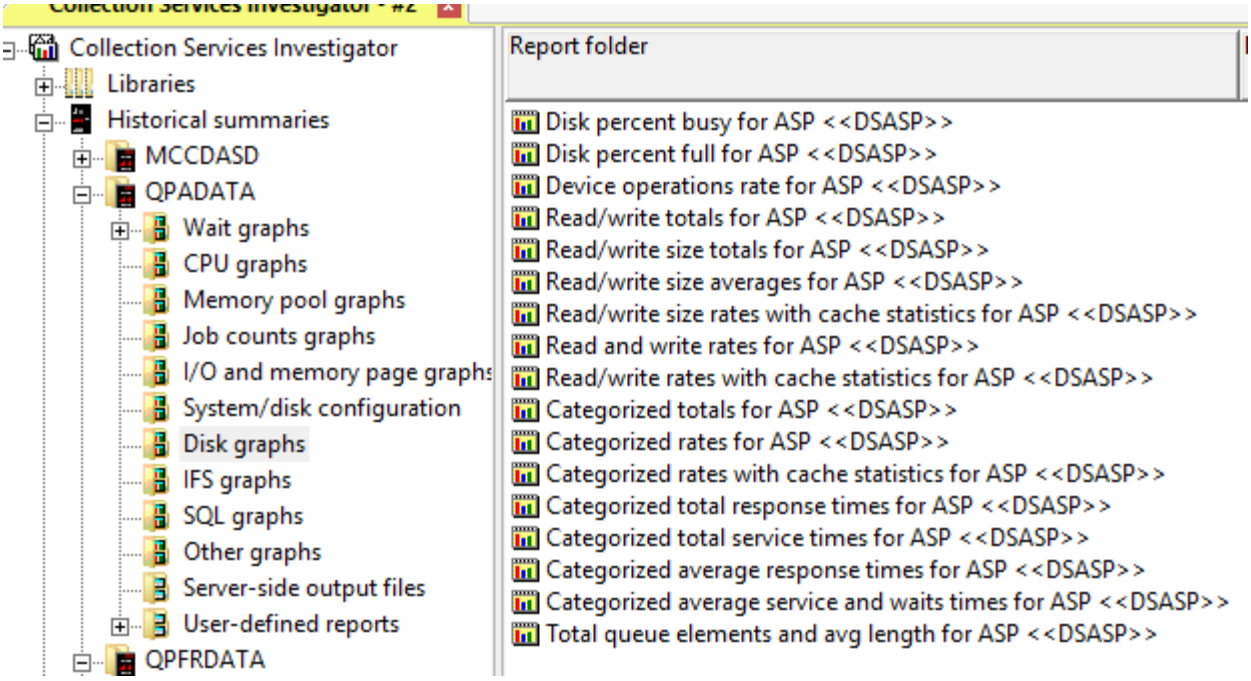
These reports show information from the QAPMCONF file in the collections included for comparison purposes as well as a report showing ASP capacities.



System/disk configuration

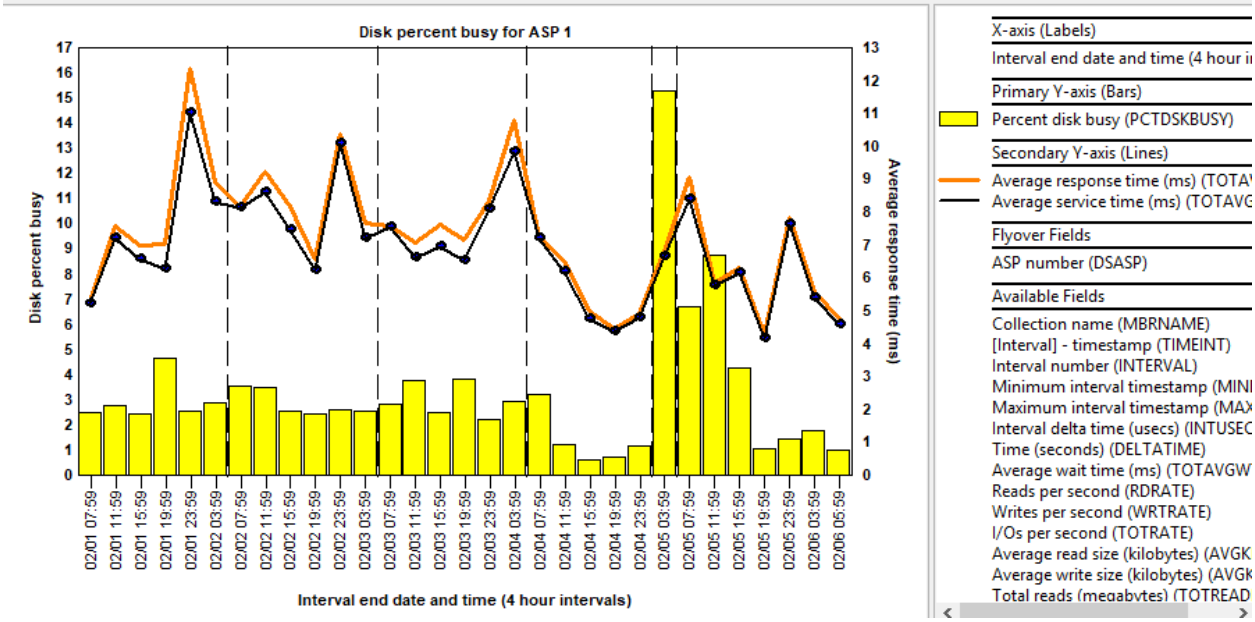
9.4.10 Disk graphs

The disk graphs folder provides high level statistics for the disk data found in the collection.



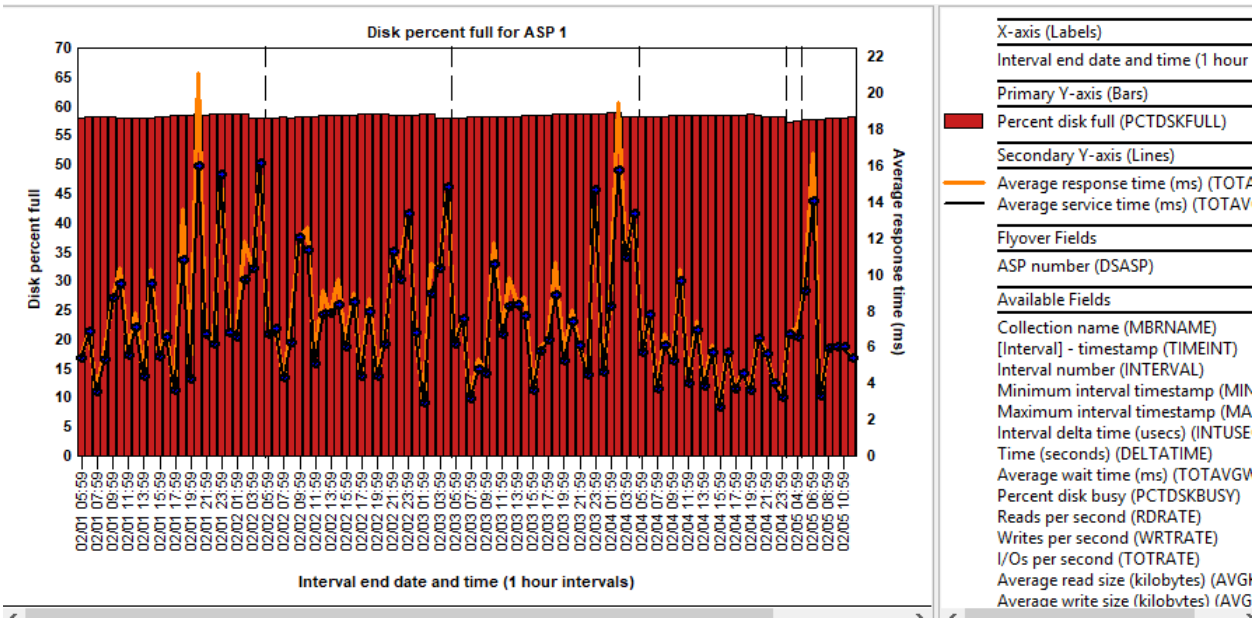
Disk graphs

9.4.10.1 Disk percent busy for ASP X



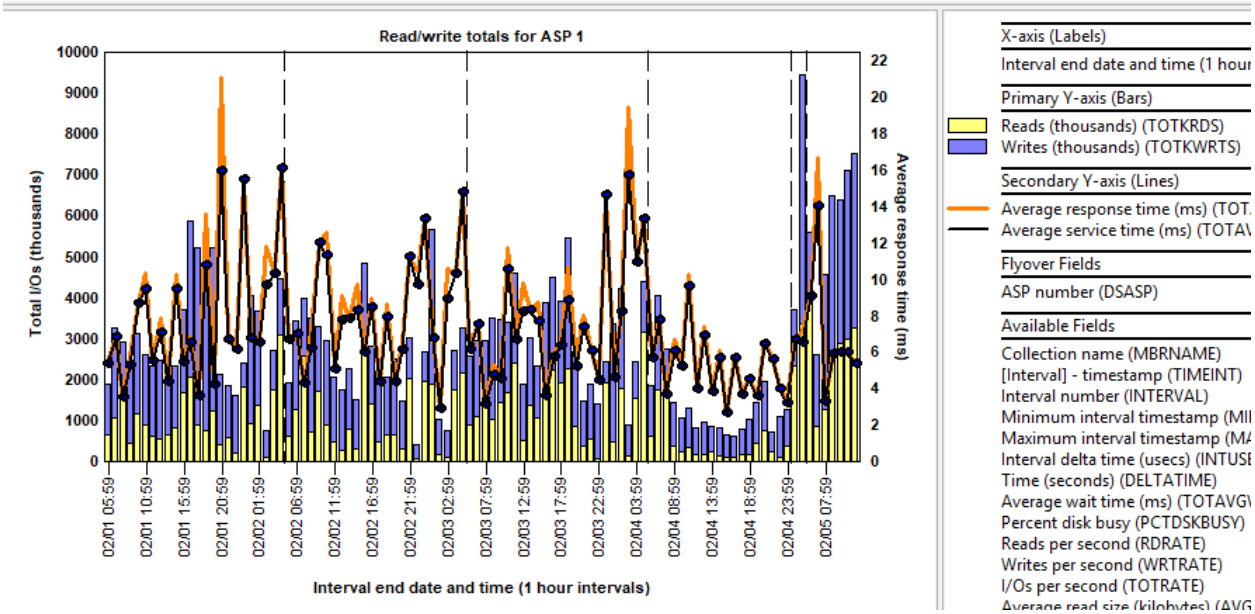
Disk percent busy for ASP X

9.4.10.2 Disk percent full for ASP X



Disk percent full for ASP X

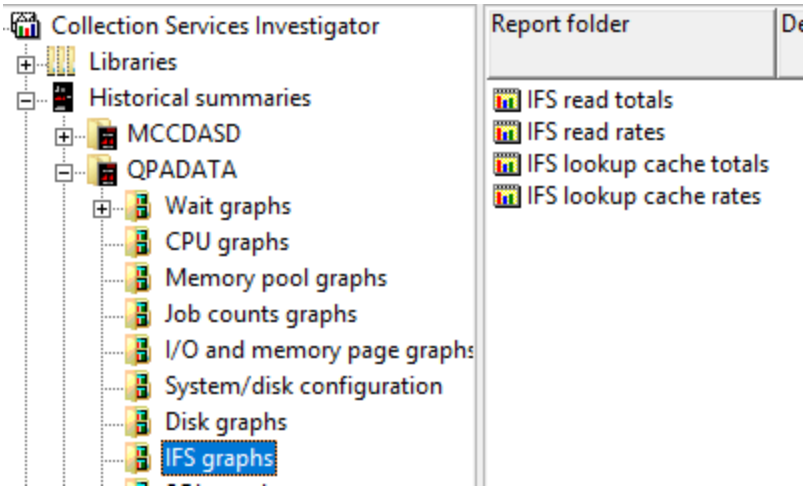
9.4.10.3 Read/write totals for ASP X



Read/write totals for ASP X

9.4.11 IFS graphs

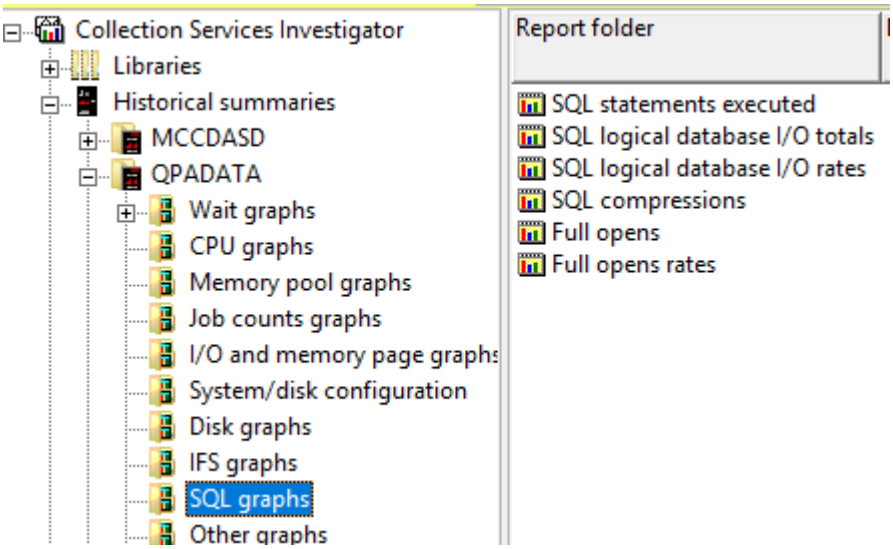
The IFS graphs folder provides statistics for IFS reads and lookup cache metrics over time.



IFS graphs

9.4.12 SQL graphs

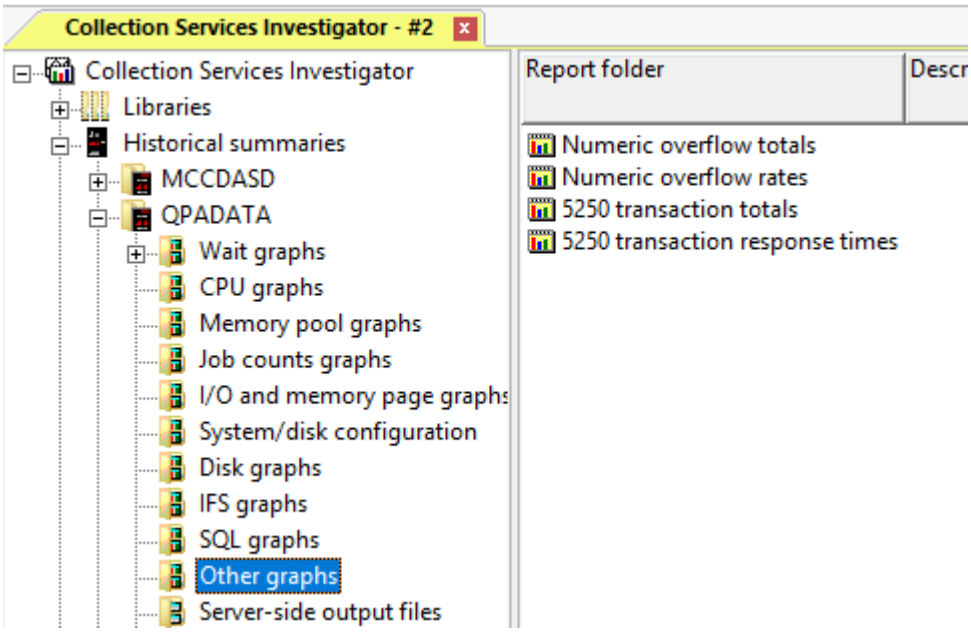
The SQL graphs folder provides metrics related to SQL statements executed as well as opens related to SQL operations or native.



SQL graphs

9.4.13 Other graphs

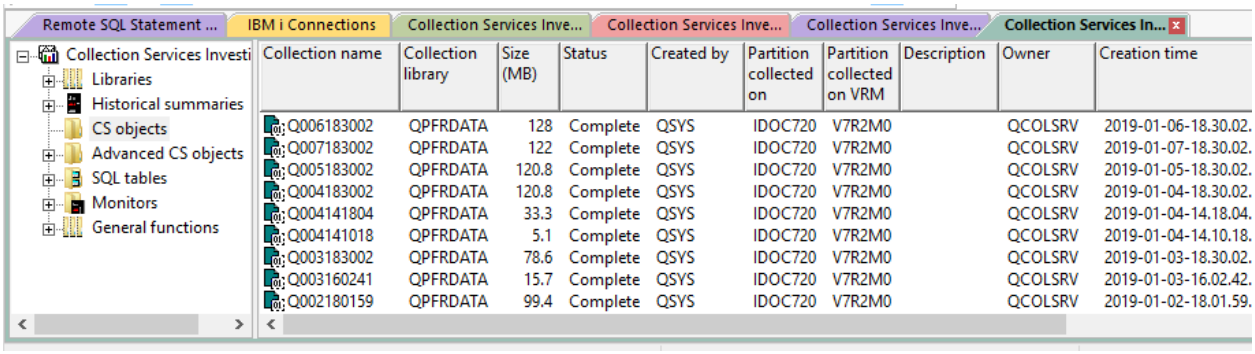
This folder provides metrics related to numeric overflows and 5250 transactions.



Other graphs

9.5 CS Objects

This folder contains the Collection Services management collection (*MGTCOL) objects found on the current system. It provides less details than the Advanced CS Objects folder, but it will display its results faster.

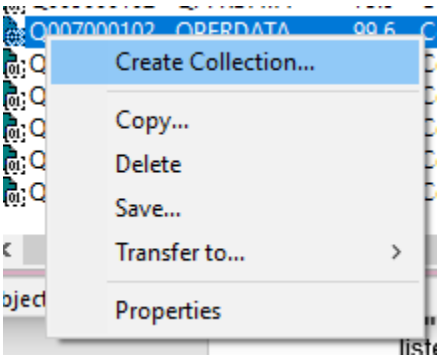


CS Objects folder

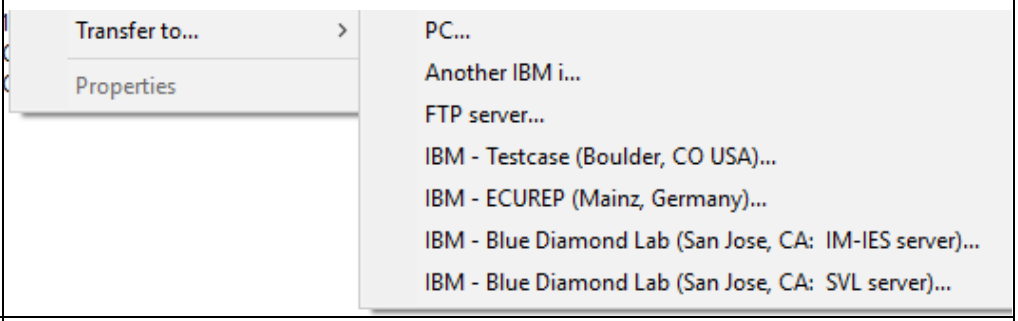
From here a user can see all the *MGTCOL objects on the system that contain Collection Services data. These can be expanded into usable database files if needed or transferred elsewhere.

9.5.1 Menu options

The menu options for a CS management collection object are:



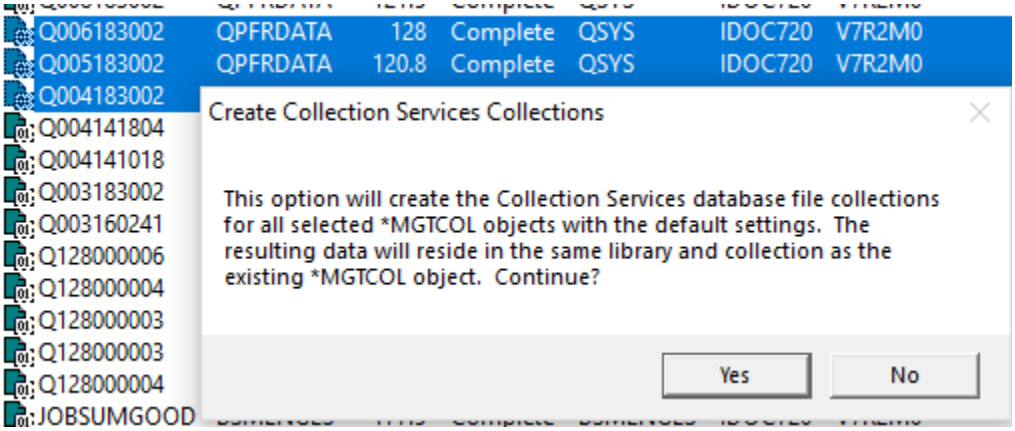
CS Objects collection menu

Option	Description
Create Collection	This option will produce the database files from the management collection object so the data can be graphed within iDoctor (or other tools.) If selecting a single object, then this will display the Create Collection Services Collection window, where the user can decide where to create the new collection and which data options to include. If selecting multiple objects, then the database files will be created with default settings and the Create Collection Services Collection window is NOT shown.
Copy	This option will copy the management collection to another library and/or object name.
Delete	This will remove the selected management collections from the system.
Save	This option is used to save the desired management collection objects into a save file.
Transfer to	These options are used to save the selected object(s) to a save file and then transfer the save file to the desired location from the options listed in the sub menu. 
Properties	The option displays properties for the management collection object.

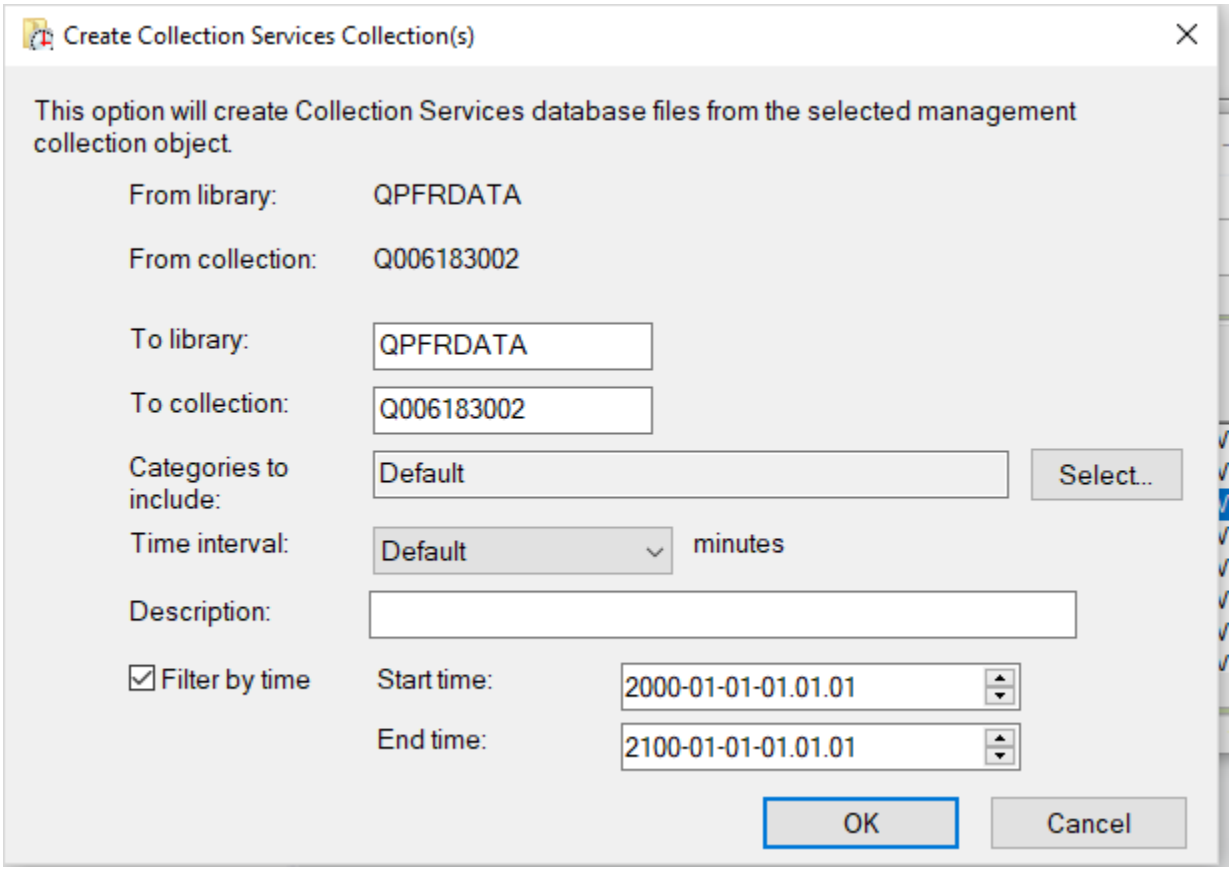
9.5.2 Create Collection Services Collection

This window is used to build the database files needed for analysis from a Collection Services management collection object.

Note: This window is only shown when selecting a **single object** from the CS Objects or Advanced CS Objects folders and using the Create Collection menu. If selecting **multiple objects**, the database files are created using the default settings instead with no prompting.



Create Collection Services Collections prompt

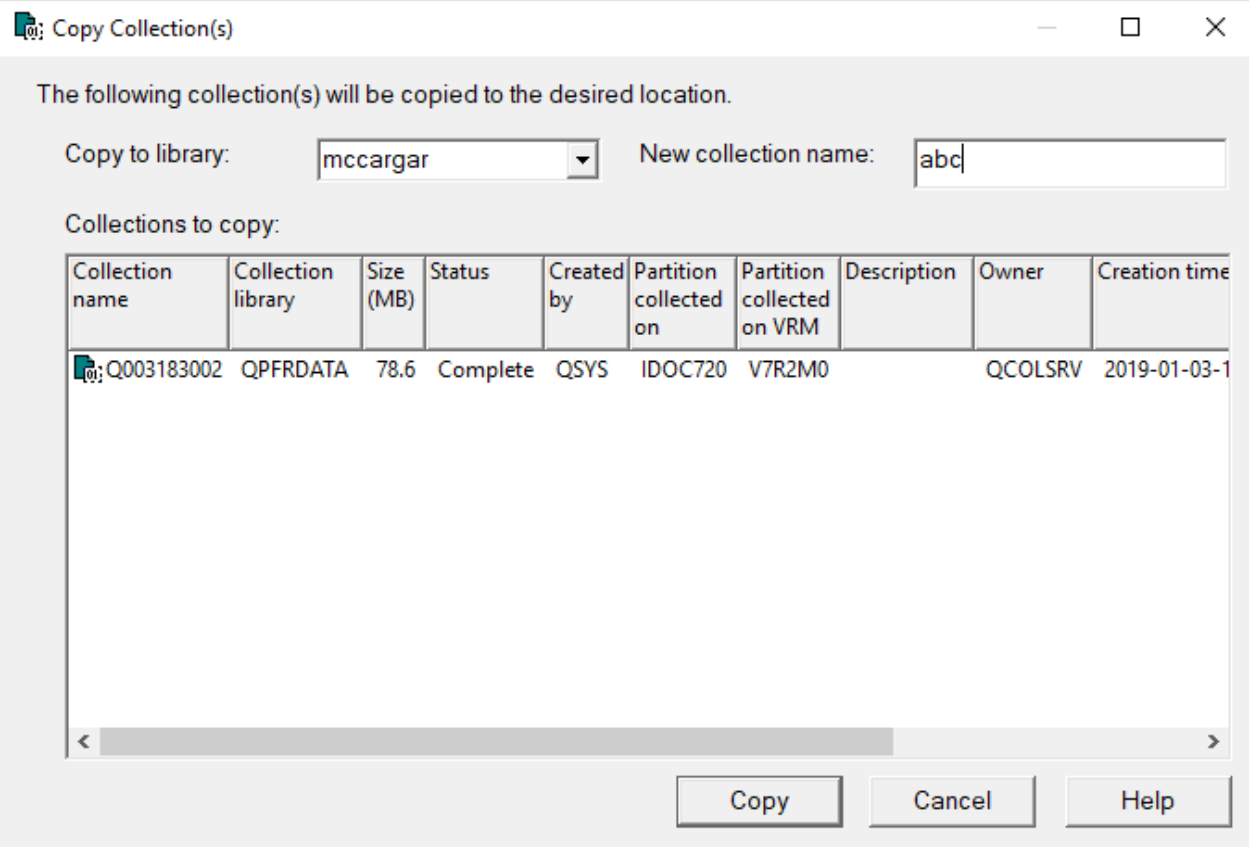


Create Collection Services Collection(s)

Option	Description
From library	The library name where the management collection object resides in.
From collection	The name of the management collection to create data for.
To library	The library name to create the Collection Services database files into.
To collection	The collection / member name to use for the database files to create. These files are all named QAPM*.
Categories to include	This value indicates the types of data to create from the management collection object. It will either contain “Default” or a list of values each representing a type of data to create. Press the Select button next to this field to configure this.
Time interval	This setting indicates the duration of each time period to create from the data. The value is in minutes and includes values < 1. .25 = 15 seconds, .5 = 30 seconds, etc.
Description	This is an optional description to give the new collection created.
Filter by time	This value allows the data in the collection to be filtered by a desired start and end time period.

9.5.3 Copy

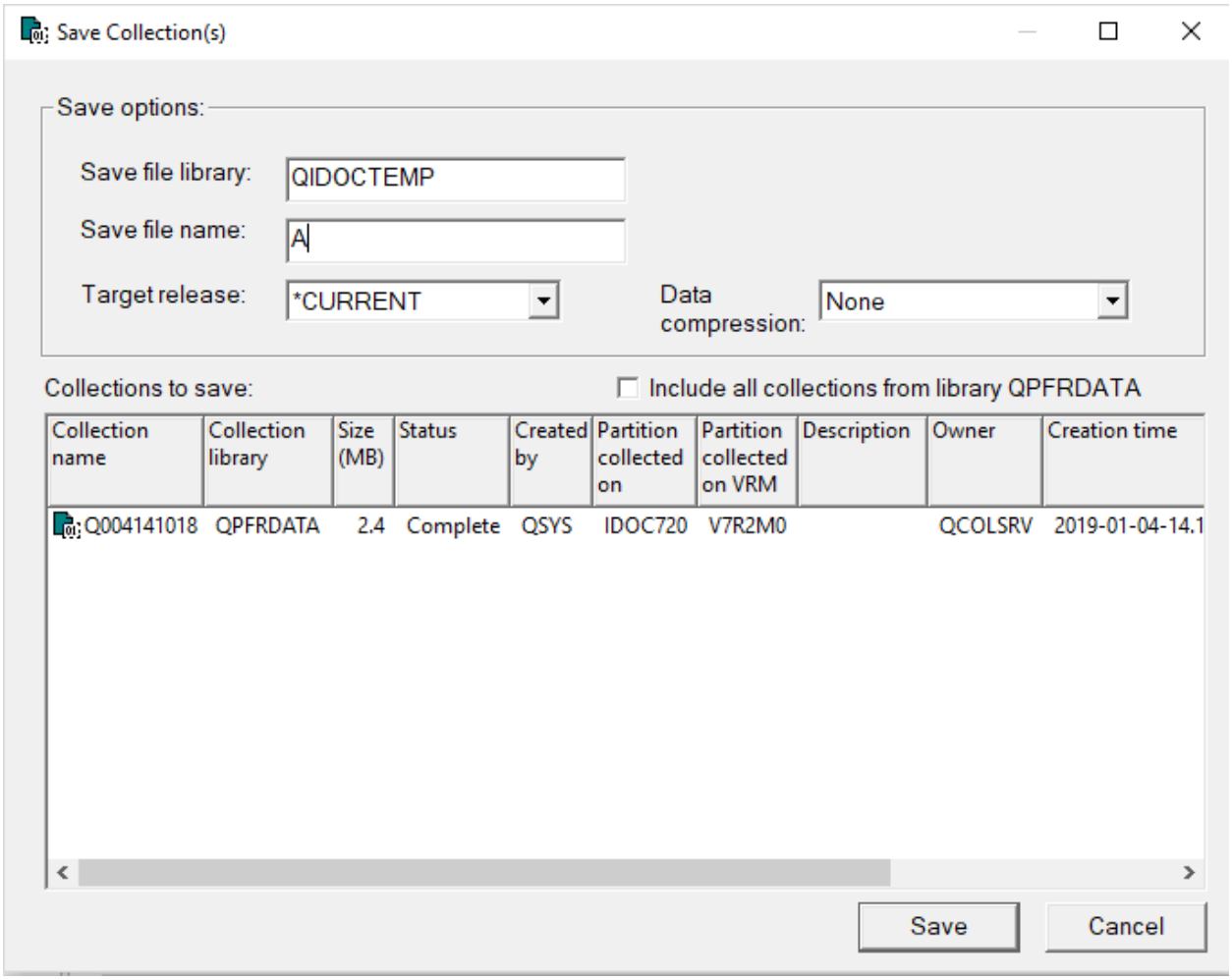
This window is shown when a user copies a management collection object to another library and/or object name.



Copy Collection(s)

9.5.4 Save

This window is used to save 1 or more collections into a save file.

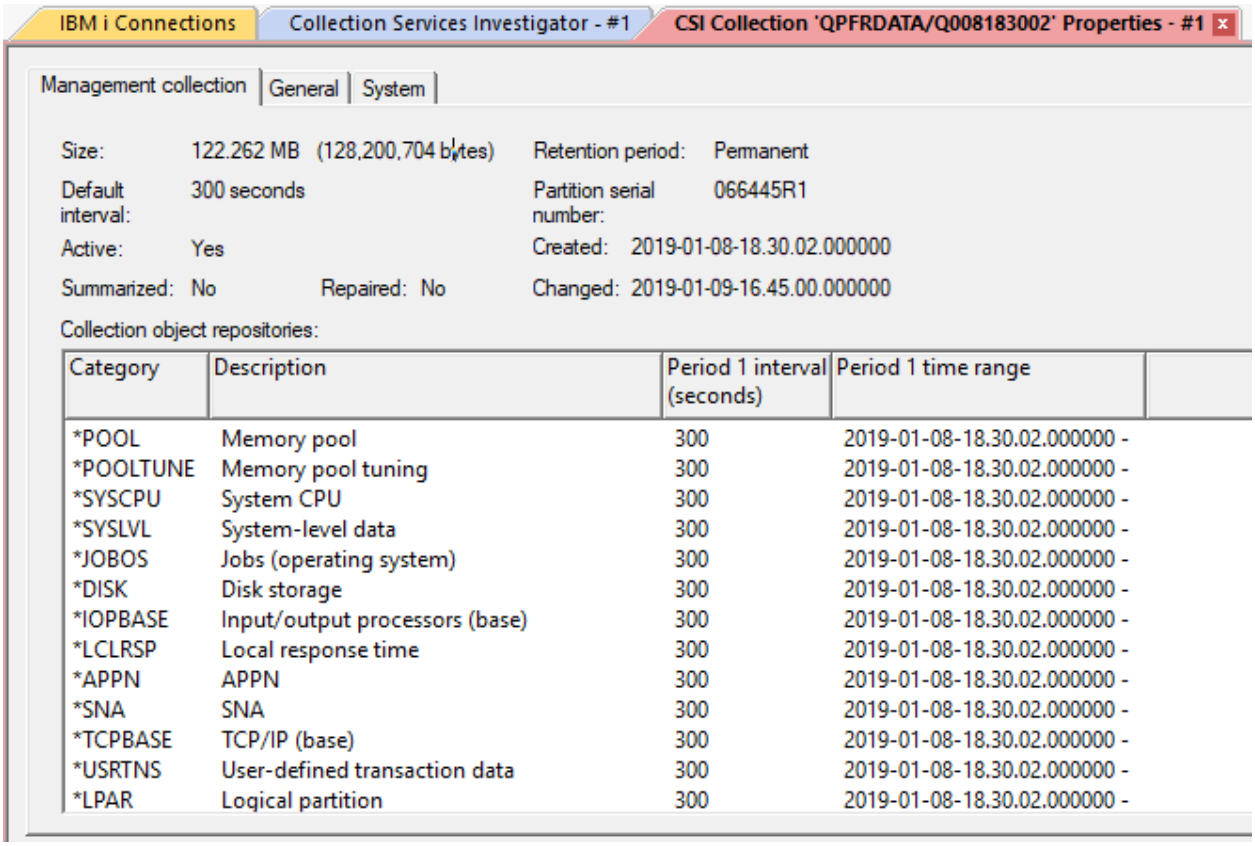


Save Collection(s)

9.5.5 Properties

The properties interface for a management collection object displays information about when it was created, its size and what types of data it contains.

This panel contains tabs for the Management collection and only if the Collection Services database files exist for this management collection object then additional tabs are shown: General, System. These tabs pull data from the database files and NOT the management collection object and are therefore not covered here.



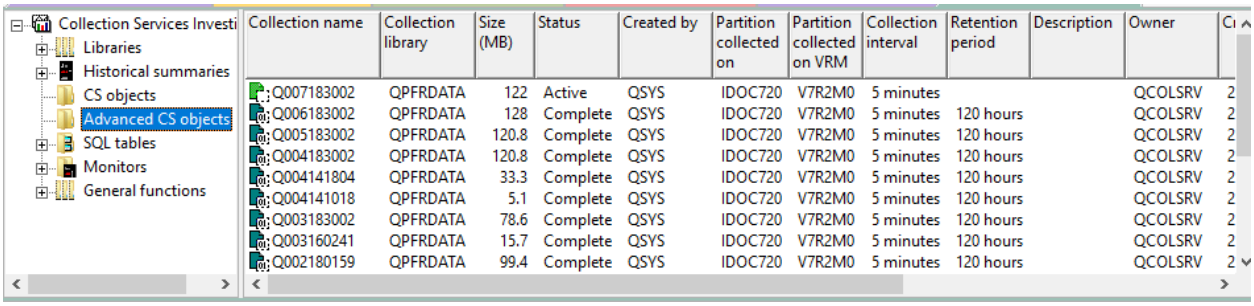
Management collection

Option	Description
Size	The size of the management collection object.
Retention period	How long the management collection object will remain on the system.
Default interval	The default duration of each collected time period within the collection.
Partition serial number	The system serial number of the system where this collection was created on.
Active	This indicates whether the collection is currently running or not.
Created	The time when the management collection object was created.
Summarized	This value indicates if the management collection object contains summary data or not.
Repaired	This value indicates if the object was repaired due to abnormal termination.
Changed	The time when the collection was last changed.
Collection object repositories	This contains a list of data collection categories that exist within the management collection and the time period of each. **Note:** In some cases, these time periods can be of different sizes (or even multiple time periods) but this will cause analysis problems with iDoctor. It is much better to ensure that all data categories have the same the period interval.

,

9.6 Advanced CS Objects

This folder is like the previous folder CS Objects except it provides additional details about the collections listed.

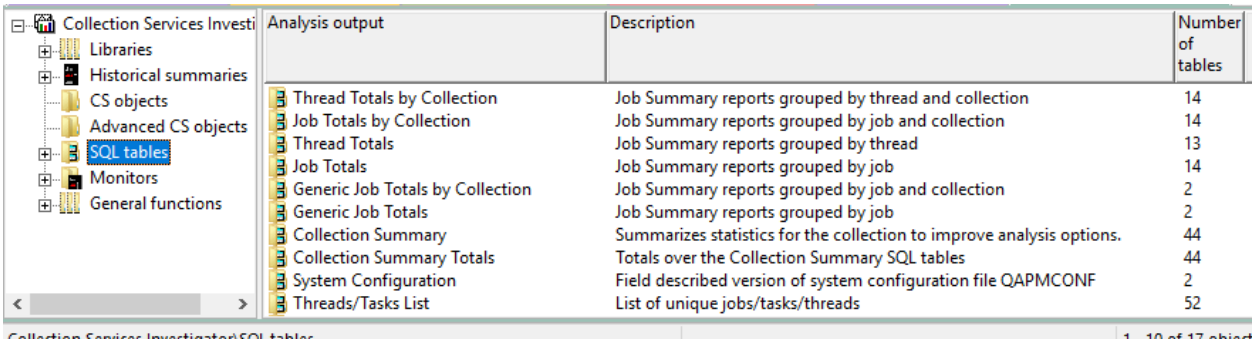


Advanced CS Objects folder

This view contains the collection interval and retention period columns for each management collection object. See the previous section for more details about the menu options you can perform against management collection objects.

9.7 SQL tables

This folder contains all the SQL tables that exist on the system generated by Collection Services Investigator analyses.



Collection Services Investigator – SQL tables folder

The folder exists in 3 places and each will filter the contents appropriately based on where it is located:

Location	Description
Under Collection Services Investigator	Entire system
Under a library	All collections in the library
Under a collection	Only this collection

For more information see the SQL Tables section in chapter 4.

9.8 Monitors

The Monitors folder allows for 24x7 data collection for PEX, Disk Watcher or Job Watcher collections. This folder does NOT contain any data for Collection Services collection.

Collection Services Investigator provides an option to summarize Collection Services data in order to more easily view data over long time periods. See the Historical Summaries folder for more information.

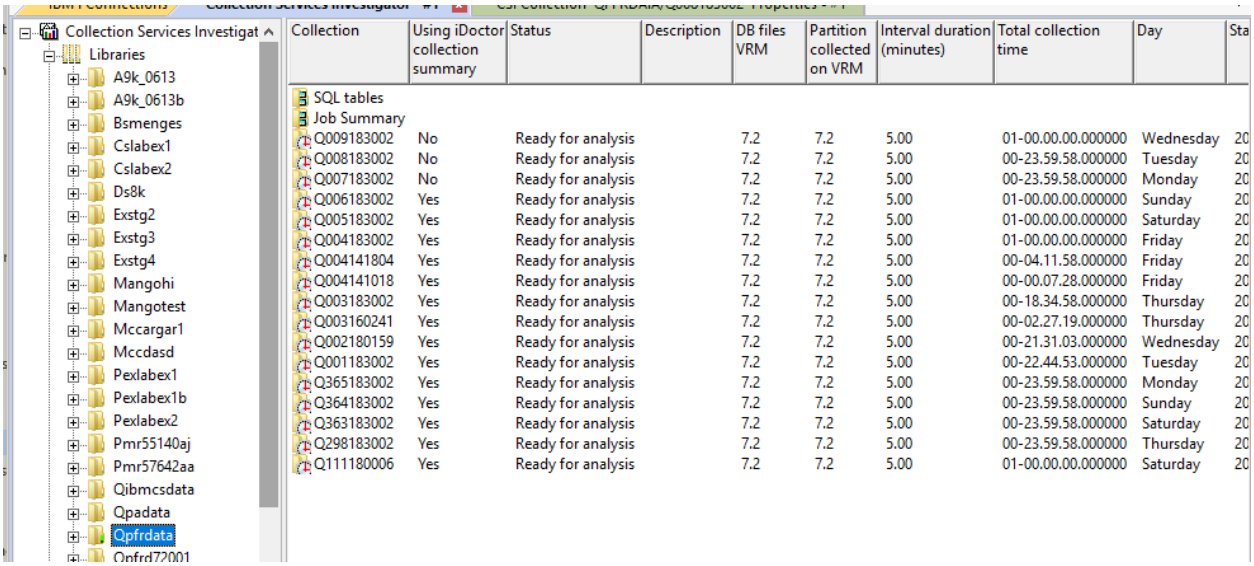
For more information about the Monitors folder see the Monitors section in chapter 4.

9.9 Collections

Moving down the tree within each Library folder are one or more collections that have been created within the current library.

Creating collections must be done using the CRTPFRDTA command (from a Collection Services management collection object) or your system needs to be setup to do this for you automatically. See the IBM i Performance Tools documentation for more information.

Collection Services collections can also be created from *MGTCOL objects viewed from the CS Objects folder.

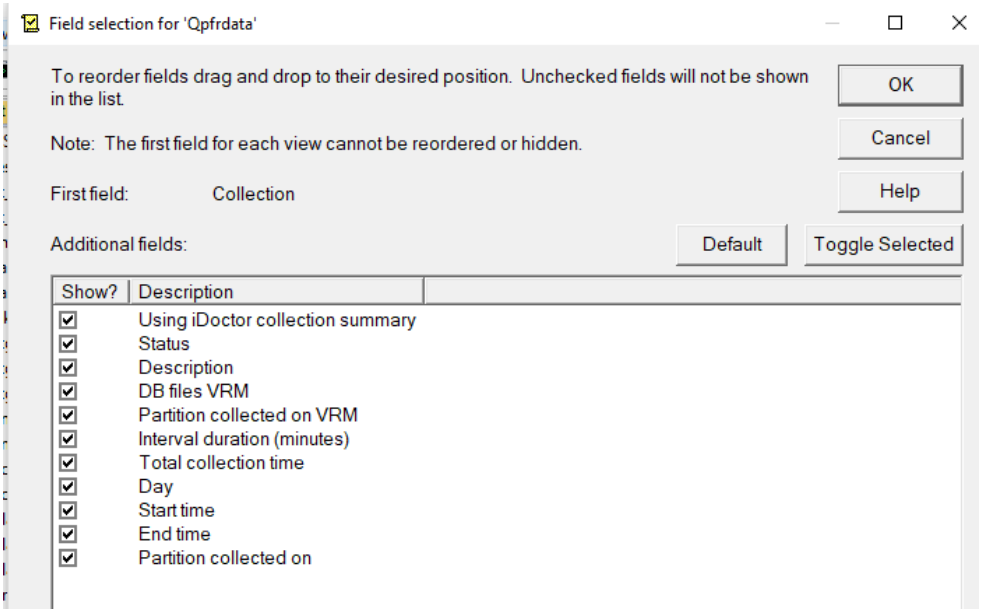


Collection Services Investigator Collections in a Library

9.9.1 Collection Fields

The list of collections displays the collection name, description, status as well as several additional fields.

Each collection in the list has a set of fields available which can be optionally reordered and displayed. To change the current field selections for the collection list, use the Select fields... menu from the library folder. A listing of the available fields and a short description is provided in the table below:



Field selection window

Field	Description
Collection	Name of the collection. This name matches the member name used in the database files named QAPM* that exist in the current library.
Using iDoctor collection summary	Indicates if the collection is using data generated by the iDoctor collection summary analysis.
Status	This column indicates if all required collection services DB files exist. If some required files are missing, then some graphing options or analyses may not be available or may not work.
Description	A text description given to the collection.
DB files VRM	The version of the database files. If this doesn't match the partition collected on VRM it means the files have been converted to a different release which sometimes causes analysis issues. **Tip:** Don't convert Collection Services files from one release to another especially if you want to analyze wait buckets. This can cause lots of issues.
Partition collected on VRM	The version of IBM i that was used to create this collection.
Interval duration	The size of each interval produced in the database files (in minutes).
Day	The day of week in which the collection **started**.
Start time	The date and time the collection started.
End time	The date and time the collection ended. **Note:** If the collection is still running this is by default 24 hours from the collection start time which may not have happened yet.
Partition collected on	The name of the partition the collection was created on.

9.9.2 Menu Options

The table below outlines the different types of operations that may be performed by right clicking on a collection within the component view.

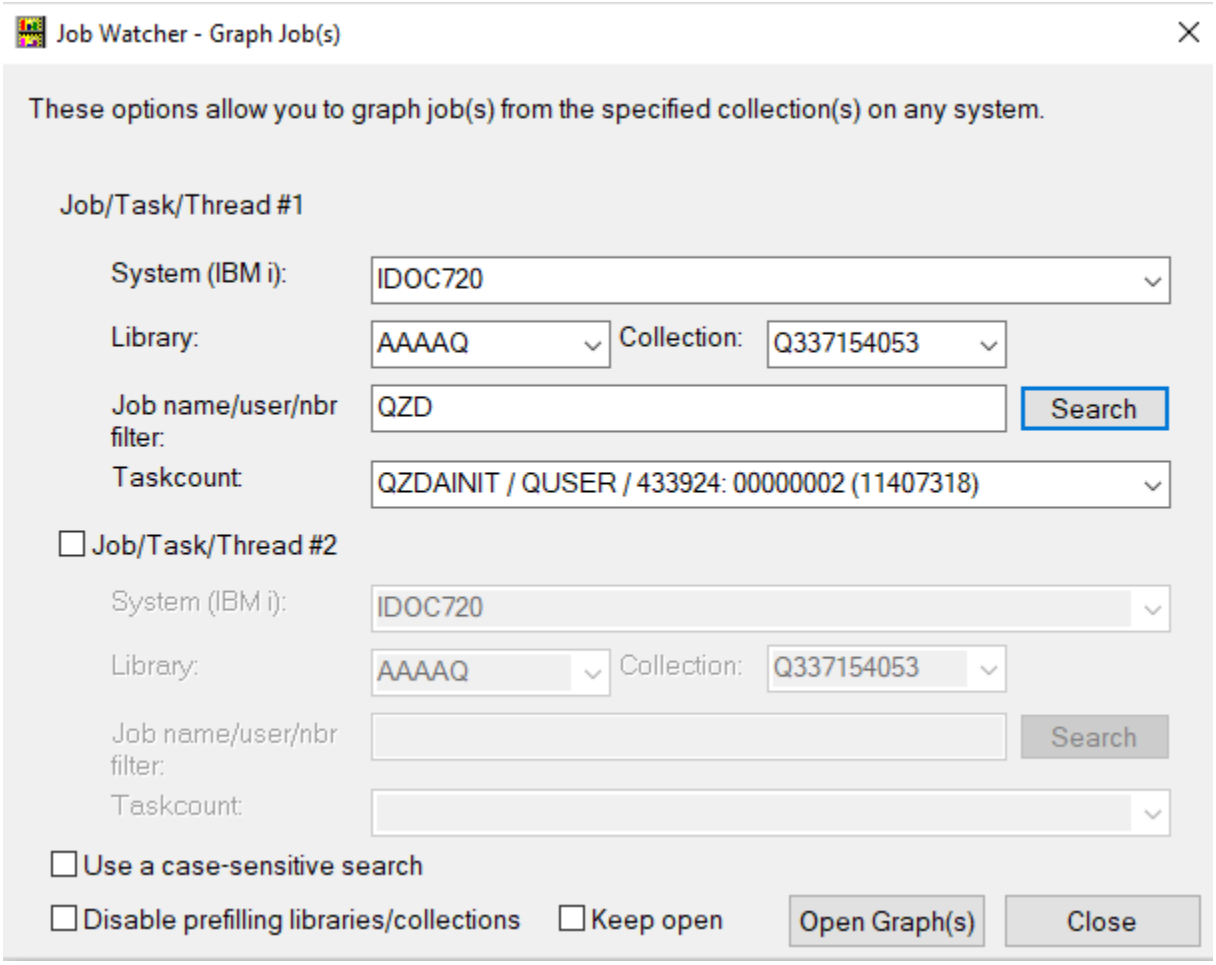
Menu Item	Description
Explore	Displays the contents of the collection folder in the right pane of the Collection Services Investigator component view.
Record Quick View	Displays the fields for a collection in the list view vertically for easier viewing. Not available from the tree side.
Analyses -> Analyze Collection	Displays the Analyze Collection window showing the available analyses that can be ran against the desired collection(s). Data generated by these analyses are stored in SQL tables which are accessible under the SQL Tables folder.
Analyses -> Run ALL default analyses	If checked, the Run ALL default analyses option will be used. All default analyses will be executed for each collection after it completes. This is NOT all analyses but only a select few that are most commonly needed.
Analyses -> Run XYZ	This lists you run a specific analysis which varies by component and VRM of the collection. See the Analyses section for a list of those available.

Favorites	This list of graphs are the ones most commonly used and are great starting points.
Wait graphs	Contains collection-wide wait summary graphs either by time interval or by thread. If unsure of where to investigate first, the Collection overview time signature under the by time interval subfolder is the best place to start.
CPU graphs	Contains collection-wide CPU graphs showing CPU utilization along with Dispatched CPU and CPU queuing.
System graphs	
Memory pool graphs	These graphs show memory pool consumption in various ways by time interval.
Job counts graphs	These graphs help you see how many total jobs / tasks and threads existed on the system during the collection. Note that the counts given for each type are normally less than they truly were because many jobs/tasks/threads never use CPU during the collection and therefore the type of work (job, task or thread) running is unknown. These are added up together in an "Always idle jobs/threads/tasks" bucket. However, if the collection uses the "force 1st interval" option then this problem will go away. But use caution as this option can take a long time in some environments and may not be advised. Contact support first if you need advice on this! **Note:** This folder will NOT appear until the Collection Summary analysis has been ran on the collection.
Temporary storage	These graphs provide metrics related to job temporary storage allocations or temporary pages allocated. **Note:** This folder only appears at 7.2+.
I/O and memory page graphs	This option provides collection-wide summary graphs showing IO operations and disk activity by time interval.
Hardware	
Disk configuration	The disk configuration reports show the layout of disk configuration by ASP -> IOP -> IOA -> Unit.
Disk graphs	The disk graphs show many views over the disk statistics with rankings by unit, path, IOA, IOP, ASP and disk type.
IFS graphs	This folder provides collection-wide summary graphs showing IFS activity by time interval.
Communications graphs	The communication graphs includes IOP utilization graphs, socket statistics graphs and SSL handshaking statistics graphs.
JVM graphs	These graphs display J9 JVM statistics.
SQL graphs	These graphs display SQL statistics but only if the appropriate data has been collected. **Tip:** For more information about enabling this data: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM+i+Technology+Updates/page/Job+Level+SQL+Metrics
Other graphs	This option provides collection-wide summary graphs showing other types of information such as state transitions and transactions by time interval.
PT1 reports	
Collection size	These reports are used to display information about the size of each collection.

Graph Job(s)	This option allows you to search for or specify a job and graph it over time using the wait bucket data. You can also specify a 2nd job in order to do a comparison of 2 jobs at once.
Search...	Performs a search over the entire collection looking for a specific piece of data specified by the user.
Generate Reports…	This option can be used to build a report of the desired set of Job Watcher tables and graphs. The report consists of a screenshot of each graph along with its title and collection information. The reports are built into a HTML page and displayed in the web browser when completed.
Launch Workload Estimator	This option performs a summary of the CPU and disk data found in the current collection and then sends it to WLE (Workload Estimator) for further analysis.
Copy URL	Creates a link to the component, library and collection that can be accessed later, or sent to another user.
Copy...	Copies one or more collections to another library. Selecting multiples is only available from the list side of the Job Watcher component view.
Delete...	Deletes a collection. Select multiple collections in order to delete more than one at a time. Selecting multiples is only available from the list side of the Job Watcher component view.
Rename	This option will rename the collection in the current library.
Save	Saves the selected collections to a save file on the system. The save file will be added to the list under the Saved collections folder under the Job Watcher icon.
Transfer to...	FTP one or more collections to the PC, another IBM, another server or to IBM. Selecting multiples is only available from the list side of the Job Watcher component view.
Properties	Use this menu to display the property pages for the collection. The property pages provide quick access to additional summary information about the collection.

9.9.3 Graph Jobs

This option allows you to search for or specify a job and graph it over time using the wait bucket data. You can also specify a 2nd job in order to do a comparison of 2 jobs at once. The data can come from two different systems if desired.

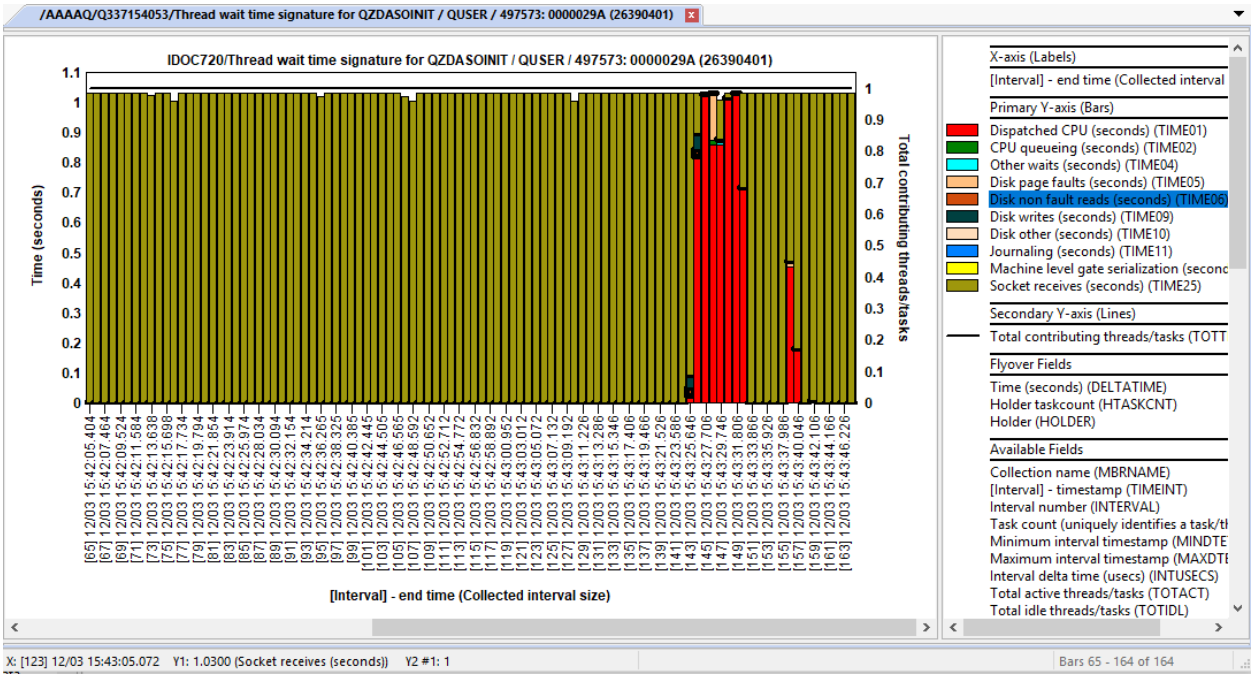


Job Watcher – Graph Job(s) window

Option	Description
Job #1 - System (IBM i)	The name of the IBM i to use for Job number 1.
Job #1 - Library	The Job Watcher collection library name. The drop-down list contains all libraries on the system that contain Job Watcher data (unless the disable prefilling libraries/collections is checked!)
Job #1 - Collection	The Job Watcher collection name to use for job #1. The drop-down list contains all collections found in the specified library.
Job #1 - Job name/user/nbr filter	This option allows you to enter in part of the job name, user, number or thread ID to search for. Pressing the Search button will build the list of taskcounts (jobs) that match the search criteria. Leave the value blank to return the 1st 1000 taskcounts in the collection.
Job #1 - Taskcount	The list of taskscounts matching the search criteria in the Job name/user/nbr filter field.

Note: The same information applies for Job #2 (see the previous table.) By selecting 2 jobs you will be automatically presented with a comparison graph of both jobs together over time.

Option	Description
Use a case-sensitive search	This option is used to perform a case-sensitive search. For some system task names this is required.
Disable prefilling libraries/collections	In some environments if you have many collections (hundreds) the drop down lists that prefilter the libraries and collections may be too slow to work effectively. Check this box and the drop-down lists will no longer be prefilled with data after making changes on this interface.
Keep open	This option if checked will cause the graph(s) to be created but this window will continue to remain open. This can be useful if you wish to kick off the graphs for multiple jobs at once before reviewing them.
Open graph(s)	This will open the specified graph's Thread wait time signature graph for the job(s) indicated.



Thread wait time signature graph opened from the Graph Job(s) window

Menu Item	Description
Explore	Displays the contents of the collection folder in the right pane of the component view.
Record Quick View	Displays the fields for a collection in the list view vertically for easier viewing. Not available from the tree side, only the list side.

Analyses -> Analyze Collection	Displays the Analyze Collection window showing the available analyses that can be ran against the desired collection(s). Data generated by these analyses are stored in SQL tables which are accessible under the SQL Tables folder.
Analyses -> Run Collection Summary	This analysis summaries the data in the collection in order to improve graphing performance and provide more options to group and manipulate the data. Job and wait bucket statistics are added up on a per interval basis and metrics from file QAPMJOBWTG are expanded into an iDoctor SQL table (QAIDRCSGAP_<<COLNAME>>) for easier processing. Note: When running this option you will be prompted for any desired filtering (time, job, subsystem, etc) you wish to perform against the data. Filtered data can be analyzed under the SQL Tables interface.
Analyses -> Run XYZ	The rest of the list of analyses is described in the Collection Services Investigator Analyses section.

Wait graphs	Contains collection-wide wait summary graphs either by time interval or by thread. If unsure of where to investigate first, the Collection overview time signature under the by time interval subfolder is the best place to start. This folder also provides reports on workload capping and seizes and locks (if the appropriate data has been collected.)
CPU graphs	Contains collection-wide CPU graphs showing CPU utilization along with Dispatched CPU and CPU queuing.
Memory pool graphs	These graphs show memory pool consumption in various ways by time interval.
Job counts graphs	These graphs provides counts of jobs/tasks/threads in several ways.
I/O and memory page graphs	This option provides collection-wide summary graphs showing IO operations and disk activity by time interval.
Disk configuration	The disk configuration reports show the layout of disk configuration by ASP -> IOP -> IOA -> Unit.

SSD candidate screening	These reports are designed to help a user determine if the current system is likely to benefit from having SSDs installed.
Disk graphs	The disk graphs show many views over the disk statistics with rankings by unit, path, IOA, IOP, ASP and disk type.
IFS graphs	This option provides collection-wide summary graphs showing IFS activity by time interval.
Communications graphs	The communication graphs includes IOP utilization graphs, socket statistics graphs and SSL handshaking statistics graphs.
Other graphs	This option provides collection-wide summary graphs showing other types of information such as state transitions and transactions by time interval.
Virtual I/O graphs	This graphs displays virtual I/O statistics.
JVM graphs	These graphs display J9 JVM statistics.
SQL graphs	These graphs display SQL statistics but only if the appropriate data has been collected. **Tip:** For more information about enabling this data: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM+i+Technology+Updates/page/Job+Level+SQL+Metrics

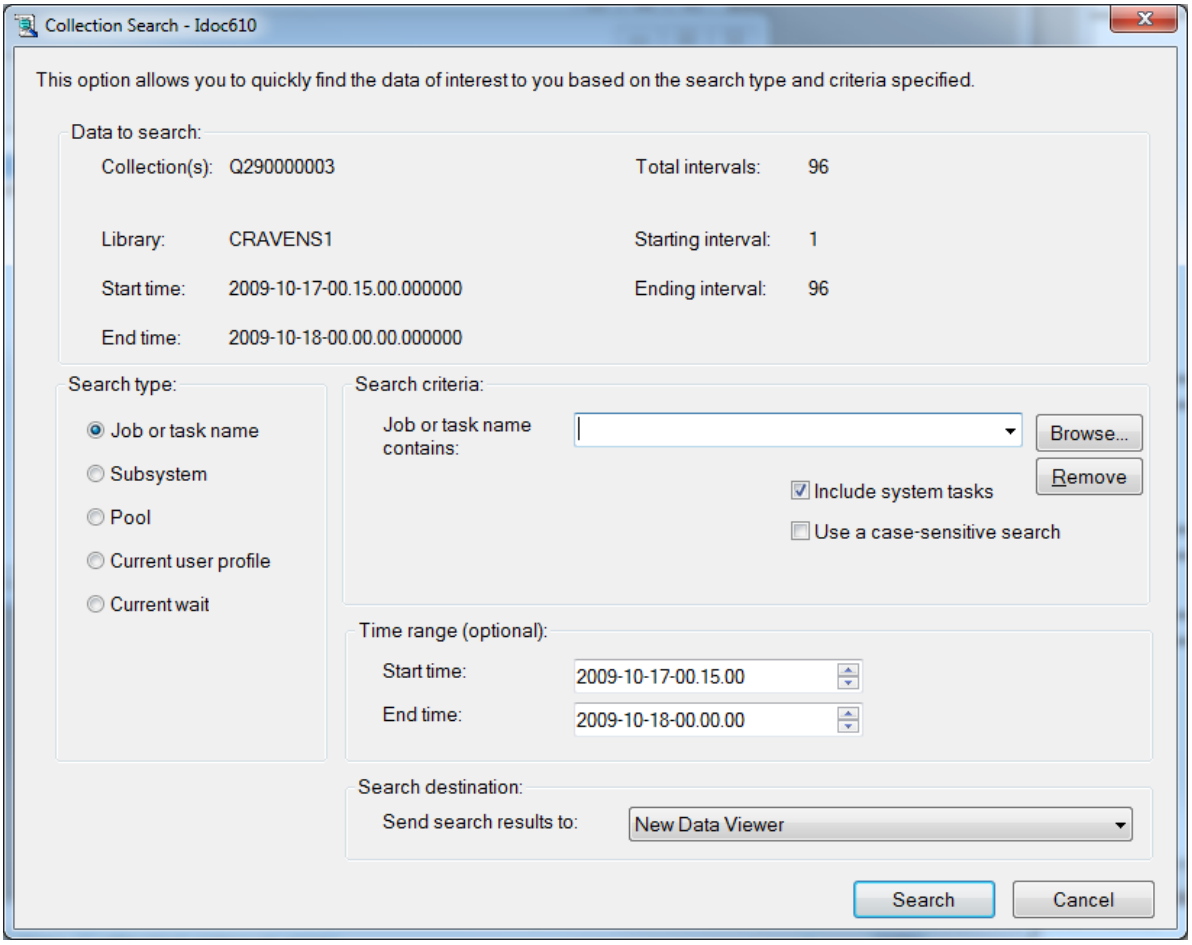
Search...	Performs a search over the entire collection looking for a specific piece of data specified by the user. If you select multiple collections before right-clicking it will search all of them.
Generate Reports…	This option can be used to build a report of the desired set of tables and graphs. The report consists of a screenshot of each graph along with its title and collection information. The reports are built into a HTML page and displayed in the web browser when completed.
Launch Workload Estimator	This option performs a summary of the CPU and disk data found in the current collection and then sends it to WLE (Workload Estimator) for further analysis.
Copy URL	Creates a link to the component and library that can be accessed later, or sent to another user.
Copy...	Copies one or more collections to another library. Selecting multiples is only available from the list side of the component view.
Delete...	Deletes a collection. Select multiple collections in order to delete more than one at a time. Selecting multiples is only available from the list side of the component view.
Save	Saves the selected collections to a save file on the system. The save file will be added to the list under the Saved collections folder.
Transfer to...	FTP one or more collections to another system. Selecting multiples is only available from the list side of the component view.
Properties	Use this menu to display the property pages for the collection. The property pages provide quick access to additional summary information about the collection.

9.9.4 Search

The Search function in Collection Services Investigator allows the user to look for a known job name, subsystem, pool, user profile, etc in order to build a report for the detailed data found in the desired collection(s). The Search window offers a browse function so the unique values found in the collection for each type can be selected from if desired.

You can search over a single collection in the library or **multiple collections** in the same library if you select multiples before right-clicking them and then use the Search menu.

An example of this interface is:



Collection Search Window

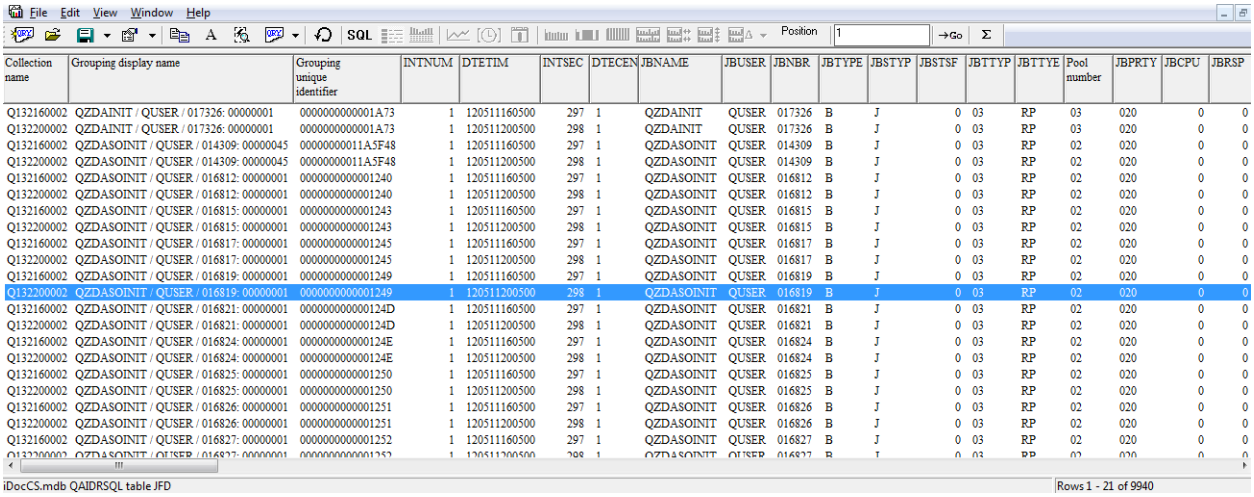
The following table describes the fields in the Collection information frame:

GUI Element	Description
Collection(s)	Name of the collection(s) to search.
Library	Library name the collections reside in.
Start time	The date and time the earliest collection started.
End time	The date and time the oldest collection ended.
Total intervals	Total number of intervals found in the collection(s).
Starting interval	The smallest interval number found in the collection(s).
Ending interval	The highest interval found in the collection(s).
Search type	The search type allows you to pick which kind of data you want to search for. Changing the selection will change the fields shown in the Search criteria area of the window as appropriate.
Search criteria	The content of this section varies depending on the search type selected. Generally you can use the Browse option to find the possible values in the collection(s) for the desired search type. The search drop down lists contain search values used from previous searches. By selecting one and pressing the Remove button you can remove the entry from the list.
Include system tasks	This option is used to include or exclude system tasks from the search results when searching by Job name, subsystem or pool. The default value is to include system tasks in the search results.
Use a case-sensitive search	If you need to search on a mixed case system task name, then check this box.
Time range	The time range fields allow you to narrow you search based on the desired time range. The default time values shown match the start time and end times shown at the top of the window.
Search destination	This drop down list allows you to pick which Data Viewer to send the search results to if multiple Data Viewers are active.

The Search criteria section allows you to enter the values appropriate for the search type selected. The fields available to search on change based on search type picked.

The Time range section allows you to narrow the search to only a specific time period. You may use this for example if a user reports a system slow down within a certain time for example.

For example performing a search using search type “Job or task name” with the value QZDA would give a report similar to this:

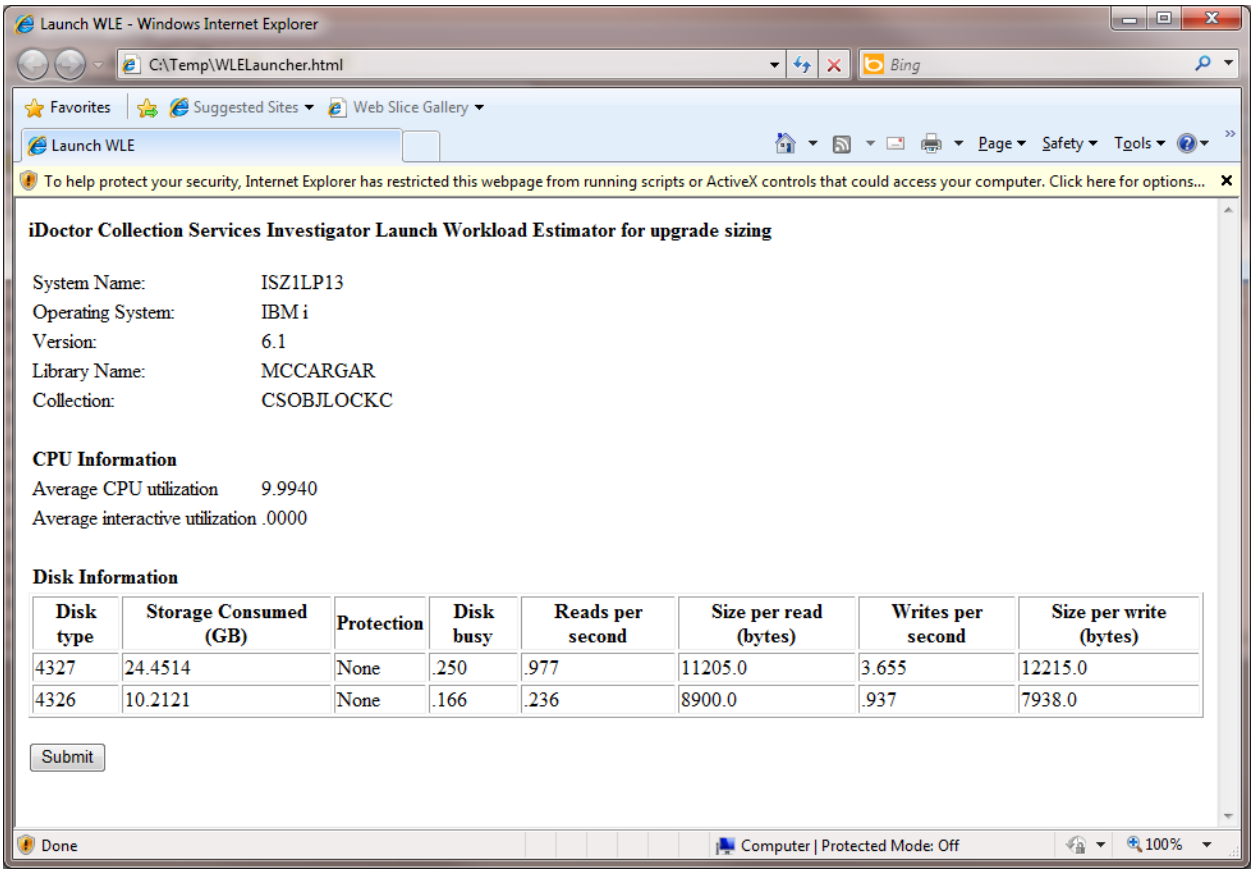


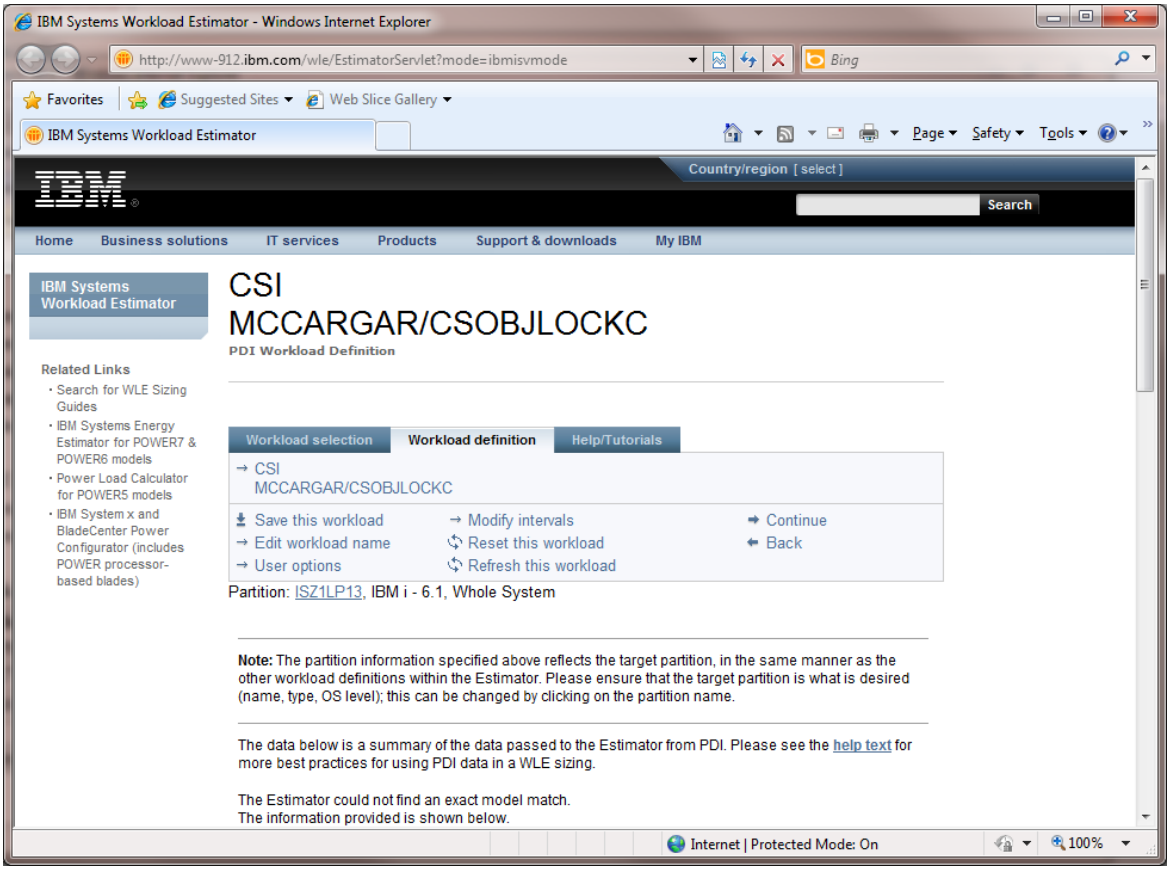
From this report there are drill down options available to view graphs for any job and interval selected. For this example, right click and choose an option under the “Selected thread” menu.

9.9.5 Launch Workload Estimator

This option summarizes the CPU and disk data in the selected collection and then opens a web browser that will send the data to WLE (Workload Estimator) for further analysis.

An example of this option in action is shown below:





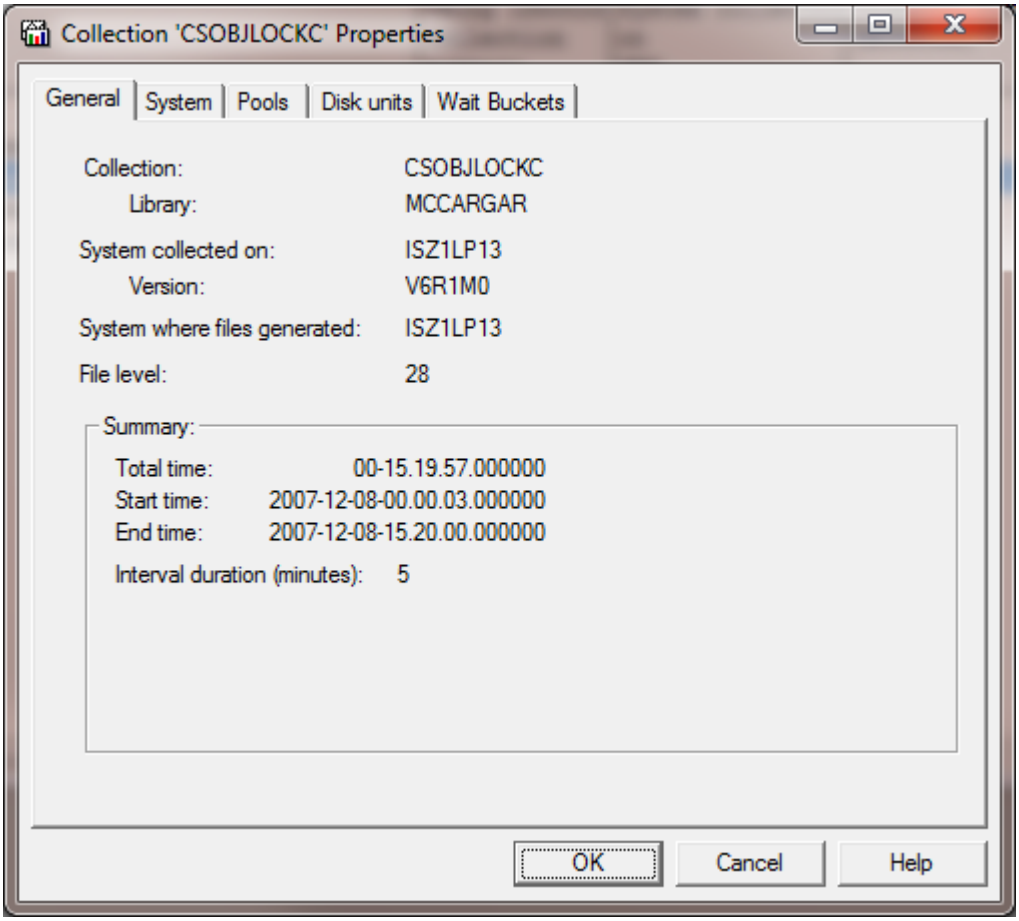
Launch Workload Estimator Example

9.9.6 Properties

This section covers the property pages for a collection. Access the property pages by right clicking on a collection and choosing the Properties menu.

9.9.6.1 General

The General property page provides basic information about the collection such as when it was created and what system.



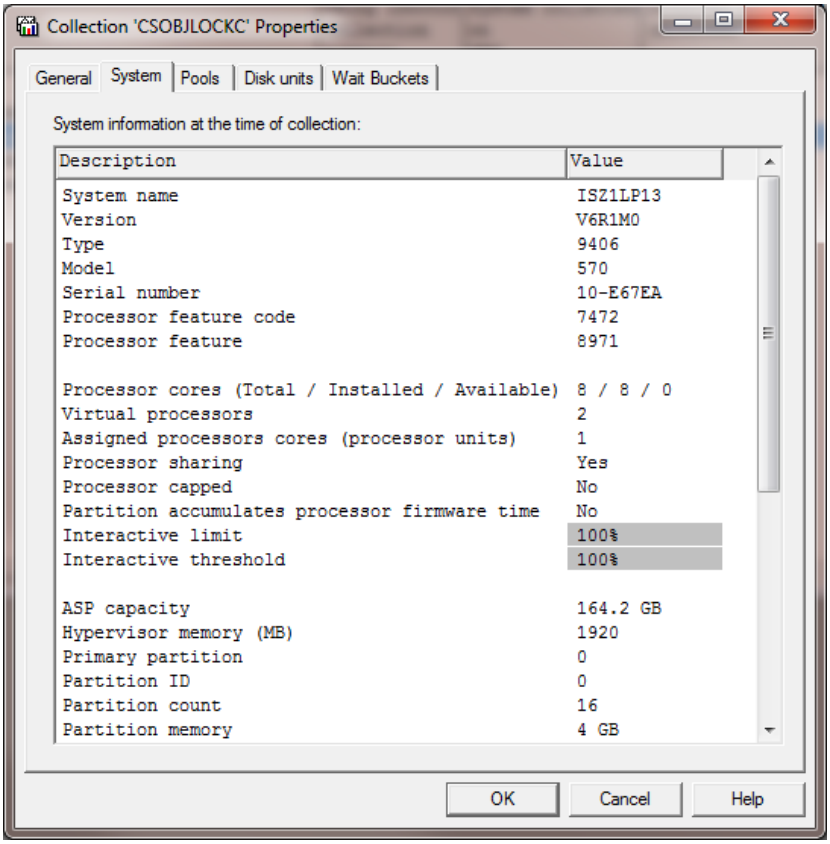
Collection Properties - General

The following information is displayed on the General property page:

GUI Element	Field Description
Collection	Name of the collection. This matches the member name used in the QAPM* files on the server in the library specified.
Library	Library the collection resides in.
System collected on	Name of the system that collection was created on.
System collected on VRM	VRM of the system the collection was created on.
File level	Version of the Collection Services files used to create the collection.
Total collection time	The total duration of the collection data in timestamp format. **Example:** 00-10.42.28.000000 is 10 hours 42 minutes, 28 seconds.
Start time	The date/time the collection started..
End time	The date/time the collection ended.
Interval duration	The size of each interval produced in the database files (in minutes).

9.9.6.2 System

The System page provides details about the system at the time the collection was started (as described in file QAPMCONF).

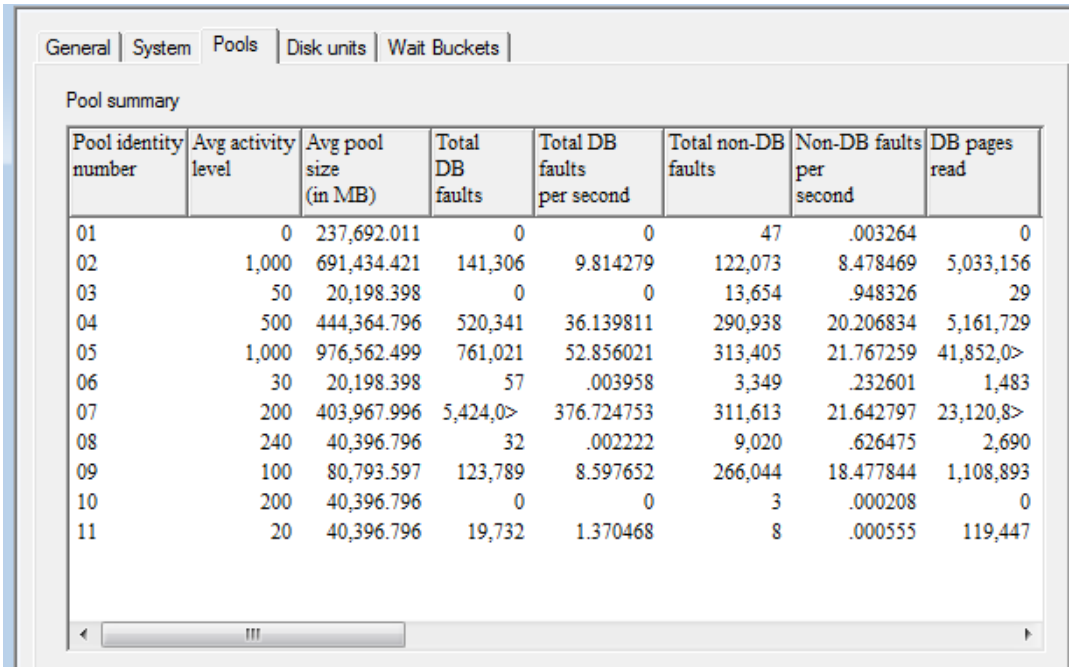


Collection Properties – System

Information in this window can also be created into a field-described SQL table using the Analyses -> Run System configuration menu for a collection.

9.9.6.3 Pools

The pools page displays a report of the memory pools that exist on the system. Similar information can also be displayed using the Memory pools graphs under the collection.



General | System | Pools | Disk units | Wait Buckets

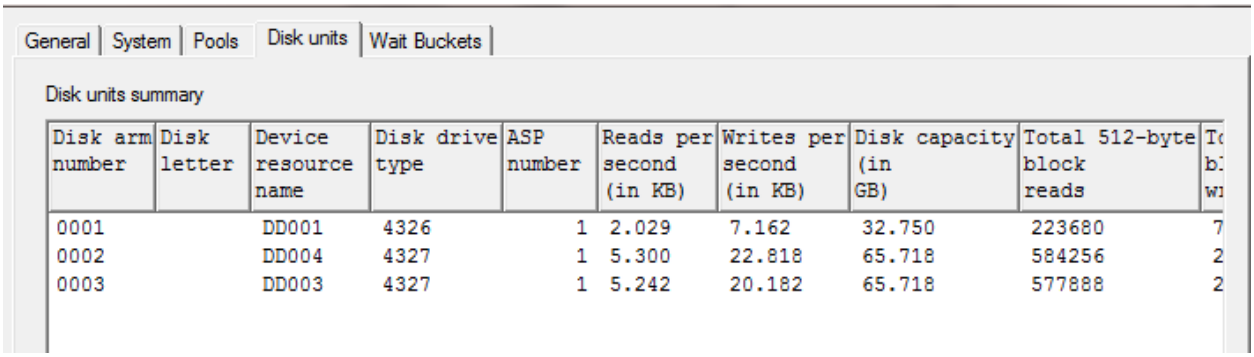
Pool summary

Pool identity number	Avg activity level	Avg pool size (in MB)	Total DB faults	Total DB faults per second	Total non-DB faults	Non-DB faults per second	DB pages read
01	0	237,692.011	0	0	47	.003264	0
02	1,000	691,434.421	141,306	9.814279	122,073	8.478469	5,033,156
03	50	20,198.398	0	0	13,654	.948326	29
04	500	444,364.796	520,341	36.139811	290,938	20.206834	5,161,729
05	1,000	976,562.499	761,021	52.856021	313,405	21.767259	41,852,0>
06	30	20,198.398	57	.003958	3,349	.232601	1,483
07	200	403,967.996	5,424,0>	376.724753	311,613	21.642797	23,120,8>
08	240	40,396.796	32	.002222	9,020	.626475	2,690
09	100	80,793.597	123,789	8.597652	266,044	18.477844	1,108,893
10	200	40,396.796	0	0	3	.000208	0
11	20	40,396.796	19,732	1.370468	8	.000555	119,447

Collection Properties – Pools

9.9.6.4 Disk units

The Disk units page displays a report over the disk units captured in the collection. The disk graphs under the collection can also be used to produce similar statistics.



General | System | Pools | Disk units | Wait Buckets

Disk units summary

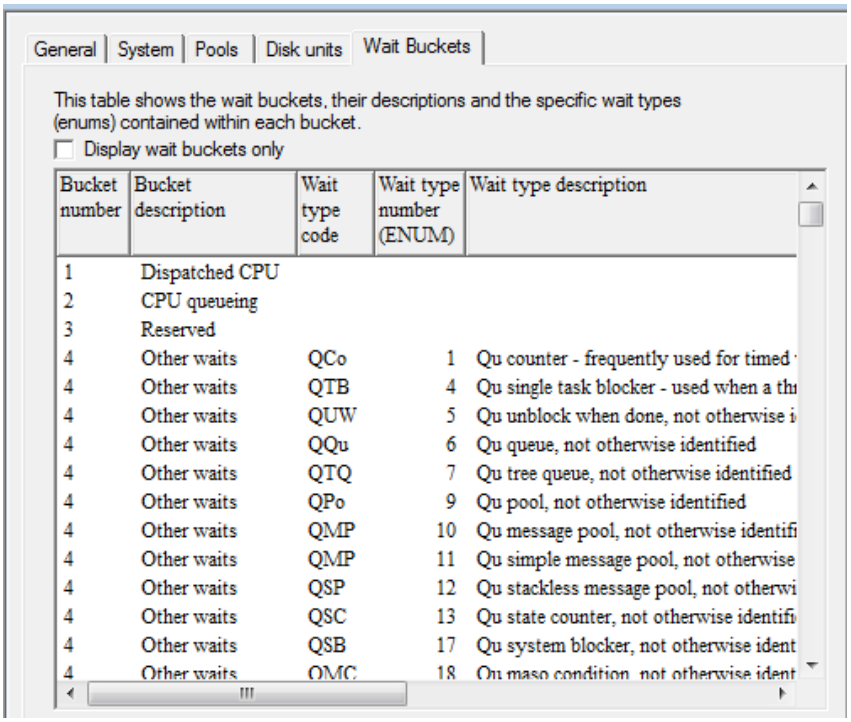
Disk arm number	Disk letter	Device resource name	Disk drive type	ASP number	Reads per second (in KB)	Writes per second (in KB)	Disk capacity (in GB)	Total 512-byte block reads	To b wr
0001		DD001	4326	1	2.029	7.162	32.750	223680	7
0002		DD004	4327	1	5.300	22.818	65.718	584256	2
0003		DD003	4327	1	5.242	20.182	65.718	577888	2

Collection Properties – Disk units

9.9.6.5 Wait Buckets

At V6R1, the wait buckets defined in the system in IBM i changed so that both Collection Services and Job Watcher utilize the same 32 wait buckets. A new bucket was also added for PASE to keep track of time spent in J9.

The wait bucket page displays the wait bucket and enums that were used during creation of the collection. These are the building blocks for the wait graphs shown in Job Watcher and Collection Services Investigator.



Collection Properties – Wait Buckets

Each specific type of wait is identified by an enum (a wait point on the system) and each enum is given a wait bucket. We can tell how much time was spent in each wait bucket for each thread during each interval. We can also tell what enum (wait) each thread was in at the end of interval and how long the thread was in that wait (the current wait).

For more information about Wait Buckets, see the Job Watcher white paper:

http://public.dhe.ibm.com/services/us/igsc/idoctor/Job_Waits_White_Paper_61_71.pdf

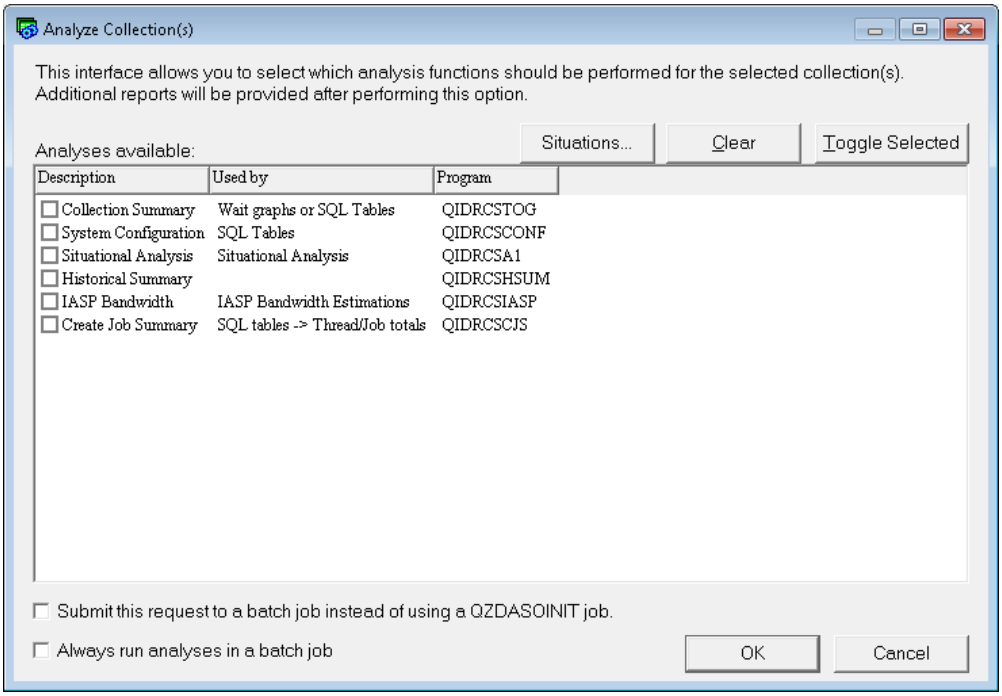
9.10 Analyses

The available Analyses in Collection Services Investigator and what they provide is described in this section.

All analyses are written as SQL stored procedures and are initiated from the Analyses menu after selecting one or more collections and right-clicking. Each analysis has a 'fast path' option that allows it to be ran without visiting the Analyze Collection window.

9.10.1 Analyze Collection Window

The Analyze Collection window presents the user with a list of available analyses that can be ran over the currently selected collection(s). It is opened using the Analyze -> Analyze Collection... menu.



Analyze Collection(s) Window

Each available analysis is presented to the user on this screen. Special options for Situational Analysis such as creating your own situations or modifying the parameters used by the IBM defined situations are accessible by clicking the Situations button.

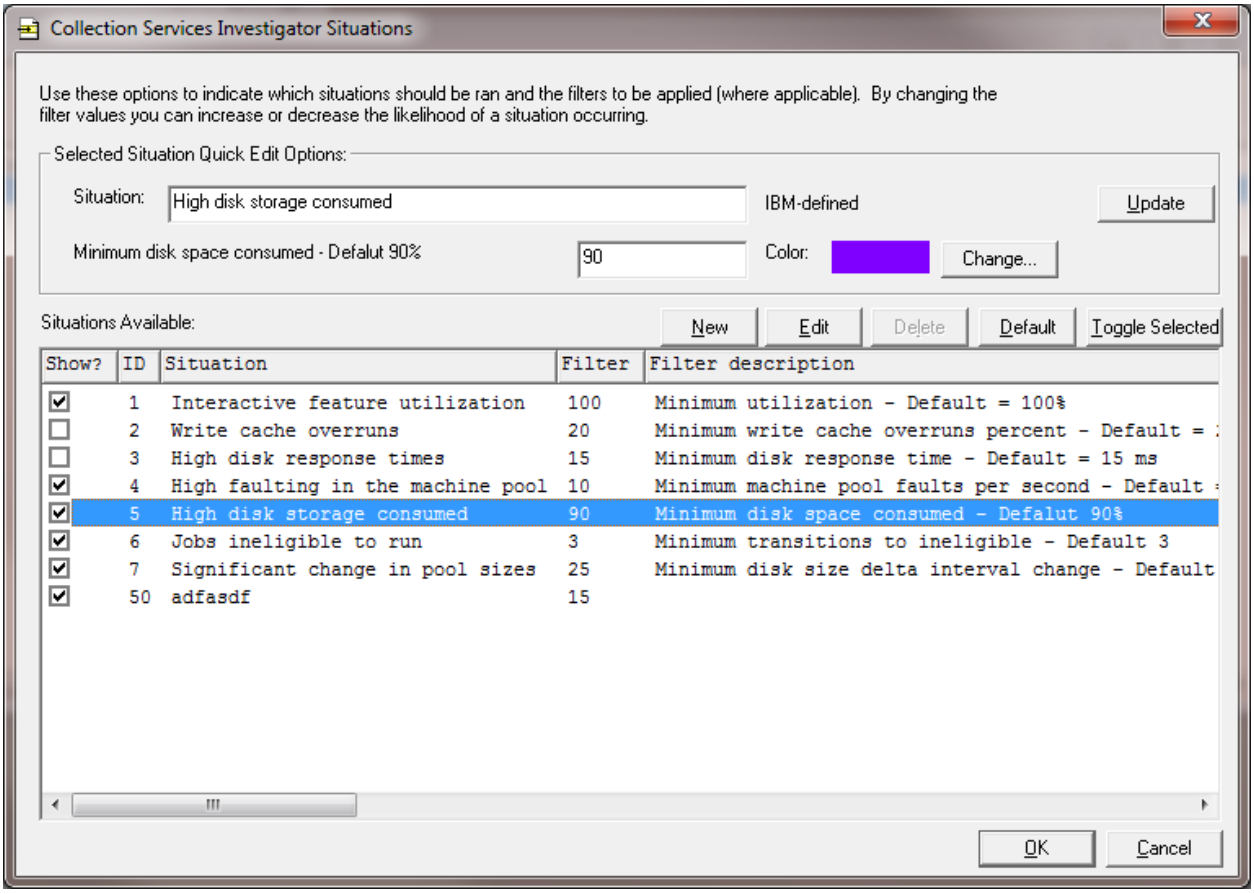
The controls on this interface and what they do is described in more detail in the following table:

Control	**Description**
Situations... button	Opens the Situations Window which allows the user to modify the parameters used by the IBM-defined situations or create new ones.
Clear button	This button unchecks all analyses that have been selected.
Toggle selected button	This button changes the checked state of all selected analyses in the list.
Analyses available list	This is the list of analyses currently available. The analyses available can vary from collection to collection depending on the OS VRM of the system the data was collected on and the data found in the collection. Checking an analysis name indicates that it will be ran when the OK button is pressed.
Submit this request to a batch job	If this option is used an SQL script will be created on the server and ran on the server in a new submitted job. The default behavior is to run the analyses from the Remote SQL Statement Status View which uses a QZDASOINIT job created by the GUI. Though this gives better status of the progress of the analysis processes, it is not ideal if the analysis is expected to take a very long time as closing the GUI would end the analysis processing as well.
Analyses run analyses in a batch job	This option is a preference linked with Preferences -> Miscellaneous tab -> “Always run analyses in a batch job”. If checked the analysis will run in a batch job instead of a Remote SQL Statement Status View.

9.10.1.1 Situations Window

The Collection Services Investigator Situations Window allows the user to control parameters used by the IBM defined situations or to create brand new user-defined situations against the Collection Services data. Using this interface you can also control which situations should be ran.

An example of this interface is:



Collection Services Investigator Situations

The controls on the interface above is described in the table below:

Control	Description
Selected situation quick edit options	This area contains controls that allow you to modify some of the attributes for the selected situation such as name, filter value and color.
Situation text box	This field allows the user to modify the name of the situation.
Update button	This button will save any changes made within the Selected situation quick edit options frame to the selected situation in the list.
Filter value text box	The filter value text box lets you modify the filter's value to use. The filter value replaces the <<FITLER>> parameter marker within the SQL statement.
Color change button	Changes the situations color shown as the background color when graphed. If multiple situations occur in a time period then the color is always red.
Situations available list	This list contains all the IBM-defined and user-defined situations. The show checkbox/column can be used to avoid running certain situations if desired. The ID number is used to uniquely identify each situation. The filter and filter description (if they exist) are used as a parameter to control whether or not a situation is triggered when the SQL query behind the situation is executed. Color column identifies the color of the situation. SQL column shows the complete SQL statement for each situation. It may be modified by pressing the Edit button.
New button	The new button displays the Situations Editor window which allows you to create your own situation.
Edit button	The edit button displays the Situations Editor window and fills in the information for the current situation.
Delete button	This button lets you delete the currently selected user-defined situations. IBM-defined situations cannot be removed.
Default button	This button removes all changes made to the IBM-defined situations and restores them to their original (shipped) state.
Toggle selected button	This button changes the checked state of all selected items in the list.

9.10.1.2 Situations Editor

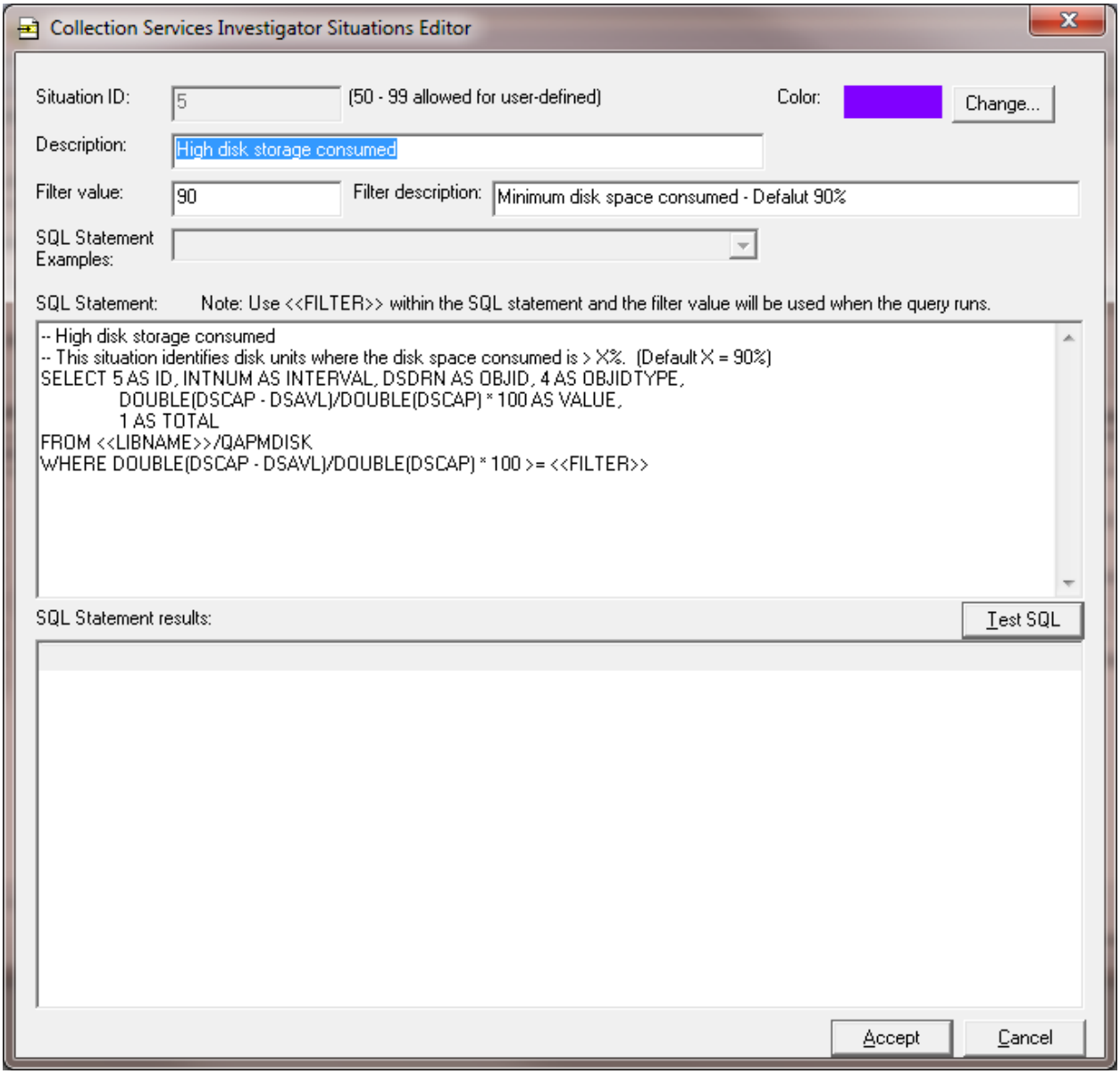
The Situations Editor window is used to create a new situation or modify an existing one. It allows the user to modify a situation to suit their individual needs. Situations are built from a special SQL that meets certain characteristics:

It must contain the following 6 fields (in this order):

1. ID = situation ID
2. INTERVAL = interval number when the situation occurred
3. OBJID = unique identifier for the job/task, disk arm, disk resource, memory pool etc. Use value of 0 if the situation applies to the entire collection.
4. OBJIDTYPE = type of value in the OBJID field
5. VALUE = the actual value of the metric that exceeded the situations filter (if applicable)
6. TOTAL = The number of jobs/threads that experienced this situation if the situation applies to the entire collection. Otherwise a value of 1 must be used.

For testing purposes, use the Test SQL button to preview results of your SQL statement before saving.

An example of this window looks like this



Collection Services Investigator Situations Editor

Control	Description
Situation ID	The situation ID must be unique and needs to be between 50-99 for user-defined situations.
Color change button	Changes the situations color shown as the background color when graphed. If multiple situations occur in a time period then the color is always red.
Description text box	This field allows the user to modify the name of the situation.
Filter value text box	The filter value text box lets you modify the filter's value to use. The filter value replaces the <<FITLER>> parameter marker within the SQL statement.
Filter description	The filter description describes the filter that has been included in the SQL statement.
SQL Statement examples	The SQL statement examples drop down box contains a list of all IBM-defined situations. Selecting one of these situations replaces the current SQL statement shown. These examples contain comments and should help you get a better idea on how to create your own situation to suit your needs.
SQL Statement	The SQL statement that performs the testing to see if the situation has been satisfied in the data. Collection Services tables should be referred to using <<LIBNAME>>/QAPMJOBMI syntax where <<LIBNAME>> is a parameter marker replaced at runtime with the current library and QAPMJOBMI is the desired Collection Services file. **Note:** Aliases will be automatically created for you to point to the current collection member and do not need to be referred to here. The SQL Statement must contain the following 6 fields (in this order): 1. ID = situation ID 2. INTERVAL = interval number when the situation occurred 3. OBJID = unique identifier for the job/task, disk arm, disk resource, memory pool etc. Use value of 0 if the situation applies to the entire collection. 4. OBJIDTYPE = type of value in the OBJID field 5. VALUE = the actual value of the metric that exceeded the situations filter (if applicable) 6. TOTAL = The number of jobs/threads that experienced this situation if the situation applies to the entire collection. Otherwise a value of 1 must be used.
Test SQL	This button executes the current SQL statement shown above against the current collection. IF any hits are found they will be shown in the SQL Statement results list.
SQL Statement results	This list contains the result set returned by running the SQL Statement shown above.
Accept button	Accepts all changes made and closes the window, returning to the previous window.

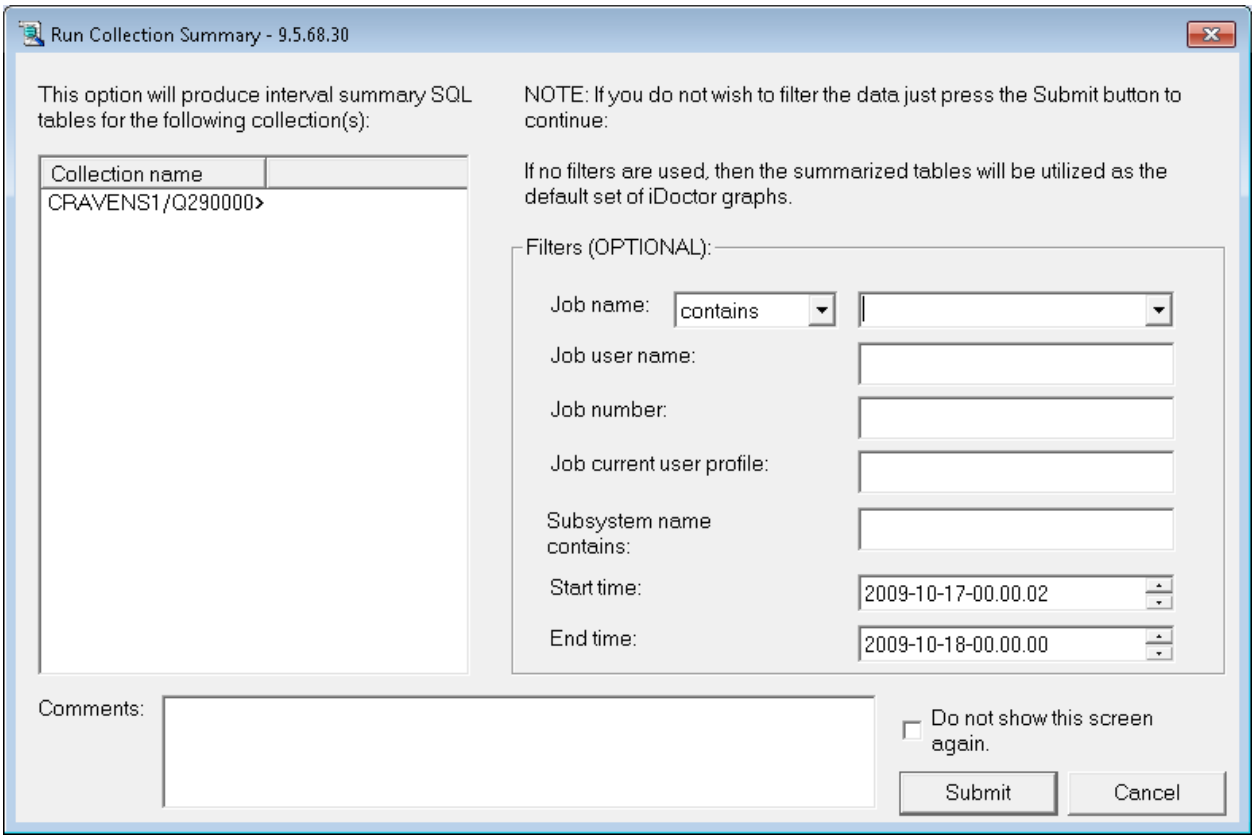
9.10.2 Collection Summary

The Collection Summary analysis (formerly known as the Summarize option) summarizes the job and wait bucket data in order to improve performance of graphs shown in Collection Services Investigator and to offer more graphing options.

9.10.2.1 Run Collection Summary Window

New in fall 2010, is the ability to create a filtered Collection Summary, or one that reduces the data by job, subsystem, time period, etc. Analyzing filtered Collection Summary is possible through the SQL Tables interface. Several graphs are available there by right-clicking the interval summary SQL table generated by this analysis.

An example of the Run Collection Summary window is shown below:



Run Collection Summary Window

This window contains the list of collections to run the analysis over, and several optional filters. You can also attach a comment to all SQL tables generated by the analysis by filling in the comments field. This comment is visible within the SQL tables interface and could be used to help better manage large number of tables.

Control	**Description**
Collection list	This is the list of all collections to be analyzed.
Job name filter	If a value is provided the data will be filtered by the given portion of the job name. Use the drop down list to control whether the job name should "contain" the value anywhere within the job name or if the job name should only "start with" the value given.
Job user name filter	This value allows the user to filter by job user name. Unlike the job name filter this must be an exact match.
Job number filter	Filter the data by job number.
Job current user profile filter	Allows the user to filter the data by current user profile.
Subsystem name contains	This value allows the user to filter on subsystem name.
Start time/end time	These fields allow the user to filter the start and end time of the resulting tables.
Comments	This field contains the comment to apply to all SQL tables generated by the analysis.
Do not show this again	Check this box if you do not wish to see this interface again. Reenable it later via the Preferences -> Confirm -> “Prompt for filtering options when running the Collection Summary Analysis…”

9.10.2.2 Features Offered

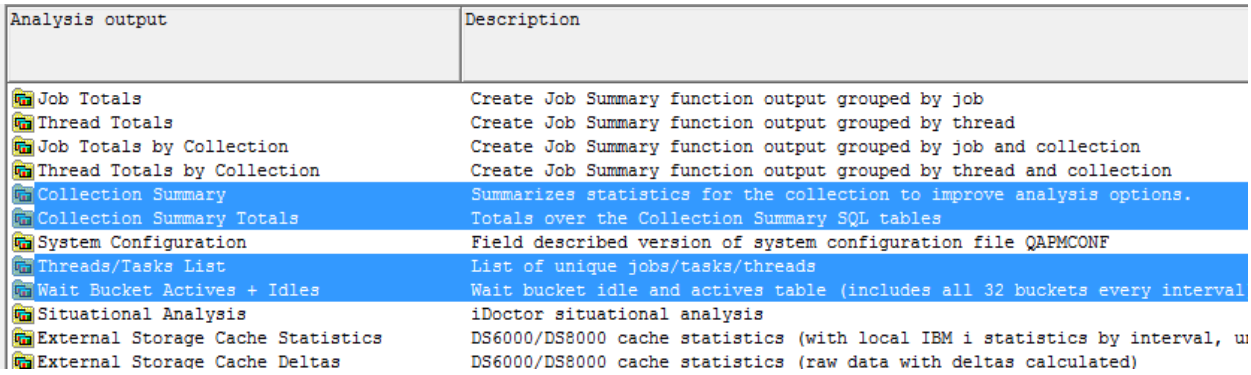
After running the analysis the following features become available in CSI:

1. Graphing multiple collections (select them and right-click to pick the graph)
2. At 6.1+, if QAPMISUM file is not available, the wait graphs will become usable.
3. SQL tables -> Collection Summary -> right-click selected table(s) to access the Wait graphs accessible under the collection but from here the data will be filtered as specified when running the analysis.
4. SQL tables -> Collection Summary Totals -> select 2 tables to perform a wait bucket comparison.
5. SQL tables -> Collection Summary Totals -> Wait graphs (various graphs showing wait buckets)
6. SQL tables -> Collection Summary Totals -> I/O graphs (graphs showing physical disk IO and logical disk IO statistics)

9.10.2.3 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

SQL Table folder	Table description	SQL table
Collection summary	Interval summary	QAIDRCSISUM_<<MBRNAME>> -or- if any parameters are defined then the name becomes: QAIDRCSISUM_<<MBRNAME>>_<<JOBINFO>>_<<CURRENTUSER>>_<<SBS>>_<<STARTTIME>>_<<ENDTIME>>
Collection summary totals	Aggregated interval summary	QAIDRCSISUMTOTALS_<<MBRNAME>> -or- if any parameters are defined then the name becomes: QAIDRCSISUMTOTALS_<<MBRNAME>>_<<JOBINFO>>_<<CURRENTUSER>>_<<SBS>>_<<STARTTIME>>_<<ENDTIME>>
Wait bucket actives + idles	Wait bucket gap file	QAIDRCSGAP_<<MBRNAME>>
Threads/tasks list	Threads/tasks list	QAIDRCSTL_<<MBRNAME>>



The Collection Summary analysis SQL tables output folders (selected) as they appear in CSI

9.10.3 System Configuration

The system configuration analysis produces a field-described QAPMCONF. QAPMCONF is a file that is very difficult to query by a novice SQL writer. After running the analysis, multiple configurations can be compared in order to find differences under the SQL tables -> System configuration folder.

9.10.3.1 Features Offered

After running the analysis the following features become available in CSI:

1) SQL tables -> System configuration -> Select 2 tables and right-click 2 -> compare configurations reports

9.10.3.2 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

SQL Tables folder	Table description	SQL table
Situational Analysis	Configuration information	QAIDRCS_CONFIG_<<MBRNAME>>

9.10.4 Situational Analysis

Situational Analysis is a function in iDoctor that looks for performance problems in a collection as an optional analysis. When ran and if situations (problems) are found in the data, they will be highlighted on the collection's high level (over time) graphs. Each situation identifies the job(s) associated with the situation and offers drill down options to view those jobs.

Users can control the Situations executed when running this analysis by using the Situations... button on the Analyze Collection window.

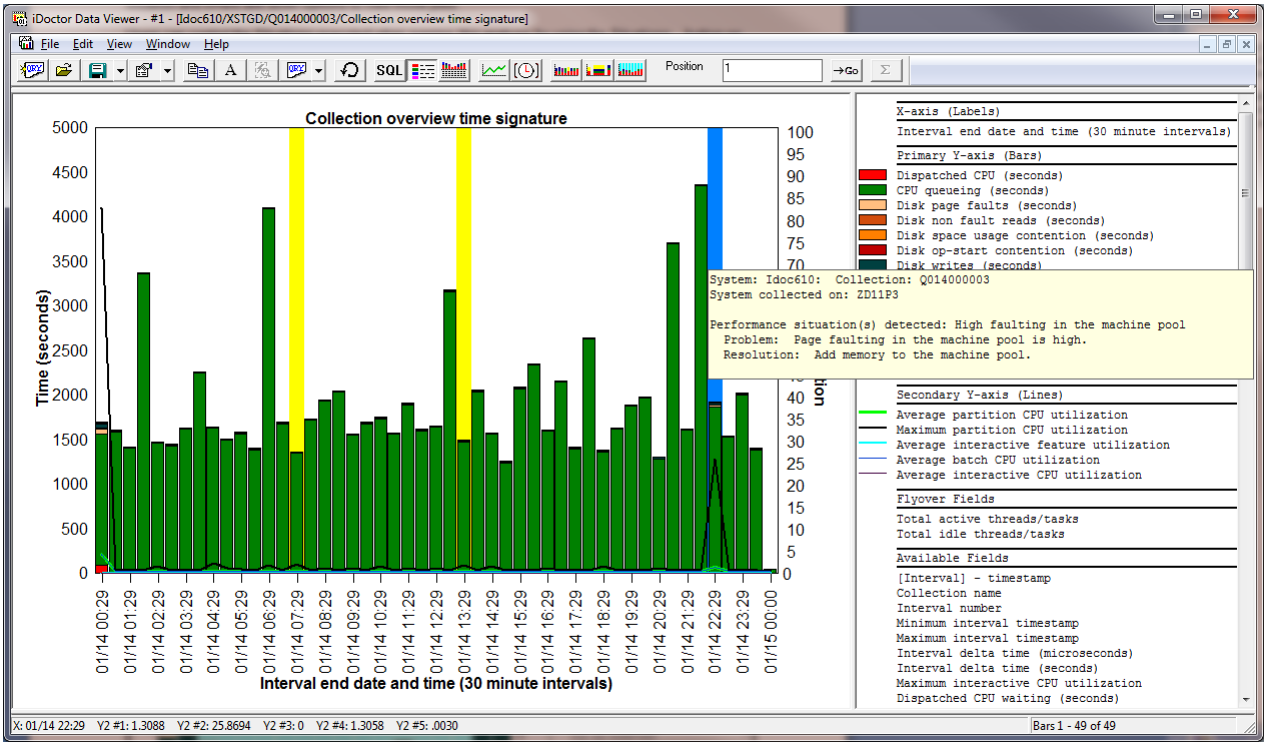
9.10.4.1 Situations Window

See the Collection Services Investigator Situations Window section for more information.

9.10.4.2 Features Offered

After running the analysis the following features become available in CSI:

1) Over time graphs in CSI will display any situations that occurred as background colors.



2) Interval summary -> Situations tab displays the situations that were found in the interval.

9.10.4.3 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

SQL Table folder	Table description	SQL table
Situational Analysis	Situational Analysis file	QAIDRCSANL_<<MBRNAME>>

9.10.5 External Storage Cache Statistics (6.1.1+)

This analysis is used to process DS6K/DS8K statistics related to cache. It can be useful to determine latency delays between the IBM i box and the external storage subsystem.

This data is only available at 6.1.1. or higher if PTF SI36359 is installed . This analyses parses data out from file QAPMXSTGV. If the file does not exist then this analysis won't be available.

For additional analysis tips see section 8.2.5 in the following RedBook.
http://www.redbooks.ibm.com/redbooks/pdfs/sg247858.pdf

9.10.5.1 Features Offered

After running the analysis the following features become available in CSI:

1) Under the collection, a new folder External Storage cache statistics appears.

These graphs provide over time, by disk unit and ASP over the external storage data. Table views are also provided to access the SQL tables.

The by time interval, by disk unit options will prompt you for desired ASP or give comparison option to compare 1 ASP with another.

2) SQL Tables -> External Storage Cache Statistics -> Select the desired table , right-click External Storage Cache Statistics menu appears offering the same graphs as shown under #1 above.

9.10.5.2 SQL Tables Generated

The list of SQL tables generated by the analysis is shown below:

SQL Table folder	Table description	SQL table
External storage cache statistics	External storage X32 by unit	QAIDRCSEXT_UNITS_<<MBRNAME>>
External storage cache deltas	External storage X32 deltas	QAIDRCSEXT_<<MBRNAME>>
External storage cache cumulative totals	External storage X32 totals (these are raw totals returned from the DS6K/DS8K box and thus keep increasing over time)	QAIDRCSEXT_TOT_<<MBRNAME>>

9.10.6 External Storage Links and Ranks Statistics (7.1+)

This analysis is used to process DS6K/DS8K statistics for links and ranks. This analyses parses data out from file QAPMXSTGD. If the file does not exist then this analysis won't be available.

For additional analysis tips see section 8.2.5 in the following RedBook.
http://www.redbooks.ibm.com/redbooks/pdfs/sg247858.pdf

9.10.6.1 Features Offered

After running the analysis the following features become available in CSI:

1) Under the collection, a new folder External Storage link and rank statistics appears.

These graphs provide over time graphs for links or ranks, and link rankings and rank rankings. Table views are also provided to access the SQL tables.

2) SQL Tables -> External Storage Links and Ranks -> Select the desired table , right-click External Storage link and rank Statistics menu appears offering the same graphs as shown under #1 above.

9.10.6.2 SQL Tables Generated

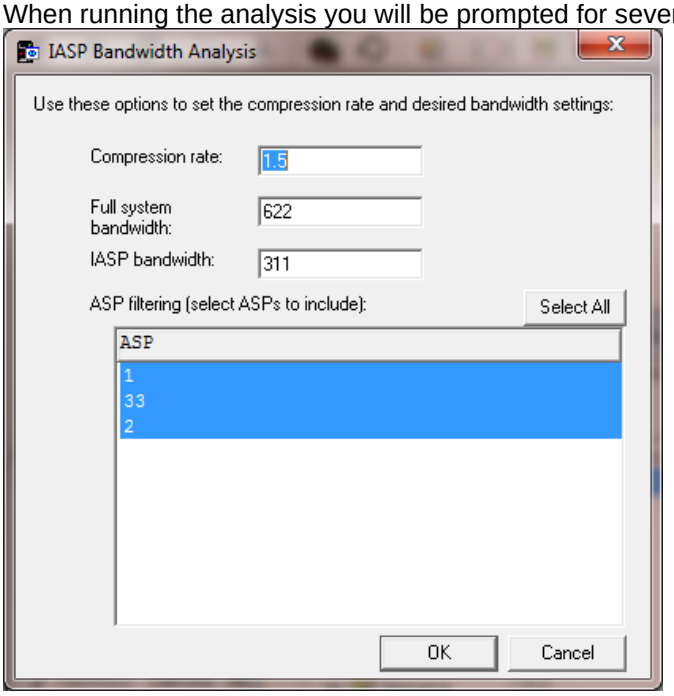
The list of SQL tables generated by the analysis is shown below:

SQL Table folder	Table description	SQL table
External storage links	Link deltas	QAIDRCSEXT_LINKD_<<MBRNAME>>
External storage ranks	Rank deltas	QAIDRCSEXT_RANKD_<<MBRNAME>>

9.10.7 IASP Bandwidth

The IASP Bandwidth analysis was created from an Excel spreadsheet created by David Frost (IBM Lab Services). Its purpose is to analyze the Collection Services data with the intent to determine if the system is a good candidate for migrating to Independent ASPs.

When running the analysis you will be prompted for several parameters:

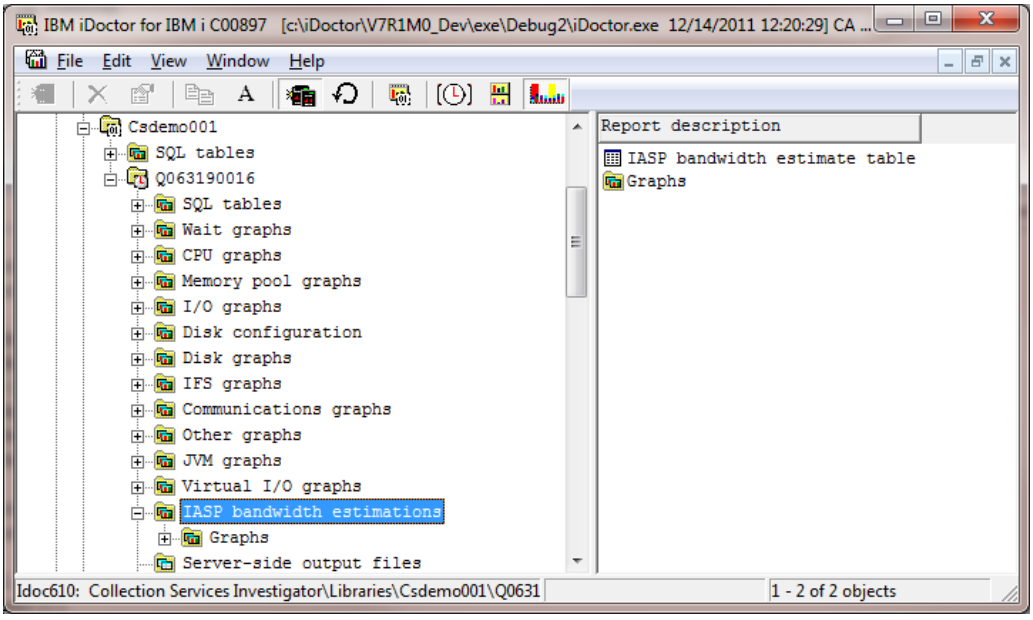


IASP Bandwidth Analysis Options

Parameter	Description
Compression rate	The estimated network (comm line) compression rate (between the system and the IASP). A value of 1 means no compression. 1.5 (the default) means 50% compression. Values less than 1 should not be used.
Full system bandwidth	Estimated bandwidth required by system without IASPs (in megabits per second). **Note:** depending on the system/data you may want to adjust this value much higher.
IASP bandwidth	Estimated bandwidth required by system with IASPs implementation (in megabits per second) **Note:** depending on the system/data you may want to adjust this value much higher.
ASP filtering	This option allows you to select which ASPs to include when running the analysis.

9.10.7.1 Features Offered

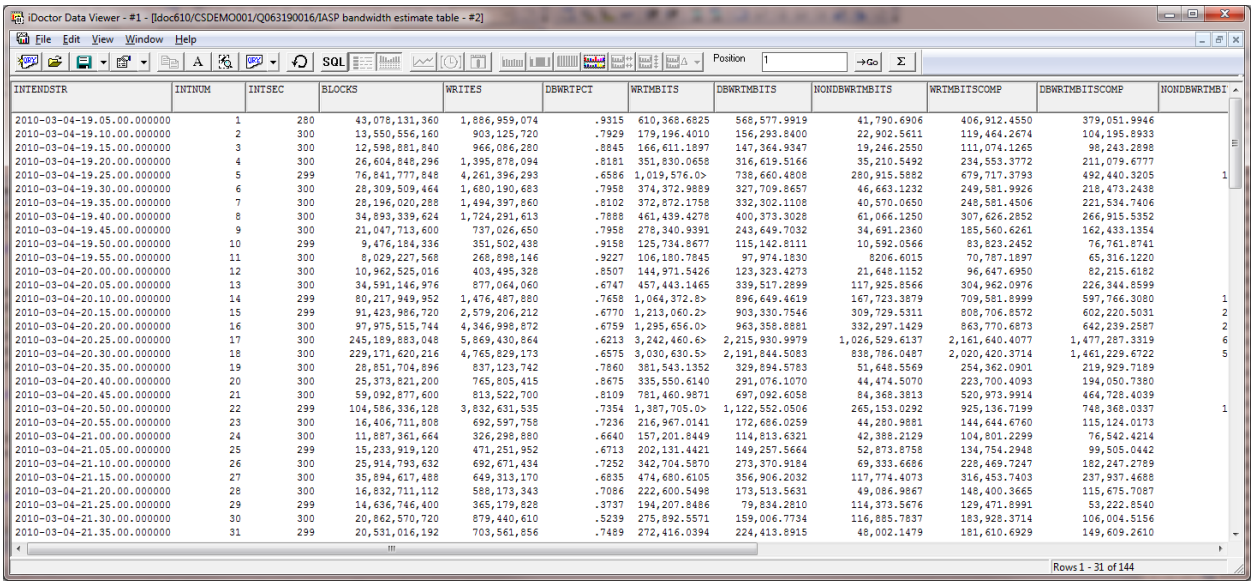
After running the analysis a new folder IASP bandwidth estimations will be available containing the table generated and a subfolder with graphs.



IASP bandwidth estimations folder

9.10.7.1.1 IASP bandwidth estimate table

This table represents the statistics generated by the analysis. The statistics generated includes the number of blocks, writes, database write percentage and various bandwidth estimates (all in megabits per second)

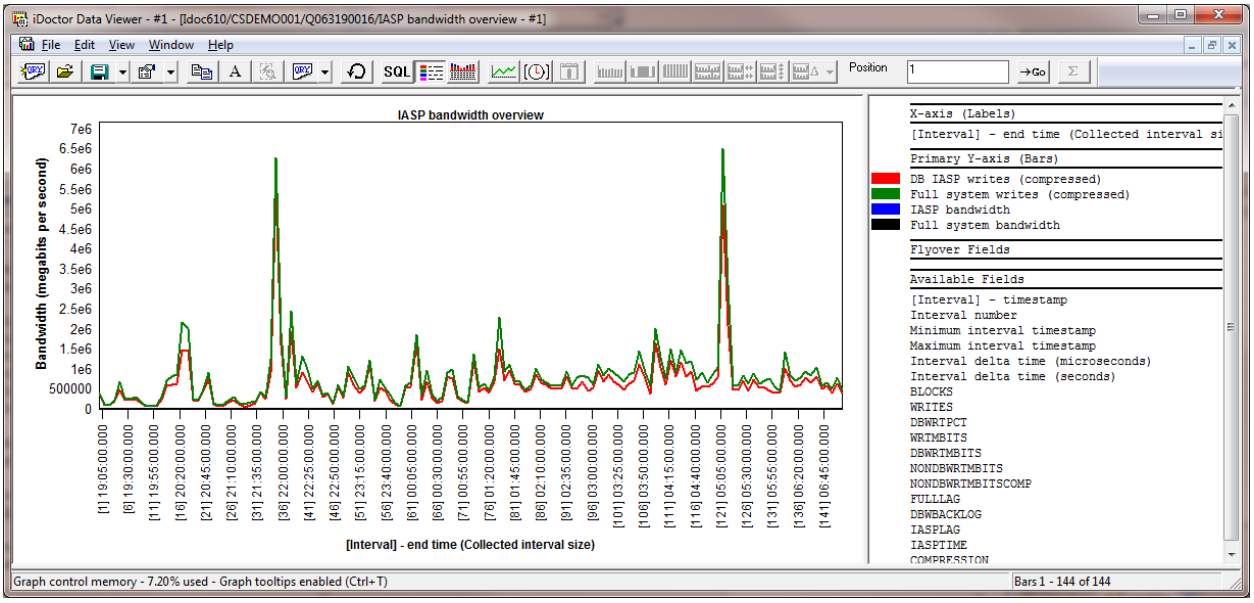


INTENDSTR	INTNUM	INTSEC	BLOCKS	WRITES	DBWRITPCT	WRITBITS	DBWRITBITS	NONDBWRITBITS	WRITBITSCOMP	DBWRITBITSCOMP	NONDBWRITBI
2010-03-04-19.05.00.000000	1	280	43,078,131,360	1,886,959,074	.9315	610,368.6825	568,577.9919	41,790.6906	406,912.4550	379,051.9946	
2010-03-04-19.10.00.000000	2	300	13,550,556,160	903,125,720	.7929	179,196.4010	156,293.8400	22,902.5611	119,464.2674	104,195.8933	
2010-03-04-19.15.00.000000	3	300	12,598,881,840	966,086,280	.8845	166,611.1897	147,364.9347	19,246.2550	111,074.1265	98,243.2898	
2010-03-04-19.20.00.000000	4	300	26,604,848,296	1,395,878,094	.8181	351,830.0658	316,619.5166	35,210.5492	234,553.3772	211,079.6777	
2010-03-04-19.25.00.000000	5	299	76,841,777,848	4,261,396,293	.6586	1,019,576.0>	738,660.4808	280,915.5882	679,717.3793	492,440.3205	1
2010-03-04-19.30.00.000000	6	300	28,309,509,464	1,680,190,683	.7958	374,372.9889	327,709.8657	46,663.1232	249,581.9926	218,473.2438	
2010-03-04-19.35.00.000000	7	300	28,196,020,288	1,494,397,860	.8102	372,872.1758	332,302.1108	40,570.0650	248,581.4506	221,534.7406	
2010-03-04-19.40.00.000000	8	300	34,893,339,624	1,724,291,613	.7888	461,439.4278	400,373.3028	61,066.1250	307,626.2852	266,915.5352	
2010-03-04-19.45.00.000000	9	300	21,047,713,600	737,026,650	.7958	278,340.9391	243,649.7032	34,691.2360	185,560.6261	162,433.1354	
2010-03-04-19.50.00.000000	10	299	9,476,184,336	351,502,438	.9158	125,734.8677	115,142.8111	10,592.0566	83,823.2452	76,761.8741	
2010-03-04-19.55.00.000000	11	300	8,029,227,568	268,898,146	.9227	106,180.7845	97,974.1830	8206.6015	70,787.1897	65,316.1220	
2010-03-04-20.00.00.000000	12	300	10,962,525,016	403,495,328	.8507	144,971.5426	123,323.4273	21,648.1152	96,647.6950	82,215.6182	
2010-03-04-20.05.00.000000	13	300	34,591,146,976	877,064,060	.6747	457,443.1465	339,517.2899	117,925.8566	304,962.0976	226,344.8599	
2010-03-04-20.10.00.000000	14	299	80,217,949,952	1,476,487,880	.7658	1,064,372.8>	896,649.4619	167,723.3879	709,581.8999	597,766.3080	1
2010-03-04-20.15.00.000000	15	299	91,423,986,720	2,579,206,212	.6770	1,213,060.2>	903,330.7546	309,729.5311	808,706.8572	602,220.5031	2
2010-03-04-20.20.00.000000	16	300	97,975,515,744	4,346,998,872	.6759	1,295,656.0>	963,358.8881	332,297.1429	863,770.6873	642,239.2587	2
2010-03-04-20.25.00.000000	17	300	245,189,883,048	5,869,430,864	.6213	3,242,460.6>	2,215,930.9979	1,026,529.6137	2,161,640.4077	1,477,287.3319	6
2010-03-04-20.30.00.000000	18	300	229,171,620,216	4,765,829,173	.6575	3,030,630.5>	2,191,844.5083	838,786.0487	2,020,420.3714	1,461,229.6722	5
2010-03-04-20.35.00.000000	19	300	28,851,704,896	837,123,742	.7860	381,543.1352	329,894.5783	51,648.5569	254,362.0901	219,929.7189	
2010-03-04-20.40.00.000000	20	300	25,373,821,200	765,805,415	.8675	335,550.6140	291,076.1070	44,474.5070	223,700.4093	194,050.7380	
2010-03-04-20.45.00.000000	21	300	59,092,877,600	813,522,700	.8109	781,460.9871	697,092.6058	84,368.3813	520,973.9914	464,728.4039	
2010-03-04-20.50.00.000000	22	299	104,586,336,128	3,832,631,535	.7354	1,387,705.0>	1,122,552.0506	265,153.0292	925,136.7199	748,368.0337	1
2010-03-04-20.55.00.000000	23	300	16,406,711,808	692,597,758	.7236	216,967.0141	172,686.0259	44,280.9881	144,644.6760	115,124.0173	
2010-03-04-21.00.00.000000	24	300	11,887,361,664	326,298,880	.6640	157,201.8449	114,813.6321	42,388.2129	104,801.2299	76,542.4214	
2010-03-04-21.05.00.000000	25	299	15,233,919,120	471,251,952	.6713	202,131.4421	149,257.5664	52,873.8758	134,754.2948	99,505.0442	
2010-03-04-21.10.00.000000	26	300	25,914,793,632	692,671,434	.7252	342,704.5870	273,370.9184	69,333.6686	228,469.7247	182,247.2789	
2010-03-04-21.15.00.000000	27	300	35,894,617,488	649,313,170	.6835	474,680.6105	356,906.2032	117,774.4073	316,453.7403	237,937.4688	
2010-03-04-21.20.00.000000	28	300	16,832,711,112	588,173,343	.7086	222,600.5498	173,513.5631	49,086.9867	148,400.3665	115,675.7087	
2010-03-04-21.25.00.000000	29	299	14,636,746,400	365,179,828	.3737	194,207.8486	79,834.2810	114,373.5676	129,471.8991	53,222.8540	
2010-03-04-21.30.00.000000	30	300	20,862,570,720	879,440,610	.5239	275,892.5571	159,006.7734	116,885.7837	183,928.3714	106,004.5156	
2010-03-04-21.35.00.000000	31	299	20,531,016,192	703,561,856	.7489	272,416.0394	224,413.8915	48,002.1479	181,610.6929	149,609.2610	

IASP bandwidth estimate table

9.10.7.1.2 IASP bandwidth overview graph

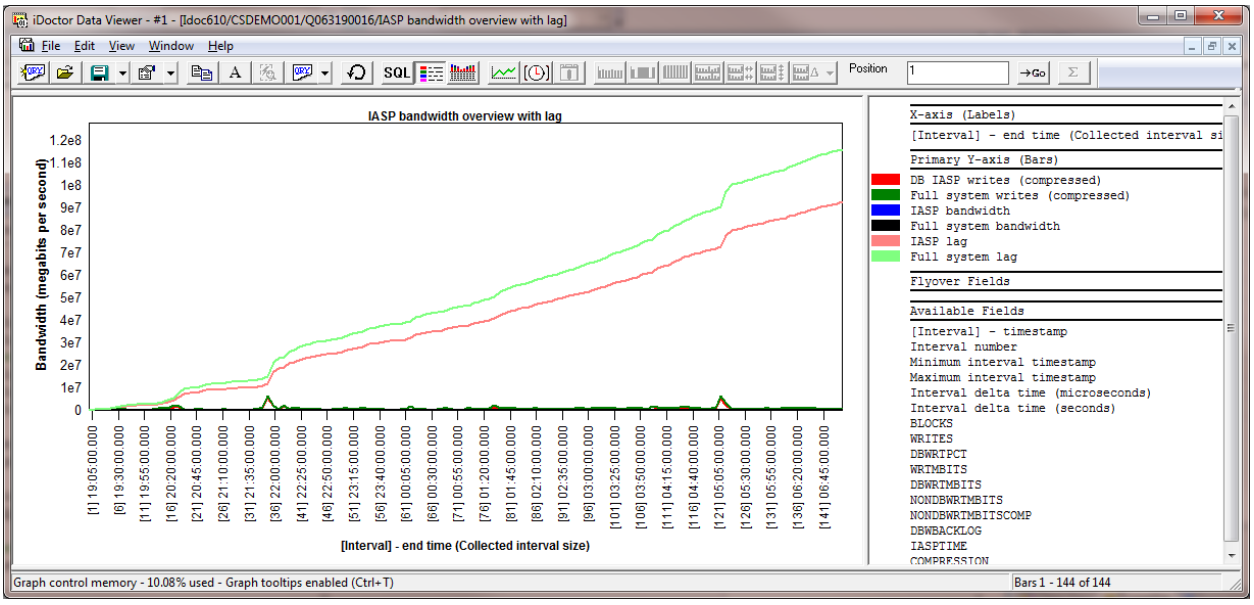
This graph displays the database writes for IASPs as well as the full system writes. The estimated bandwidth numbers provided when running the analysis are also given (not visible here because they were way too small.)



IASP bandwidth estimate graph

9.10.7.1.3 IASP bandwidth overview graph with lag

This graph is like the previous graph except it includes how much bandwidth lag there would be based on the parameter estimation values given when the analysis was created.



IASP bandwidth estimate graph with lag

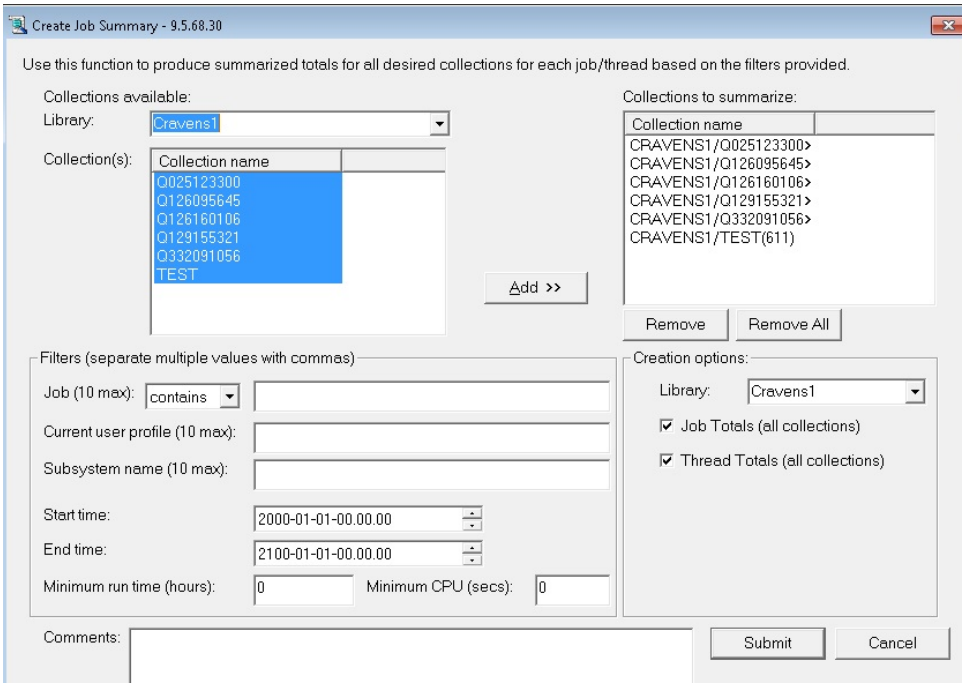
9.10.8 Create Job Summary

The Create Job Summary analysis allows a user to build tables that add up job statistics across 1 or more collections. These summaries are stored in the SQL Tables folder in iDoctor.

Note: If the collections specified have not already been summarized (i.e. the Collection Summary analysis has not yet been ran), it will be ran automatically by running this analysis.

Advanced Usage Note: At 5.4, the wait bucket times do NOT include wait bucket idle time in intervals where the job used zero CPU. In some types of threads/tasks this may be significant. At 6.1 (with latest builds), these times are included and are derived from file QAPMJOBWTG.

An example of this interface is:



Create Job Summary Analysis Window

The following table describes the various components of this window.

GUI Element	**Description**
Library	The name of the library to select available collections from.
Collection(s) list	Displays the list of collections available in the current library to select from.
Add >>	Adds the selected collections in the Collection(s) available list to the Collections to summarize list.

Job (10 max)	This field indicates which jobs to include in the reports by specifying a portion of the job name. If you leave this field blank all jobs will be included in the report. Up to 10 job name values may be entered. Separate multiple values with a comma. The drop down list allows you to specify if each value used should be a “contains” comparison or a “starts with” comparison.
Current user profile (10 max)	This is the list of up to 10 current user profiles to filter the analysis output on.
Subsystem name (10 max)	This is the list of up to 10 subsystem names to filter the analysis output on.
Start and end time	Use these fields to filter the data by time.
Minimum run time (hours)	If you wish to filter the job data by a minimum time the job ran, then enter a value in hours.
Minimum CPU (secs)	If you wish to filter the job data by a minimum CPU time used, then enter a value in seconds.

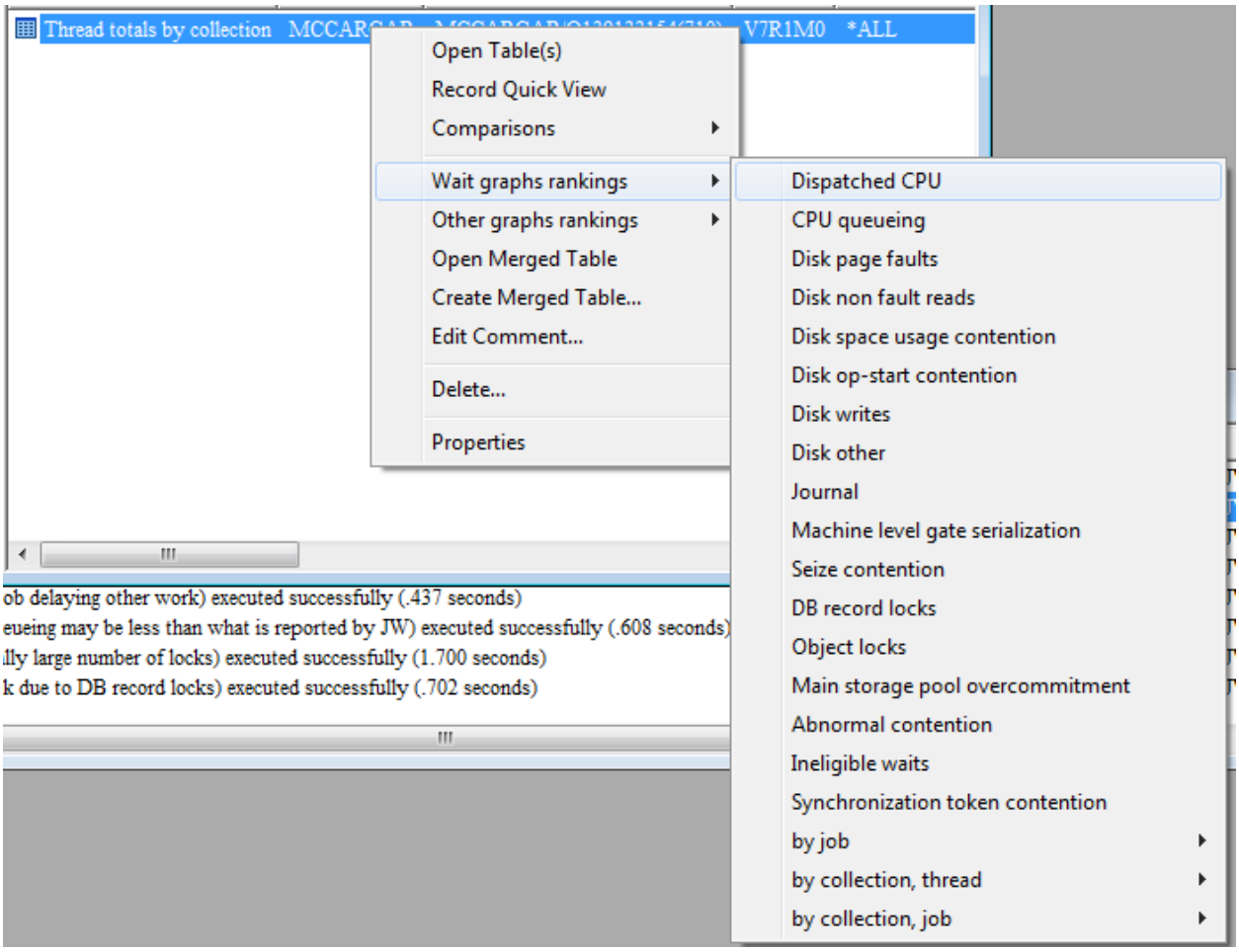
Comments	The comments field is used to apply a comment to all SQL tables generated by this analysis.
Collections to summarize	The list of collections to include in the analysis.
Remove / Remove All buttons	Removes collections from the Collections to Summarize list.
Creation options: library	This field allows the user to specify a different library than the current one for the SQL tables generated.
Job totals option	If checked a report will be generated that summarizes the data across all collections by job .
Threads totals option	If checked a report will be generated that summarizes the data across all collections by thread/taskcount.

Pressing the Submit button will run the analysis over the desired collections and place the results in the SQL tables repository. One folder is created for each of the SQL table creation options checked.

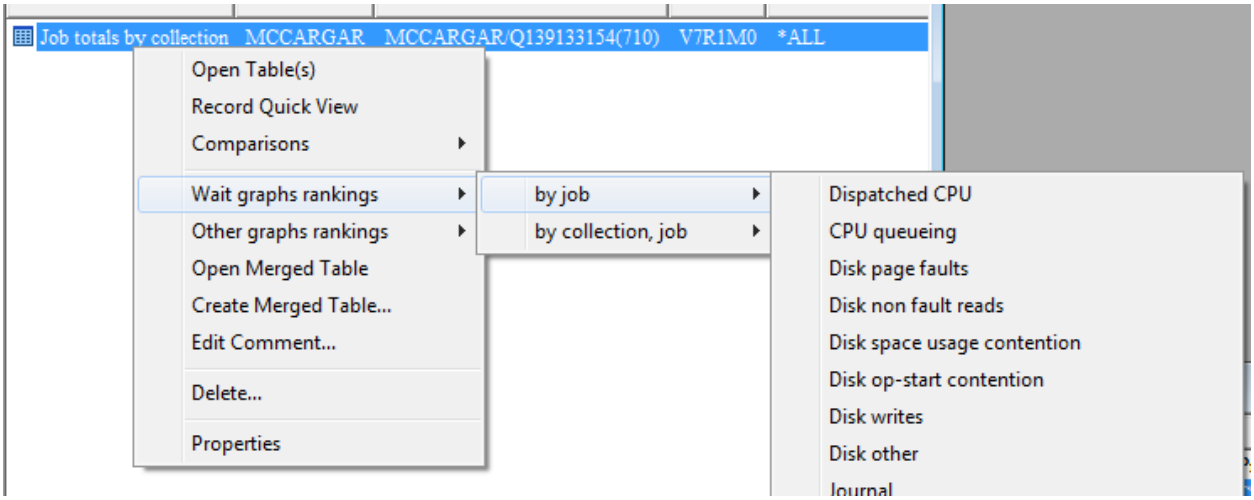
When running the analysis the progress is shown in the Remote SQL Statement Status View.

9.10.8.1.1 Features Offered

After running the analysis the following features become available in under the SQL Tables folder by right-clicking the SQL tables.



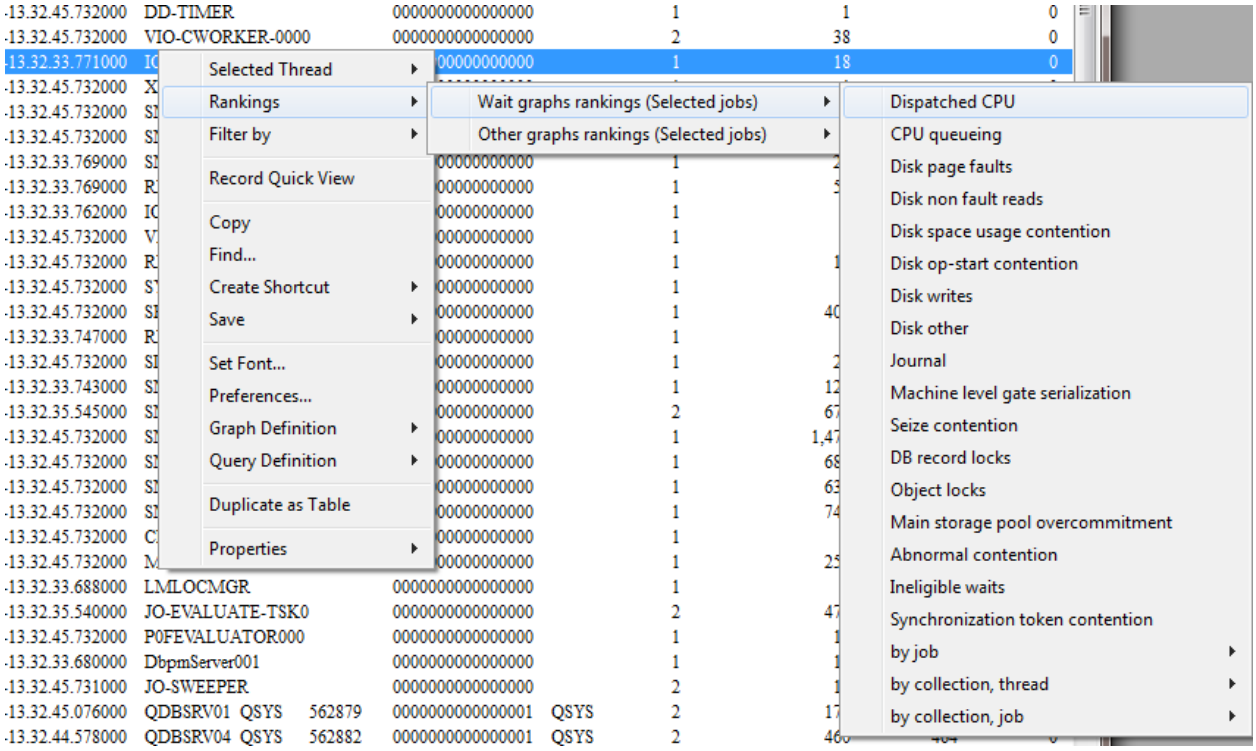
Thread totals SQL tables graphing options



Job totals SQL tables graphing options

Opening a Create Job Summary SQL table and right-clicking individual records provides these additional options:

1. Selected thread (or job) over time
2. Rankings (for either selected jobs, all jobs, or jobs matching a generic job name filter)
3. Filter by menu (to configure how the rankings graphs should be filtered)



9.10.8.2 Statistics provided

The data included in this analysis is generated from job statistics from files QAPMJOBMI, QAPMJOBOS, QAPMJOBWT and QAPMJOBWTG (at 6.1+ only). The columns included in this analysis are:

Collection name (except in the job totals for all collections reports)
OSVRM
Duration of job in hours - (job may have ran longer outside the collected data)
Elapsed seconds
Start of job included 1-Yes, 0-No - (whether the job started during the collection)
End of job included 1-Yes, 0-No - (whether the job ended during the collection)
Start timestamp
End timestamp
Fully qualified job name or task name
Thread ID
Current user profile (max value, there could be multiples)
Job pool
Job subsystem
Thread CPU microseconds
Job CPU microseconds
Synchronous database reads
Synchronous non database reads
Synchronous database writes
Synchronous non database writes
Asynchronous database reads
Asynchronous non database reads
Asynchronous database writes
Asynchronous non database writes
Active to wait transitions
Wait to ineligible transitions
Active to ineligible transitions
I/O pending page faults
Waits for asynchronous writes
Page faults
Allocated DASD pages
Deallocated DASD pages
Binary overflows
Decimal overflows
Floating point overflows
Stream file reads
Stream file writes
New mainstore frames stolen
Successful removes
Max prepared statement areas used
Intervals SQL statements running
Total threads active since job start
Total threads created since job start
Communication file writes
Communication file reads
Logical database writes
Logical database reads
Logical database others
Application input queueing
Application input queueing transactions
Resource usage time in microsecs
Resource usage transactions
Display I/O response time in microsecs
Display I/O transactions
IFS symbolic link reads
IFS directory reads
IFS lookup cache hits
IFS lookup cache misses
IFS opens

IFS directory creates
IFS non directory creates
IFS directory deletes
IFS non directory deletes
Socket reads
Socket writes
Socket bytes read
Socket bytes written
Fully opened SQL cursors
Pseudo closed SQL cursors
Maximum activation groups
Task count - (unique identifier for the job/thread/task. In the CS database files this is called TDE)
Initial thread task count - (unique identifier for the primary job/thread.)
Wait bucket times
Wait bucket counts - (these are how many unique occurrences there were within each bucket)

9.10.8.2.1 SQL Tables Generated

This function generates 1 table for each type of report indicated to be created (where <<X>> is a unique number):

SQL Table folder	Table description	SQL table
Job totals	Job totals	QAIDRCS4SUM_<<X>>
Thread totals	Thread totals	QAIDRCS3SUM_<<X>>
Job totals by collection	Job totals by collection	QAIDRCS2SUM_<<X>>
Thread totals by collection	Thread totals by collection	QAIDRCS1SUM_<<X>>

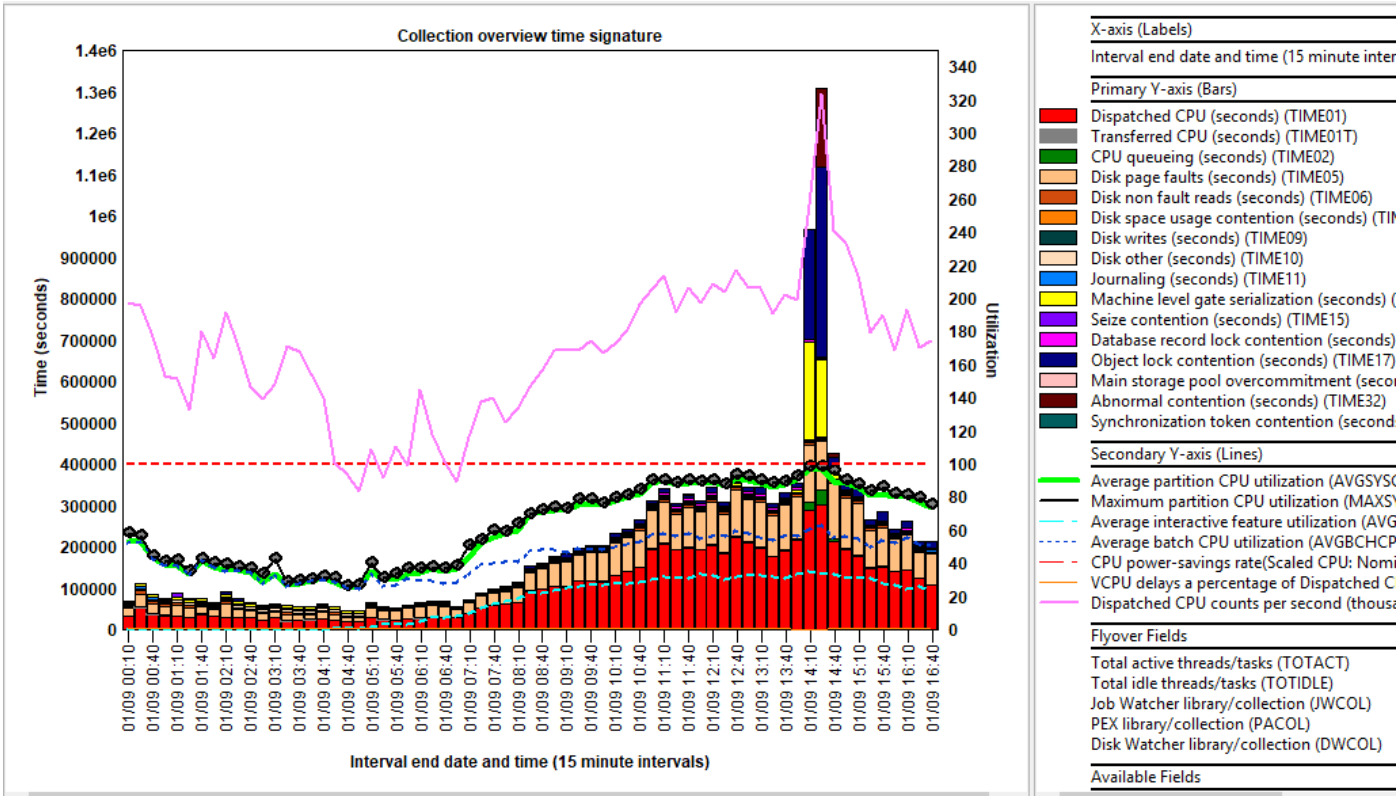
9.11 Graph notes

This section discusses the types of graphs found in a Collection Services Investigator collection and how to use them.

It also covers briefly a discussion on “interesting” vs “idle” waits and on CPU related fields shown in many Collection Services Investigator graphs.

9.11.1 CPU notes

Many of the Overview graphs show different types of CPU utilization and other CPU related fields discussed below:



Collection overview time signature

Average partition CPU utilization (green line) – This is the average CPU utilization for each summarized interval as collected by the LPAR.

Maximum partition CPU utilization (black line) – Because each bar in the graph could contain data from several intervals, this is the highest partition average CPU utilization that occurred. It is not a true maximum that occurred but the maximum of the average that occurred in those intervals being summarized.

Average interactive feature utilization (light blue dashed line) – This is CPU utilization of all interactive jobs found during the collection.

Average batch CPU utilization (blue dashed line) – This is CPU utilization of all batch jobs found during the collection.

CPU power-saving rates (Scaled CPU : Nominal CPU) (dashed red line) – This value shows the effect of CPU power saving, also known as CPU scaling that occurred over time. It is multiplied by 100 so it will scale with the rest of the fields on the secondary Y-axis and will typically be at the top of the graph.

VCPU delays as a percentage of Dispatched CPU – This value shows the amount of time virtual processors spent waiting to run in relation to the dispatched CPU time used.

The formula for this is (((SYPTREADY + SYPTLATEN) / DOUBLE(1000000)) / TIME01) * 100.00 and is multiplied by 100 so it will graph well with the other metrics on the secondary Y-axis.

SYPTREADY	Virtual processor thread wait ready time. The elapsed time in microseconds that ready to run threads of the partition's virtual processors waited to be dispatched while entitled capacity was exhausted.
SYPTLATEN	Virtual processor thread dispatch latency. The elapsed time in microseconds that ready to run threads of the partition's virtual processors waited to be dispatched while entitled capacity was not exhausted and a physical processor was not available.

TIME01	This is the Dispatched CPU time spent (in seconds)

Dispatched CPU counts per second – This value indicates the number of times the CPU entered a run state as a rate per second (in thousands)

9.11.2 "Interesting" wait buckets

Job Watcher (and Collection Services Investigator) both make use the IBM i wait buckets instrumented on the system. There are 32 wait buckets, and these cover all possible wait states that a job can be in.

Many of these buckets are associated with idle waits and generally are not of any interest from a performance standpoint. For that reason, many iDoctor graphs only show the wait buckets that are interesting for a performance perspective.

The interesting wait buckets are:

Bucket	Description
1	Dispatched CPU
2	CPU queueing
5	Disk page faults
6	Disk non-fault reads
7	Disk space usage contention
8	Disk op-start contention
9	Disk writes
10	Disk other
11	Journaling
14	Machine level gate serialization
15	Seize contention
16	Database record lock contention
17	Object lock contention
18	Ineligible waits
19	Main storage pool overcommitment
20	Journal save while active (7.2+ only)
31	Synchronization token contention
32	Abnormal contention

9.11.3 CPU power-savings rate (scaled CPU : nominal CPU)

This field shows how much CPU was increased or reduced based on power (energy) savings features. iDoctor shows this value as a percentage so that it can exist on the same graph as CPU utilization (100% value in iDoctor equals 1.0 value shown in PDI.)

For more information:

http://ibmsystemsmag.blogs.com/i_can/2010/03/i-can-understand-scaled-cpu-time.html

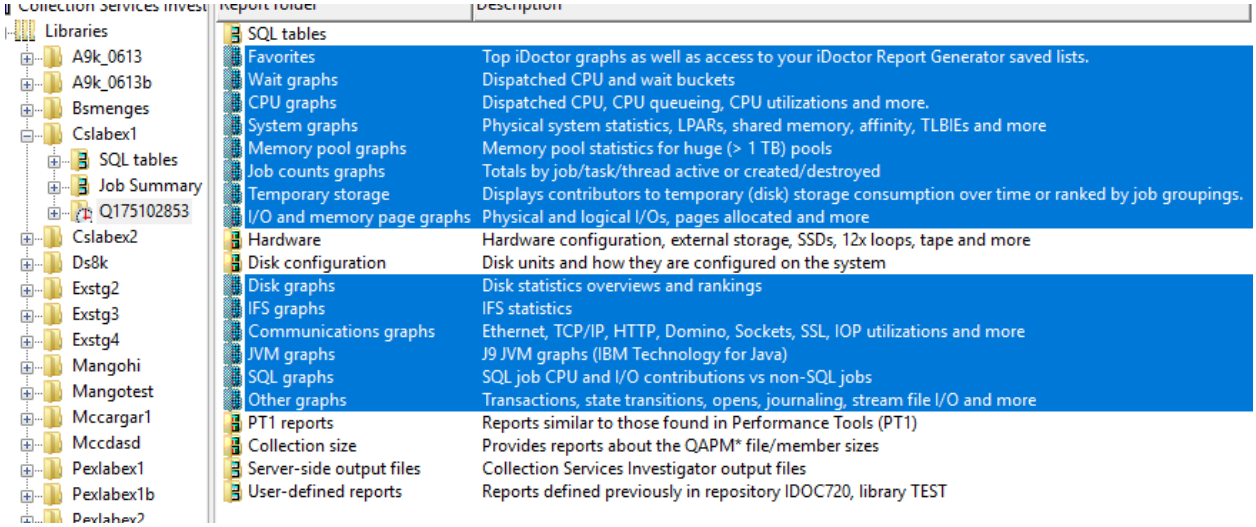
9.11.4 Workload capping delays as a percentage of CPUQ

This field shows what percentage of CPU queuing (green bar color) on the graph was from workload capping delays. It is only available if workload capping data is present. A special graph called Collection overview with workload capping time signature will be shown if the data is available.

9.11.5 Overview graphs

The overview graphs are time-based and summarize the data in various ways over time. These show the data over the entire collection.

These graphs are contained within several folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the desired menus.



Overview Graphs in Collection Services Investigator (selected)

Each folder contains a series of graphs. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Often most of these graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one. You can also use the Graph Compare icon on the toolbar of the Main Window in order to perform comparisons between graphs.

Tip #1: Use the clock icon on the toolbar to change the default Time interval size. This is useful if you have many thousands of intervals and wish to group those intervals into fewer bars than would be shown if you graphed at the Collected interval size.

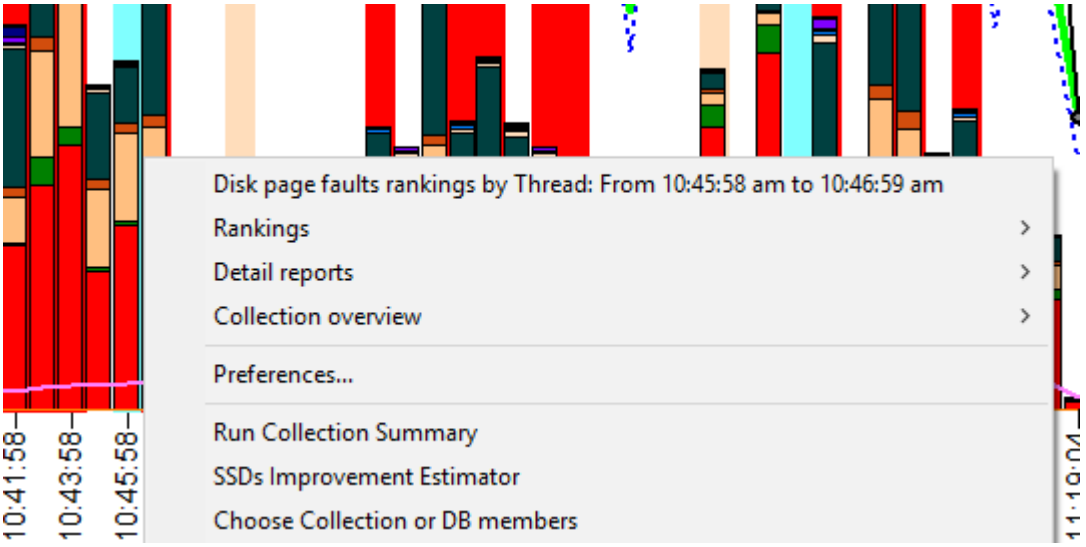
Tip #2: You can make time selections by clicking the 1st bar, hold down the shift key and clicking the last bar of the desired time range and right-click a bar and wait bucket color within that time period in order to drill down into the jobs experiencing the most amount of time in that wait bucket you right-clicked on.

Tip #3: If you wish to group job data in the ranking graphs with groupings larger than thread (by generic job, etc) you must run the Collection Summary analysis first.

9.11.5.1 Drilling down into Rankings graphs

When drilling down into ranking graphs (from Collection-wide graphs) you can select the desired time period of interest by holding down the shift key and clicking the 1st and last bars of the desired time period. Then right-click on one of the bars in the time period and pick the desired drill down graph.

This action will look something like this:



Drilling down from a Collection Overview Time Signature graph into ranking graphs

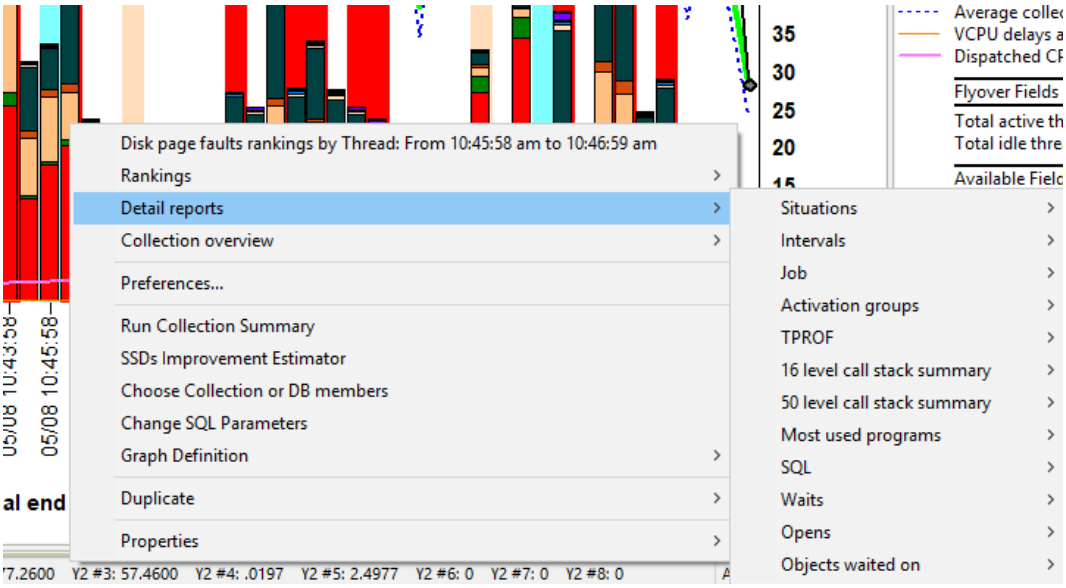
The default action at the top is Thread signatures ranked by disk page faults. This is because the bucket right-clicked on is the disk page faults time bucket. If it had been CPU time then it would give a default drilldown of Thread signatures ranked by Dispatched CPU instead.

However in the event that you don't want to drill down into wait buckets, you can use one of the Rankings menus and pick from there the desired graph you want.

9.11.5.2 Drilling down into Detail reports

Another drilldown option from the Collection-wide graphs is found under a menu called "Detail reports". This menu offers a series of table views that provide quick access to much of the raw data found in the collection.

An example of this menu and list of report categories it contains is:



Detail reports menu options

These reports are based on either the single interval or time range selected.

9.11.5.3 Run Job Summary option

The Create Job Summary menu option allows the user to create a Job Summary for only the time period selected.

PLEASE NOTE: This option is only shown if a time range (> 1 interval) has been selected.

9.11.5.4 Split Collection option

The Split Collection option allows the user to create a new collection from the selected time period in the current one.

PLEASE NOTE: This option is only shown if a time range (> 1 interval) has been selected.

9.11.5.5 Run Collection Summary

This option allows the user to run a filtered Collection Summary analysis for the currently selected interval or time period.

See the section on the Run Collection Summary interface for more information.

9.11.5.6 SSDs Improvement Estimator

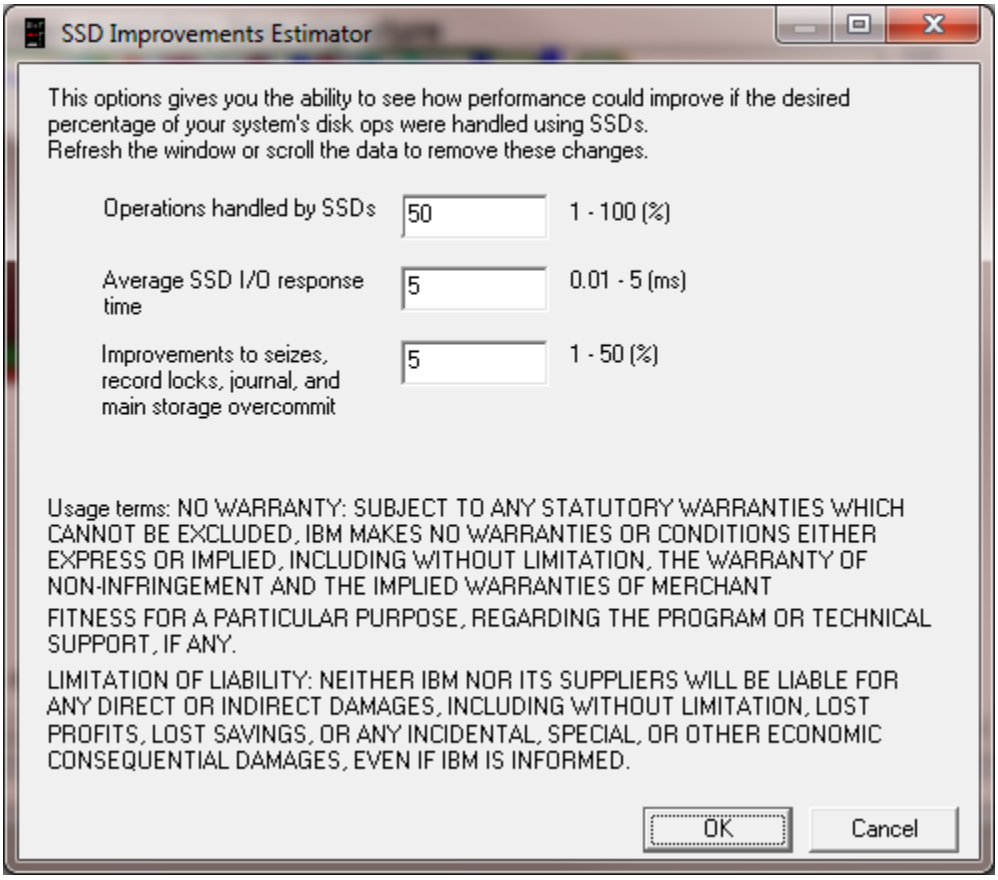
The SSDs improvement estimator is intended to show the user how a wait bucket (over time) graph might look if SSDs were added to the current system. It alters wait bucket graphs in CSI/JW to show possible disk time reductions if SSDs were installed based on % of SSDs, avg SSD I/O response time and estimated improvements to other types of waits besides disk.

Here are a few tips when using this option:

1) This function primary modifies disk wait times (buckets Disk page faults and disk non fault reads) If these times are not present on the graph then don't bother.

2) Use the clock icon to summarize the data so that all the data you wish to estimate improvements for is on the current page of the graph. As soon as you scroll the graph the changes are lost.

An example of the interface follows:



The estimations given are truly that and may vary significantly from the data captured after SSDs are installed.

9.11.6 Ranking graphs

In Collection Services Investigator the ranking graphs are ranking jobs by a desired metric using a job grouping such as by thread, by job, by user name, etc. Collection Services Investigator currently has 12 job groupings used. All 12 are available only if the Collection summary analysis has been ran on a collection. Otherwise the only ranking graph type available is **by thread**.

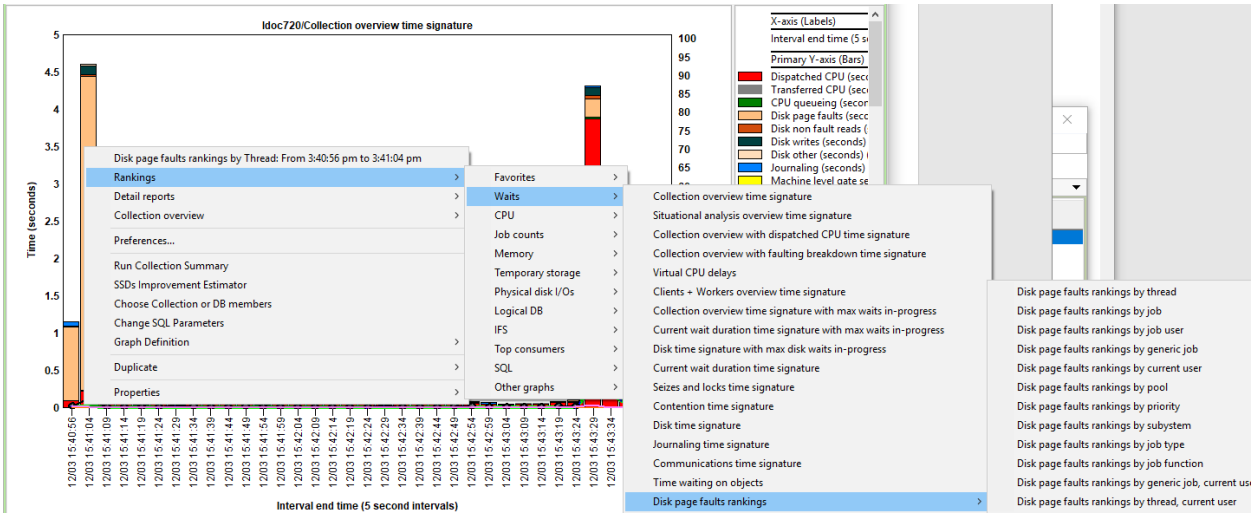
The types of job groupings available in Collection Services Investigator are:

by thread
by job
by job user
by generic job
by current user
by pool
by priority
by subystem
by job type
by job function
by generic job, current user
by thread, current user

Job groupings list

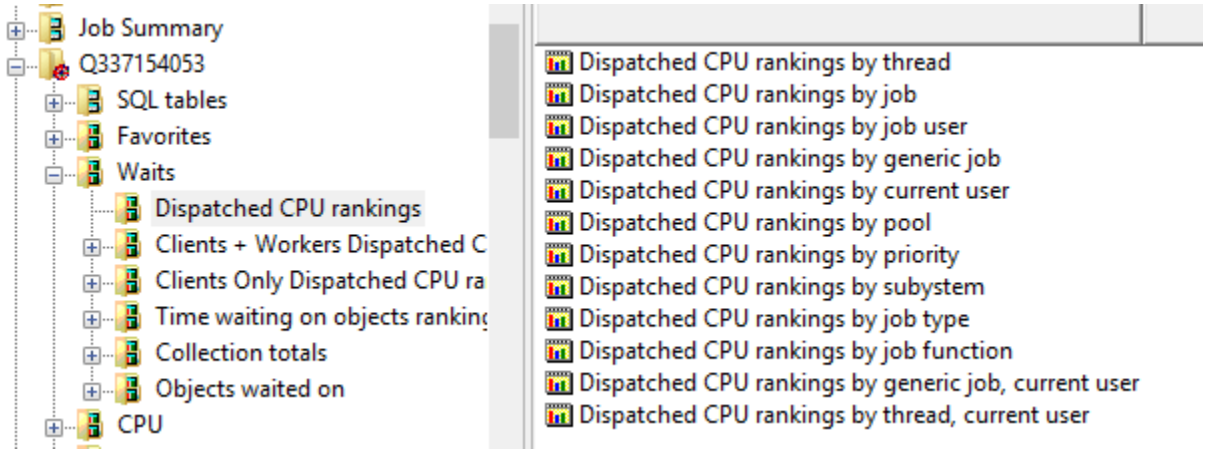
Ranking graphs are accessed in 1 of 2 ways:

3) As a drill-down from an overview graph under the Rankings -> (Pick a metric) rankings -> By job grouping X menu:



Rankings drill down options from a Collection overview time signature

4) Directly under the collection within one of the ranking graphs folders such as Waits -> Dispatched CPU rankings.



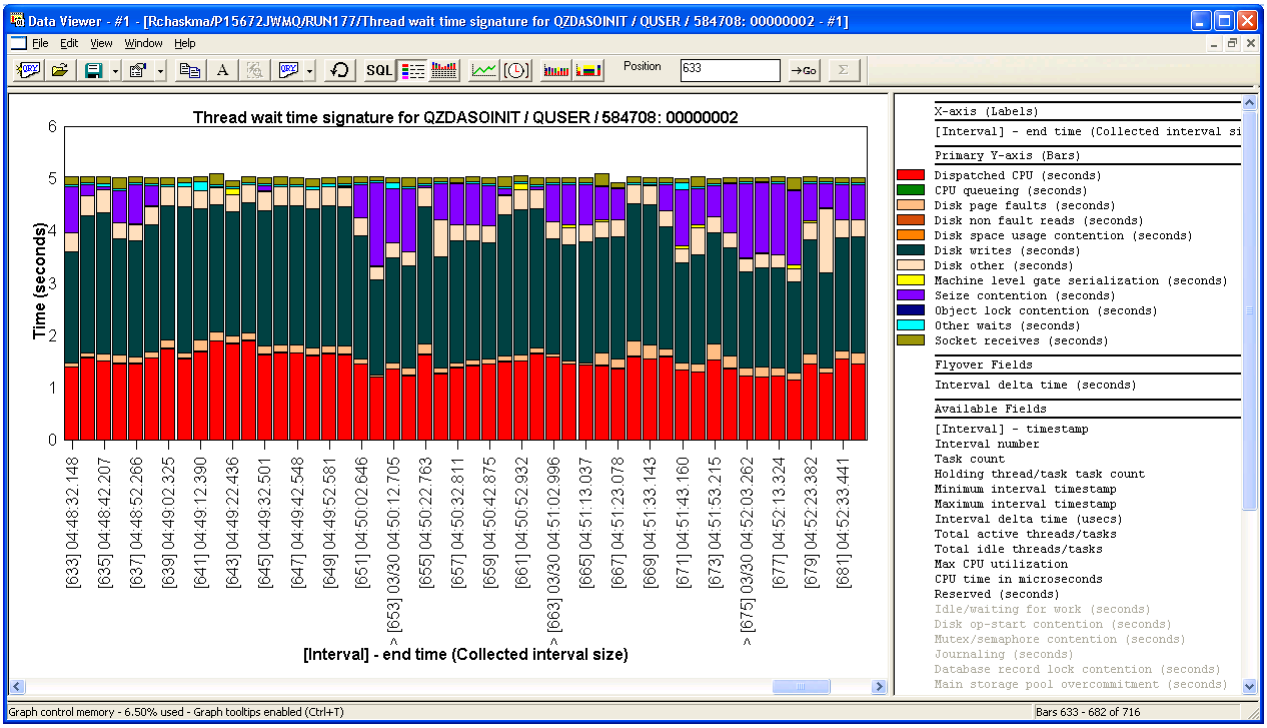
Waits -> Dispatched CPU rankings folder

From a ranking graph, users can select one or more job groupings and graph that selection over time.

9.11.7 Selection over time graphs

This graph type shows the currently selected thread (or job, generic job, user, etc) over time. The size of time interval is configurable (greater than the collected size only) by using the clock icon on the toolbar.

By default, when opening this type of Collection Services Investigator graph the data will be shown at the collected interval size for the best level of granularity. However, if this requires too much scrolling to the see a big picture view of this job, use the clock icon to effectively “zoom out”.



Selected Thread graph for a QZDASOINIT job

If viewing a Thread this type of graph contains a visual indicator that the thread had a holder for a particular interval. This is a ^ symbol on the X axis. There are 3 shown in the example above.

From any of these intervals you can double-click to go to the Interval details and view the call stack and holder information. Or you can also right-click the desired interval and a “Holder” menu will appear allow you to view the Holder’s call stack, perform a Holder chase or to graph the holder job over time.

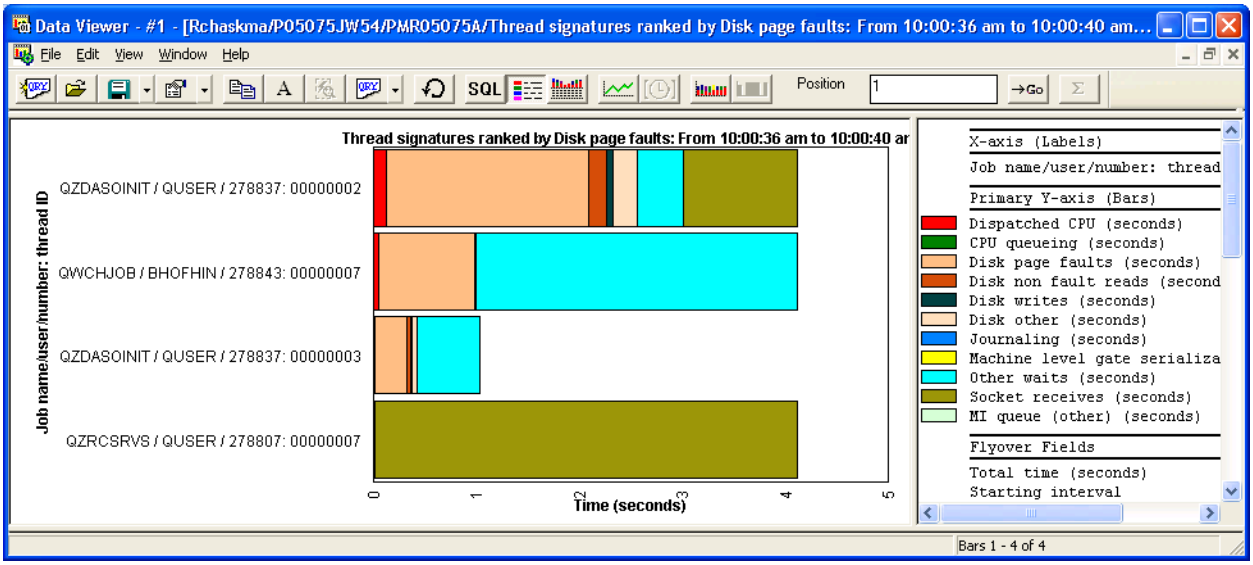
9.11.7.1 Drilling up

From the Selection over time graphs thread graphs you can select a time period of interest and right-click to have the same Rankings graphs options available to you but over the new time period. This allows you to navigate through your job over time, find something of interest and then compare that time period with the rest of the jobs on the system.

9.12 Rankings Graphs (via the Collection-Wide graphs)

This section covers the ranking graphs in Collection Services Investigator that are available as drilldowns from any of the Collection-wide graphs.

Ranking graphs show a list of objects (jobs, threads, units, etc) ranked by the desired metric.



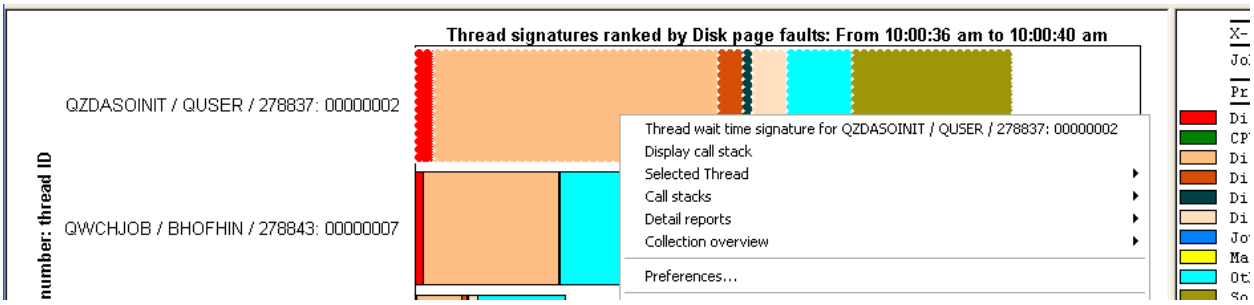
Threads ranked by disk page faults

Note: If the collection has NOT been summarized then the only ranking graphs available are the Thread ranking graphs.

9.12.1 Drilling down to Selected Thread/Job/etc graphs

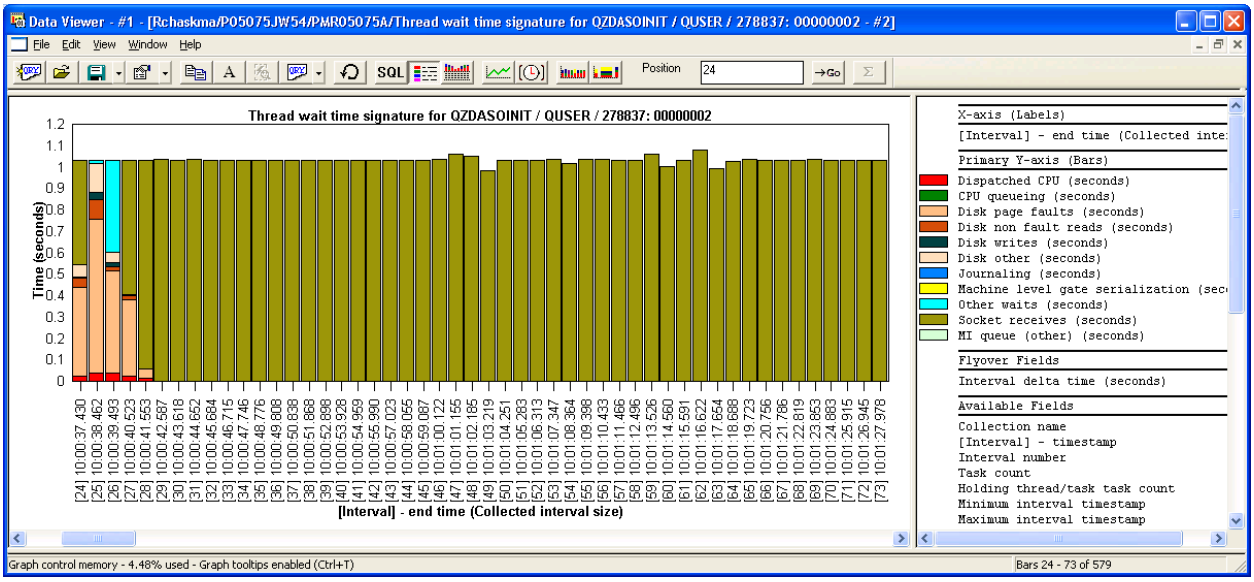
When drilling down from rankings graphs the default (top) menu is to show the current thread/job/job user/generic job, etc in a time interval graph. By default the time interval size is set to the collected interval size even if this is not your default preference. This was done deliberately since most users prefer to work with the collected interval size when dealing with this type of graph.

However, you can easily change this if desired by clicking the clock icon on the Data Viewer's toolbar.



Drilldown from a thread rankings graph to show a single job over time

To drill down simply right-click a job/thread/etc and choose the desired option. Typically the top menu is what you want, but you could also use the Selected Thread menu (name varies depending on the graph grouping) if you need to select a different graph than the one you are currently working with.



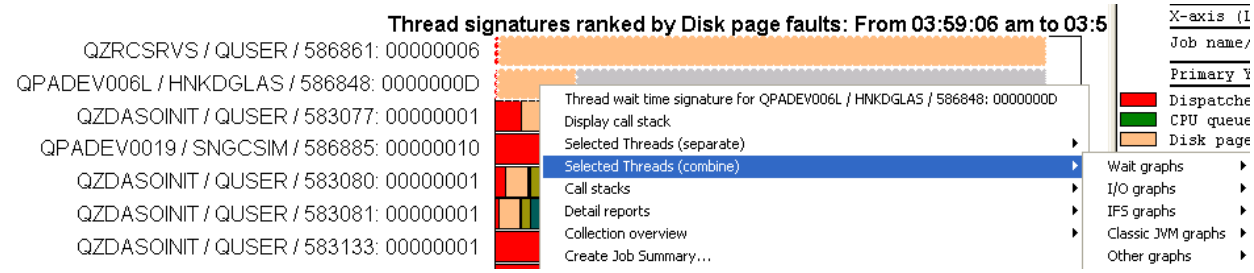
Thread wait time signature for a specific thread

In the example above the graph is showing the wait bucket times for a single thread over time. Also the graph is automatically scrolled to the start of the time interval displayed in the previous rankings graph.

9.12.2 Analyzing multiple threads/jobs/etc

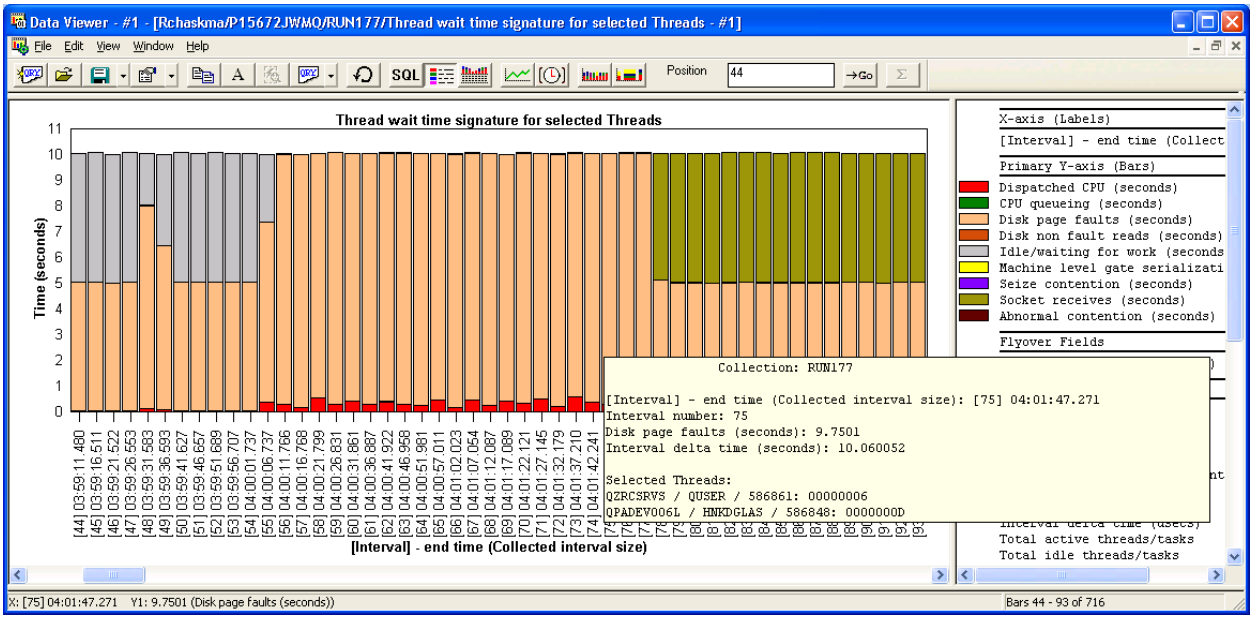
From a rankings graph you can hold down the Ctrl key and click multiple jobs/threads (depending on the graph grouping) and then when right-clicking you will have the option to combine the data from multiple jobs/threads/etc into a single graph. Just choose the menu option called Selected Thread (combine).

A result of this action looks like this:



Selected Threads (combine) menu

The resulting wait graph over time for these threads will combine the wait times from all selections into a single graph.



Graph combining data from multiple threads

The flyover will include the selections that make up the graph.

Also if you wish to select multiple jobs and have a different graph created for each one, (in one step), use the Selected Thread (Separate) menu.

9.12.3 Display call stack menu

The Display Call Stack menu from a rankings graph allows the user to quickly go to the call stack for the 1st interval in the time period indicated at the top of the rankings graph for the selected thread/task.

9.12.4 Call stacks menu

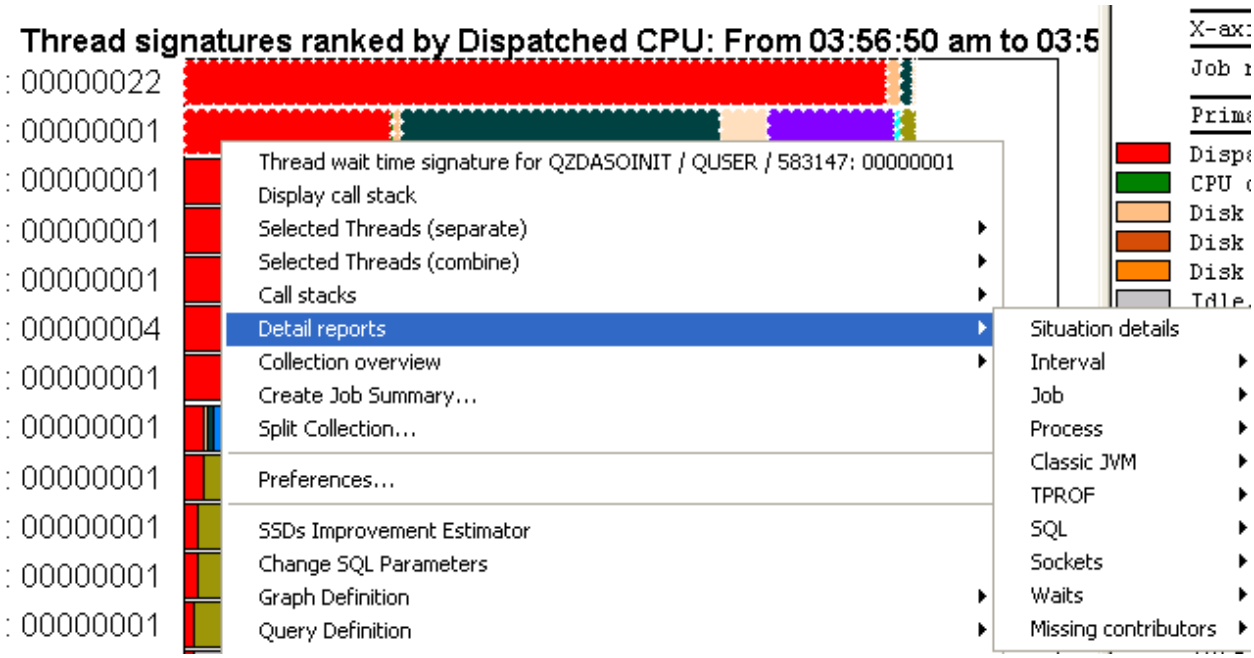
The Call Stacks menu is a special option that allows the user to display all the call stacks for all threads in the selected job from a rankings graph in various ways. This option was initially designed for use when analyzing JVM (Java), but could be used for any type of job if you pick the appropriate menu option from the list.

Menu	Description
Call stacks for selected job, interval X	This report shows all threads call stacks for the current job for the first interval of the time range selected (in the rankings graph)
Call stacks for selected job, interval X to Y	This report shows all threads call stacks for the current job for the entire time period selected in the rankings graph. Be careful not to pick too large of a time period for this one as the report could take a long time to generate.
Classic JVM call stacks for selected job, interval X	This report shows all threads call stacks for the current job for the first interval of the time range selected (in the rankings graph). This report contains extra fields specific to the classic JVM data. If the job selected is not a classic JVM job then this option won't provide data.

9.12.5 Drilling down into Detail reports

Another drilldown option from the Rankings graphs is found under a menu called "Detail reports". This menu offers a series of table views that provide quick access to many of the raw data found in the collection.

An example of this menu and list of report categories it contains is:



All of these reports are filtered on the time period selected by the rankings graph as well as the current selections (jobs/threads) from the graph.

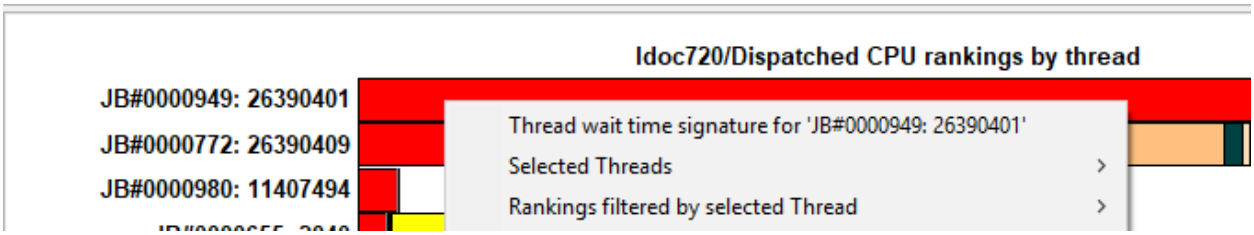
PLEASE NOTE: This drill down option is only visible if the preference on the Job Watcher tab called "Display advanced reporting options" is enabled.

9.12.6 Collection overview menu

This is identical to the menu described previously.

9.12.7 Job selection overtime graphs

These graphs are initiated as a drill-down from any ranking graph. Typically, these can be most-easily accessed by right-clicking on a job in the ranking graph and using the first menu option that appears. This will by default show the same graph but for the selection and over time.



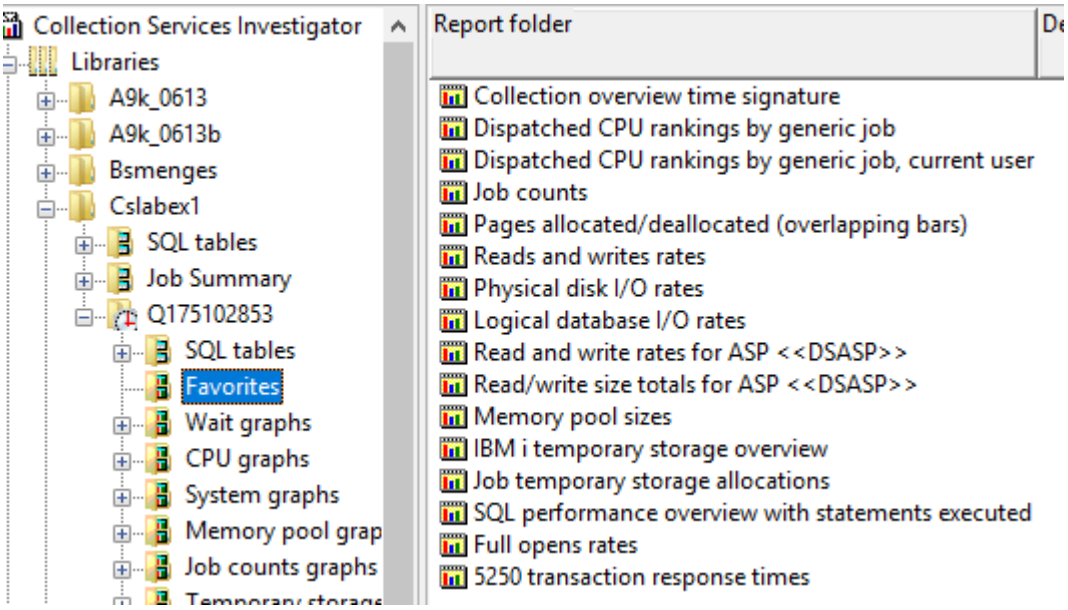
Dispatched CPU rankings drill down options

Tip: If another graph type is desired then use the Selected Threads menu or Rankings filtered by selected Thread options.

9.13 Favorites

This folder contains a list of graphs most commonly used and are great starting points. If you are new to iDoctor these are your recommended graphs to use.

Note: Some graphs only appear at certain releases or require the Collection Summary analysis to be ran.

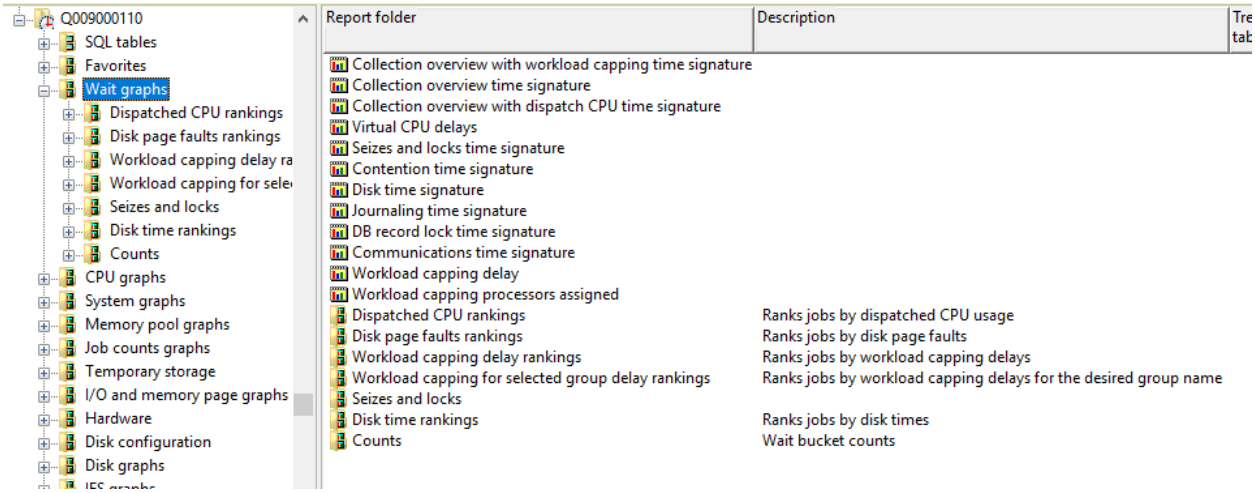


Favorites folder

Graph name	Notes
Collection overview time signature	
Dispatched CPU rankings by generic job	
Dispatched CPU rankings by generic job, current user	
Job counts	
Pages allocated/deallocated - overlapping bars	
Read and writes rates	
Physical disk I/O rates	
Logical database I/O rates	
Read and write rates for ASP X	
Read/write size totals for ASP X	
Memory pool sizes	
IBM i temporary storage overview	
Job temporary storage allocations	
SQL performance overview with statements executed	
Full opens rates	
5250 transaction response times	

9.14 Wait graphs

These graphs show running and waiting time across all jobs in the collection over time. These graphs are wait bucket graphs which divides up the wait times into various buckets. These buckets contain enums which are the individual wait types grouped into each bucket. The wait buckets and enums are visible from the Wait Buckets tab of the Collection's Properties.



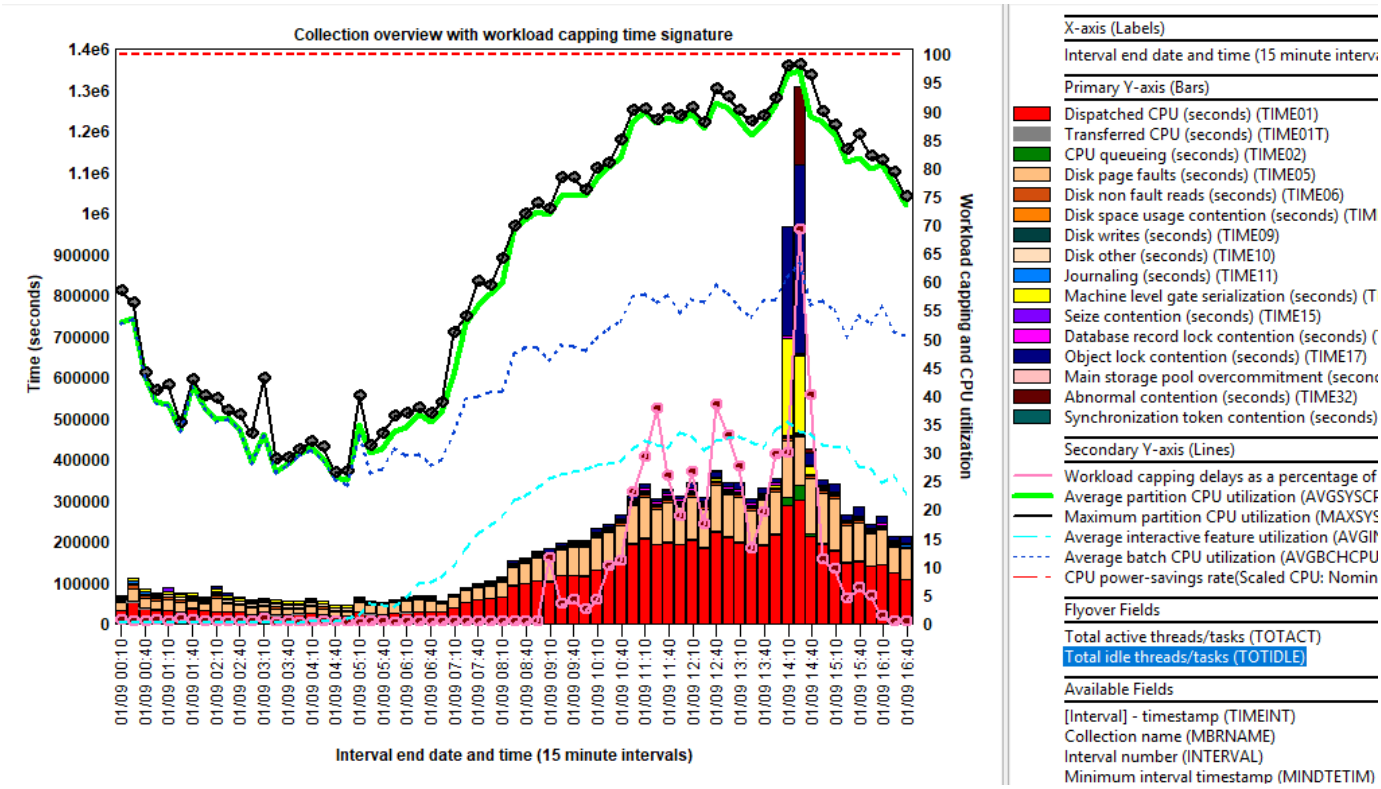
Wait graphs folder

Typically, the graph most users start with is the Collection overview time signature graph or one of its variations.

9.14.1 Collection overview with workload capping time signature

This graph displays the collection overview time signature graph containing the "interesting" wait bucket and CPU time except it also contains a value showing the effects of workload capping as a percentage of total CPU queuing. The higher the value, the more workload capping delays that occurred. In this example, the pink line shows workload capping delays.

Note: This graph will only appear if file QAPMSYSWLC exists.

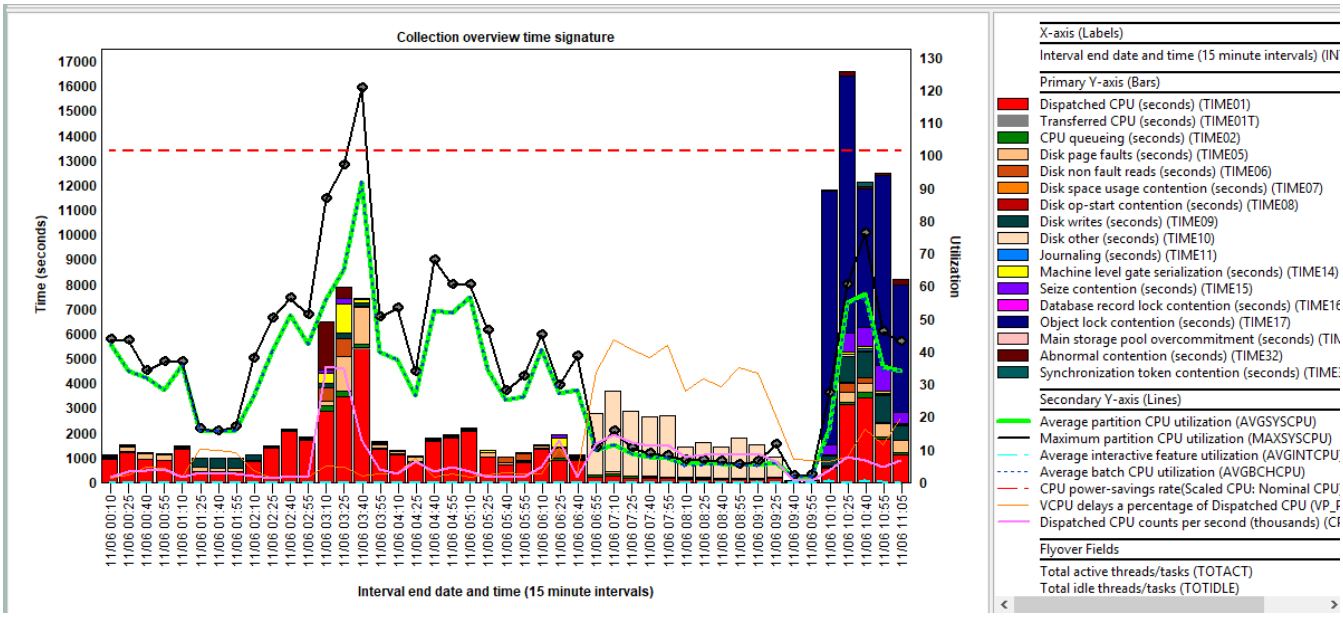


Collection overview with workload capping time signature

If desired, the user could select the desired time period, right-click and graph the threads or other job groupings that contributed to the workload capping delays via the **Rankings -> Wait graphs -> Workload capping delay rankings** menu.

9.14.2 Collection overview time signature

This graph shows CPU time and the [“interesting” wait bucket] times added together across all jobs on the system. In the example below, a user could right-click an interval(s) on the desired wait bucket to drill down and show the jobs that experienced the highest amount of time in the desired wait bucket.



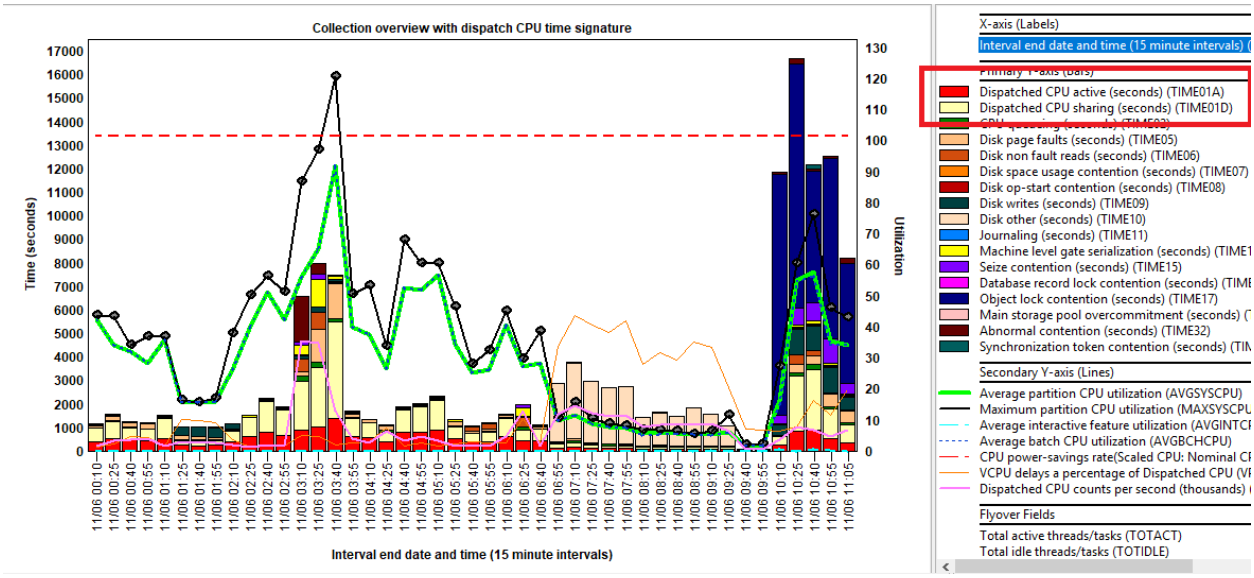
Collection overview time signature

9.14.3 Collection overview with dispatch CPU time signature

This graph is identical to the Collection overview time signature graph except the CPU time is divided into 2 different buckets:

Dispatched CPU active (red) – This is time spent burning CPU.

Dispatched CPU sharing (light yellow) – This is a type of time we can measure where we are dispatched to the processor but NOT actually burning CPU. It is being shared with other tasks. This is perfectly normal to have a large amount of time spent in this bucket and does not likely indicated a performance problem. It is provided for advanced users.



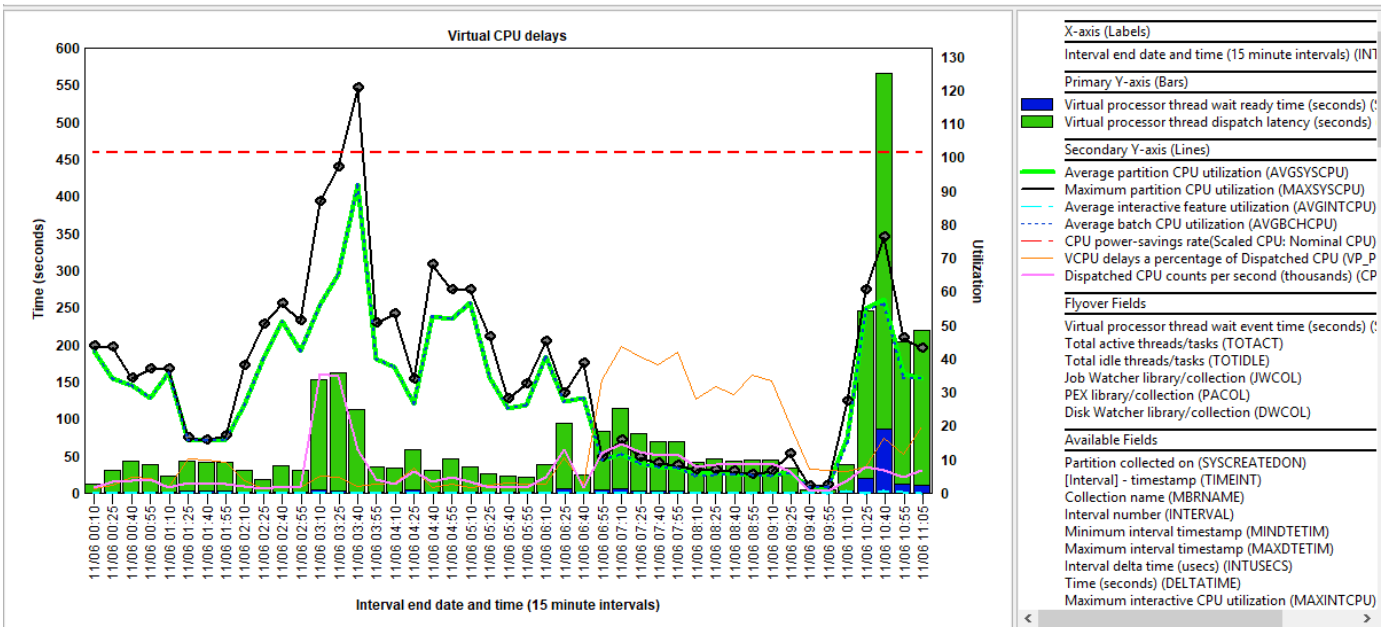
Collection overview with dispatch CPU time signature

9.14.4 Virtual CPU delays

This graph highlights the amount of time spent by virtual processors waiting on 2 types of waits.

These fields are wait ready time and dispatch latency time. The detailed descriptions for the underlying fields are copied from the Knowledge center below:

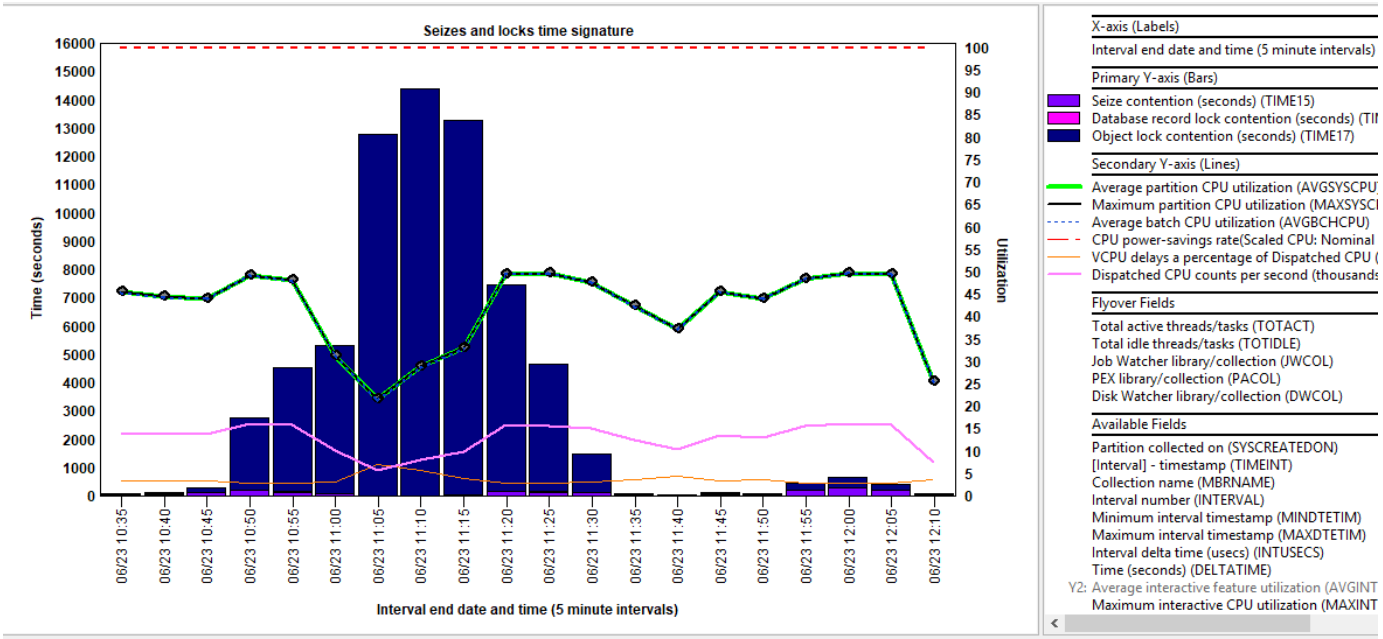
SYPTREADY	Virtual processor thread wait ready time. The elapsed time in microseconds that ready to run threads of the partition's virtual processors waited to be dispatched while entitled capacity was exhausted.
SYPTLATEN	Virtual processor thread dispatch latency. The elapsed time in microseconds that ready to run threads of the partition's virtual processors waited to be dispatched while entitled capacity was not exhausted and a physical processor was not available.



Virtual CPU delays

9.14.5 Seizes and locks time signature

This graph just shows seizes, record locks and object lock contention times.



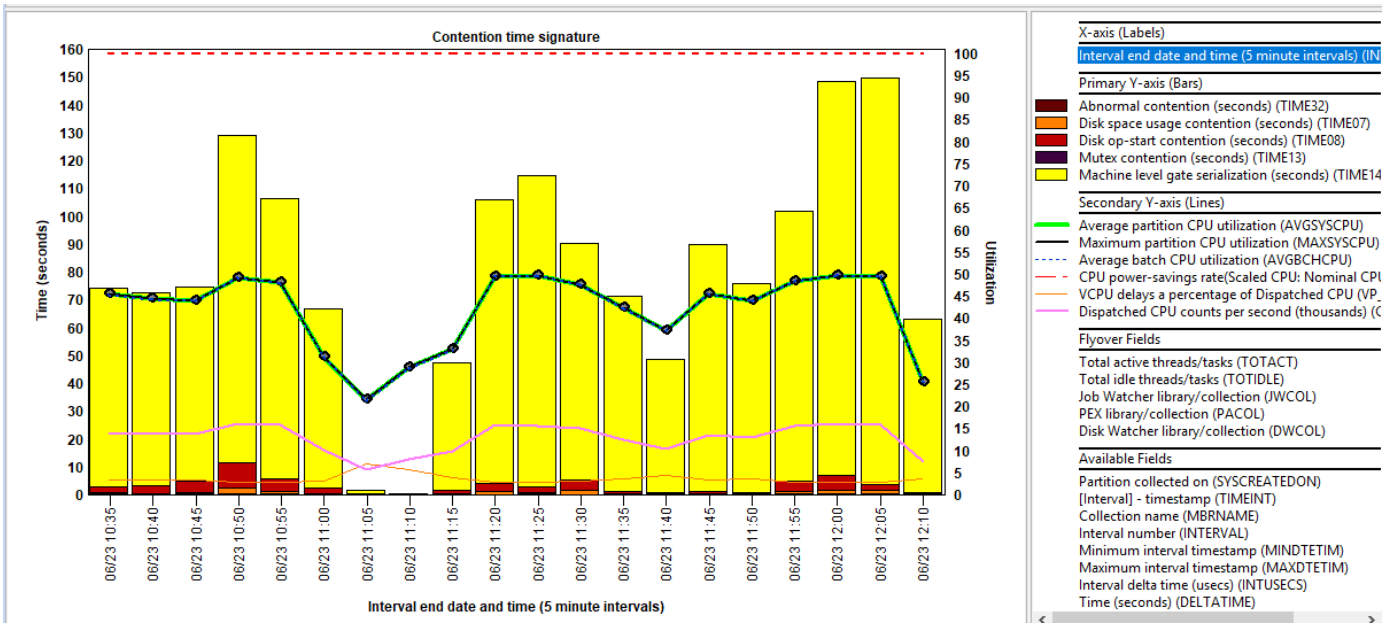
Seizes and locks time signature

9.14.6 Contention time signature

This graph only shows several wait buckets that are often associated with some type of contention on the system. It shows the following wait buckets:

1) Abnormal contention (32)

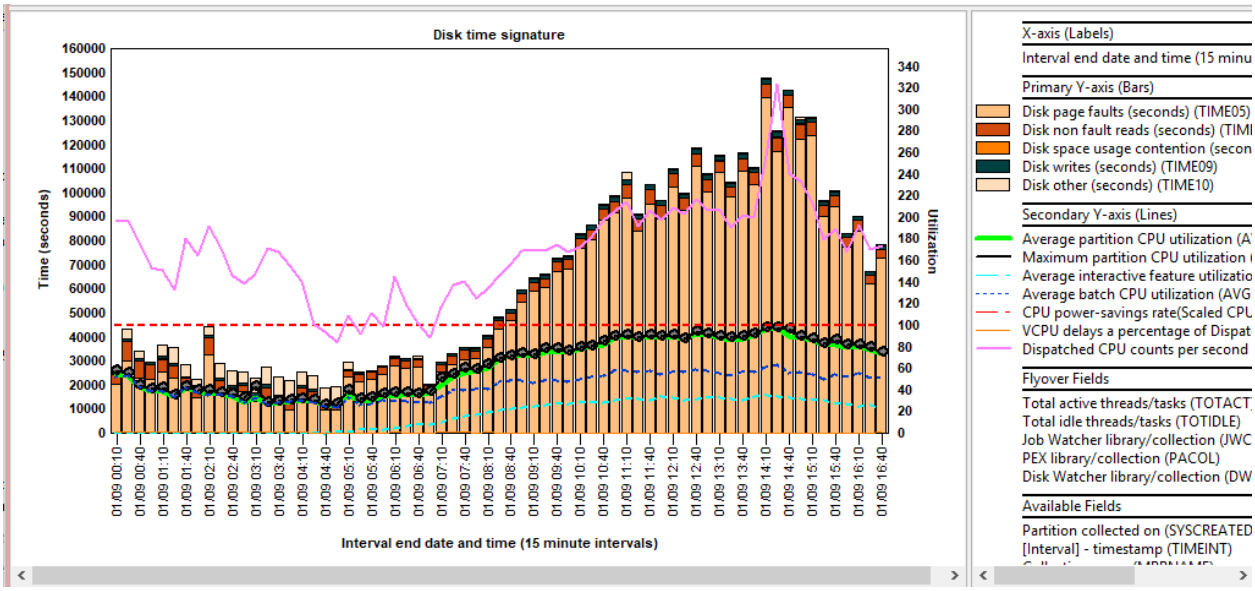
2) Disk space usage contention (7)
3) Disk op-start contention (8)
4) Mutex contention (13)
5) Machine level gate serialization (14)



Contention time signature

9.14.7 Disk time signature

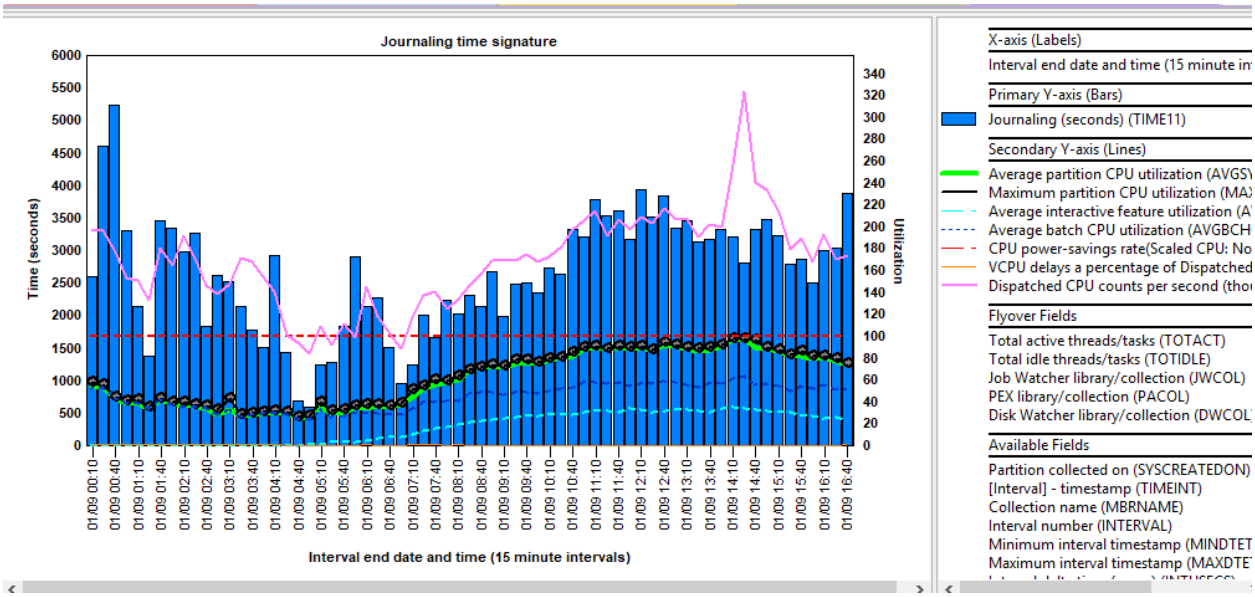
This graph just shows wait bucket times related to disk I/O or disk contention.



Disk time signature

9.14.8 Journaling time signature

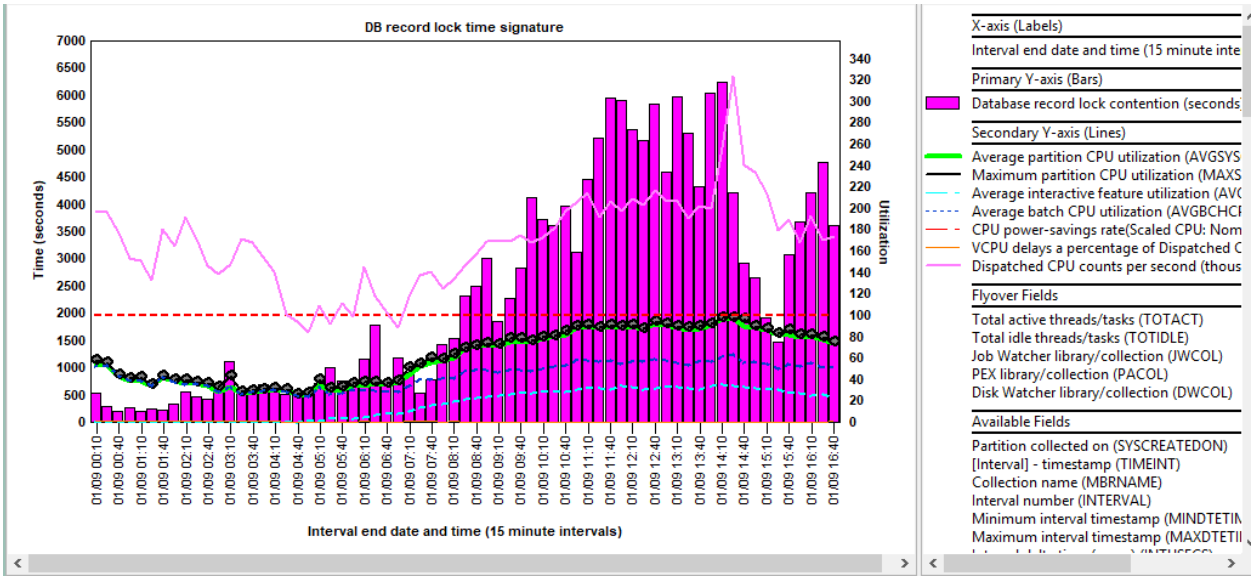
This graph displays waits related to journaling. Two wait buckets are included: Journaling and Journal save while active. If the later wait bucket is NOT visible, this just means no time was spent in that wait bucket.



Journaling time signature

9.14.9 DB record lock time signature

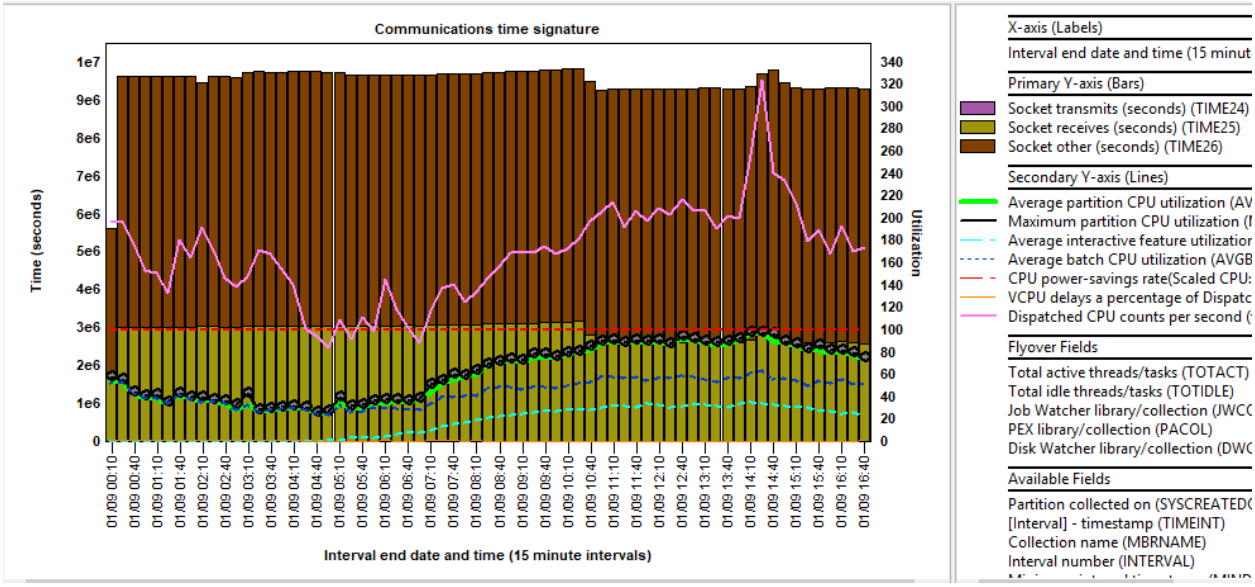
This graph shows the total time spent in Database record locks.



DB record lock time signature

9.14.10 Communications time signature

These waits indicate time waiting to receive or send data or other types of socket waits. An example of a socket receive is what a QZRCSRVS job (iDoctor remote command job servicing the GUI) will do when it is idle waiting for requests from the PC. Once the job receives data over the comm line other types of ***non-idle*** waits (CPU, disk IO) will be shown.

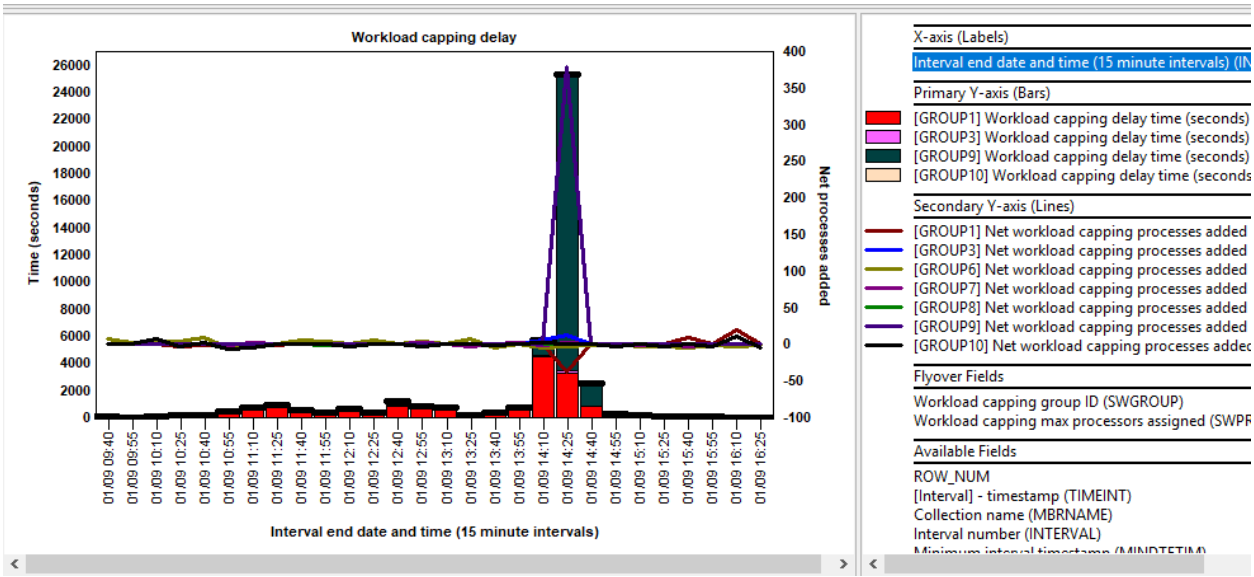


Communications time signature

9.14.11 Workload capping delay

This graph shows the various workload capping groups defined on the system during the collection and how much times jobs did not run in those groups because of workload capping. The secondary Y-axis also shows the net number of jobs/tasks/threads were added or removed (negative values) to each group.

Note: This graph will only appear if file QAPMSYSWLC exists.

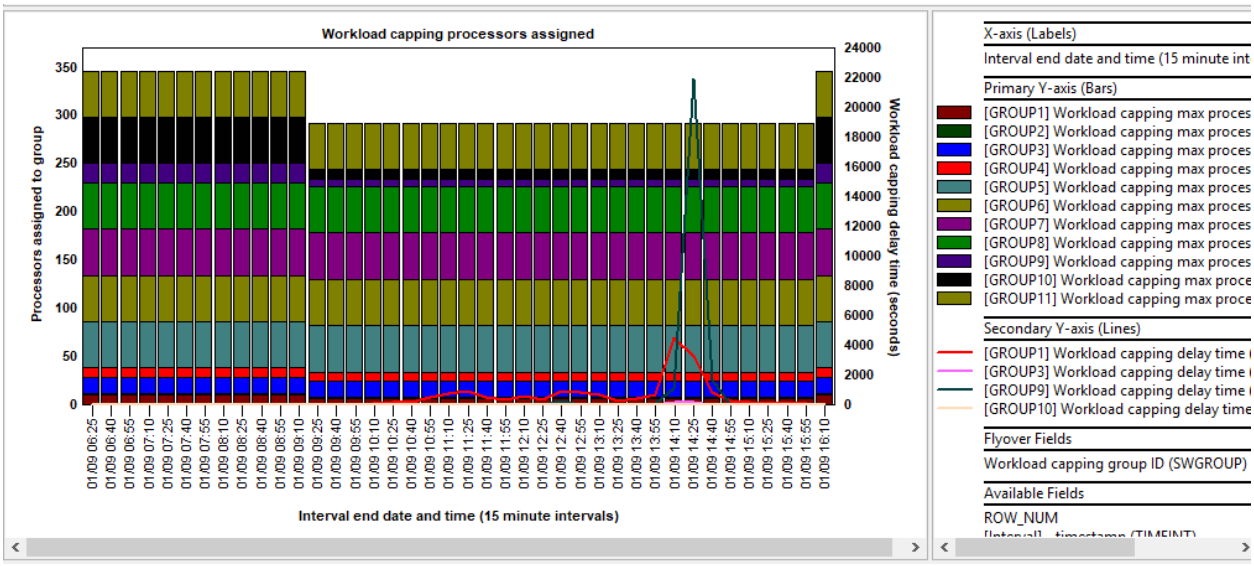


Workload capping delay

9.14.12 Workload capping processors assigned

This graph shows the various workload capping groups defined on the system during the collection and how many processors were assigned to each group. The secondary Y-axis also shows the amount of workload capping delay time experienced by each group.

Note: This graph will only appear if file QAPMSYSWLC exists.

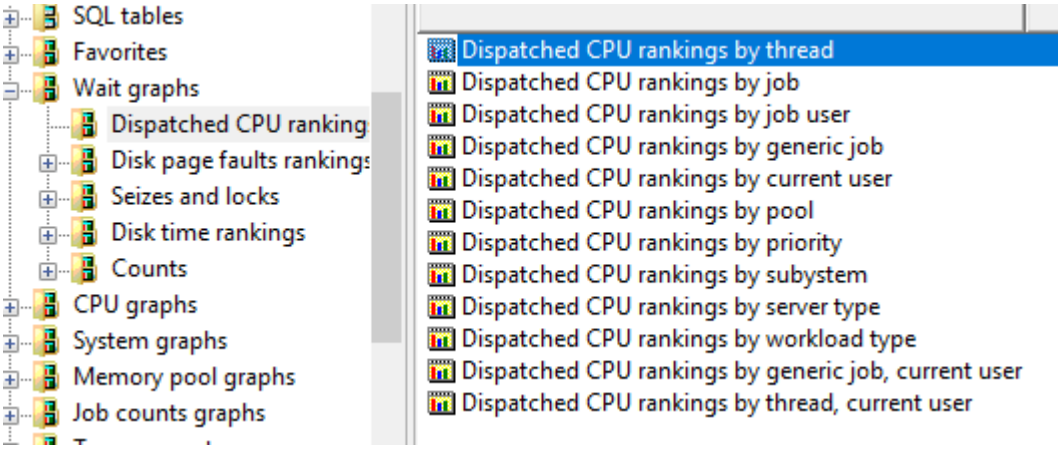


Workload capping processors assigned

9.14.13 Dispatched CPU rankings

Under the Wait graphs folder is the Dispatched CPU rankings subfolder which contains a set of wait bucket ranking graphs sorted by Dispatched CPU time over the entire collection. One graph is listed for each type of job grouping available.

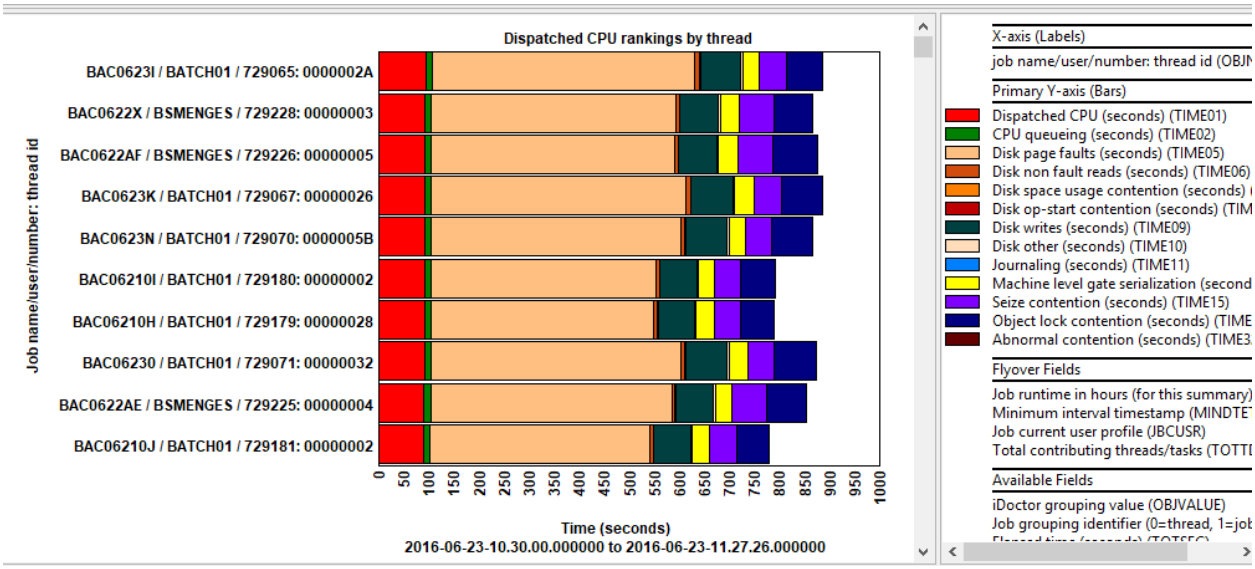
The graph only shows data for jobs that used at least some CPU during the collection.



Wait graphs -> Dispatched CPU rankings

9.14.13.1 Dispatched CPU rankings by thread

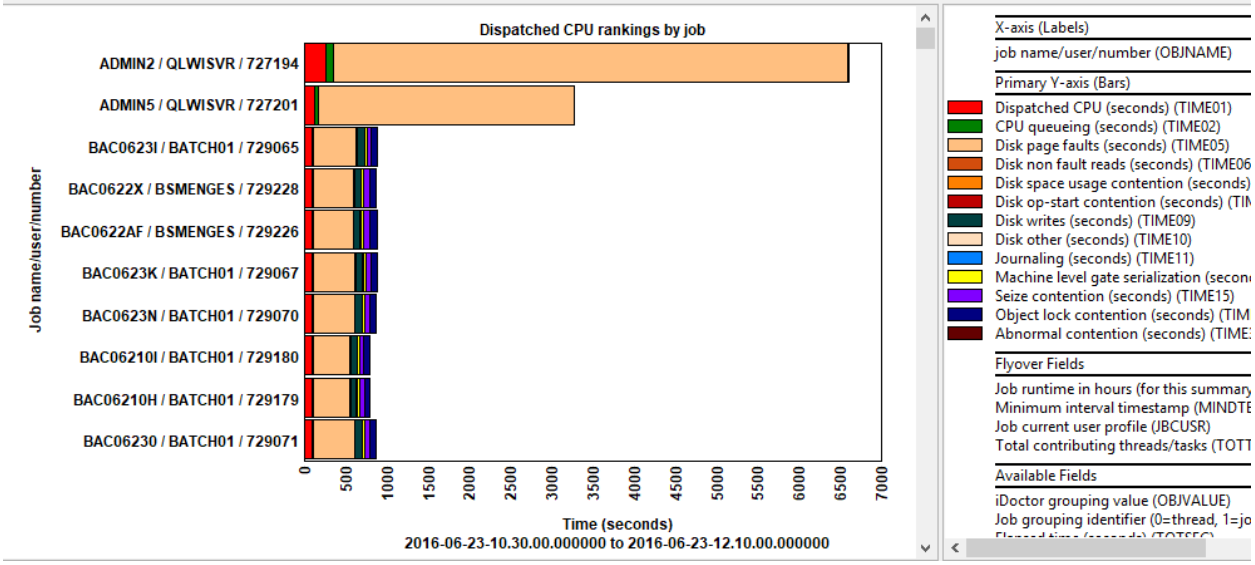
This graph shows the job/threads or system tasks that used the most CPU during the collection. Right-click to show options to graph it over time.



Dispatched CPU rankings by thread

9.14.13.2 Dispatched CPU rankings by job

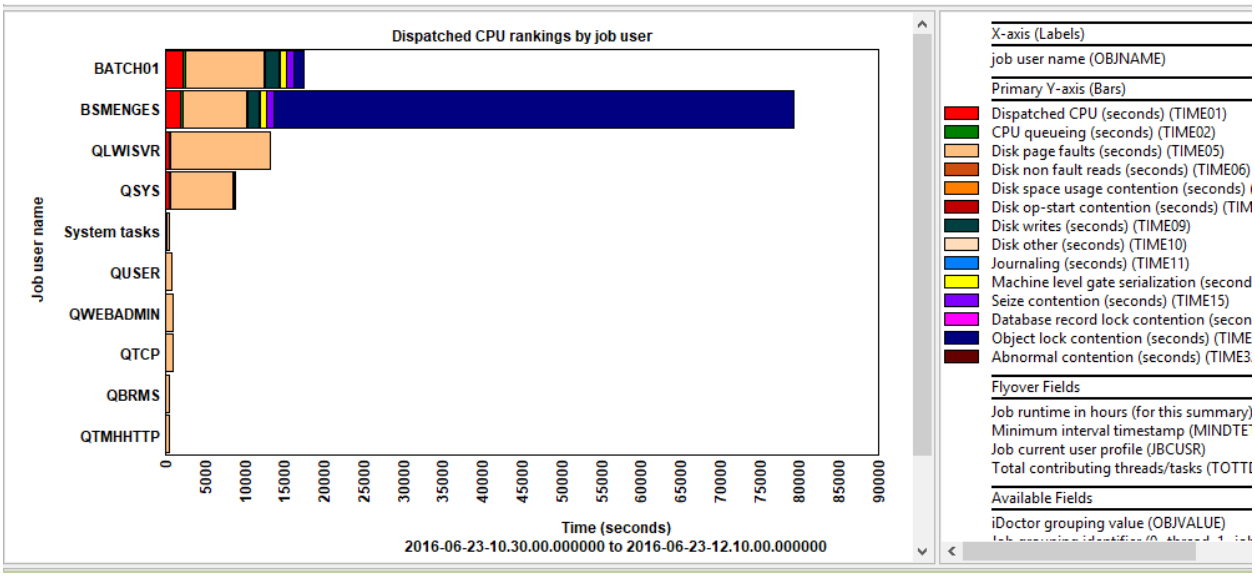
This graph shows the jobs (all threads added together) that used the most CPU during the collection.



Dispatched CPU rankings by job

9.14.13.3 Dispatched CPU rankings by job user

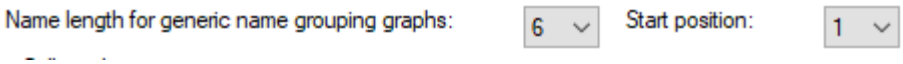
This graph shows the job user name (all jobs having the same user name added together) that used the most CPU during the collection.

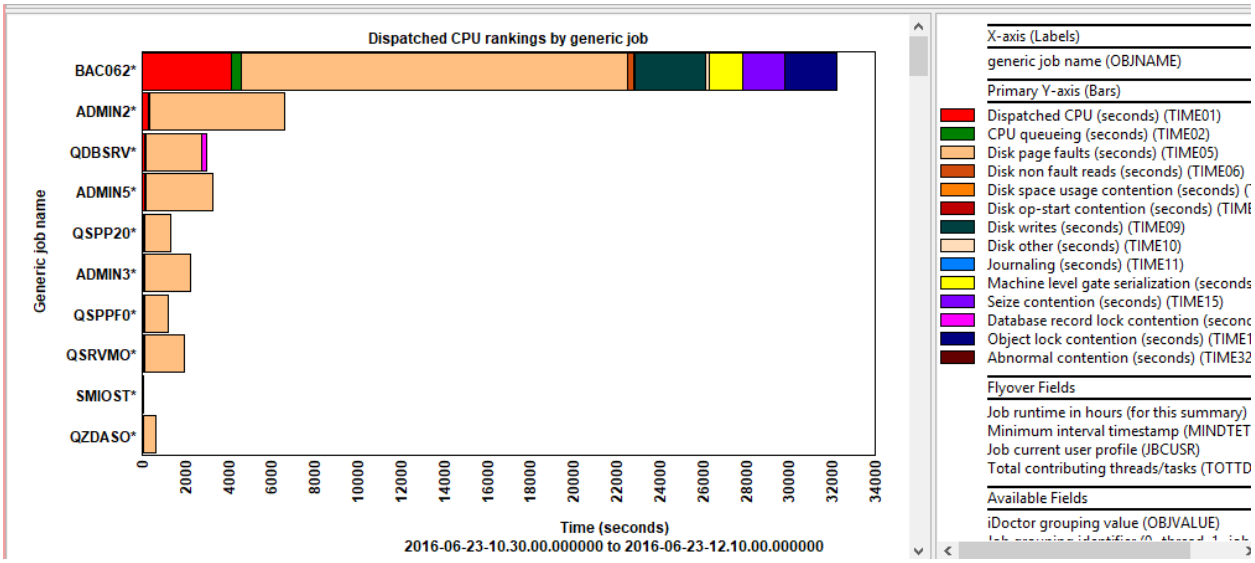


Dispatched CPU rankings by job user

9.14.13.4 Dispatched CPU rankings by generic job

This graph ranks the selected time period's wait bucket data by Dispatched CPU and generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

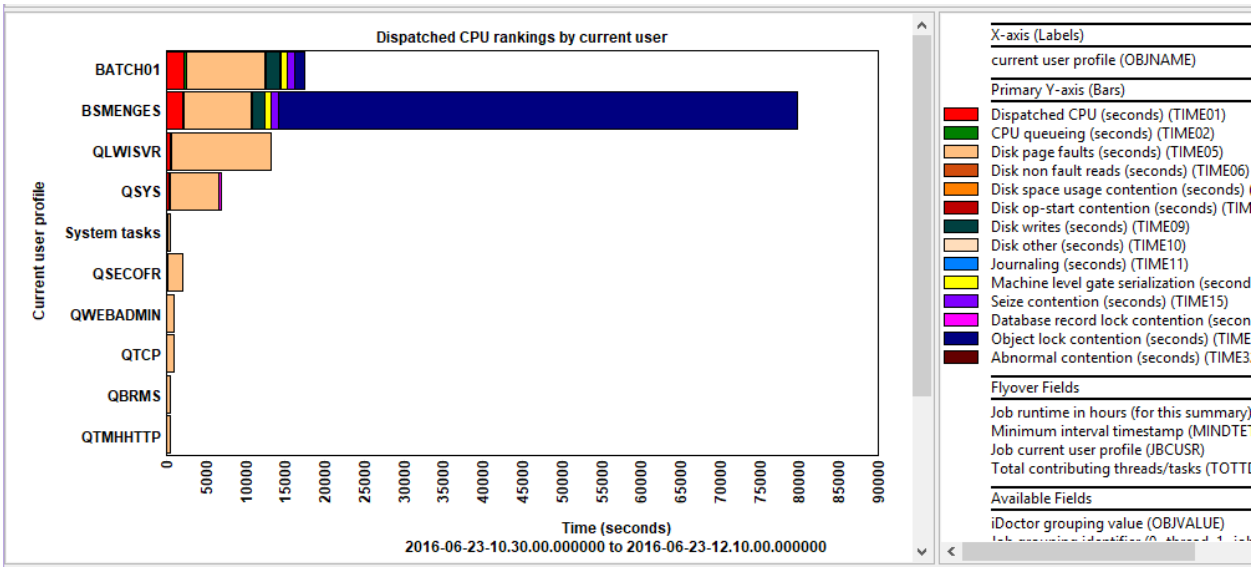




Dispatched CPU rankings by generic job

9.14.13.5 Dispatched CPU rankings by current user

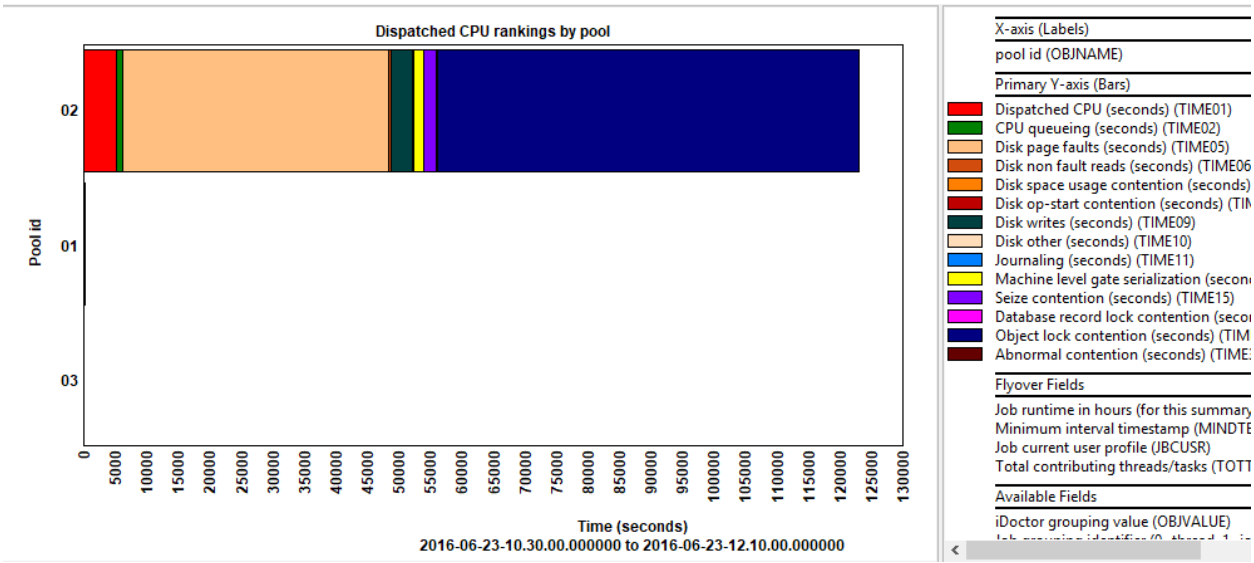
This graph ranks the selected time period's wait bucket data by Dispatched CPU and current user profile.
Note: All system tasks are grouped together into one "**System tasks"** record within this report.



Dispatched CPU rankings by current user

9.14.13.6 Dispatched CPU rankings by pool

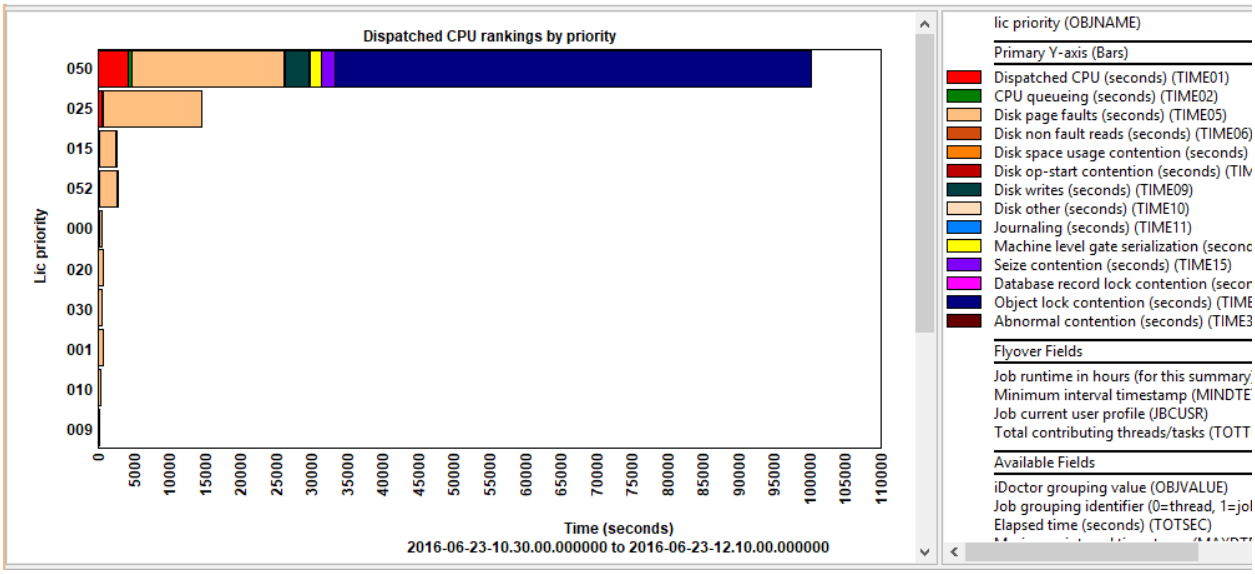
This graph shows the pool ID (all jobs in each pool added together) that used the most CPU during the collection.



Dispatched CPU rankings by pool

9.14.13.7 Dispatched CPU rankings by priority

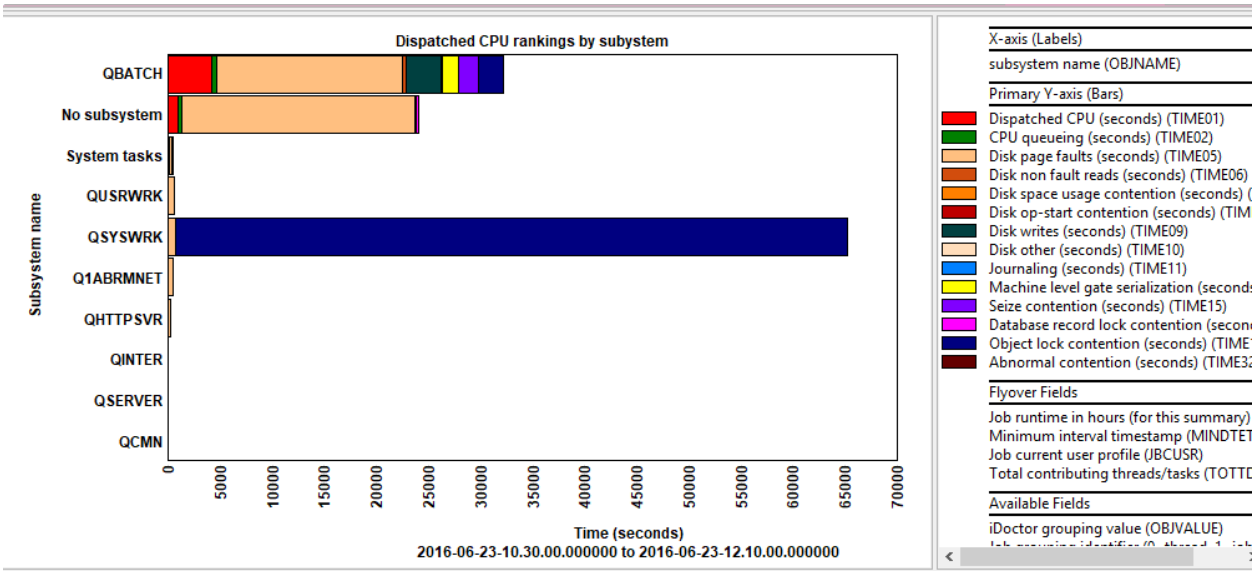
This graph shows the job priority (all job time periods that had the same priority added together) that used the most CPU during the collection.



Dispatched CPU rankings by priority

9.14.13.8 Dispatched CPU rankings by subsystem

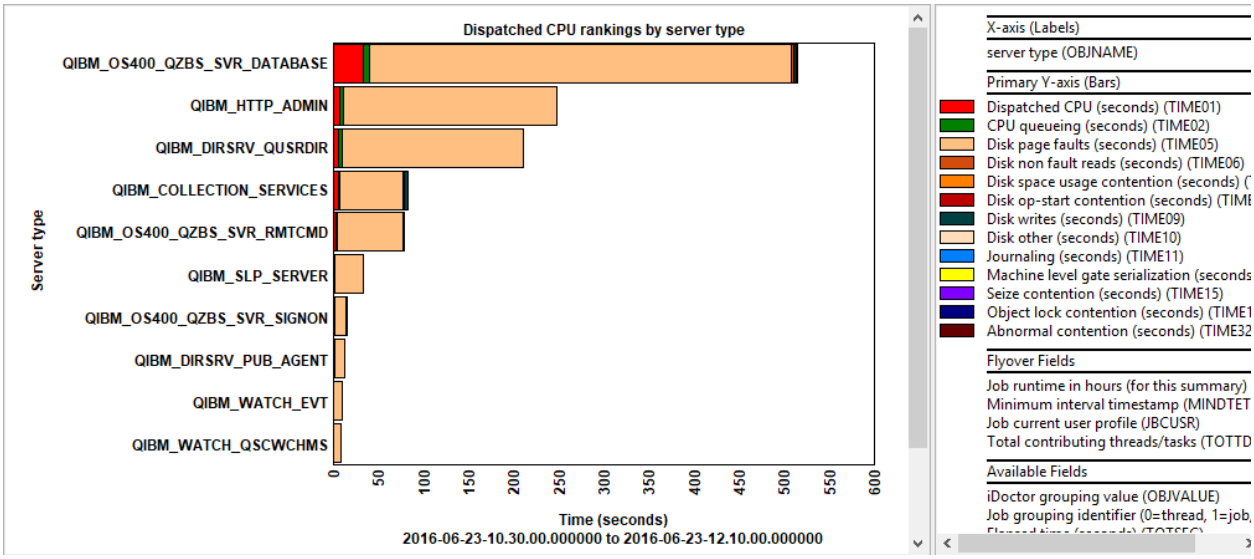
This graph shows the subsystems that used the most CPU during the collection.



Dispatched CPU rankings by subsystem

9.14.13.9 Dispatched CPU rankings by server type

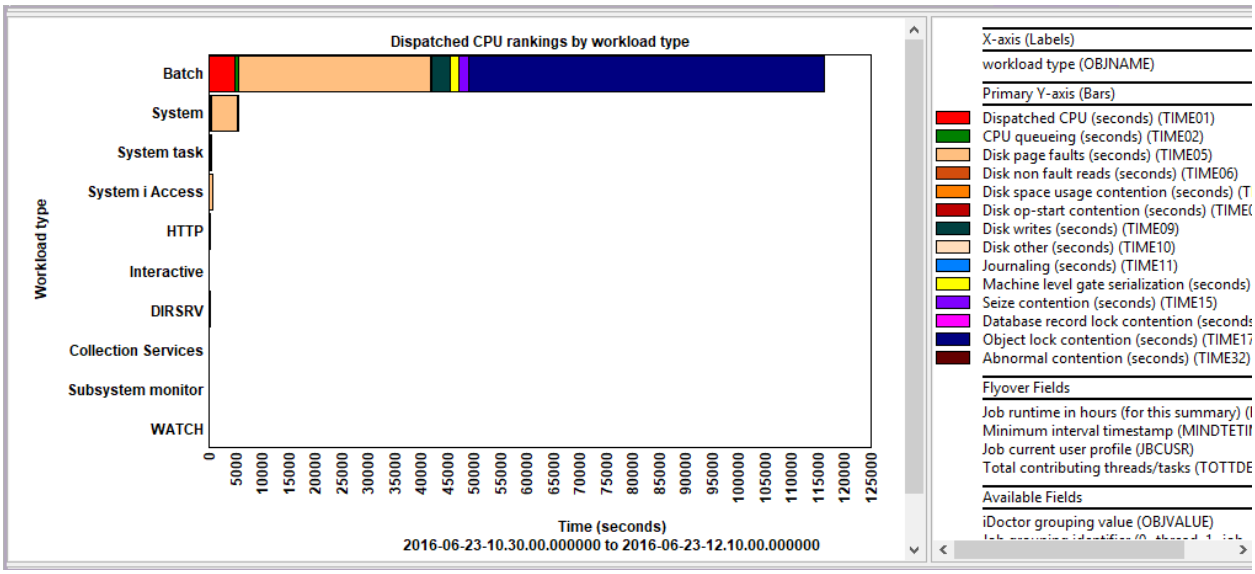
This graph ranks the Dispatched CPU time contributions during the collection by server type.



Dispatched CPU rankings by server type

9.14.13.10 Dispatched CPU rankings by workload type

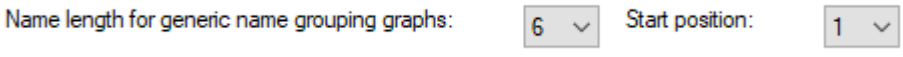
This graph ranks the Dispatched CPU time contributions by workload type.

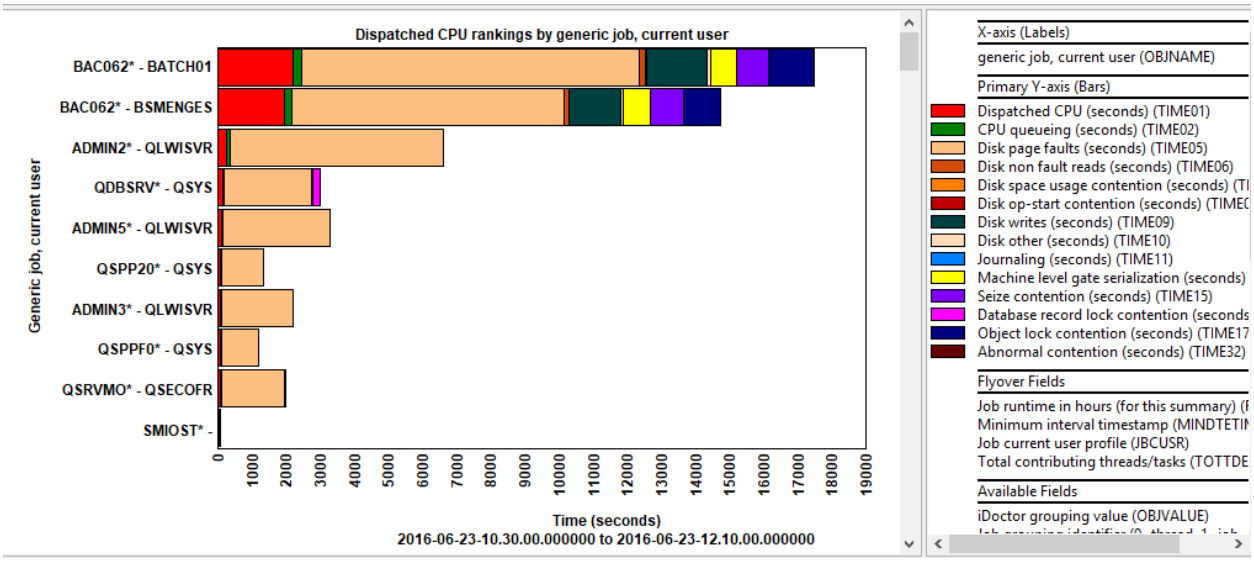


Dispatched CPU rankings by workload type

9.14.13.11 Dispatched CPU rankings by generic job, current user

This graph ranks the selected time period's wait bucket data by Dispatched CPU and a combination of generic job and current user profile. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

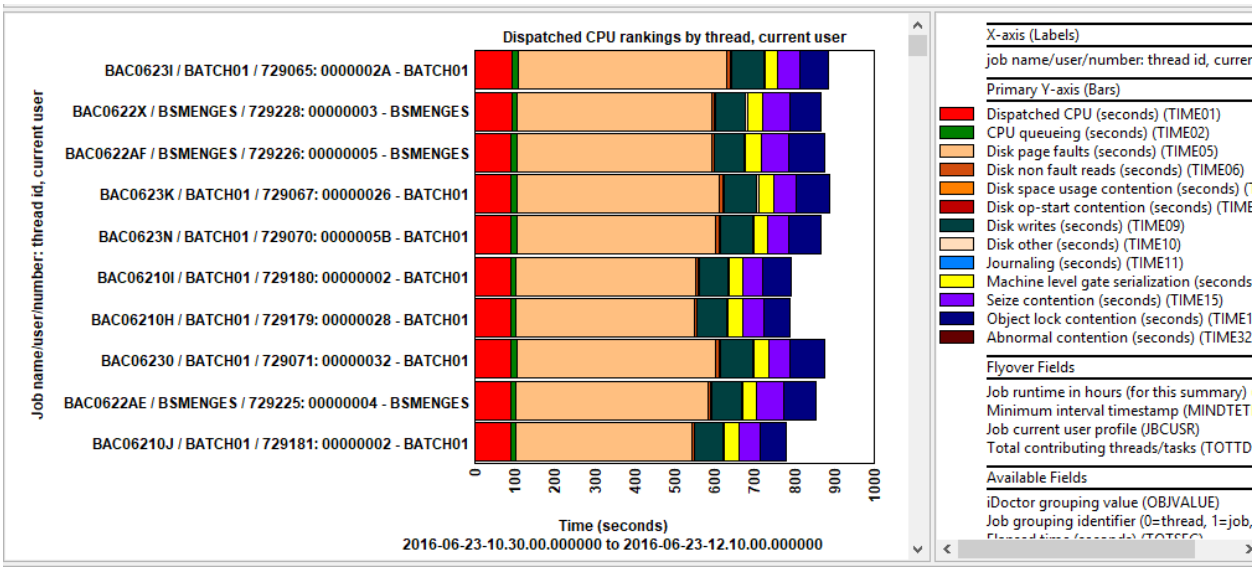




Dispatched CPU rankings by generic job, current user

9.14.13.12 Dispatched CPU rankings by thread, current user

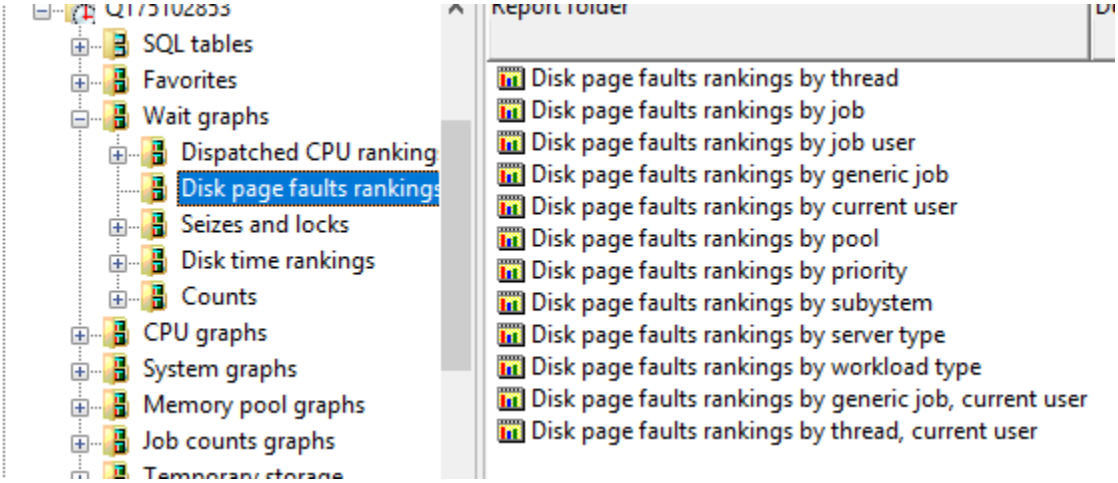
This graph shows a combination of thread and current user ranked by those that used the most CPU during the collection.



Dispatched CPU rankings by thread, current user

9.14.14 Disk page fault rankings

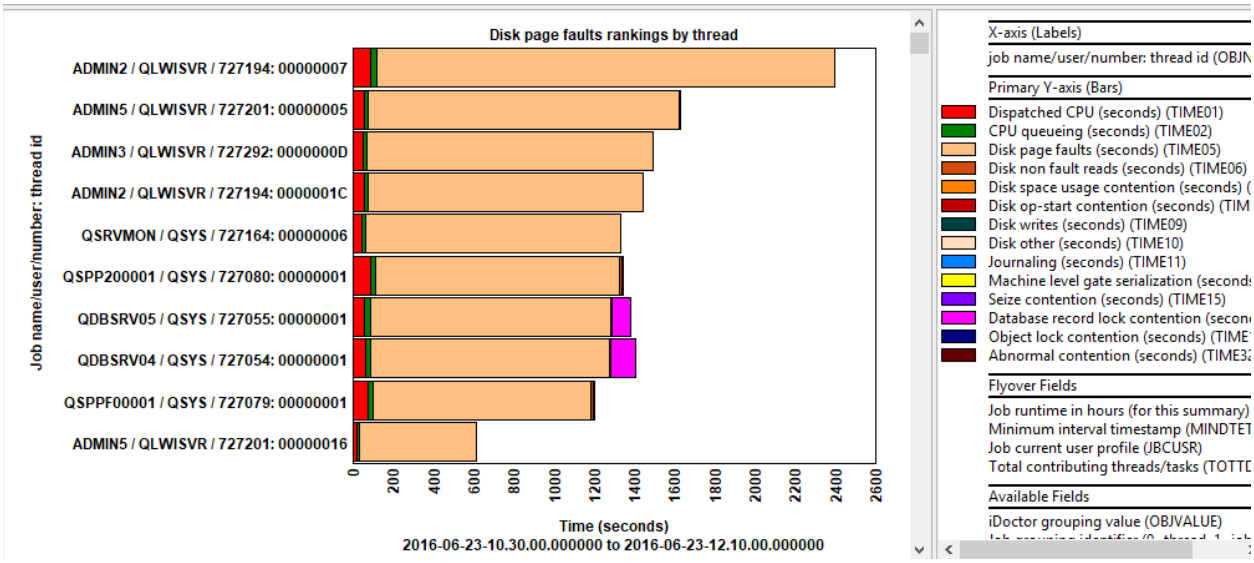
Under the Wait graphs folder is the Disk page fault rankings subfolder which contains a set of wait bucket ranking graphs sorted by disk page faults time over the entire collection. One graph is listed for each type of job grouping available.



Wait graphs -> Disk page faults rankings

The graph only shows data for jobs that had at least some disk page fault time during the collection.

These graphs are the same as the Dispatched CPU rankings graphs with the only difference being these graphs rank disk page faults instead of Dispatched CPU time.

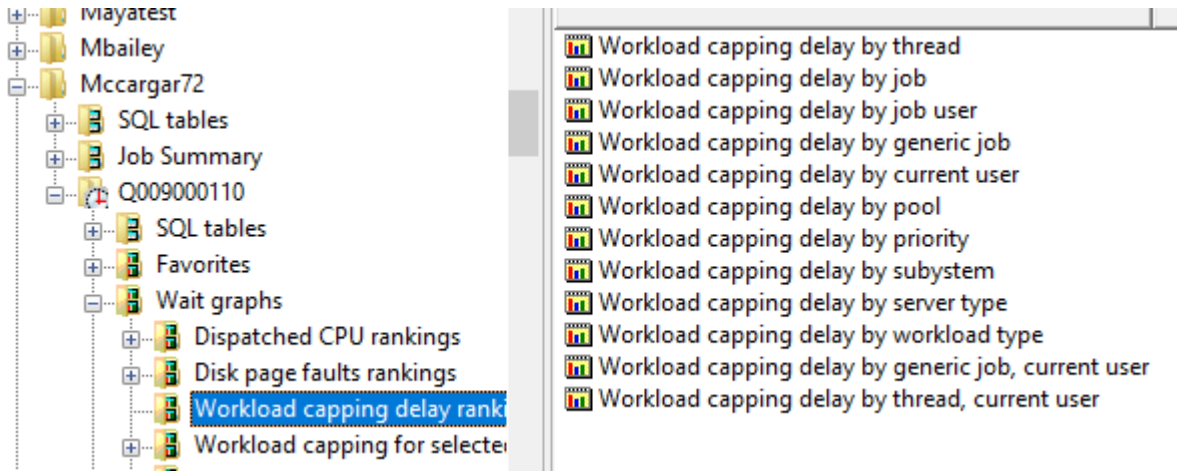


Disk page faults rankings by thread

9.14.15 Workload capping delay rankings

This set of graphs show the effects of workload capping for the desired job grouping and ranks the jobs by those experiencing the most delay time. These graphs include data for ANY workload capping groups found in the collection.

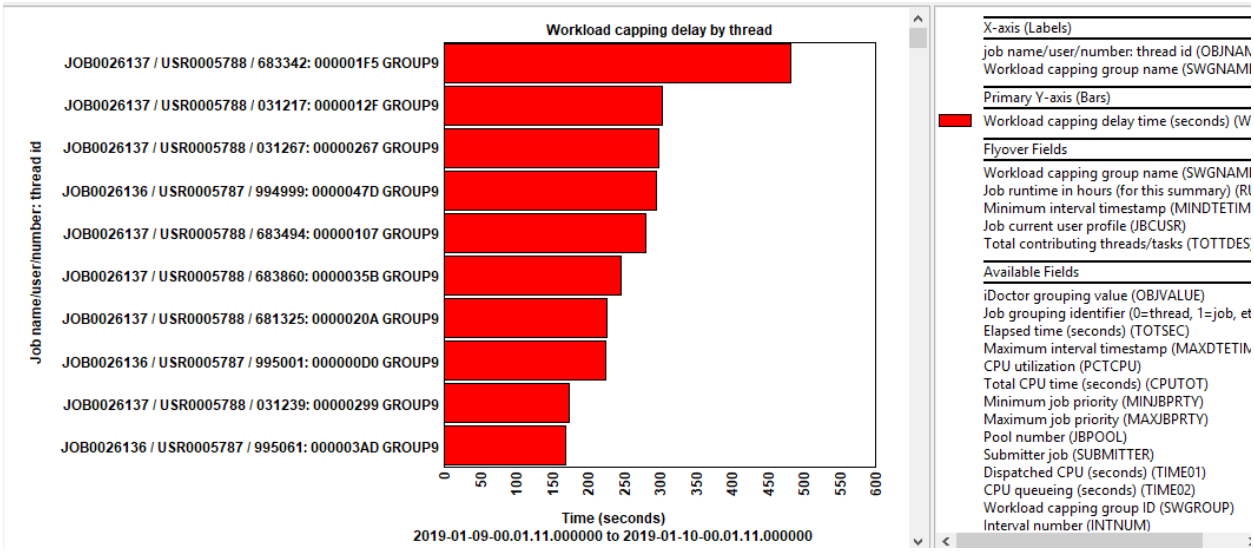
NOTE: The data in these graphs is grouped by the job grouping AND workload capping group name.



Wait graphs -> Workload capping delay rankings graphs

9.14.15.1 Workload capping delay by thread

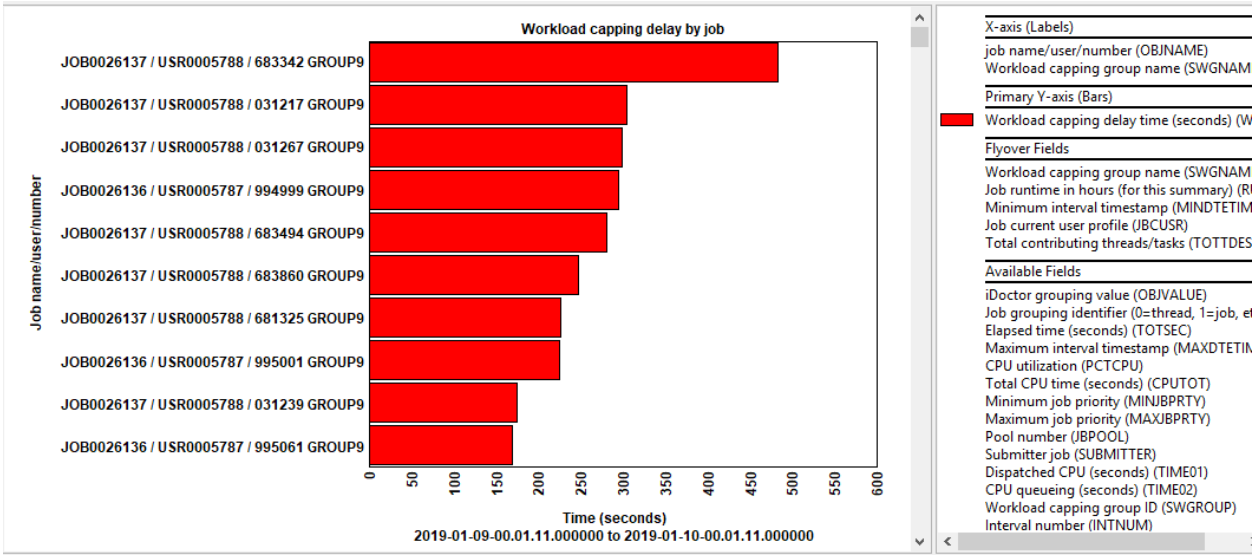
This graph shows the job/thread and workload capping group name that experienced the most delays due to the effects of workload capping.



Workload capping delay by thread

9.14.15.2 Workload capping delay by job

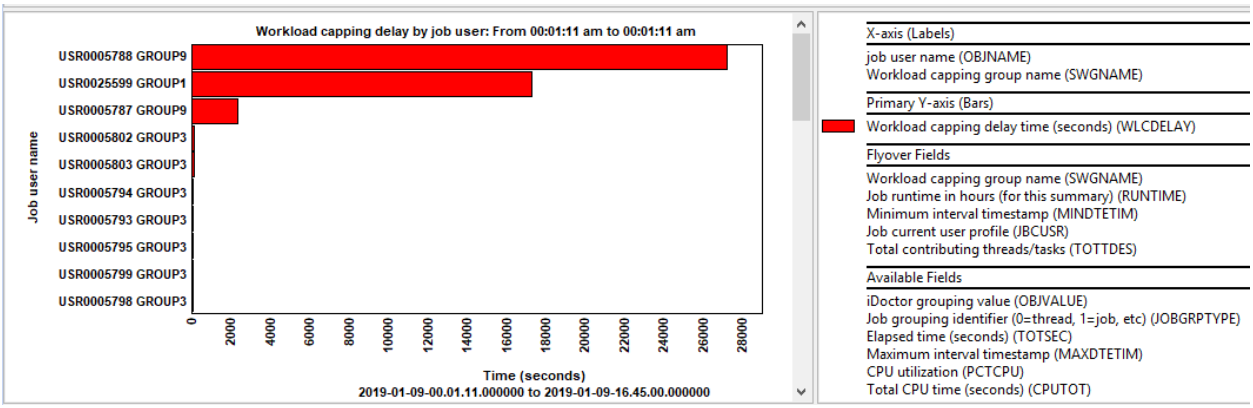
This graph shows the jobs (all threads added together) and workload capping group name that experienced the most delays due to the effects of workload capping.



Workload capping delay by job

9.14.15.3 Workload capping delay by job user

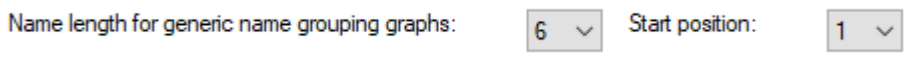
This graph shows the job users and workload capping group name that experienced the most delays due to the effects of workload capping.

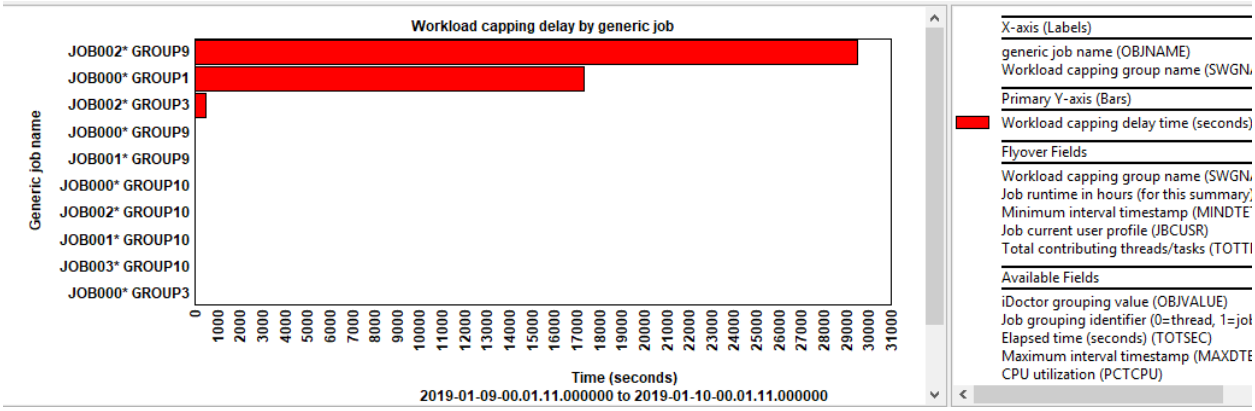


Workload capping delay by job user

9.14.15.4 Workload capping delay by generic job

This graph ranks the selected time period's workload capping delays by workload capping group name and generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

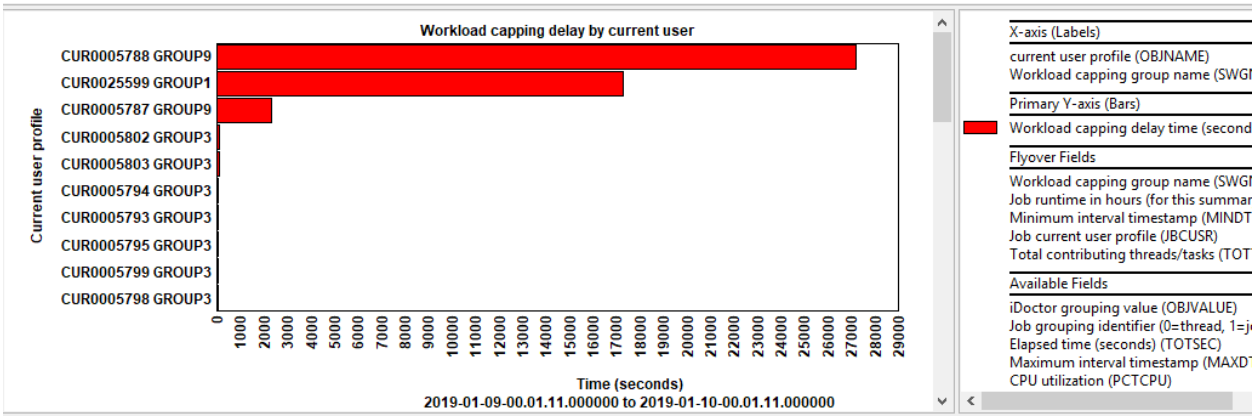




Workload capping delay by generic job

9.14.15.5 Workload capping delay by current user

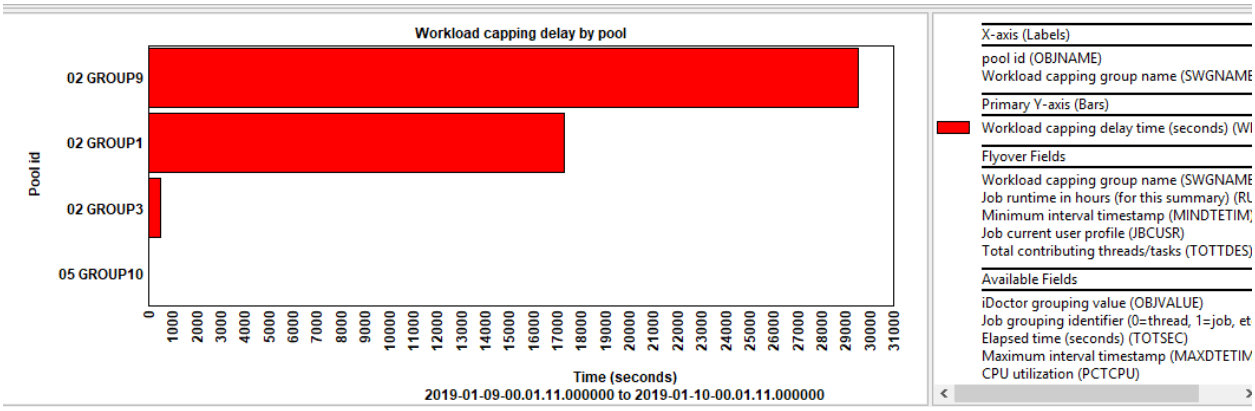
This graph ranks the selected time period's workload capping delays by workload capping group name and current user profile.



Workload capping delay by current user

9.14.15.6 Workload capping delay by pool

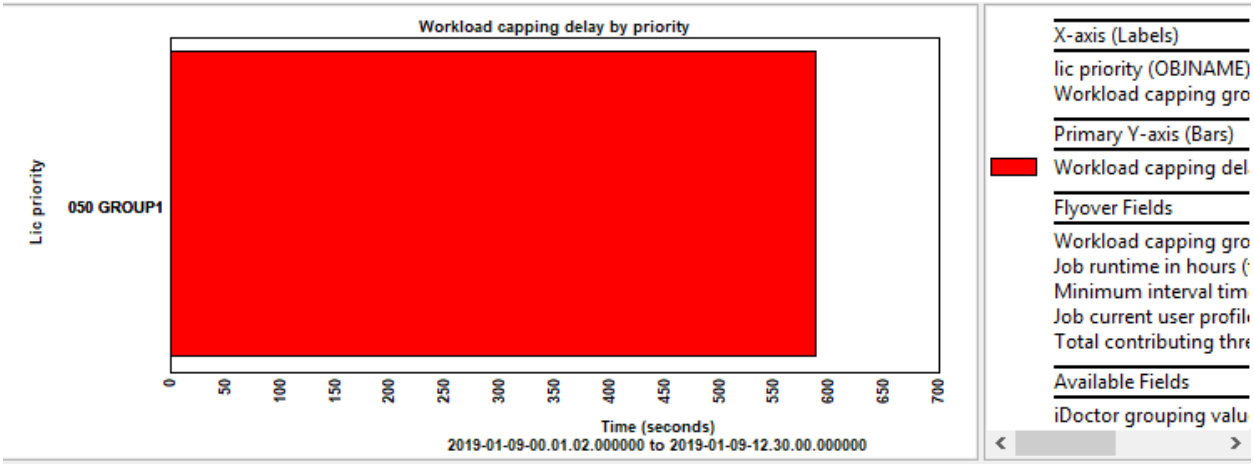
This graph ranks the selected time period's workload capping delays by workload capping group name and memory pool.



Workload capping delay by pool

9.14.15.7 Workload capping delay by priority

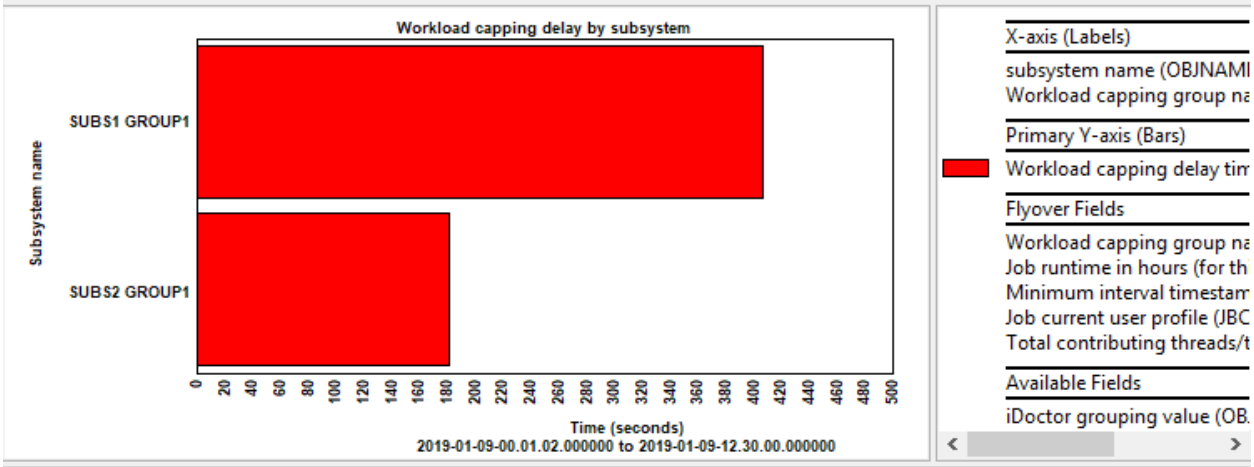
This graph ranks the selected time period's workload capping delays by workload capping group name and LIC priority.



Workload capping delay by priority

9.14.15.8 Workload capping delay by subsystem

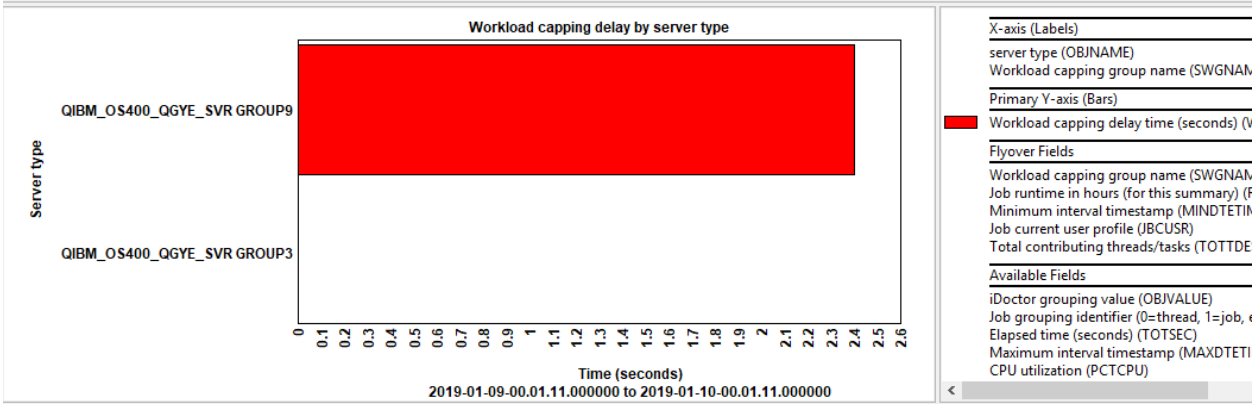
This graph ranks the selected time period's workload capping delays by workload capping group name and subsystem name.



Workload capping delay by subsystem

9.14.15.9 Workload capping delay by server type

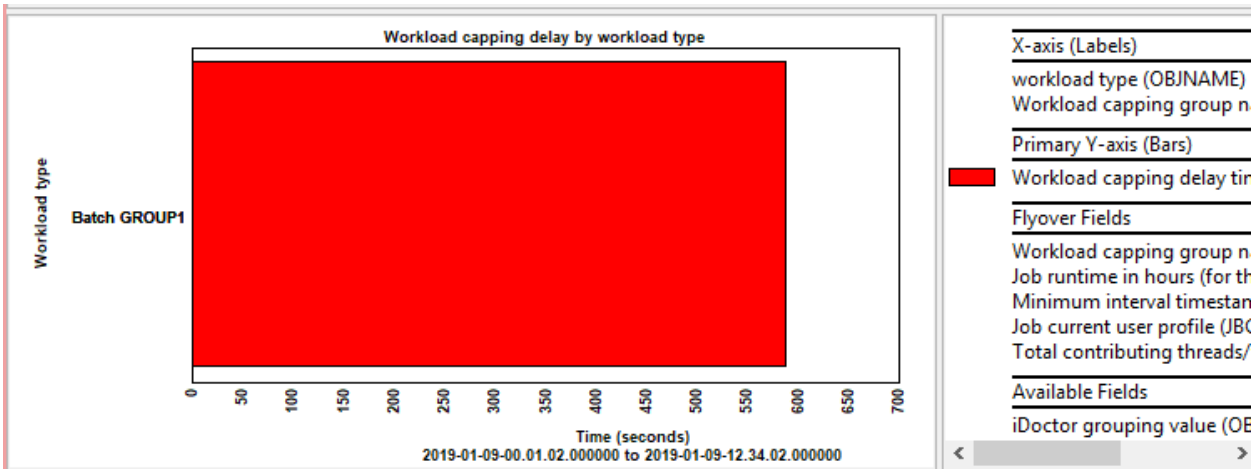
This graph ranks the selected time period's workload capping delays by workload capping group name and server type.



Workload capping delay by server type

9.14.15.10 Workload capping delay by workload type

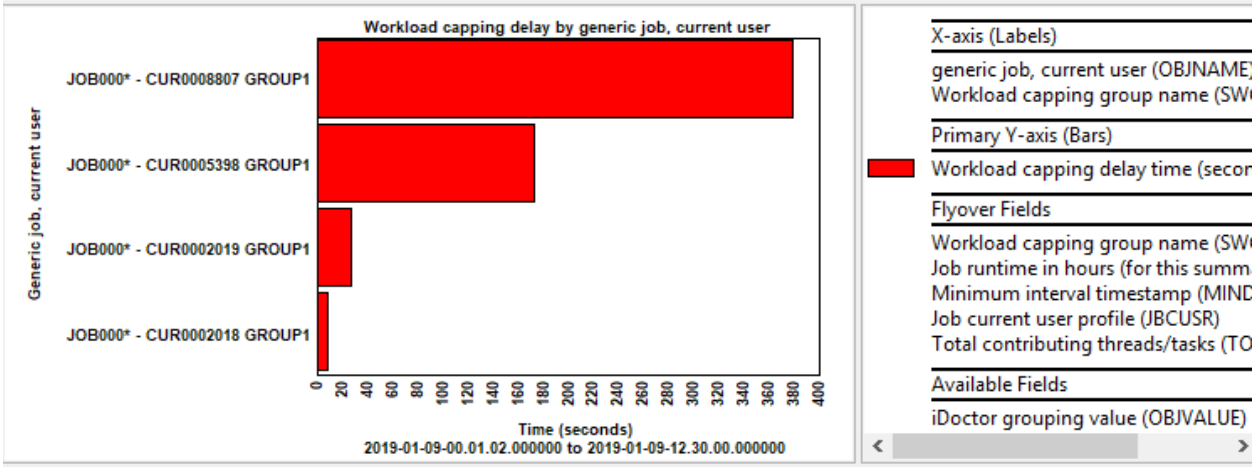
This graph ranks the selected time period's workload capping delays by workload capping group name and server type.



Workload capping delay by workload type

9.14.15.11 Workload capping delay by generic job, current user

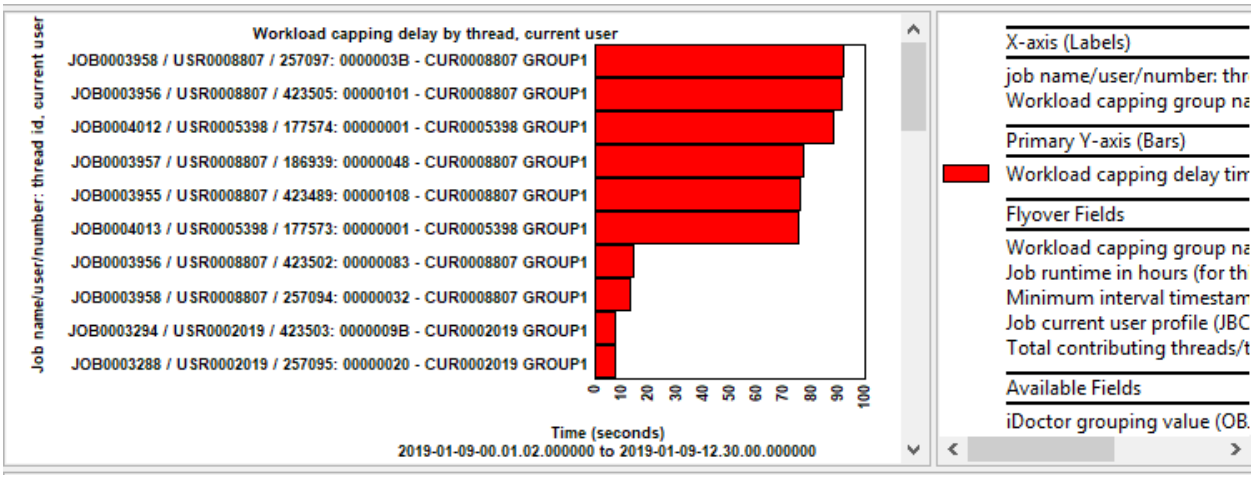
This graph ranks the selected time period's workload capping delays by workload capping group name and generic job/current user combination.



Workload capping delay by generic job, current user

9.14.15.12 Workload capping delay by thread, current user

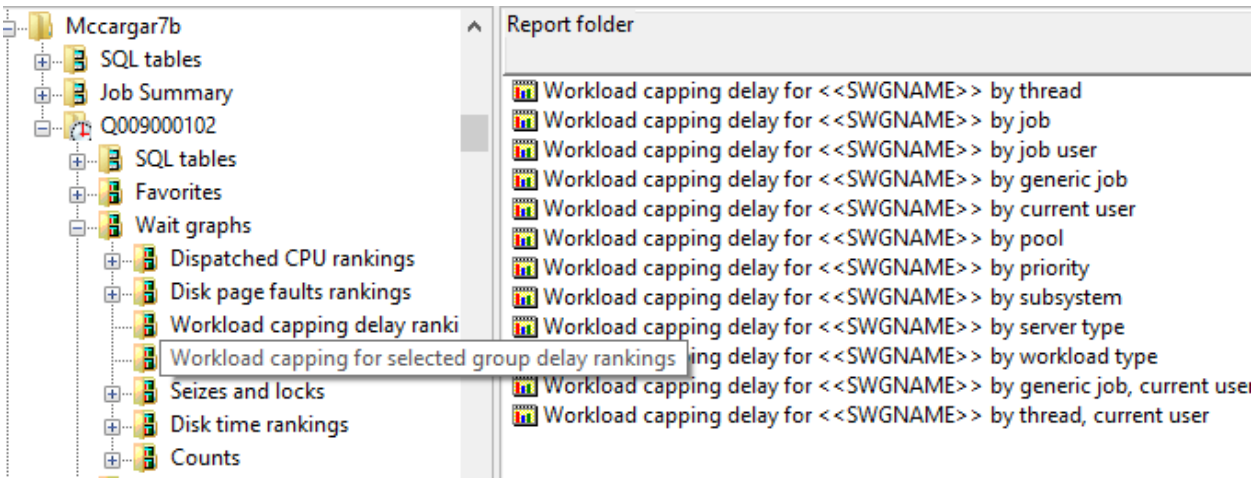
This graph ranks the selected time period's workload capping delays by workload capping group name and thread/current user combination.



Workload capping delay by thread, current user

9.14.16 Workload capping for selected group delay rankings

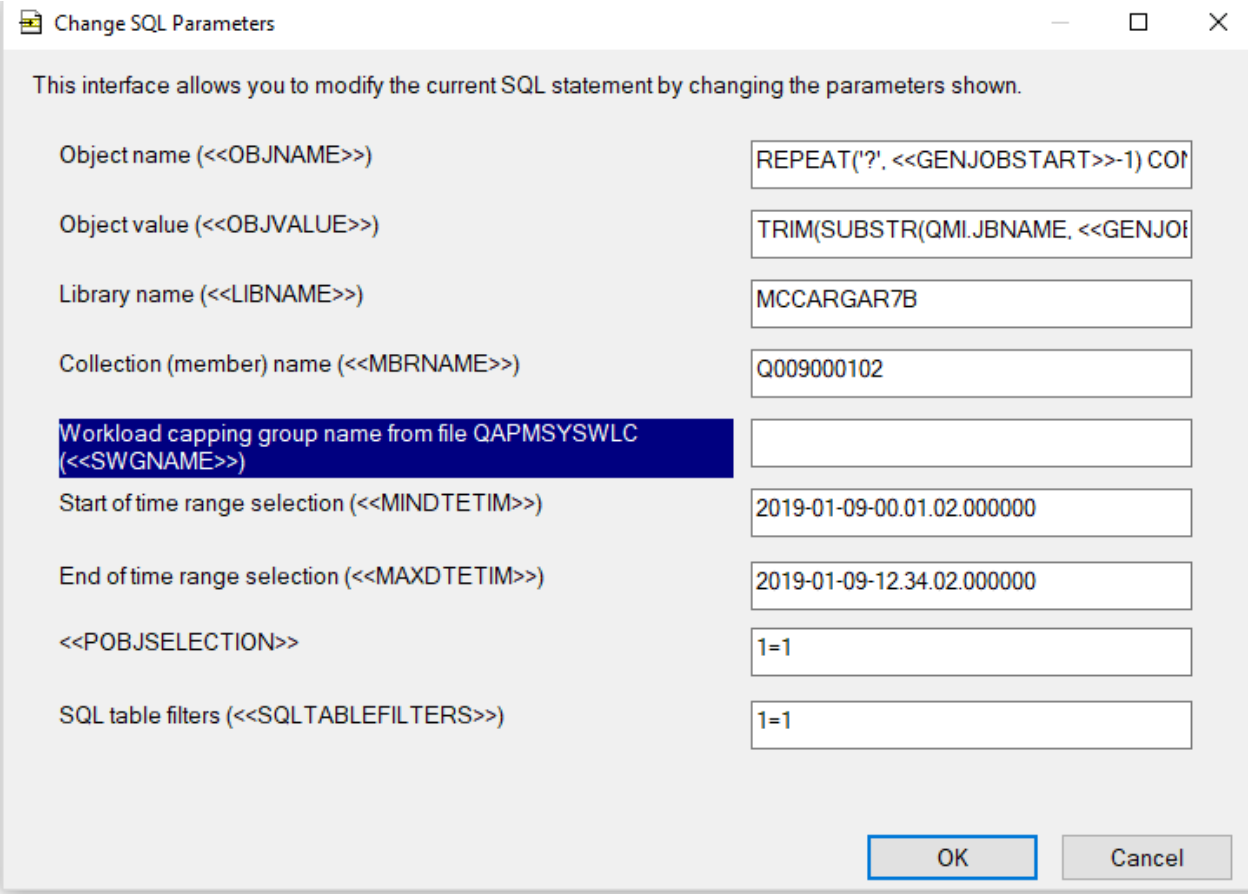
This set of graphs show the effects of workload capping for the desired workload capping group. You will be prompted to enter the desired group name value before the data for this graph is displayed in order to filter the results.



Wait graphs -> Workload capping for selected group delay rankings graphs

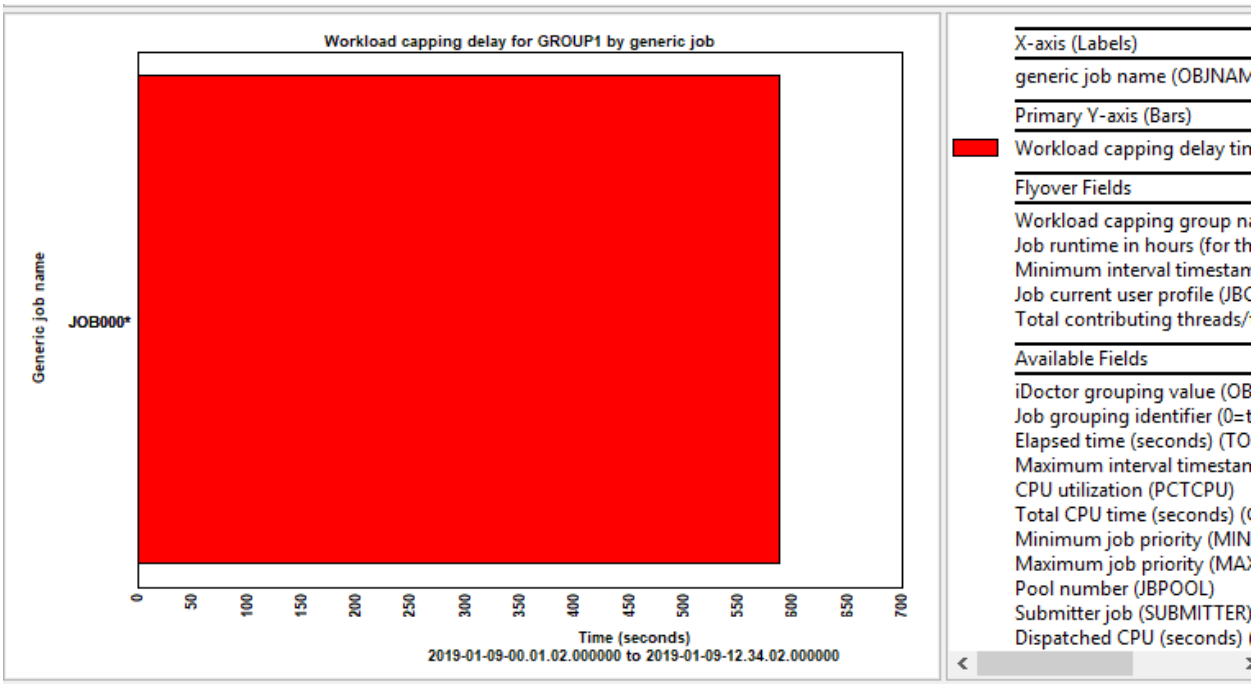
To determine the group names available either view the QAPMSYSWLC file or view any of the graphs in the Workload capping delay rankings folder.

When opening any graph in this folder the following window will be shown:



Change SQL Parameters -> Prompting for the workload capping group name

As an example, this graph shows the generic jobs having workload capping delays for ‘GROUP1’.

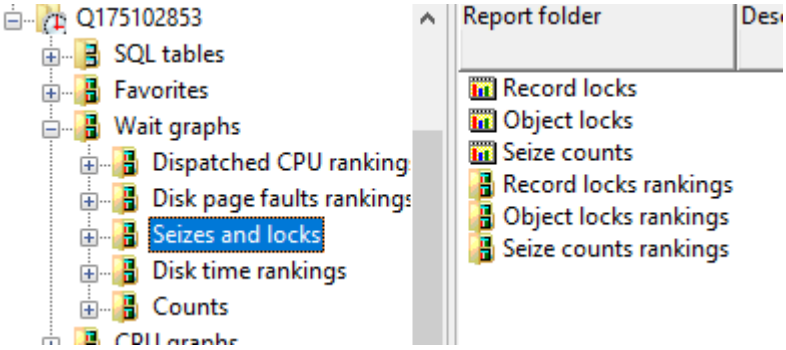


Workload capping delay for GROUP1 by generic job

9.14.17 Seizes and locks

These graphs show counts and the jobs experiencing record locks, object locks and seizes.

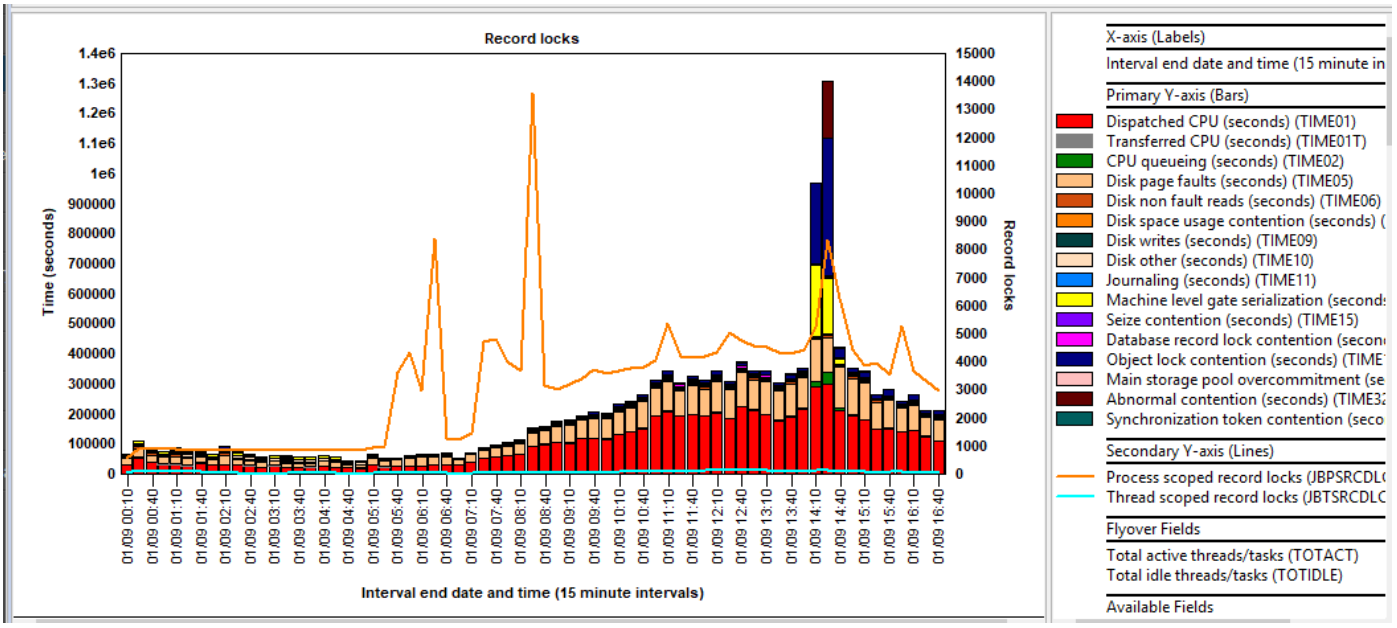
Note: This folder will only appear if the Collection Summary analysis has been ran on this collection.



Wait graphs -> Seizes and locks folder

9.14.17.1 Record locks

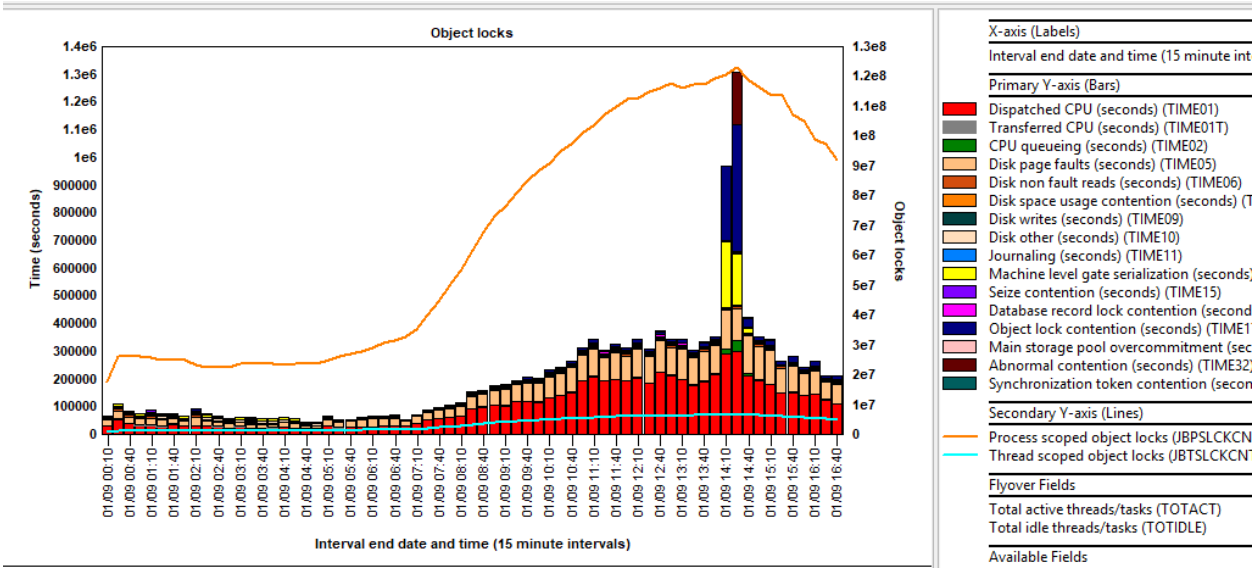
This graph is like the collection overview time signature but contains the number of process scoped record locks and thread scoped record locks on the secondary Y-axis.



Record locks

9.14.17.2 Object locks

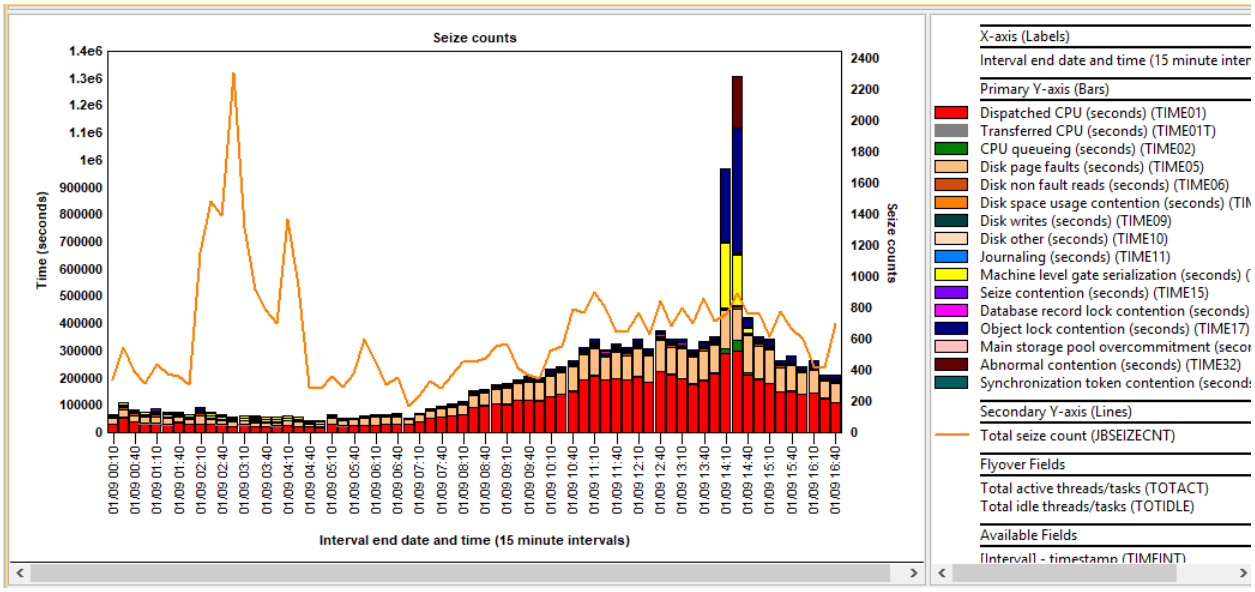
This graph is like the collection overview time signature but contains the number of process scoped object locks and thread scoped object locks on the secondary Y-axis.



Object locks

9.14.17.3 Seize counts

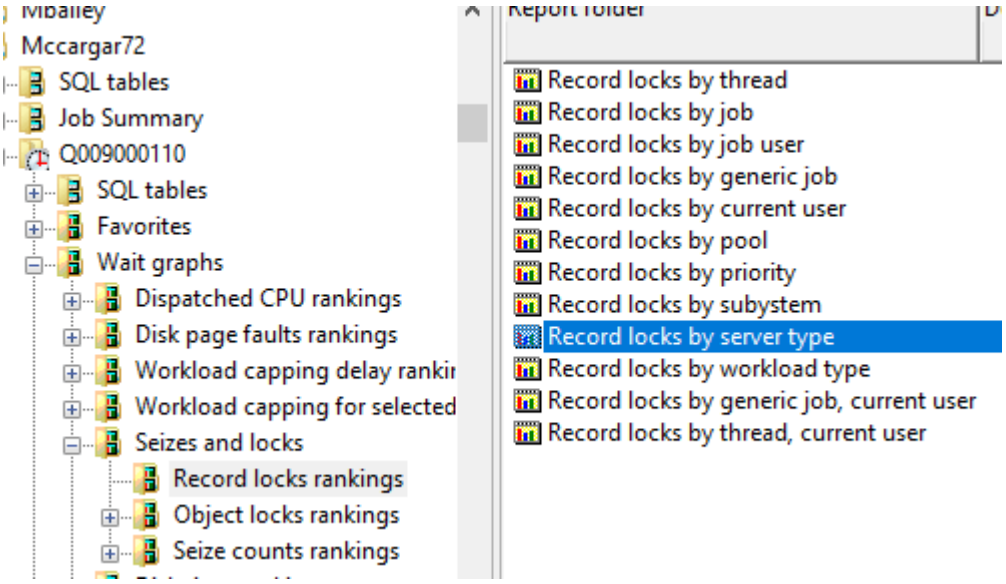
This graph is like the collection overview time signature but contains the total number of seizes on the secondary Y-axis.



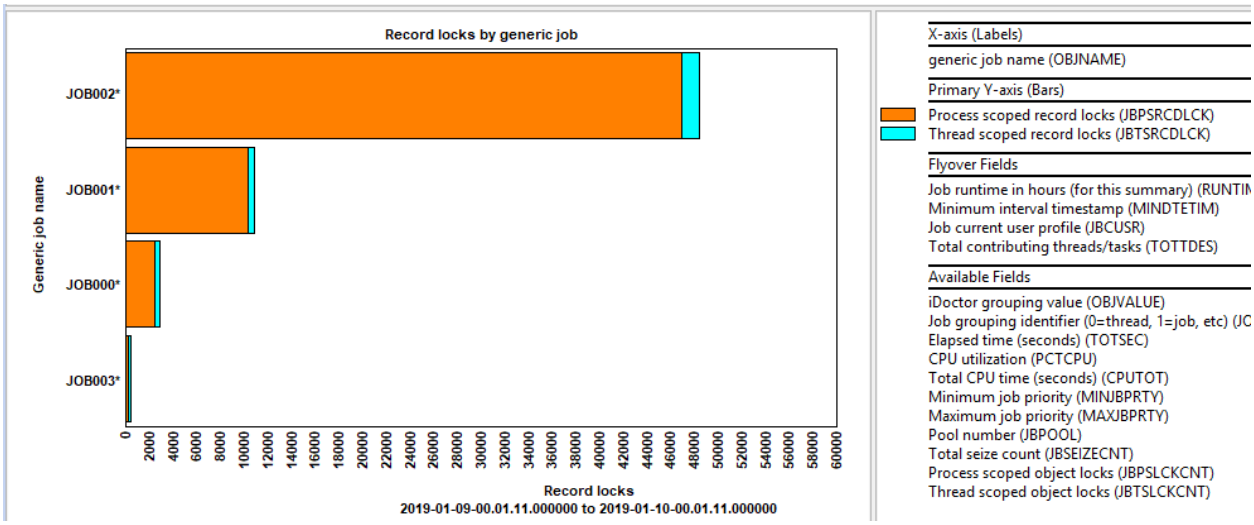
Seize counts

9.14.17.4 Record lock rankings

The record lock rankings graphs allow the user to group the record lock totals by any of the job groupings types available.



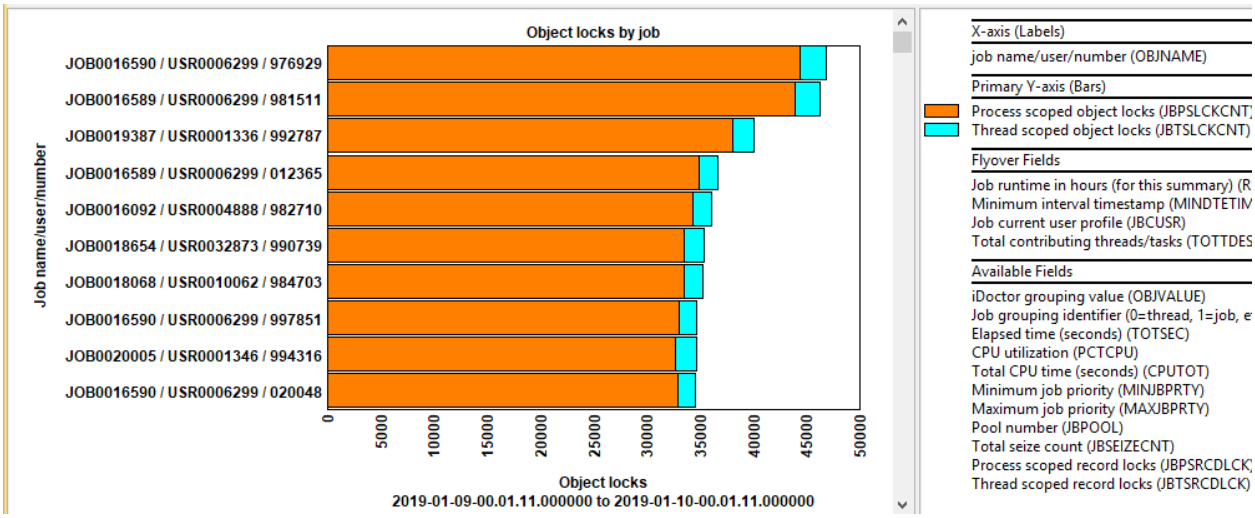
Waits graphs -> Seizes and locks -> Record locks rankings



Record locks by generic job

9.14.17.5 Object lock rankings

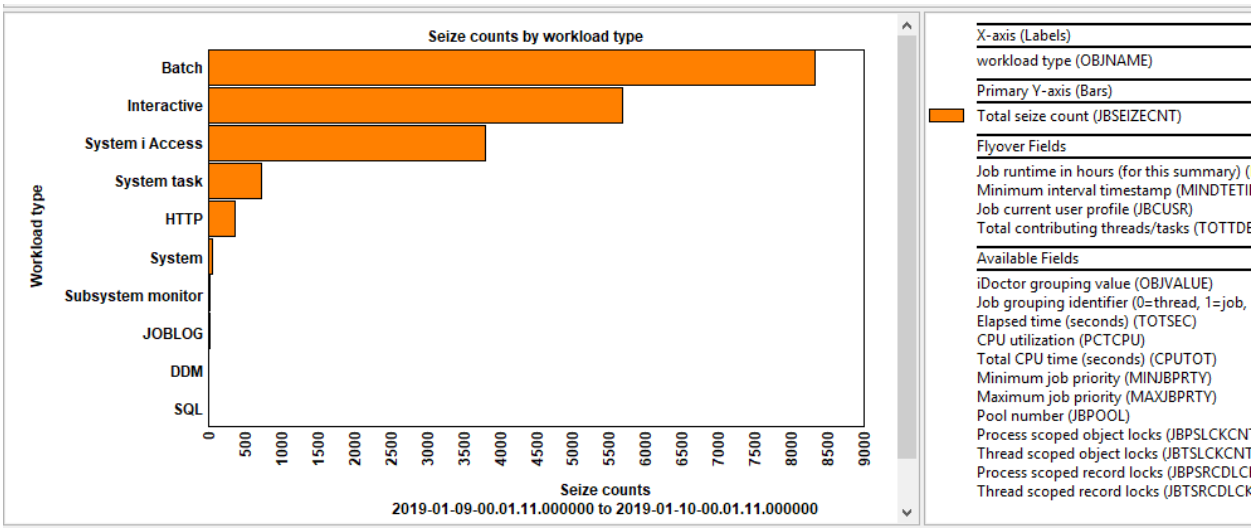
The object lock rankings graphs allow the user to group the object locks by any of the job groupings types available.



Object locks by job

9.14.17.6 Seize counts rankings

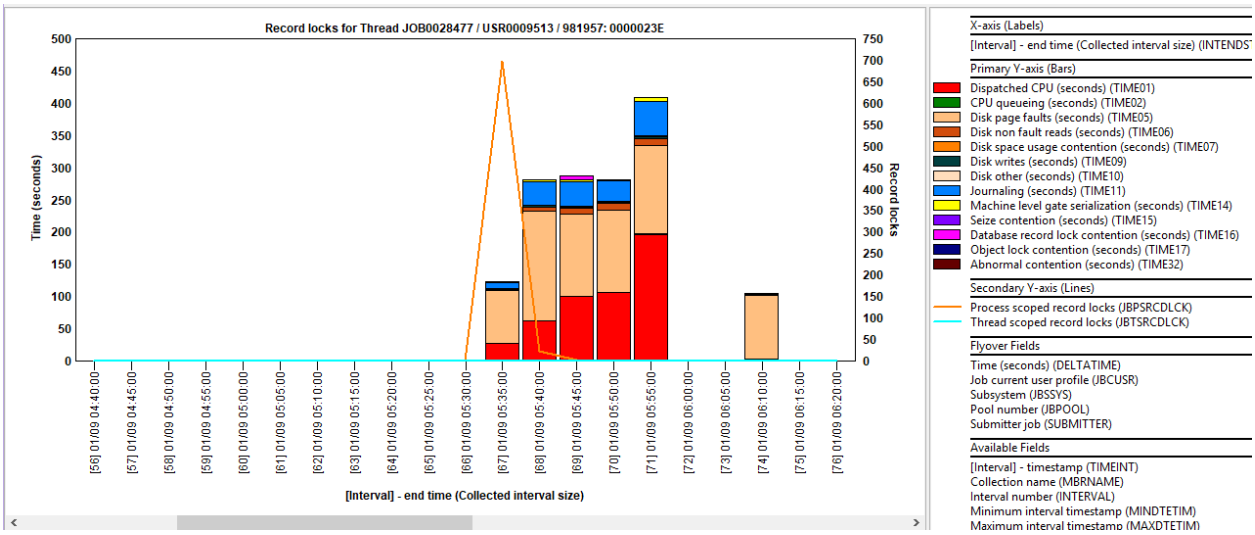
The seize counts rankings graphs allow the user to group the seize counts by any of the job groupings types available.



Seize counts by workload type

9.14.17.7 Selected job grouping over time graphs

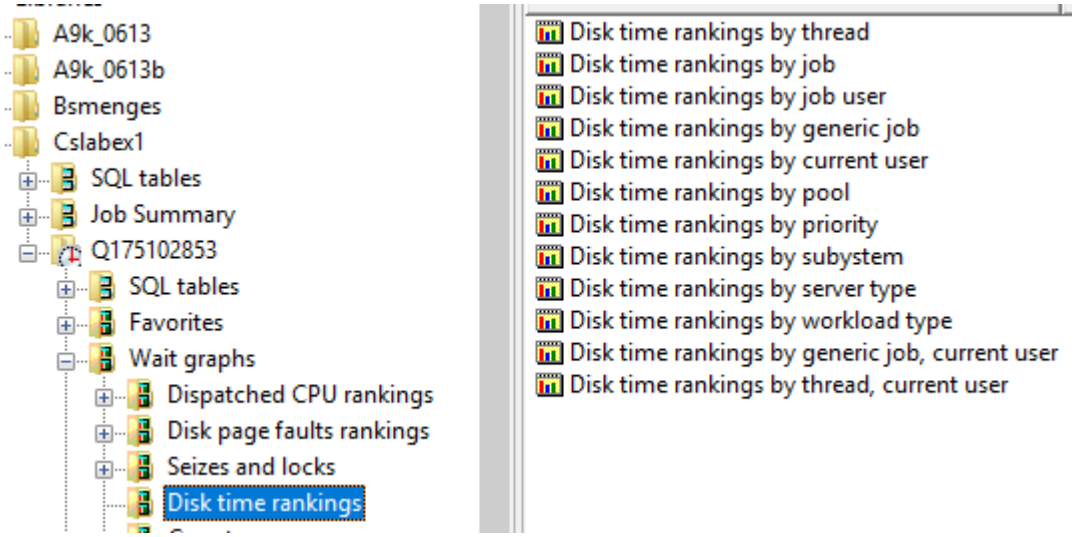
Any of the 3 types of rankings charts above can be drilled down from by right-clicking a job / current user / etc and picking the 1st option. The result graph will contain the wait buckets times for the desired item and the seizes or lock counts on the secondary Y-axis. It will look something like this:



Record locks for Thread JOB0028477

9.14.18 Disk time rankings

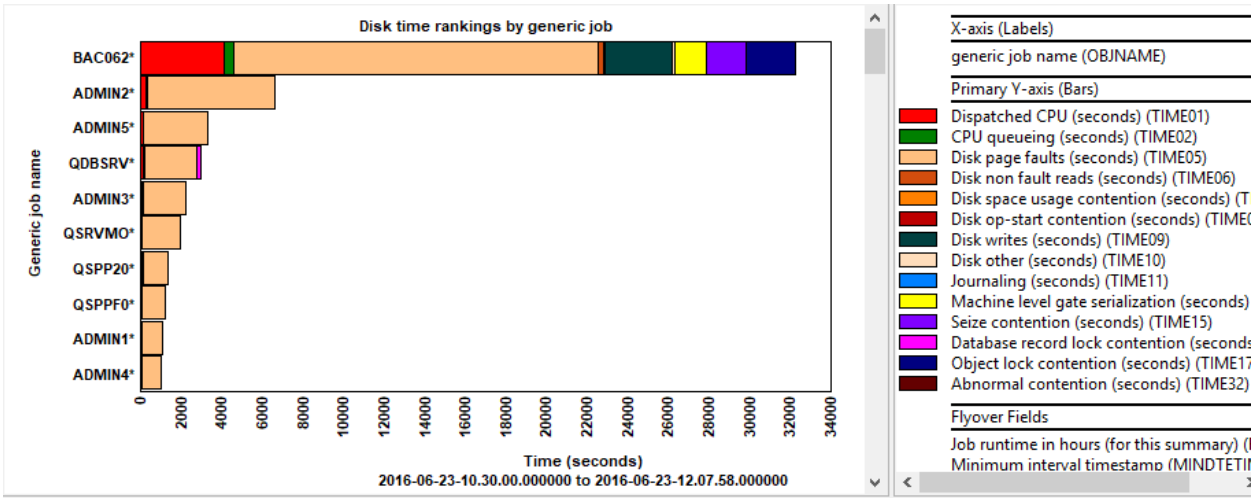
Under the Wait graphs folder is the Disk time rankings subfolder which contains a set of wait bucket ranking graphs sorted by total disk times (all disk wait buckets 5 through 10 added together) over the entire collection. One graph is listed for each type of job grouping available.



Wait graphs -> Disk time rankings

The graph only shows data for jobs that had at least some disk time during the collection.

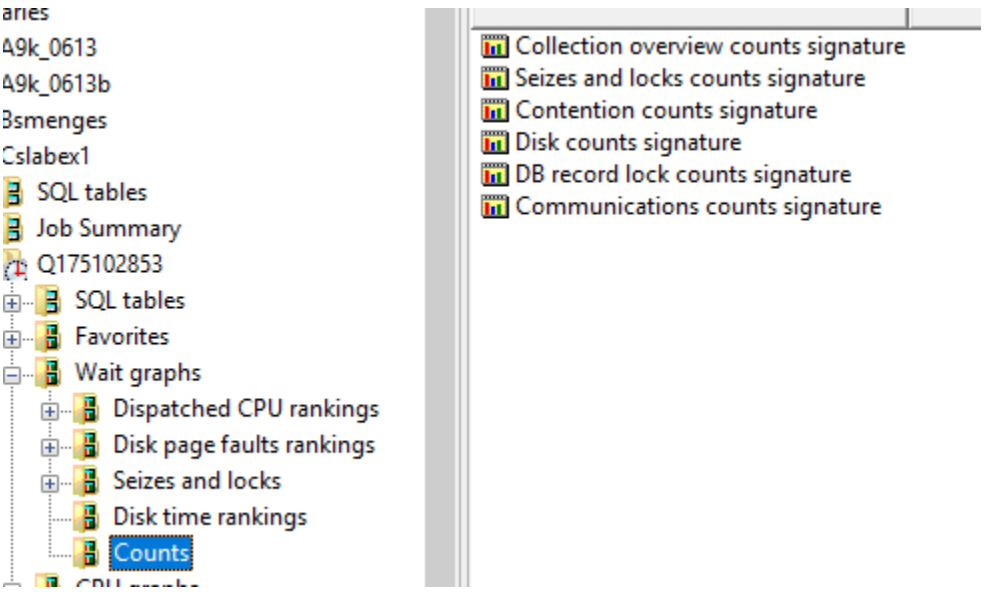
These graphs are the same as the Dispatched CPU rankings graphs with the only difference being these graphs rank disk times (all disk wait buckets 5 through 10 added together) instead of Dispatched CPU time.



Disk time rankings by generic job

9.14.19 Counts

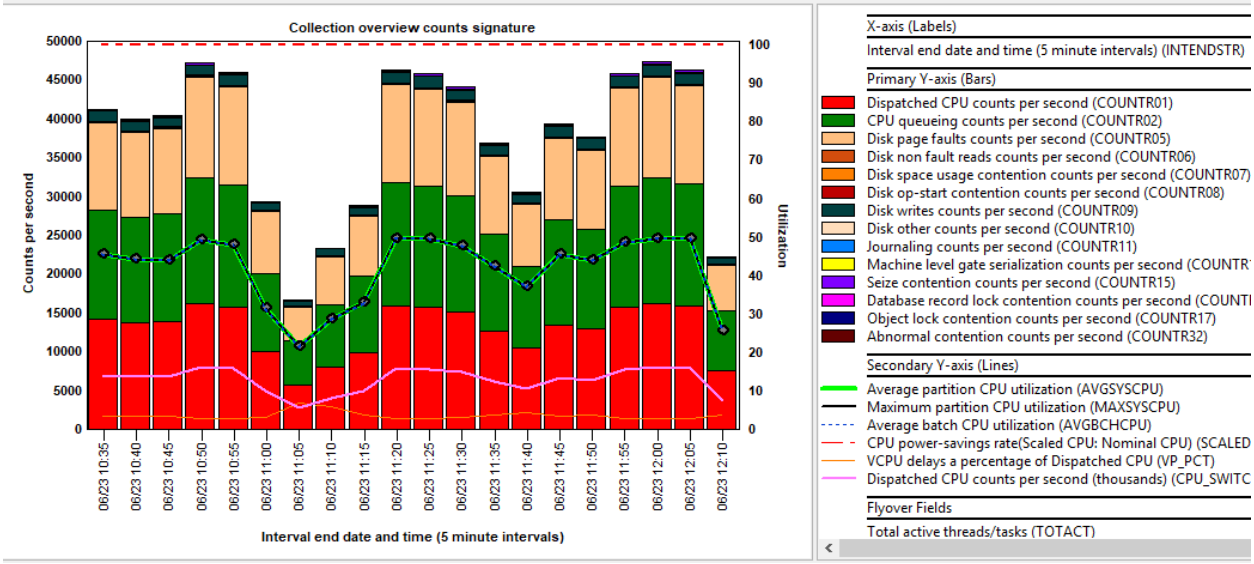
This series of graphs shows the number of individual waits that occurred per seconds within each wait bucket over time. This metric is referred to as Counts per second (i.e. Disk write counts per second).



Wait graphs -> Counts folder

9.14.19.1 Collection overview counts signature

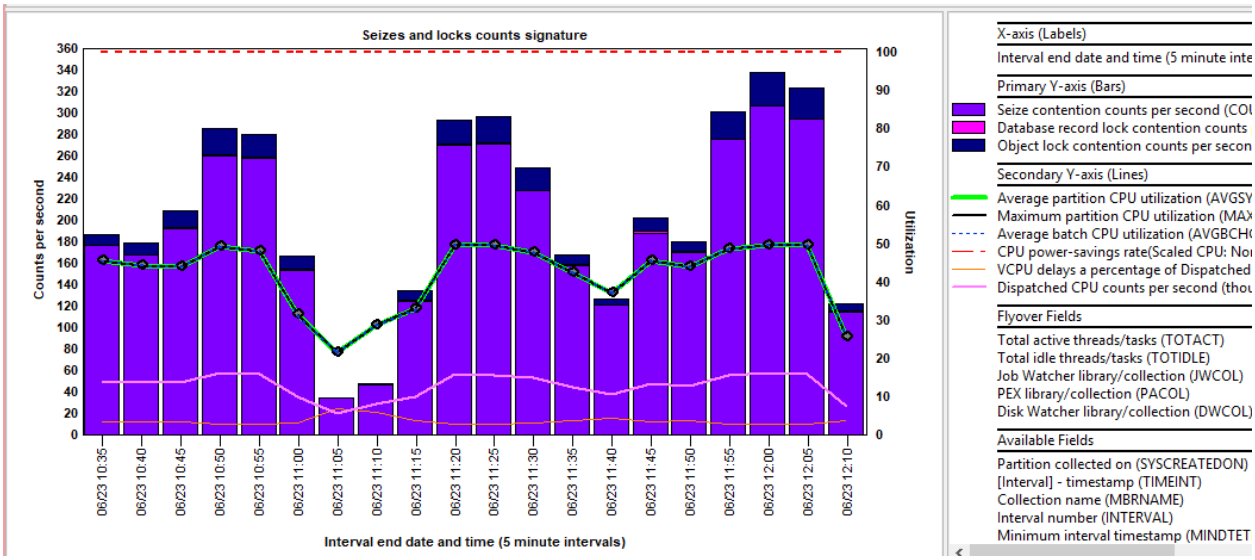
This graph shows CPU and the “most interesting” wait bucket counts per second added together across all jobs on the system.



Collection overview counts signature

9.14.19.2 Seizes and locks counts signature

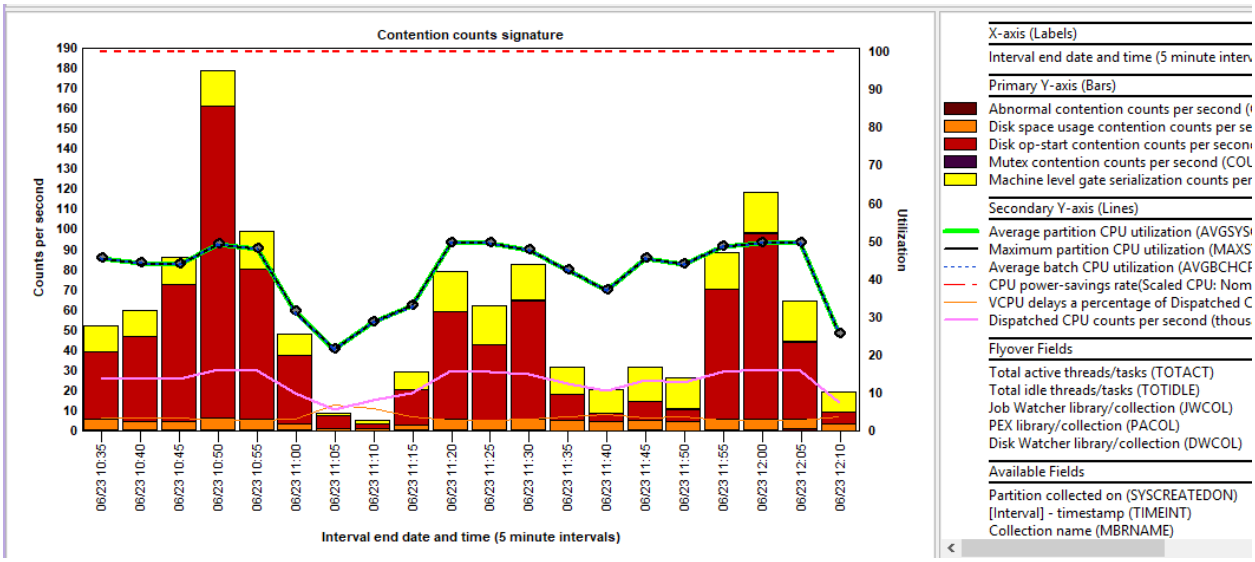
This graph just shows seizes, record locks and object lock counts per second.



Seizes and locks counts signature

9.14.19.3 Contention counts signature

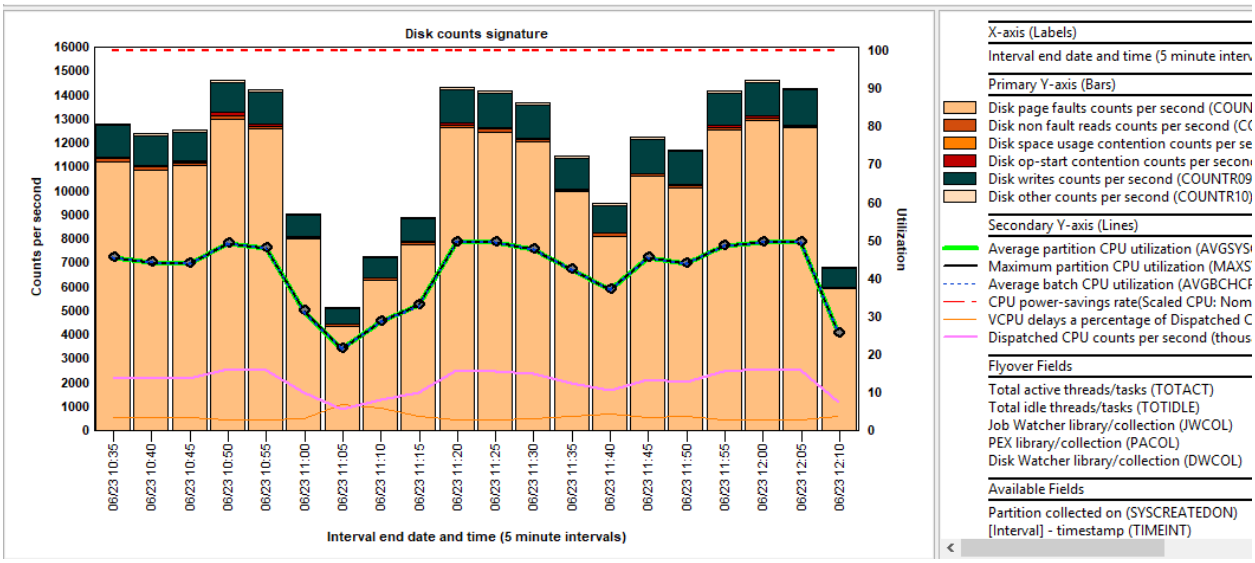
This graph only shows wait buckets that are usually (but not always) associated with some type of contention on the system. Occasionally there are system tasks that use enums in these buckets as their normal “idle” wait when they shouldn’t.



Contention counts signature

9.14.19.4 Disk counts signature

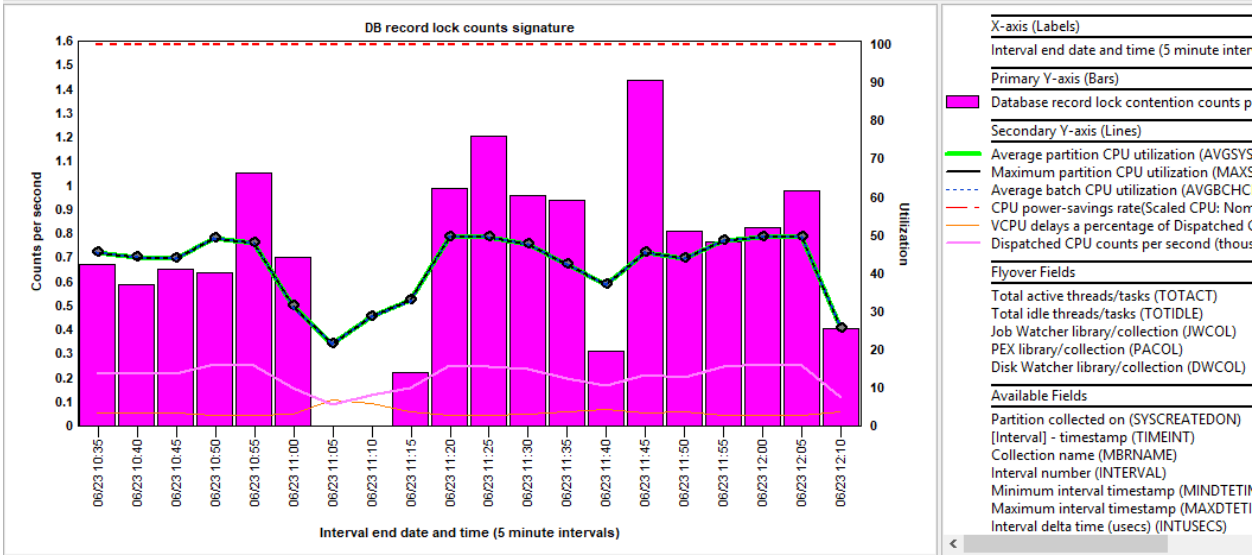
This graph just shows wait buckets related to disk I/O or disk contention.



Disk counts signature

9.14.19.5 DB record lock counts signature

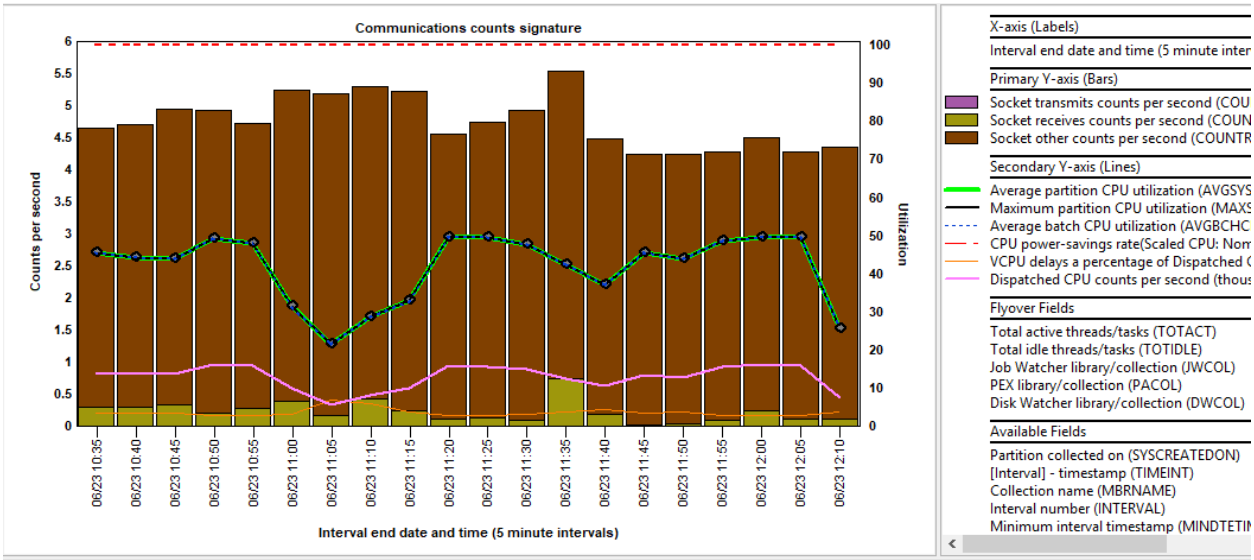
This graph only shows DB record lock counts per second.



DB record lock counts signature

9.14.19.6 Communications counts signature

These waits indicate time waiting to receive or send data or other types of socket waits. An example of a socket receive is what a QZRCSRVS job (iDoctor remote command job servicing the GUI) will do when it is idle waiting for requests from the PC. Once the job receives data over the comm line other types of ***non-idle*** waits (CPU, disk IO) will be shown.

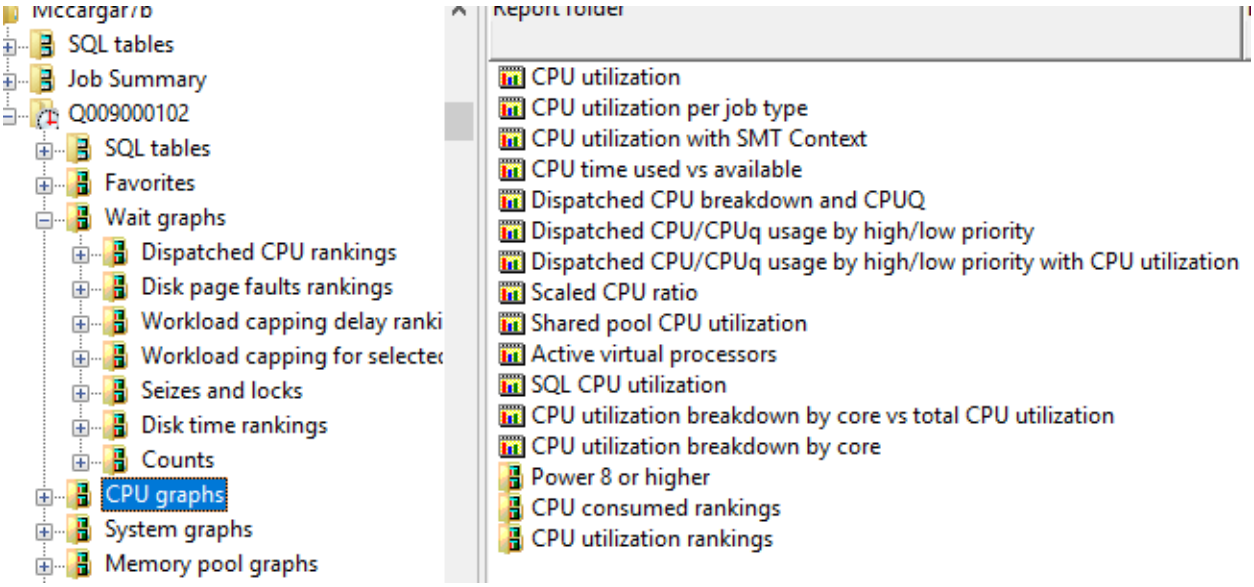


Communications counts signature

9.15 CPU graphs

These graphs display the various types of CPU related metrics available in Collection Services.

Note: Many graphs in this folder will classify low priority jobs as jobs at priority 30 to 99. High priority jobs are classified as having a priority of 0 to 29.

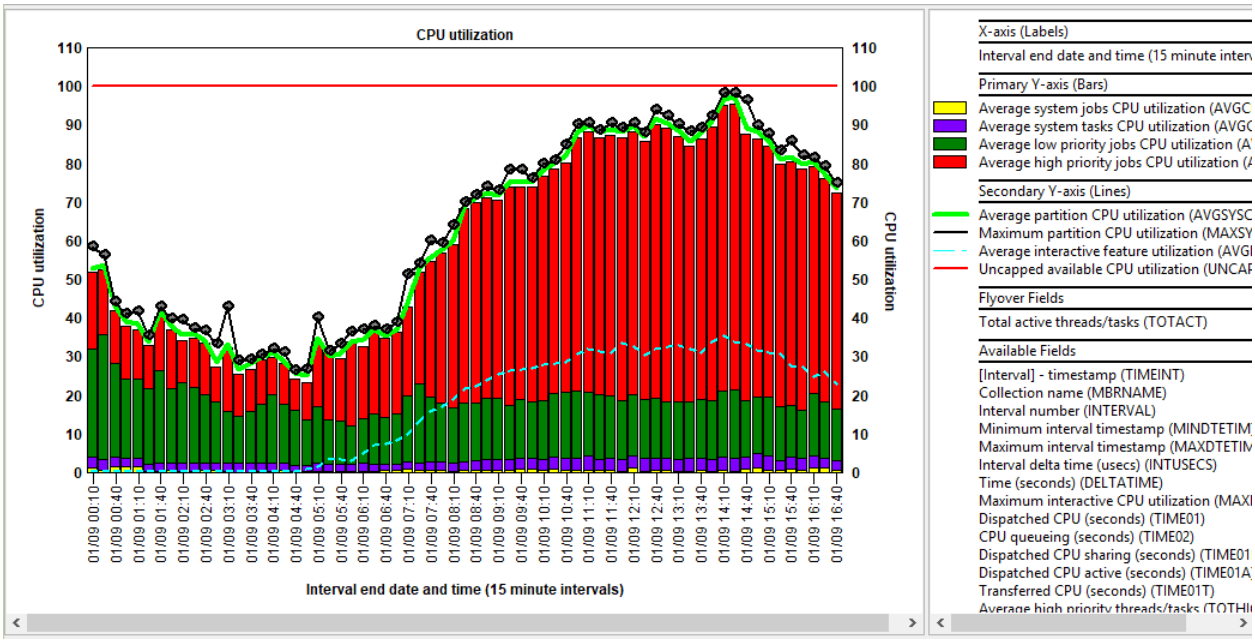


CPU graphs folder

9.15.1 CPU utilization

This graph breaks down the types of jobs and tasks that are using the CPU into 4 different categories:

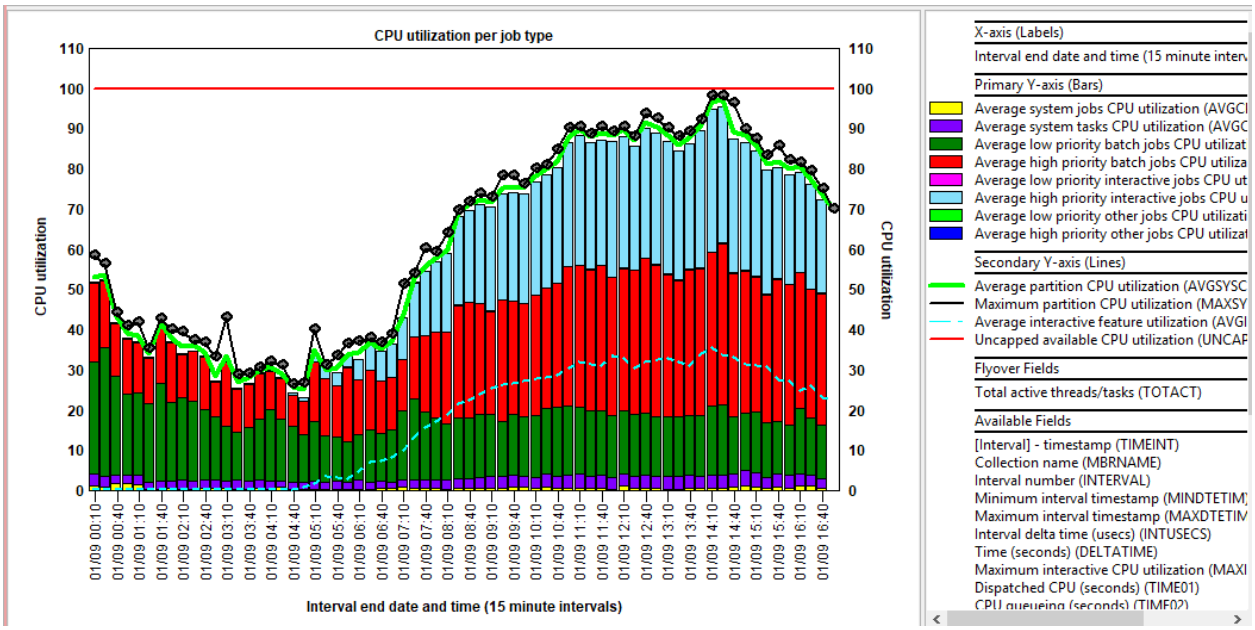
1. System jobs
2. System tasks
3. Low priority jobs (excludes the system/IBM i jobs)
4. High priority jobs (excludes the system/IBM i jobs)



CPU utilization

9.15.2 CPU utilization per job type

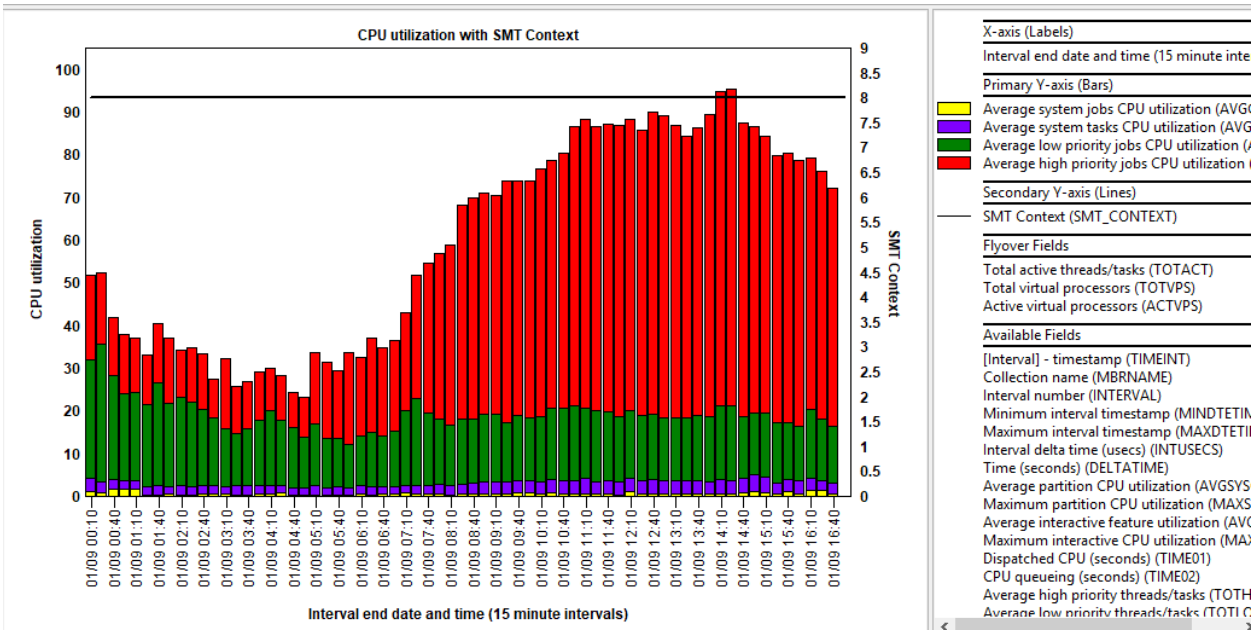
This graph is like the previous graph except it breaks down the categories of jobs into 8 types for more granularity.



CPU utilization per job type

9.15.3 CPU utilization with SMT context

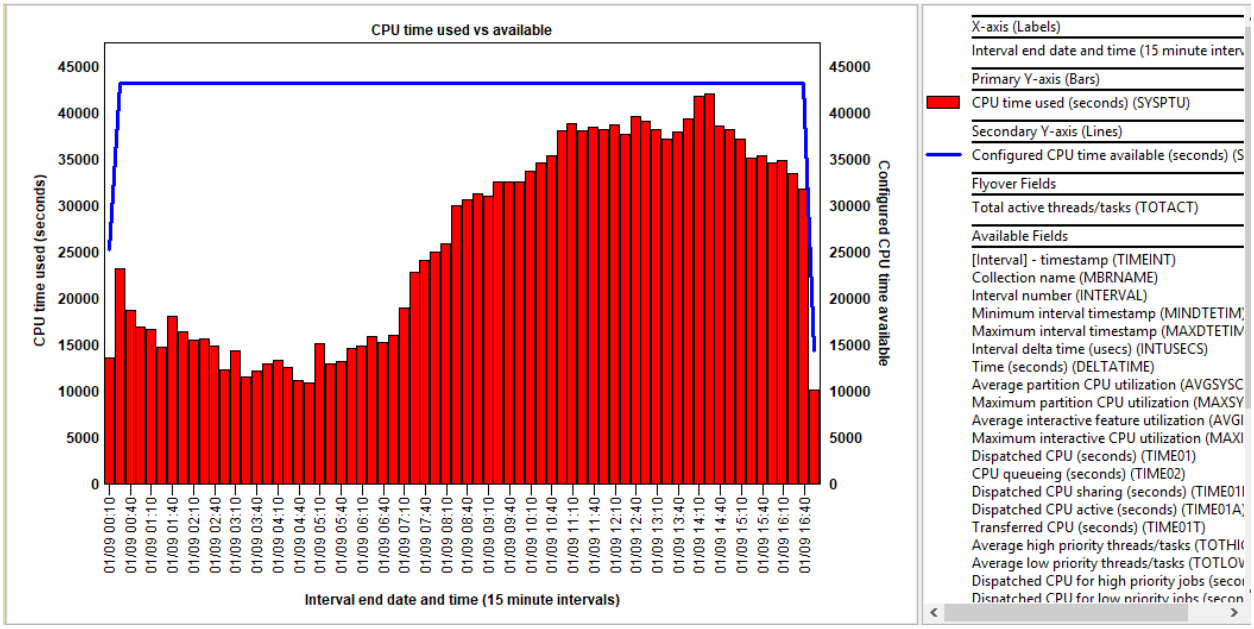
This graph is the same as the CPU utilization graph except the secondary Y-axis displays the SMT context. This refers to the number of hardware threads used in relation to simultaneous multithreading.



CPU utilization with SMT contex

9.15.4 CPU time used vs available

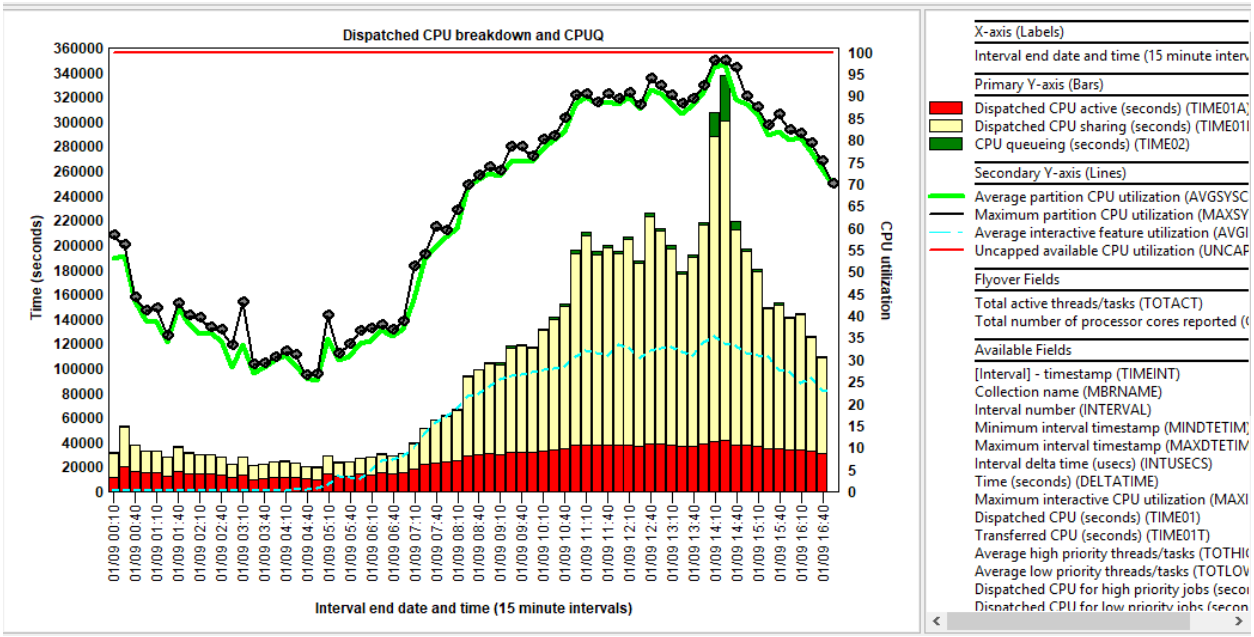
This graph displays the amount of CPU time used vs the configured CPU time available.



CPU time used vs available

9.15.5 Dispatched CPU breakdown and CPUQ

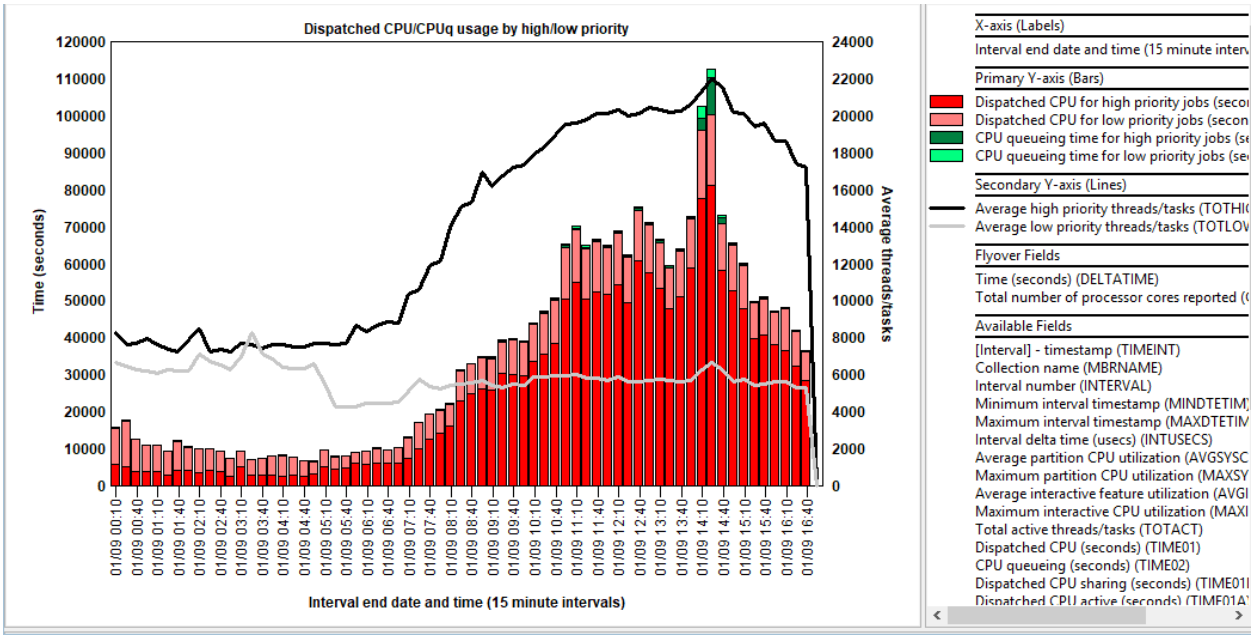
This graph displays a breakdown of Dispatched CPU showing both the active and sharing components of Dispatched CPU time along with CPU queuing. The typical CPU utilization fields are shown on the secondary Y-axis.



Dispatched CPU breakdown and CPUQ

9.15.6 Dispatched CPU/CPUQ usage by high/low priority

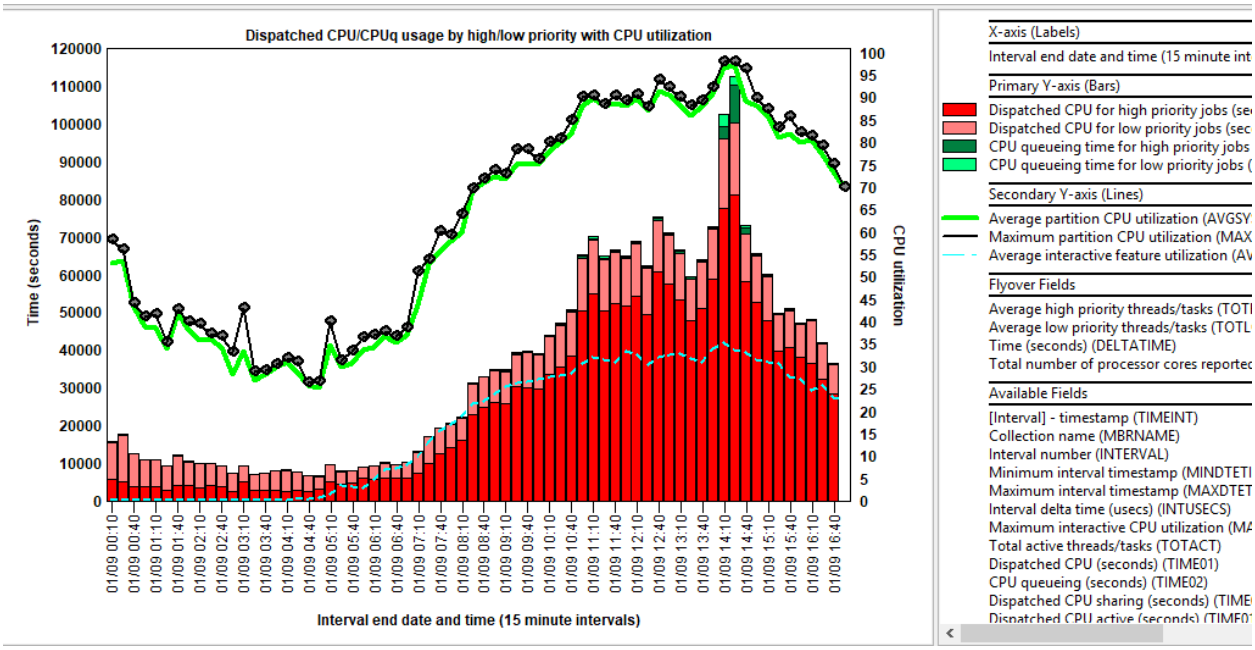
This graph shows the Dispatched CPU and CPU queuing for both high and low priority jobs. The secondary Y-axis displays a count of the number of high and low priority threads or tasks.



Dispatched CPU/CPUQ usage by high/low priority

9.15.7 Dispatched CPU/CPUQ usage by high/low priority with CPU utilization

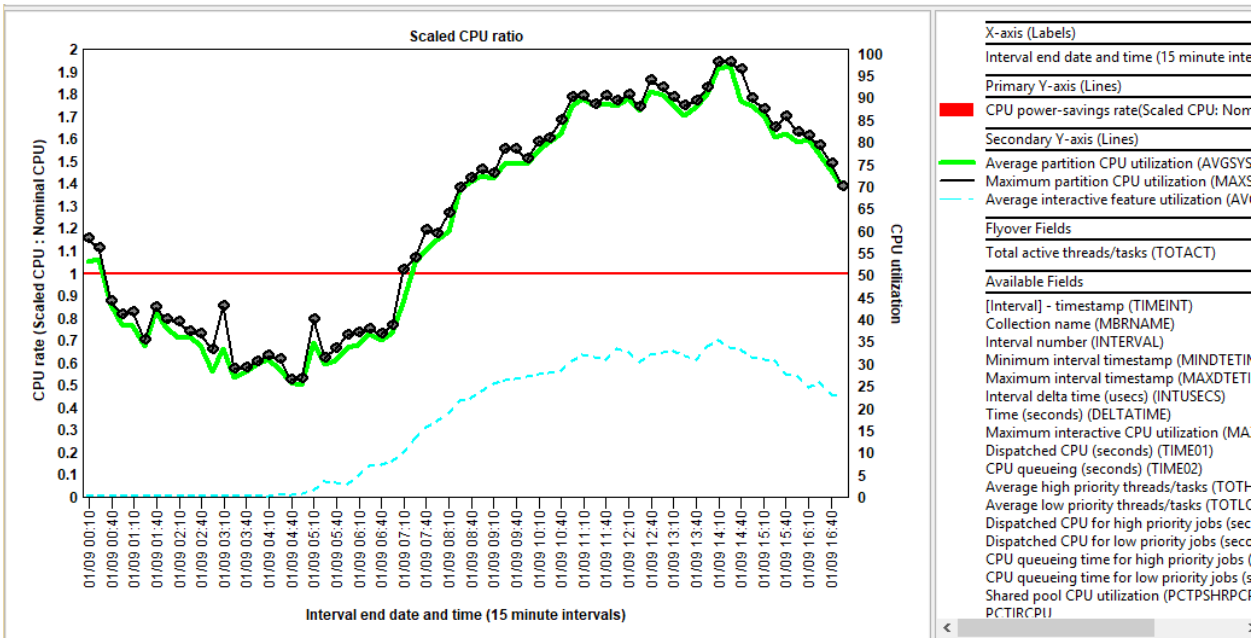
This graph is the same as the previous graph except the 2nd Y-axis contains the CPU utilization fields.



Dispatched CPU/CPUQ usage by high/low priority with CPU utilization

9.15.8 Scaled CPU ratio

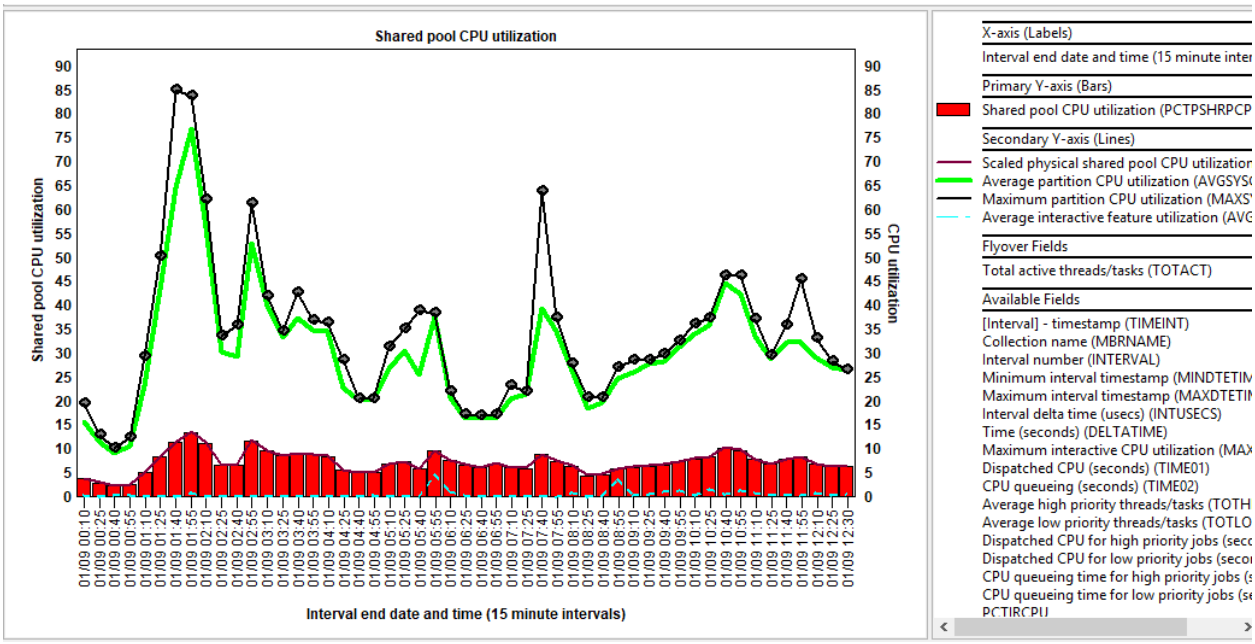
This graph shows the effects on the CPU because of energy saving features. If the value is 1 then there is no impact on the CPU.



Scaled CPU ratio

9.15.9 Shared pool CPU utilization

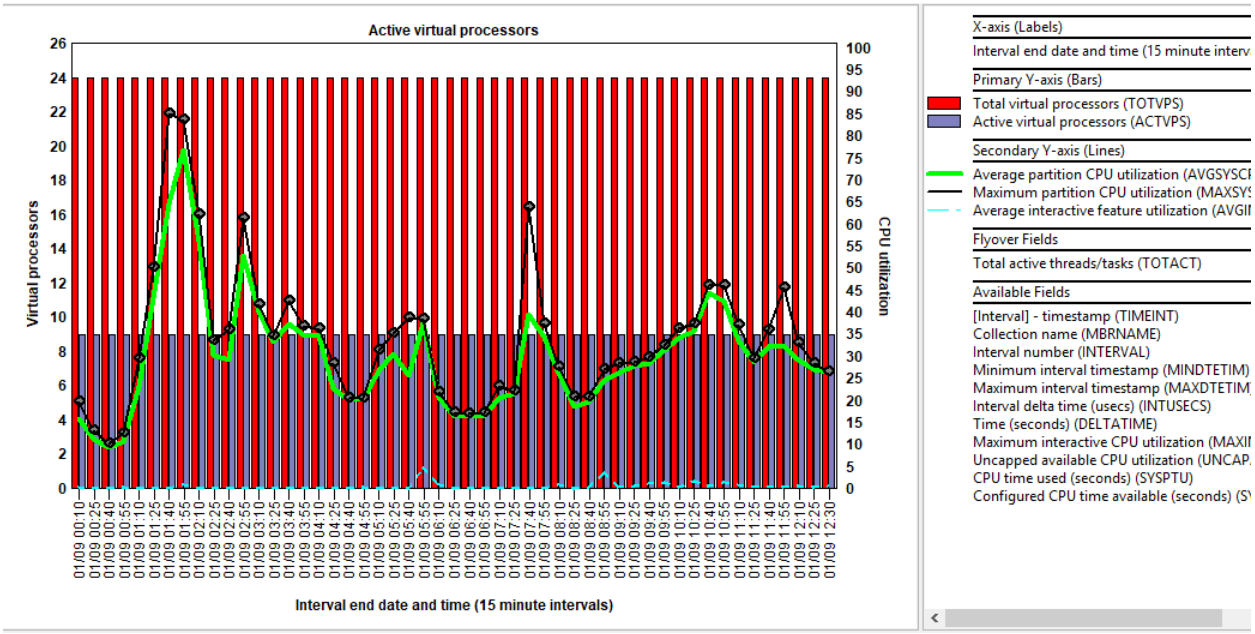
This graph shows the amount of shared pool CPU utilization (Primary Y-axis) in relation to the normal CPU utilization fields (Secondary Y-axis.)



Shared pool CPU utilization

9.15.10 Active virtual processors

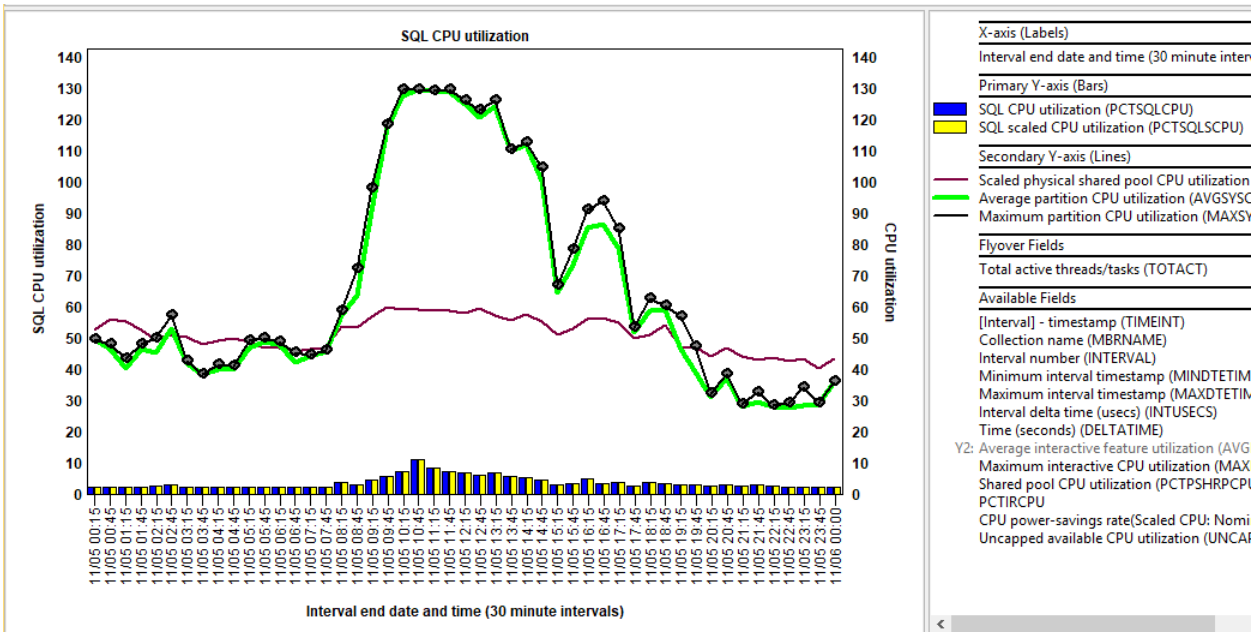
This graph shows the total vs active virtual processors over time along with the CPU utilization fields on the secondary Y-axis.



Active virtual processors

9.15.11 SQL CPU utilization

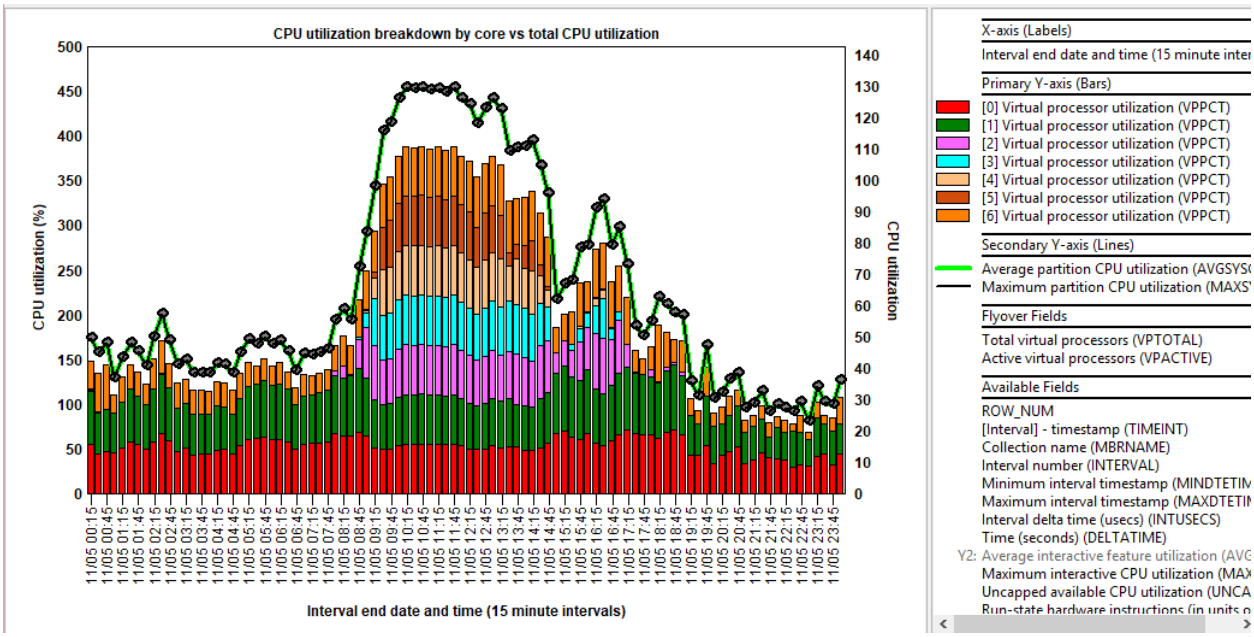
This graph shows the amount of CPU utilization dedicated to SQL processing with or without scaled CPU.



SQL CPU utilization

9.15.12 CPU utilization breakdown by core vs total CPU utilization

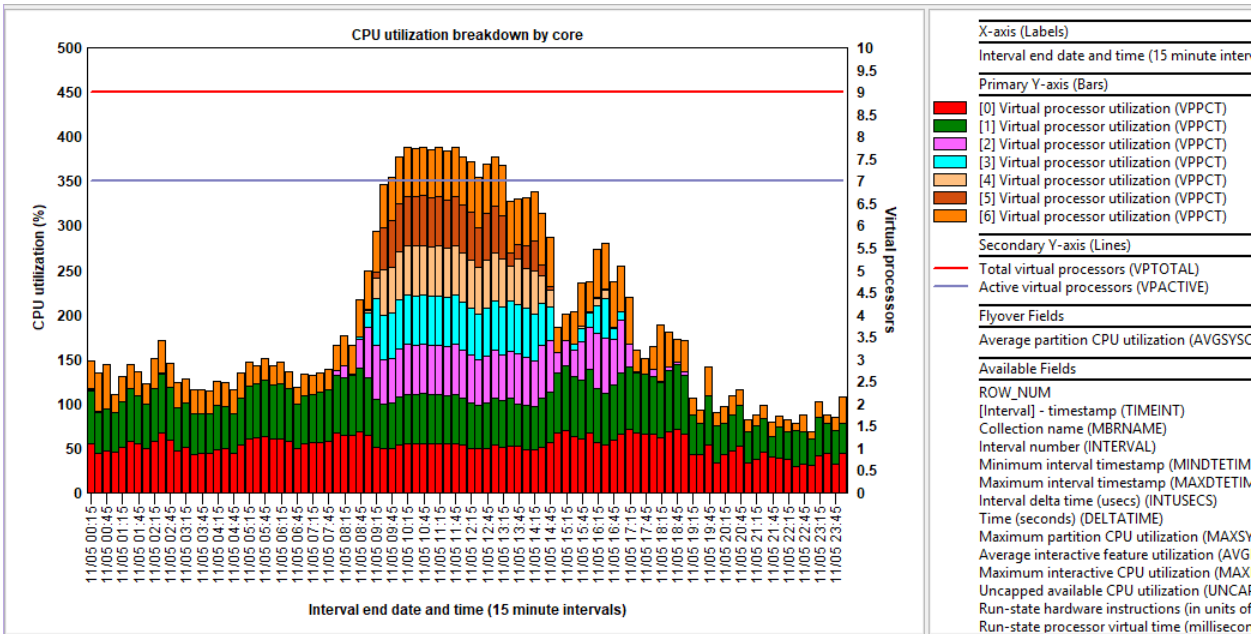
This graph displays CPU utilization but broken down for each virtual processor.



CPU utilization breakdown by core vs total CPU utilization

9.15.13 CPU utilization breakdown by core

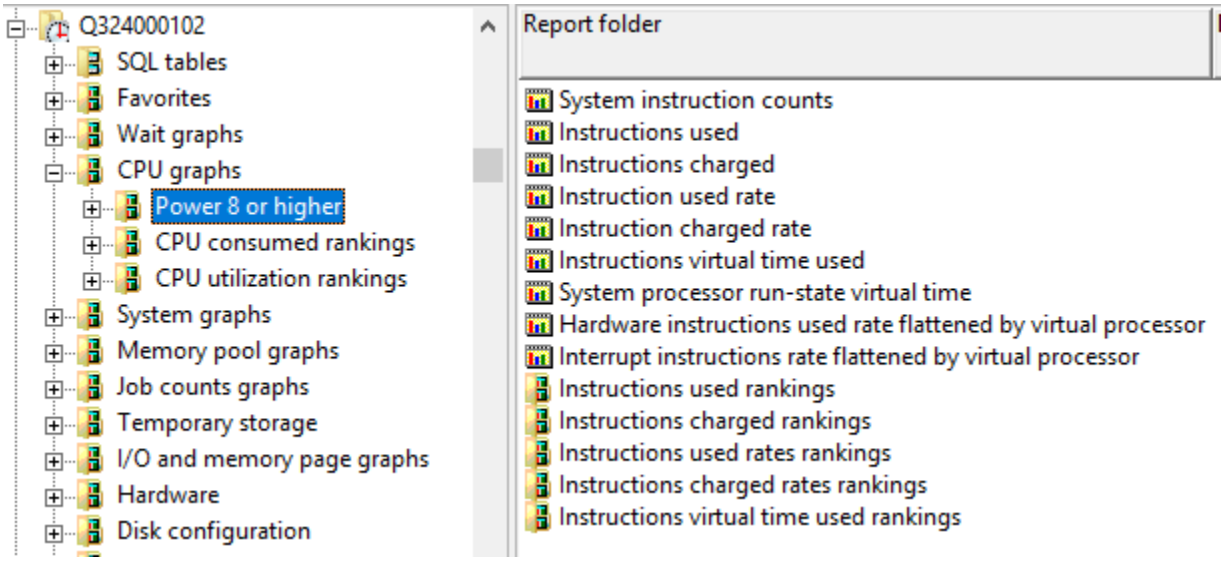
This graph is the same as previous graph except the secondary Y-axis displays the total active and available virtual processors instead of the overall CPU utilization.



CPU utilization breakdown by core

9.15.14 Power 8 or higher

This folder contains a set of graphs relating to systems with Power 8 or higher hardware only. Systems that do NOT have Power 8 hardware will not show any data within these graphs.



CPU graphs -> Power 8 or higher

9.15.14.1 System instruction counts

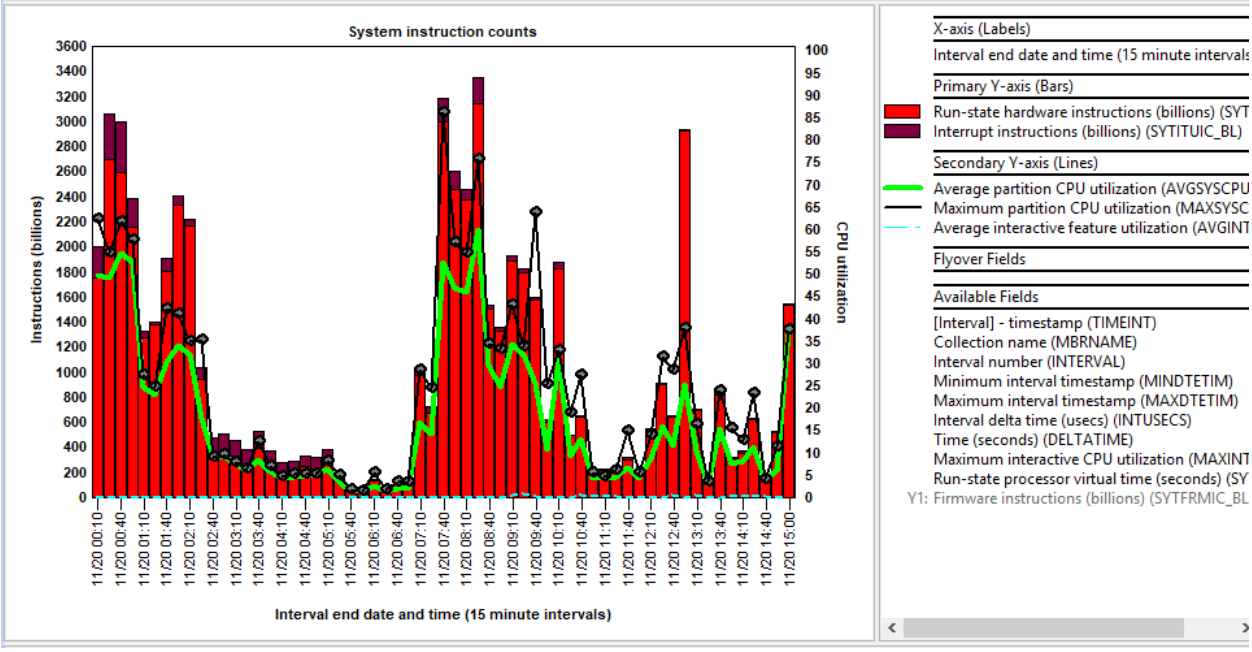
This graph displays instruction counts exerpienced on the system during the collection. It includes the folowing types of data:

Run-state hardware instructions (billions) - Hardware instructions, completed by and summed for processor threads in run state.

Interrupt instructions (billions) - Hardware instructions, completed by and summed for processor threads in interrupt state.

Firmware instructions (billions) - Hardware instructions, completed by firmware

Note: The values shown on this graph are available only for partitions running on Power8® and later hardware. However, it is not available when running in Power7® or Power6® mode.



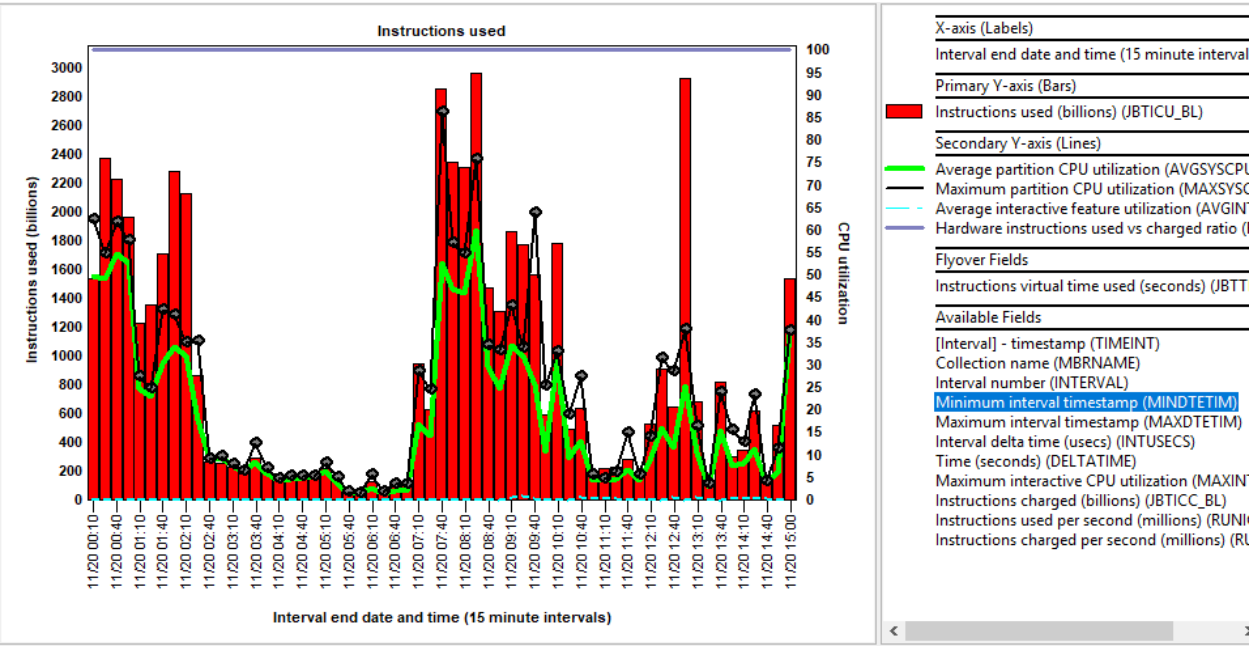
System instruction counts

9.15.14.2 Instructions used

This graph is like the previous graph but shows the total instructions used rather than breaking them out into 3 categories as shown above. The secondary Y-axis on this graph also includes a value for Hardware instructions used vs charged ratio.

Used instructions are based on the total thread instruction count used without regard to how server tasks work is charged during each interval.

Note: The values shown on this graph are available only for partitions running on Power8® and later hardware. However, it is not available when running in Power7® or Power6® mode.



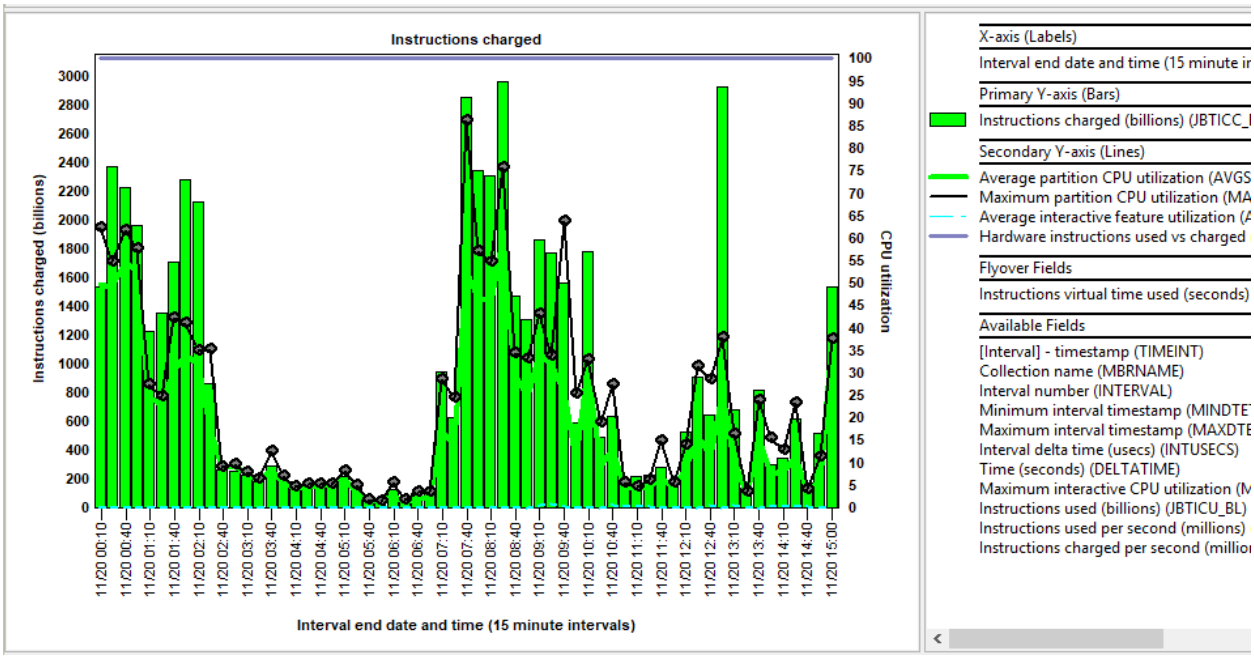
Instructions used

9.15.14.3 Instructions charged

This graph shows the instructions charged and will typically look like the previous graph (Instructions used), assuming the Hardware instructions used vs charged ratio is at 100%.

Charged instructions are based on the total thread instruction count charged and includes instructions completed by server tasks that worked on behalf of threads they are working for each interval.

Note: The values shown on this graph are available only for partitions running on Power8® and later hardware. However, it is not available when running in Power7® or Power6® mode.



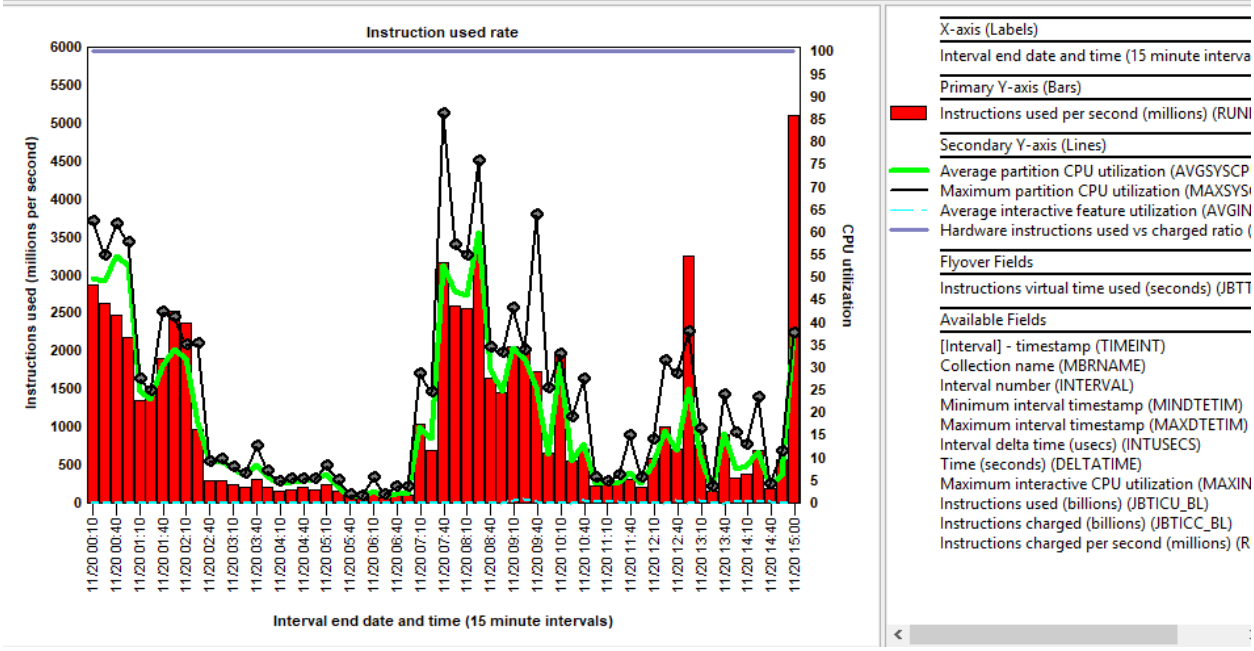
Instructions charged

9.15.14.4 Instructions used rate

This graph shows the instructions used rate per second.

Used instructions are based on the total thread instruction count used without regard to how server tasks work is charged during each interval.

Note: The values shown on this graph are available only for partitions running on Power8® and later hardware. However, it is not available when running in Power7® or Power6® mode.



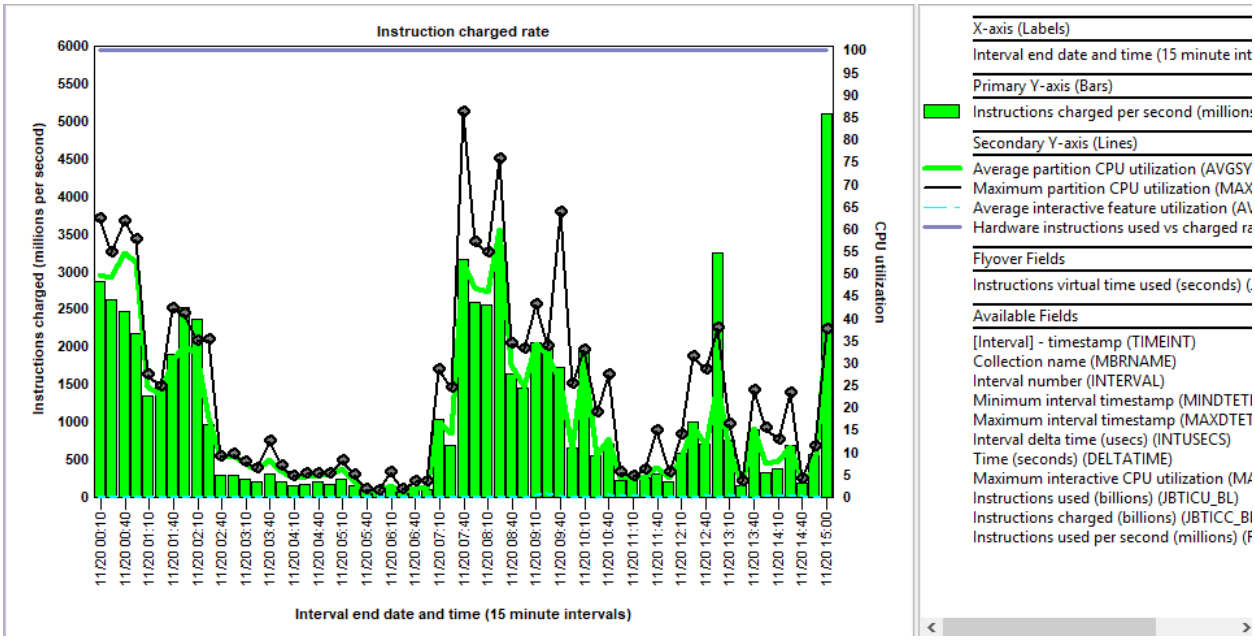
Instructions used rate

9.15.14.5 Instructions charged rate

This graph shows the instructions charged rate per second over time during the collection.

Charged instructions are based on the total thread instruction count charged and includes instructions completed by server tasks that worked on behalf of threads they are working for each interval.

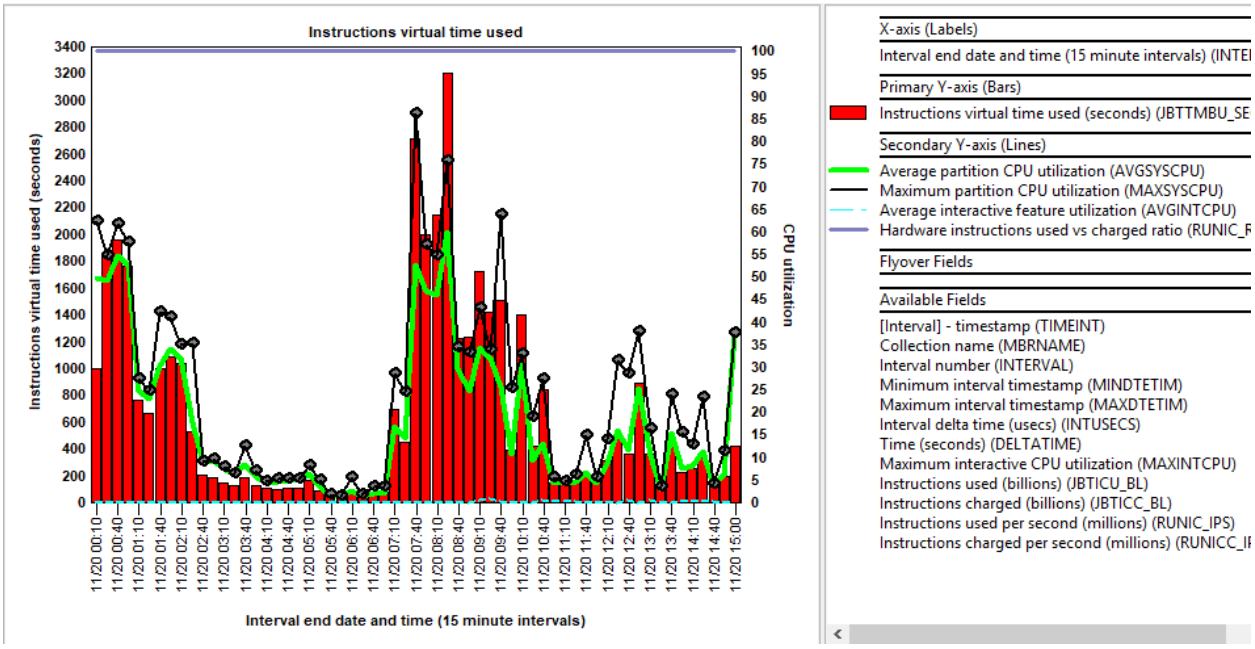
Note: The values shown on this graph are available only for partitions running on Power8® and later hardware. However, it is not available when running in Power7® or Power6® mode.



Instructions charged rate

9.15.14.6 Instructions virtual time used

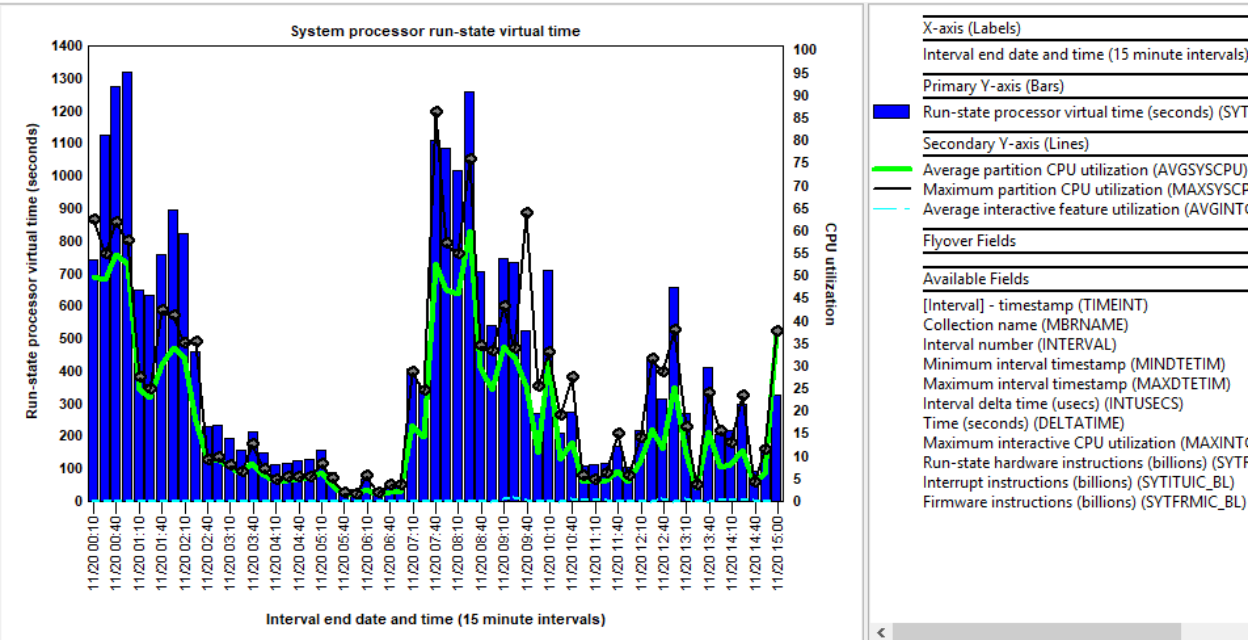
This graph shows the total virtual time used for all threads. This is the amount of time that a processor was running hardware instructions on behalf of jobs/threads on the system.



Instructions virtual time used

9.15.14.7 System processor run-state virtual time

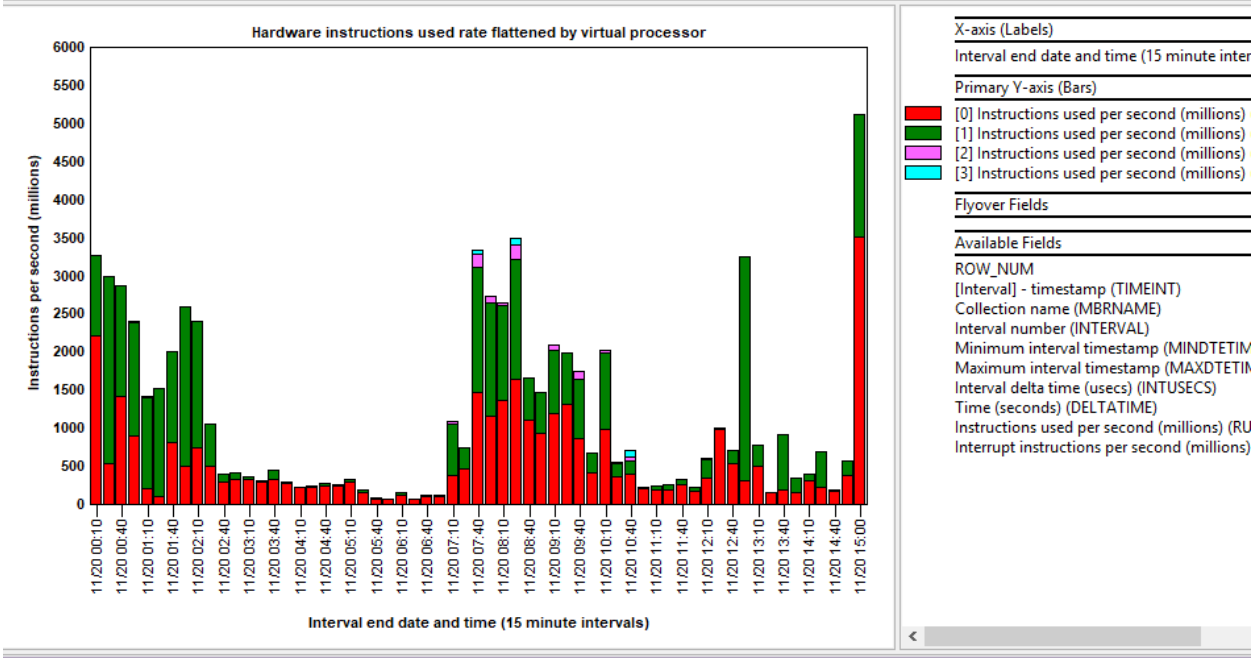
This graph displays the non-idle processor virtual time for the system. The amount of run state physical processor time is summed for all processors. A processor is in run state when any of its threads are in the run state; a processor is in idle state when all of its threads are in idle state



System processor run-state virtual time

9.15.14.8 Hardware instructions used rate flattened by virtual processor

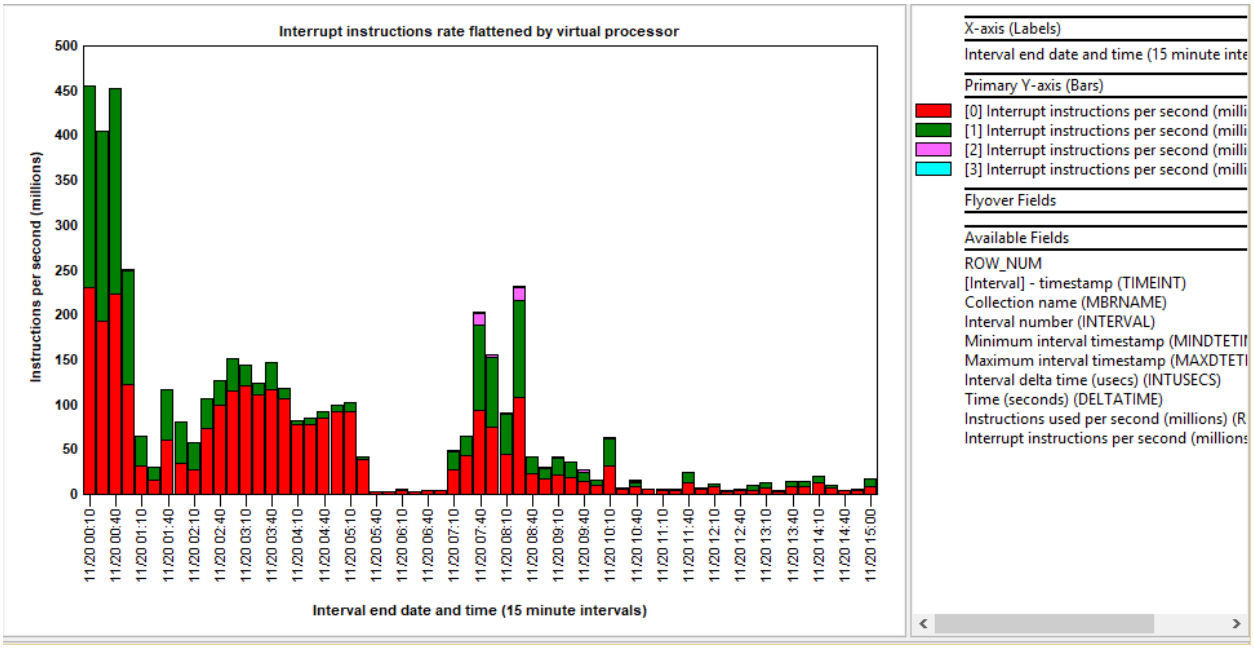
This graph shows the number of hardware instructions used for each virtual processor over time. This value is a rate per second (in millions of instructions per second).



Hardware instructions used rate flattened by virtual processor

9.15.14.9 Interrupt instructions rate flattened by virtual processor

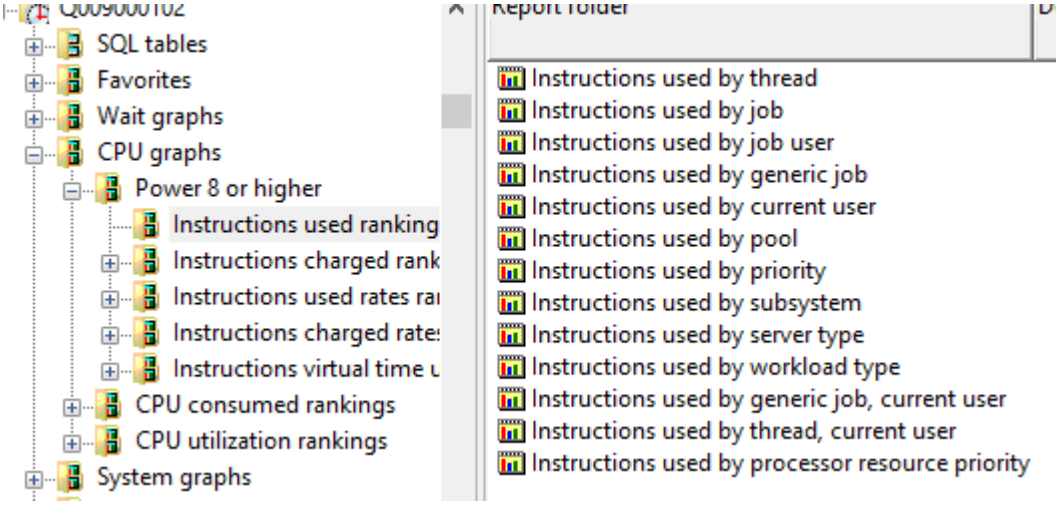
This graph shows the number of interrupt instructions used for each virtual processor over time. This value is a rate per second (in millions of instructions per second).



Interrupt instructions rate flattened by virtual processor

9.15.14.10 Instructions used rankings

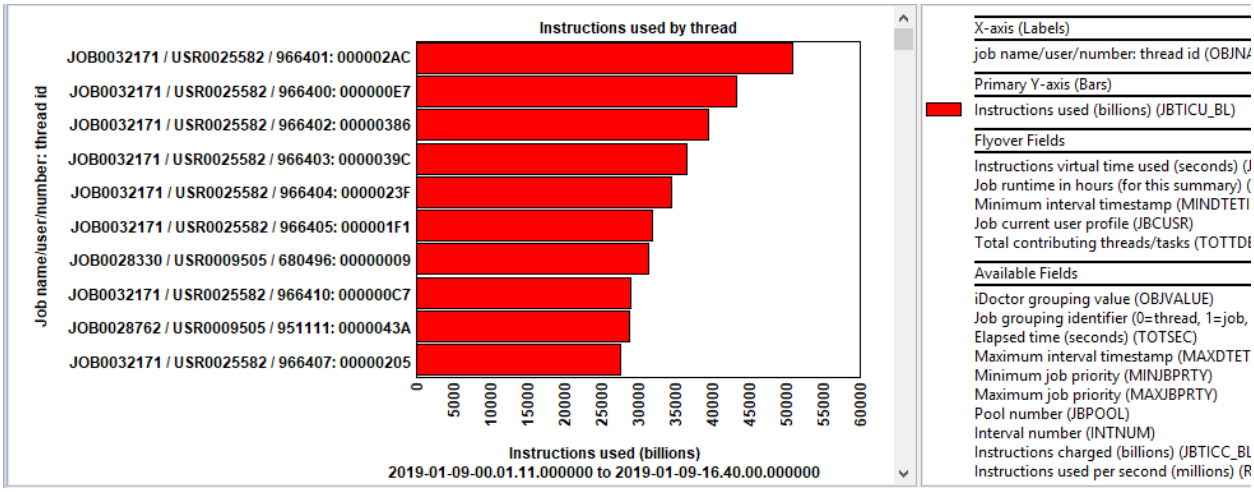
These graphs rank the instructions used by one of several available types of job groupings available.



CPU graphs -> Power 8 or higher -> Instructions used rankings

9.15.14.10.1 Instructions used by thread

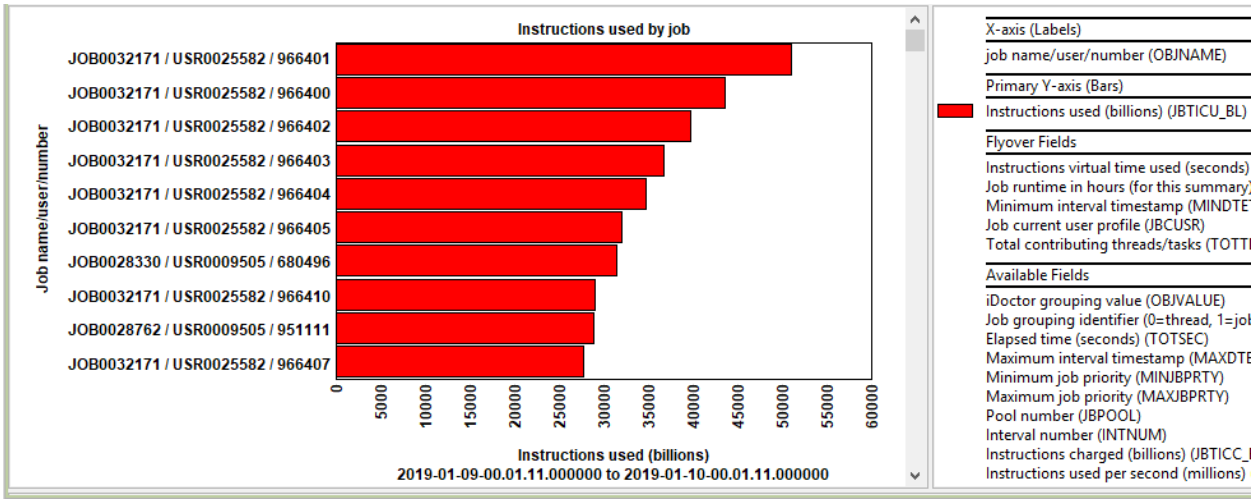
This graph shows the job/threads or system tasks that had the most instructions used during the collection or time selection if this is used via a drill-down.



Instructions used by thread

9.15.14.10.2 Instructions used by job

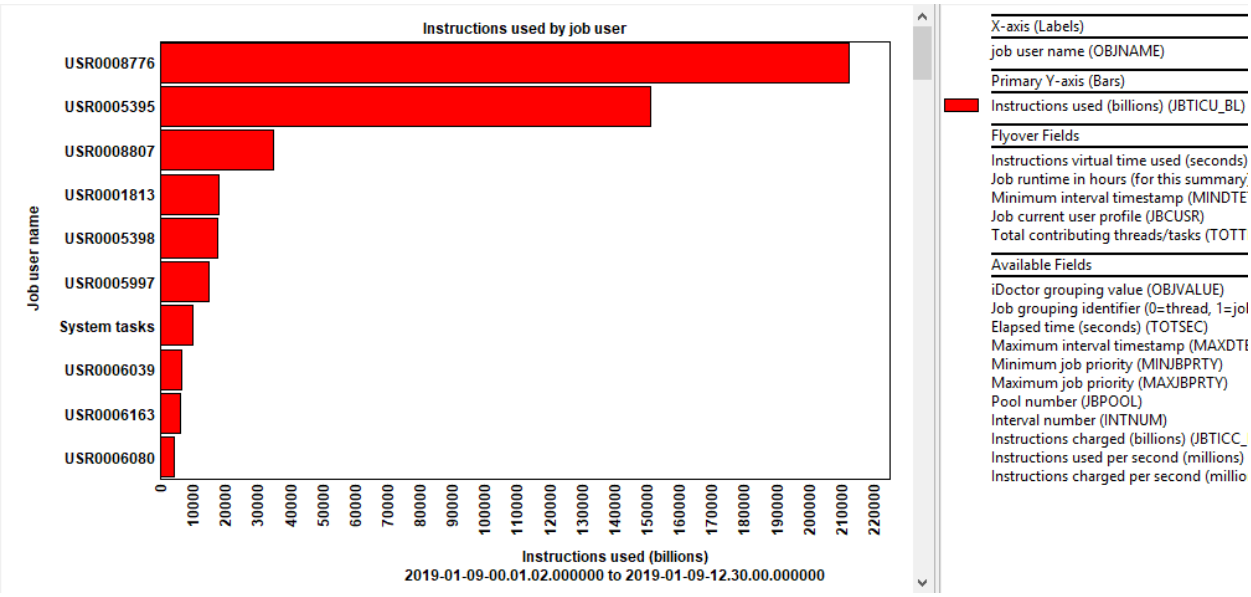
This graph shows the jobs (all threads added together) that used the most instructions during the collection.



Instructions used by job

9.15.14.10.3 Instructions used by job user

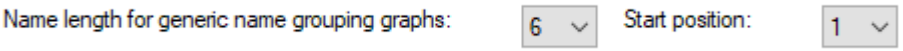
This graph shows the job user name (all jobs having the same user name added together) that used the most instructions during the collection.

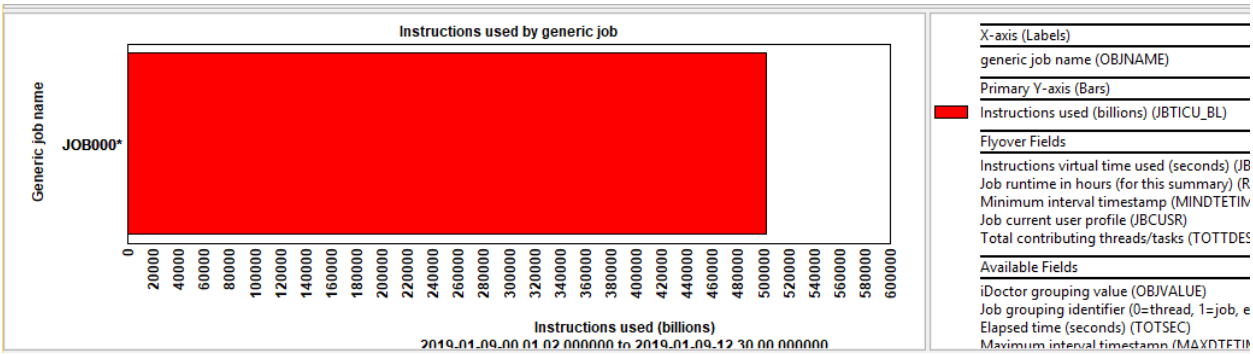


Instructions used by job user

9.15.14.10.4 Instructions used by generic job

This graph ranks the selected time period's instructions used by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

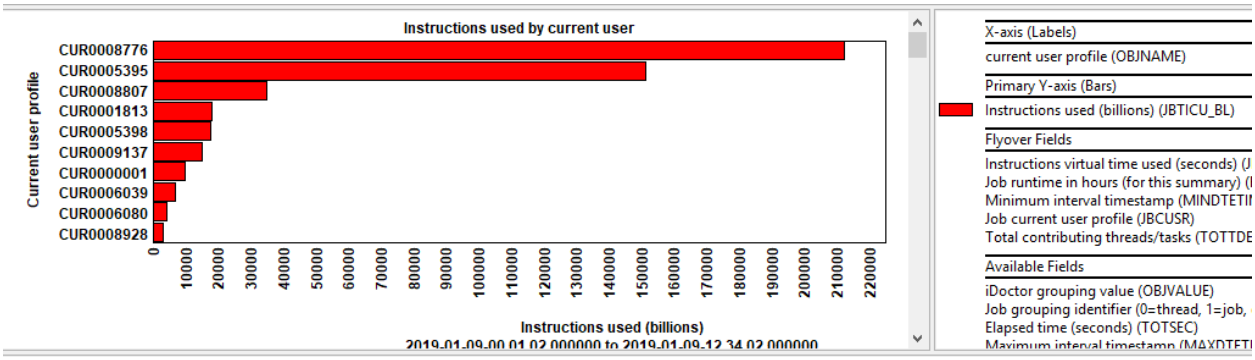




Instructions used by generic job

9.15.14.10.5 Instructions used by current user

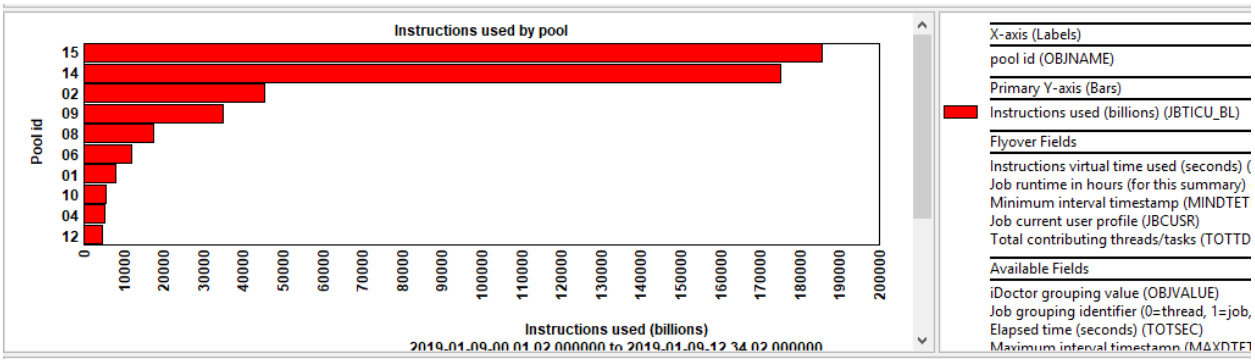
This graph ranks the selected time period's instructions used by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



Instructions used by current user

9.15.14.10.6 Instructions used by pool

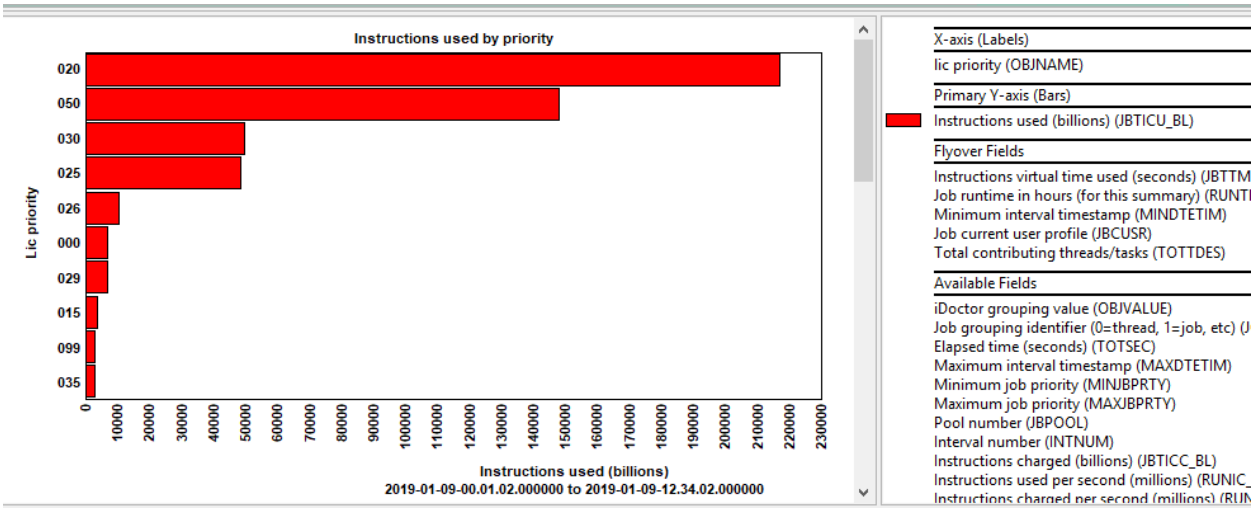
This graph shows the pool ID (all jobs in each pool added together) that used the most instructions during the collection.



Instructions used by pool

9.15.14.10.7 Instructions used by priority

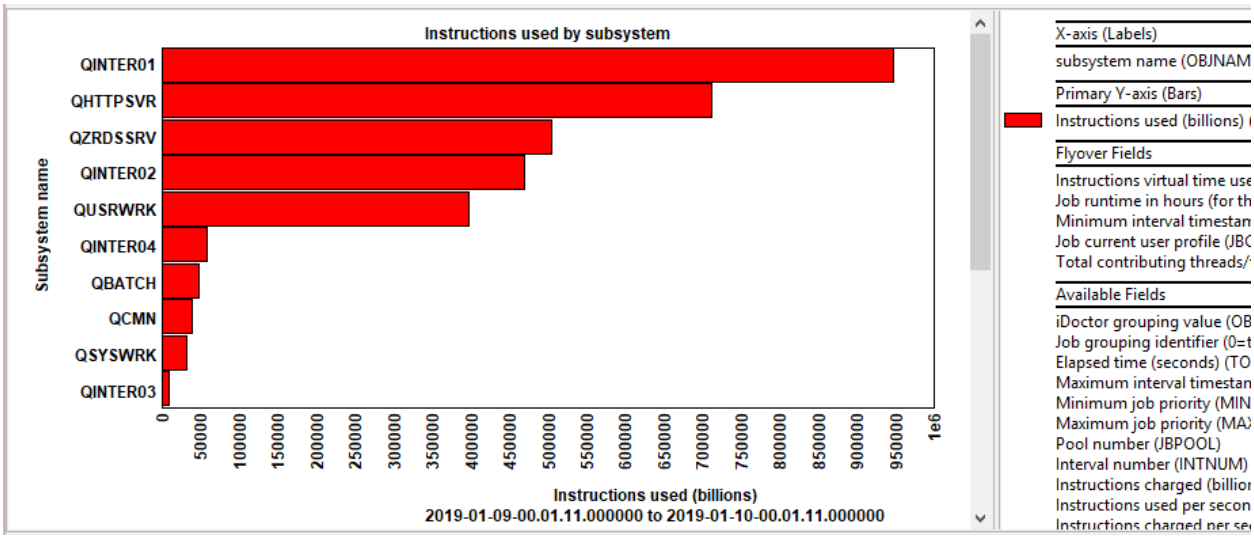
This graph shows the job priority (all job time periods that had the same priority added together) that used the most instructions during the collection.



Instructions used by priority

9.15.14.10.8 Instructions used by subsystem

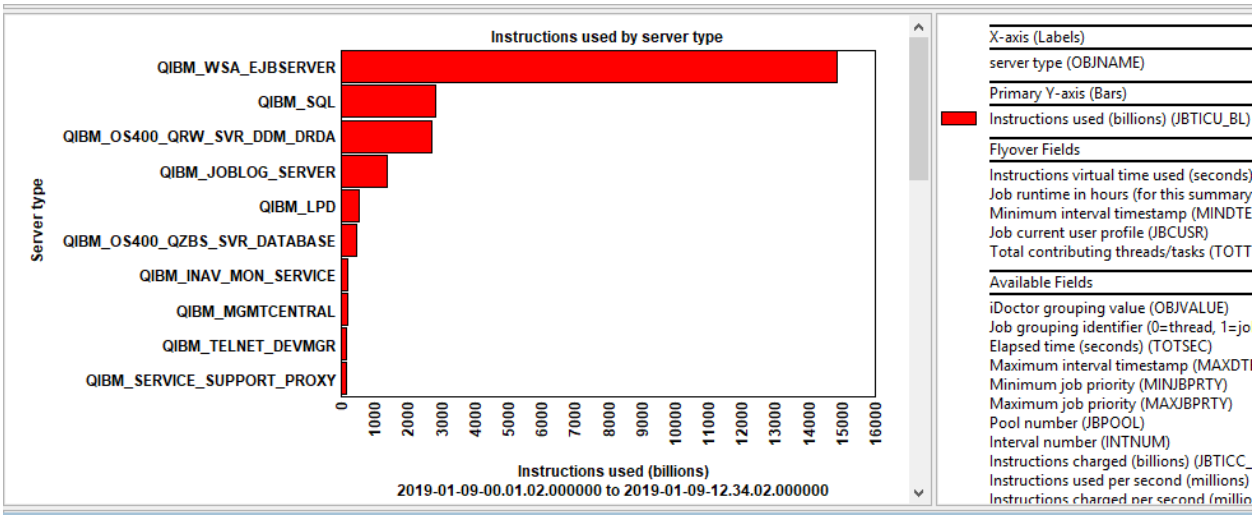
This graph shows the subsystems that used the most instructions during the collection.



Instructions used by subsystem

9.15.14.10.9 Instructions used by server type

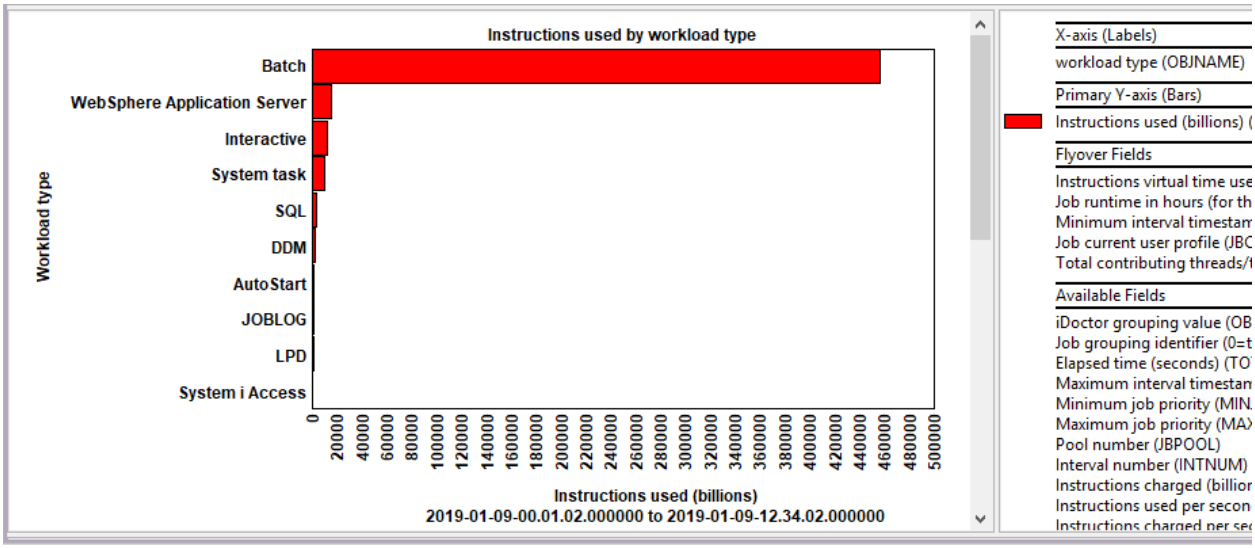
This graph ranks the instructions used during the collection or selected time period by server type.



Instructions used by server type

9.15.14.10.10 Instructions used by workload type

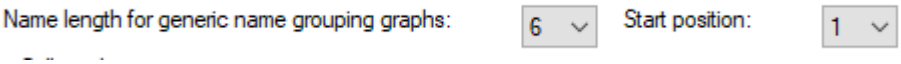
This graph ranks the instructions used during the collection or selected time period by workload type.

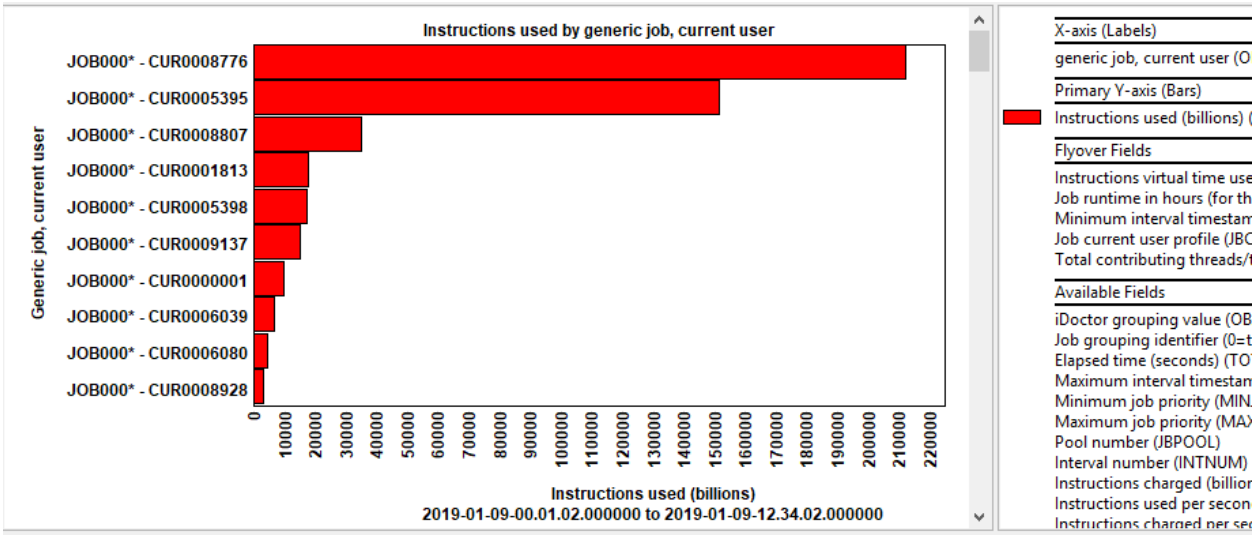


Instructions used by workload type

9.15.14.10.11 Instructions used by generic job, current user

This graph ranks the selected time period's instructions used by a combination of generic job and current user profile. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

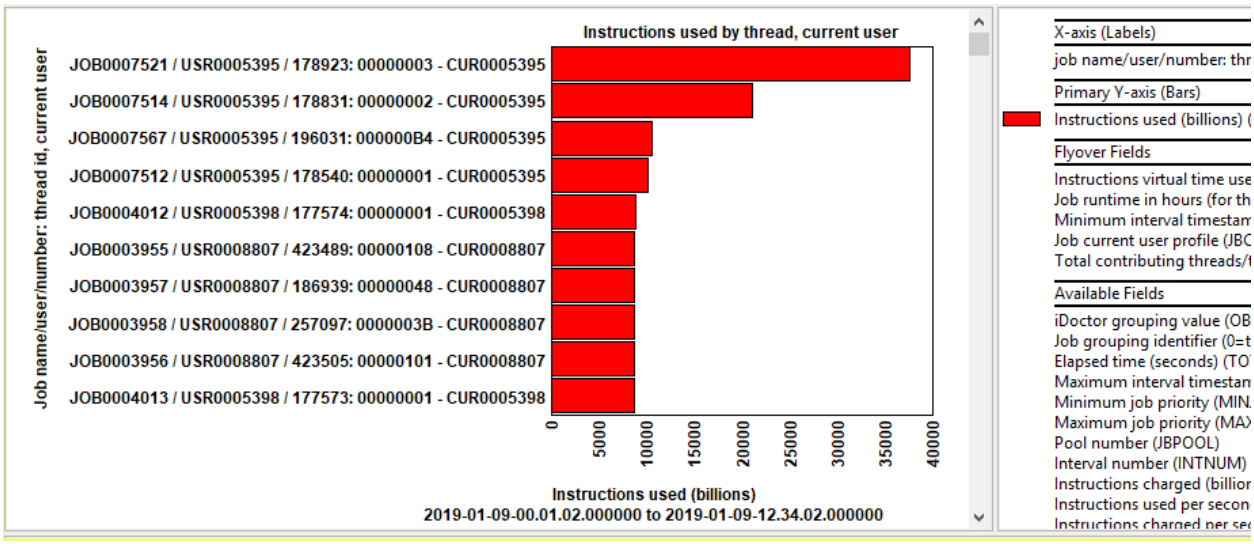




Instructions used by generic job, current user

9.15.14.10.12 Instructions used by thread, current user

This graph shows a combination of thread and current user ranked by those that used the most instructions during the collection.



Instructions used by thread, current user

9.15.14.10.13 Instructions used by processor resource priority

This graph ranks the instructions used by processor resource priority. This is a setting on the CHGJOB command parameter (PRCRSCPTY.)

If your system has simultaneous multithreading (SMT) enabled, the processor resources priority will be used by the operating system to determine the relative importance of the job when it is dispatched compared to other jobs that are dispatched. Performance may vary depending on processor model, partition placement, processor utilization and other system settings and configuration.

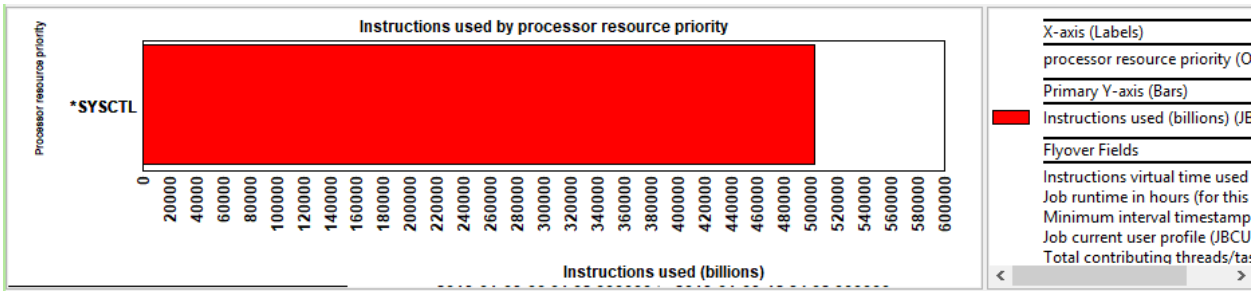
This value will consist of one of the following:

***SYSCTL -** The system determines the processor resources priority for the job and its threads.

***HIGH -** The operating system will isolate this job and its threads, when possible, to processors with fewer threads running concurrently.

***NORMAL -** The job and its threads will be dispatched uniformly across the maximum number of available processors.

***LOW -** The job and its threads will be dispatched to processors with other low priority jobs executing, when possible.

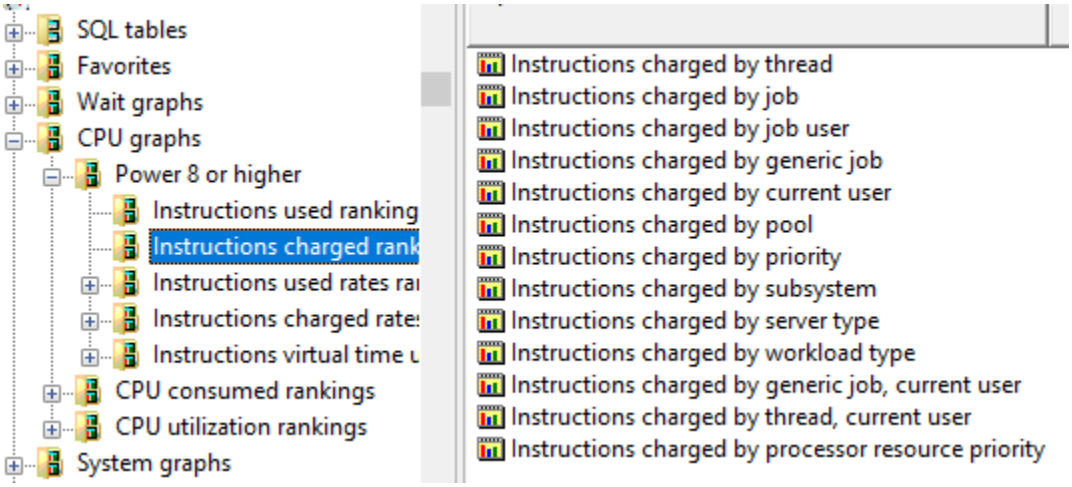


Instructions used by processor resource priority

9.15.14.11 Instructions charged rankings

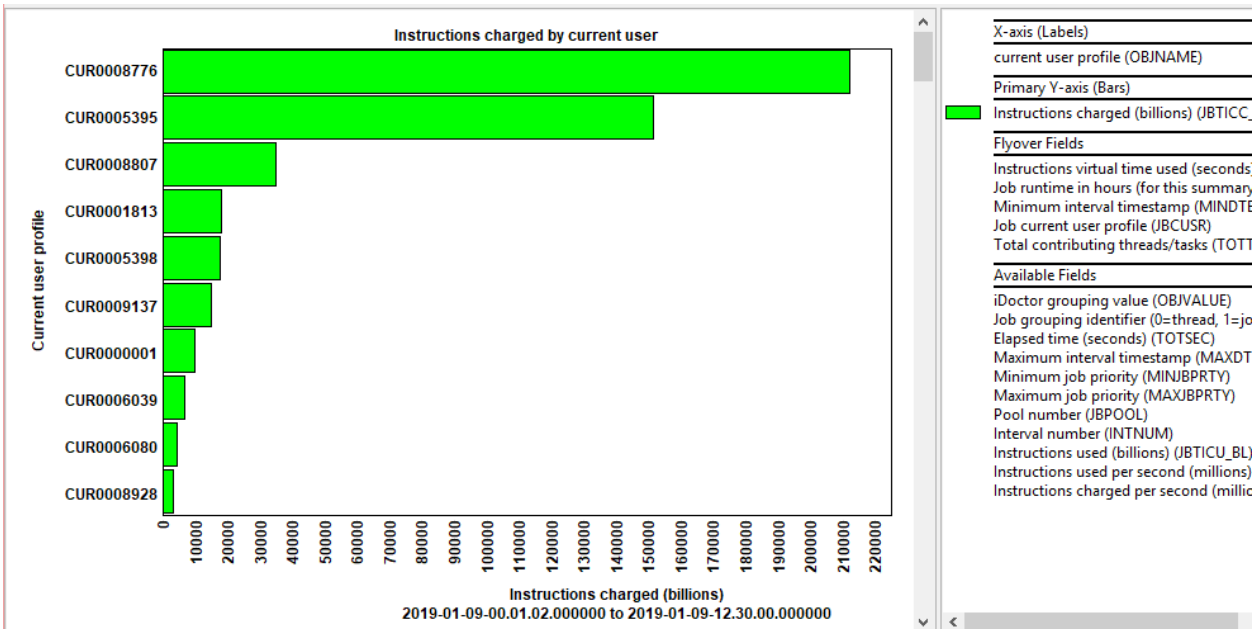
These graphs rank the instructions charged by one of several available types of job groupings available.

Charged instructions are based on the total thread instruction count charged and includes instructions completed by server tasks that worked on behalf of threads they are working for each interval.



CPU graphs -> Power 8 or higher -> Instructions charged rankings

An example is shown below:

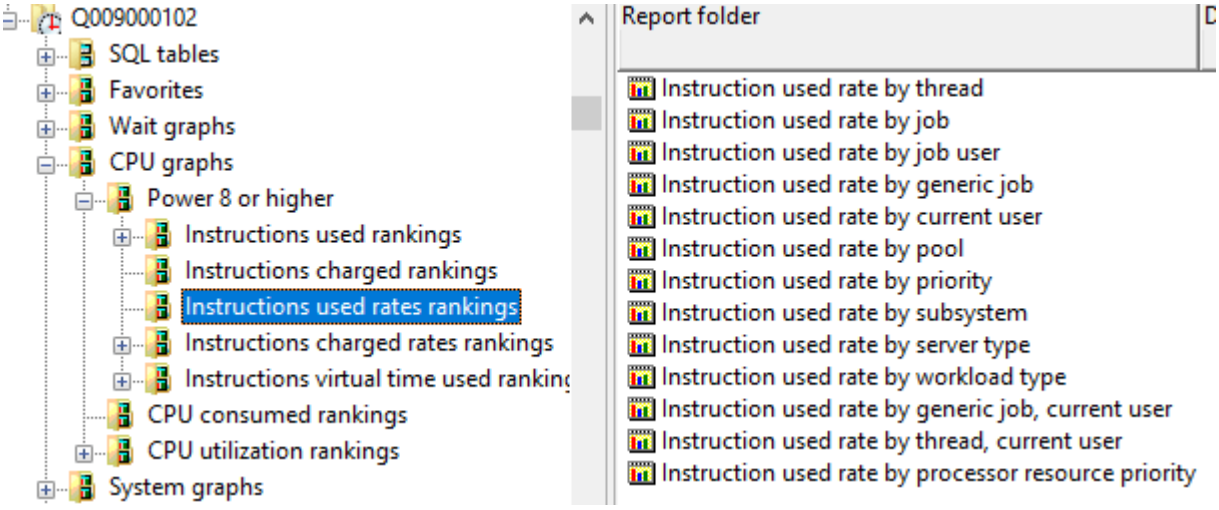


Instructions charged by current user

For more examples see the Instructions used rankings section.

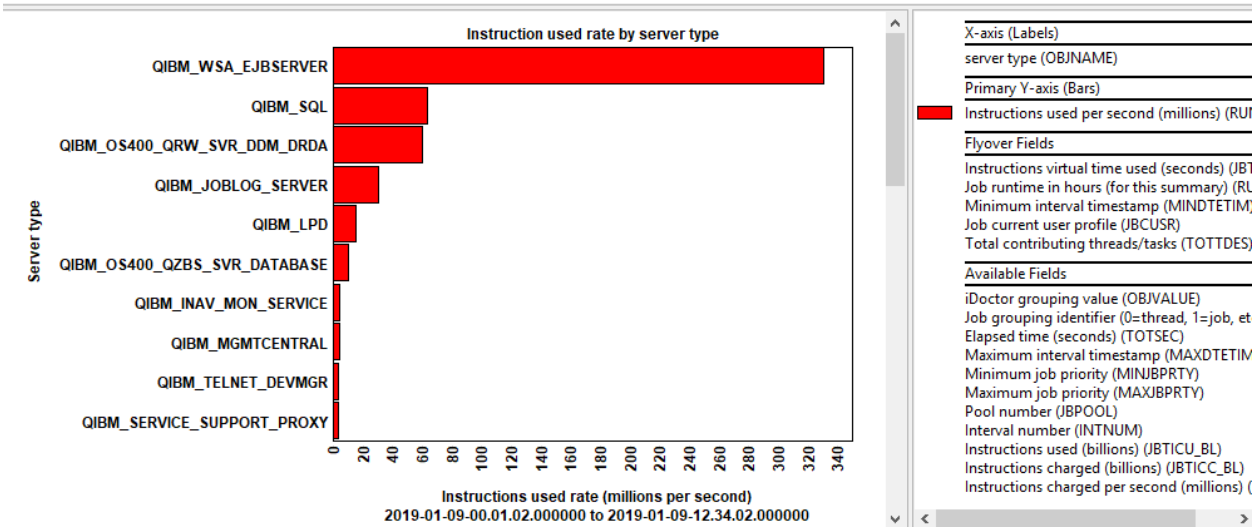
9.15.14.12 Instructions used rates rankings

These graphs rank the instructions used per second by one of several available types of job groupings available.



CPU graphs -> Power 8 or higher -> Instructions used rates rankings

An example is shown below:

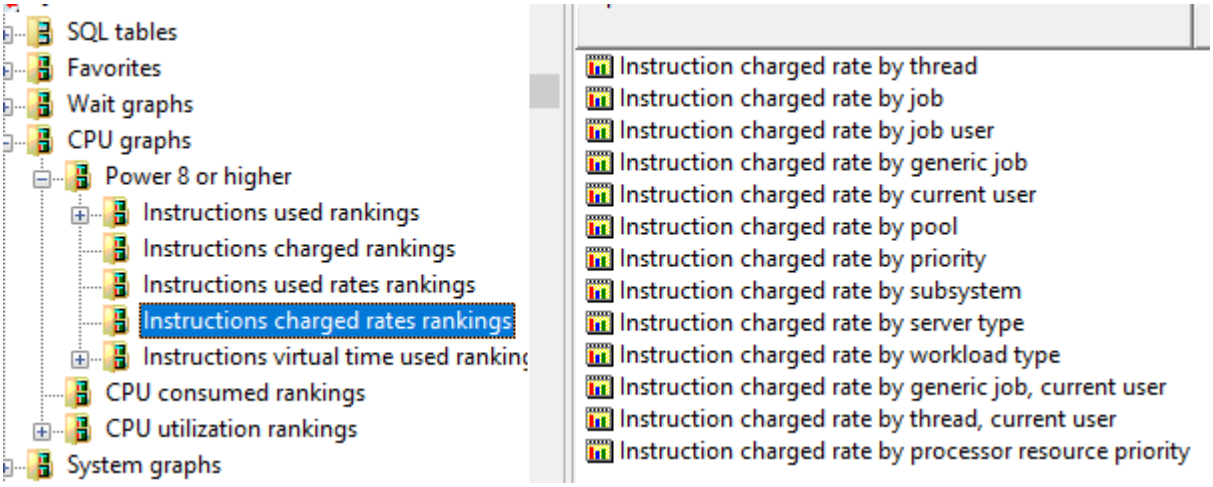


Instructions used rate by server type

For more examples see the Instructions used rankings section.

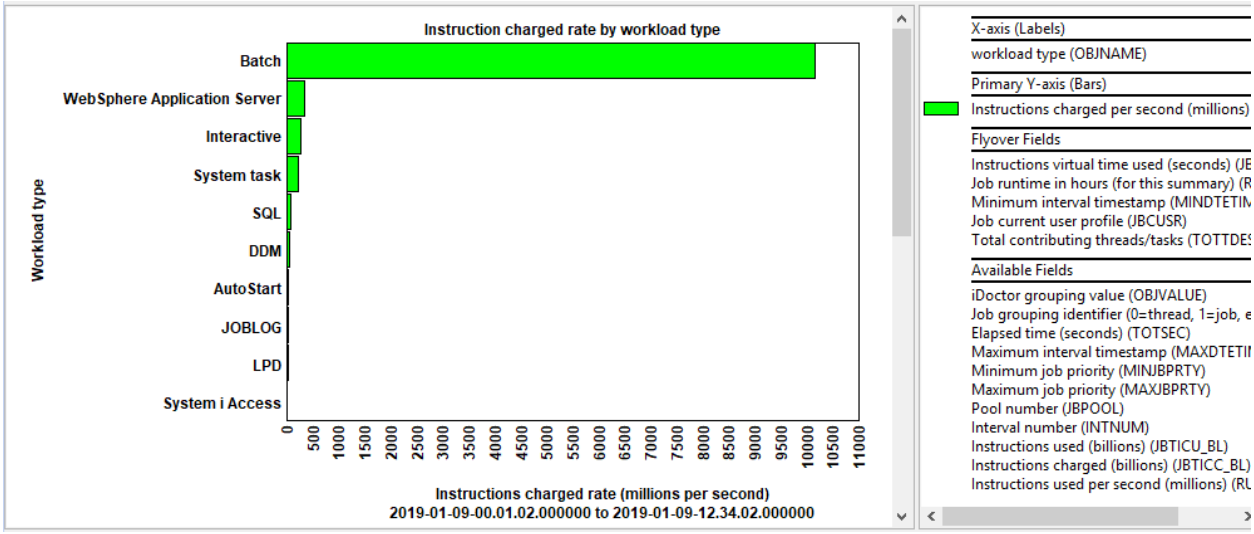
9.15.14.13 Instructions charged rates rankings

These graphs rank the instructions charged per second by one of several available types of job groupings available.



CPU graphs -> Power 8 or higher -> Instructions charged rates rankings

An example is shown below:

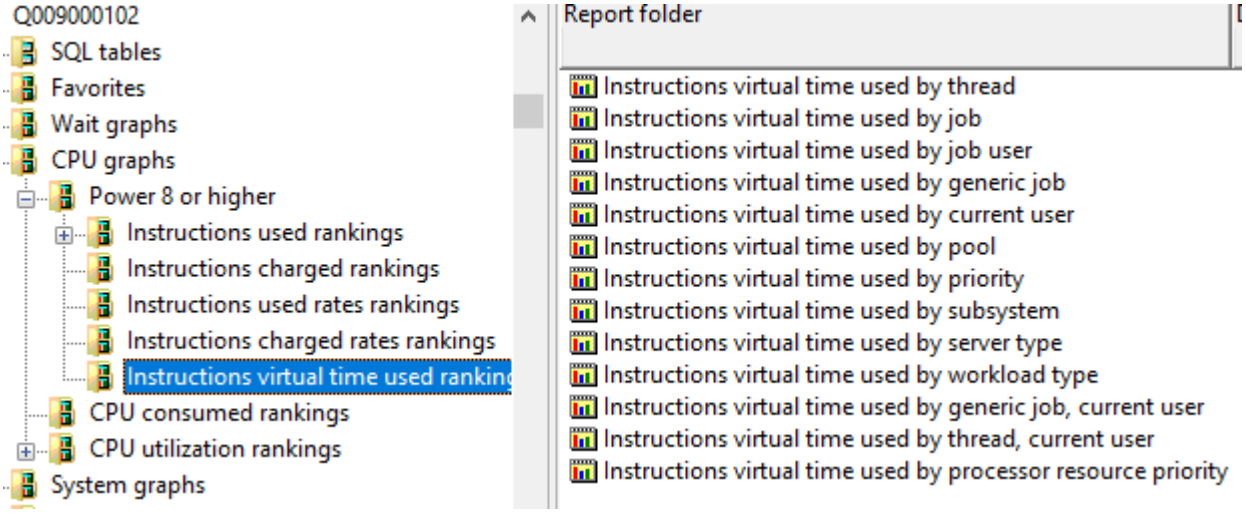


Instructions charged rate by workload type

For more examples see the Instructions used rankings section.

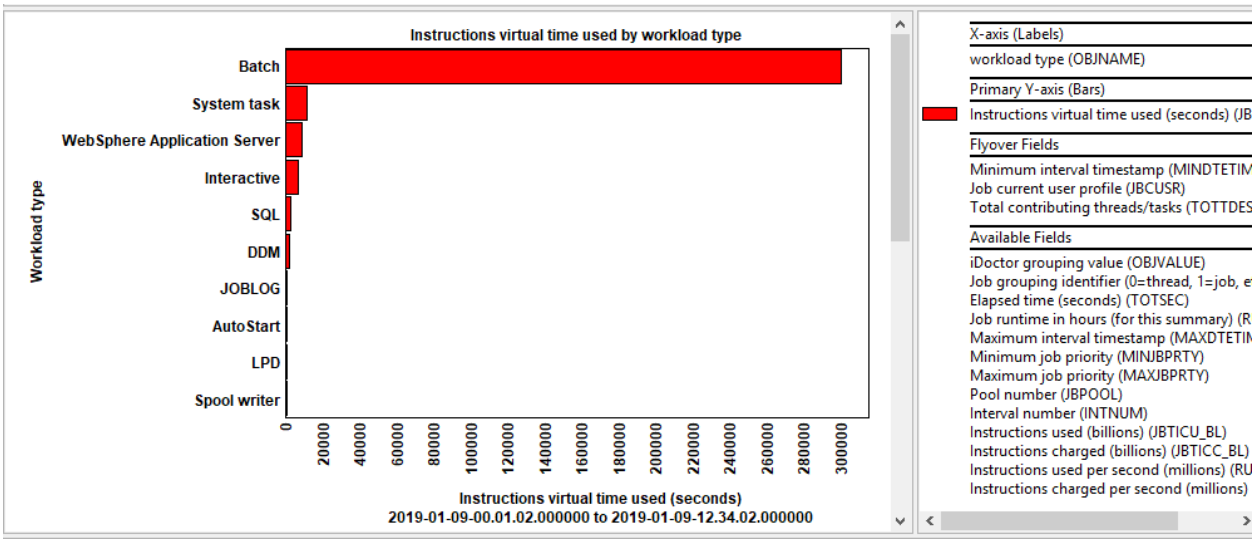
9.15.14.14 Instructions virtual time used rankings

These graphs rank the instructions virtual time used ranked by one of several available types of job groupings available. This is the amount of time that a processor was running hardware instructions on behalf of jobs/threads on the system.



CPU graphs -> Power 8 or higher -> Instructions virtual time used rankings

An example is shown below:

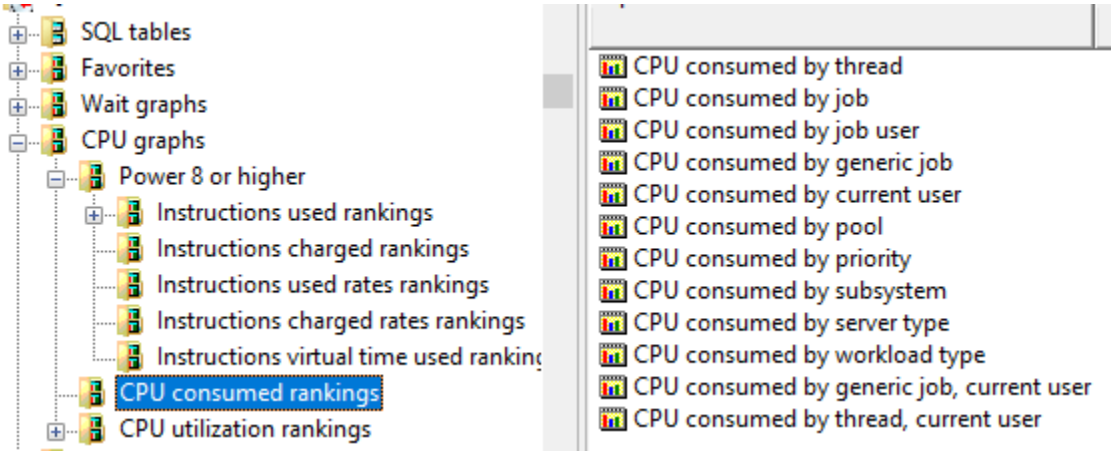


Instructions virtual time used by workload type

For more examples see the Instructions used rankings section.

9.15.15 CPU consumed rankings

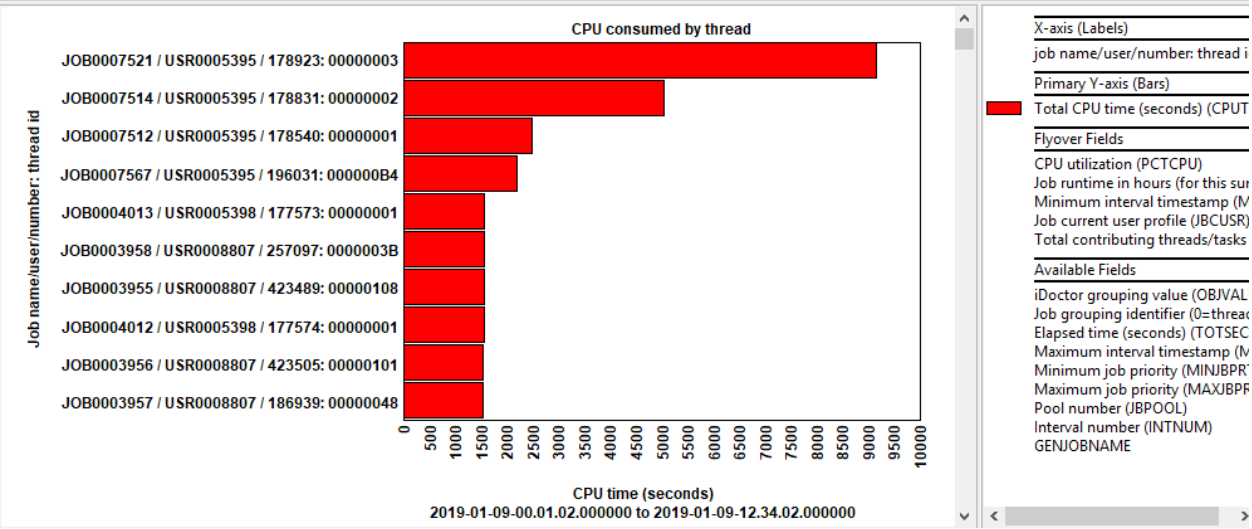
These graphs display the amount of “job” CPU (not the same as Dispatched CPU) grouped by one of many job groupings:



CPU graphs -> CPU consumed rankings

9.15.15.1 CPU consumed by thread

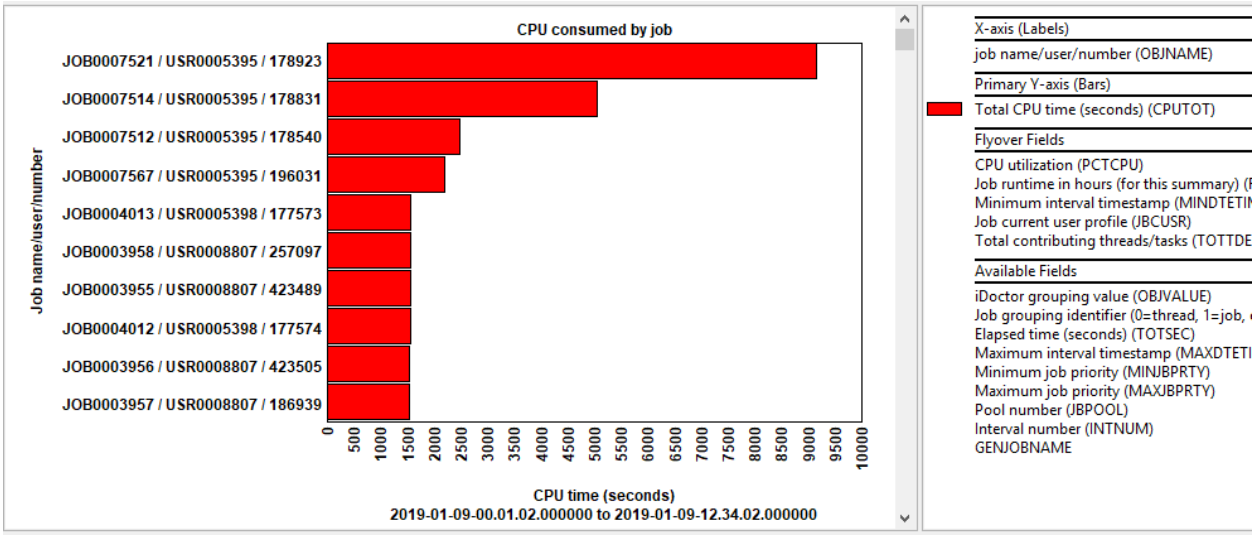
This graph shows the job/threads or system tasks that had the most CPU consumed during the collection or time selection if this is used via a drill-down.



CPU consumed by thread

9.15.15.2 CPU consumed by job

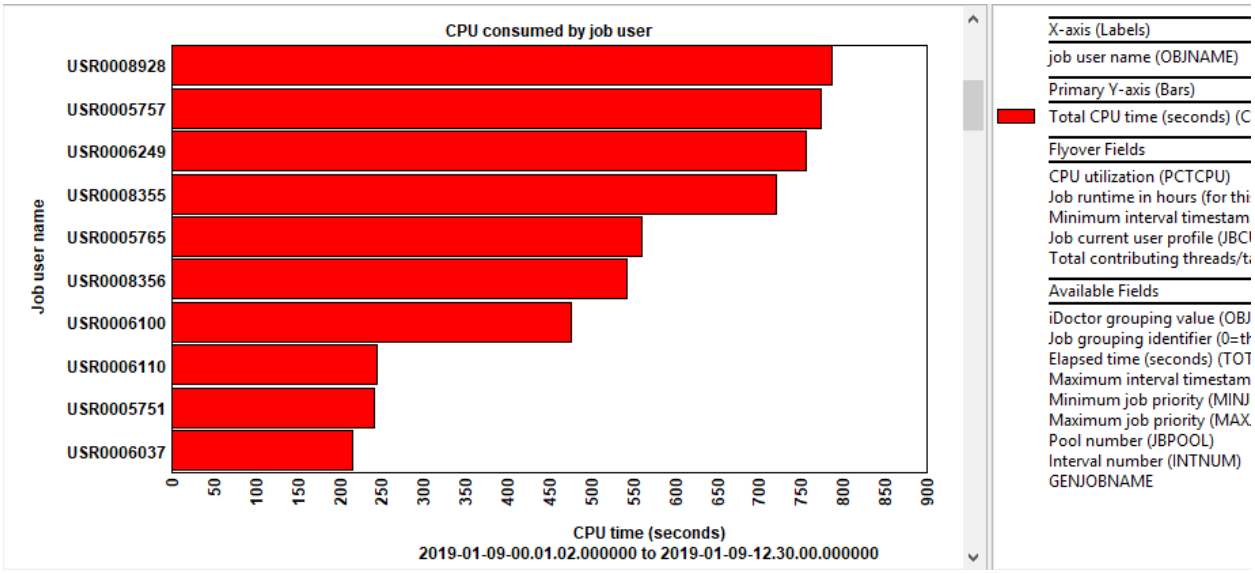
This graph shows the jobs (all threads added together) that consumed the most CPU during the collection.



CPU consumed by job

9.15.15.3 CPU consumed by job user

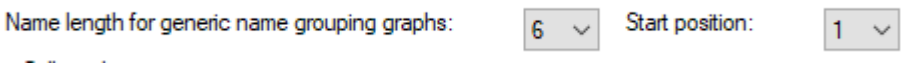
This graph shows the job user name (all jobs having the same user name added together) that used the most CPU during the collection.

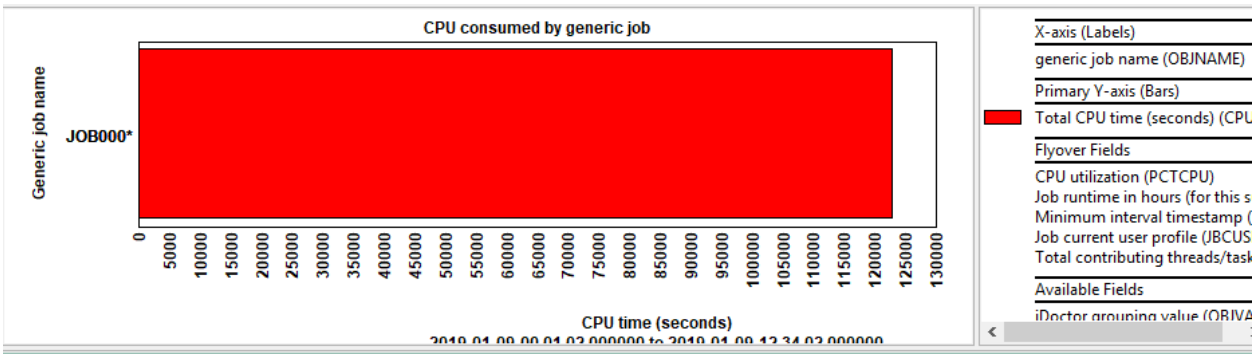


CPU consumed by job user

9.15.15.4 CPU consumed by generic job

This graph ranks the selected time period's CPU consumed by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

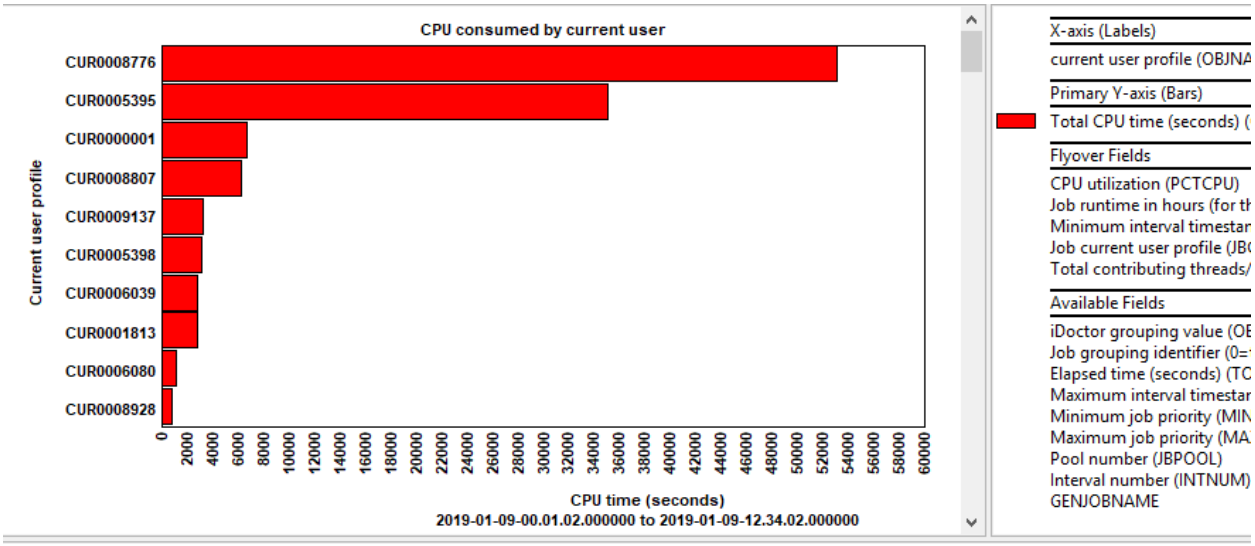




CPU consumed by generic job

9.15.15.5 CPU consumed by current user

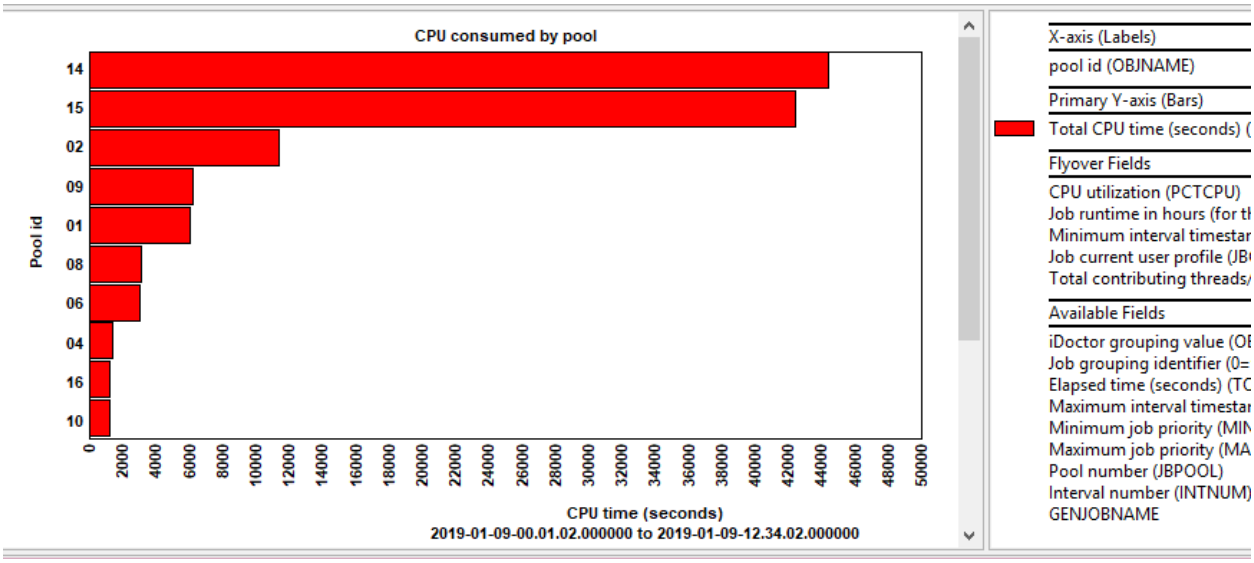
This graph ranks the selected time period's CPU consumed by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



CPU consumed by current user

9.15.15.6 CPU consumed by pool

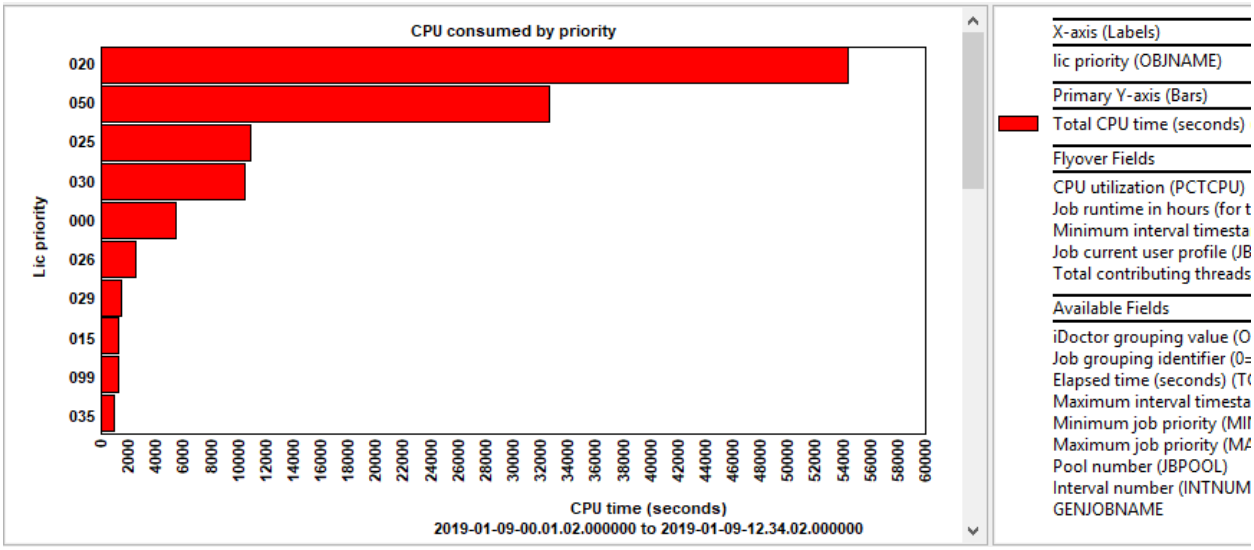
This graph shows the pool ID (all jobs in each pool added together) that used the most CPU during the collection.



CPU consumed by pool

9.15.15.7 CPU consumed by priority

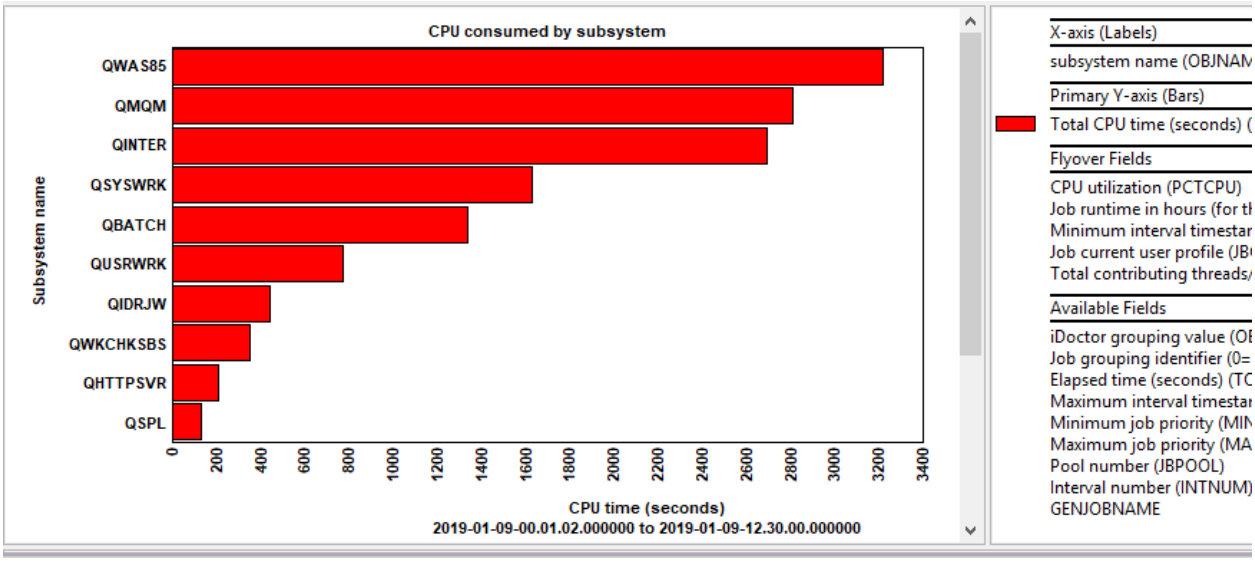
This graph shows the job priority (all job time periods that had the same priority added together) that used the most CPU during the collection.



CPU consumed by priority

9.15.15.8 CPU consumed by subsystem

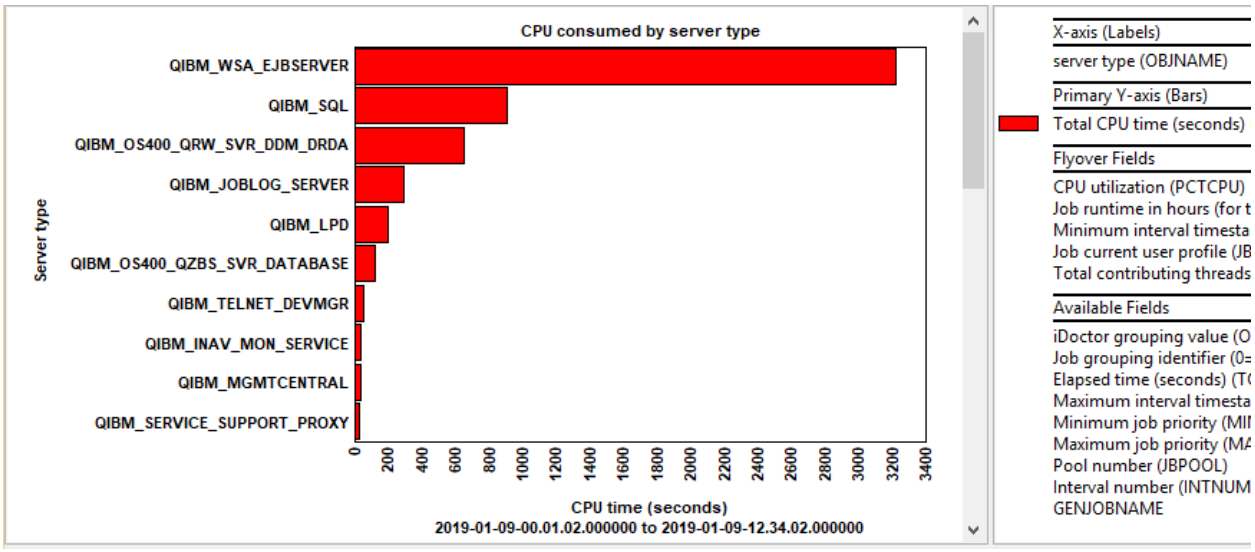
This graph shows the subsystems that used the most CPU during the collection.



CPU consumed by subsystem

9.15.15.9 CPU consumed by server type

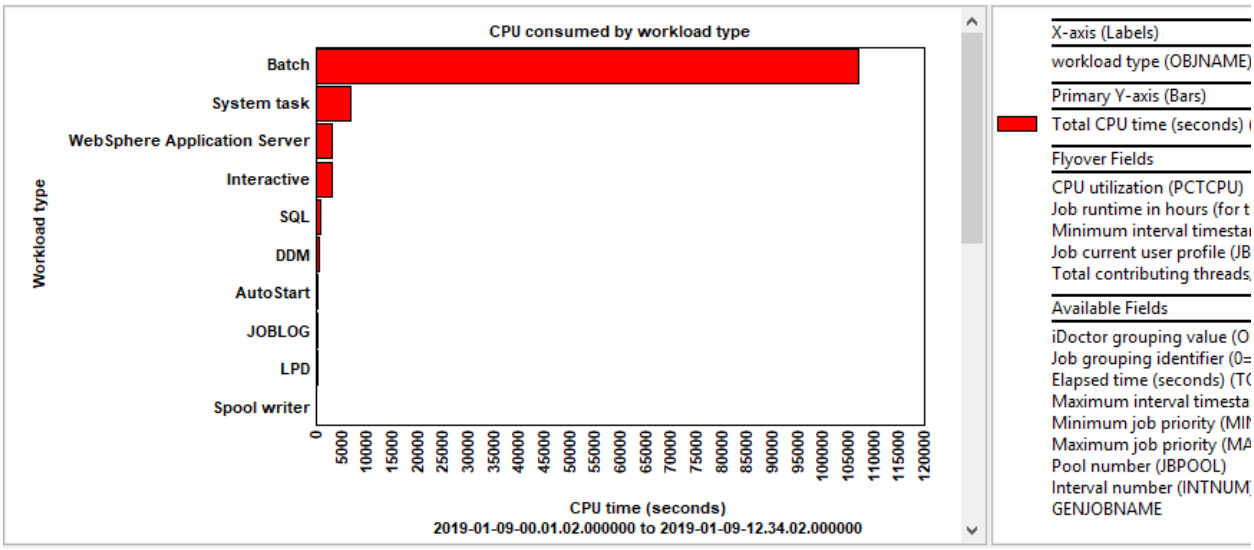
This graph ranks the CPU consumed during the collection or selected time period by server type.



CPU consumed by server type

9.15.15.10 CPU consumed by workload type

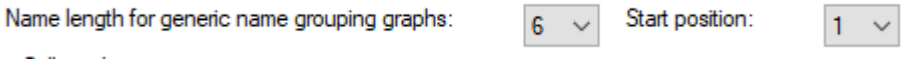
This graph ranks the CPU consumed during the collection or selected time period by workload type.

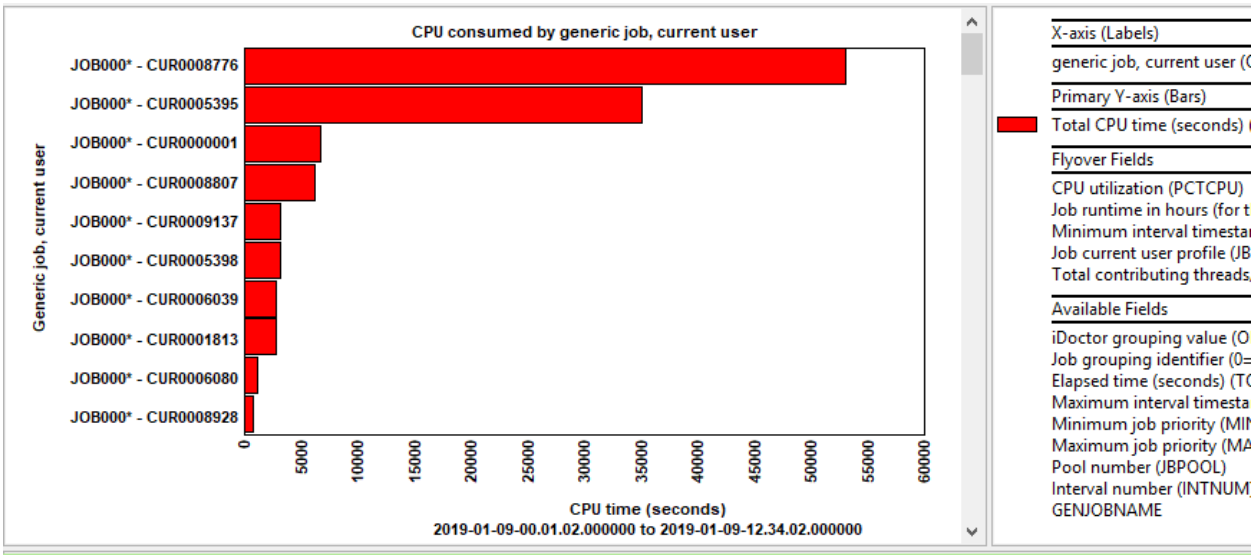


CPU consumed by workload type

9.15.15.11 CPU consumed by generic job, current user

This graph ranks the selected time period's CPU consumed by a combination of generic job and current user profile. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

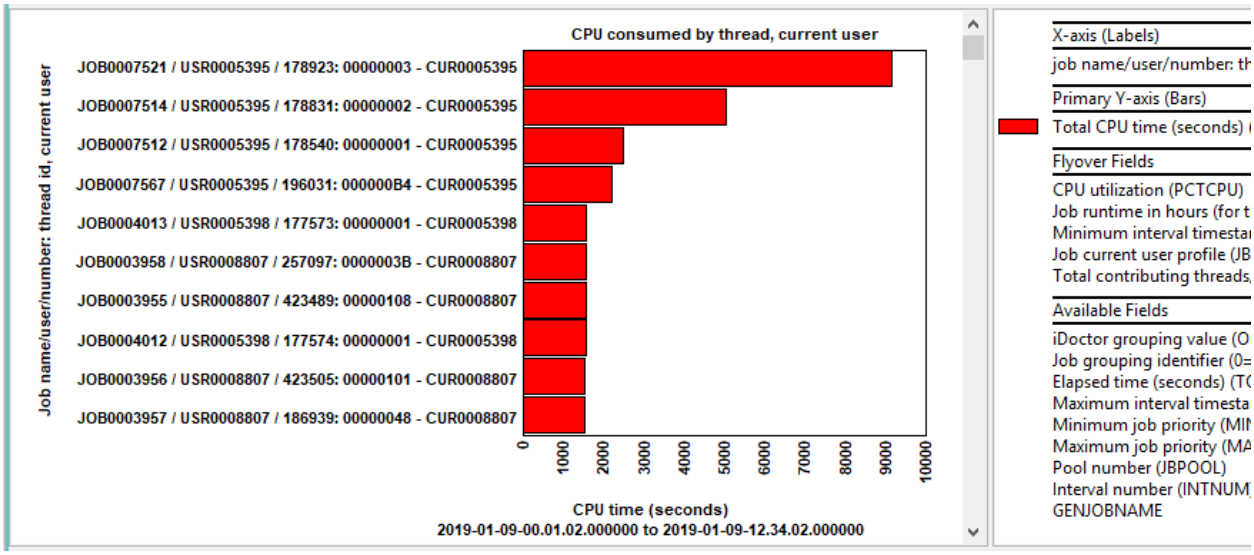




CPU consumed by generic job, current user

9.15.15.12 CPU consumed by thread, current user

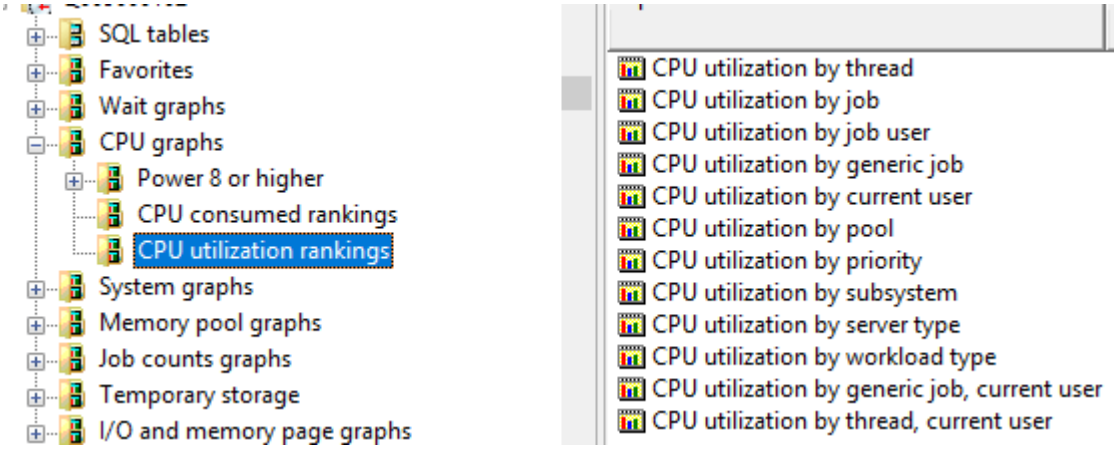
This graph shows a combination of thread and current user ranked by those that used the most CPU during the collection.



CPU consumed by thread, current user

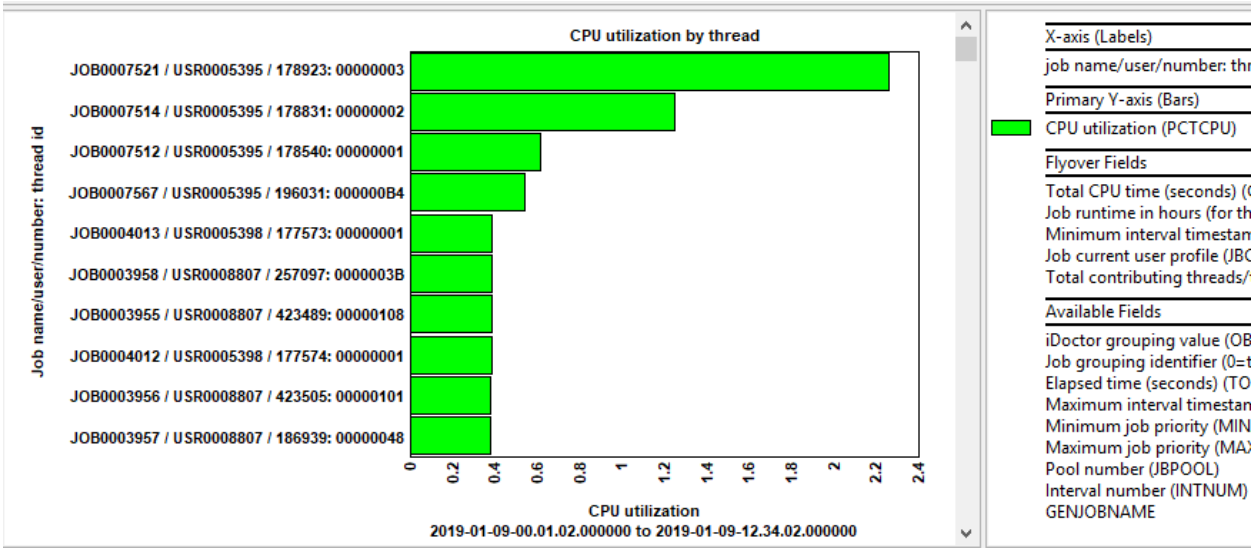
9.15.16 CPU utilization rankings

These graphs display the amount of CPU utilization grouped by one several job groupings:



CPU graphs -> CPU utilization rankings

An example of this graph is shown below:



CPU utilization by thread

For more examples see the CPU consumed rankings section.

9.16 System graphs

These graphs show several key statistics (CPU, memory) covering the entire system and all partitions.

Additional metrics are also provided for TLBIEs, Interrupts and/or partition affinity but will only appear if the data of these types has been captured.

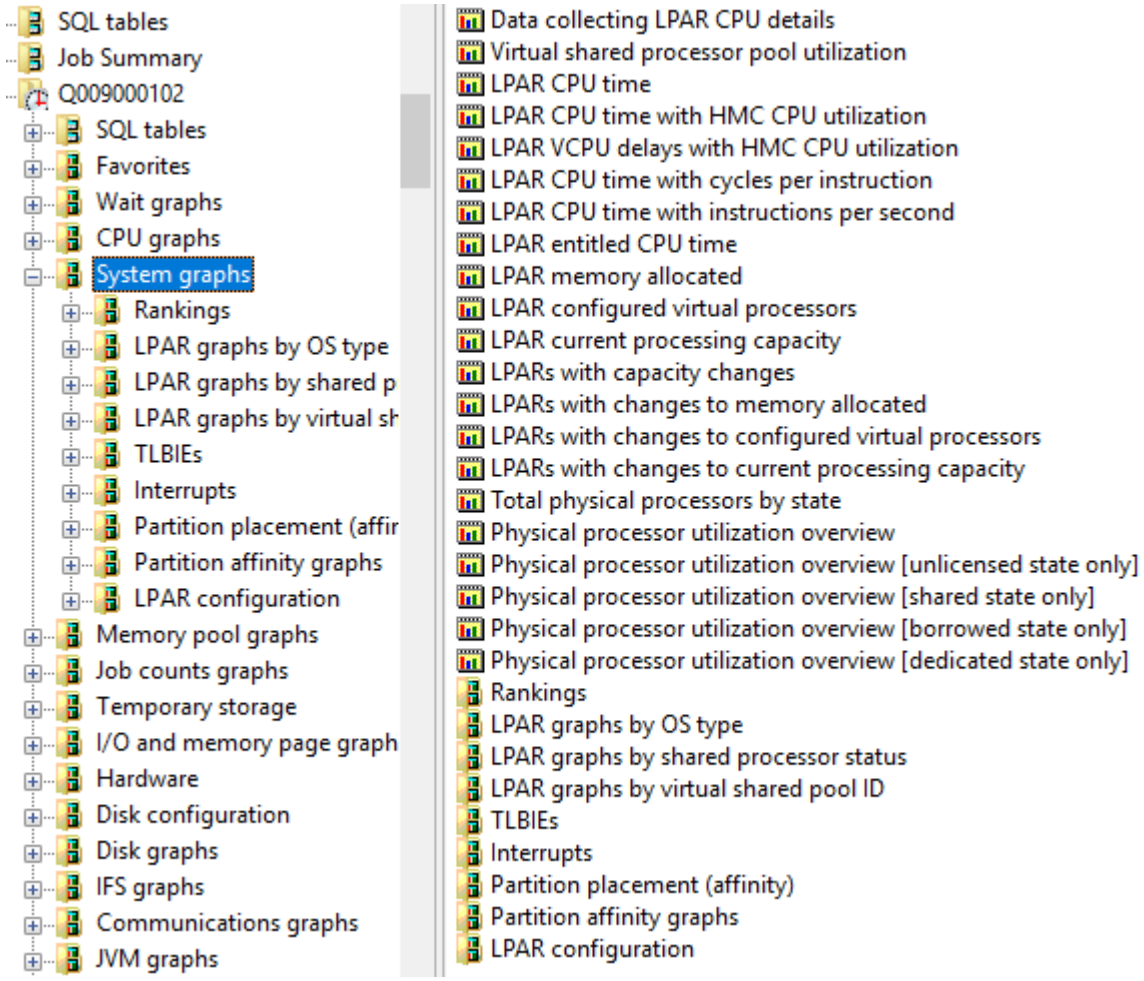
The CPU data is captured through the HMC and stored in Collection Services file QAPMLPARH.

For more information on collecting the CPU statistics for all partitions within Collection Services see this web page:

https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzahx/rzahxcollectdisplaycpuforallpartitions.htm

Shared memory pool data is also captured through the HMC and stored in file QAPMSHRMP. Data is generated only when a partition is defined to use a shared memory pool. Data is reported for both the partition's use of the pool as well as pool metrics that are the sum of activity caused by all partitions using the pool.

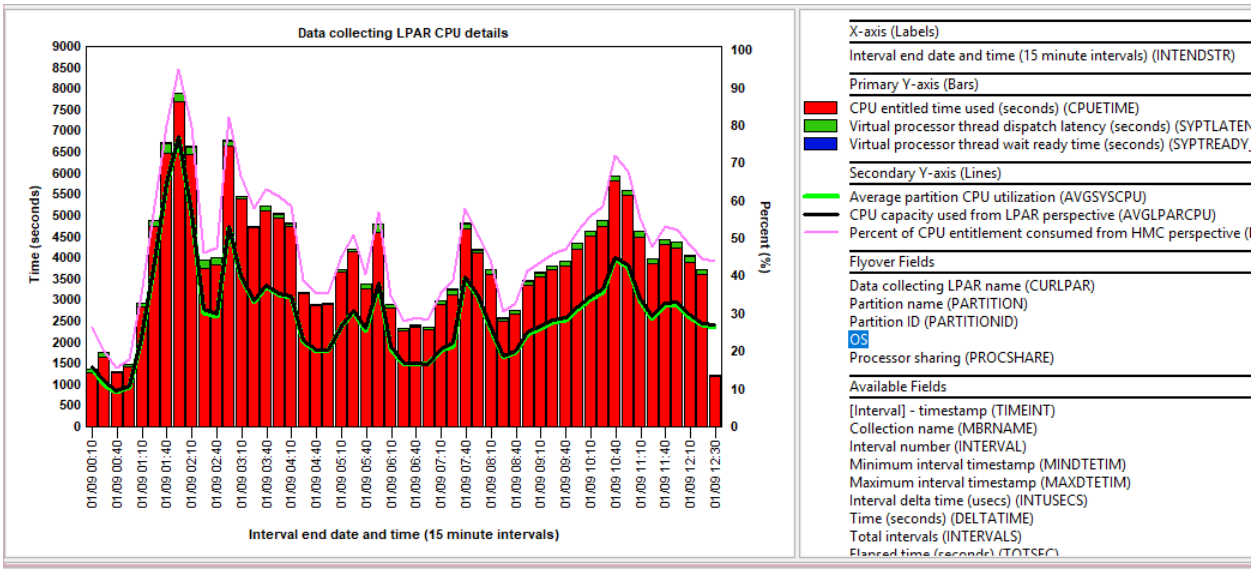
Note: The graphs and graph folders shown in the System graphs folder may vary greatly depending on the data available.



System graphs folder

9.16.1 Data collecting LPAR CPU details

This graph will show CPU entitled for the current LPAR as well as any virtual processor delays (thread dispatch latency and thread wait ready time.) The secondary Y-axis contains additional metrics for CPU capacity used from the LPAR's perspective or CPU entitlement consumption from the HMC's perspective.

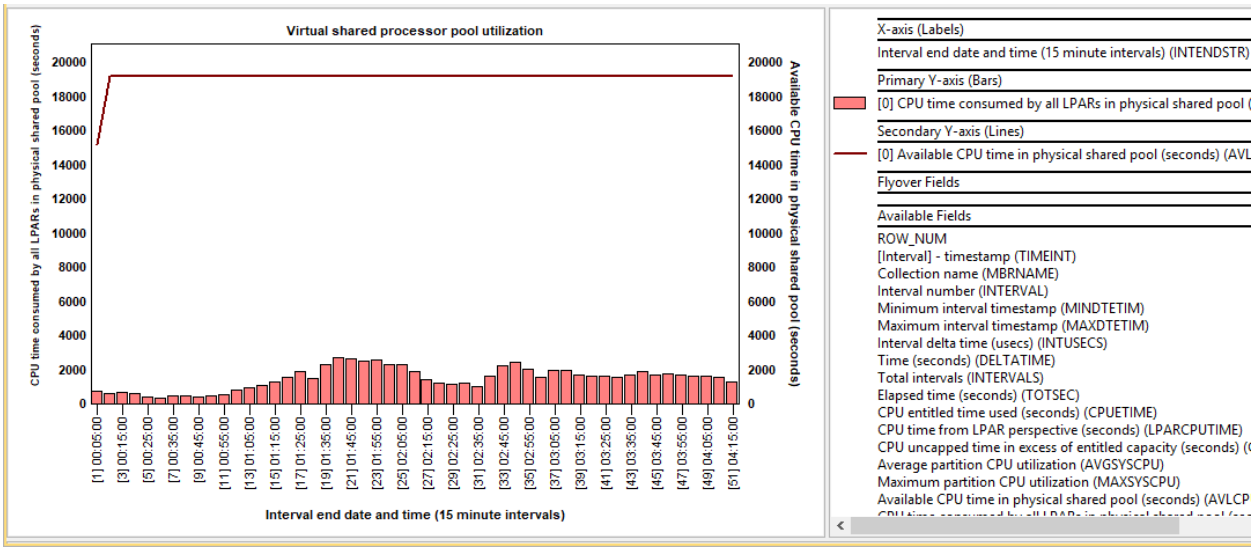


Data collecting LPAR CPU details

9.16.2 Virtual shared processor pool utilization

This graph is used to graph the shared processor pool utilization for all LPARs. It shows the time consumed on the primary Y-axis and the available CPU time in the physical shared pool on the secondary Y-axis.

Note: This is a flattened graph and potentially multiple pools could be shown here each with a different color (unlike the example below.) The pool number is shown within brackets [0] in the graph legend.

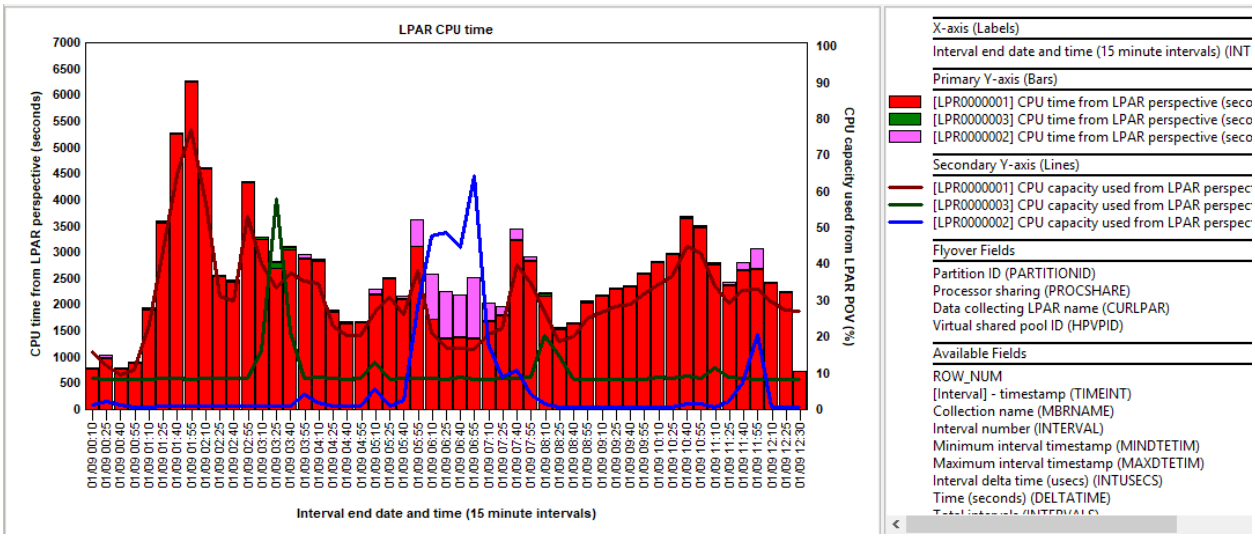


Virtual shared processor pool utilization

9.16.3 LPAR CPU time

This graph shows the total CPU time used by each partition along with the average CPU utilization (from each LPAR's perspective) over time on the secondary Y-axis.

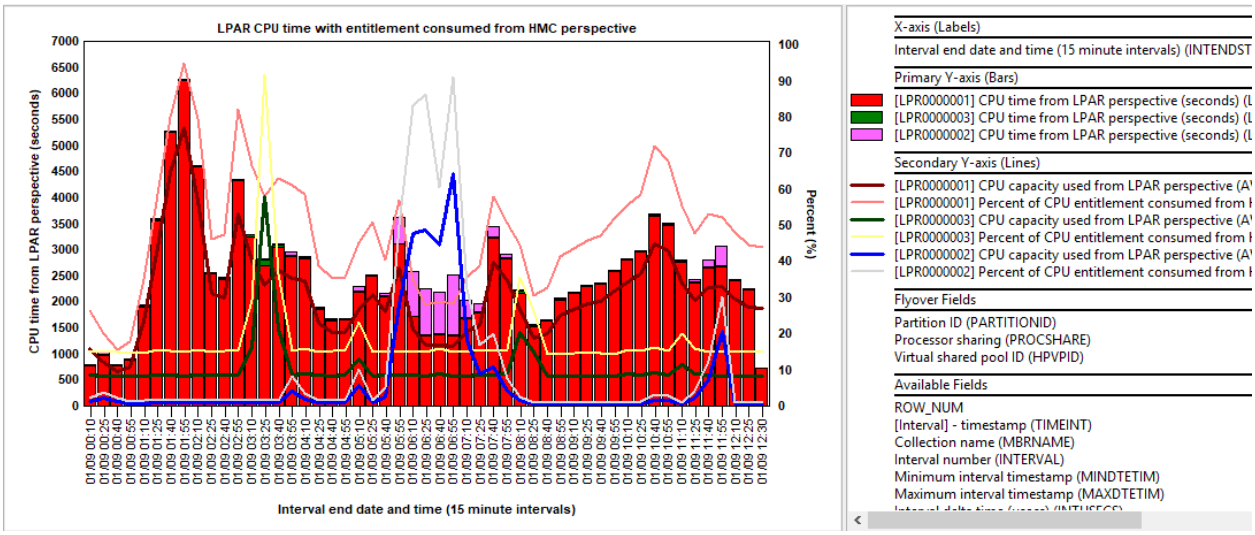
Tip: If you have a large number of LPARs and bars/colors, you can click the desired system/field in the legend to highlight a particular value. By default a maximum of 10 top LPAR contributors will be shown on the graph and an “Other contibutors” bucket will contain all the rest.



LPAR CPU time

9.16.4 LPAR CPU time with HMC CPU utilization

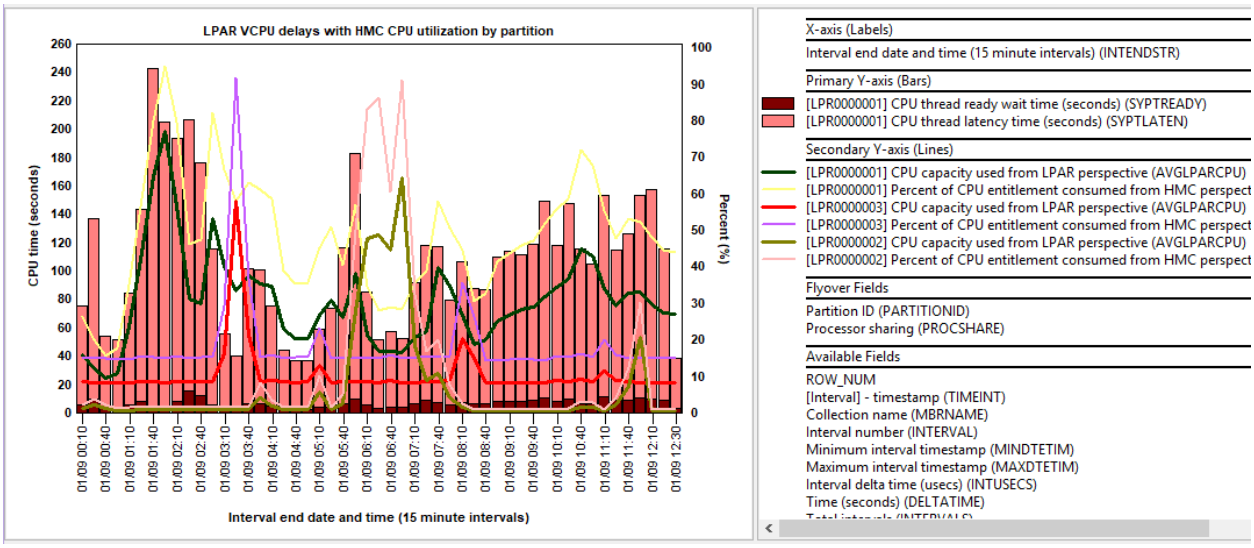
This graph is the same as the previous one but also includes the percent of CPU entitlement consumed from the HMC’s perspective (i.e. HMC CPU utilization) on the secondary Y-axis.



LPAR CPU time with HMC CPU utilization

9.16.5 LPAR VCPU delays with HMC CPU utilization

This graph displays virtual thread delays (thread ready wait time and latency time) that are known for each LPAR as well as CPU utilization for each LPAR from both HMC or LPAR perspectives.

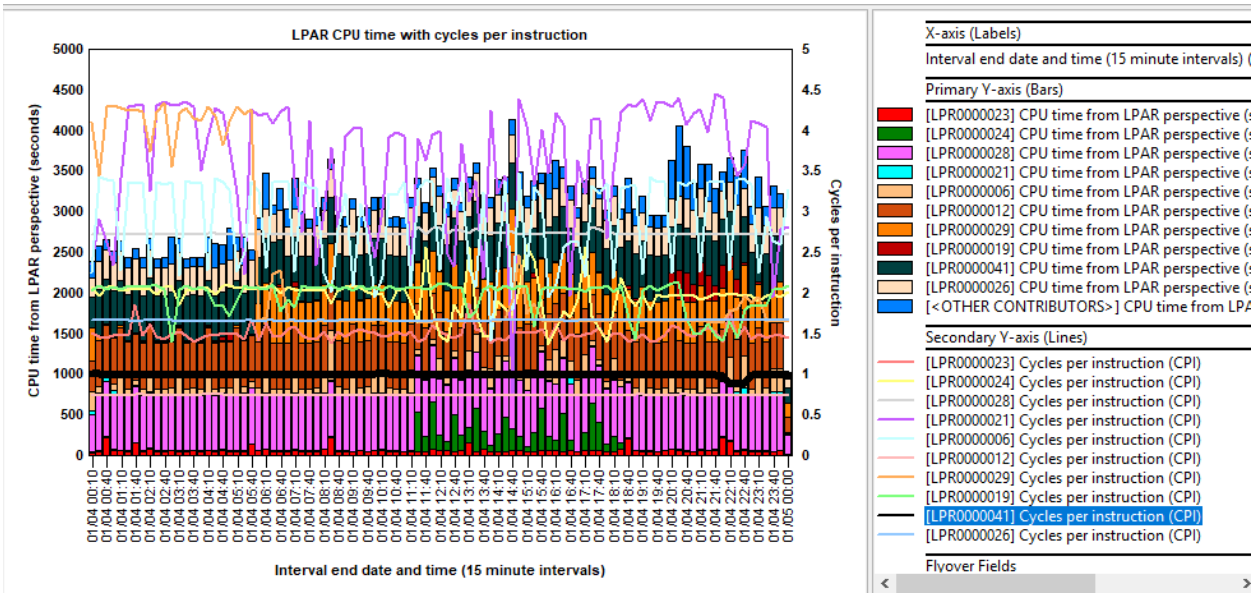


LPAR VCPU delays with HMC CPU utilization

9.16.6 LPAR CPU time with cycles per instruction

This graph is the same as the LPAR CPU time graph but also contains the cycles per instructions (CPI) on the secondary Y-axis.

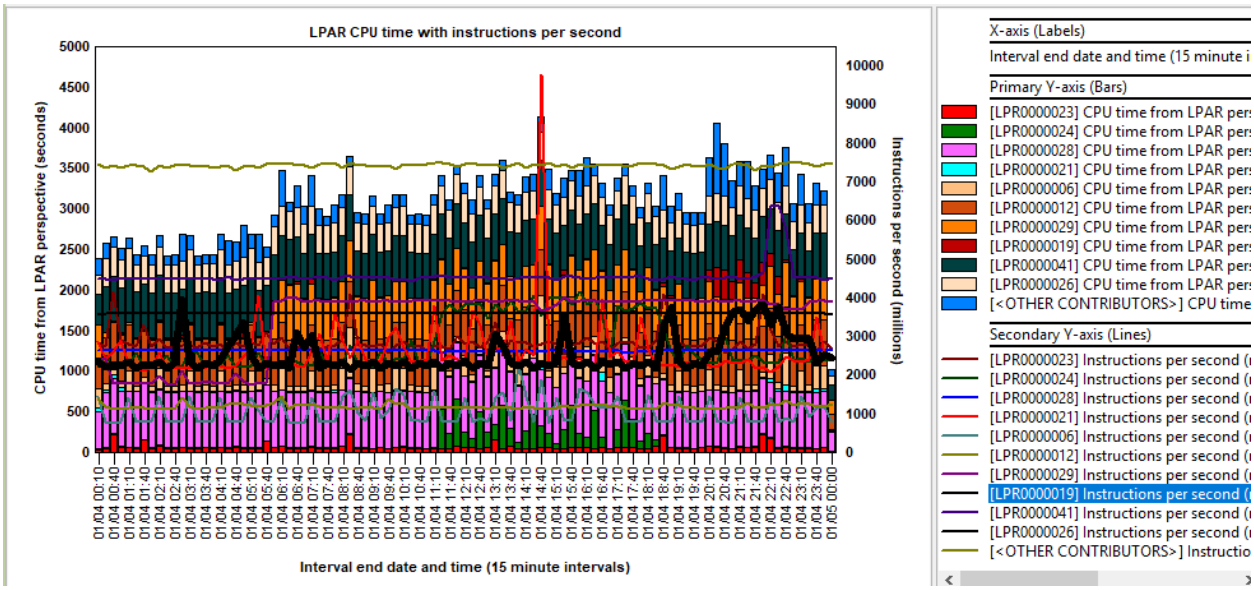
Note: CPI is available only for partitions running on Power6® and Power7® hardware. Otherwise the values are not available and will be 0.



LPAR CPU time with cycles per instruction

9.16.7 LPAR CPU time with instructions per second

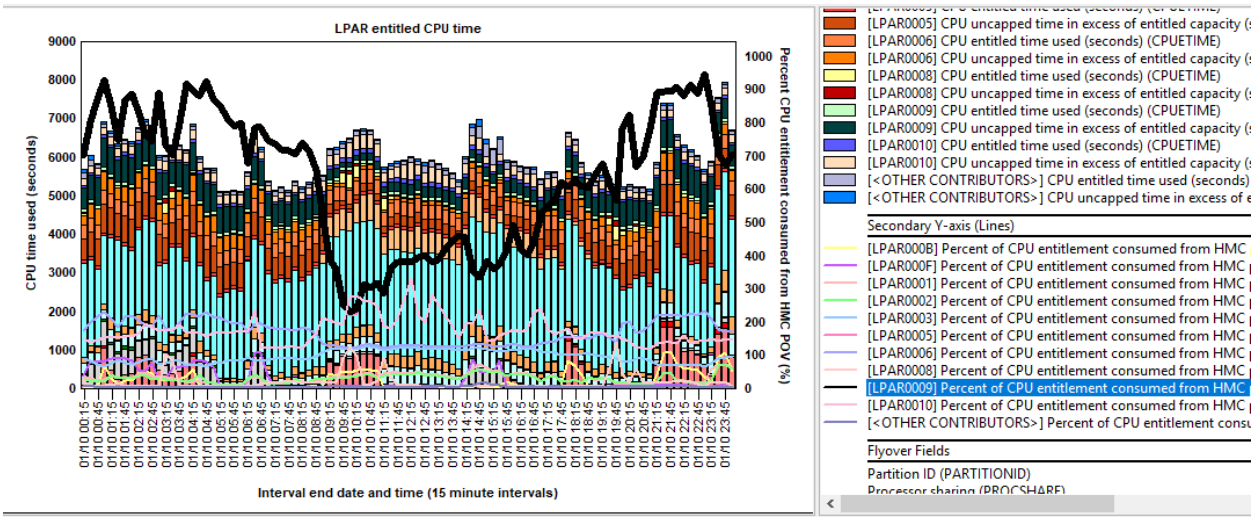
This graph is the same as the LPAR CPU time graph but also contains the instructions per second (IPS) on the secondary Y-axis.



LPAR CPU time with instructions per second

9.16.8 LPAR entitled CPU time

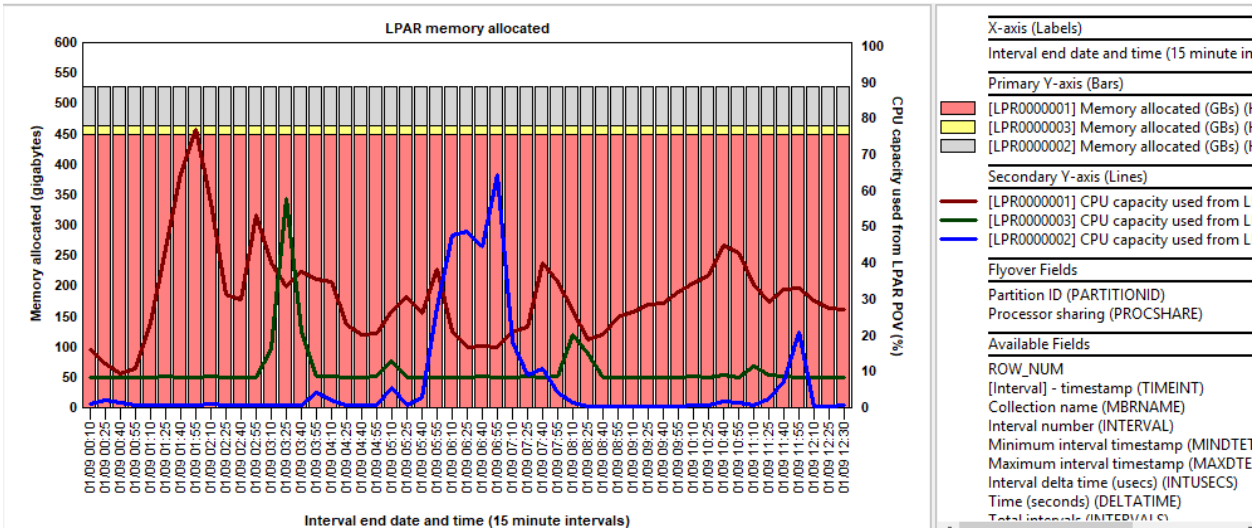
This graph shows entitled CPU time and uncapped time in excess of entitled capacity per LPAR on the primary Y-axis and the HMC CPU utilization on the secondary Y-axis.



LPAR entitled CPU time

9.16.9 LPAR memory allocated

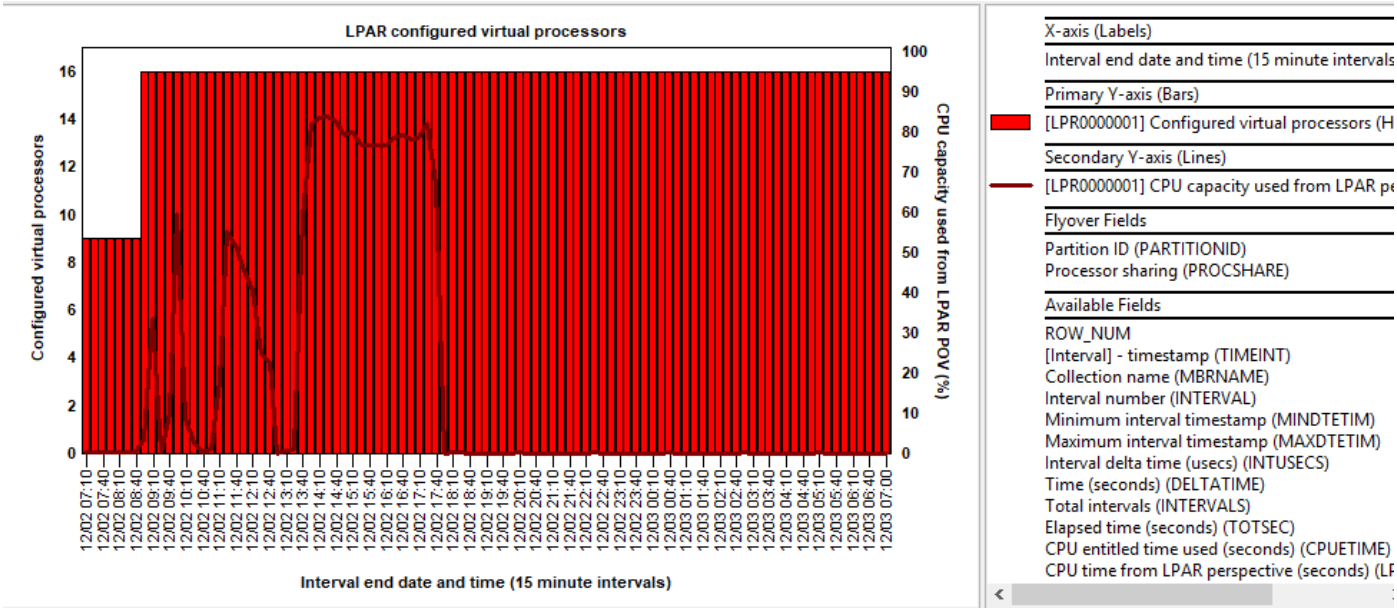
This graph shows the amount of memory allocated per partition (in GBs.) The secondary Y-axis shows the CPU utilization from the LPAR's perspective.



LPAR memory allocated

9.16.10 LPAR configured virtual processors

This graph displays the configured (not necessarily active) virtual processors for each LPAR over time. The secondary Y-axis displays CPU utilization from each LPAR's perspective.

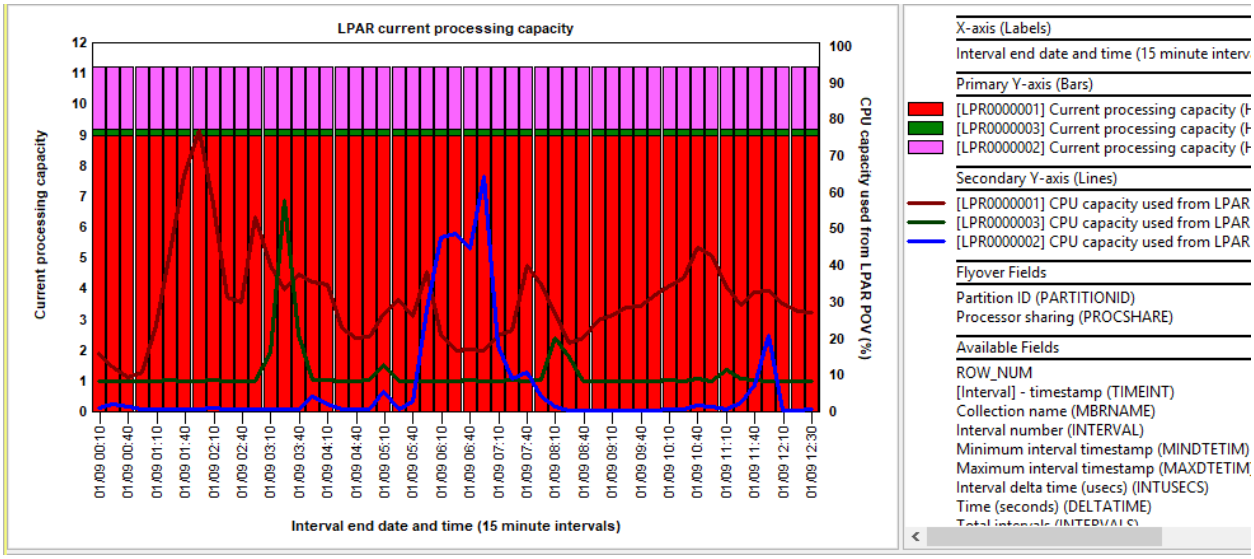


LPAR configured virtual processors

9.16.11 LPAR current processing capacity

This graph displays the current processing capacity for each LPAR over time. This is defined as the number of processor units currently allocated to the partition.

The secondary Y-axis displays CPU utilization from each LPAR's perspective.



LPAR current processing capacity

9.16.12 LPARS with capacity changes

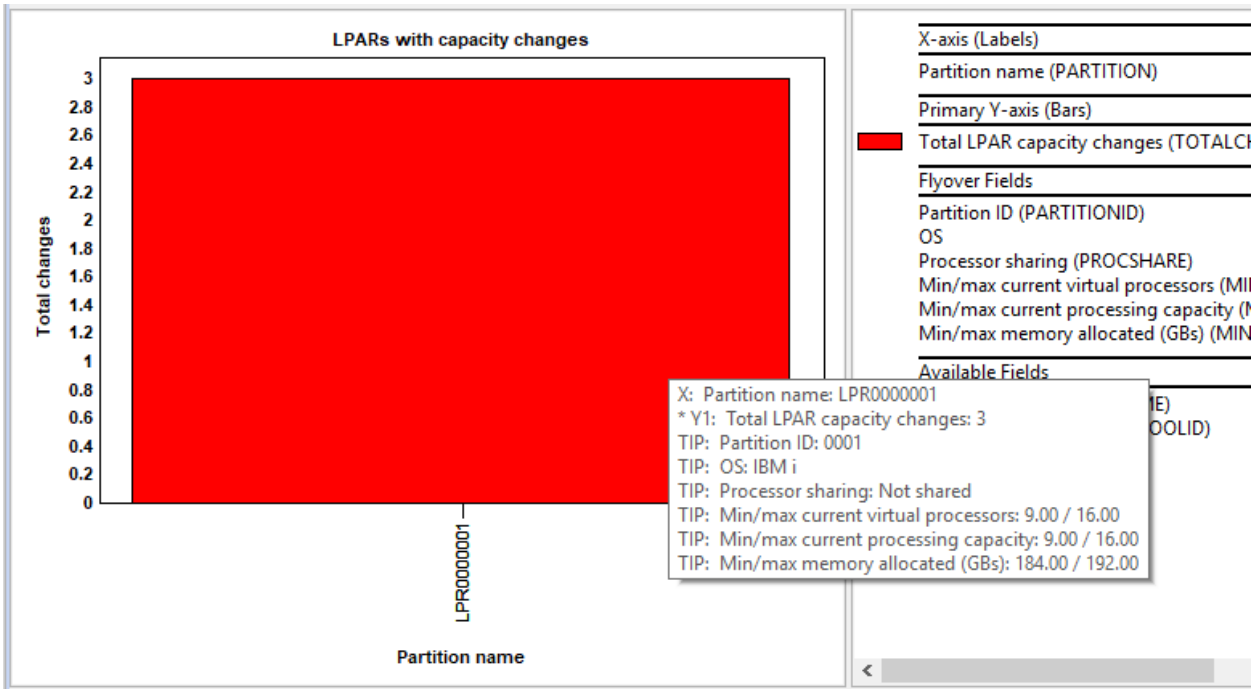
This graph checks for any changes to 3 configuration settings during the collection for all LPARs. These settings are:

1) Configured virtual processors
2) Current processing capacity
3) Memory allocated (GBs)

For each LPAR this graph will display the total configuration changes that occurred (max of 3.)

The graph flyover can be used to see the min and max values found in the data for each setting as well.

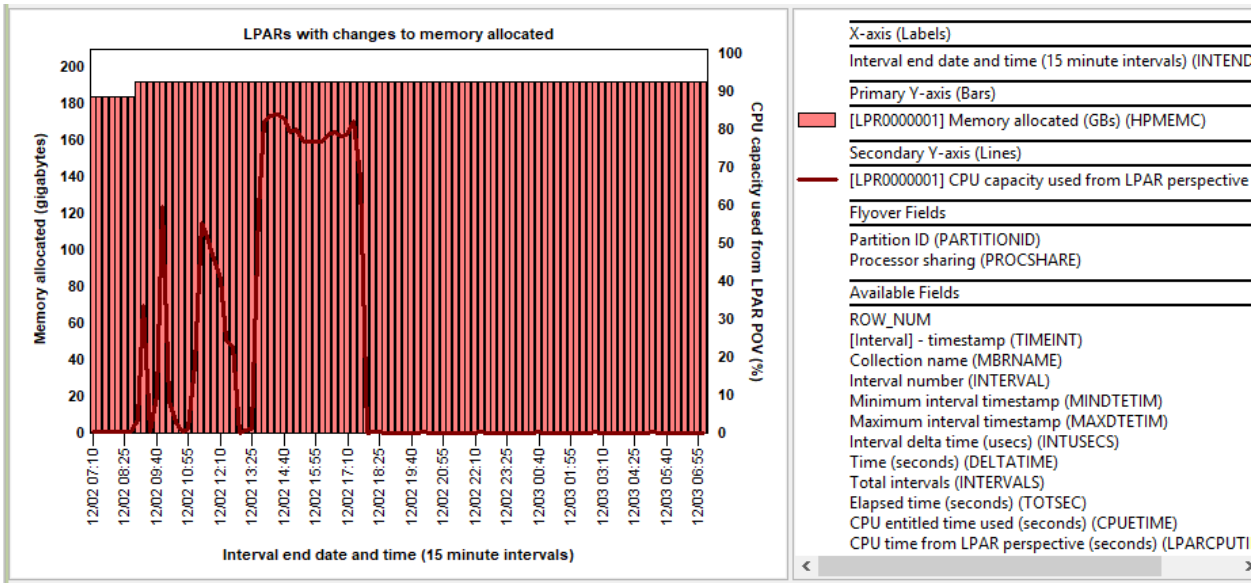
Note: This graph will typically contain no data because configuration changes are rare.



LPARS with capacity changes

9.16.13 LPARS with changes to memory allocated

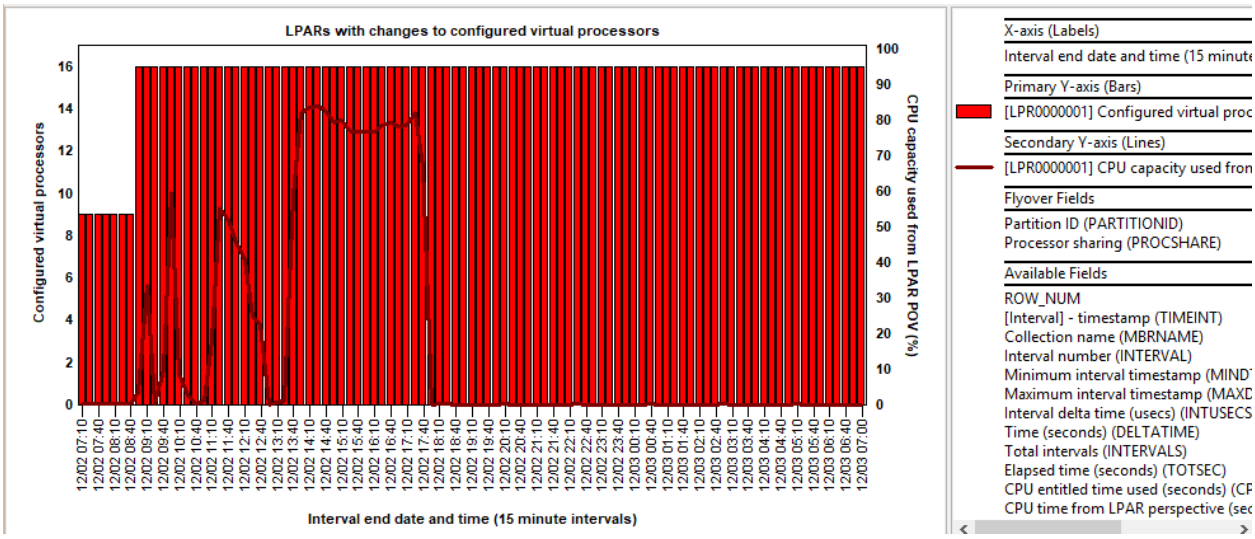
This graph will display the memory allocated but only for LPARs that had changes to that setting during the collection.



LPARS with changes to memory allocated

9.16.14 LPARS with changes to configured virtual processors

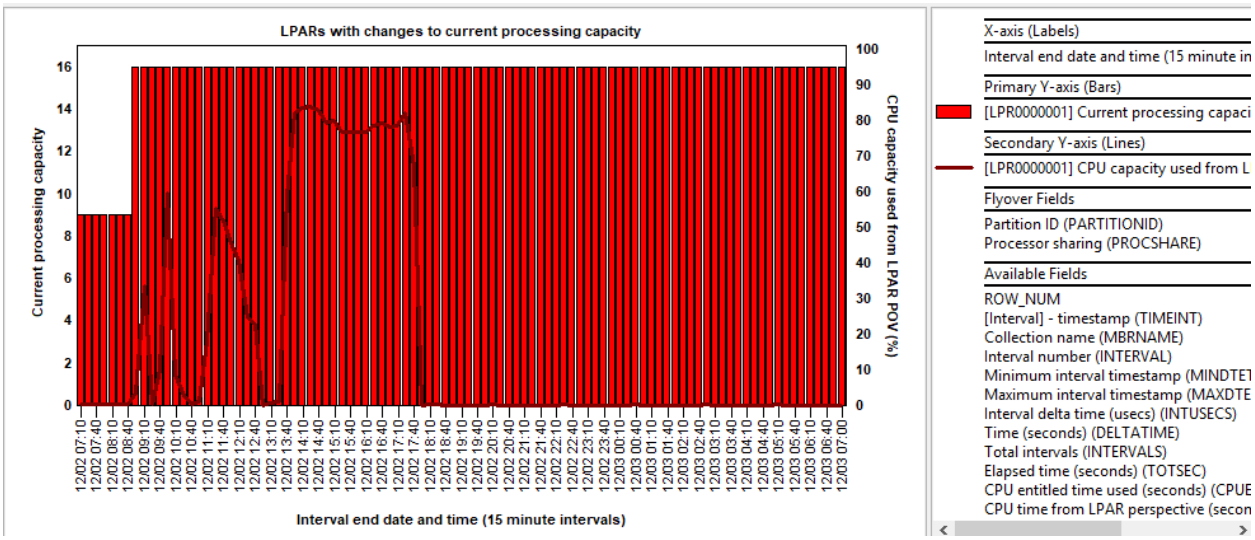
This graph will display the configured virtual processors but only for LPARs that had changes to that setting during the collection.



LPARS with changes to configured virtual processors

9.16.15 LPARS with changes to current processing capacity

This graph will display the current processing capacity but only for LPARs that had changes to that setting during the collection.



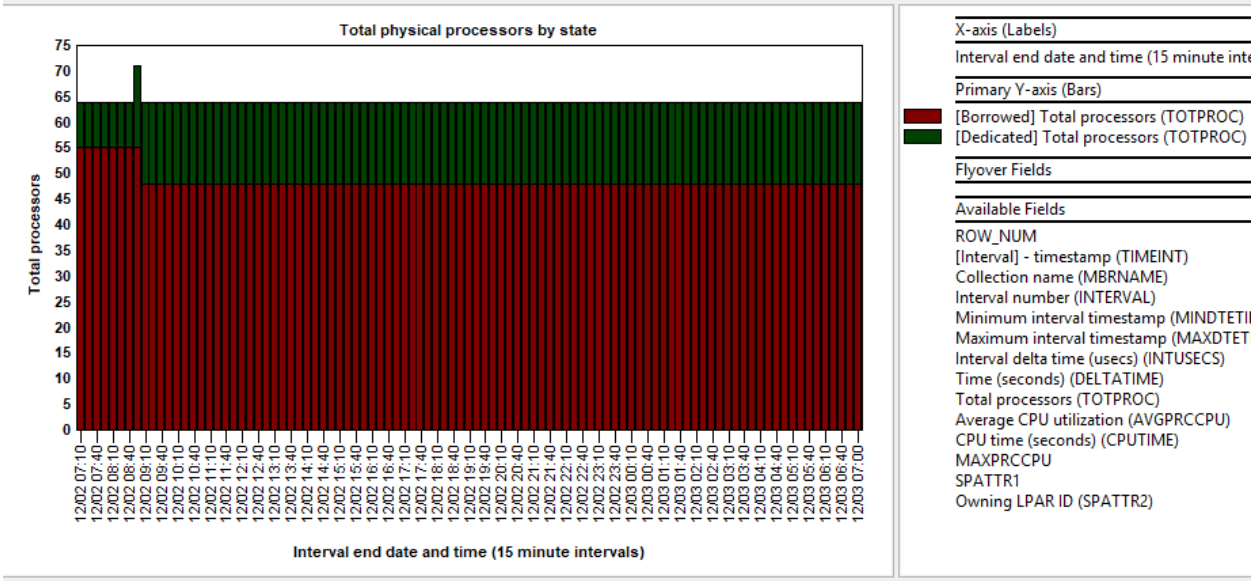
LPARS with changes to current processing capacity

9.16.16 Total physical processors by state

This graph displays the total processors by state on the physical system. Each processor state will show a different color and its description will be listed in the graph legend.

The processor states are:

- '1' = **Not installed**
- '2' = **Guarded off** (This processor has some runtime recoverable errors or it has been check stopped).
- '3' = **Unlicensed** (This processor is installed, but the license has not been purchased).
- '4' = **Shared**
- '5' = **Borrowed** (If this processor is not assigned to any partition, it has been taken from a dedicated partition that is set to donate its unused processors).
- '6' = **Dedicated**

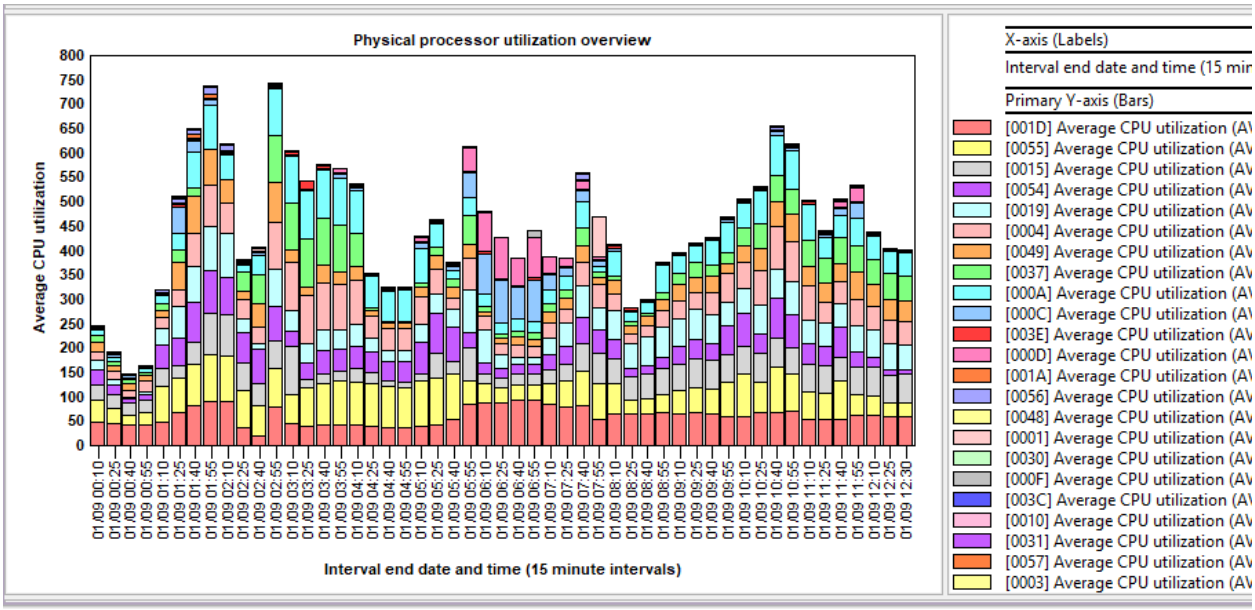


Total physical processors by state

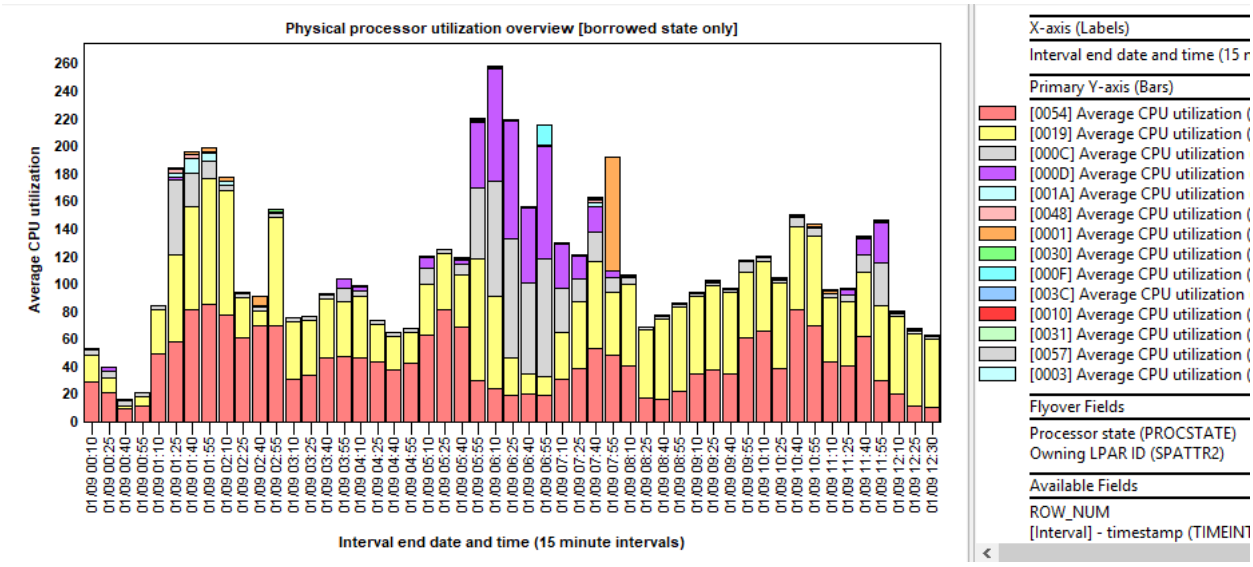
9.16.17 Physical processor utilization overview graphs

These graph shows the CPU utilization for different types of physical processors depending on the processor state shown in brackets next to the graph name.

Some examples follow:



Physical processor utilization overview

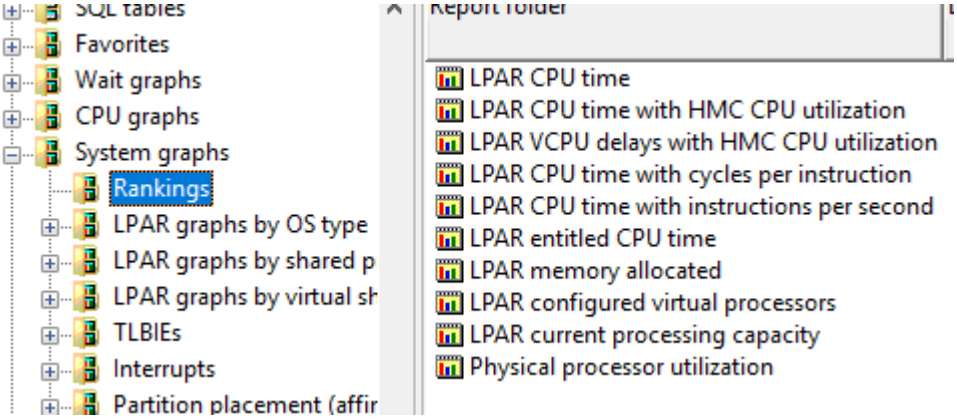


Physical processor utilization overview [borrowed state only]

9.16.18 Rankings

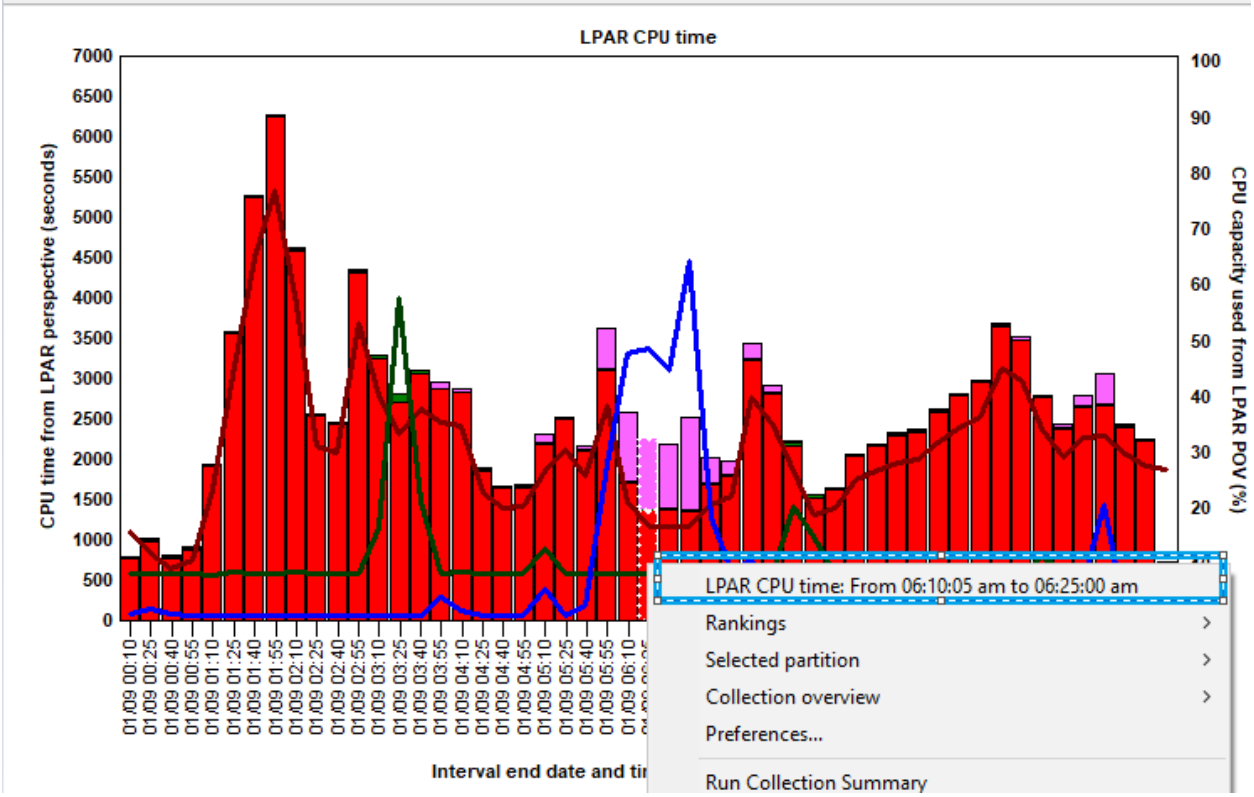
These graphs are like the graphs in the previous section but instead of showing data over time, the partitions (or processors) are ranked.

These graphs can be accessed either as a drill down from the previous graphs or over the entire collection.



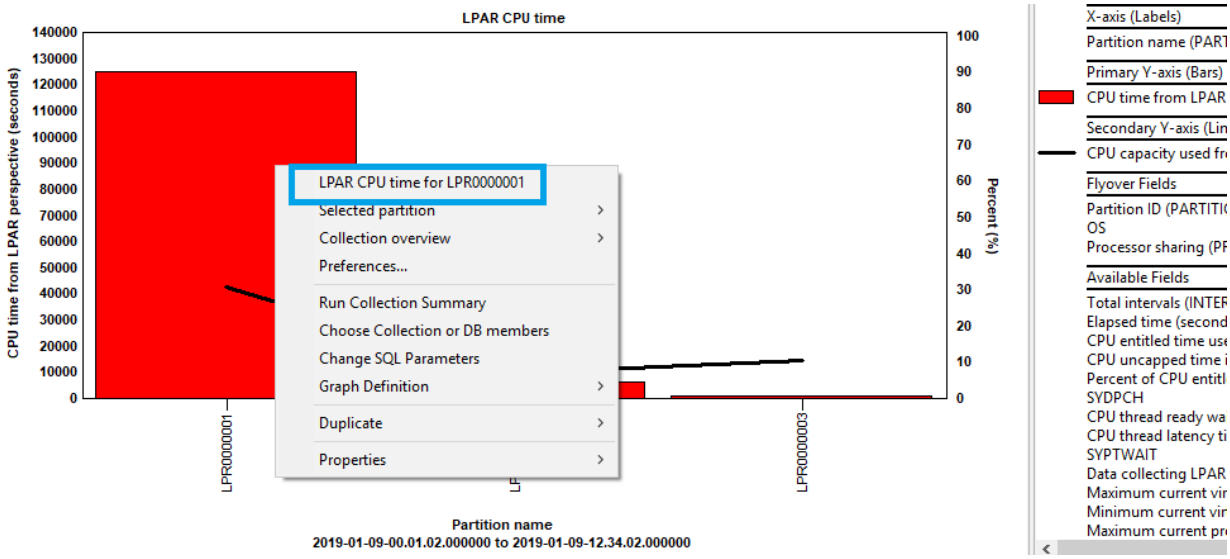
System graphs -> Rankings

Tip #1: From one of the System (overview) graphs, select a time period and right-click to drill into a graph showing LPARs ranked for the desired time period.



Example (Tip #1)

Tip #2: From one of the rankings graphs, select a partition and right-click to graph that LPAR over time.

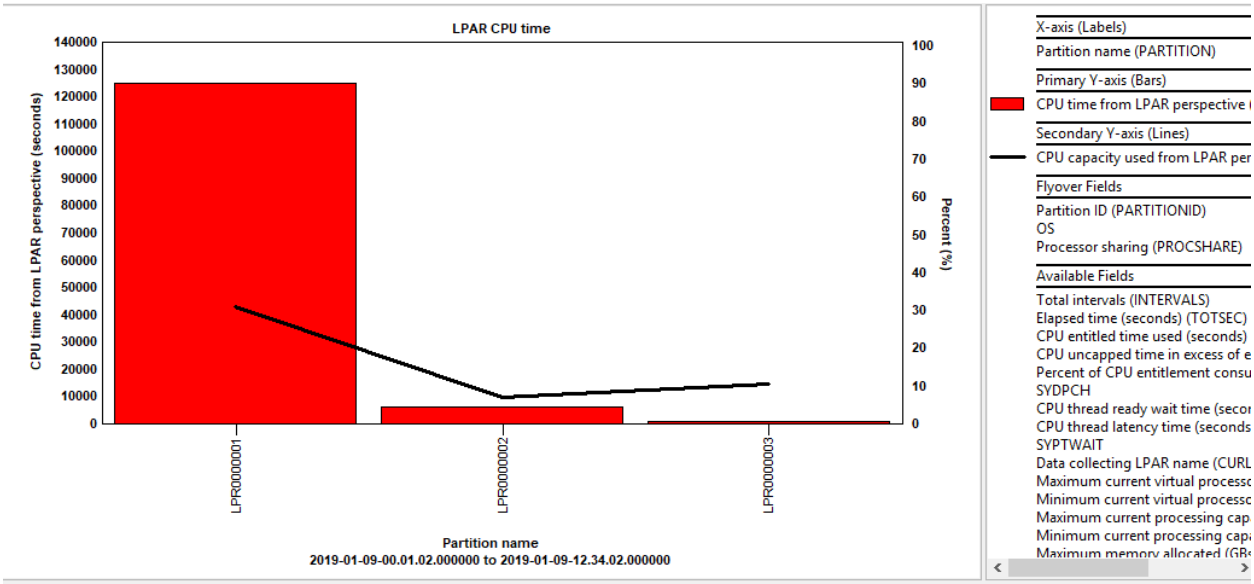


Example (Tip #2)

9.16.18.1 LPAR CPU time

This graph shows each partition's total CPU time and average CPU utilization (from the LPAR's perspective) during the collection.

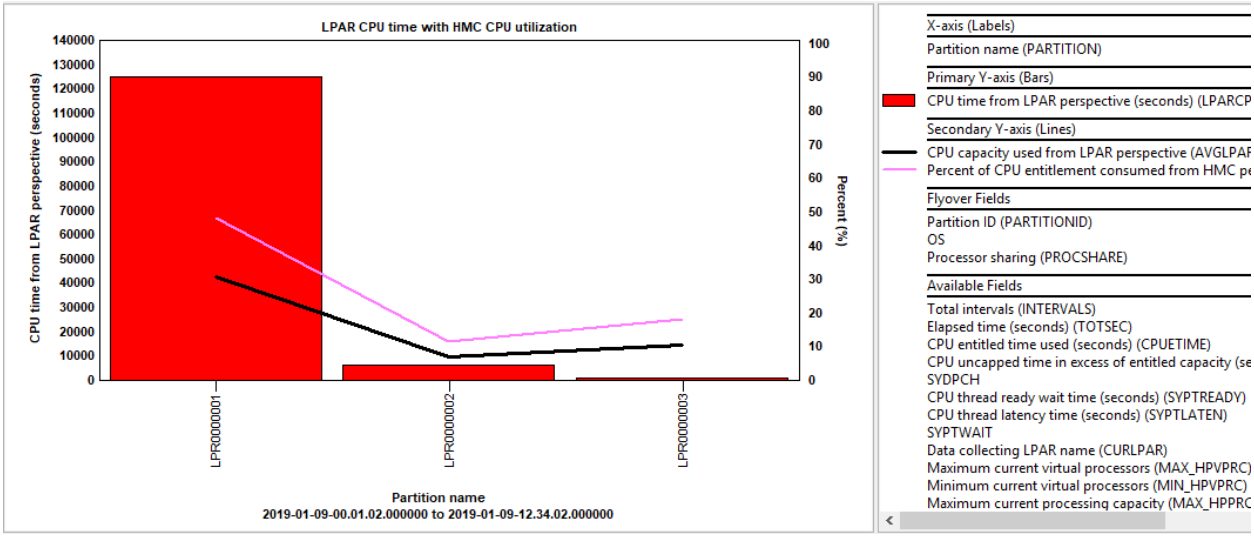
Tip: If you don't wish to see vertical bars (for readability purposes), click the toggle graph format button to switch it to horizontal bars. The Y2-axis will be removed, however.



LPAR CPU time

9.16.18.2 LPAR CPU time with HMC CPU utilization

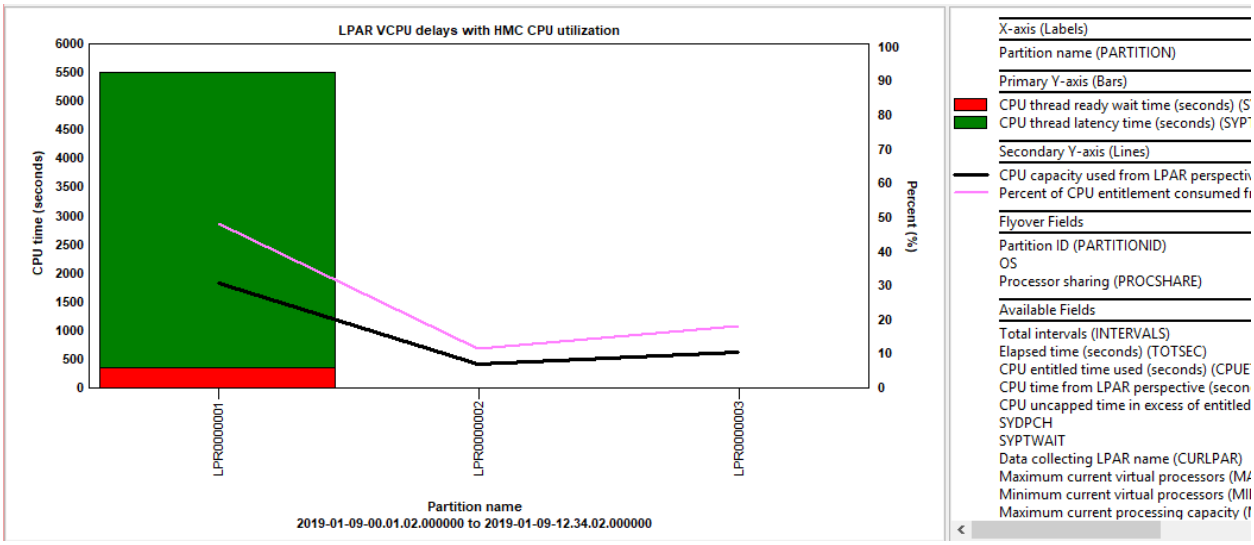
This graph is the same as the previous one but also includes average CPU utilization from the HMC's perspective for comparison purposes with the CPU utilization from the LPAR's perspective.



LPAR CPU time with HMC CPU utilization

9.16.18.3 LPAR VCPU delays with HMC CPU utilization

This graph displays virtual thread delays (thread ready wait time and latency time) that are known for each LPAR as well as aveage CPU utilization for each LPAR from both HMC or LPAR perspectives.

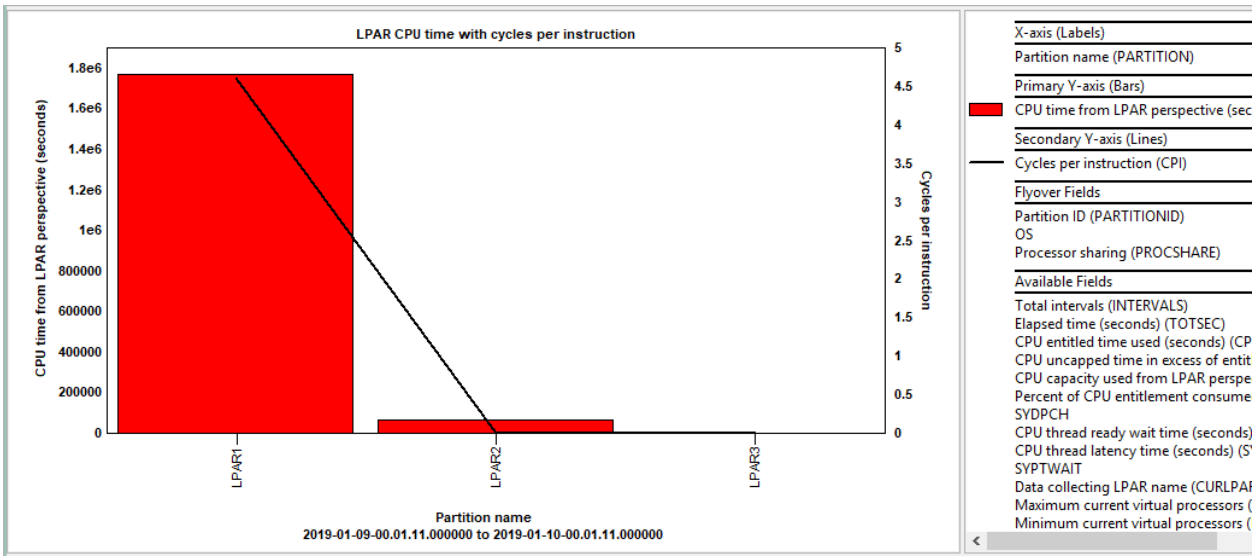


LPAR VCPU delays with HMC CPU utilization

9.16.18.4 LPAR CPU time with cycles per instruction

This graph is the same as the LPAR CPU time graph but also contains the cycles per instructions (CPI) on the secondary Y-axis.

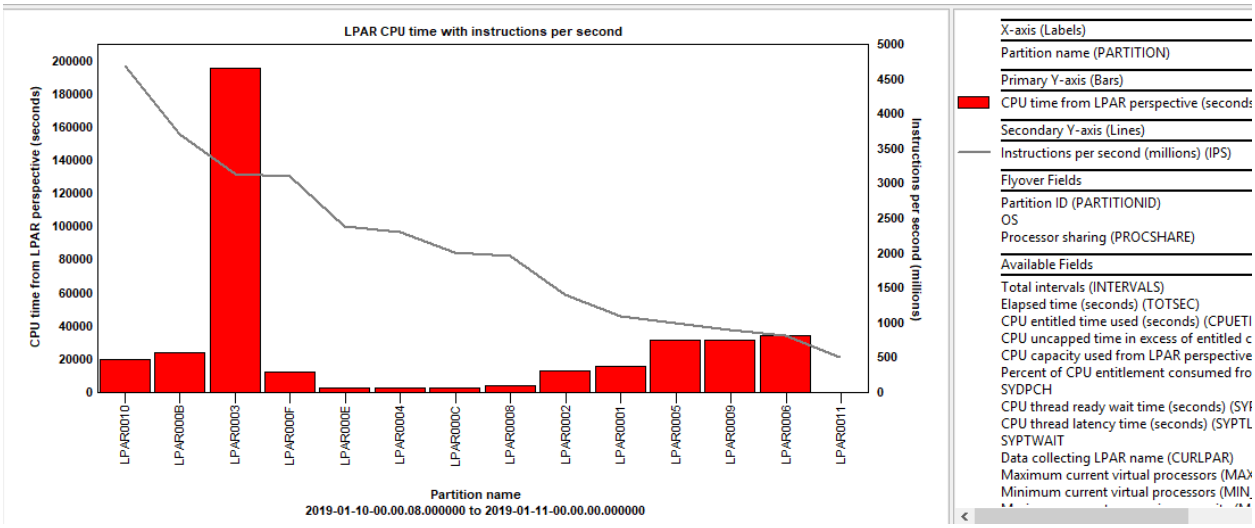
Note: CPI is available only for partitions running on Power6® and Power7® hardware. Otherwise the values are not available and will be 0.



LPAR CPU time with cycles per instruction

9.16.18.5 LPAR CPU time with instructions per second

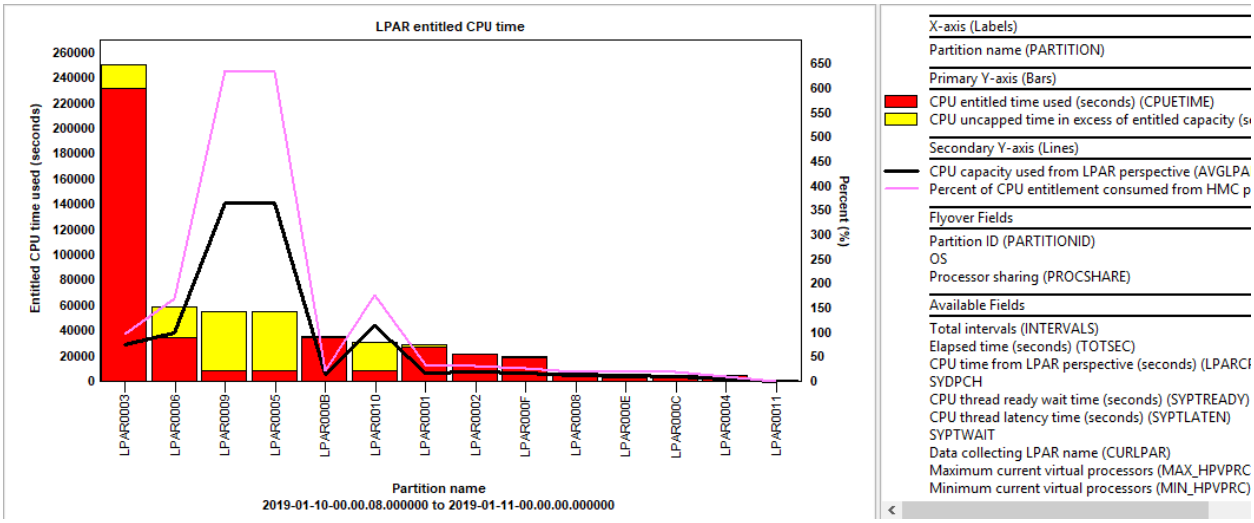
This graph is the same as the LPAR CPU time graph but also contains the instructions per second (IPS) on the secondary Y-axis.



LPAR CPU time with instructions per second

9.16.18.6 LPAR entitled CPU time

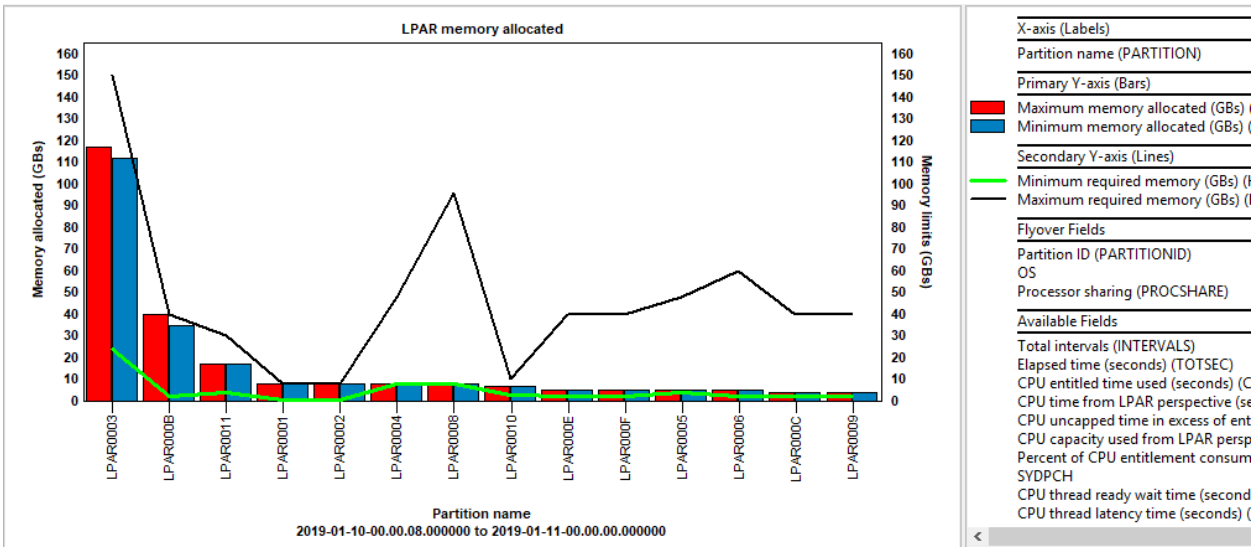
This graph shows entitled CPU time and uncapped time in excess of entitled capacity per LPAR on the primary Y-axis and the HMC CPU utilization on the secondary Y-axis.



LPAR entitled CPU time

9.16.18.7 LPAR memory allocated

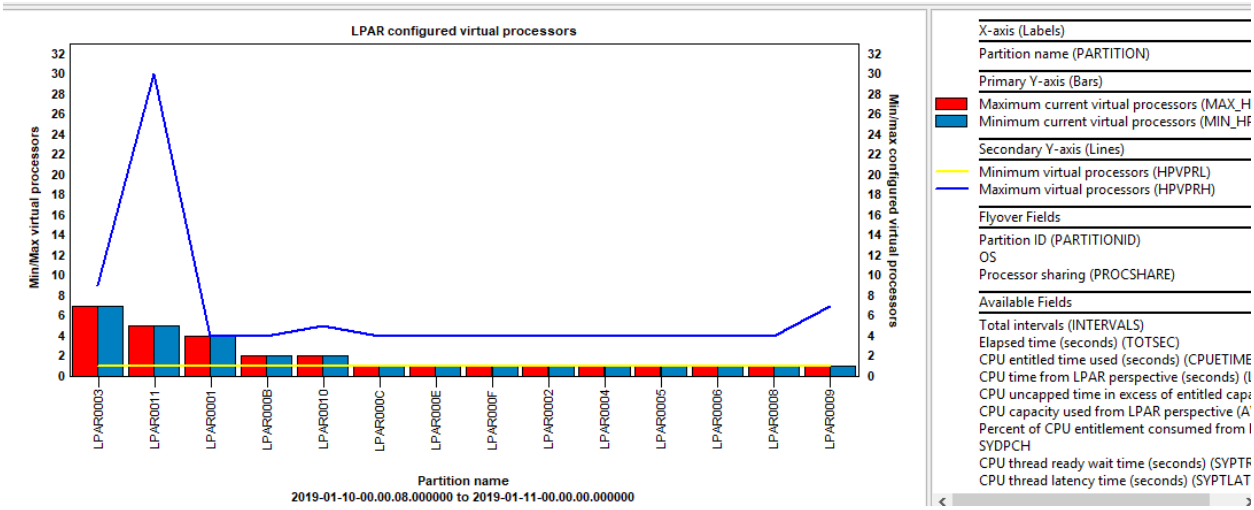
The graph shows the min/max memory allocated as well as the min/max memory required. All values shown are in gigabytes (GBs).



LPAR memory allocated

9.16.18.8 LPAR configured virtual processors

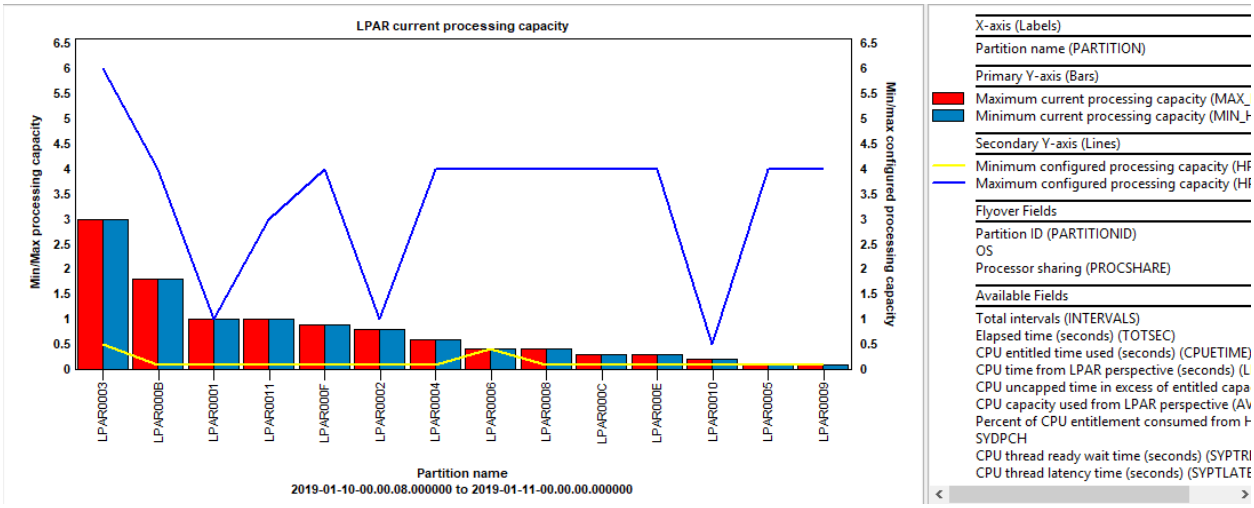
This graph ranks the configured (not necessarily active) virtual processors for each LPAR. The secondary Y-axis displays CPU utilization from each LPAR's perspective.



LPAR configured virtual processors

9.16.18.9 LPAR current processing capacity

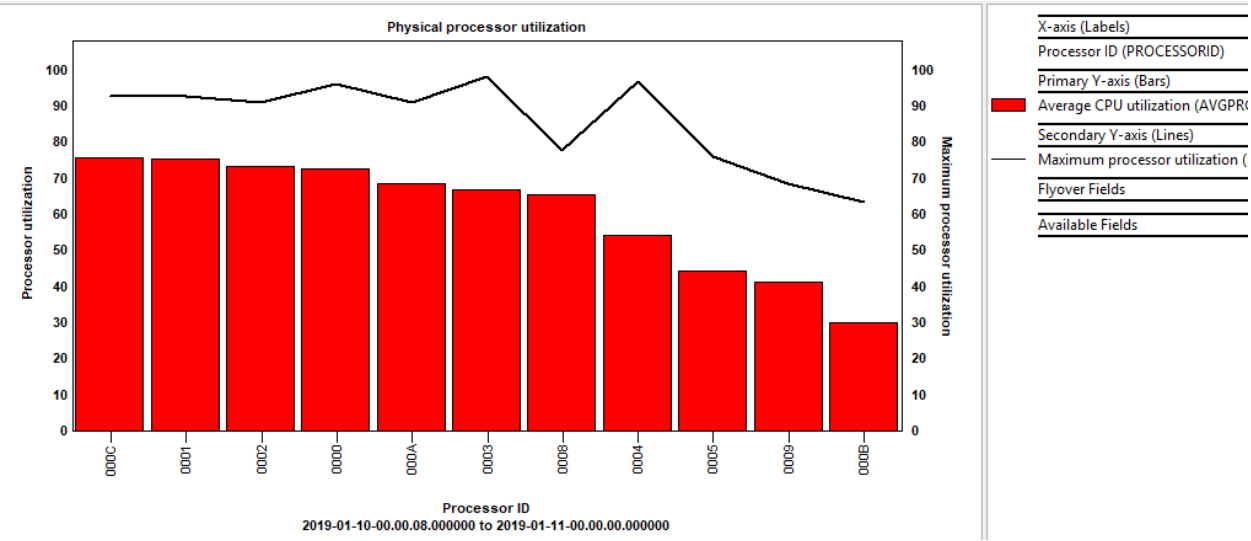
This graph ranks the min/max current processing capacity for each LPAR. This is defined as the number of processor units currently allocated to the partition. The min and max configured values are also shown on the secondary Y-axis.



LPAR current processing capacity

9.16.18.10 Physical processor utilization

This graph ranks each processor by the average CPU utilization. The processor ID is shown for each. The 2nd Y-axis displays the maximum CPU utilization for each processor.



Physical processor utilization

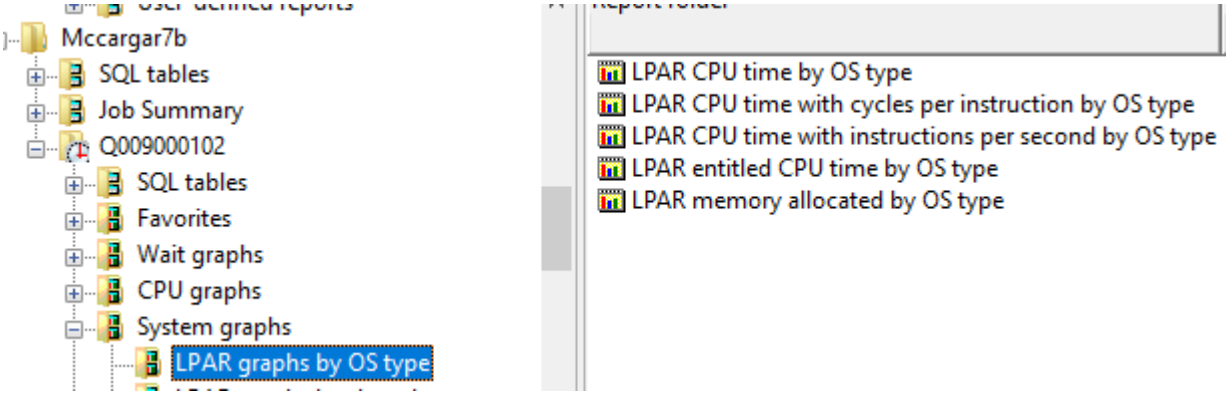
9.16.19 LPAR graphs by OS type

This folder contains a set of graphs like those under the Systems graphs folder but flattened by the type of system.

The possible OS types are:

- IBM i
- Other OS
- Unknown

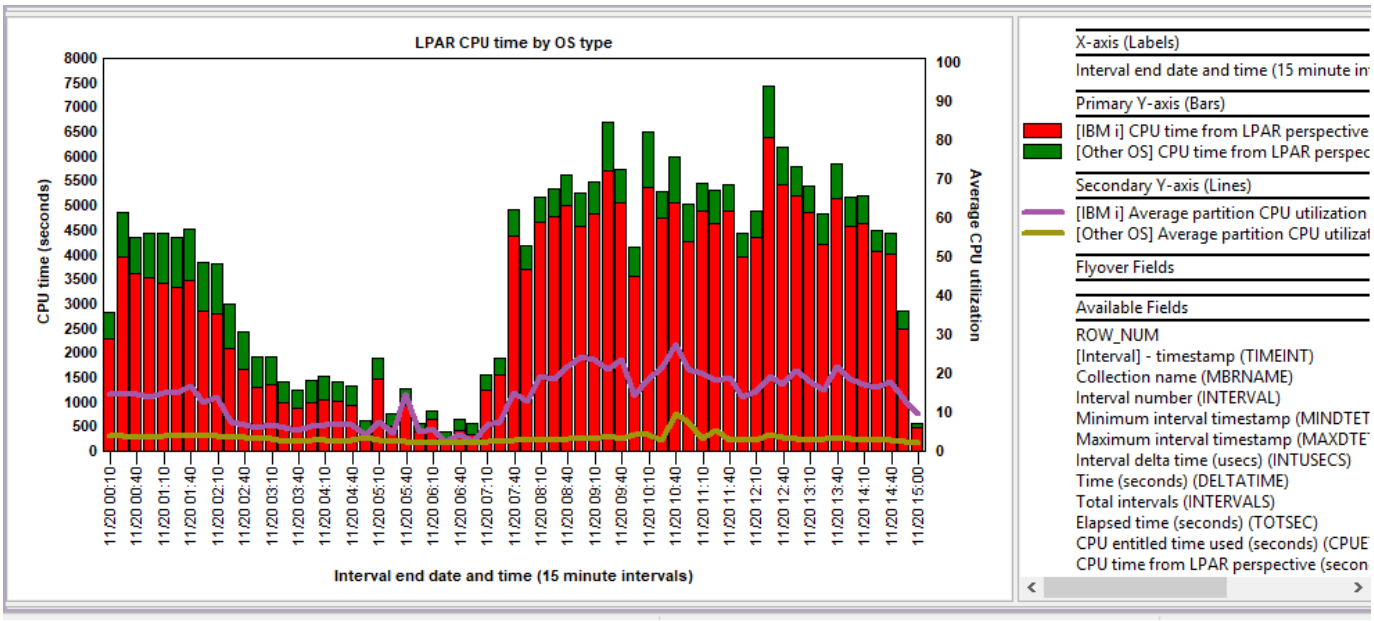
Note: Typically, the unknown type is only rare shown.



System graphs -> LPAR graphs by OS type

9.16.19.1 LPAR CPU time by OS type

This graph displays total CPU time consumed from all LPARs of each OS type on the primary Y-axis. The secondary Y-axis contains the average CPU utilization for each LPAR.

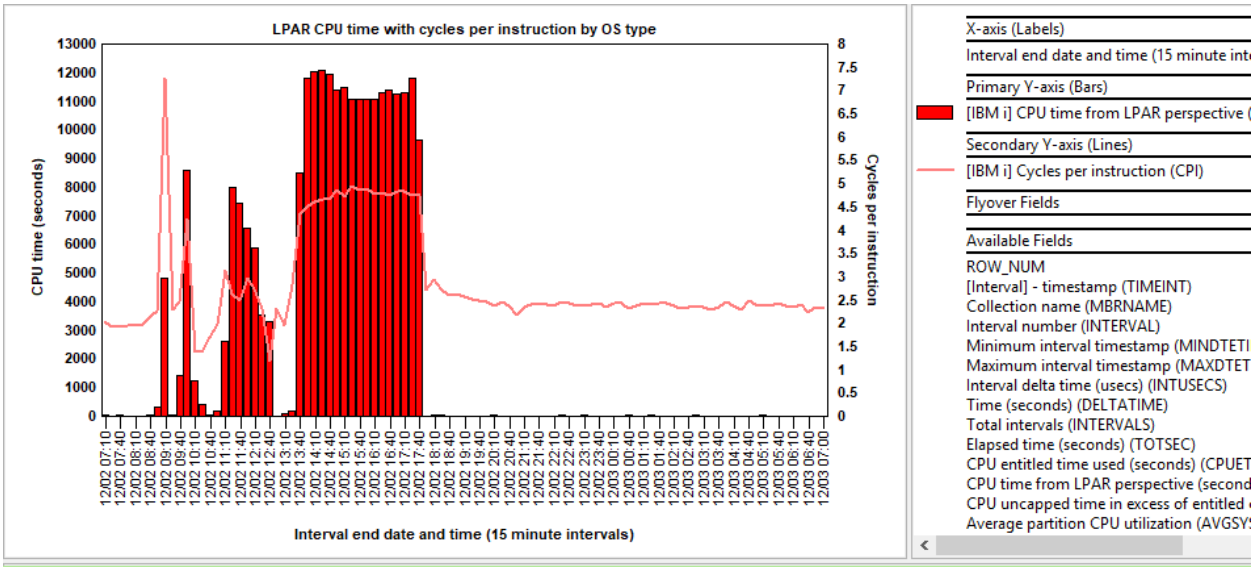


LPAR CPU time by OS type

9.16.19.2 LPAR CPU time with cycles per instruction by OS type

This graph is the same as the previous one except includes average CPI on the secondary Y-axis per OS type.

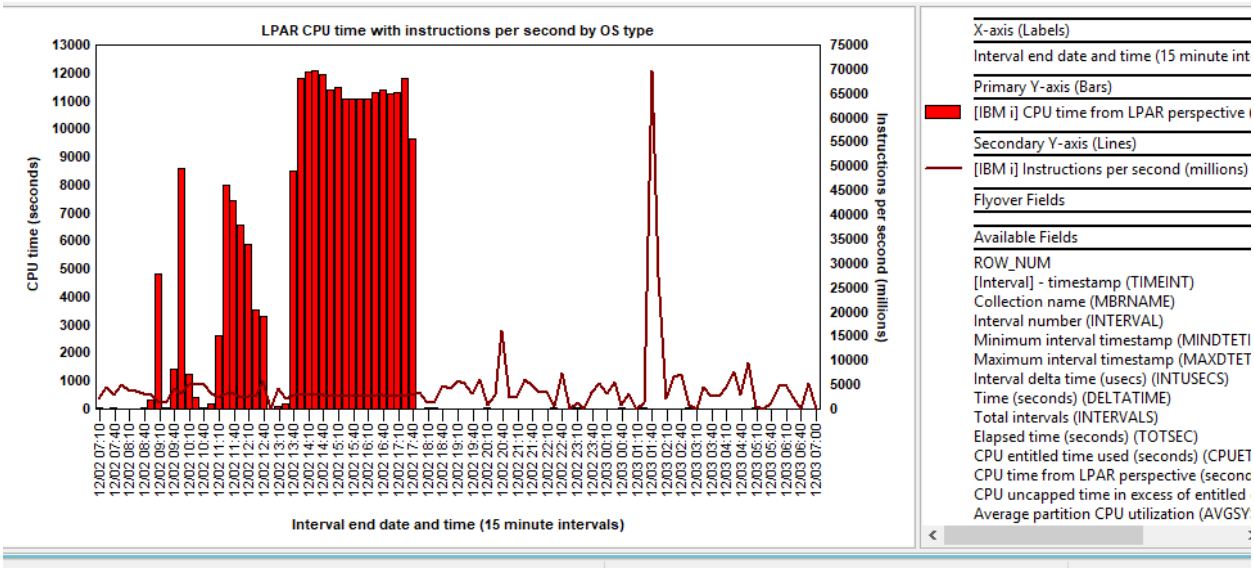
Note: CPI is available only for partitions running on Power6® and Power7® hardware. Otherwise the values are not available and will be 0.



LPAR CPU time with cycles per instruction by OS type

9.16.19.3 LPAR CPU time with instructions per second by OS type

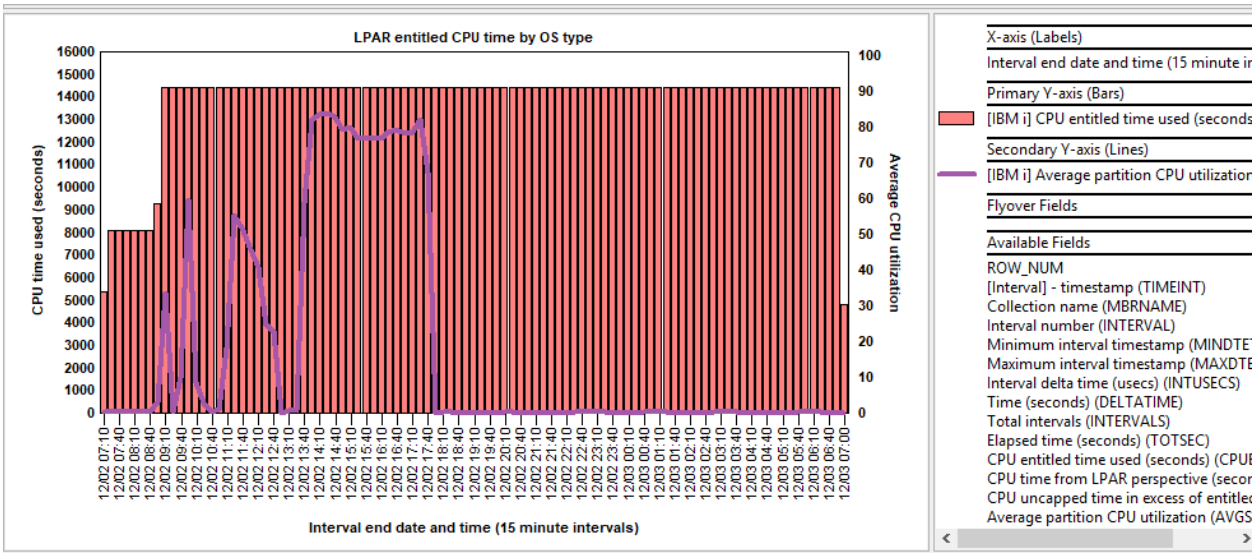
This graph is the same as the previous once except includes average instructions per second on the secondary Y-axis by OS type.



LPAR CPU time with instructions per second by OS type

9.16.19.4 LPAR entitled CPU time by OS type

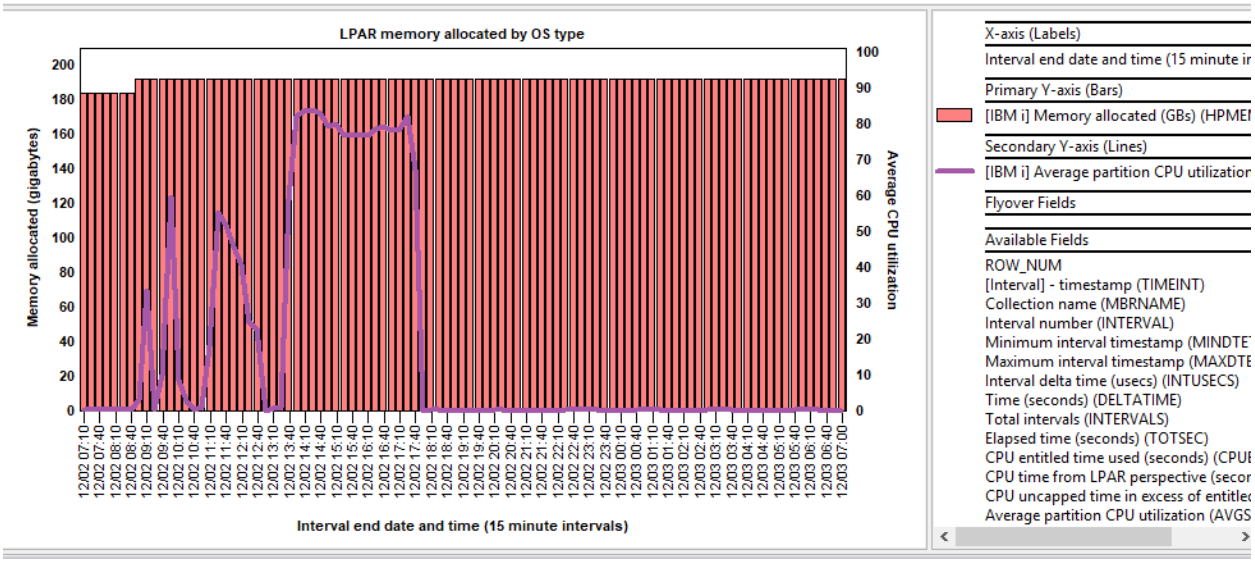
This graph shows the total CPU entitled time used per OS type and uncapped time in excess of capacity on the primary Y-axis and the average CPU utilization on the secondary Y-axis.



LPAR entitled CPU time by OS type

9.16.19.5 LPAR memory allocated by OS type

This graph shows the total memory allocated for all LPARs of each OS type on the primary Y-axis and the average CPU utilization per OS type on the secondary Y-axis.



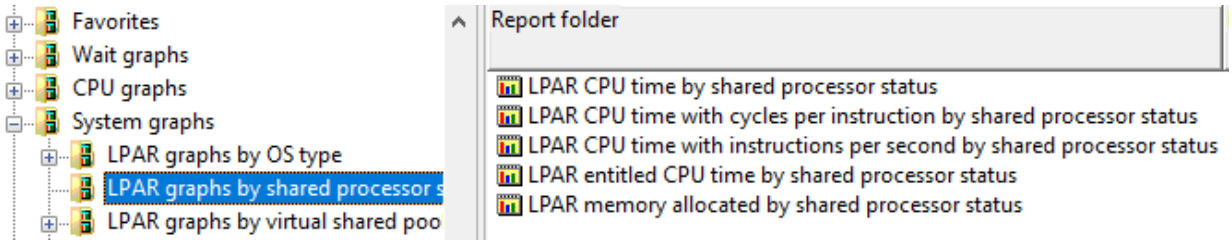
LPAR memory allocated by OS type

9.16.20 LPAR graphs by shared processor status

This folder contains a set of graphs like those under the Systems graphs folder but flattened by shared processor status.

The possible statuses for the processors are:

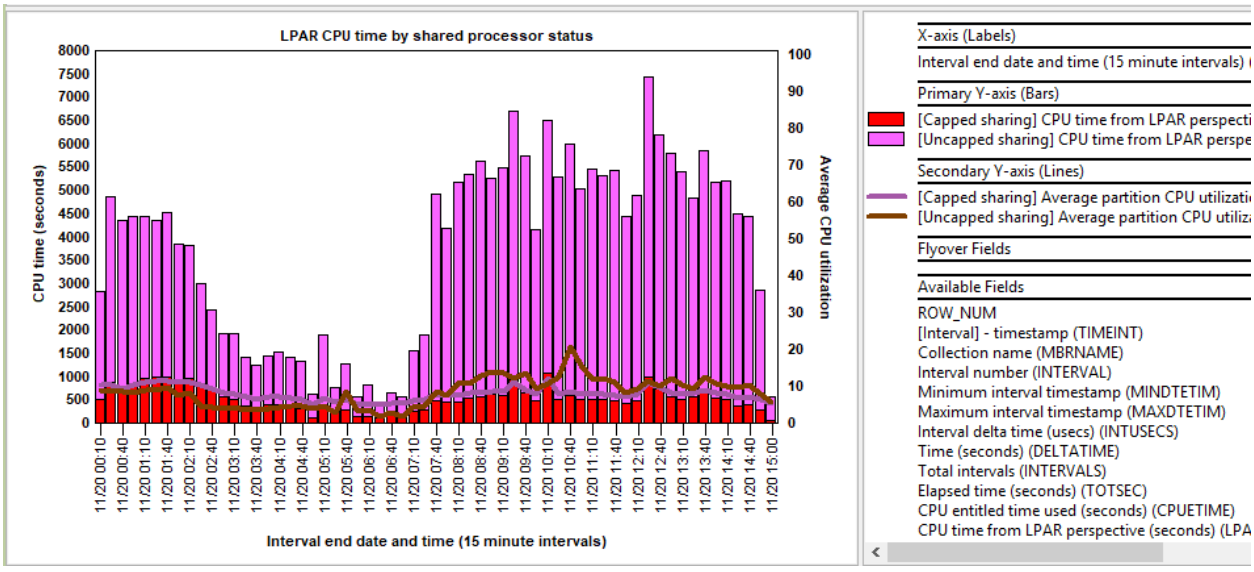
- Not shared
- Capped sharing
- Uncapped sharing



System graphs -> LPAR graphs by shared processor status

9.16.20.1 LPAR CPU time by shared processor status

This graph displays total CPU time consumed from all LPARs of each processor status on the primary Y-axis. The secondary Y-axis contains the average CPU utilization for each LPAR.

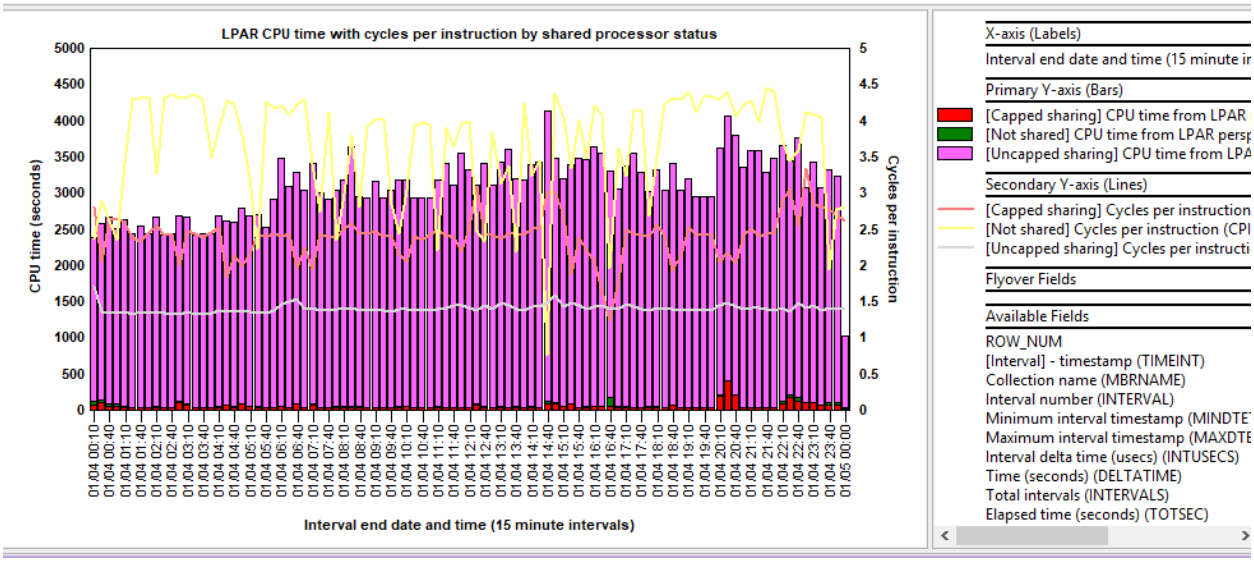


LPAR CPU time by shared processor status

9.16.20.2 LPAR CPU time with cycles per instruction by shared processor status

This graph is the same as the previous one except includes average CPI on the secondary Y-axis per processor status.

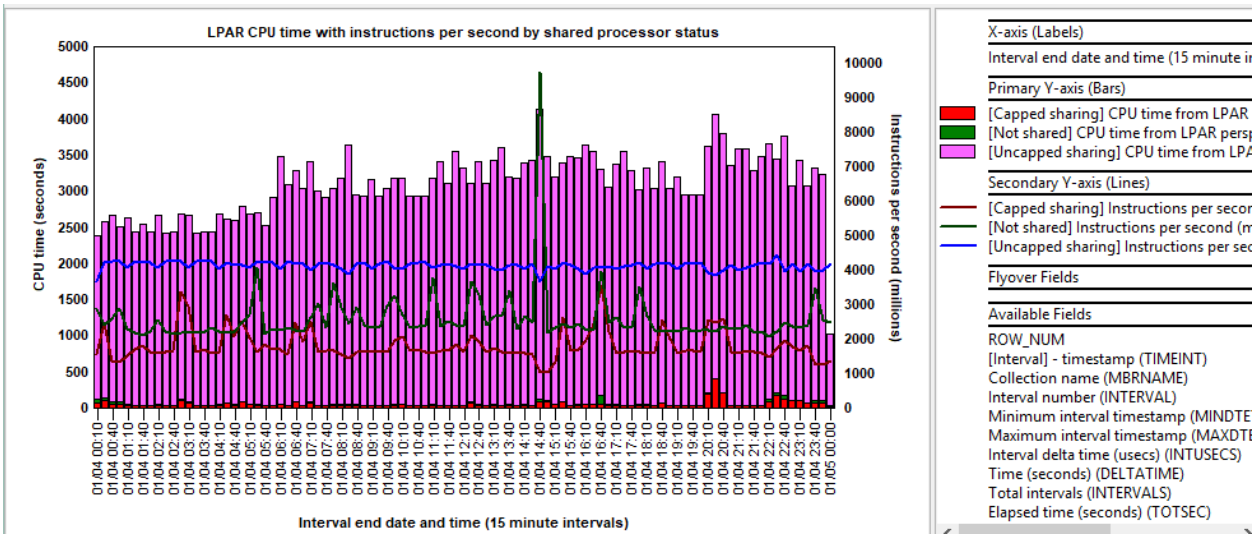
Note: CPI is available only for partitions running on Power6® and Power7® hardware. Otherwise the values are not available and will be 0.



LPAR CPU time with cycles per instruction by shared processor status

9.16.20.3 LPAR CPU time with instructions per second by shared processor status

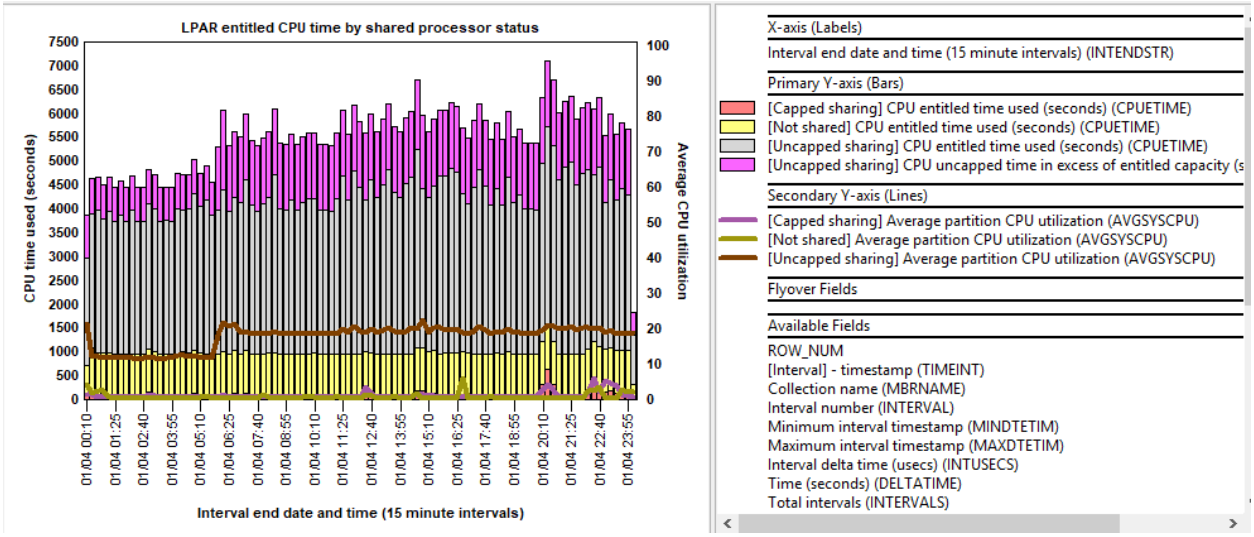
This graph is the same as the previous one except includes average instructions per second on the secondary Y-axis per processor status.



LPAR CPU time with instructions per second by shared processor status

9.16.20.4 LPAR entitled CPU time by shared processor status

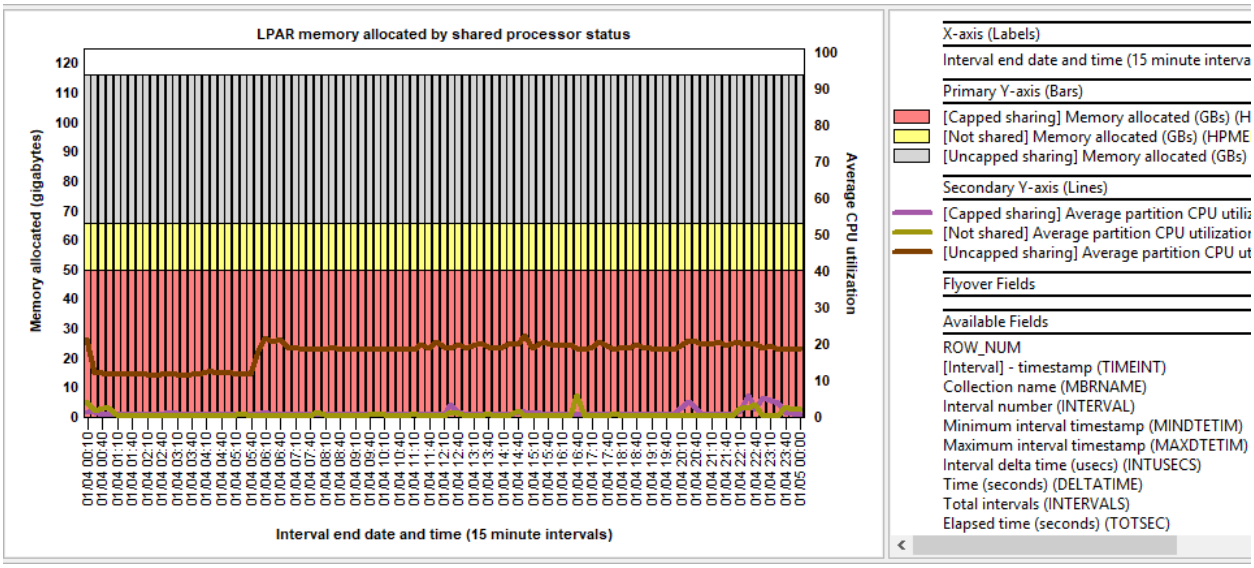
This graph shows (per processor status) the total CPU entitled time used and uncapped time in excess of capacity on the primary Y-axis and the average CPU utilization on the secondary Y-axis.



LPAR entitled CPU time by shared processor status

9.16.20.5 LPAR memory allocated by shared processor status

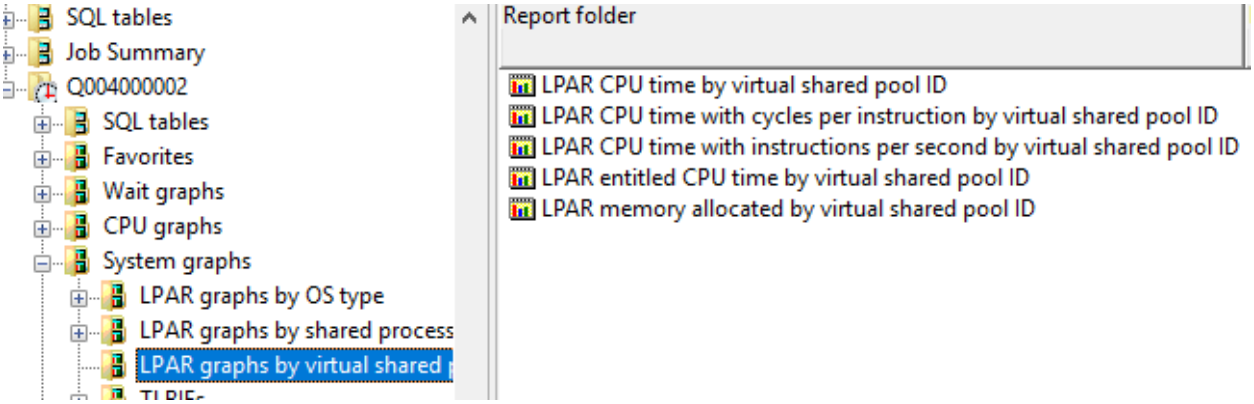
This graph shows the total memory allocated for all LPARs of each processor status on the primary Y-axis and the average CPU utilization per status on the secondary Y-axis.



LPAR memory allocated by shared processor status

9.16.21 LPAR graphs by virtual shared pool ID

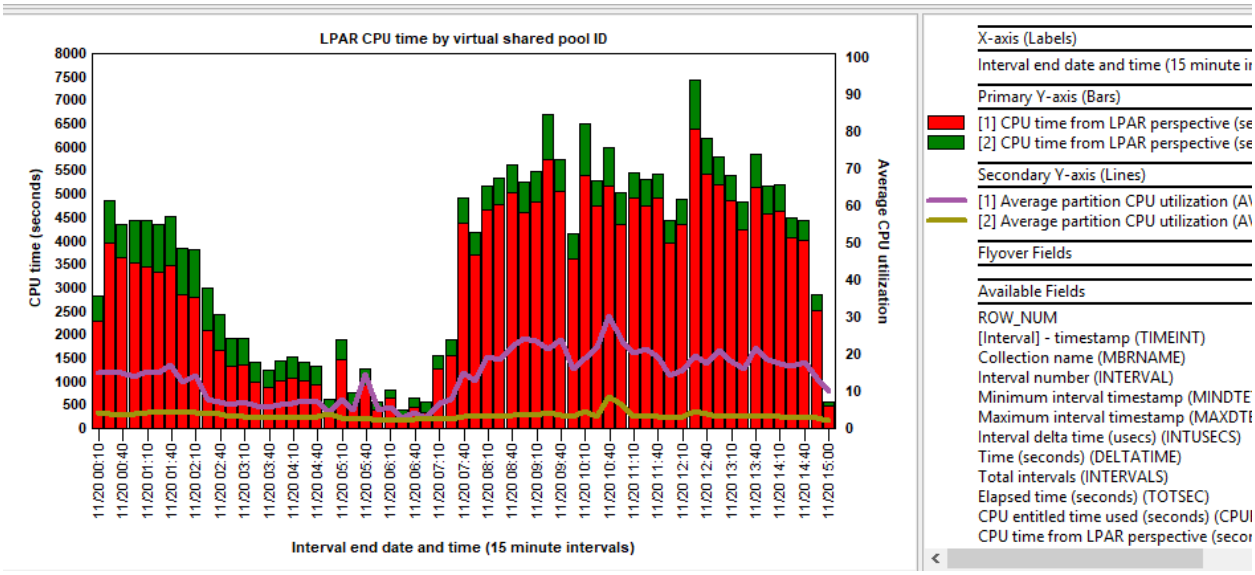
This folder contains a set of graphs like those under the Systems graphs folder but flattened by virtual shared pool ID.



System graphs -> LPAR graphs by virtual shared pool ID

9.16.21.1 LPAR CPU time by virtual shared pool ID

This graph displays total CPU time consumed from all LPARs within each virtual shared pool on the primary Y-axis. The secondary Y-axis contains the average CPU utilization per virtual shared pool.

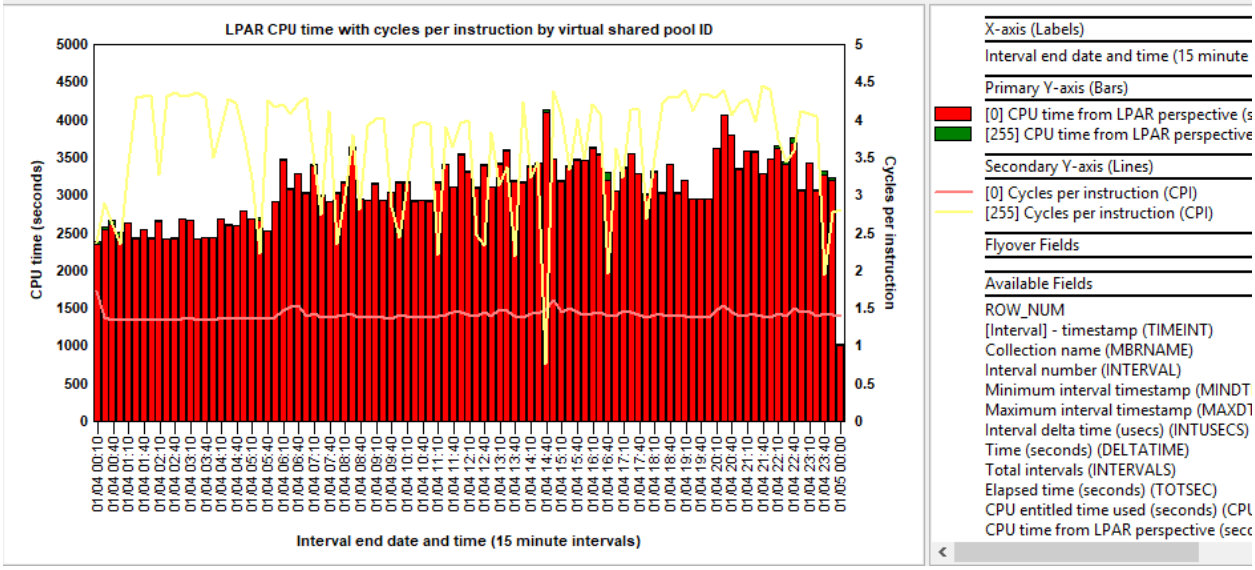


LPAR CPU time by virtual shared pool ID

9.16.21.2 LPAR CPU time with cycles per instruction by virtual shared pool ID

This graph is the same as the previous one except includes average CPI on the secondary Y-axis per virtual shared pool ID.

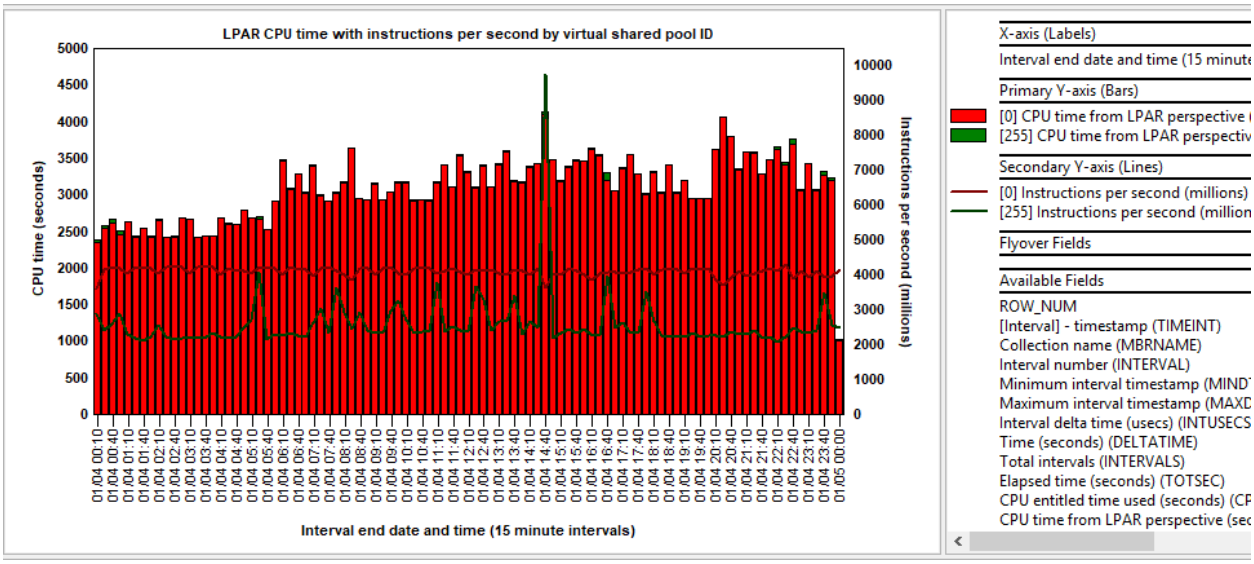
Note: CPI is available only for partitions running on Power6® and Power7® hardware. Otherwise the values are not available and will be 0.



LPAR CPU time with cycles per instruction by virtual shared pool ID

9.16.21.3 LPAR CPU time with instructions per second by virtual shared pool ID

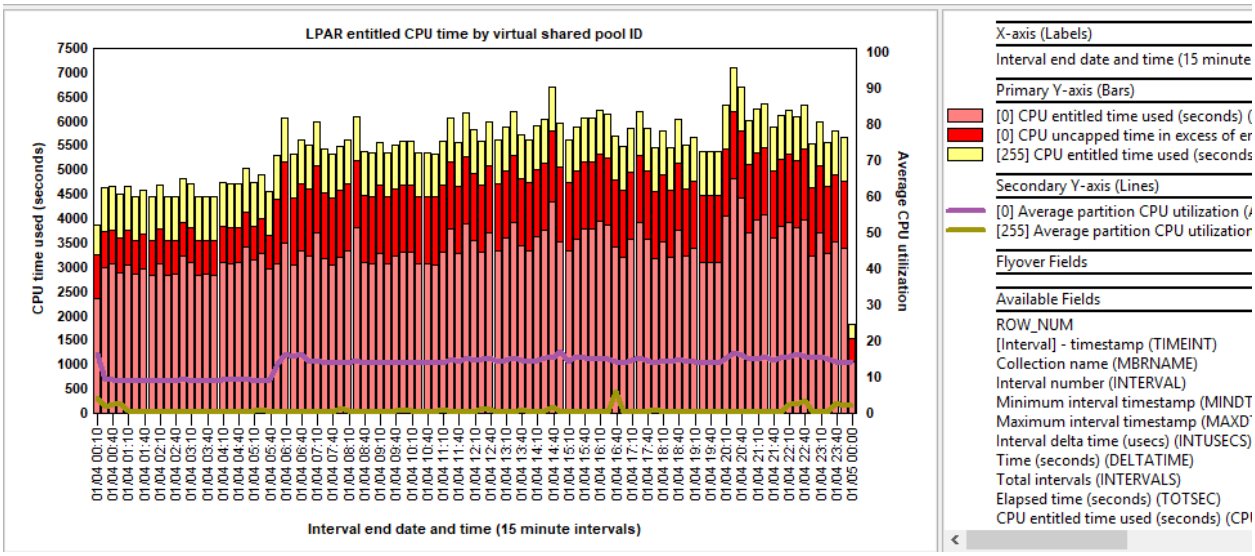
This graph is the same as the previous one except includes average instructions per second on the secondary Y-axis per virtual shared pool.



LPAR CPU time with instructions per second by virtual shared pool ID

9.16.21.4 LPAR entitled CPU time by virtual shared pool ID

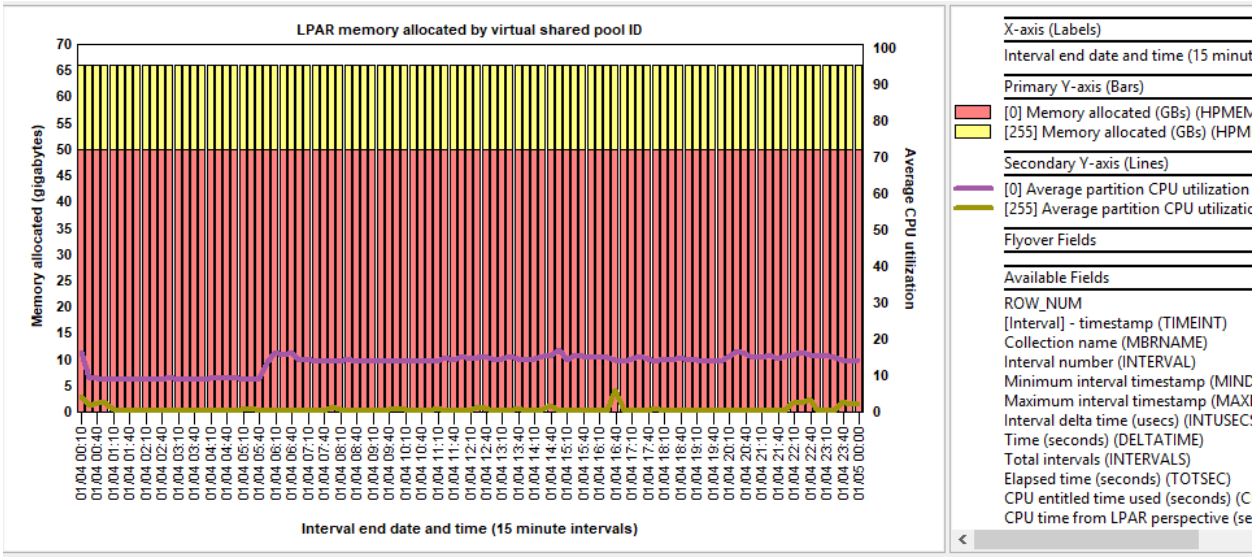
This graph shows (per shared pool ID) the total CPU entitled time used and uncapped time in excess of capacity on the primary Y-axis and the average CPU utilization on the secondary Y-axis.



LPAR entitled CPU time by virtual shared pool ID

9.16.21.5 LPAR memory allocated by virtual shared pool ID

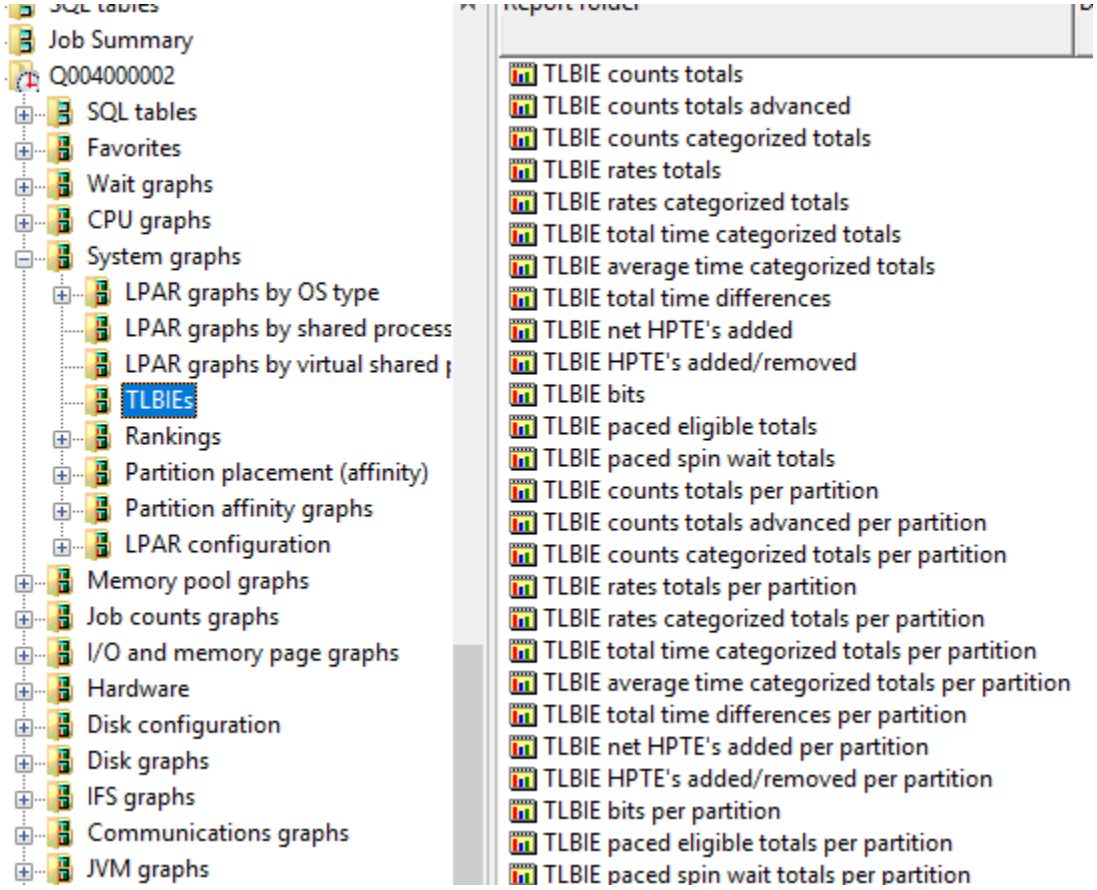
This graph shows the total memory allocated for all LPARs in each shared pool on the primary Y-axis and the average CPU utilization per virtual shared pool on the secondary Y-axis.



LPAR memory allocated by virtual shared pool ID

9.16.22 TLBIEs

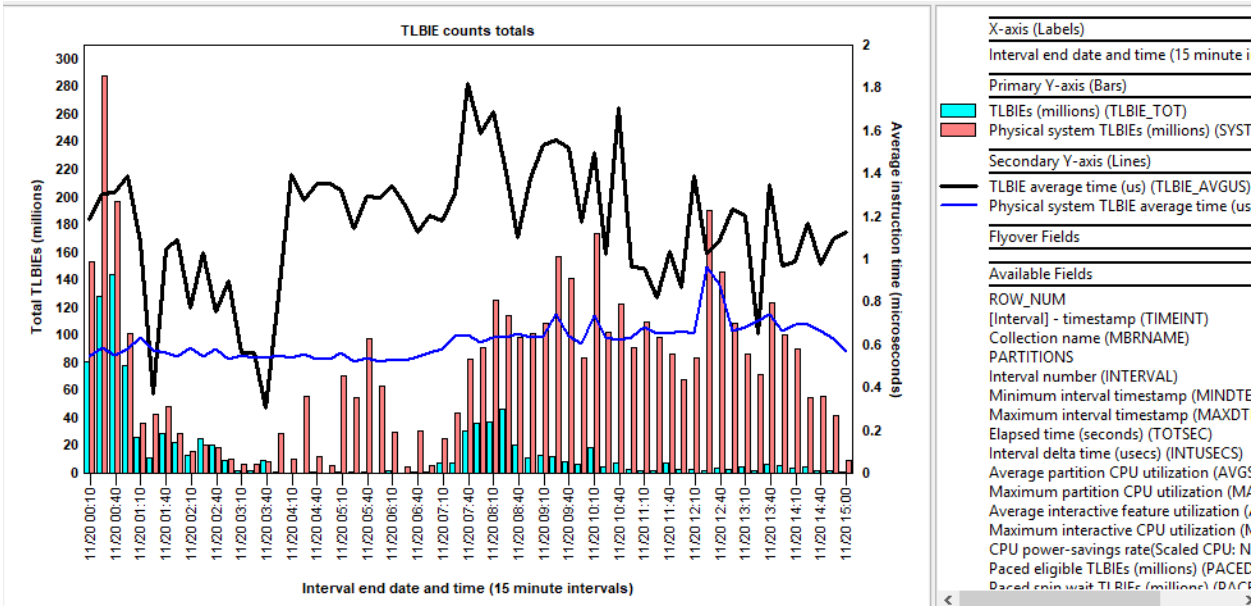
These graphs cover various metrics analyzing Translation Look-aside Buffer Invalidate Entries Instructions (TLBIEs) that occur on the system. Typically, these are intended for advanced users or IBM support.



System graphs -> TLBIEs folder

9.16.22.1 TLBIE counts totals

This graph compares total TLBIEs on the system vs the current LPAR on the primary Y-axis and the average time in microseconds on the secondary Y-axis.



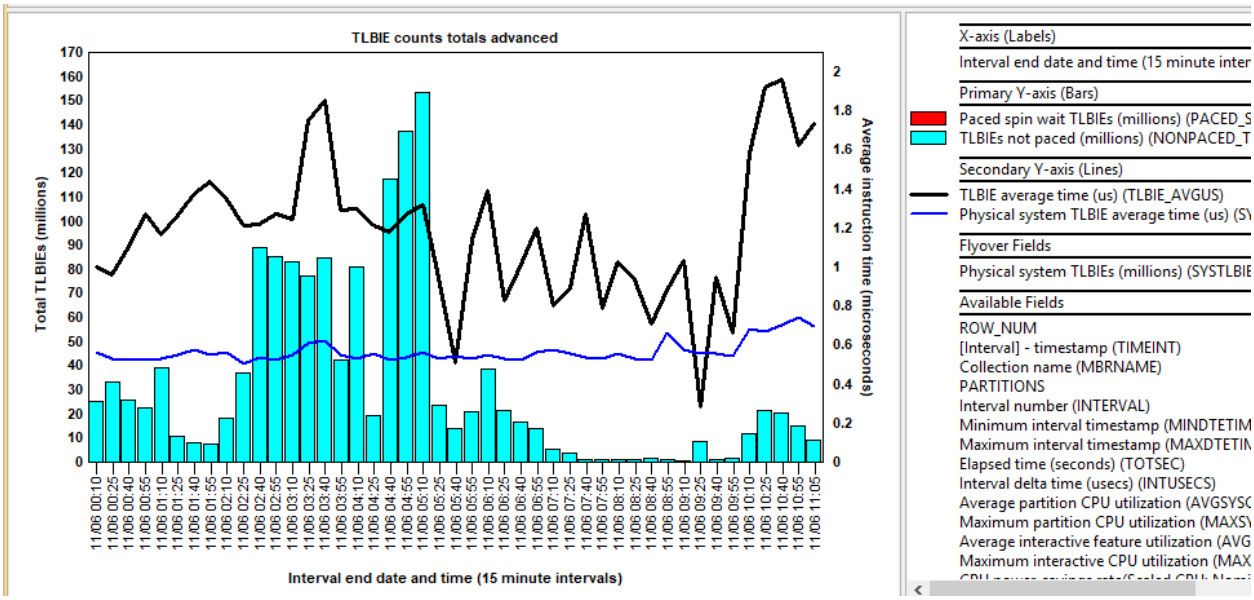
TLBIE counts totals

9.16.22.2 TLBIE counts total advanced

This graph breaks down the TLBIE counts occurring on the LPAR into 3 types:

1) Paced spin wait TLBIEs
2) Paced non-spin wait TLBIEs
3) TLBIEs not paced

The secondary Y-axis displays the average time of the LPAR and physical system in microseconds.



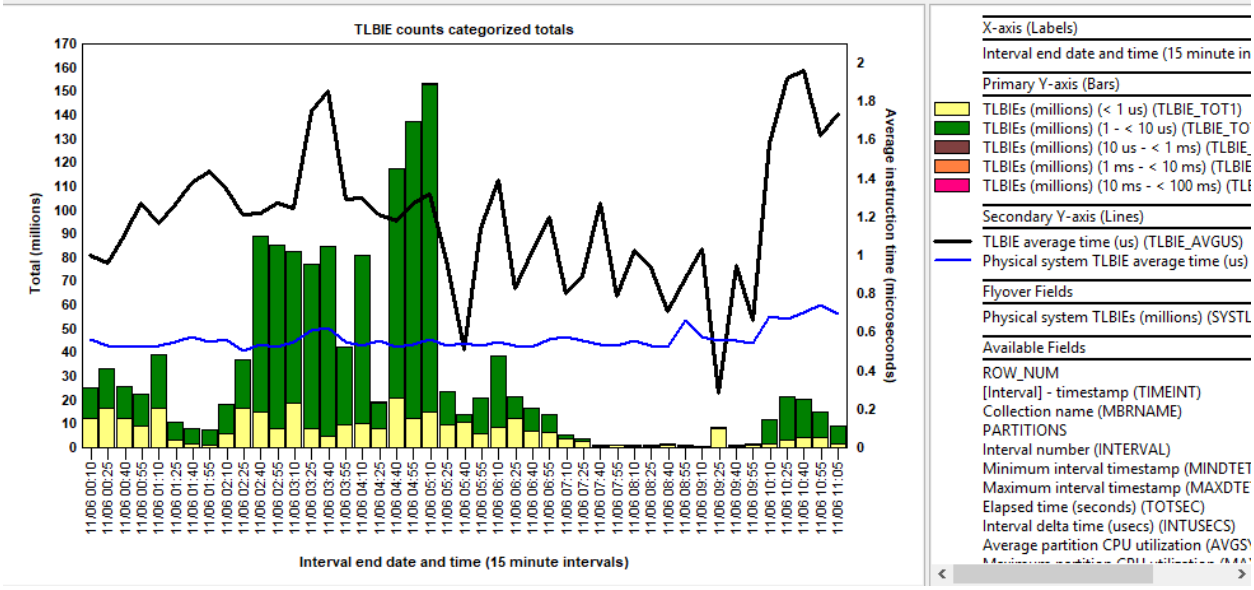
TLBIE counts total advanced

9.16.22.3 TLBIE counts categorized totals

This graph breaks out the TLBIE counts into 6 categories based on the average time of each.

1) <1 microseconds
2) 1 to < 10 microseconds
3) 10 microseconds to < 1 millisecond
4) 1 millisecond to < 10 milliseconds
5) 10 milliseconds to < 100 milliseconds
6) >= 100 milliseconds

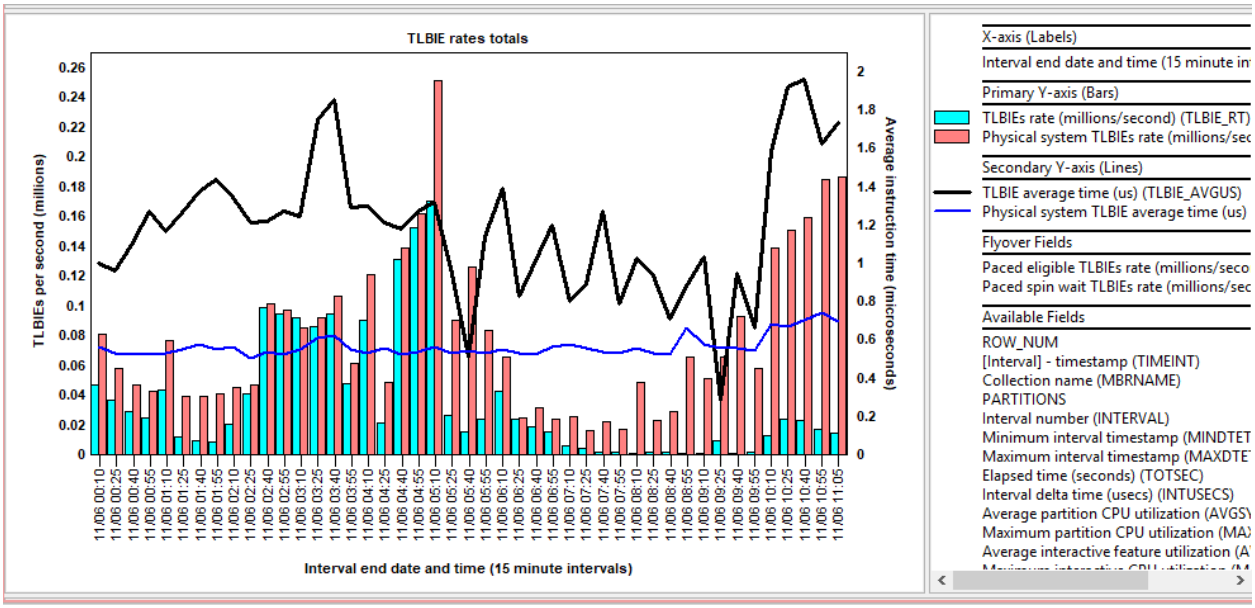
The secondary Y-axis displays the average time of the LPAR and physical system in microseconds.



TLBIE counts categorized totals

9.16.22.4 TLBIE rates totals

This graph compares TLBIE rate per second on the system vs the current LPAR on the primary Y-axis and the average time in microseconds on the secondary Y-axis.



TLBIE rates totals

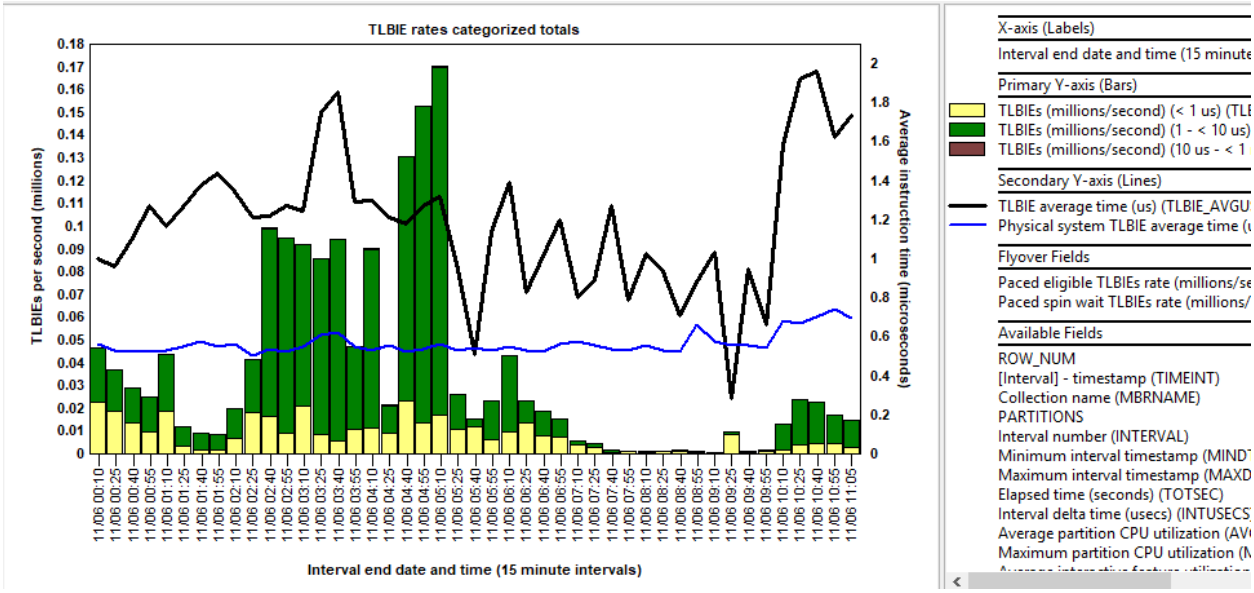
9.16.22.5 TLBIE rates categorized totals

This graph breaks out the TLBIE rate per second into 6 categories based on the average time of each.

1) <1 microseconds
2) 1 to < 10 microseconds
3) 10 microseconds to < 1 millisecond

4) 1 millisecond to < 10 milliseconds
5) 10 milliseconds to < 100 milliseconds
6) >= 100 milliseconds

The secondary Y-axis displays the average time of the LPAR and physical system in microseconds.

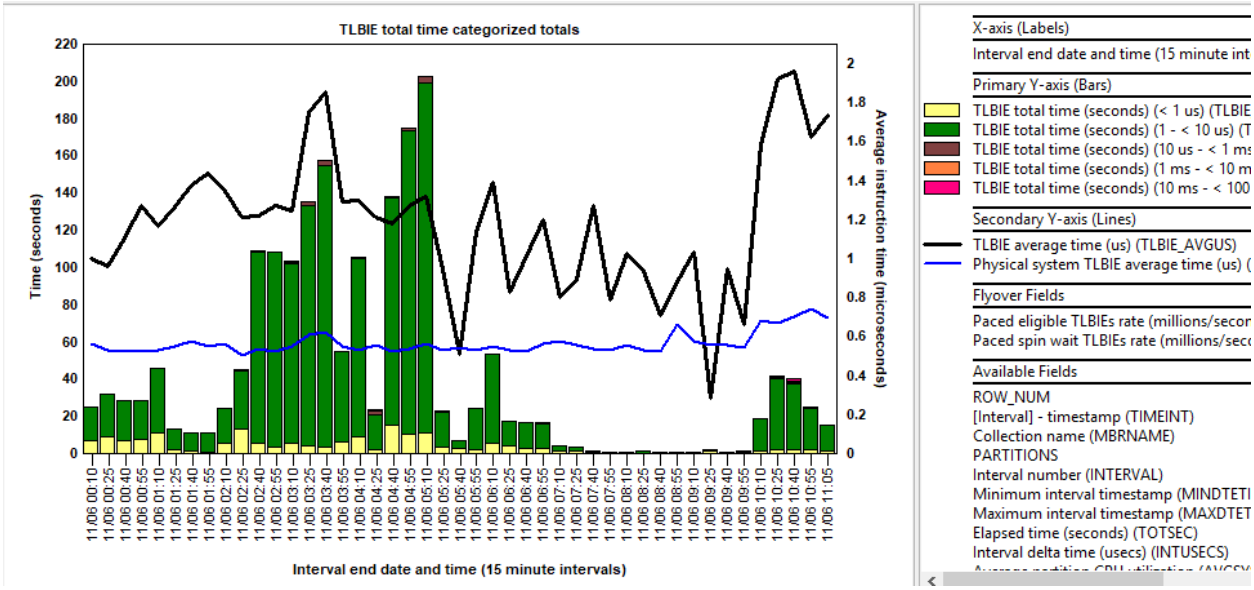


TLBIE rates categorized totals

9.16.22.6 TLBIE total time categorized totals

This graph shows the total time spent executing TLBIE instructions for each of the 6 average response time categories.

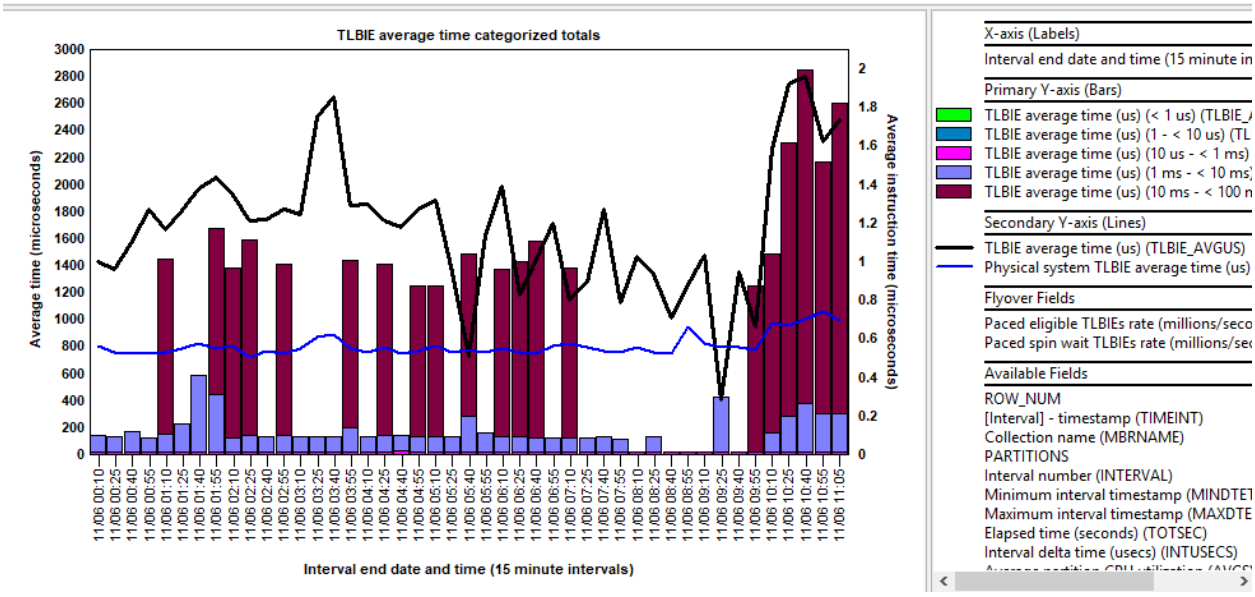
The secondary Y-axis displays the average time of the LPAR and physical system in microseconds.



TLBIE total time categorized totals

9.16.22.7 TLBIE average time categorized totals

This graph shows the average time spent for the TLBIEs within each of the 6 response time buckets.

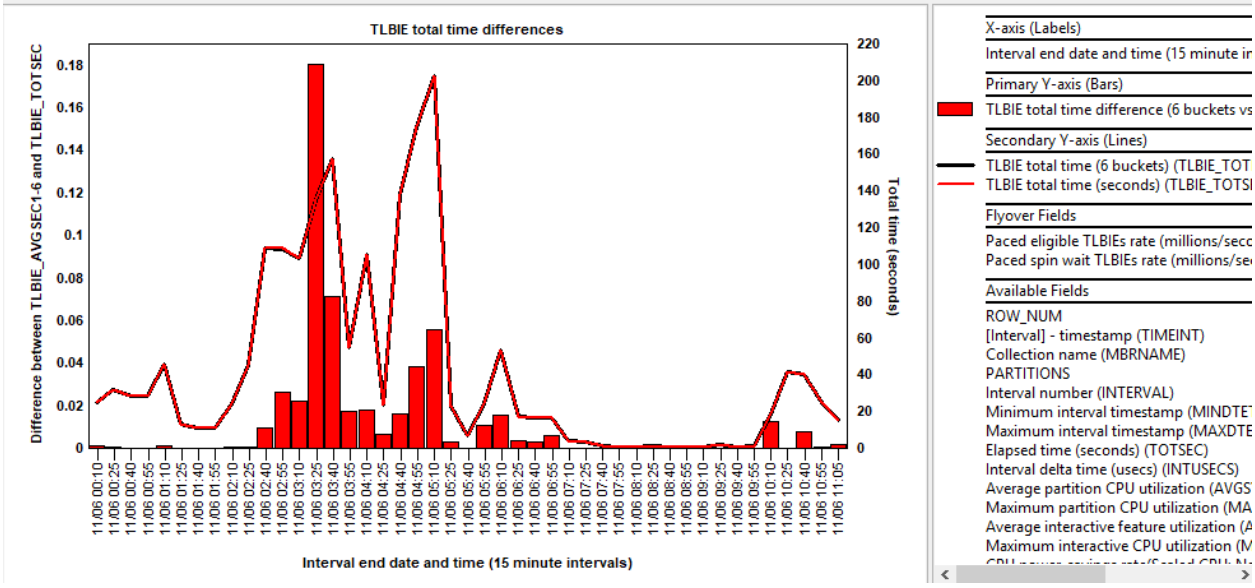


TLBIE average time categorized totals

9.16.22.8 TLBIE total time differences

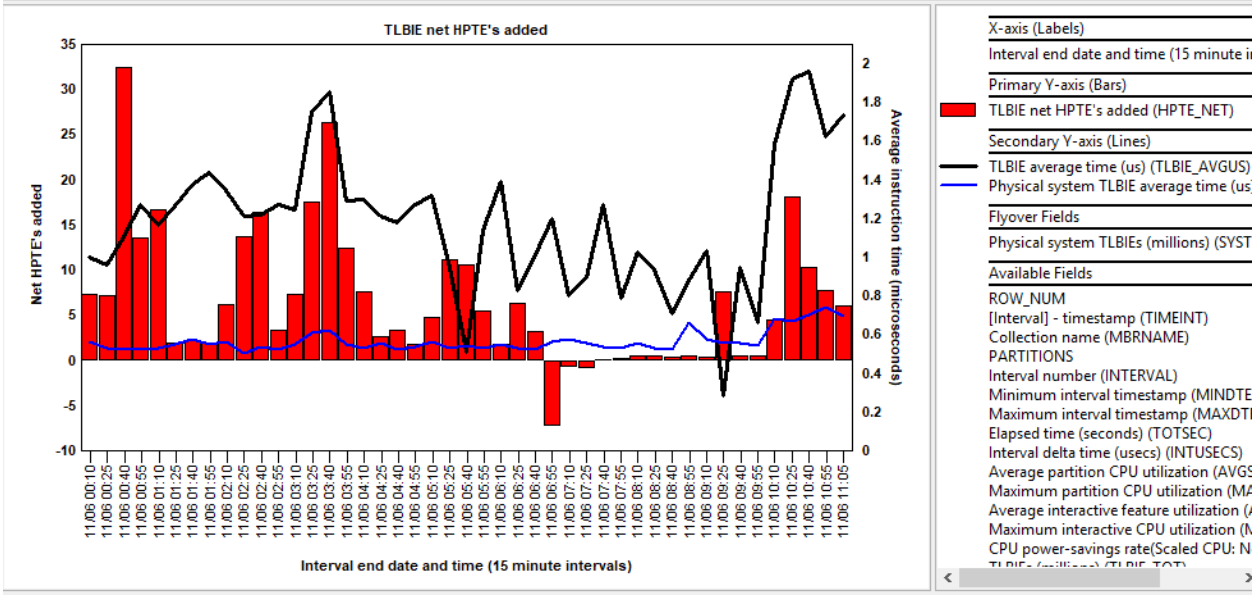
This graph is used to compare the total time of each of the 6 buckets added together vs the overall total time.

Note: *Use the TLBIE total time differences graph to check that the latest TR was applied before collecting the data if necessary. This graph compares the categorized (6 bucket total times added together) vs the TLBIE total time and ideally should show values < 1 second or even 0. If this is NOT the case then the latest TR updates should be applied or some graphs will show incorrect values:*



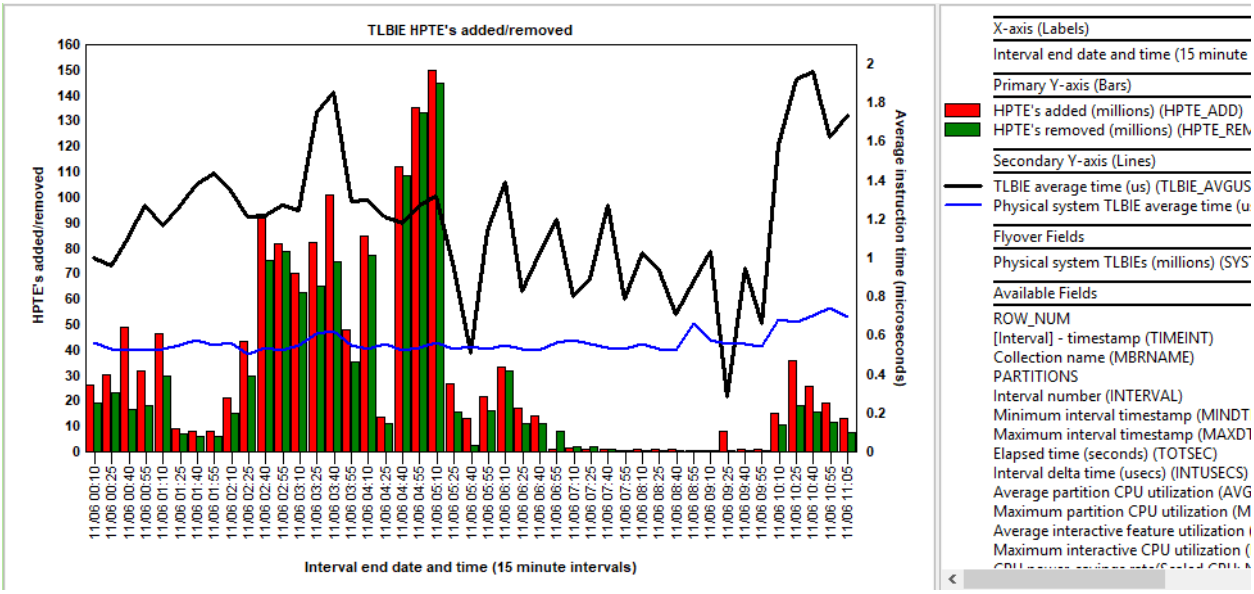
TLBIE total time differences

9.16.22.9 TLBIE net HPTE's added



TLBIE net HPTE's added

9.16.22.10 TLBIE HPTE's added/removed



TLBIE HPTE's added/removed

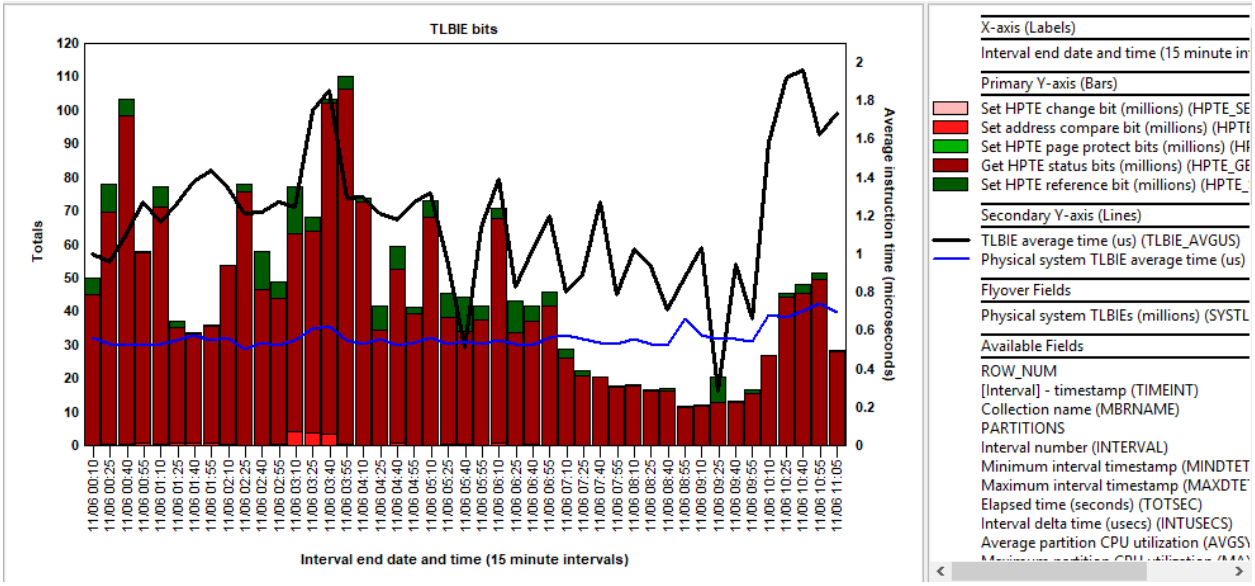
9.16.22.11 TLBIE bits

This graph compares the numbers of times various TLBIE settings (i.e. bits) were used. The bits checked are the following:

1) Set HPTE change bit
2) Set HPTE storage key bit
3) Set address compare bit
4) Set HPTE page protect bits

5) Get HPTE status bits
6) Set HPTE reference bit

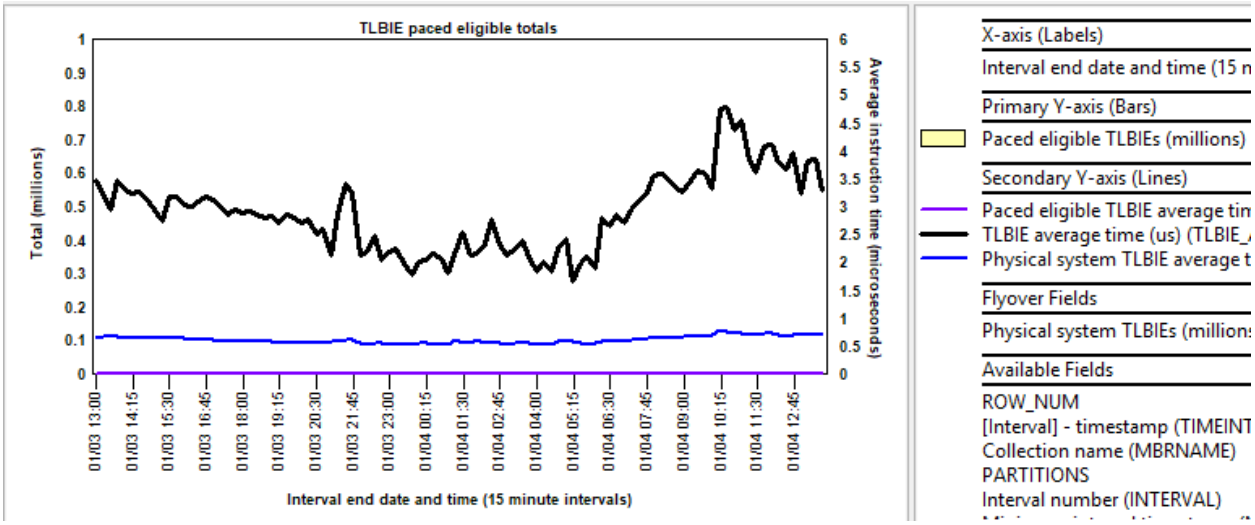
Note: If some of these options are not listed in the graph, this means the value of that setting was zero for the entire period shown.



TLBIE bits

9.16.22.12 TLBIE paced eligible totals

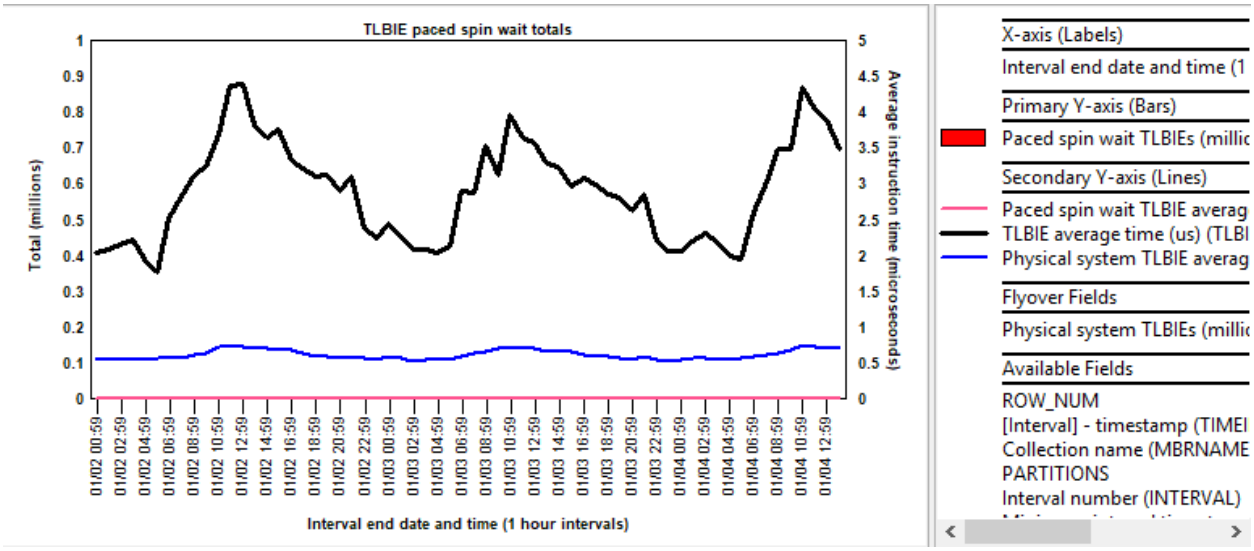
This graph displays the total paced eligible TLBIEs found during the collection. Typically, it is zero.



TLBIE paced eligible totals

9.16.22.13 TLBIE paced spin wait totals

This graph displays the total paced spin wait TLBIEs found during the collection. Typically, it is zero.

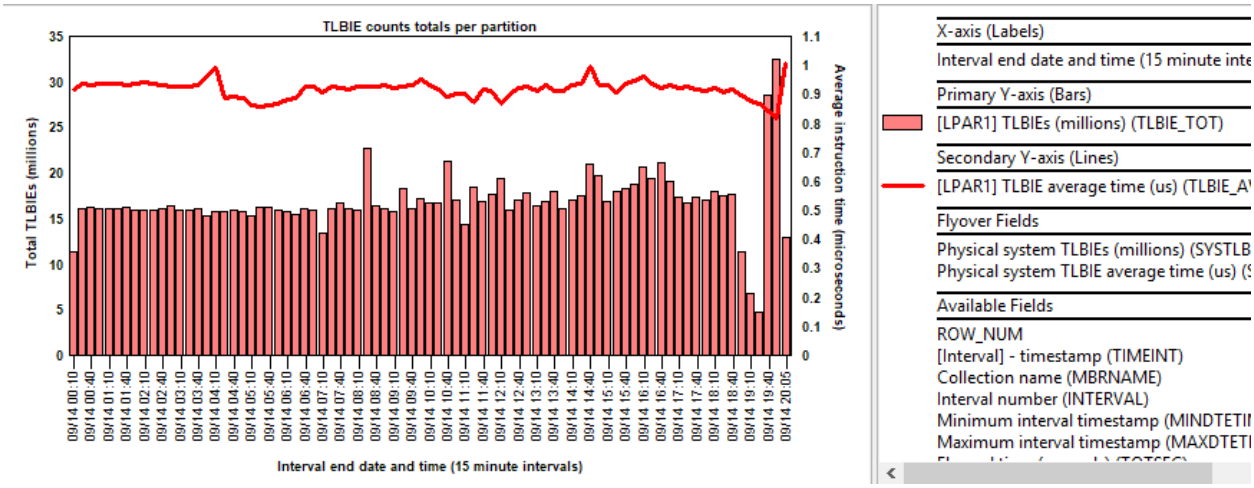


TLBIE paced spin wait totals

9.16.22.14 Per partition graphs

The remaining graphs are the same as the ones described previously except each color represents a different LPAR found in the data. Typically, in order to get data from multiple LPARs into a single graph you must copy the collection (performance data) from different LPARs all into the same library and use the option to graph multiple collections at once.

An example:



TBLIE counts totals per partition

The list of graphs is:

TLBIE counts totals per partition

TLBIE counts totals advanced per partition

TLBIE counts categorized totals per partition

TLBIE rates totals per partition

TLBIE rates categorized totals per partition

TLBIE total time categorized totals per partition

TLBIE average time categorized totals per partition

TLBIE total time differences per partition

TLBIE net HPTE's added per partition

TLBIE HPTE's added/removed per partition

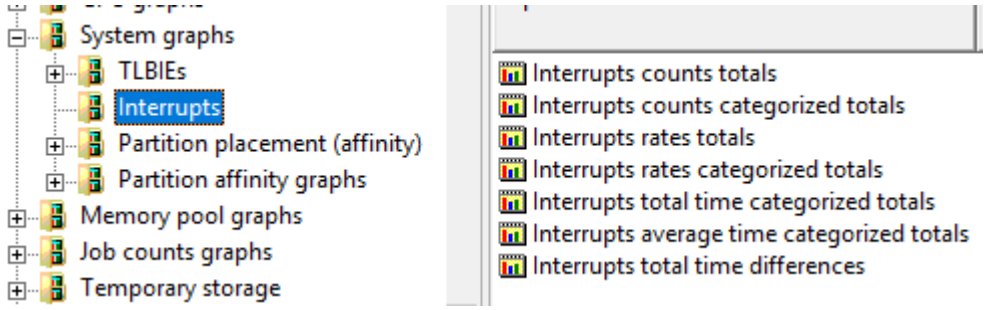
TLBIE bits per partition

TLBIE paced eligible totals per partition

TLBIE paced spin wait totals per partition

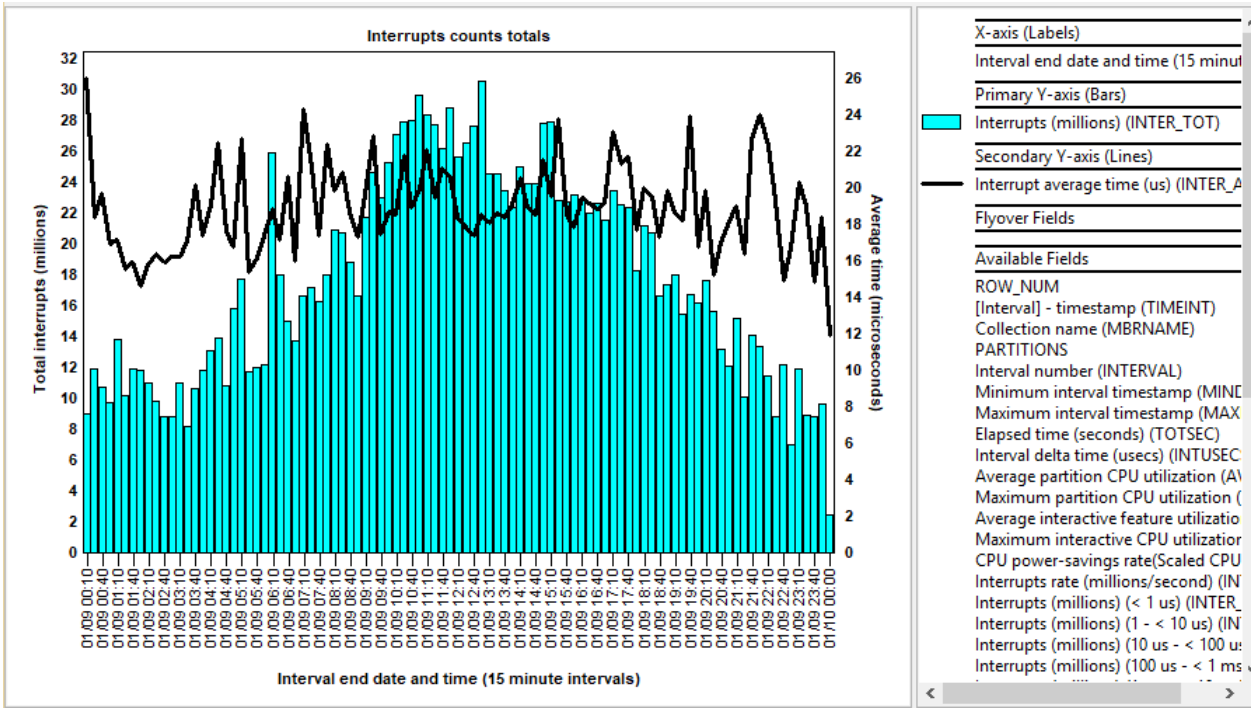
9.16.23 Interrupts

This folder contains a set of graphs to measure the number of interrupts that occurred and how much time they took to occur.



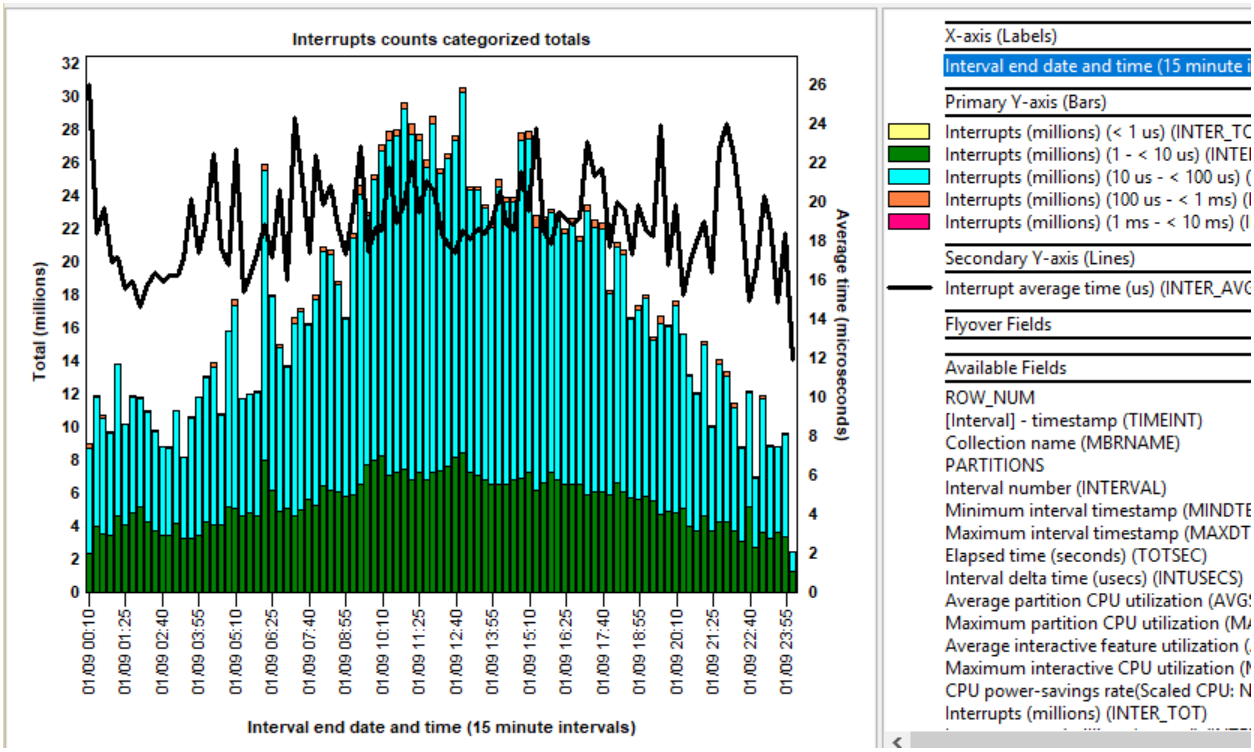
System graphs -> Interrupts

9.16.23.1 Interrupts counts totals



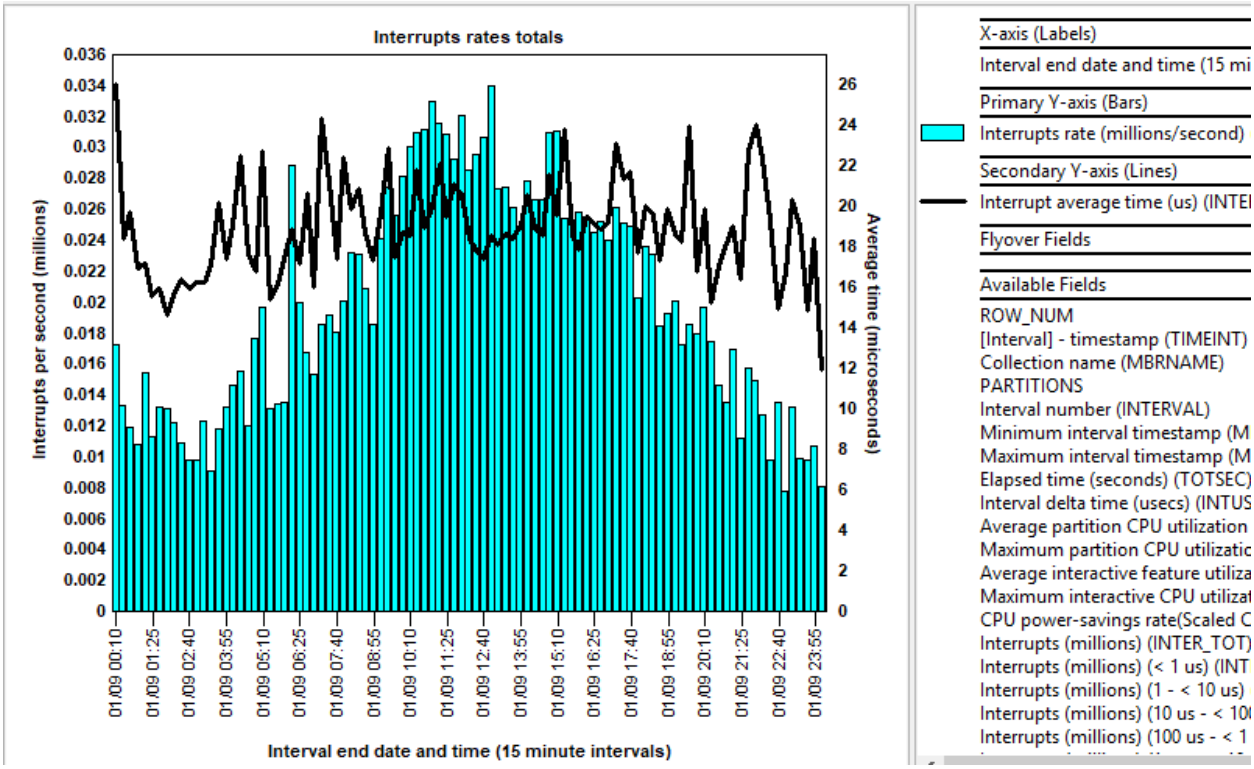
Interrupts counts totals

9.16.23.2 Interrupts counts categorized totals



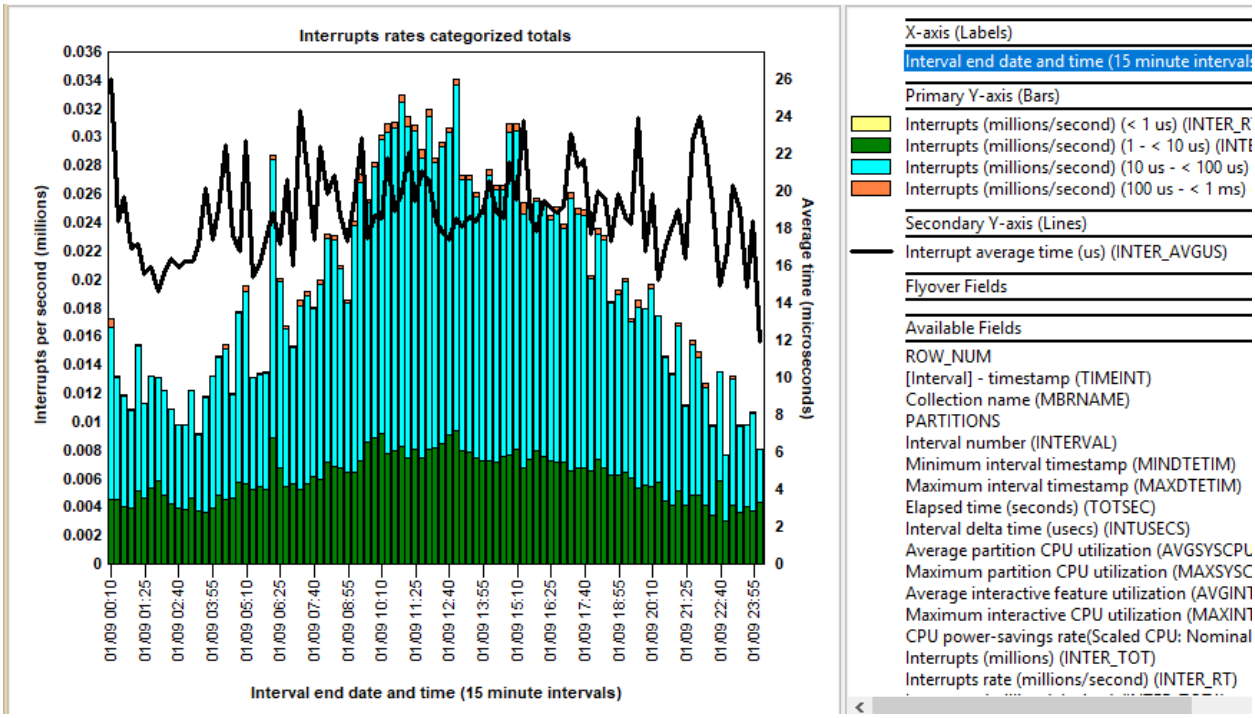
Interrupts counts categorized totals

9.16.23.3 Interrupts rates totals



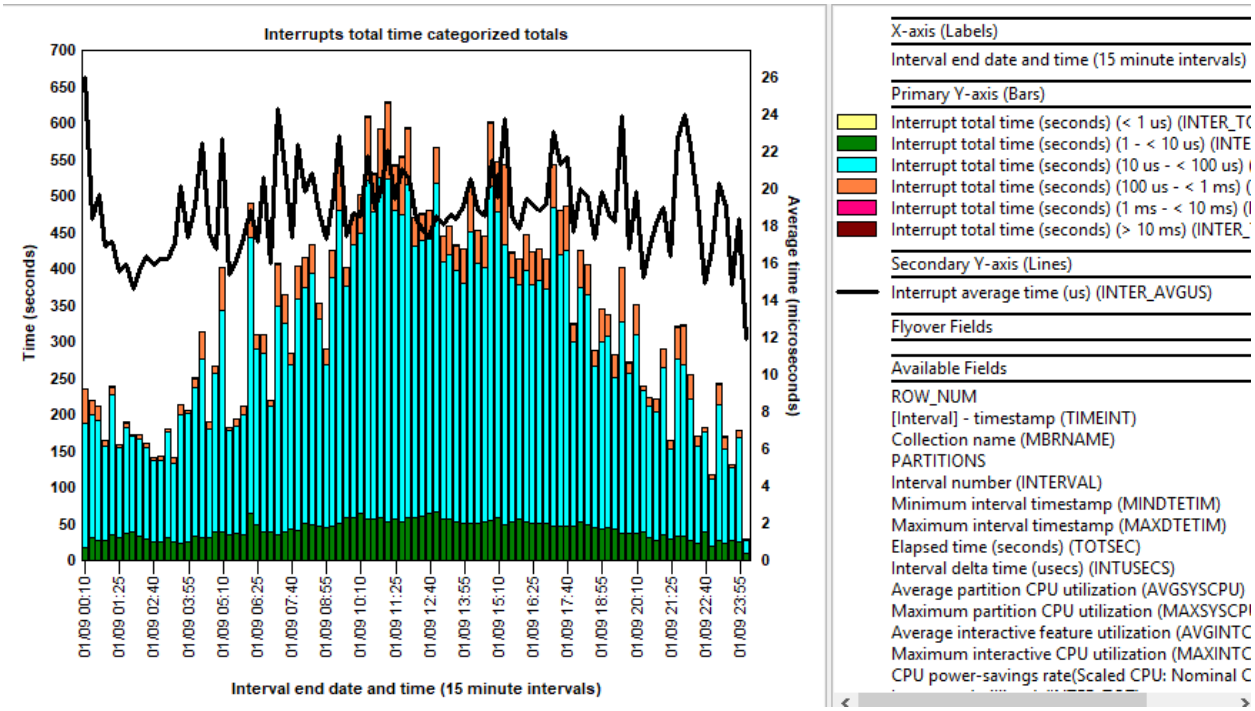
Interrupts rates totals

9.16.23.4 Interrupts rates categorized totals



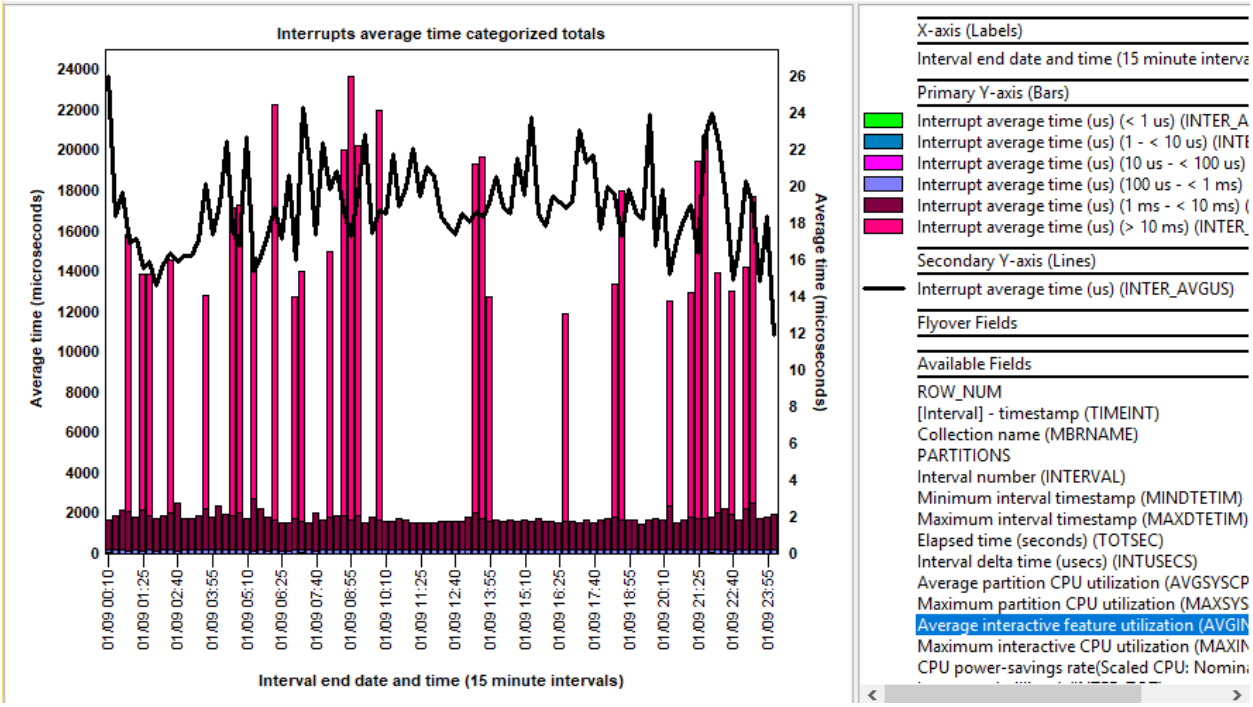
Interrupts rates categorized totals

9.16.23.5 Interrupts total time categorized totals



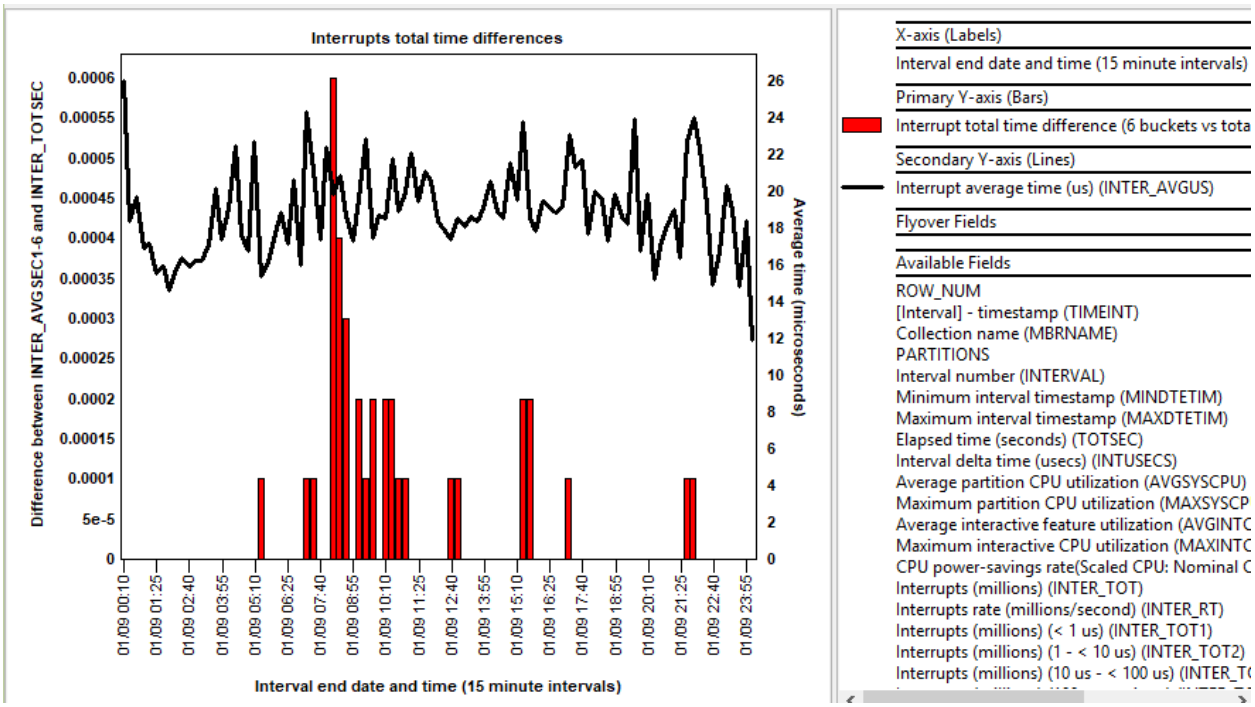
Interrupts total time categorized totals

9.16.23.6 Interrupts average time categorized totals



Interrupts average time categorized totals

9.16.23.7 Interrupts total time differences



Interrupts total time differences

9.16.24 Partition placement (affinity)

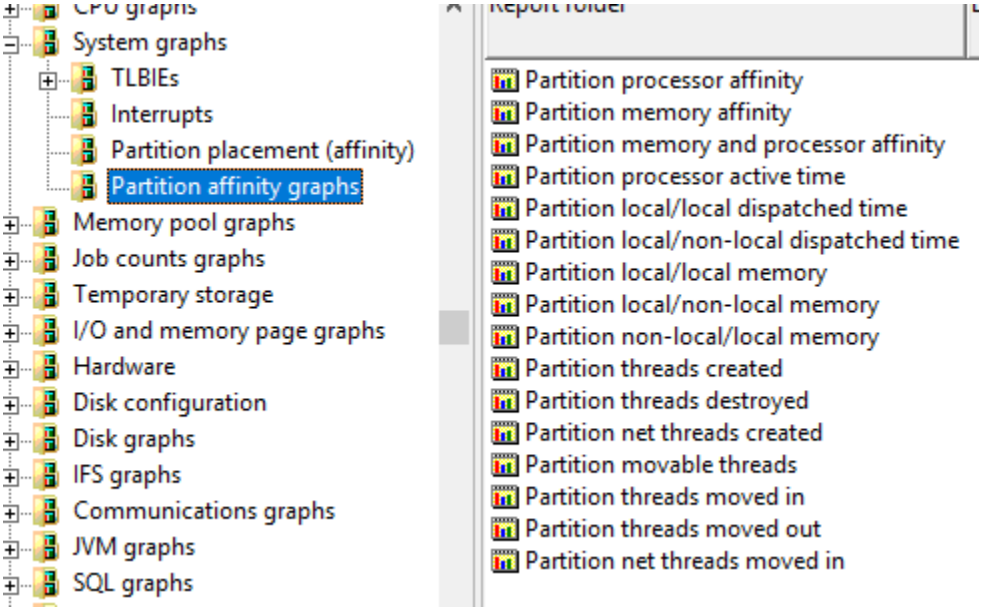
These reports show the CPU and memory layout in terms of the book/node placement on the partition.

System graphs -> Partition placement (affinity)

9.16.25 Partition affinity graphs

These graphs display CPU and memory layout in terms of book/node placement over time during the collection. Additional graphs are provided that show thread counts over time per book/node combination.

These are intended primarily for IBM use only due to potential data accuracy issues.



System graphs -> Partition affinity graphs

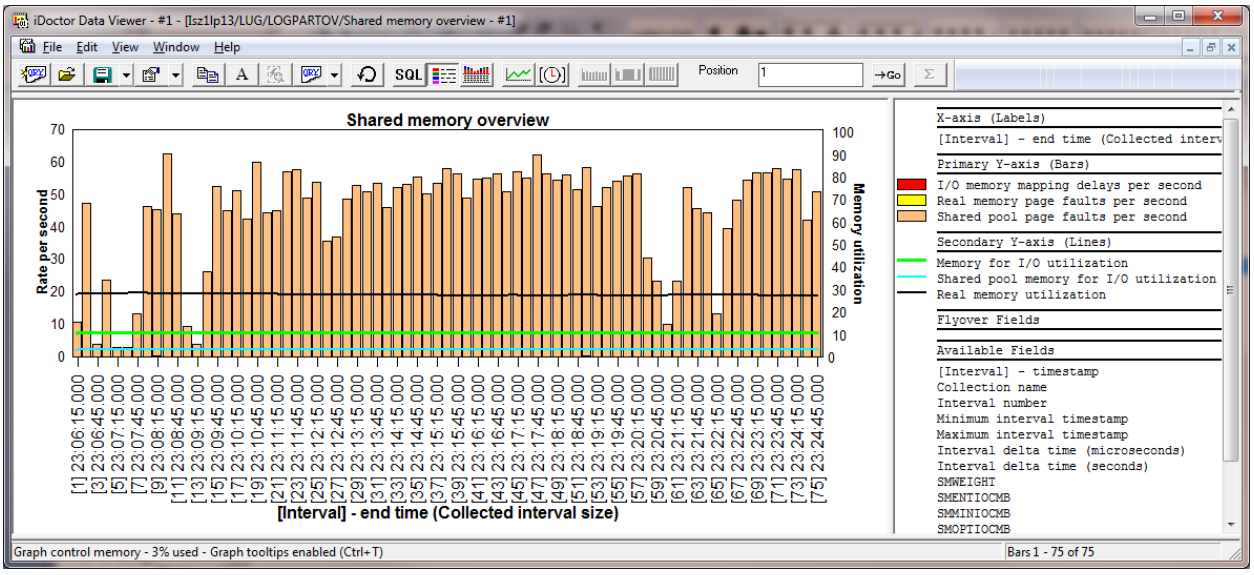
9.16.26 Shared memory graphs

These graphs report information about shared memory pool usage. Data is reported for both the partition's use of the pool as well as pool metrics that are the sum of activity caused by all partitions using the pool. The data behind these graphs is created in file QAPMSHRMP.

Note: This data requires a POWER6 system with firmware level xx340_075 (or higher.)

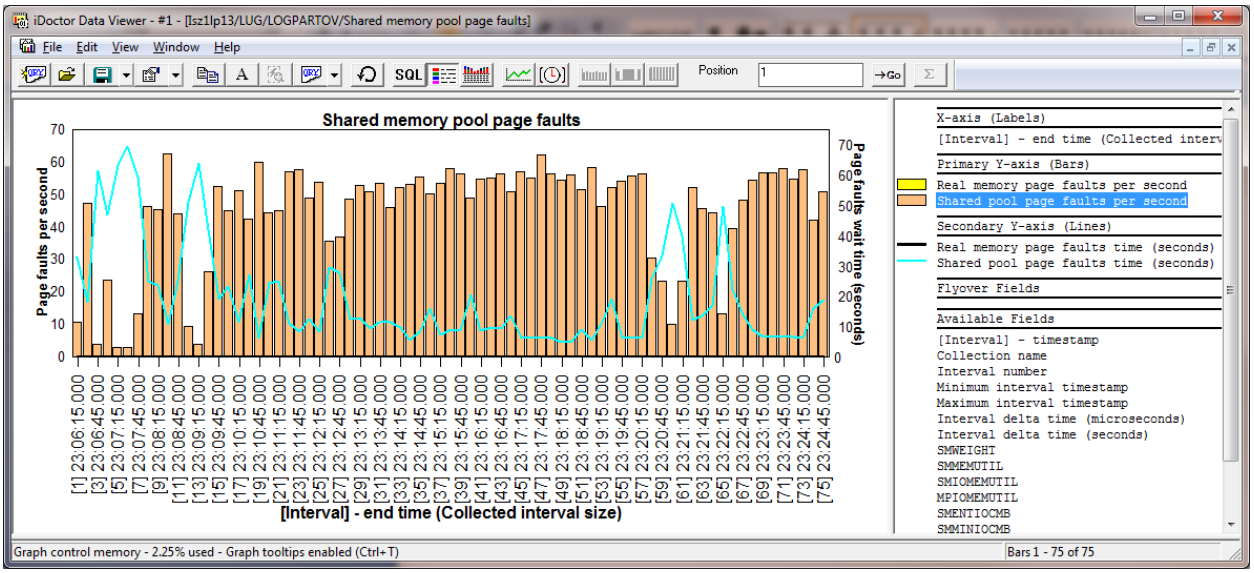
9.16.26.1 Shared memory overview

This graph shows the rate of I/O memory mapping delays, real memory page faults and shared pool page faults.



Shared memory overview

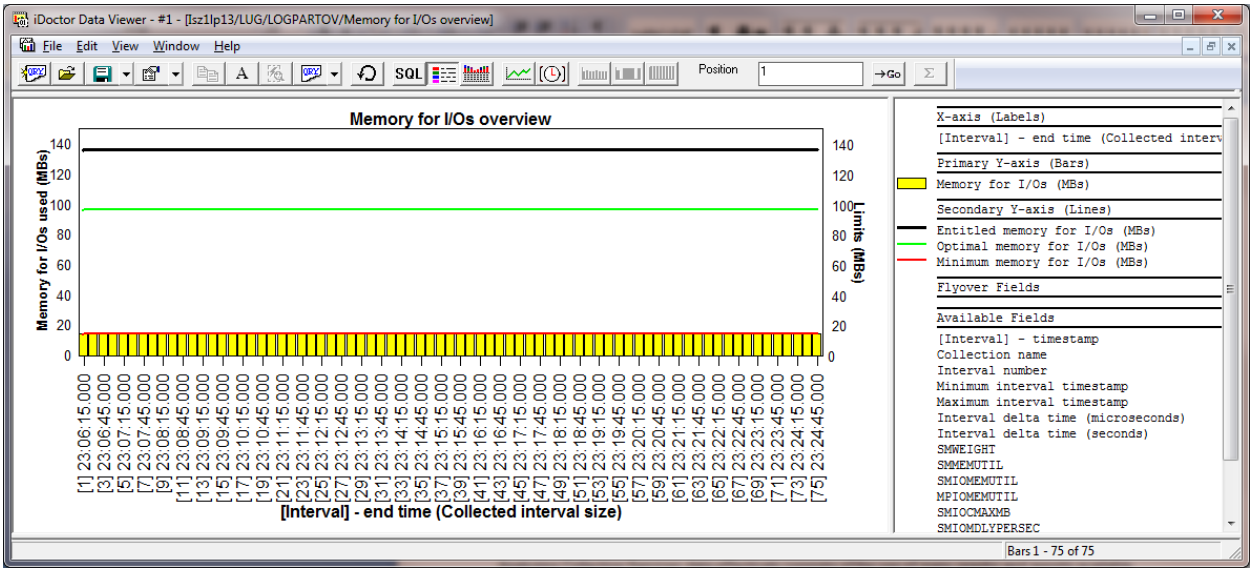
9.16.26.2 Shared memory pool page faults



Shared memory pool page faults

9.16.26.3 Memory for I/Os overview

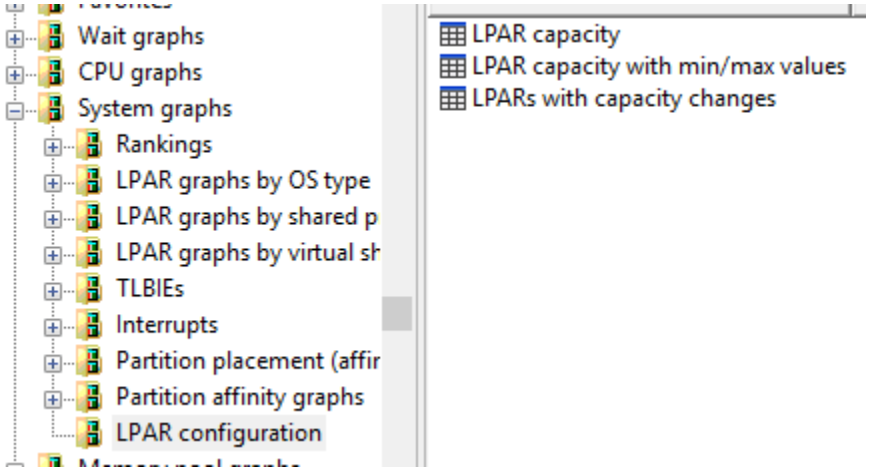
This graph shows entitled, optional, actual and maximum memory entitled to perform I/Os.



Memory for I/Os overview

9.16.27 LPAR configuration

This folder contains reports that provide detailed information about how the LPAR settings like current processing capacity are set.



Systems graphs -> LPAR configuration

9.16.27.1 LPAR capacity

This report shows the basic information about each LPAR found in the collection.

Collection name (MBRNAME)	Partition name (PARTITION)	Partition ID (PARTITIONID)	Virtual shared pool ID (HPPOOLI...	OS	Processor sharing (PROCSHARE)	Configured virtual processors (HPVPRC)	Current processing capacity (HPPRCCC)	Memory allocated (GBs) (HPMEMC_GBS)
Q010000007	LPAR0003	0003	1	IBM i	Uncapped sharing	7	3	112
Q010000007	LPAR0003	0003	1	IBM i	Uncapped sharing	7	3	117
Q010000007	LPAR000B	000B	0	IBM i	Uncapped sharing	2	1.8000	40
Q010000007	LPAR000B	000B	0	IBM i	Uncapped sharing	2	1.8000	35
Q010000007	LPAR0001	0001	0	Other OS	Uncapped sharing	4	1	8
Q010000007	LPAR0011	0011	0	Other OS	Uncapped sharing	5	1	17
Q010000007	LPAR000F	000F	0	IBM i	Uncapped sharing	1	.9000	5
Q010000007	LPAR0002	0002	0	Other OS	Uncapped sharing	1	.8000	8
Q010000007	LPAR0004	0004	1	IBM i	Uncapped sharing	1	.6000	8
Q010000007	LPAR0006	0006	1	IBM i	Uncapped sharing	1	.4000	5
Q010000007	LPAR0008	0008	1	IBM i	Uncapped sharing	1	.4000	8
Q010000007	LPAR000C	000C	0	IBM i	Uncapped sharing	1	.3000	4
Q010000007	LPAR000E	000E	0	IBM i	Uncapped sharing	1	.3000	5
Q010000007	LPAR0010	0010	0	Other OS	Uncapped sharing	2	.2000	7
Q010000007	LPAR0005	0005	1	IBM i	Uncapped sharing	1	.1000	5
Q010000007	LPAR0009	0009	1	IBM i	Uncapped sharing	1	.1000	3.7500
Q010000007	LPAR0007	0007	1	IBM i	Uncapped sharing	0	0	0
Q010000007	LPAR000A	000A	1	IBM i	Uncapped sharing	0	0	0
Q010000007	LPAR000D	000D	0	IBM i	Uncapped sharing	0	0	0
Q010000007	LPAR0012	0012	0	Other OS	Uncapped sharing	0	0	0
Q010000007	LPAR0013	0013	0	IBM i	Uncapped sharing	0	0	0

LPAR capacity

9.16.27.2 LPAR capacity with min/max values

This report is the same as the previous one except shows the min/max configured values detected during the collection for each LPAR.

Collection name (MBRNAME)	Partition name (PARTITION)	Virtual shared pool ID (HPVPI...	OS	Processor sharing (PROCSHARE)	Configured virtual processors (HPVPRC)	Current processing capacity (HPPRCCC)	Memory allocated (GBs) (HPMEMC_GBS)	Min/max configured virtual processors (MINMAX_HPVPR)	Min/max configured processing capacity (MINMAX_HPPRCC)	Min/max configured memory required (GBs) (MINMAX_HPMEM)
Q010000007	LPAR0003	1	IBM i	Uncapped sharing	7	3	112	1.00 / 9.00	.50 / 6.00	24 / 150
Q010000007	LPAR0003	1	IBM i	Uncapped sharing	7	3	117	1.00 / 9.00	.50 / 6.00	24 / 150
Q010000007	LPAR000B	0	IBM i	Uncapped sharing	2	1.8000	40	1.00 / 4.00	.10 / 4.00	2 / 40
Q010000007	LPAR000B	0	IBM i	Uncapped sharing	2	1.8000	35	1.00 / 4.00	.10 / 4.00	2 / 40
Q010000007	LPAR0001	0	Other OS	Uncapped sharing	4	1	8	1.00 / 4.00	.10 / 1.00	0 / 8
Q010000007	LPAR0011	0	Other OS	Uncapped sharing	5	1	17	1.00 / 30.00	.10 / 3.00	4 / 30
Q010000007	LPAR000F	0	IBM i	Uncapped sharing	1	.9000	5	1.00 / 4.00	.10 / 4.00	2 / 40
Q010000007	LPAR0002	0	Other OS	Uncapped sharing	1	.8000	8	1.00 / 4.00	.10 / 1.00	0 / 8
Q010000007	LPAR0004	1	IBM i	Uncapped sharing	1	.6000	8	1.00 / 4.00	.10 / 4.00	8 / 48
Q010000007	LPAR0006	1	IBM i	Uncapped sharing	1	.4000	5	1.00 / 4.00	.40 / 4.00	2 / 60
Q010000007	LPAR0008	1	IBM i	Uncapped sharing	1	.4000	8	1.00 / 4.00	.10 / 4.00	8 / 96
Q010000007	LPAR000C	0	IBM i	Uncapped sharing	1	.3000	4	1.00 / 4.00	.10 / 4.00	2 / 40
Q010000007	LPAR000E	0	IBM i	Uncapped sharing	1	.3000	5	1.00 / 4.00	.10 / 4.00	2 / 40
Q010000007	LPAR0010	0	Other OS	Uncapped sharing	2	.2000	7	1.00 / 5.00	.10 / .50	3 / 10
Q010000007	LPAR0005	1	IBM i	Uncapped sharing	1	.1000	5	1.00 / 4.00	.10 / 4.00	4 / 48
Q010000007	LPAR0009	1	IBM i	Uncapped sharing	1	.1000	3.7500	1.00 / 7.00	.10 / 4.00	2 / 40
Q010000007	LPAR0007	1	IBM i	Uncapped sharing	0	0	0	.00	.00	0
Q010000007	LPAR000A	1	IBM i	Uncapped sharing	0	0	0	.00	.00	0
Q010000007	LPAR000D	0	IBM i	Uncapped sharing	0	0	0	1.00 / 7.00	.10 / 4.00	0
Q010000007	LPAR0012	0	Other OS	Uncapped sharing	0	0	0	.00	.00	0
Q010000007	LPAR0013	0	IBM i	Uncapped sharing	0	0	0	.00	.00	0

LPAR capacity with min/max values

9.16.27.3 LPARs with capacity changes

This report identifies the LPARs that had configuration changes to memory, virtual processors or current processing capacity settings during the collection.

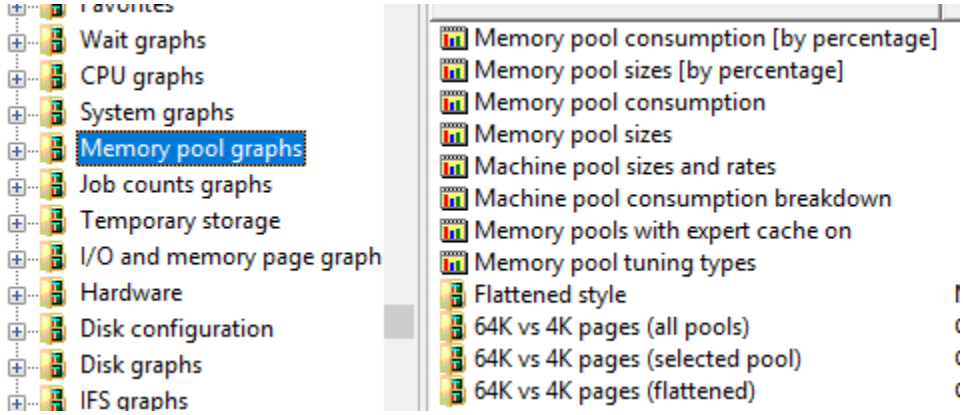
Collection name (MBRNAME)	Total LPAR capacity changes (TOTALCHANGES)	Partition name (PARTITION)	Partition ID (PARTITIONID)	Virtual shared pool ID (HPPOOLI...	OS	Processor sharing (PROCSHARE)	Min/max current virtual processors (MINMAX_HPVPR...	Min/max current processing capacity (MINMAX_HPPRCC...	Min/max memory allocated (GBs) (MINMAX_HPMEM
Q010000007	1	000B	000B	0	IBM i	Uncapped sharing	2.00	1.80	35.00 / 40.00
Q010000007	1	0003	0003	1	IBM i	Uncapped sharing	7.00	3.00	112.00 / 117.00
Q010000007	0	000A	000A	1	IBM i	Uncapped sharing	.00	.00	.00
Q010000007	0	000C	000C	0	IBM i	Uncapped sharing	1.00	.29	4.00
Q010000007	0	000D	000D	0	IBM i	Uncapped sharing	.00	.00	.00
Q010000007	0	000E	000E	0	IBM i	Uncapped sharing	1.00	.29	5.00
Q010000007	0	000F	000F	0	IBM i	Uncapped sharing	1.00	.90	5.00
Q010000007	0	0001	0001	0	Other OS	Uncapped sharing	4.00	1.00	8.00
Q010000007	0	0002	0002	0	Other OS	Uncapped sharing	1.00	.80	8.00
Q010000007	0	0004	0004	1	IBM i	Uncapped sharing	1.00	.59	8.00

LPARs with capacity changes

9.17 Memory pool graphs

These graphs display memory pool statistics over time. Several variations are available.

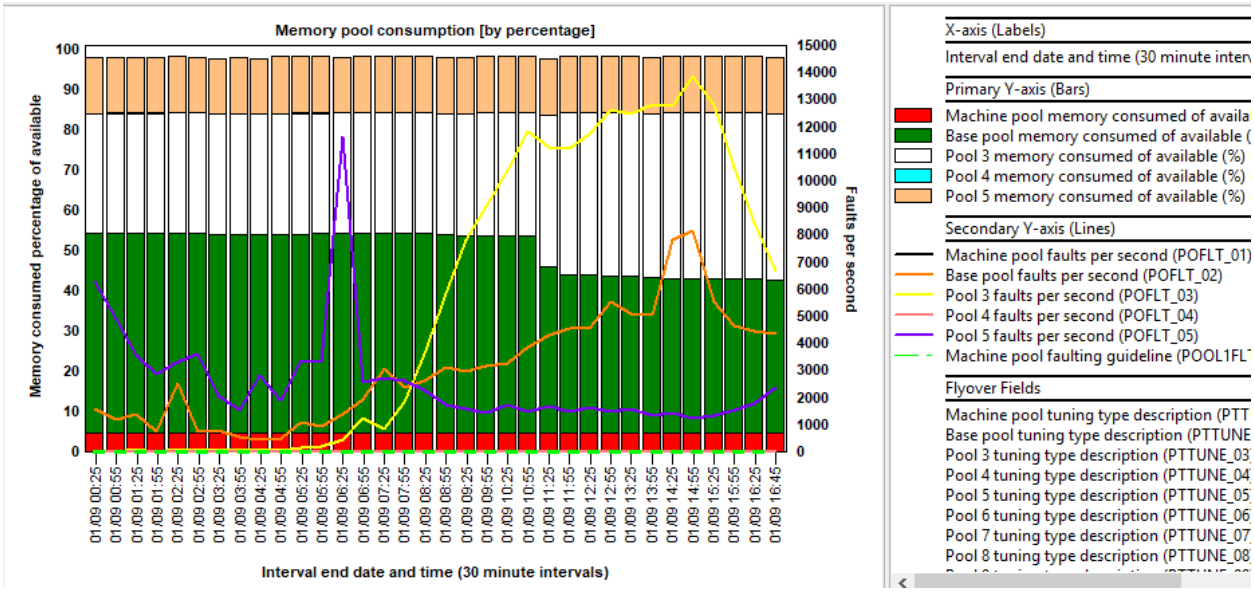
Note: In order for these graphs to show memory pools > 1 TB you will need to have the latest Collection Services PTFs and QAPMPOOLB file needs to be updated as well. If this is NOT done and you have huge memory pool sizes, then the data shown will likely be wrong and capped at the max size the old file format allowed.



Memory pool graphs

9.17.1 Memory pool consumption [by percentage]

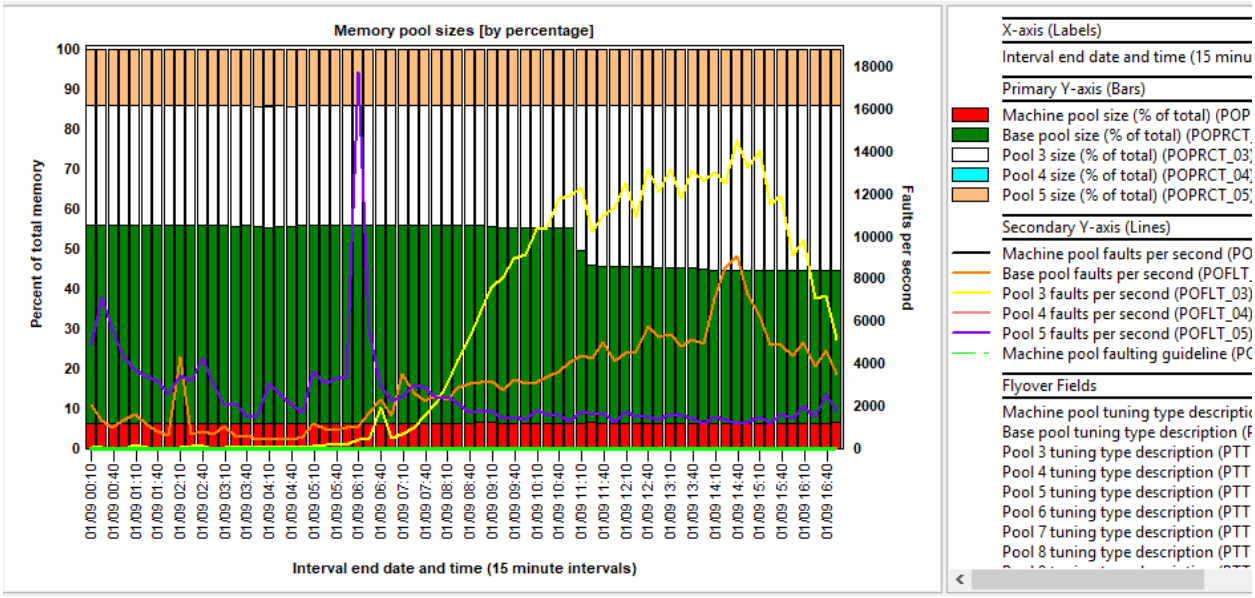
This graph displays up to a max of 24 pools on the primary Y-axis showing the memory consumed and up to 6 pools on the 2nd Y-axis showing faults per second. The memory consumed is shown as a percentage of the total memory available.



Memory pool consumption [by percentage

9.17.2 Memory pool sizes [by percentage]

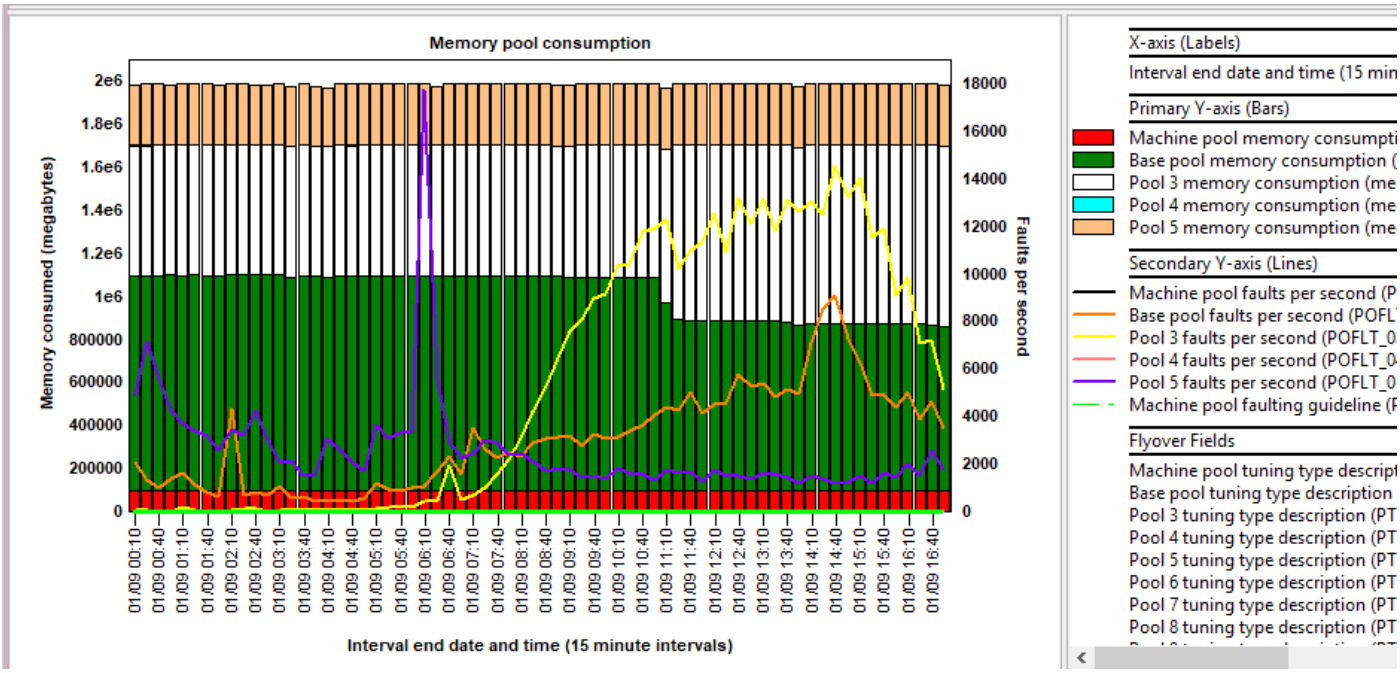
This graph is like the previous one except memory pool sizes (as a percentage of the total) is shown on the primary Y-axis.



Memory pool sizes [by percentage]

9.17.3 Memory pool consumption

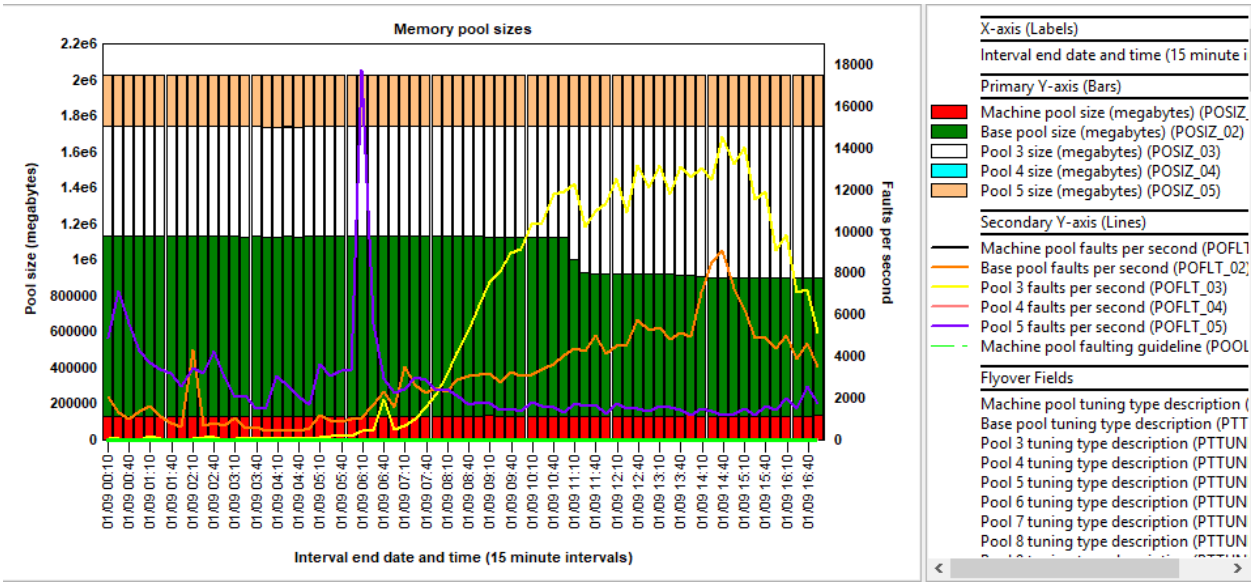
This graph displays up to a max of 24 pools on the primary Y-axis showing the memory consumed and up to 6 pools on the 2nd Y-axis showing faults per second. The memory consumed is shown in megabytes.



Memory pool consumption

9.17.4 Memory pool sizes

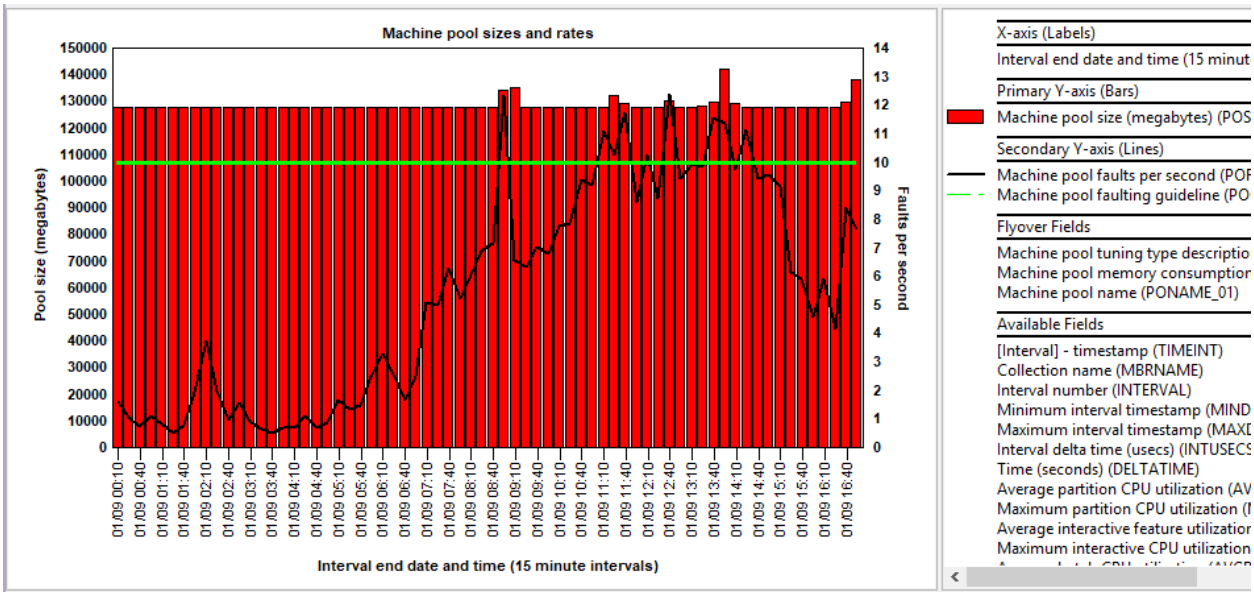
This graph is like the previous one except memory pool sizes (in megabytes) is shown on the primary Y-axis.



Memory pool sizes

9.17.5 Machine pool sizes and rates

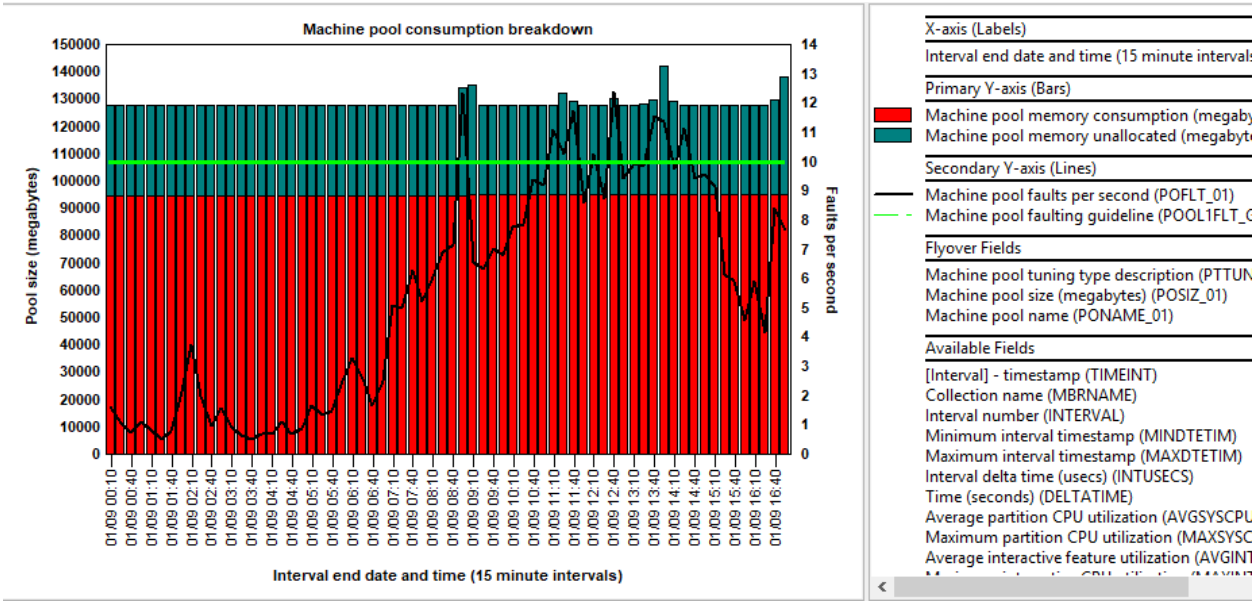
This graph shows the pool size and faulting rates for the machine (base) pool. A machine pool faulting guideline is also displayed as a green line (with a constant value of 10 faults per second.)



Machine pool sizes and rates

9.17.6 Machine pool consumption breakdown

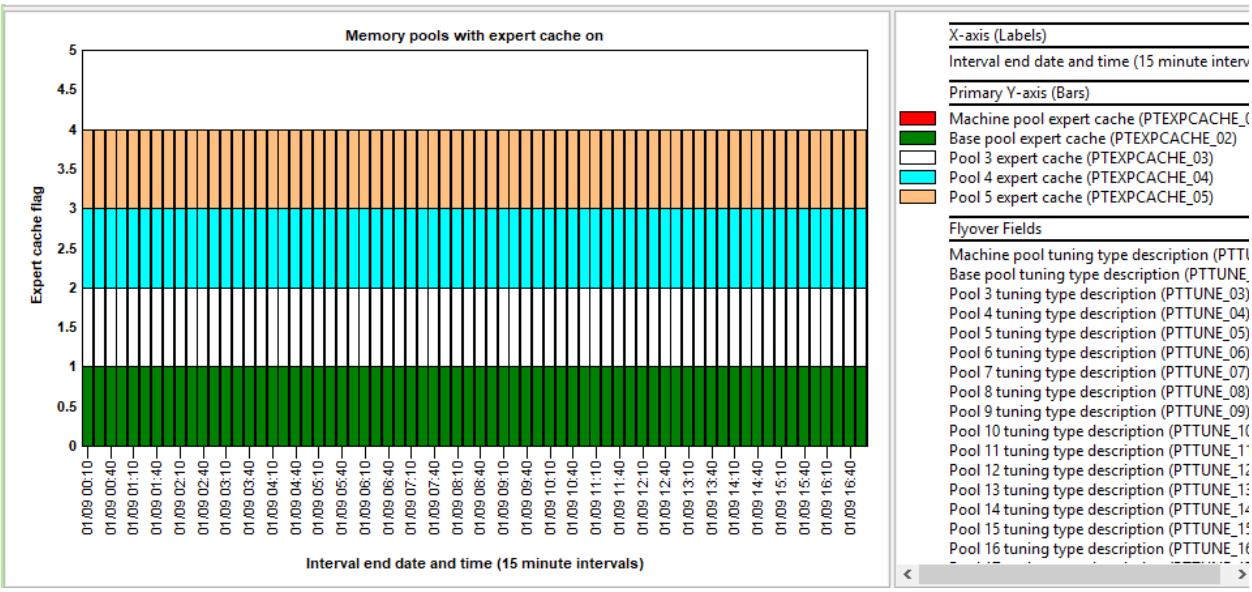
This graph displays the machine pool space consumed and unallocated memory together over time along with the faulting rate on the secondary Y-axis.



Machine pool consumption breakdown

9.17.7 Memory pools with expert cache on

This graph simply displays all the memory pools with expert cache turned on. This is also known as the "Calculated" pool tuning type.



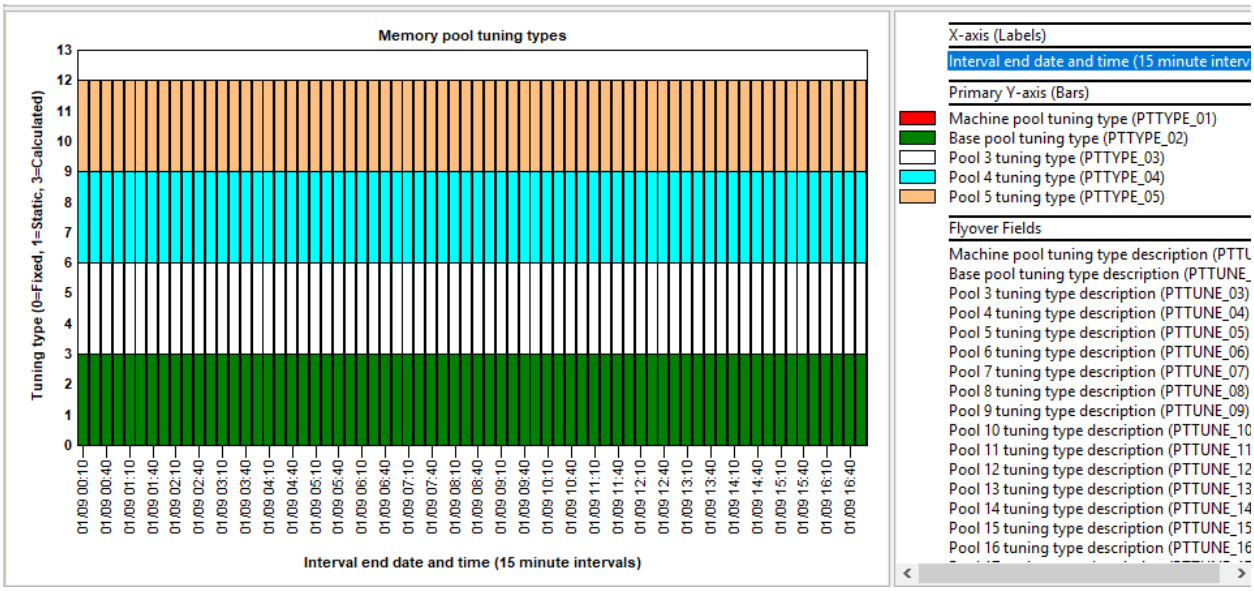
Memory pools with expert cache on

9.17.8 Memory pools tuning types

This graph is similar to the previous one except also shows the other possible memory pool tuning types besides expert cache/calculated. Each type is given a numeric value and graphed accordingly based on the type.

The types are:

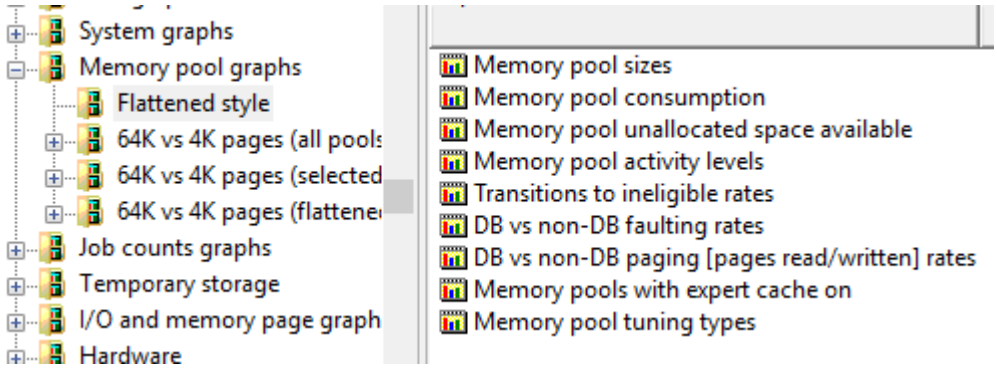
- 1 = static
- 2 = dynamic
- 3 = calculated / (i.e. expert cache on)



Memory pools tuning types

9.17.9 Flattened style

These graphs are like the previous set described except there are no limits to the number of memory pools they can show. Each pool will be given its own color.

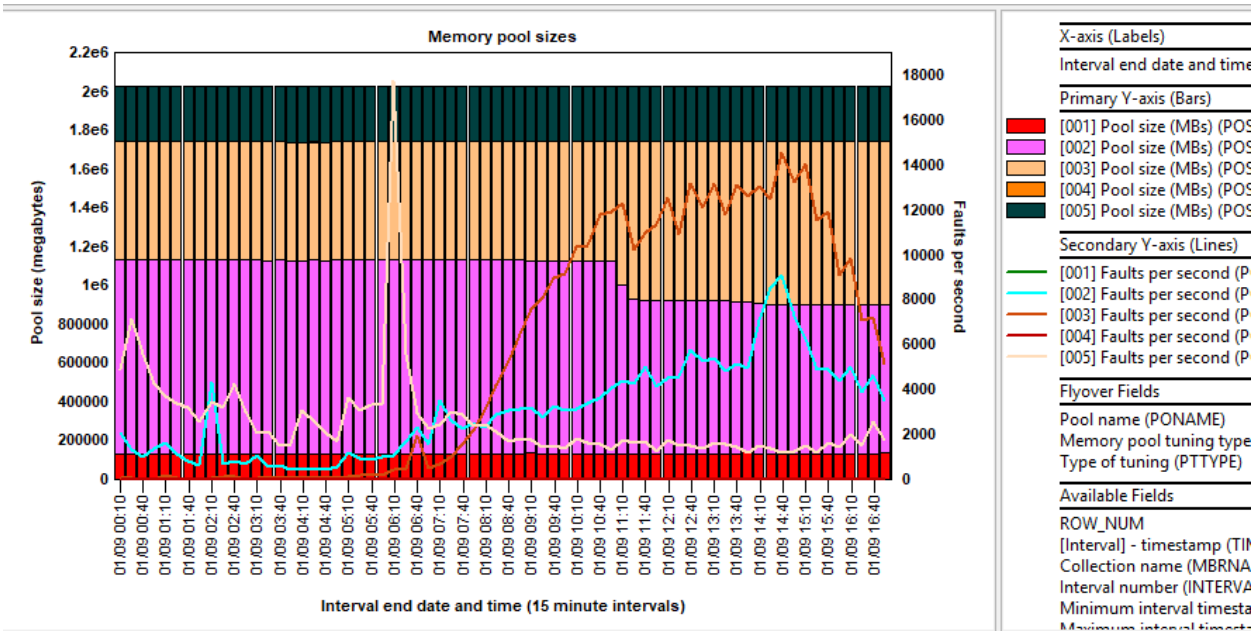


Memory pool graphs -> Flattened style

Tip: From these graphs you can right-click a pool to be able to graph any of the job statistics just for the desired pool. Use the “Selected Pool” menu for an over-time graph, and the “Rankings filtered by selected pool” to show a rankings graph filtered by the desired pool.

9.17.9.1 Memory pool sizes

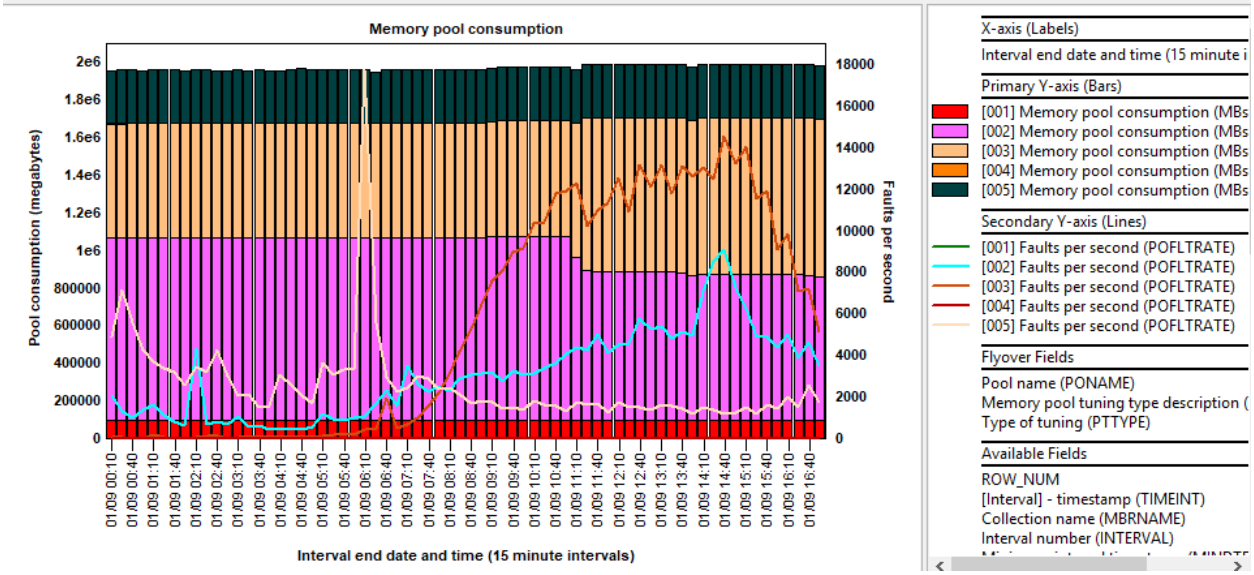
This graph displays the memory pool sizes (in megabytes) for each pool over time with the faults per second on the secondary Y-axis.



Memory pool sizes

9.17.9.2 Memory pool consumption

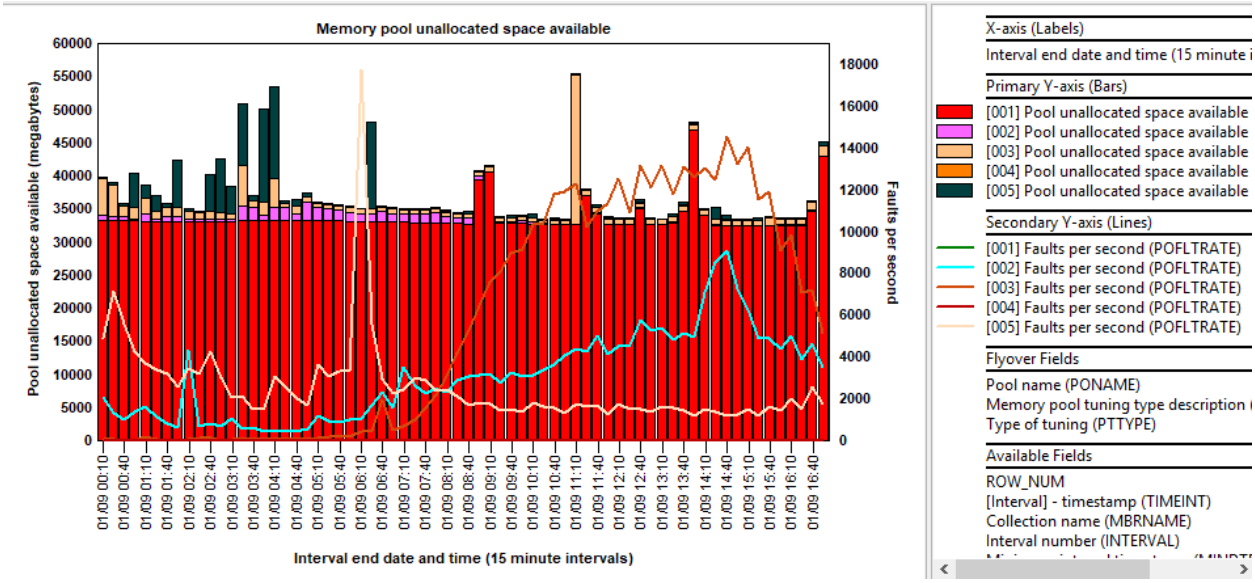
This graph displays the memory pool consumption (in megabytes) for each pool over time with the faults per second on the secondary Y-axis.



Memory pool consumption

9.17.9.3 Memory pool unallocated space available

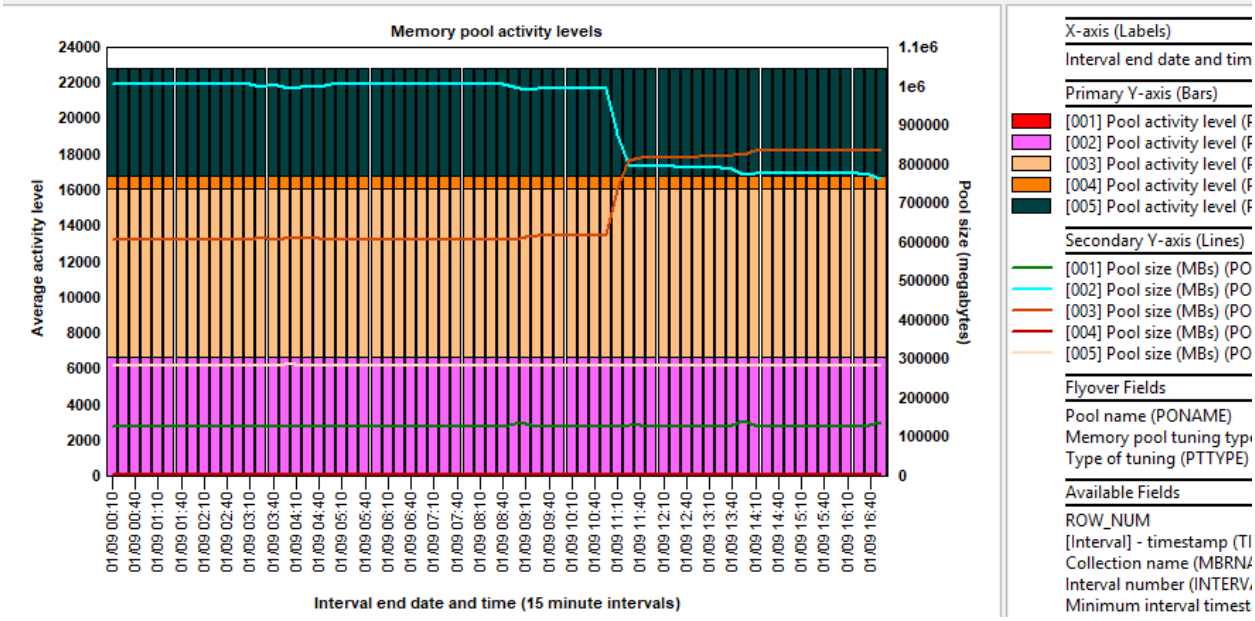
This graph displays the unallocated space available for each memory pool in megabytes. The secondary Y-axis displays the fault per second.



Memory pool unallocated space available

9.17.9.4 Memory pool activity levels

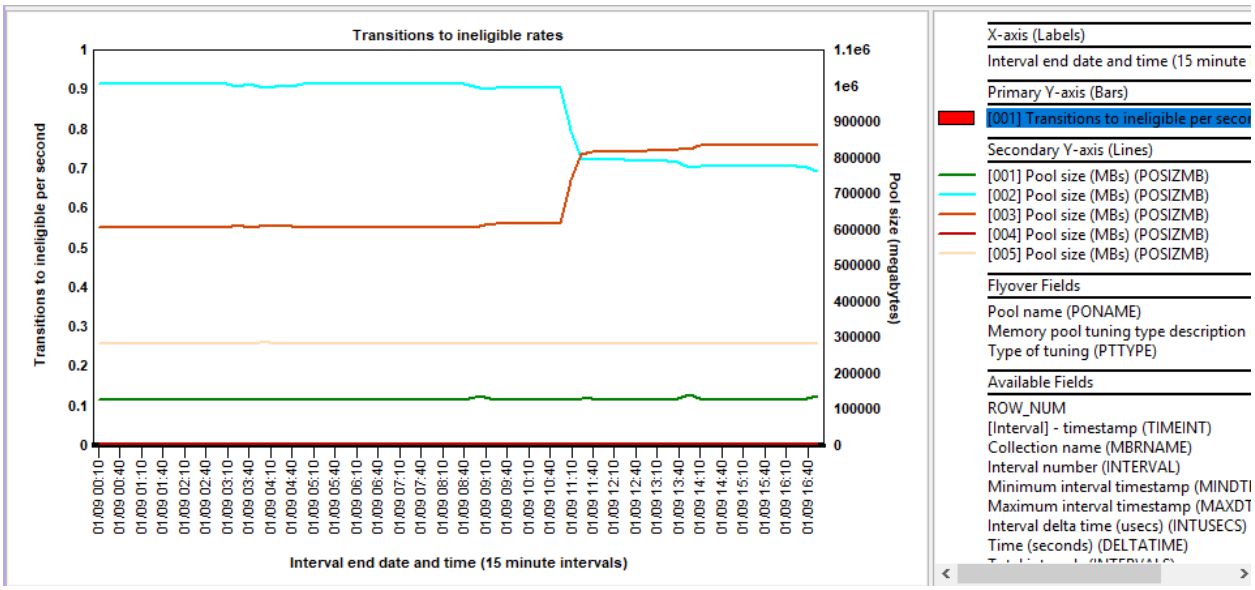
This graph displays the average pool activity level for all pools on the primary Y-axis and the pool sizes on the 2nd Y-axis.



Memory pool activity levels

9.17.9.5 Transitions to ineligible rates

This graph displays the rates of state transitions to ineligible per second.

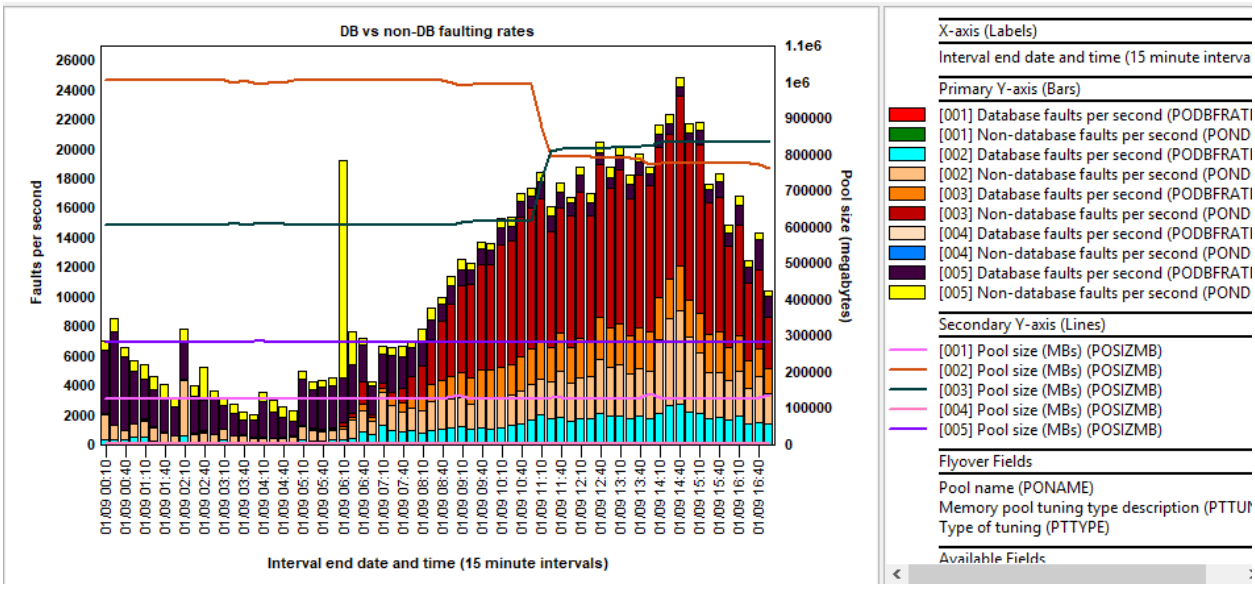


Transitions to ineligible rates

9.17.9.6 DB vs non-DB faulting rates

This graph shows for every pool:

1. Database faults per second
2. Non-database faults per second
3. Pool size (in megabytes) on the secondary Y-axis.

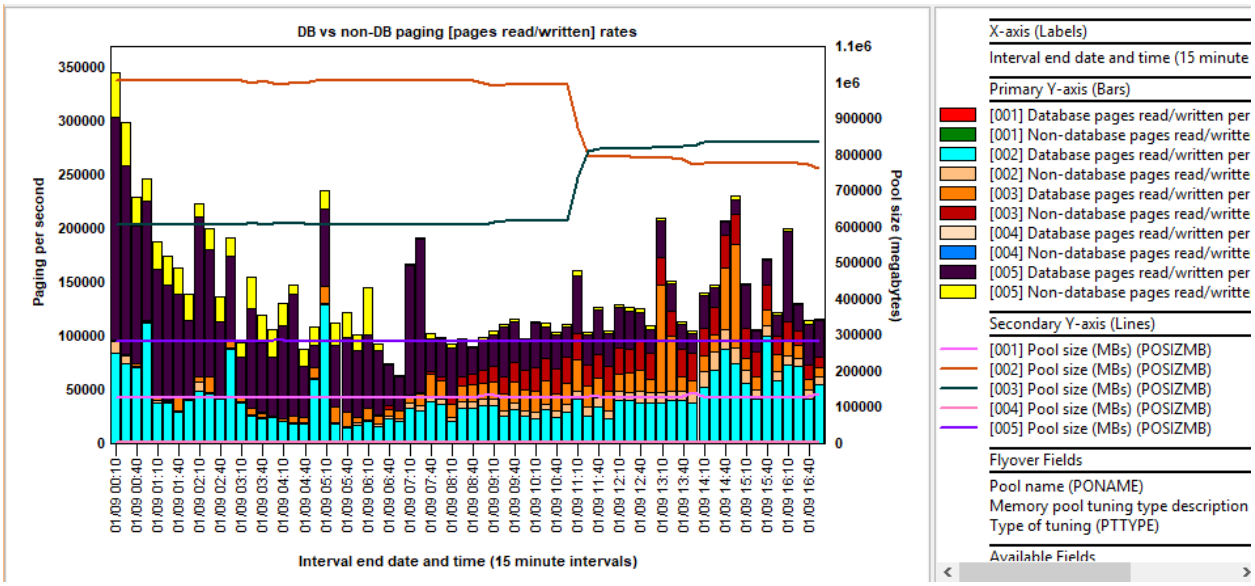


DB vs non-DB faulting rates

9.17.9.7 DB vs non-DB paging [pages read/written] rates

This graph shows for every pool:

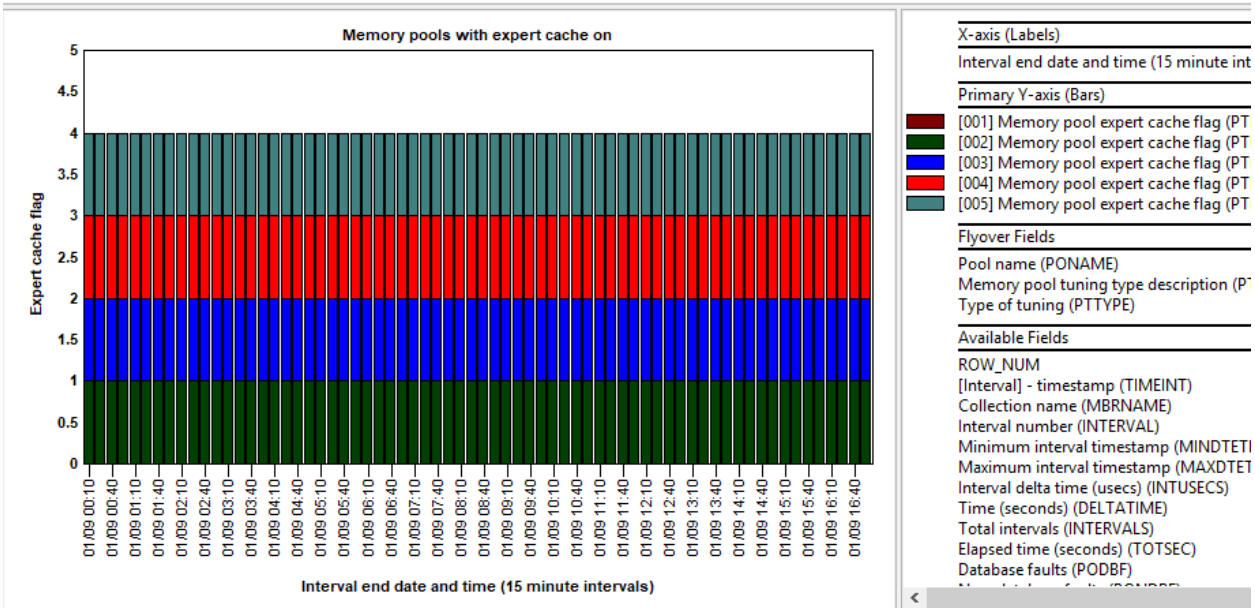
1. Database paging (pages read/written) per second
2. Non-database paging (pages read/written) per second
3. Pool size (in megabytes) on the 2nd Y-axis



DB vs non-DB paging [pages read/written] rates

9.17.9.8 Memory pools with expert cache on

This graph simply displays all the memory pools with expert cache turned on. This is also known as the "Calculated" pool tuning type.



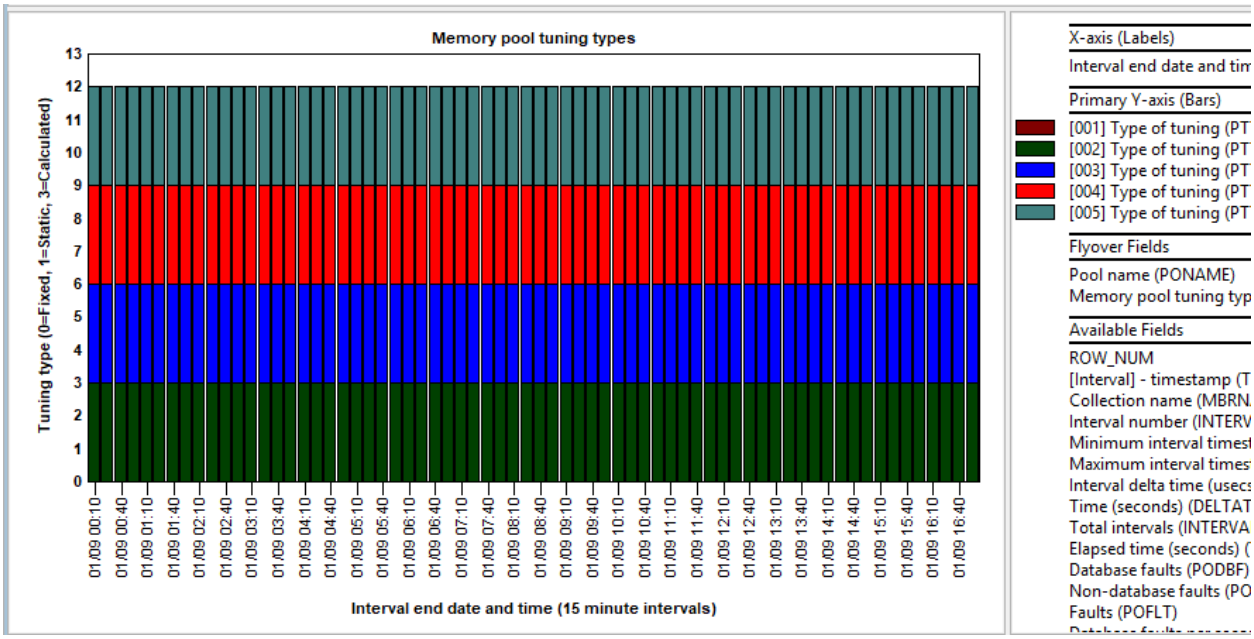
Memory pools with expert cache on

9.17.9.9 Memory pools tuning types

This graph is similar to the previous one except also shows the other possible memory pool tuning types besides expert cache/calculated. Each type is given a numeric value and graphed accordingly based on the type.

The types are:

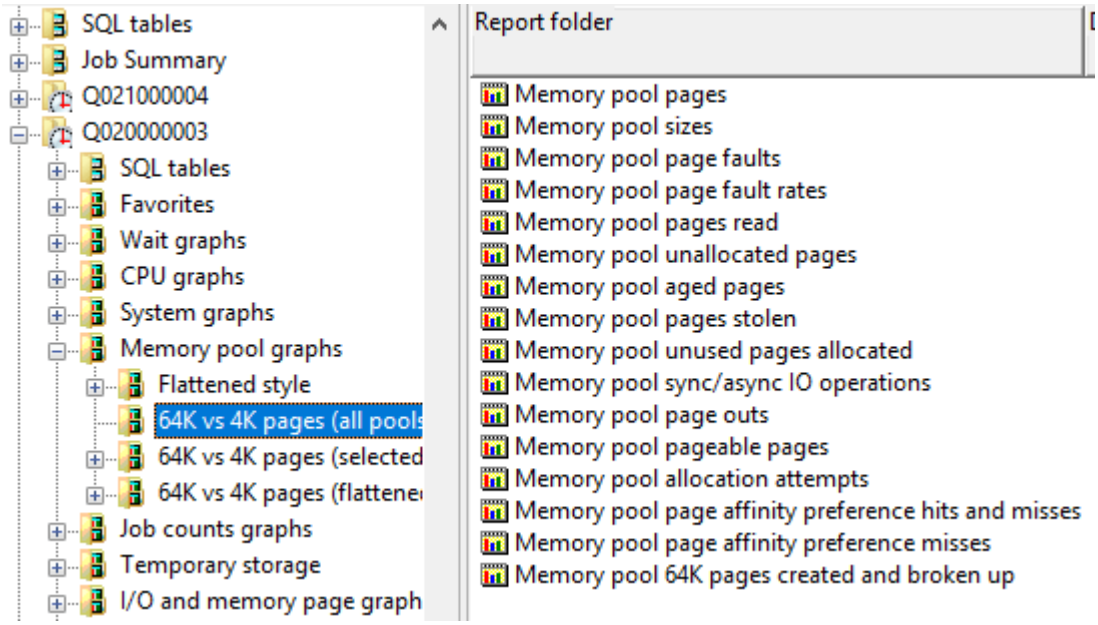
- 1 = static
- 2 = dynamic
- 3 = calculated / (i.e. expert cache on)



Memory pools tuning types

9.17.10 64K vs 4K pages (all pools)

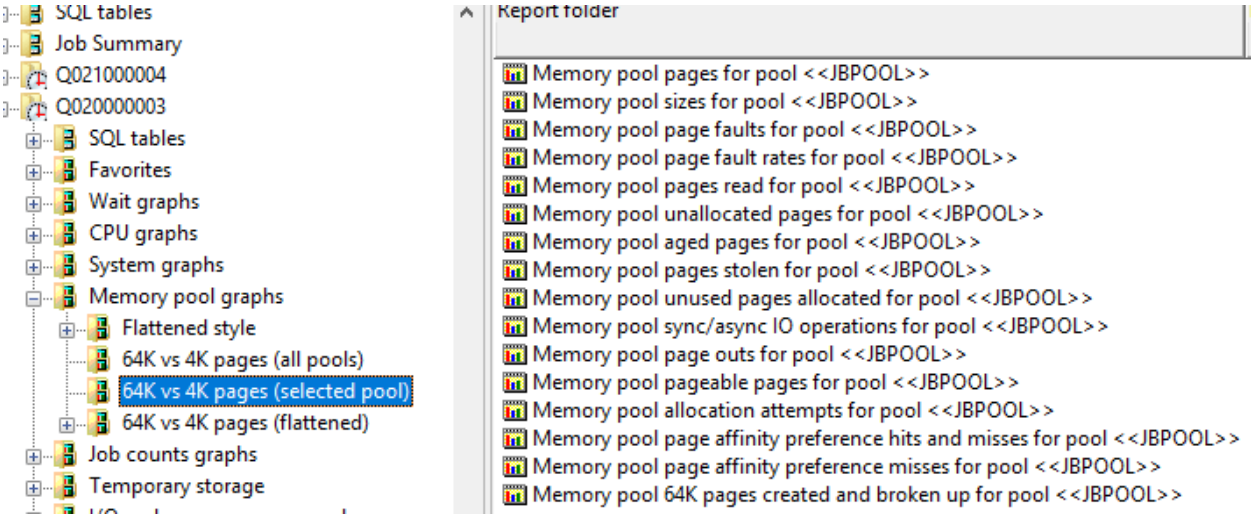
These graphs are generally intended for IBM service use only. They compare the use of 64K vs 4K page usage in all the memory pools added up together.



Memory pool graphs -> 64K vs 4K pages (all pools)

9.17.11 64K vs 4K pages (selected pool)

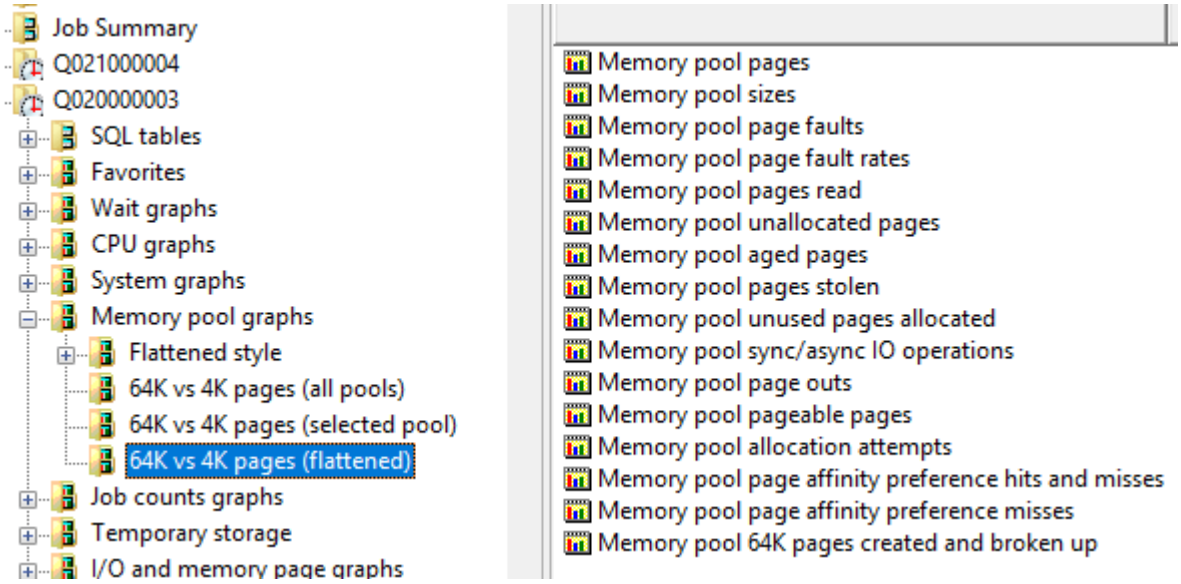
These graphs are generally intended for IBM service use only. They compare the use of 64K vs 4K page usage in the desired memory pool. Upon opening one of these graphs you will be prompted to enter the desired memory pool number.



Memory pool graphs -> 64K vs 4K pages (selected pool)

9.17.12 64K vs 4K pages (flattened)

These graphs are generally intended for IBM service use only. They compare the use of 64K vs 4K page usage in all memory pools, but each will be graphed with different colors.

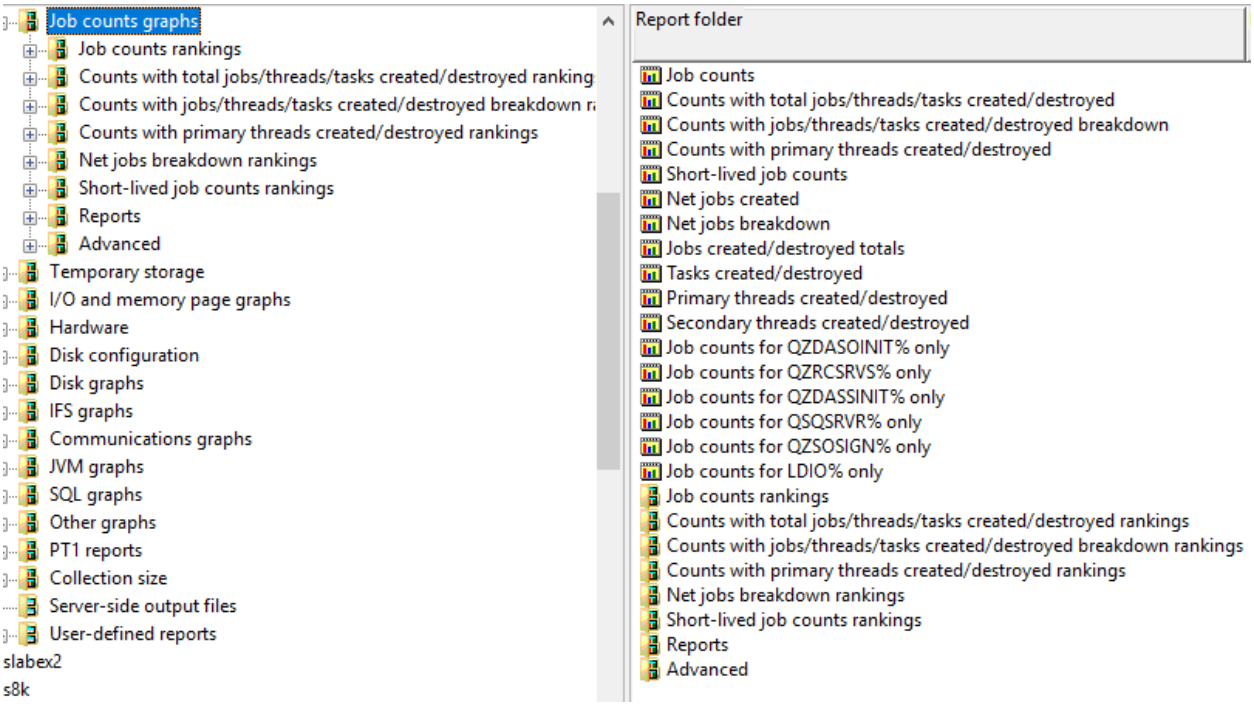


Memory pool graphs -> 64K vs 4K pages (flattened)

9.18 Job counts graphs

These graphs display the number of jobs/tasks/threads that exist on the system in several different ways. The Job counts rankings subfolder lets you generate list of job counts by a job grouping (by job, by job user, generic job, etc.)

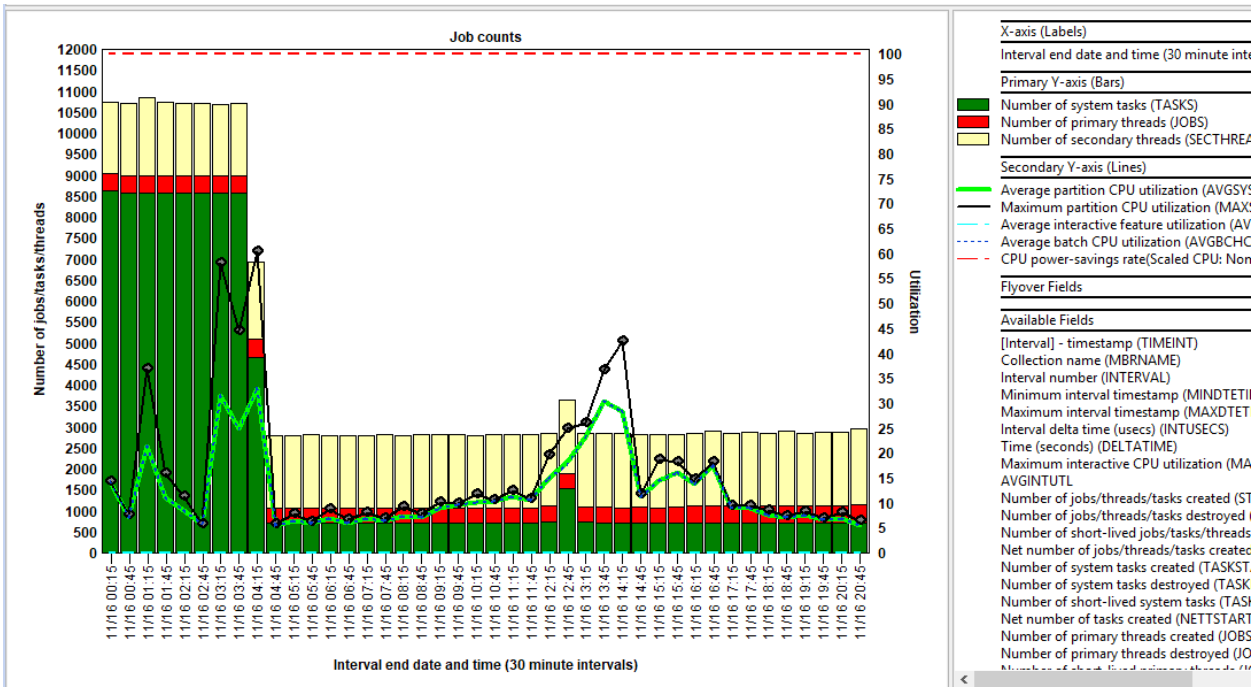
Note: This folder will only appear if the Collection Summary analysis has been executed.



Job counts graphs

9.18.1 Job counts

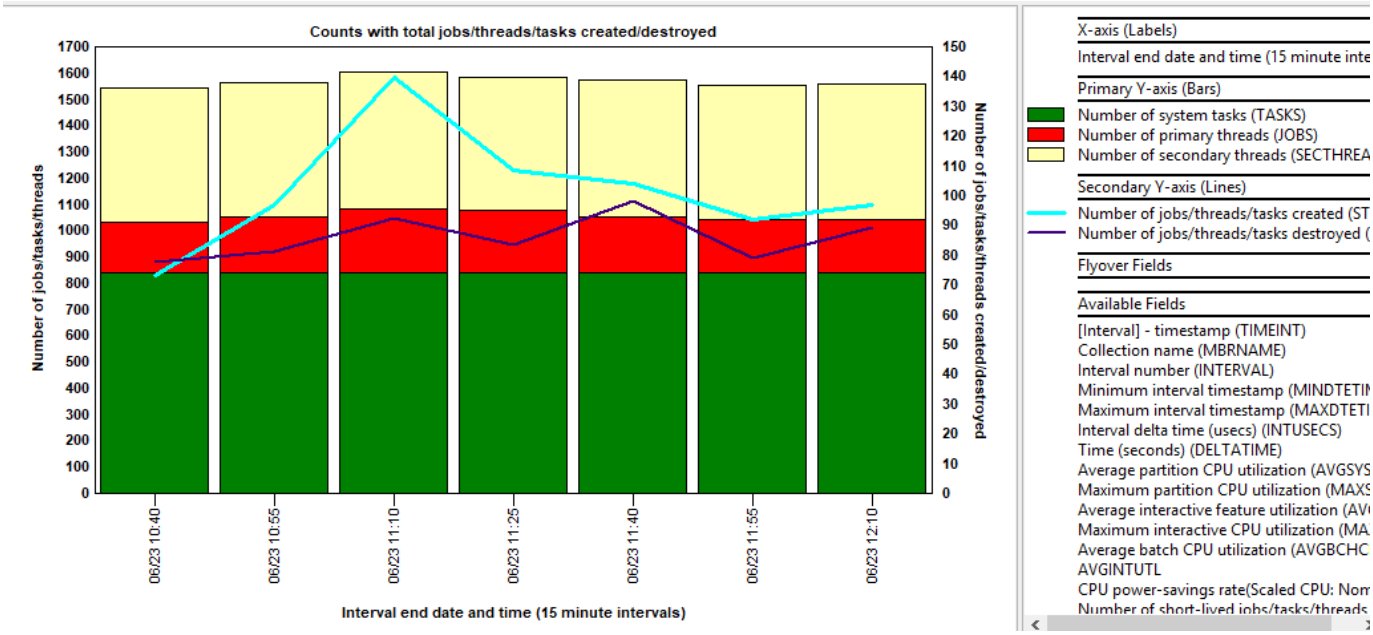
This graph simply shows the number of system tasks, processes (primary threads) and secondary threads over time. The secondary Y-axis displays the CPU utilization fields.



Job counts

9.18.2 Counts with total jobs/threads/tasks created/destroyed

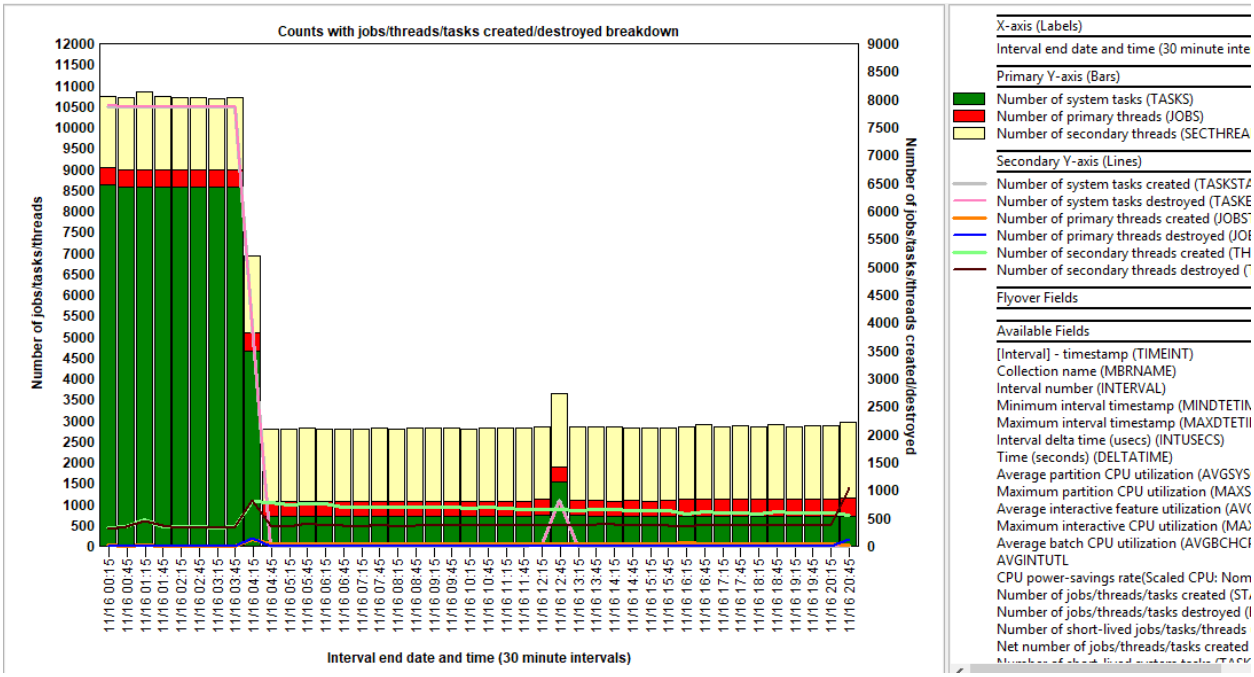
This graph is the same as the previous one except the secondary Y-axis will instead contain the total number of jobs/tasks/threads created and destroyed.



Counts with total jobs/threads/tasks created/destroyed

9.18.3 Counts with jobs/threads/tasks created/destroyed breakdown

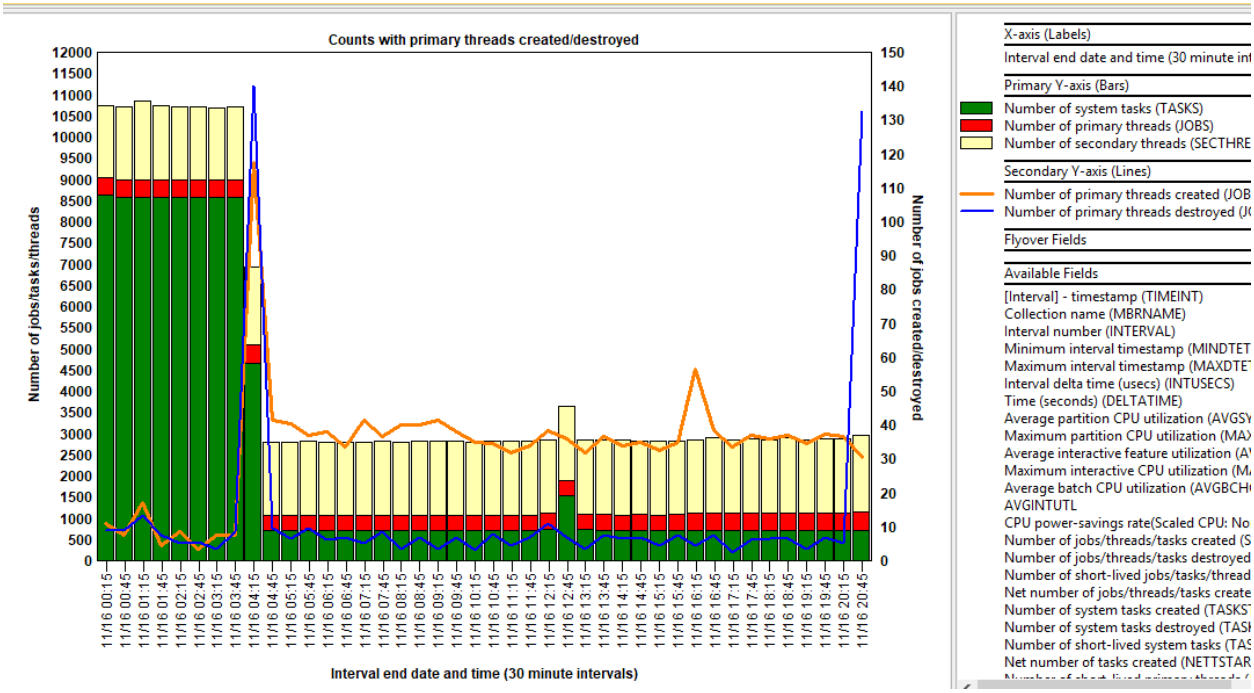
This graph is the same as the prior one except breaks out the created / destroyed counts into 3 categories: system tasks, primary threads and secondary threads.



Counts with jobs/threads/tasks created/destroyed breakdown

9.18.4 Counts with primary threads created/destroyed

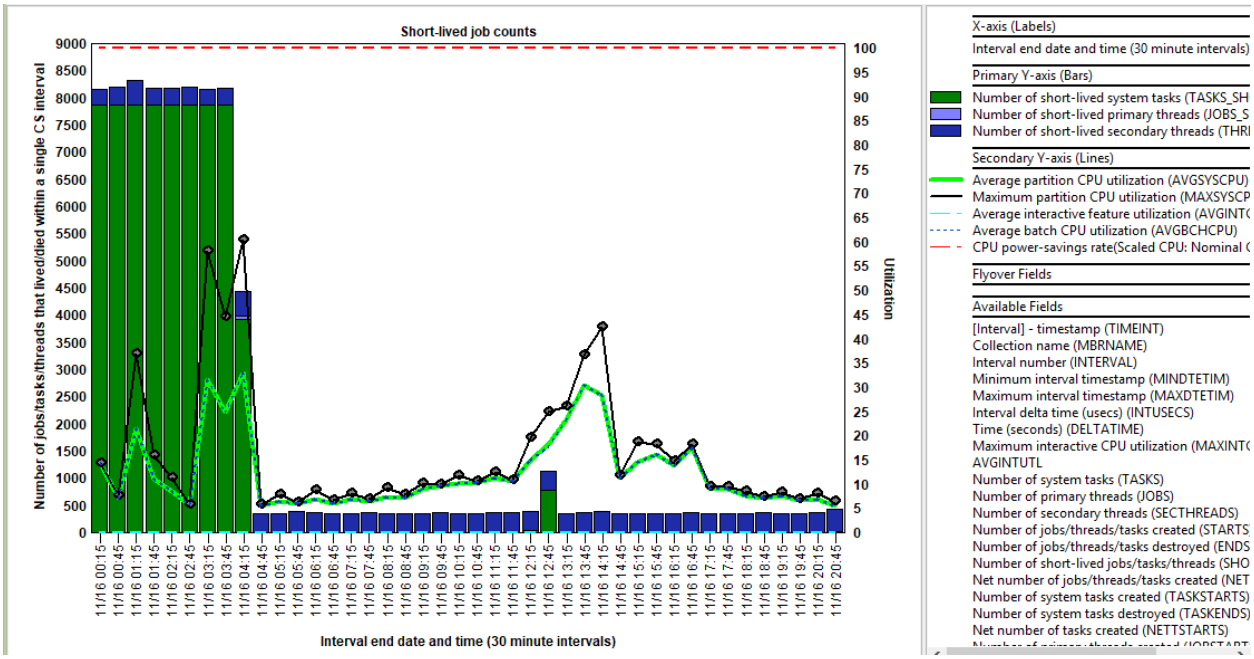
This graph is like the previous one except the secondary Y-axis will only contain the counts for the primary threads created and destroyed.



Counts with primary threads created/destroyed

9.18.5 Short-lived job counts

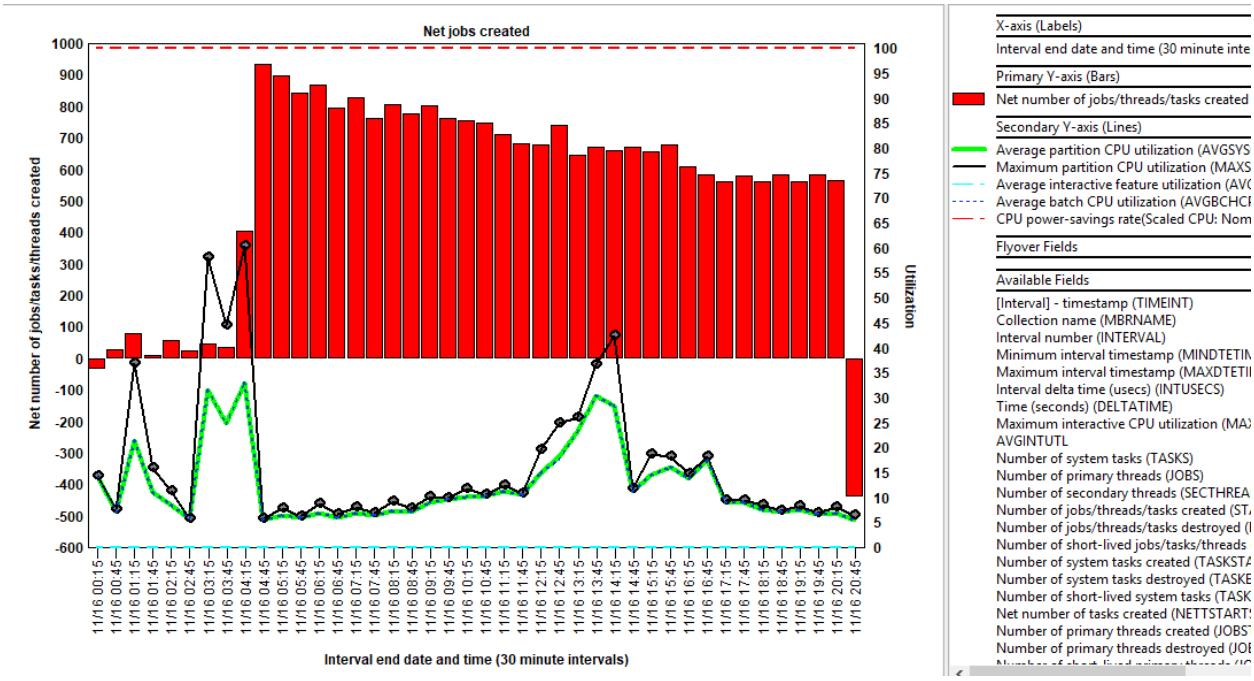
This graph displays all jobs/tasks/threads that only survived for 1 collection services data collection interval or less. This varies but is often set to 5 minutes or 15 minutes.



Short-lived job counts

9.18.6 Net jobs created

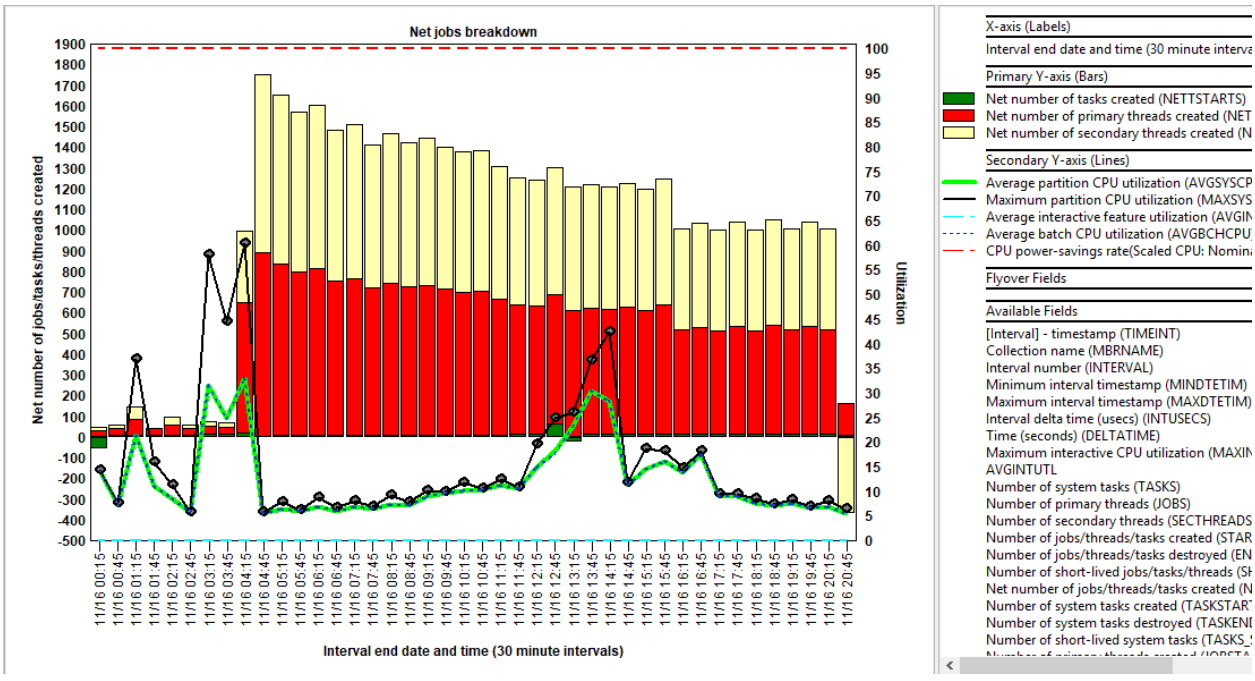
This graph displays the overall net number of jobs/tasks/threads created over time.



Net jobs created

9.18.7 Net jobs breakdown

This graph is the same as the previous graph except the net created values for each of tasks, jobs and secondary threads are shown separately.

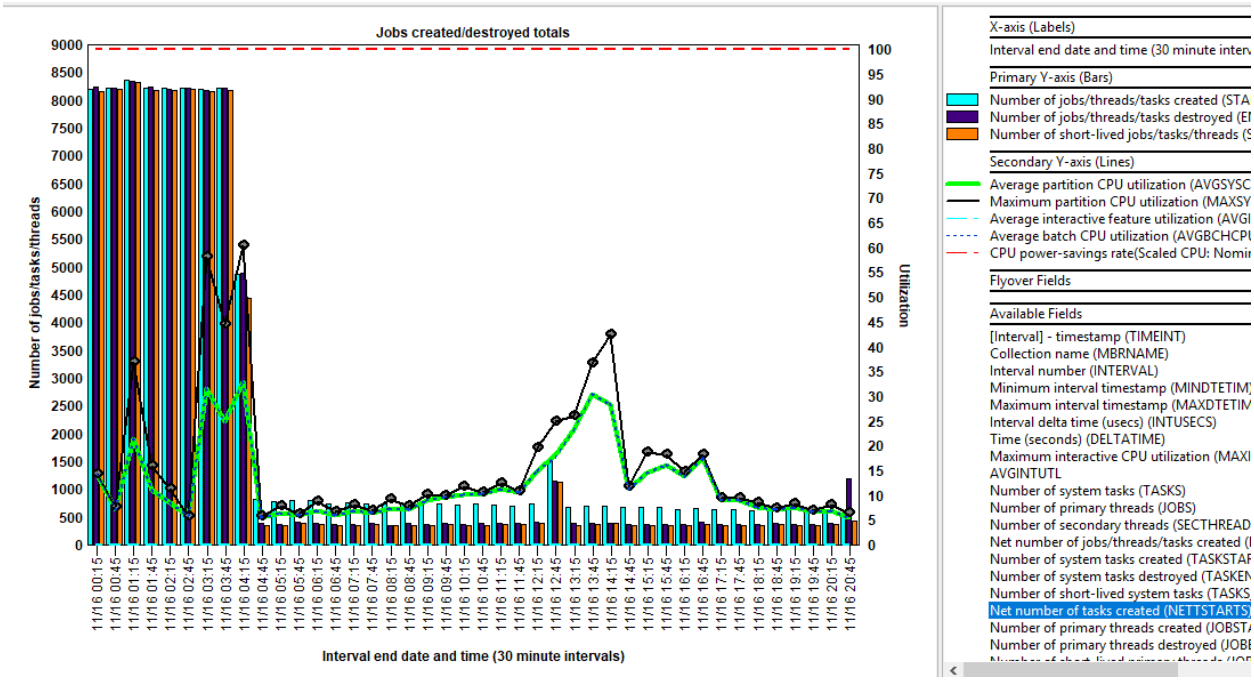


Net jobs breakdown

9.18.8 Jobs created/destroyed totals

This graph displays the following 3 values with side-by-side vertical bars per interval:

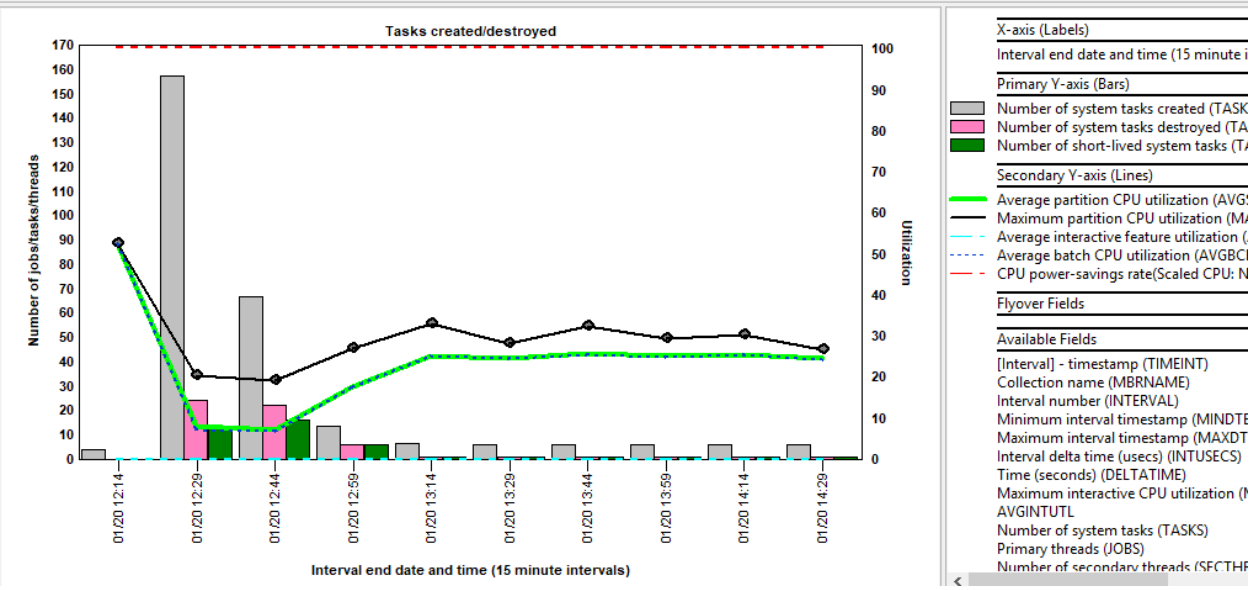
1) Jobs/threads/tasks created
2) Jobs/threads/tasks destroyed
3) Short-lived jobs/tasks/threads (**Note:** These are also included in the totals for #1 and #2.)



Jobs created/destroyed totals

9.18.9 Tasks created/destroyed

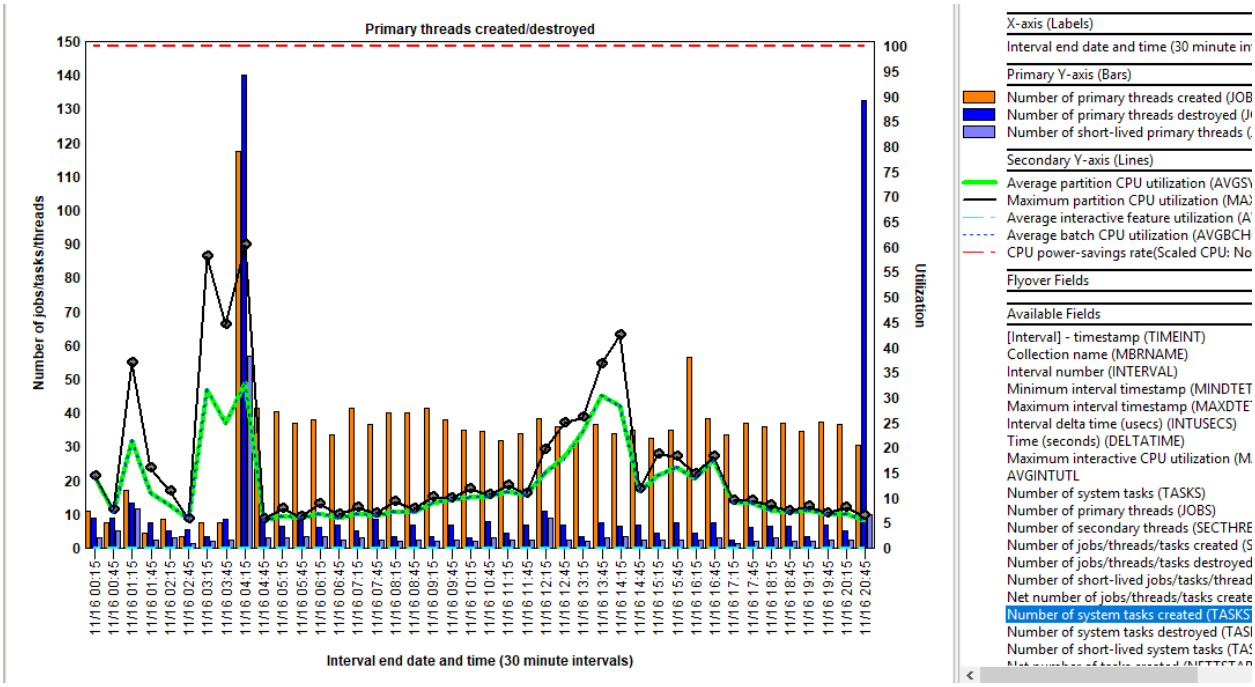
This graph is the same as the Jobs created/destroyed totals graph but only includes numbers for system tasks.



Tasks created/destroyed

9.18.10 Primary threads created/destroyed

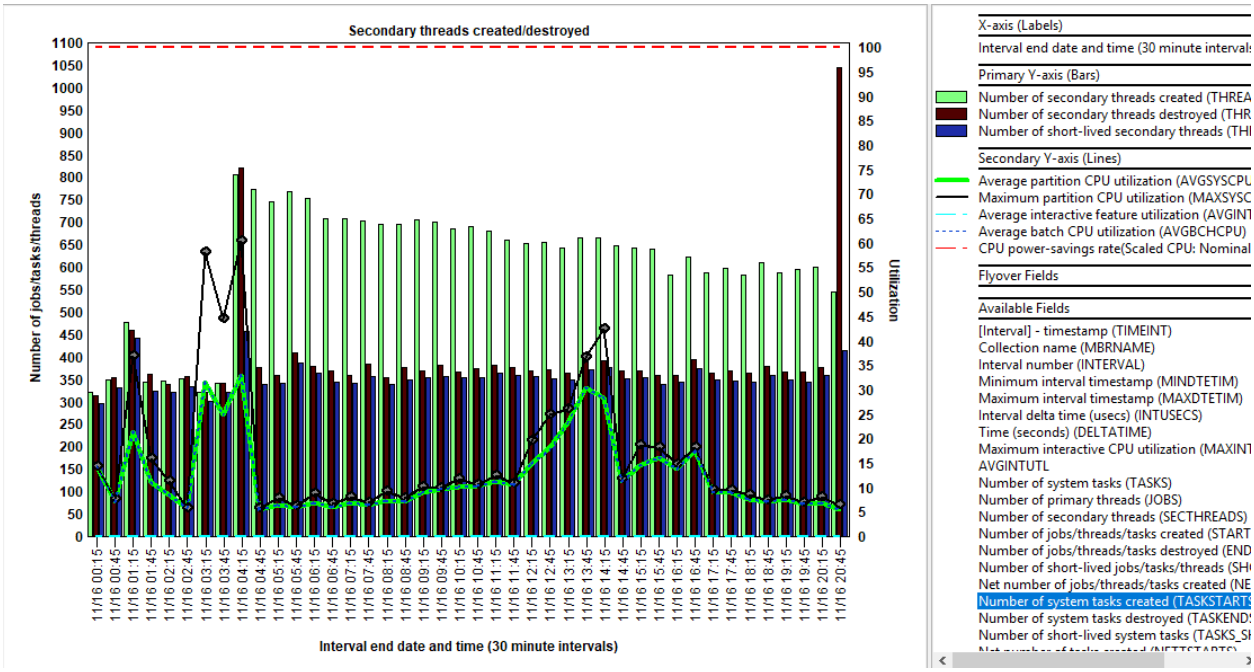
This graph is the same as the Jobs created/destroyed totals graph but only includes numbers for primary threads (i.e. processes.)



Primary threads created/destroyed

9.18.11 Secondary threads created/destroyed

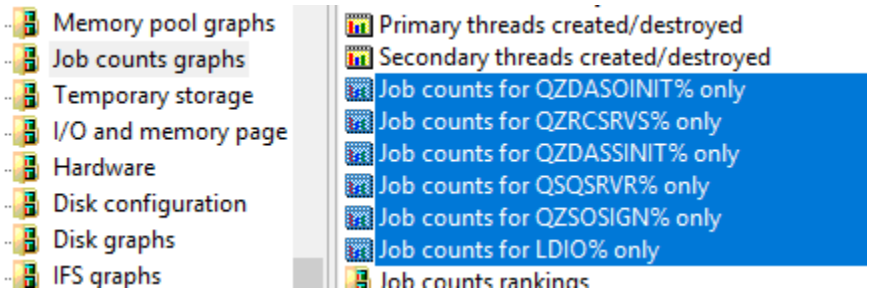
This graph is the same as the Jobs created/destroyed totals graph but only includes numbers for secondary threads.



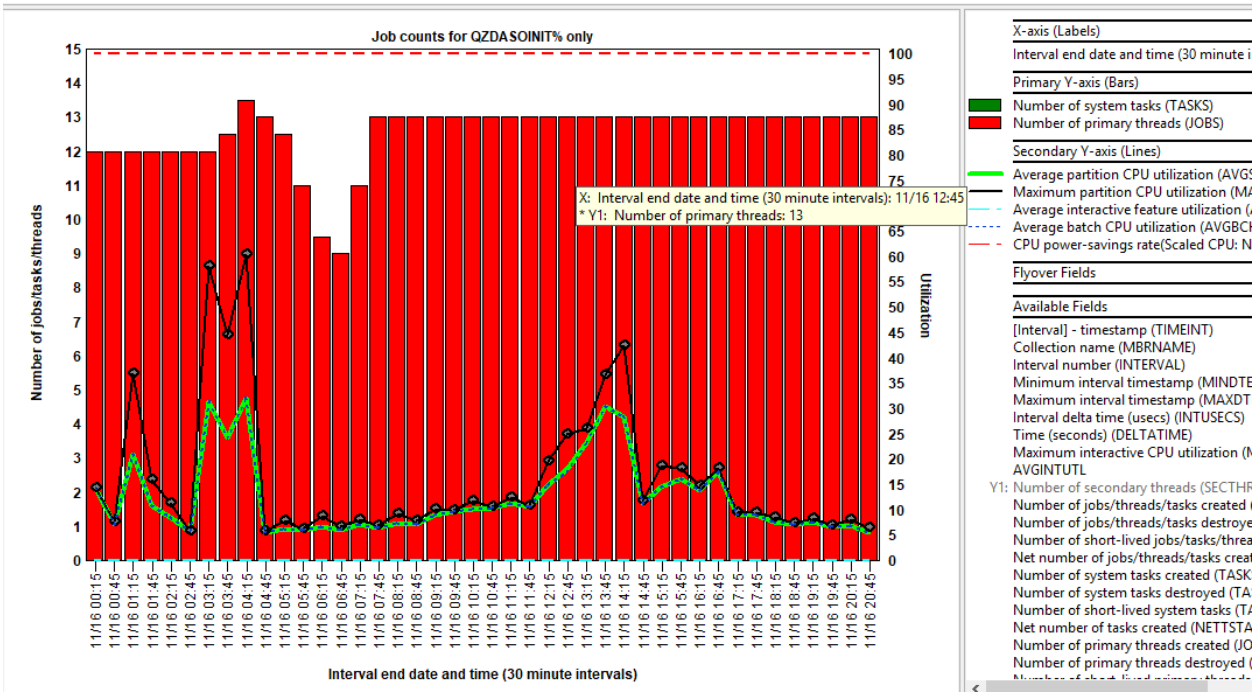
Secondary threads created/destroyed

9.18.12 Job counts for <JOBNAME%> only

These graphs include the following options in the Job counts graphs folder and filter by the specified generic job name:



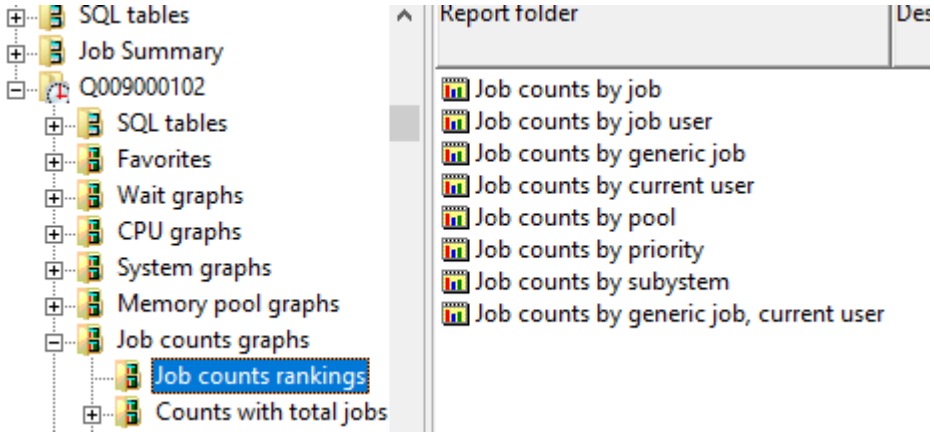
Job counts graphs -> Job counts for <JOBNAME%> only graphs



Job counts for QZDASOINIT% only

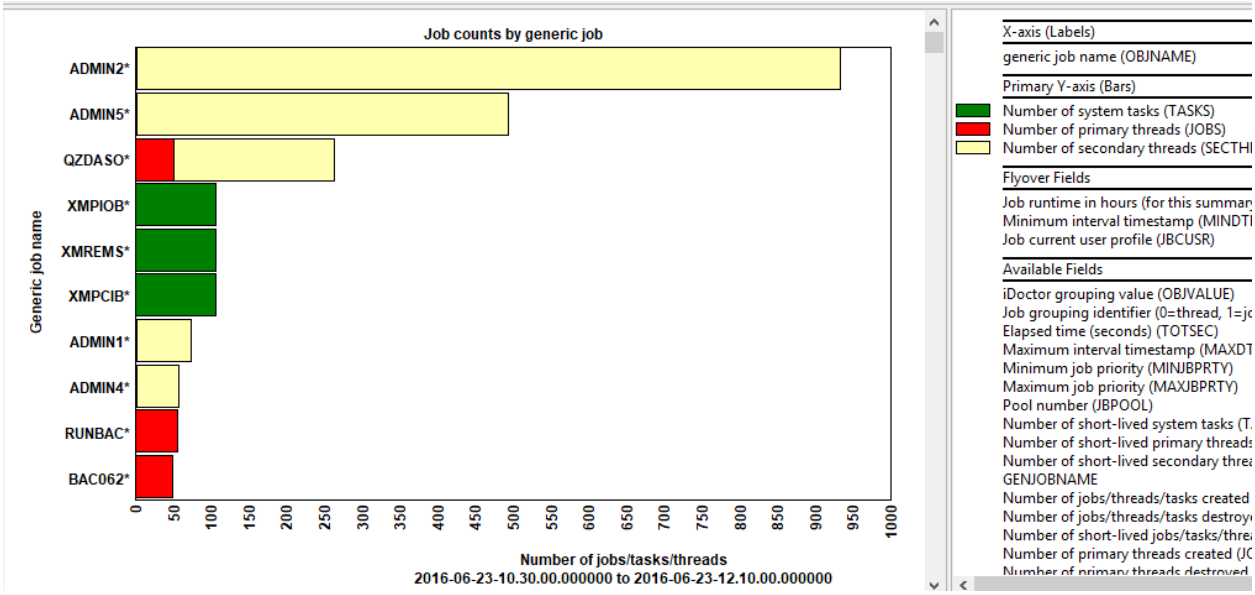
9.18.13 Job counts rankings

These graphs show the total job counts for the job groupings shown below:



Job counts graphs -> Job counts rankings

An example is shown below:

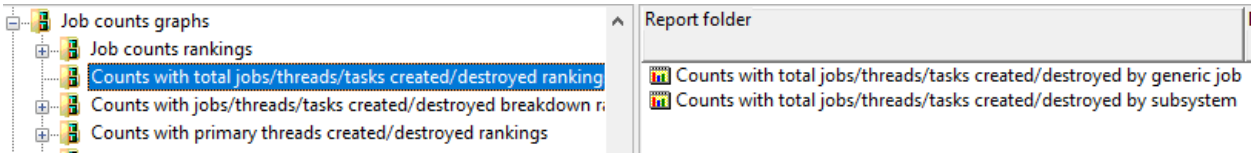


Job counts by generic job

Tip: These graphs show data for the entire collection, but you can also drill down into these from the previous graphs in the Job counts folder.

9.18.14 Counts with total jobs/threads/tasks created/destroyed rankings

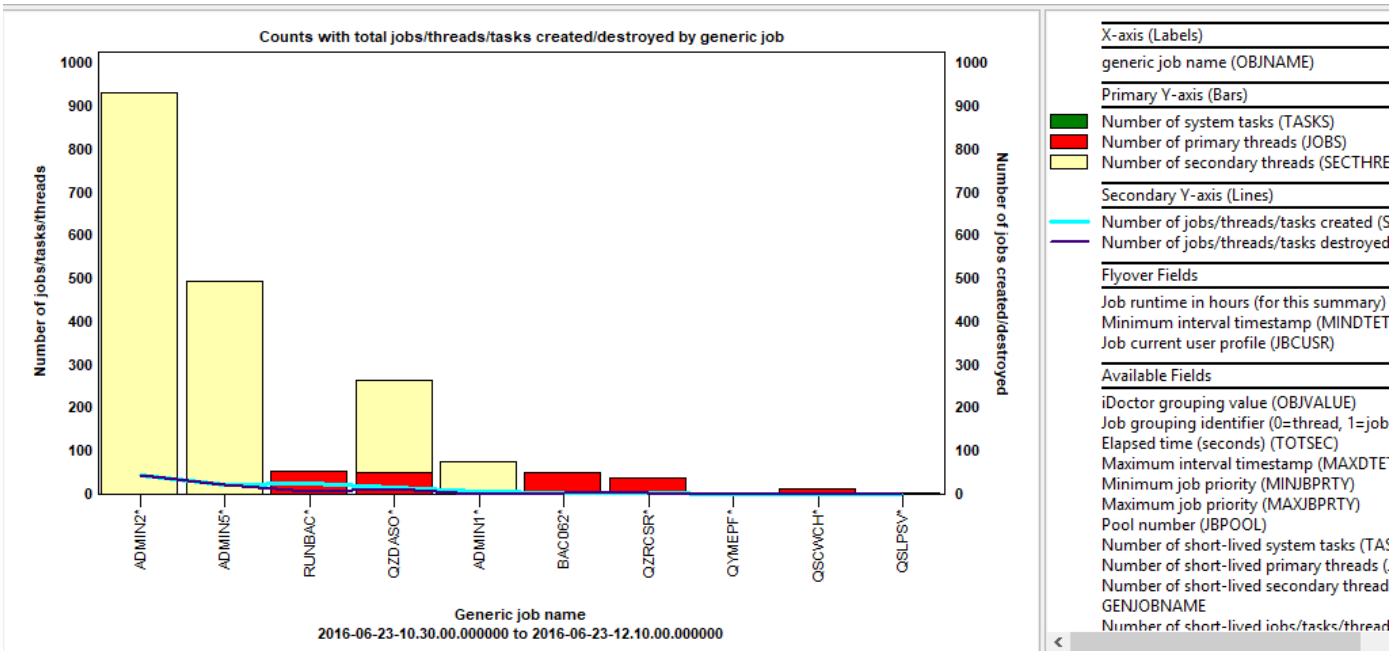
These graphs rank the job counts for job/threads/tasks and also display the total jobs created/destroyed on the secondary Y-axis. These graphs are also provided as a drill down from the Job counts -> Counts with total jobs/threads/tasks created/destroyed graph.



Job counts graphs -> Counts with total jobs/threads/tasks created/destroyed rankings

9.18.14.1 Counts with total jobs/threads/tasks created destroyed by generic job

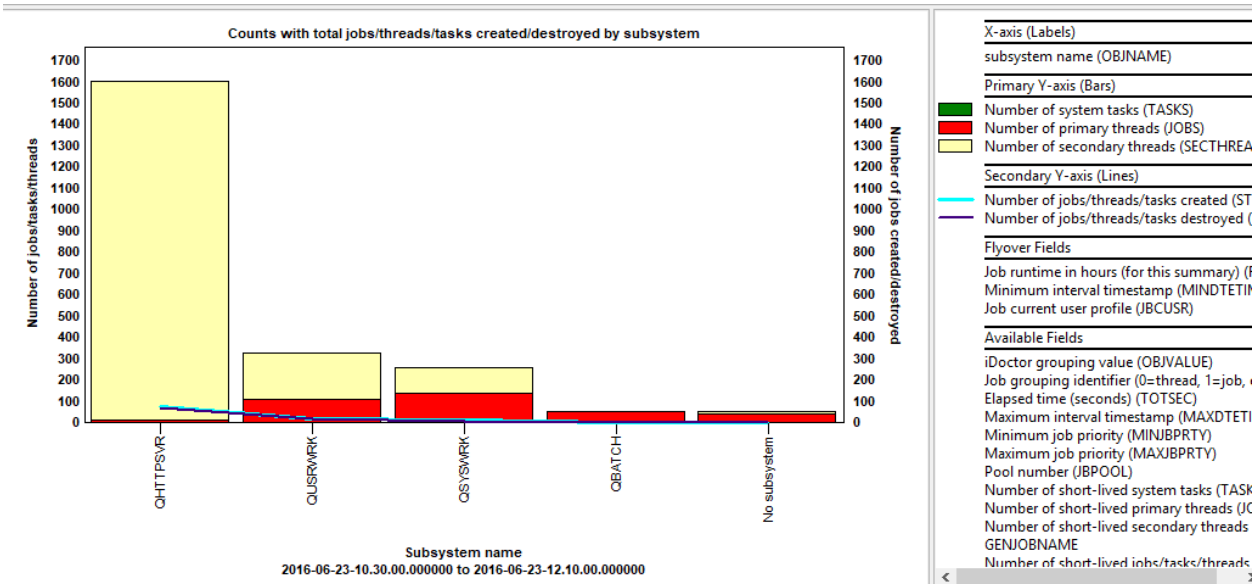
This graph shows the total jobs/threads/tasks for each generic job name along with the total created and destroyed on the secondary Y-axis.



Counts with total jobs/threads/tasks created/destroyed by generic job

9.18.14.2 Counts with jobs/threads/tasks created/destroyed by subsystem

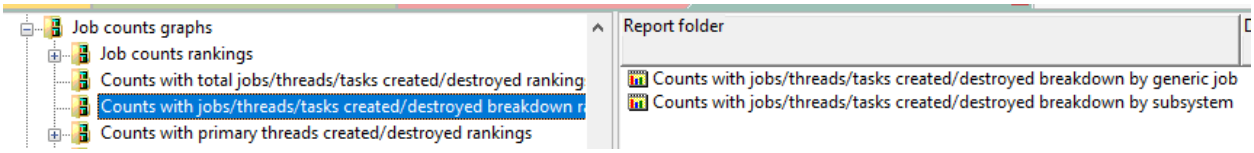
This graph shows the total jobs/threads/tasks for each subsystem along with the total created and destroyed on the secondary Y-axis.



Counts with total jobs/threads/tasks created/destroyed by subsystem

9.18.15 Counts with jobs/threads/tasks created/destroyed breakdown rankings

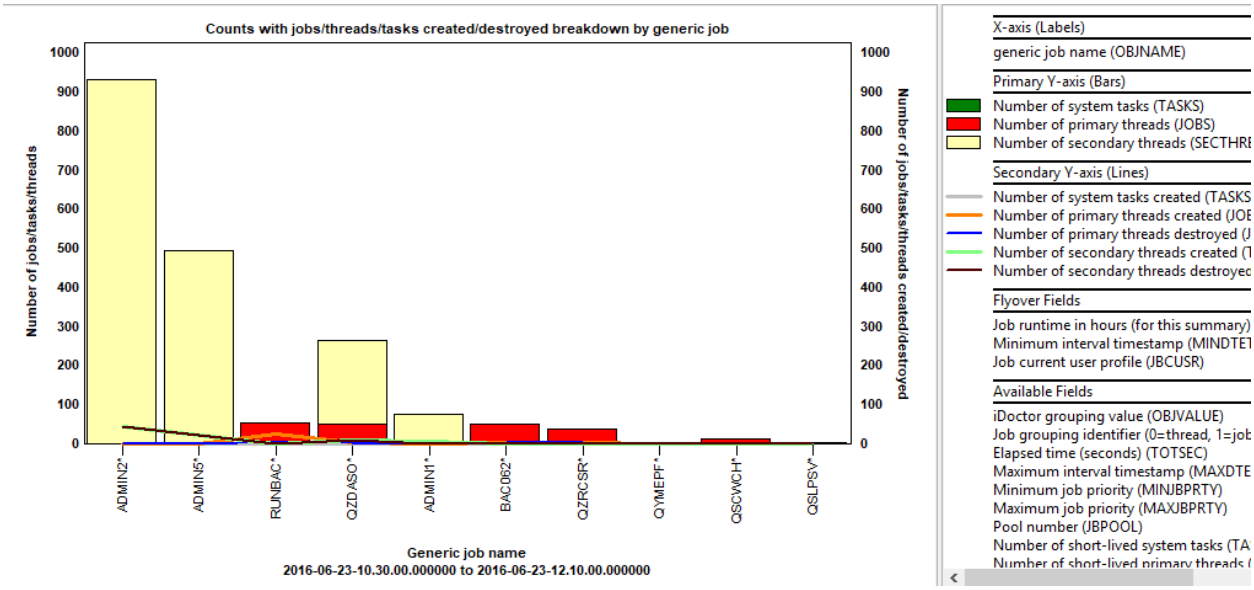
These graphs rank the job counts for job/threads/tasks and displays each of jobs, tasks and threads created/destroyed on the secondary Y-axis. These graphs are also provided as a drill down from the Job counts -> Counts with jobs/threads/tasks created/destroyed breakdown graph.



Job counts graphs -> Counts with jobs/threads/tasks created/destroyed breakdown rankings

9.18.15.1 Counts with jobs/threads/tasks created/destroyed breakdown by generic job

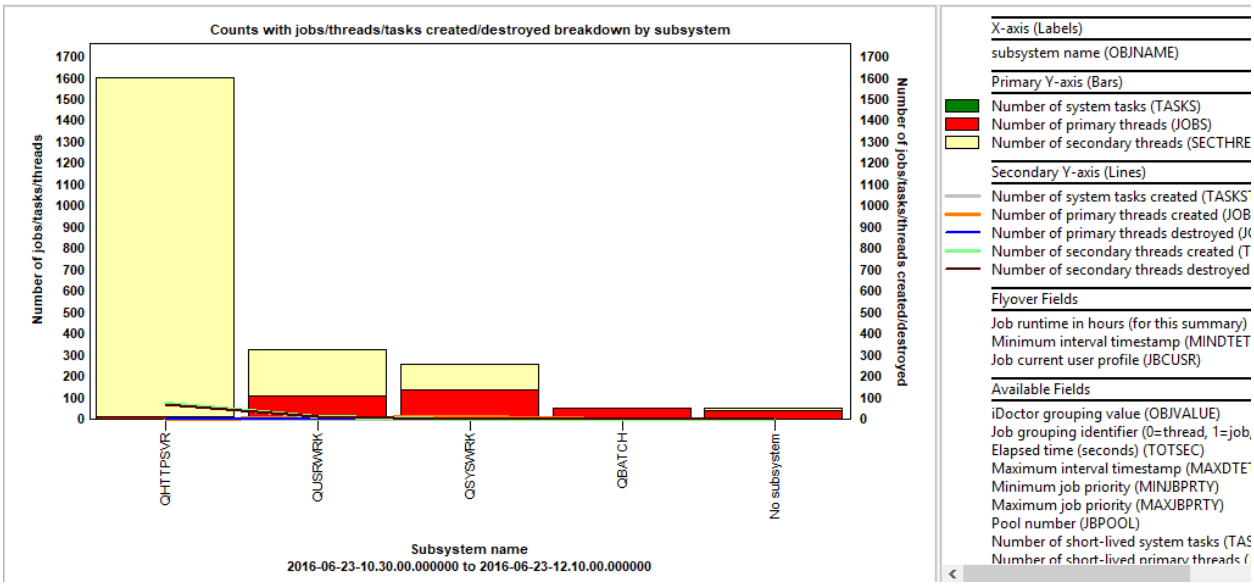
This graph shows the total jobs/threads/tasks for each generic job name and displays each of jobs, tasks and threads created/destroyed on the secondary Y-axis.



Counts with jobs/threads/tasks created/destroyed breakdown by generic job

9.18.15.2 Counts with jobs/threads/tasks created/destroyed by subsystem

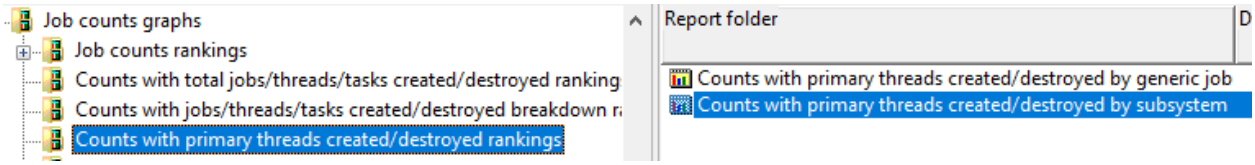
This graph shows the total jobs/threads/tasks for each subsystem and displays each of jobs, tasks and threads created/destroyed on the secondary Y-axis.



Counts with jobs/threads/tasks created/destroyed breakdown by subsystem

9.18.16 Counts with primary threads created/destroyed rankings

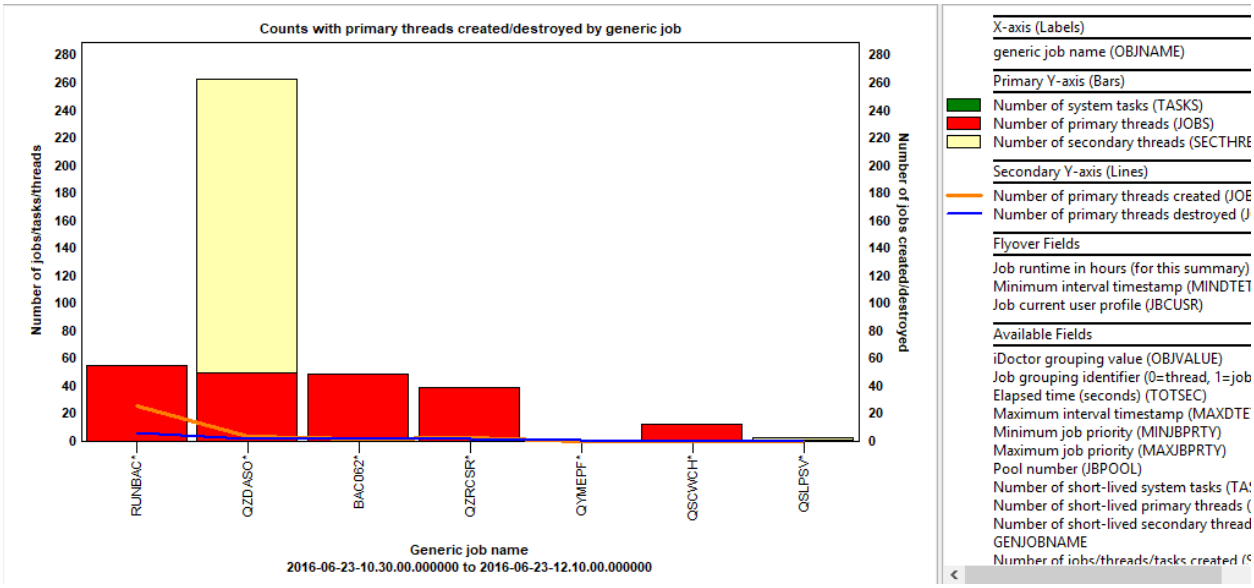
These graphs rank the job counts for job/threads/tasks and shows the primary threads created/destroyed on the secondary Y-axis. These graphs are also provided as a drill down from the Job counts -> Counts with primary threads created/destroyed graph.



Job counts graphs -> Counts with primary threads created/destroyed rankings

9.18.16.1 Counts with primary threads created destroyed by generic job

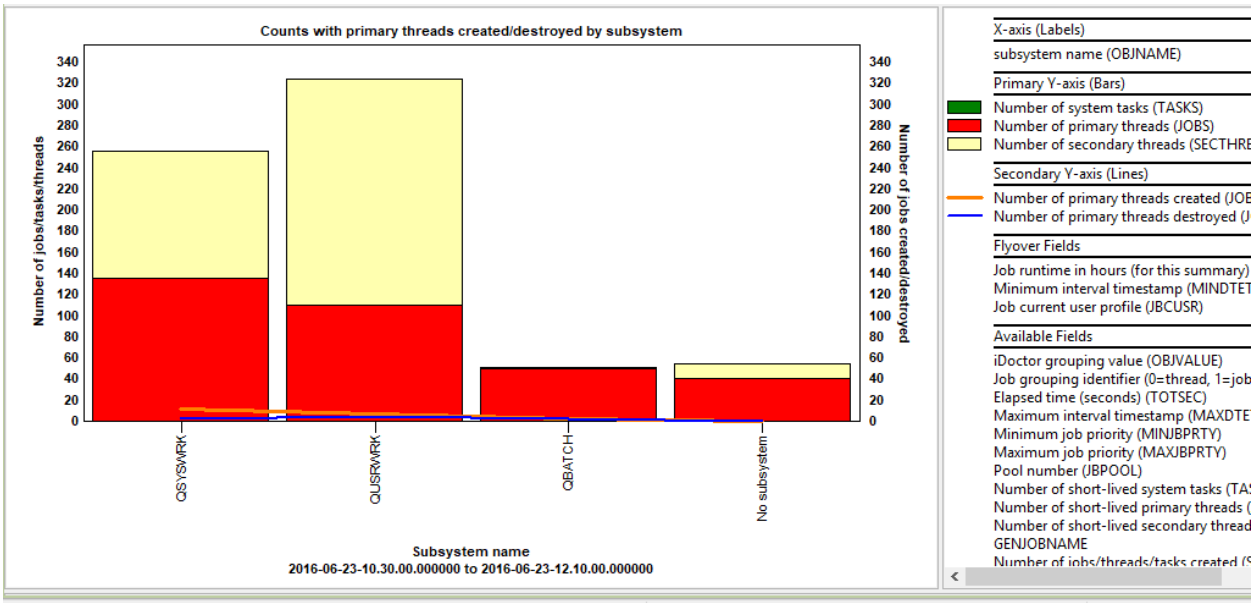
This graph shows the total jobs/threads/tasks for each generic job name along with the total primary threads created and destroyed on the secondary Y-axis.



Counts with primary threads created/destroyed by generic job

9.18.16.2 Counts with jobs/threads/tasks created/destroyed by subsystem

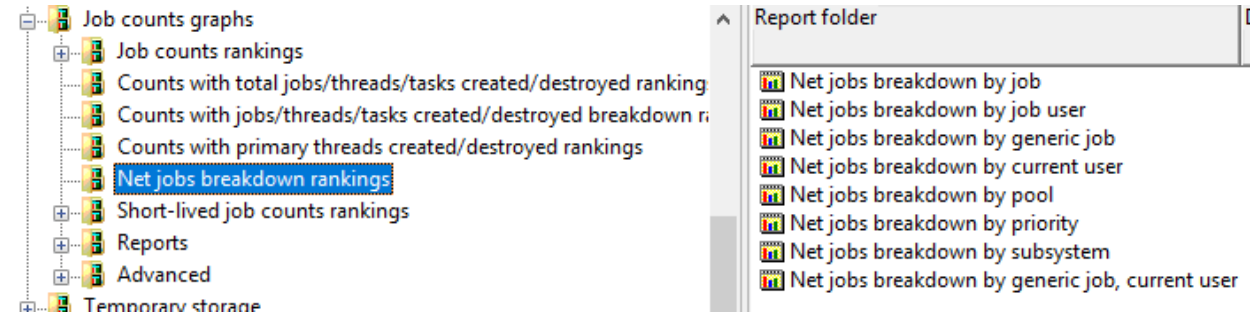
This graph shows the total jobs/threads/tasks for each subsystem along with the total primary threads created and destroyed on the secondary Y-axis.



Counts with primary threads created/destroyed by subsystem

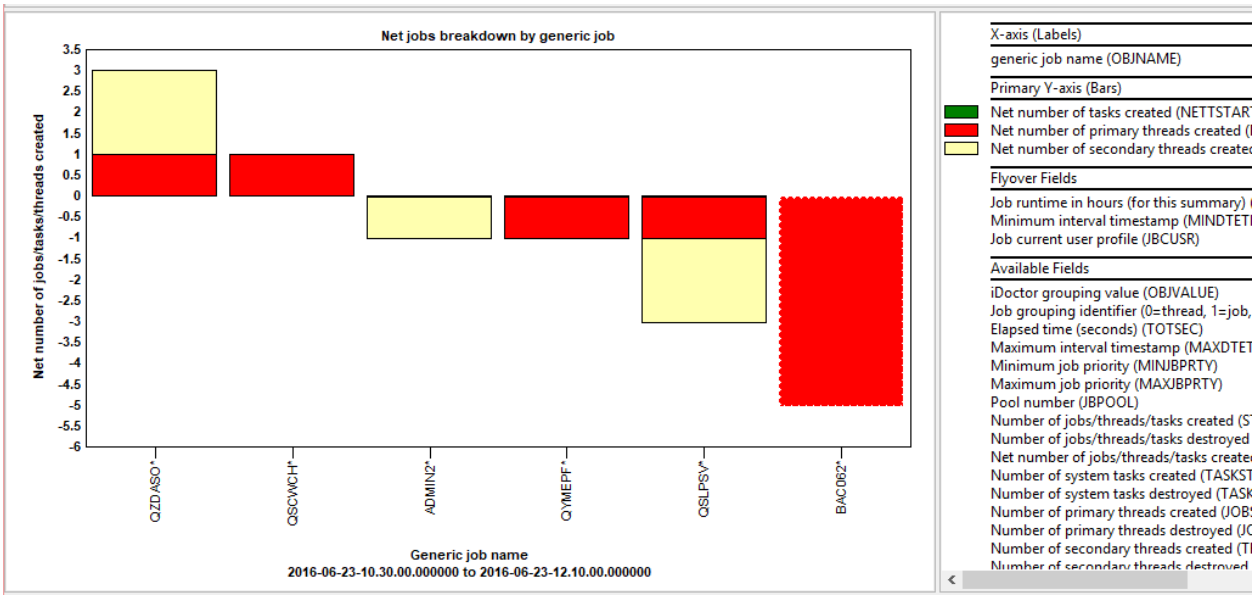
9.18.17 Net jobs breakdown rankings

These graphs display the net number of jobs/tasks/threads created as a rankings chart for any of the job groupings shown below:



Job counts graphs -> Net jobs breakdown rankings

An example is shown below:



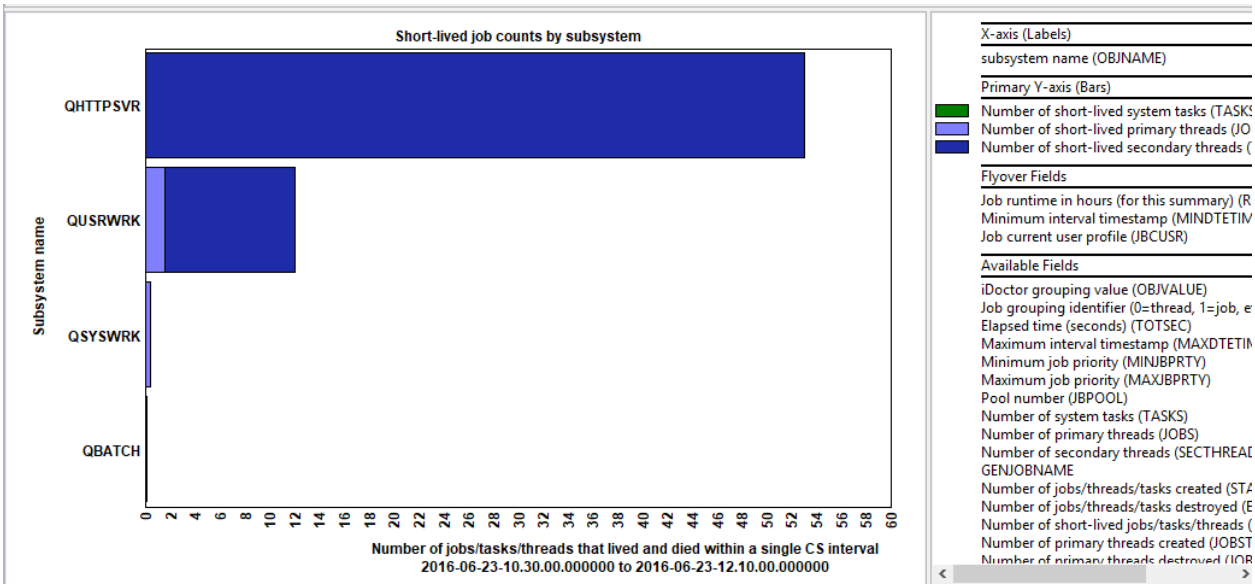
Net jobs breakdown by generic job

9.18.18 Short-lived counts rankings

These graph display counts of the short-lived jobs/tasks/threads as a rankings chart for any of the job groupings shown below:

Job counts graphs -> Short-lived job counts rankings

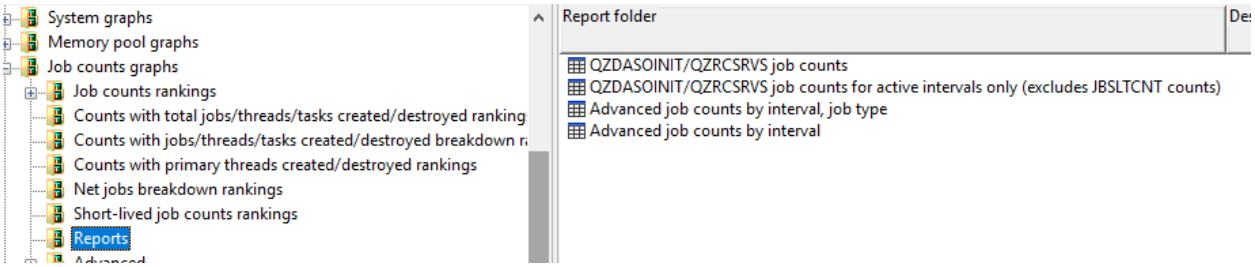
The short-lived job counts refer to jobs/threads/tasks that only lived for a single Collection Services data collection interval or less. An example is shown below:



Short-lived job counts by subsystem

9.18.19 Reports

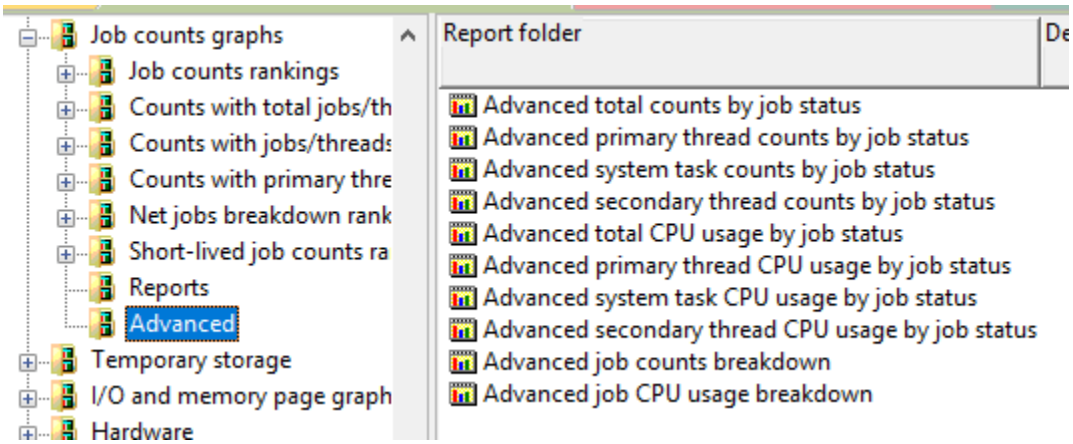
This folder contains table views showing advanced options.



Job counts graphs -> Reports

9.18.20 Advanced

This folder contains additional graphs for IBM service or advanced users.



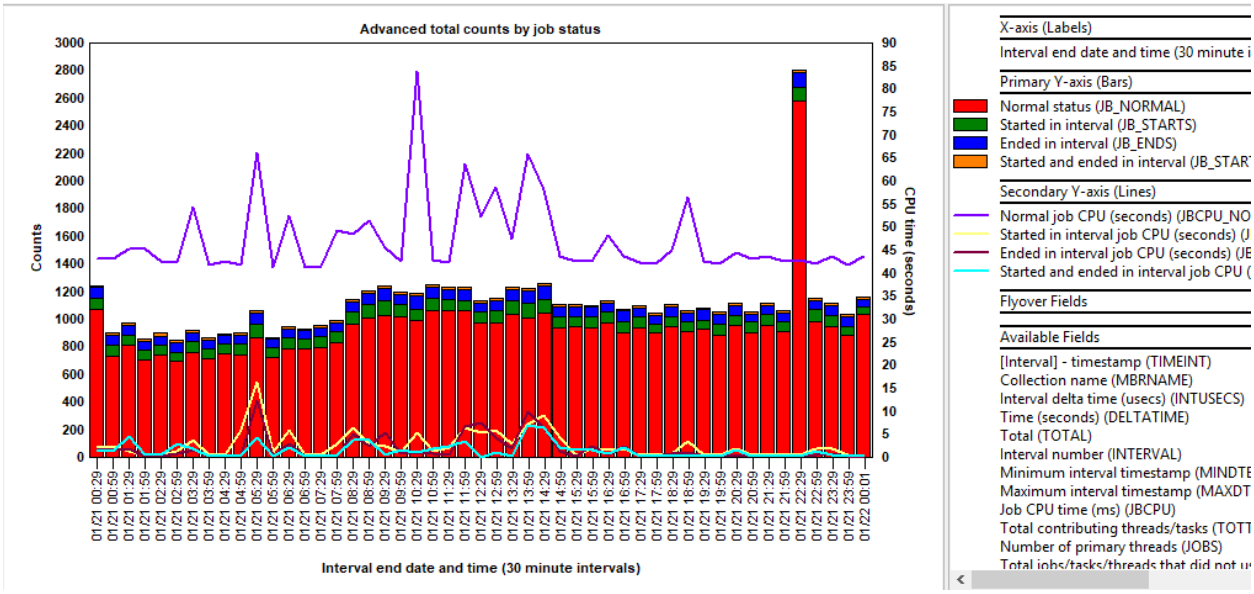
Job counts graphs -> Advanced

9.18.20.1 Advanced total job counts by job status

This graph breaks down the job counts based on the job's status with the following values on the primary Y-axis:

1) Normal status (job did not start or end this interval)
2) Started in interval
3) Ended in interval
4) Started and ended in interval

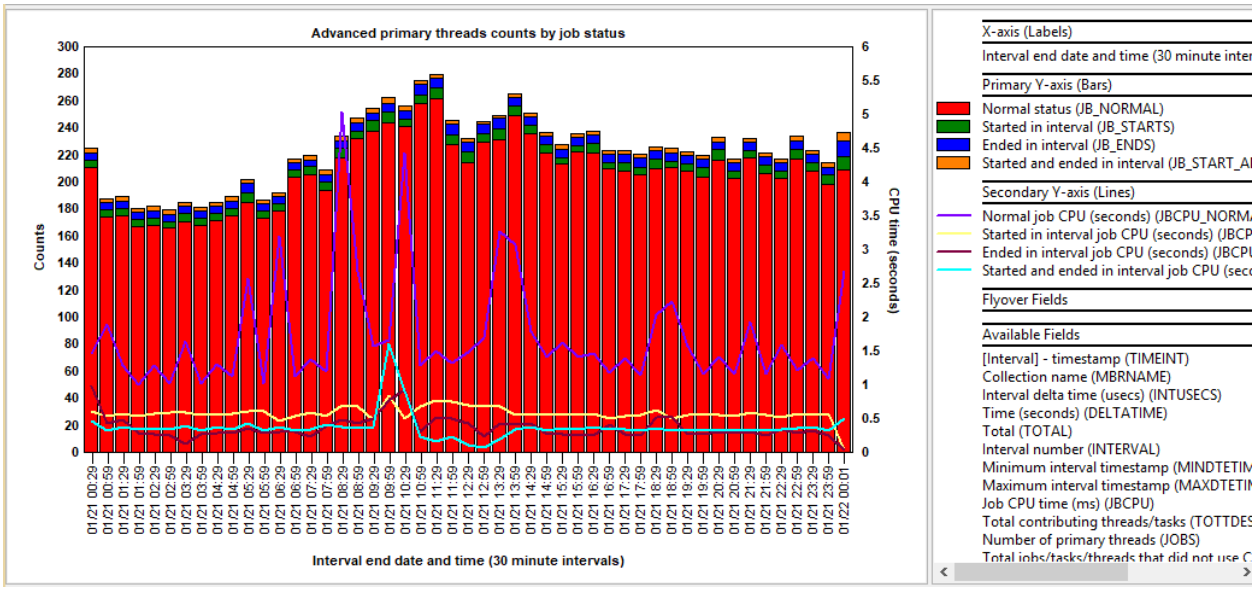
The secondary Y-axis provides the CPU time consumed for the jobs in each of the 4 categories described above.



Advanced total job counts by job status

9.18.20.2 Advanced primary thread counts by job status

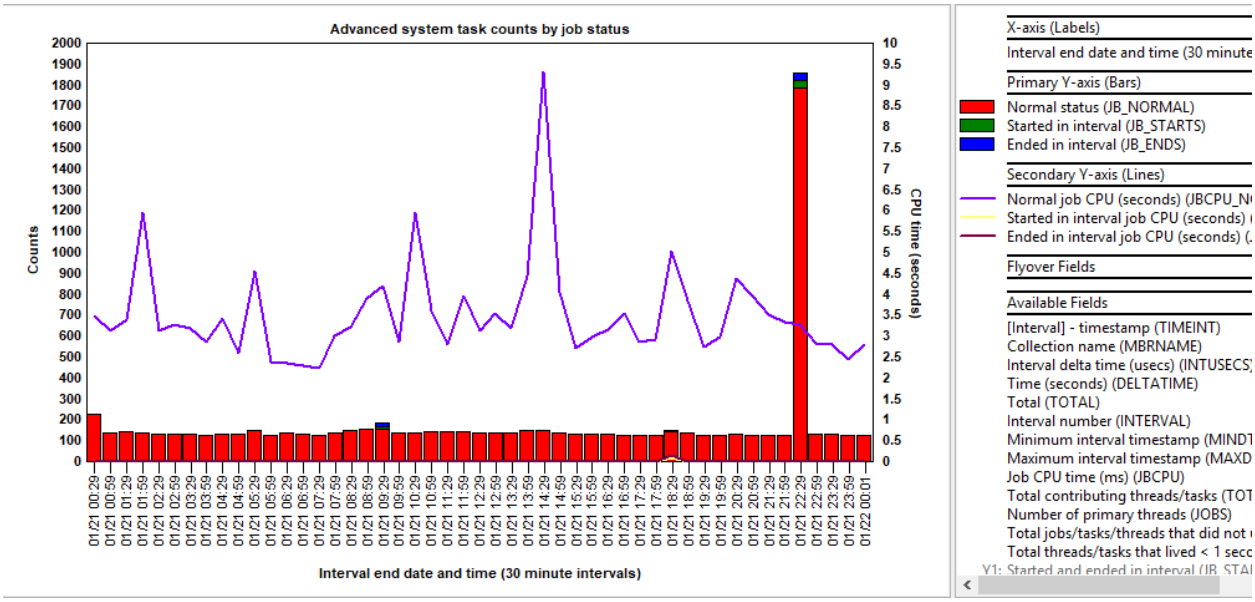
This graph is the same as the Advanced total job counts by job status graph except the metrics shown only apply to primary threads.



Advanced primary thread counts by job status

9.18.20.3 Advanced system task counts by job status

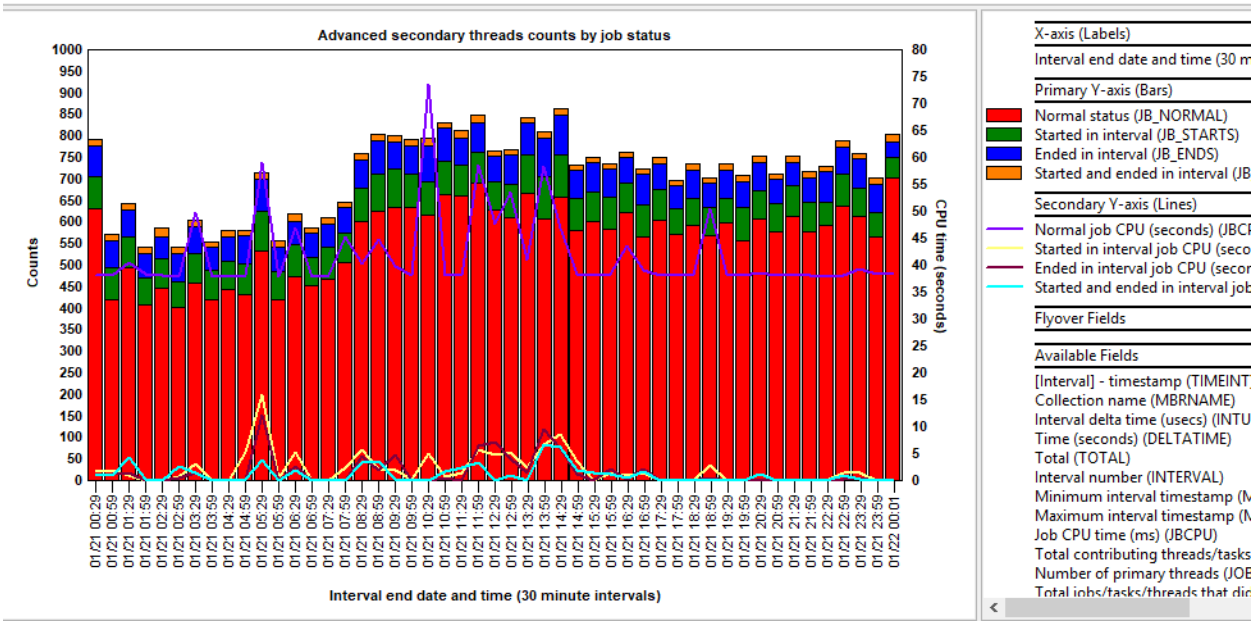
This graph is the same as the Advanced total job counts by job status graph except the metrics shown only apply to system tasks.



Advanced system task counts by job status

9.18.20.4 Advanced secondary thread counts by job status

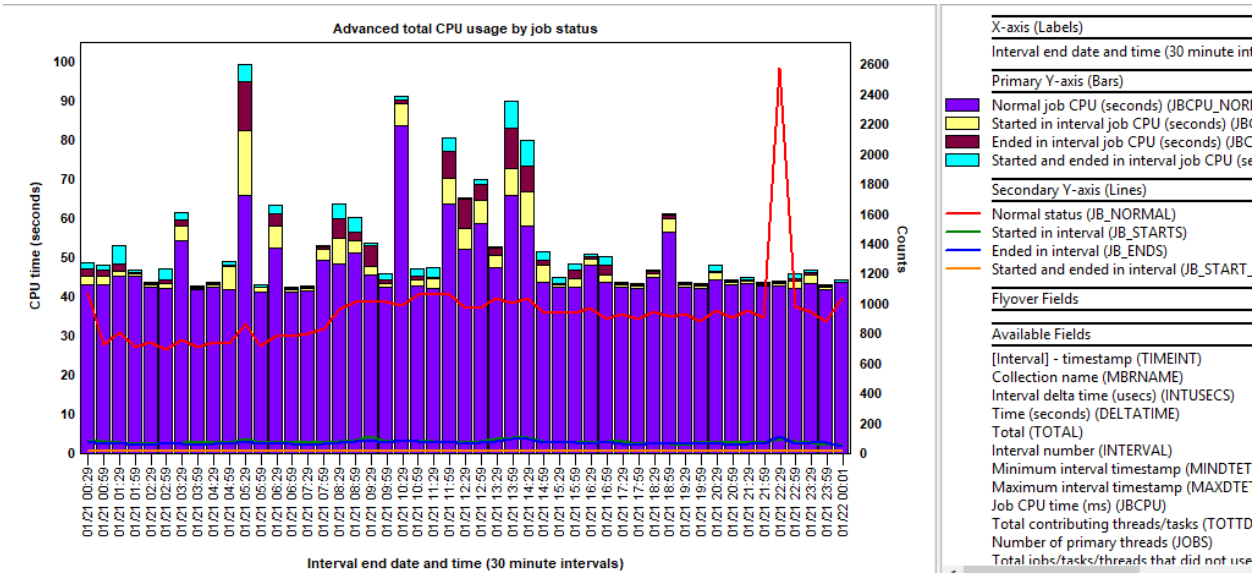
This graph is the same as the Advanced total job counts by job status graph except the metrics shown only apply to secondary threads.



Advanced secondary thread counts by job status

9.18.20.5 Advanced total CPU usage by job status

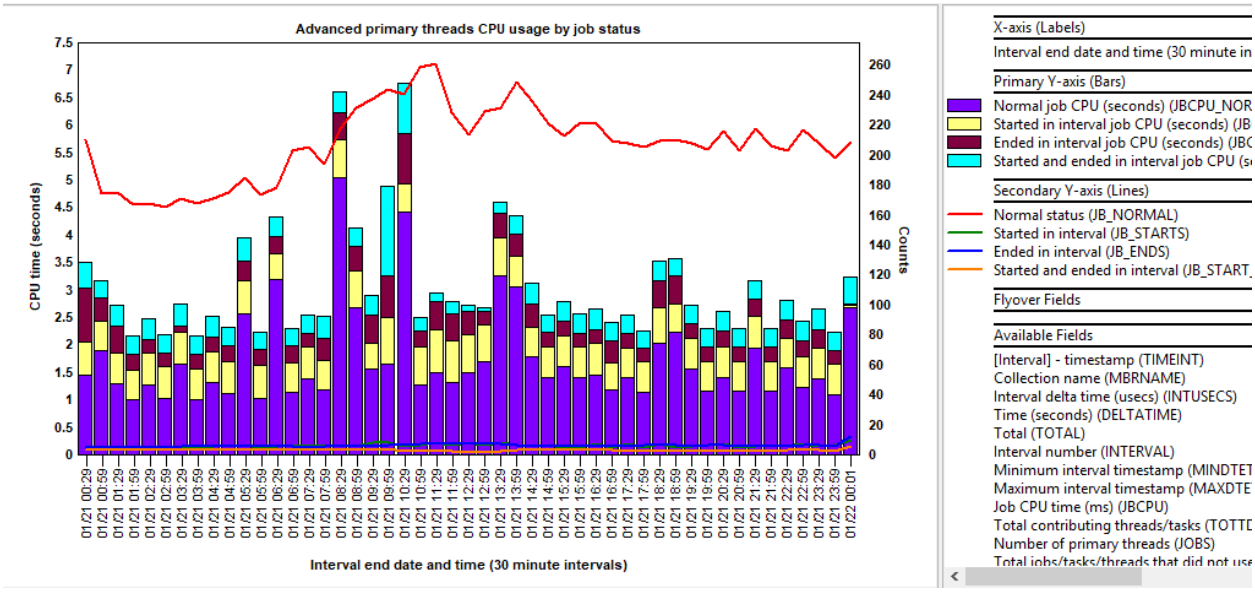
This graph is the same as Advanced total job counts by job status except the Y1-Y2 axes are flipped around.



Advanced total CPU usage by job status

9.18.20.6 Advanced primary threads CPU usage by job status

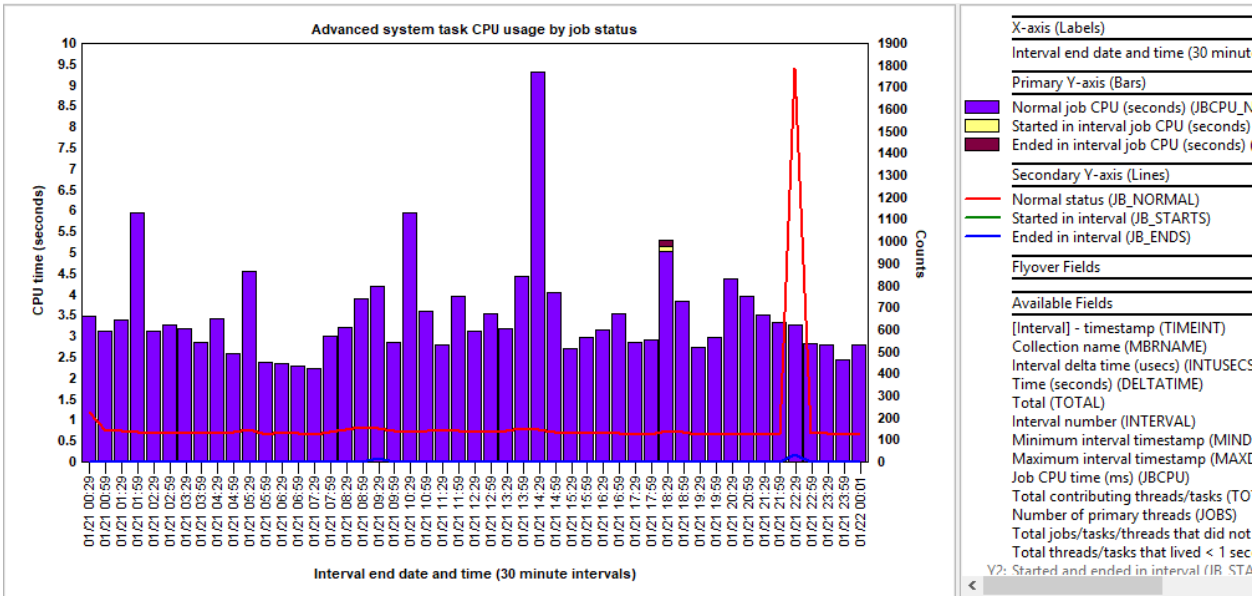
This graph is the same as Advanced primary thread counts by job status except the Y1-Y2 axes are flipped around.



Advanced primary threads CPU usage by job status

9.18.20.7 Advanced system task CPU usage by job status

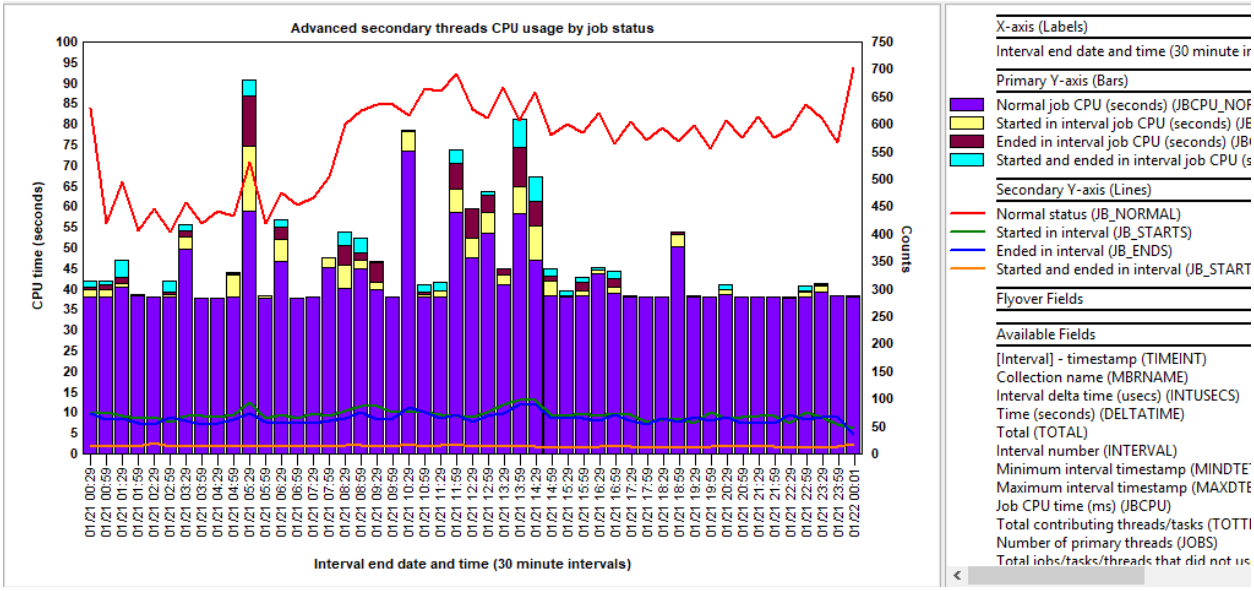
This graph is the same as Advanced system task counts by job status except the Y1-Y2 axes are flipped around.



Advanced system task CPU usage by job status

9.18.20.8 Advanced secondary threads CPU usage by job status

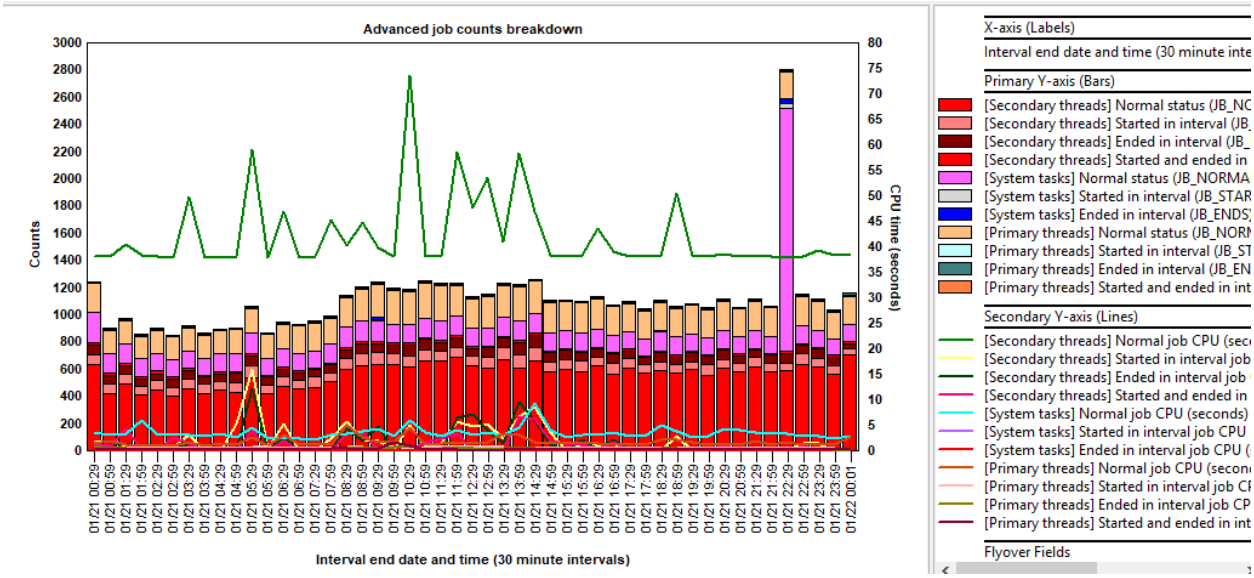
This graph is the same as Advanced secondary thread counts by job status except the Y1-Y2 axes are flipped around.



Advanced secondary threads CPU usage by job status

9.18.20.9 Advanced job counts breakdown

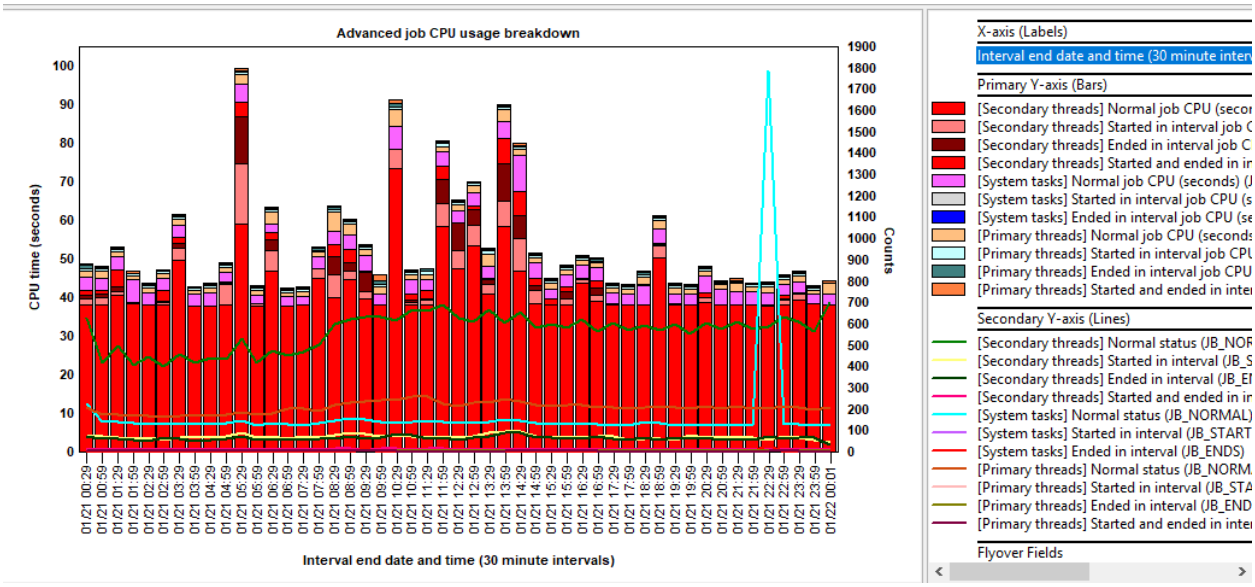
This graph combines many of the metrics in the advanced job counts graphs together into a single chart.



Advanced job counts breakdown

9.18.20.10 Advanced job CPU usage breakdown

This graph combines many of the metrics in the advanced job CPU usage graphs together into a single chart.

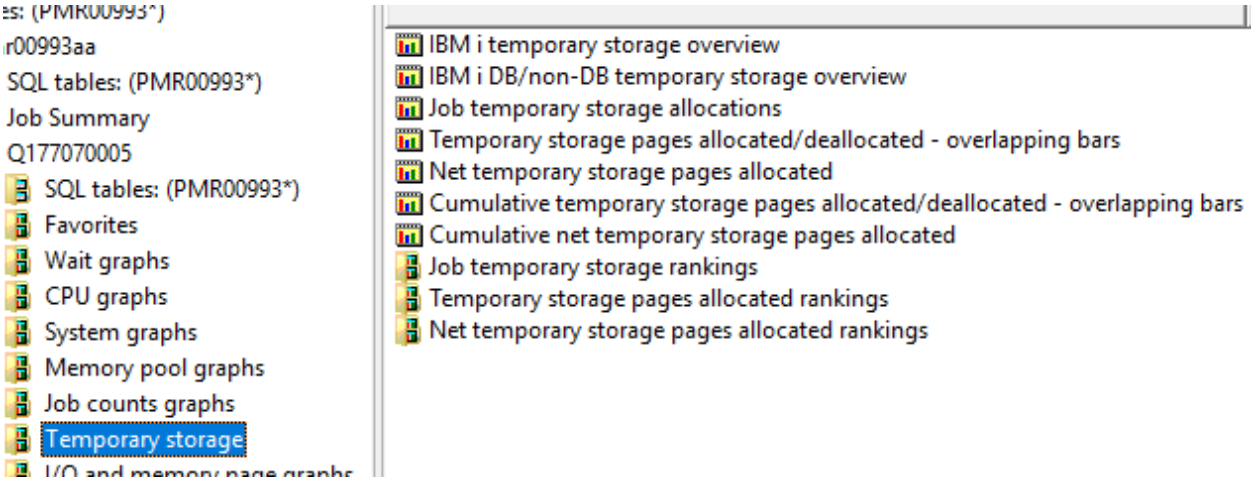


Advanced job CPU usage breakdown

9.19 Temporary storage

This folder contains the graphs showing temporary storage consumption on the system or ranked on a per job or job grouping basis. These metrics are only available at IBM i 7.2 and higher.

Note: The 1st 3 graphs are available whether the Collection Summary analysis has been ran or not. The rest of the graphs require the Collection Summary analysis to be ran in order to appear in this folder.



Temporary storage

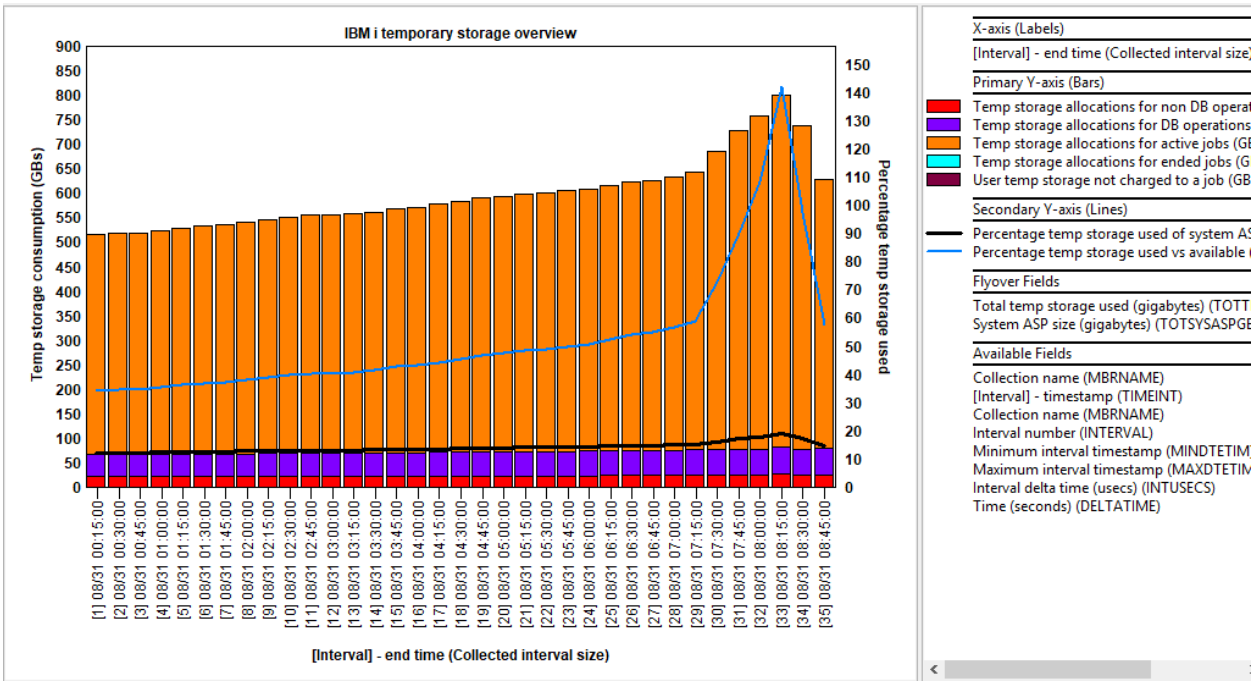
9.19.1 IBM i temporary storage overview

Collection Services provides the data used in this graph to break down the various contributions to temporary storage. You can find more details about these fields in the knowledge center for table qapmsystem:
https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_72/rzahx/rzahxqapmsystem.htm

The following fields are provided on the primary Y-axis:

SYOSTMPGB	Current® temporary storage allocated for non-database operations by IBM® i. A snapshot of the total temporary storage currently allocated for non-database operations across the system, expressed in gigabytes (GBs.)
SYDBTMPGB	Current temporary storage allocated for database operations by IBM i. A snapshot of the total temporary storage currently allocated for database operations across the system, expressed in gigabytes (GBs.)
SYAJOBTMPGB	Current temporary storage charged to active jobs. A snapshot of the total temporary storage currently charged to active jobs, expressed in gigabytes (GBs.)
SYEJOBTMPGB	Current temporary storage charged to ended jobs. A snapshot of the total temporary storage currently charged to ended jobs, expressed in gigabytes (GBs.)
SYUSERTMPGB	Current user temporary storage. A snapshot of the total user temporary storage currently allocated, expressed in gigabytes (GBs.) Only the user temporary storage not charged to any job is accounted for here.

The secondary Y-axis displays the percentage of temp storage used of the system ASP and another value showing percentage used vs available.

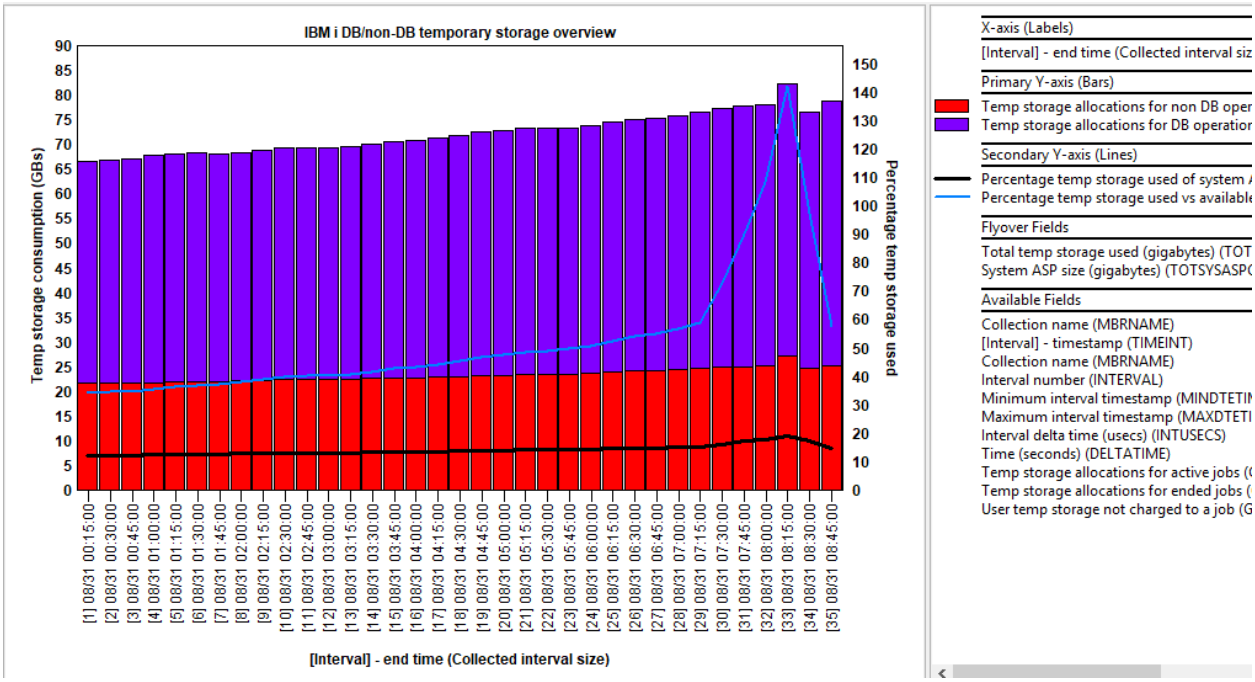


IBM i temporary storage overview

9.19.2 IBM i DB/non-DB temporary storage overview

This graph is the same as the previous one but just compares the two first columns from the primary Y-axis.

The secondary Y-axis displays the percentage of temp storage used of the system ASP and another value showing percentage used vs available.

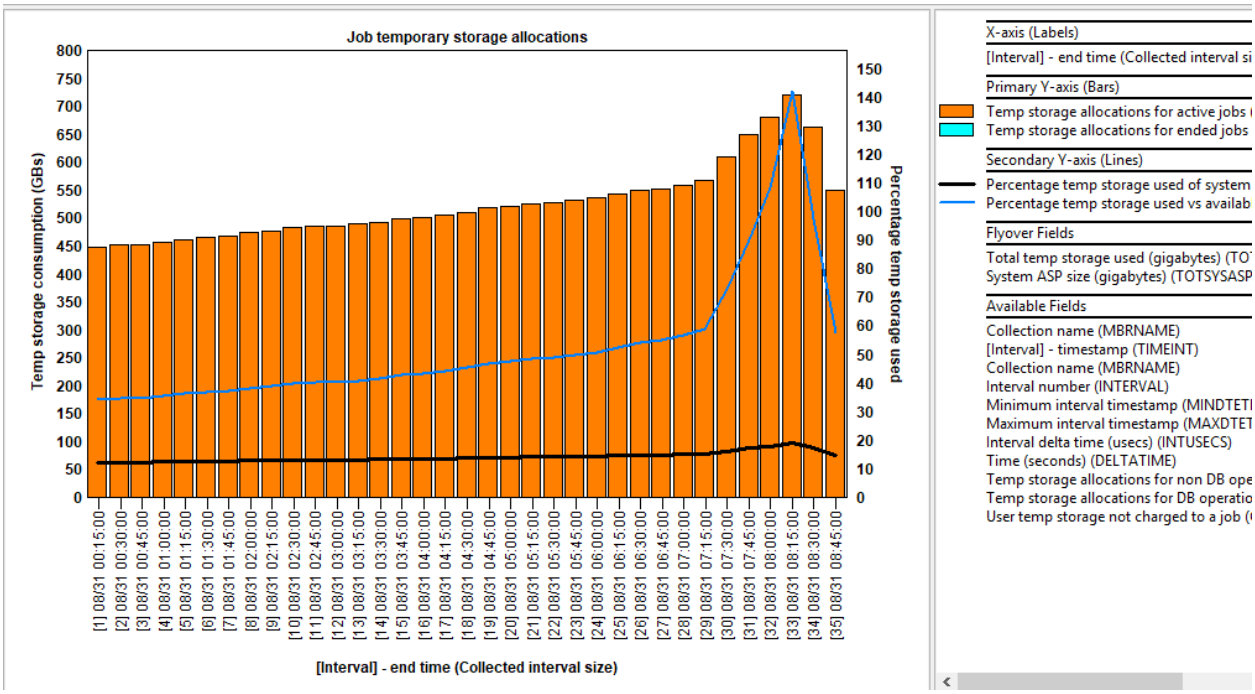


IBM i DB/non-DB temporary storage overview

9.19.3 Job temporary storage allocations

This graph includes only the temporary storage allocations attributed to active jobs and ended jobs.

The secondary Y-axis displays the percentage of temp storage used of the system ASP and another value showing percentage used vs available.



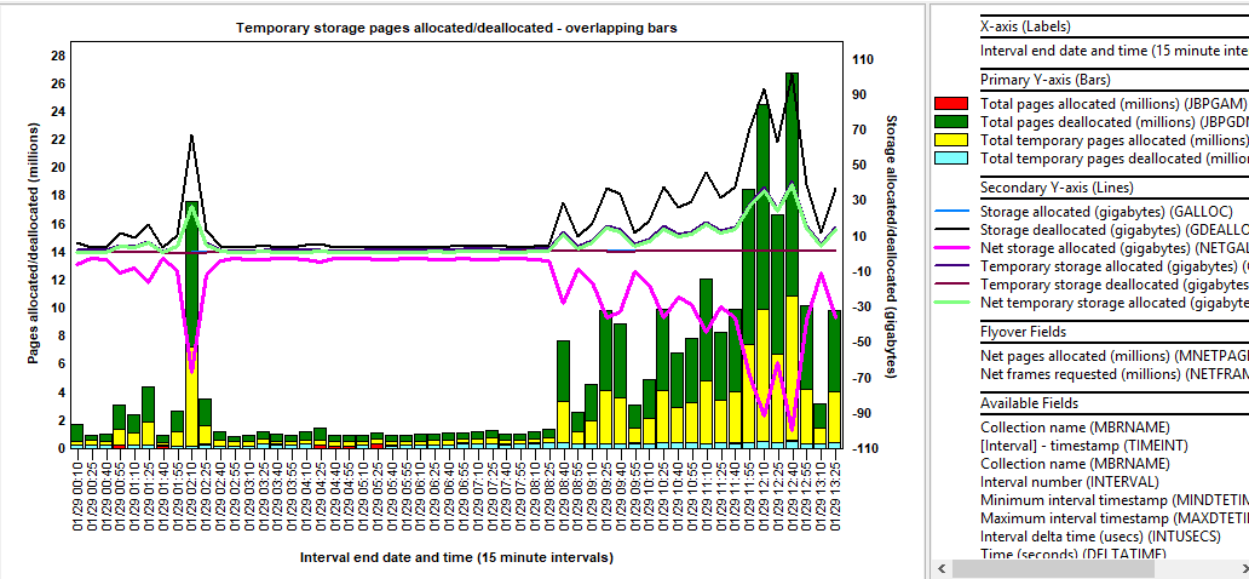
Job temporary storage allocations

9.19.4 Temporary storage pages allocated/deallocated - overlapping bars

This graph shows the following metrics but using overlapping bars. This means the values shown at the top of each bar are the largest. The values shown on the primary Y-axis are:

1) Total pages allocated (millions)
2) Total pages deallocated (millions)
3) Total temporary pages allocated (millions)
4) Total temporary pages deallocated (millions)

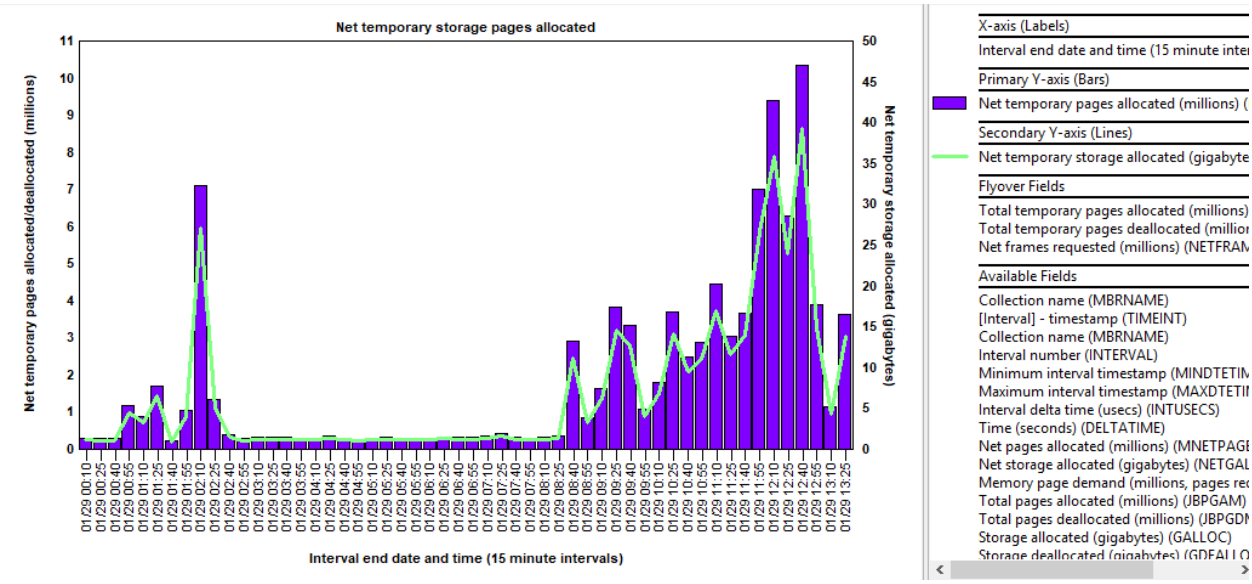
The secondary Y-axis adds up the total storage of these pages in various ways.



Temporary storage pages allocated/deallocated - overlapping bars

9.19.5 Net temporary storage pages allocated

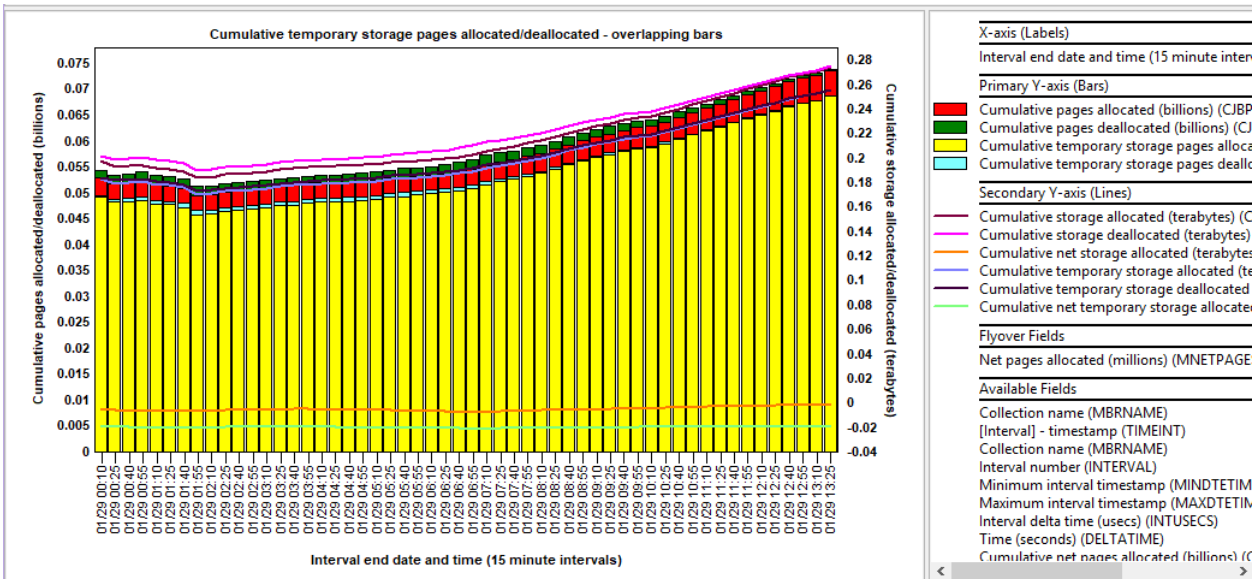
This graph just shows the net temporary storage pages allocated (in millions) on the primary Y-axis with the secondary Y-axis containing the net temporary storage allocations (in gigabytes.)



Net temporary storage pages allocated

9.19.6 Cumulative temporary storage pages allocated/deallocated - overlapping bars

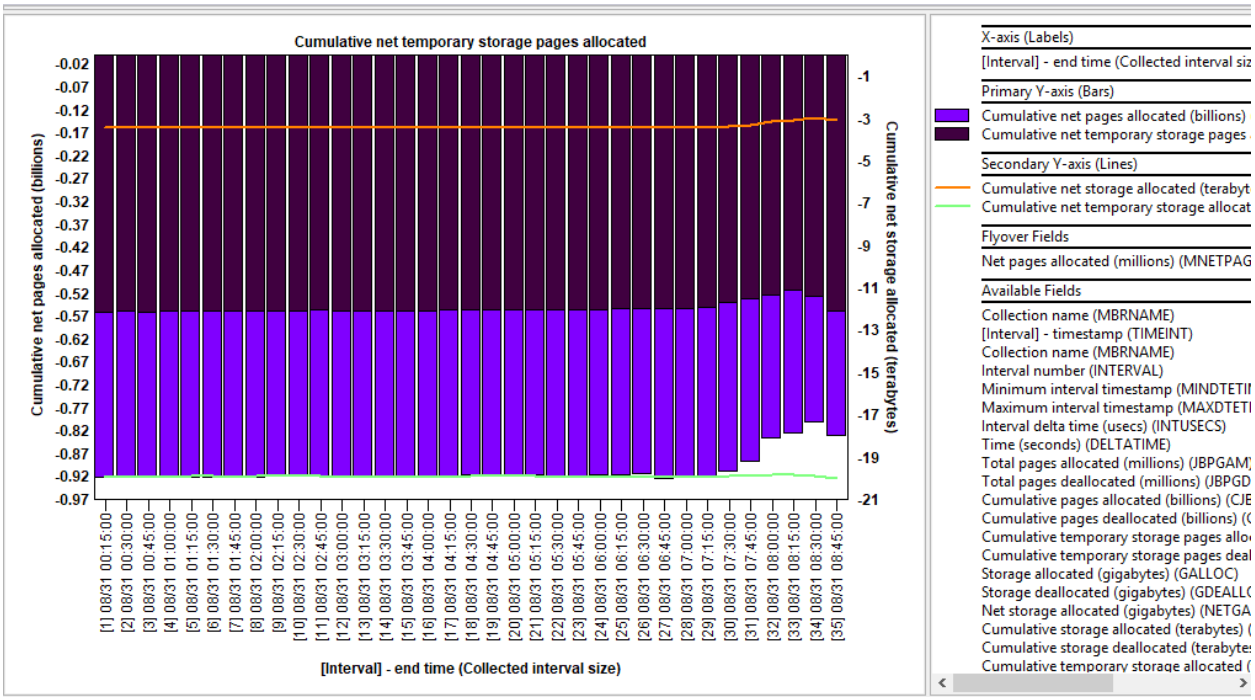
This graph displays cumulative totals for pages allocated /deallocated and temporary storage pages allocated/deallocated. The cumulative value is added up for each job and is the total since each job started.



Cumulative temporary storage pages allocated/deallocated - overlapping bars

9.19.7 Cumulative net temporary storage pages allocated

This graph displays the cumulative net pages allocated and temporary storage pages allocated over time. The secondary Y-axis displays the cumulative size of these values in terrabytes.

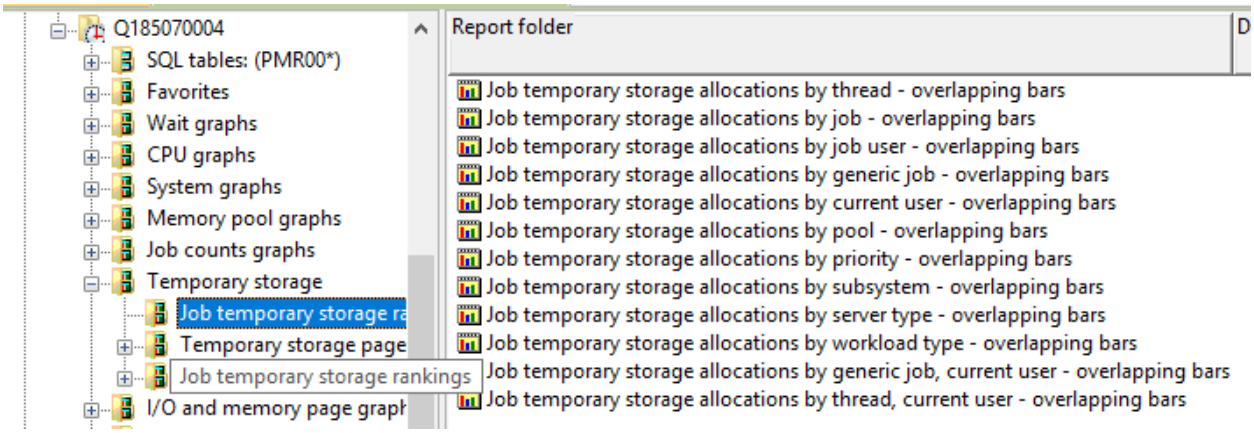


Cumulative net temporary storage pages allocated

9.19.8 Job temporary storage rankings

These graphs show the job temporary storage allocations in various job groupings as shown below:

These graphs can be accessed as a drill down from the temporary storage overview graphs (for the time period selected) or from here.

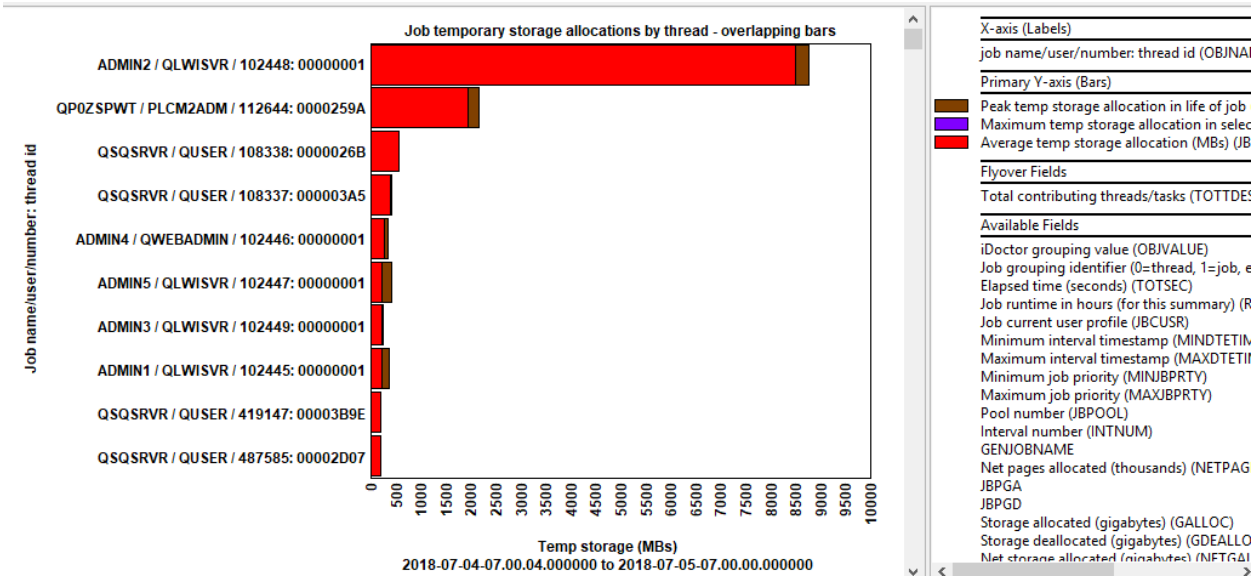


Temporary storage -> Job temporary storage rankings

Note: These graphs use overlapping bars and so the maximum values will be listed at the end of the bar on the right side.

9.19.8.1 Job temporary storage allocations by thread – overlapping bars

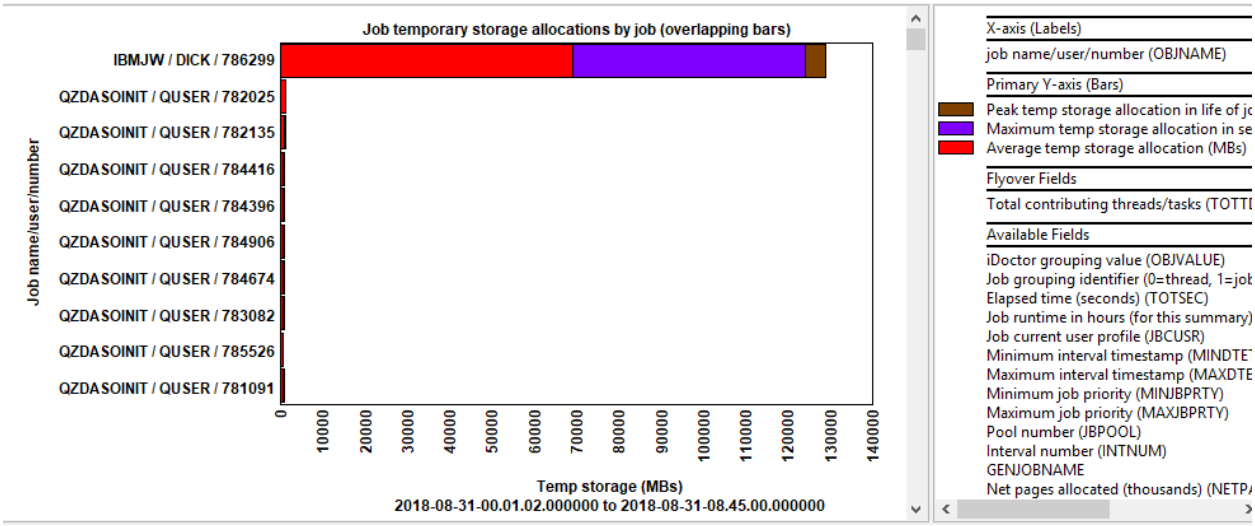
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by job/task/thread.



Job temporary storage allocations by thread - overlapping bars

9.19.8.2 Job temporary storage allocations by job - overlapping bars

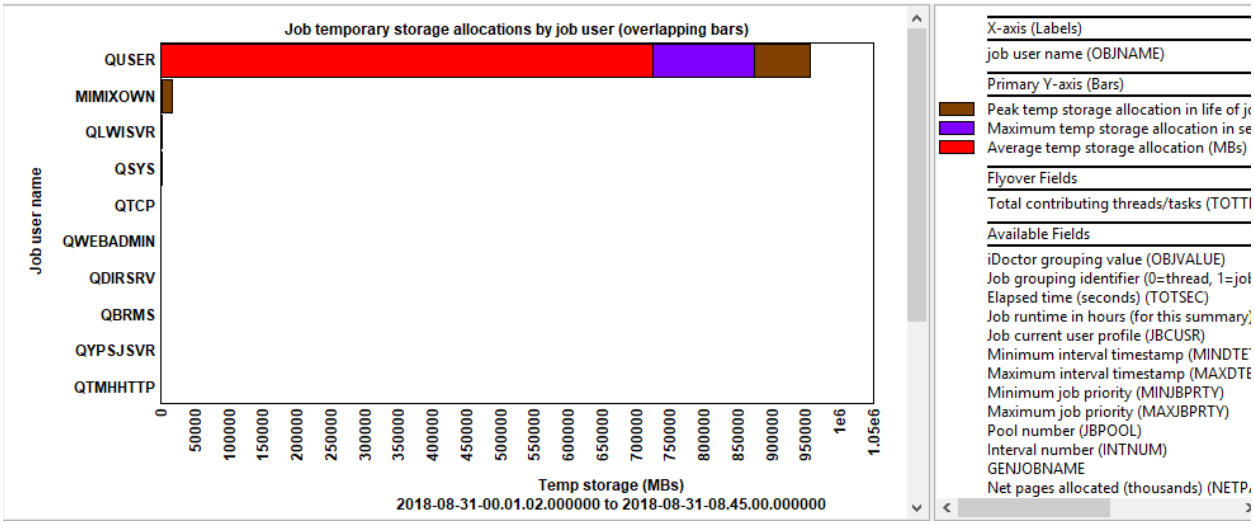
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by job.



Job temporary storage allocations by job - overlapping bars

9.19.8.3 Job temporary storage allocations by job user - overlapping bars

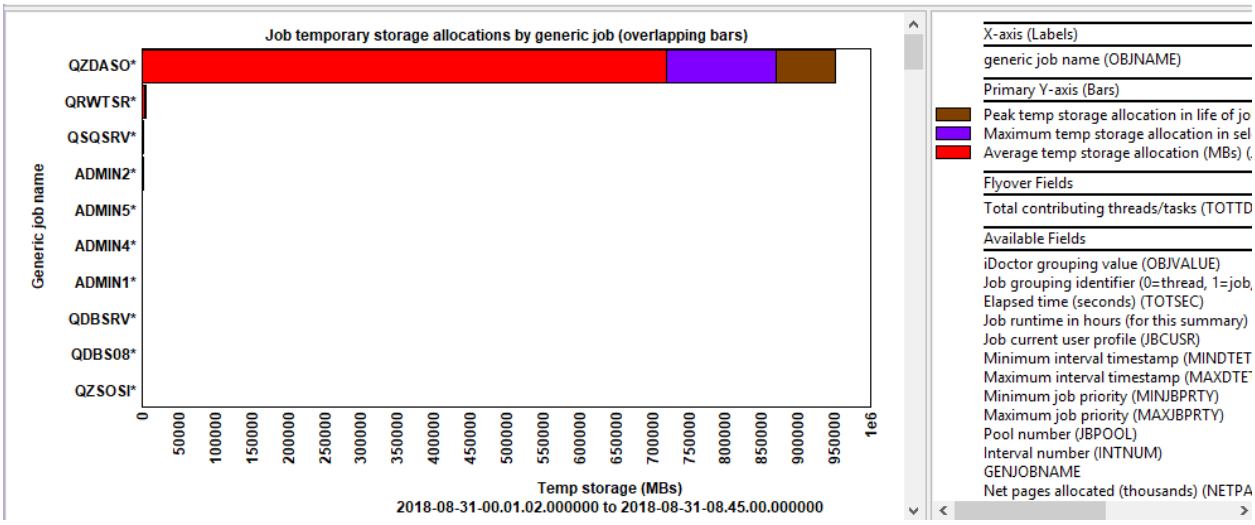
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by job user.



Job temporary storage allocations by job user - overlapping bars

9.19.8.4 Job temporary storage allocations by generic job - overlapping bars

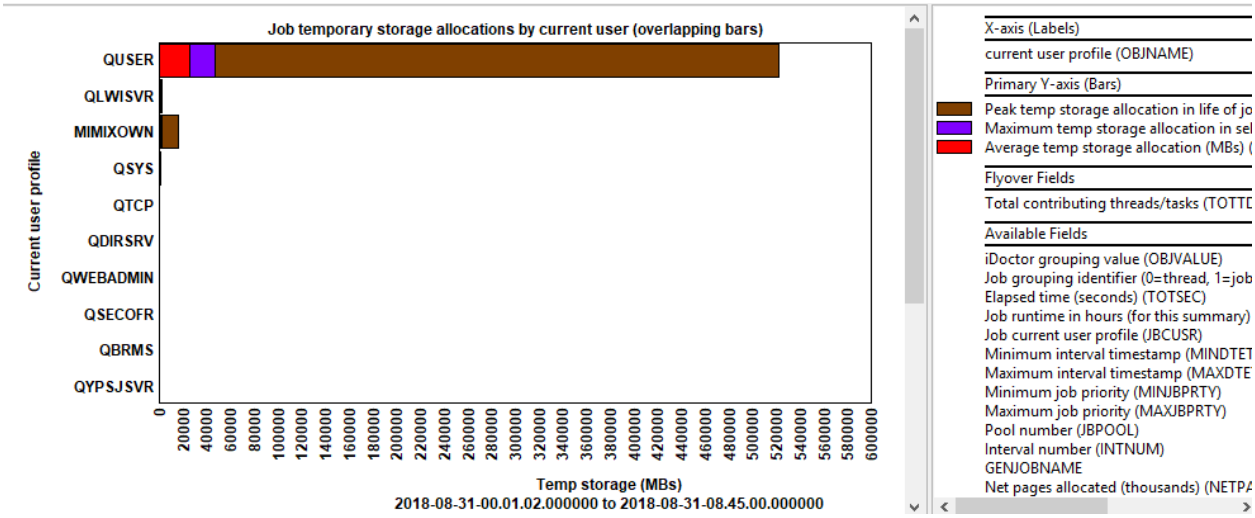
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by generic job or task name.



Job temporary storage allocations by generic job - overlapping bars

9.19.8.5 Job temporary storage allocations by current user - overlapping bars

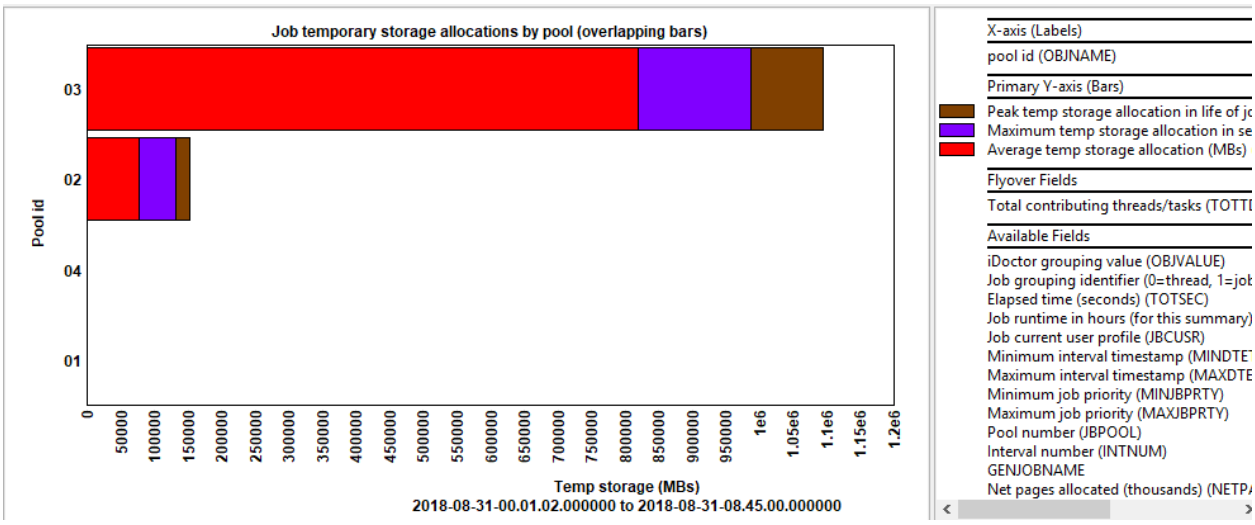
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by current user.



Job temporary storage allocations by current user - overlapping bars

9.19.8.6 Job temporary storage allocations by pool - overlapping bars

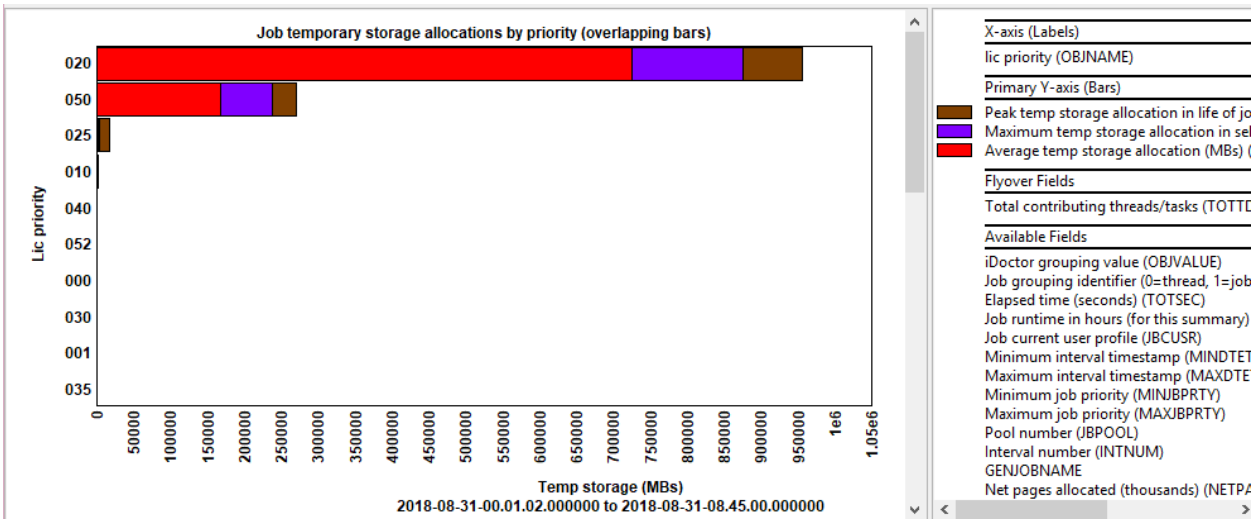
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by pool.



Job temporary storage allocations by pool - overlapping bars

9.19.8.7 Job temporary storage allocations by priority - overlapping bars

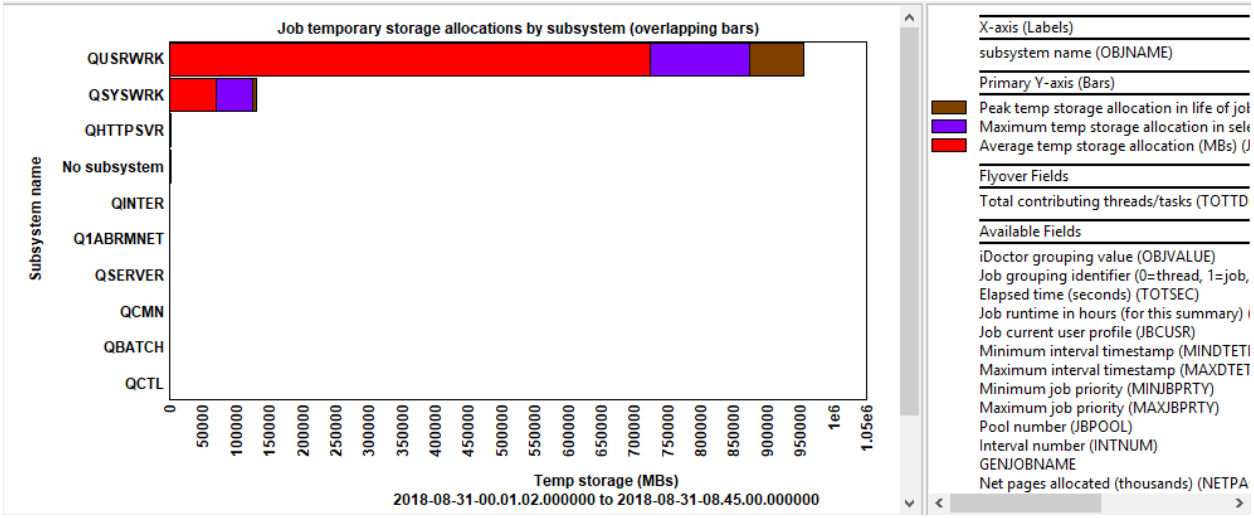
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by LIC priority.



Job temporary storage allocations by priority - overlapping bars

9.19.8.8 Job temporary storage allocations by subsystem - overlapping bars

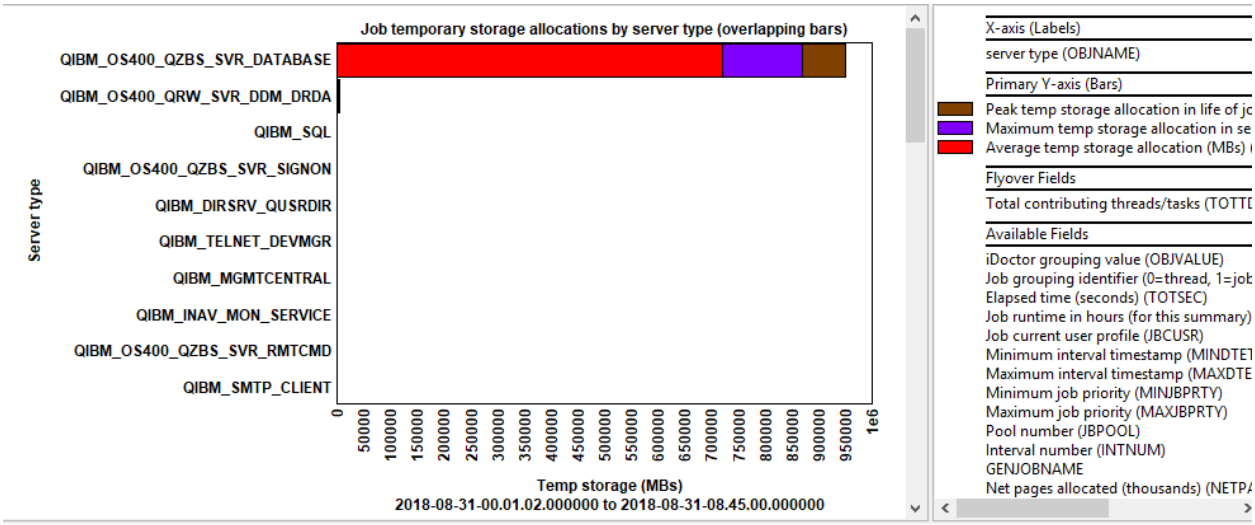
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by subsystem.



Job temporary storage allocations by subsystem - overlapping bars

9.19.8.9 Job temporary storage allocations by server type - overlapping bars

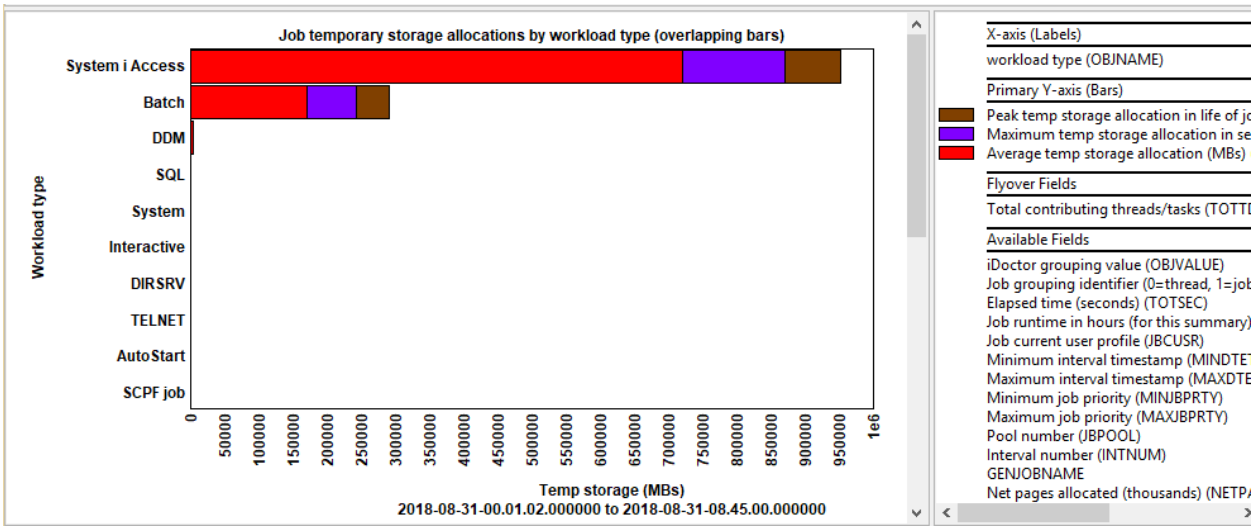
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by server type.



Job temporary storage allocations by server type - overlapping bars

9.19.8.10 Job temporary storage allocations by workload type - overlapping bars

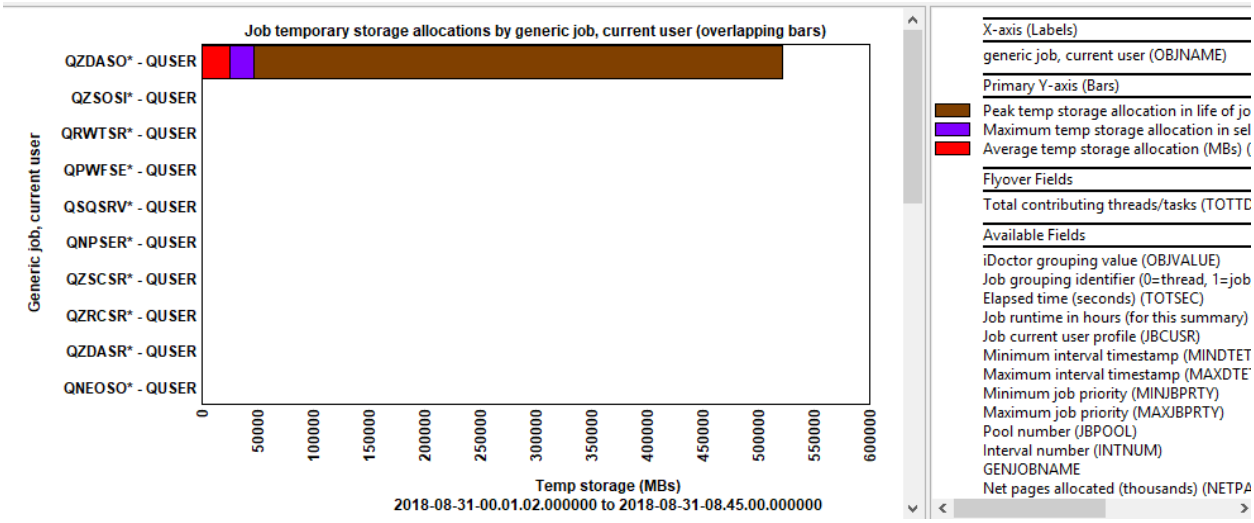
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by workload type.



Job temporary storage allocations by workload type - overlapping bars

9.19.8.11 Job temporary storage allocations by generic job, current user - overlapping bars

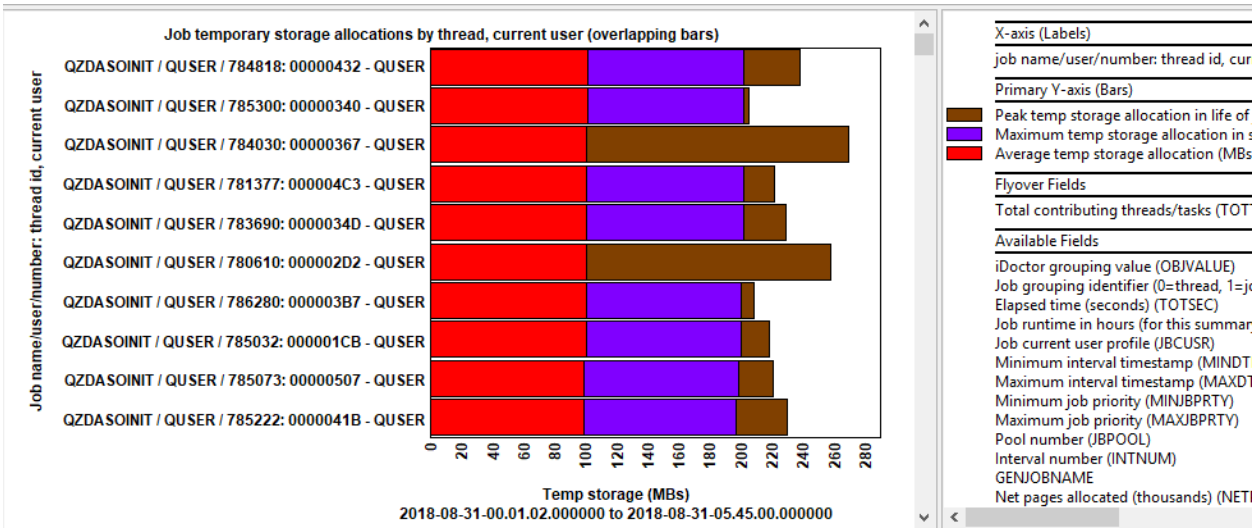
This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by generic job and current user combination.



Job temporary storage allocations by generic job, current user - overlapping bars

9.19.8.12 Job temporary storage allocations by thread, current user - overlapping bars

This graph shows peak, maximum and average temporary storage allocations in megabytes for the time period shown and ranked by job/task/thread and current user combination.

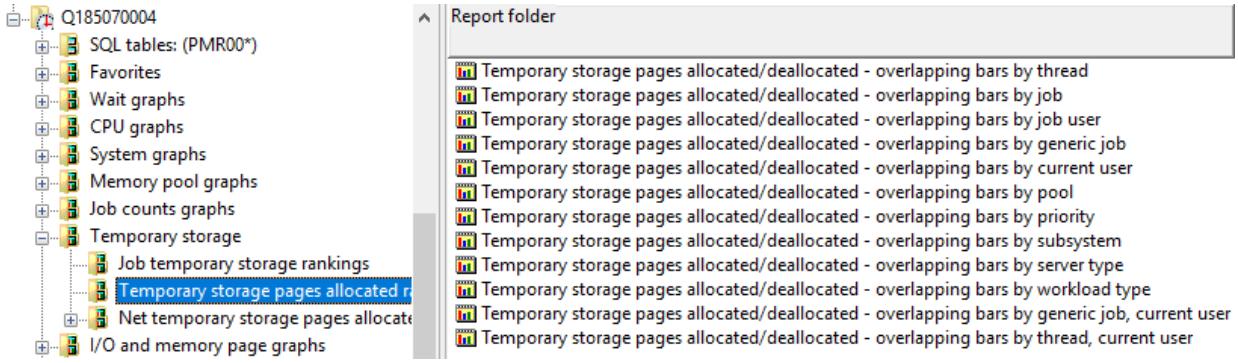


Job temporary storage allocations by thread, current user - overlapping bars

9.19.9 Temporary storage pages allocated rankings

These graphs show the temporary storage pages allocated/deallocated in various job groupings as shown below:

These graphs can be accessed as a drill down from the temporary storage overview graphs (for the time period selected) or from here.



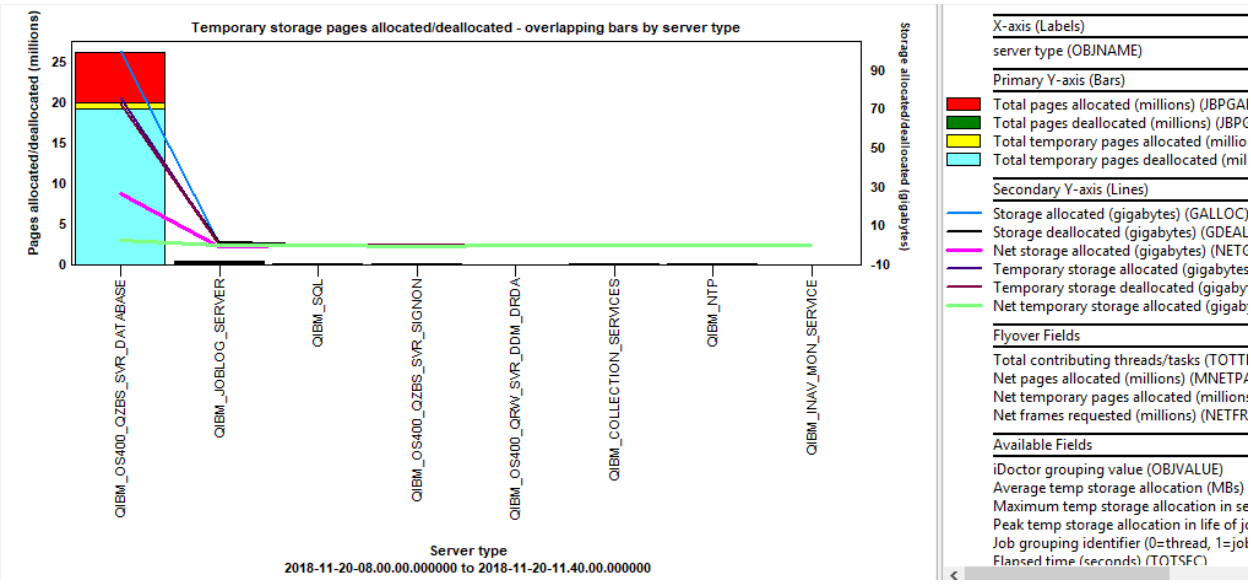
Temporary storage -> Temporary storage pages allocated rankings

These graphs show the following metrics but using overlapping bars. This means the values shown at the top of each bar are the largest. The values shown on the primary Y-axis are:

1) Total pages allocated (millions)
2) Total pages deallocated (millions)
3) Total temporary pages allocated (millions)
4) Total temporary pages deallocated (millions)

The secondary Y-axis adds up the total storage of these pages in various ways.

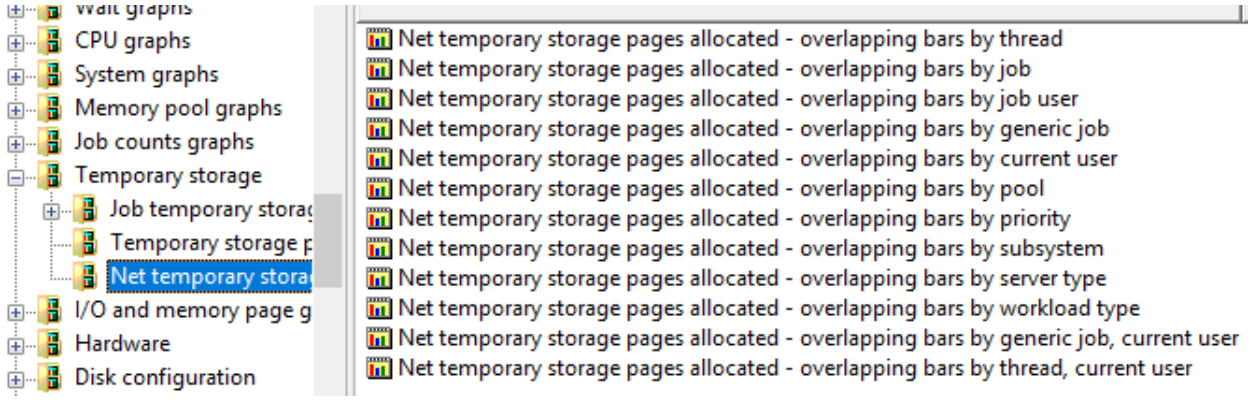
An example is shown below:



Temporary storage pages allocated/deallocated - overlapping bars by server type

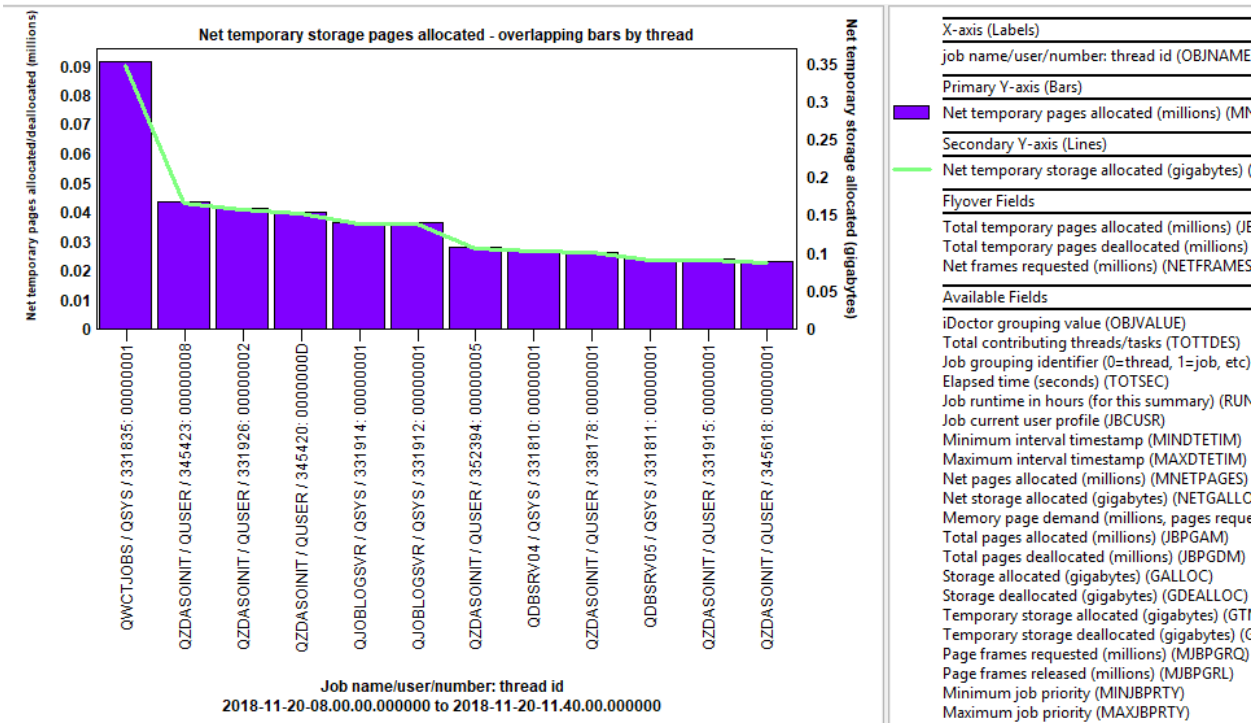
9.19.10 Net temporary storage pages allocated rankings

This graph just shows the net temporary storage pages allocated (in millions) on the primary Y-axis with the secondary Y-axis containing the net temporary storage allocations (in gigabytes.) These are ranked by the various job groupings shown below:



Temporary storage -> Net temporary storage pages allocated rankings

An example is shown below:

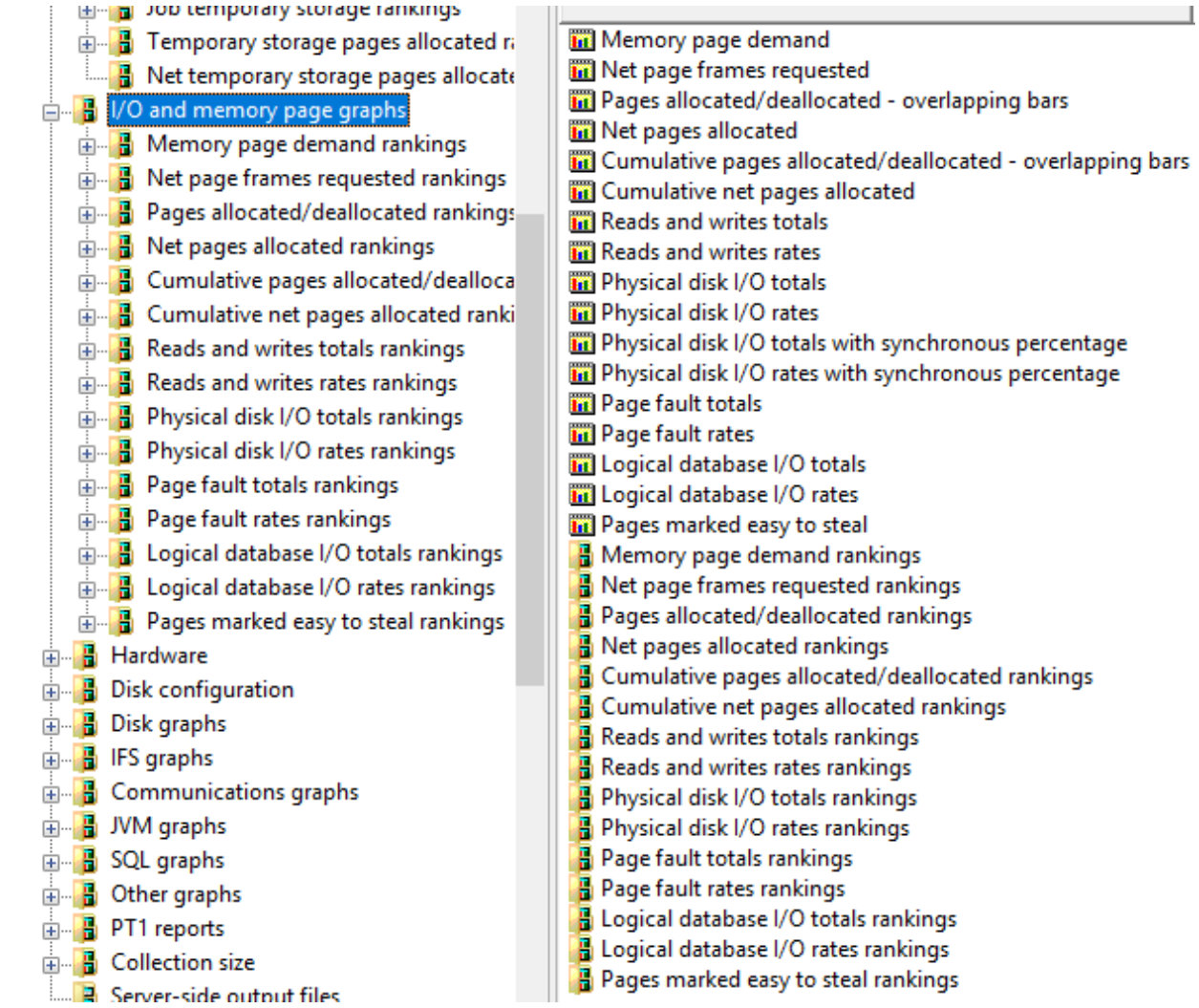


Net temporary storage pages allocated - overlapping bars by thread

9.20 I/O and memory page graphs

These graphs show physical and logical I/Os, pages allocated and page faults as either totals or rates per second over time. Memory page demand and net page frames requested are also included in this folder.

Tip: You can right-click a time interval or interest and pick the top drilldown menu option to graph the same data as a (job/thread/generic job/etc) rankings graph.

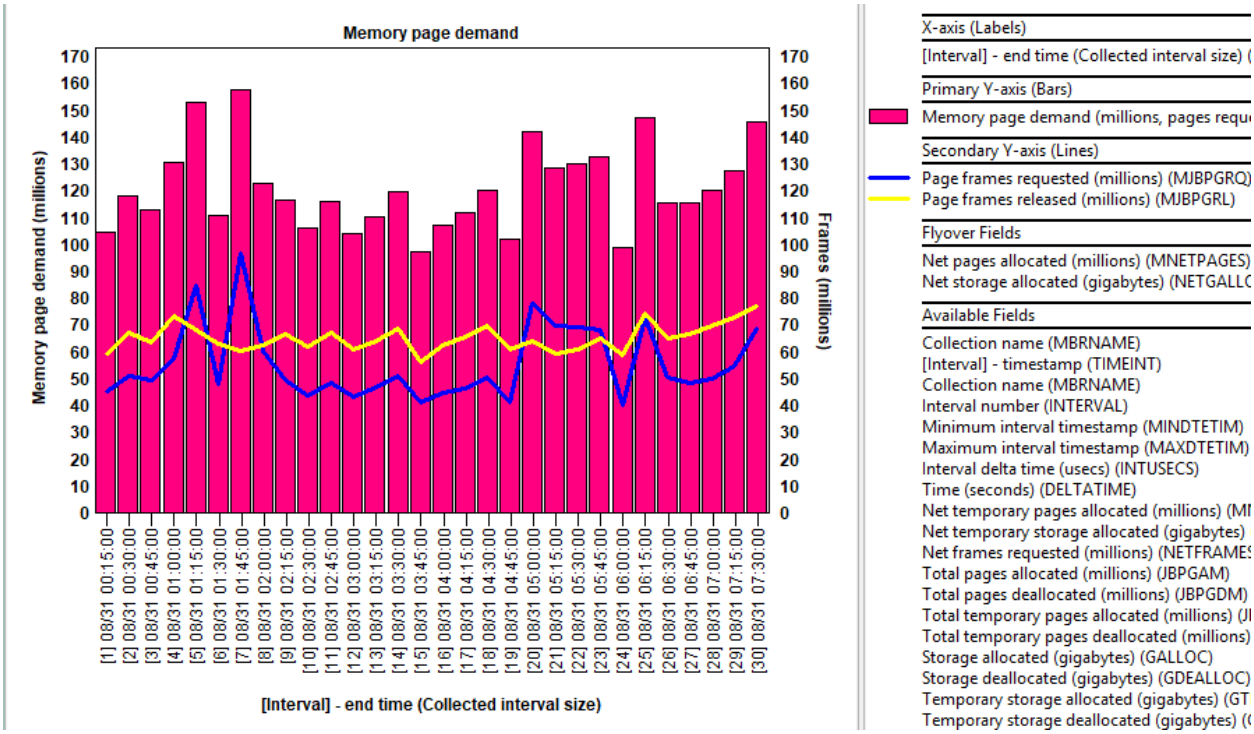


I/O and memory page graphs

9.20.1 Memory page demand

This graph displays the memory pages requested and released which gives a sense of how much memory is being utilized by the jobs captured by Collection Services.

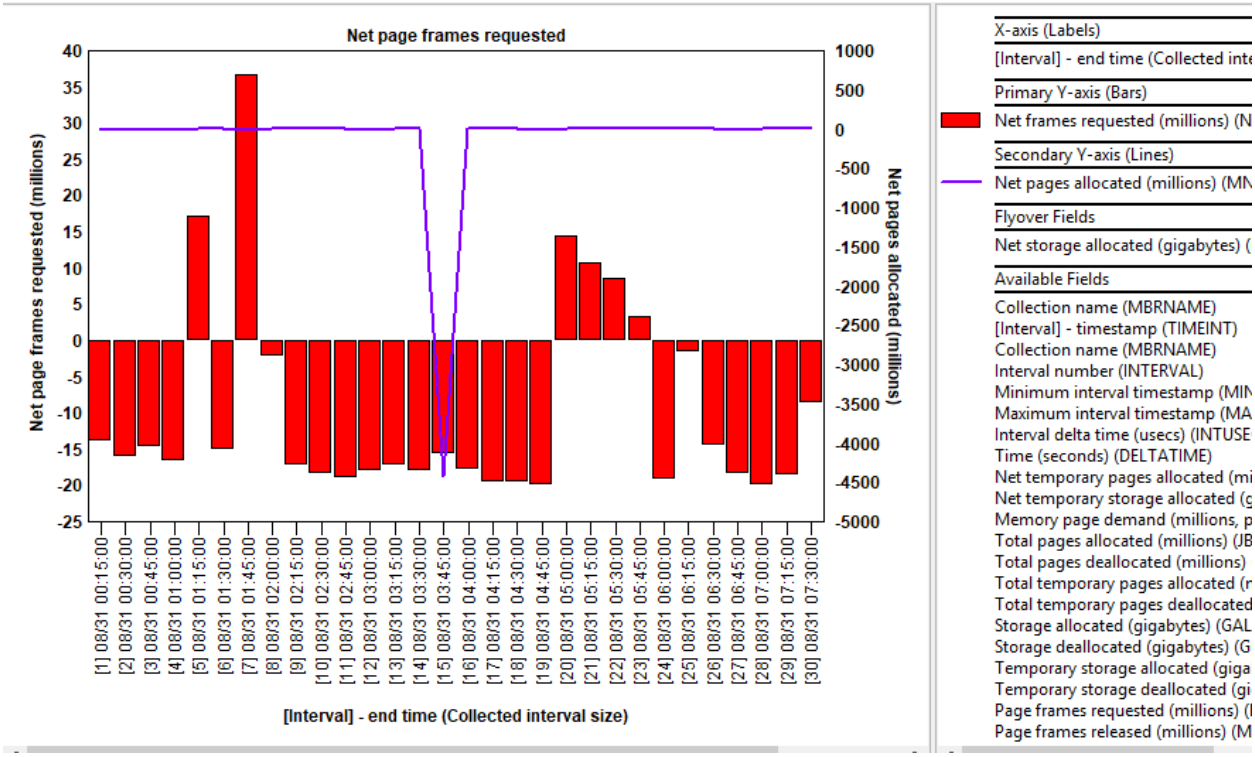
The bars display the total of the page frames requested and released. The secondary Y-axis displays the 2 values as separate lines instead.



Memory page demand

9.20.2 Net page frames requested

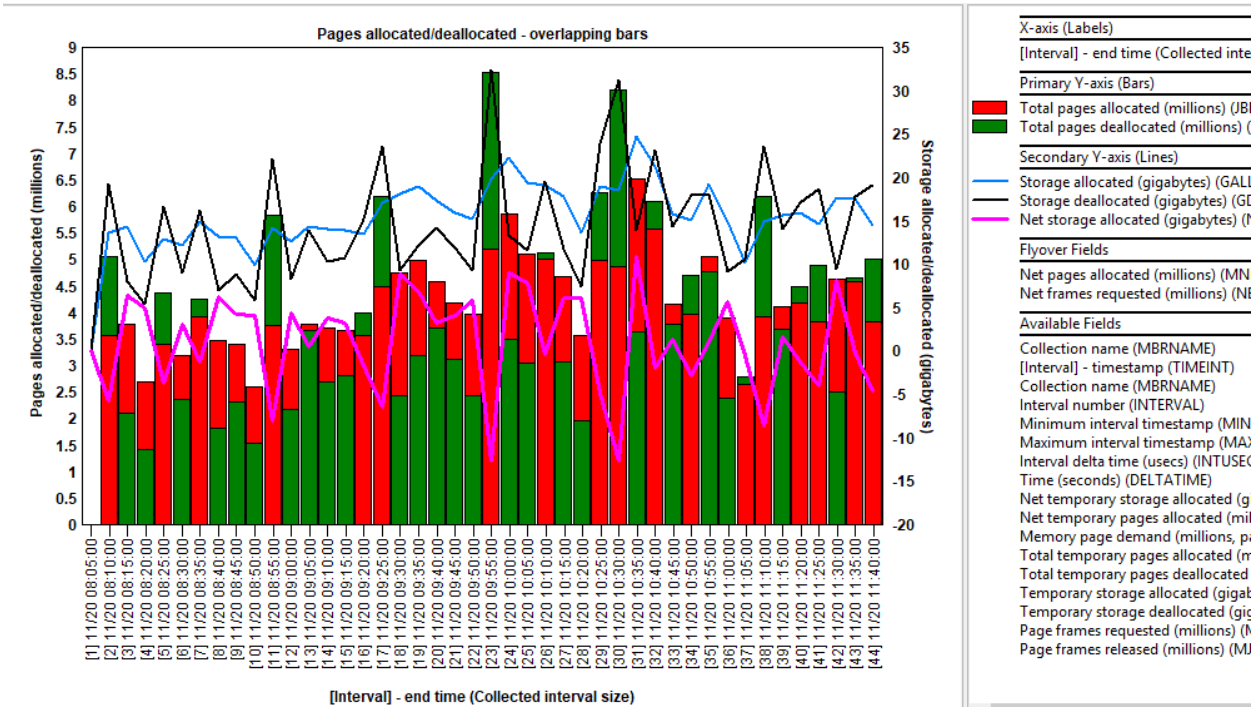
This graph is like the memory page demand graph except it just shows the Net memory pages requested. A negative value indicates that more pages were released than requested in that time interval.



Net page frames requested

9.20.3 Pages allocated/deallocated - overlapping bars

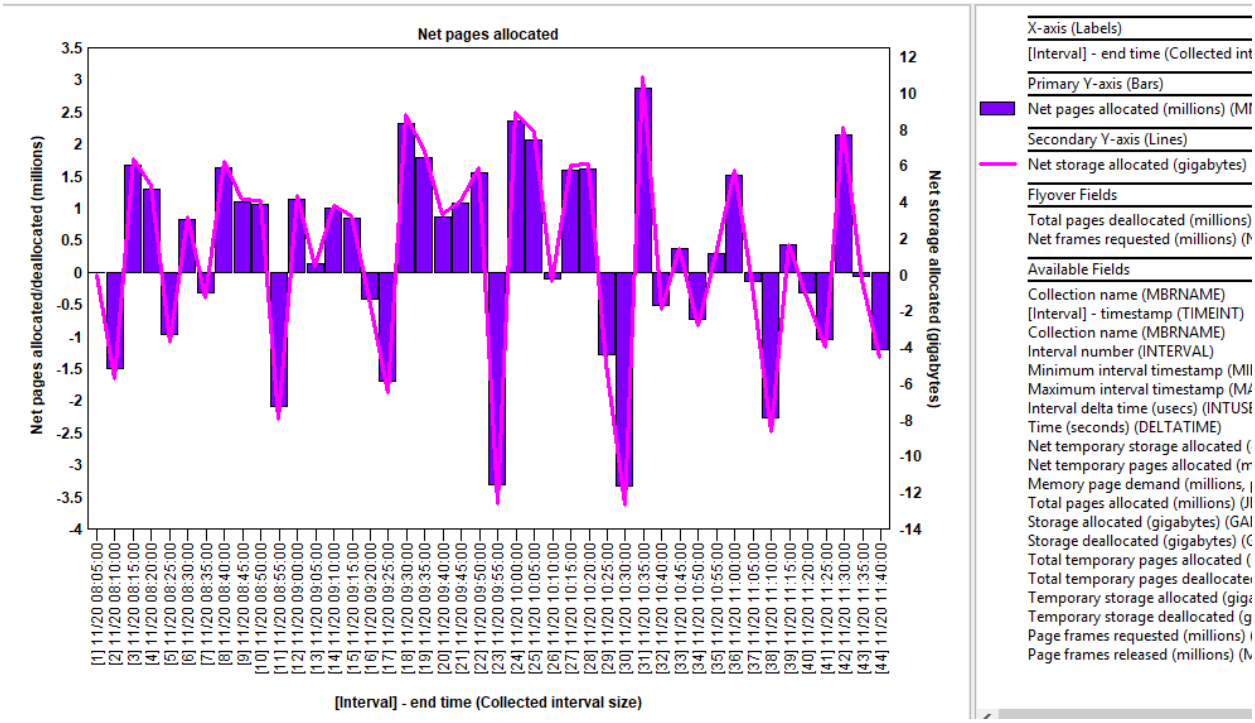
This graph displays the total size of pages allocated and deallocated (in millions) shown with overlapping bars. The larger value will be on top for each bar. The net pages allocated is shown on the secondary Y-axis.



Pages allocated/deallocated - overlapping bars

9.20.4 Net pages allocated

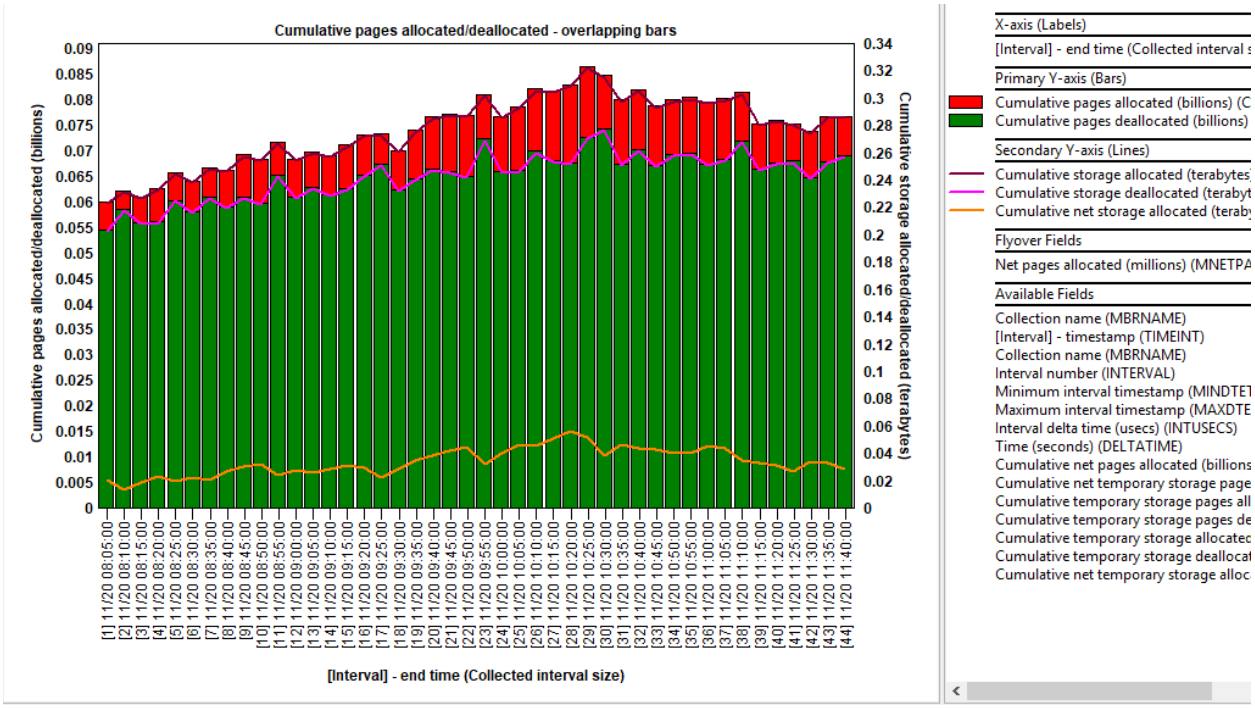
This graph shows the net pages allocated (in millions) on the primary Y-axis with the secondary Y-axis showing the net storage allocated (in gigabytes.)



Net pages allocated

9.20.5 Cumulative pages allocated/deallocated - overlapping bars

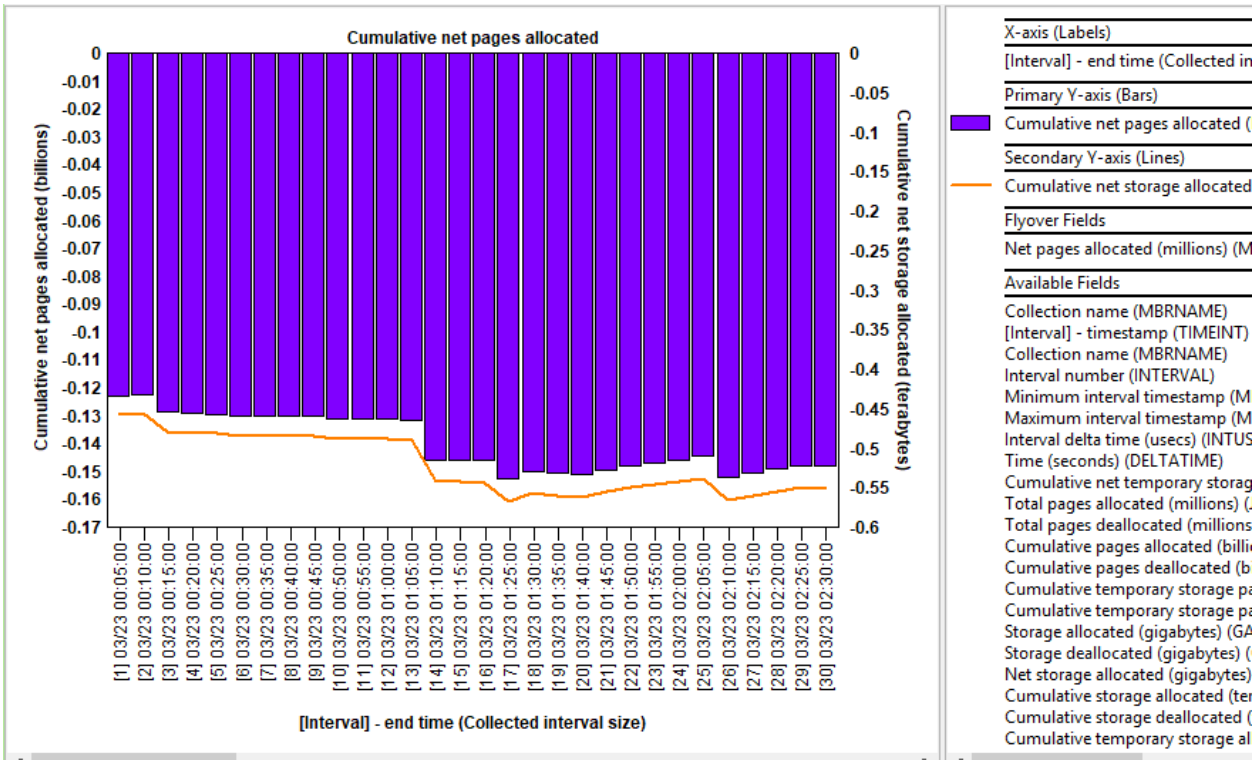
This graph displays the cumulative pages allocated/deallocated with overlapping bars. The larger value will be on top for each bar.



Cumulative pages allocated/deallocated - overlapping bars

9.20.6 Cumulative net pages allocated

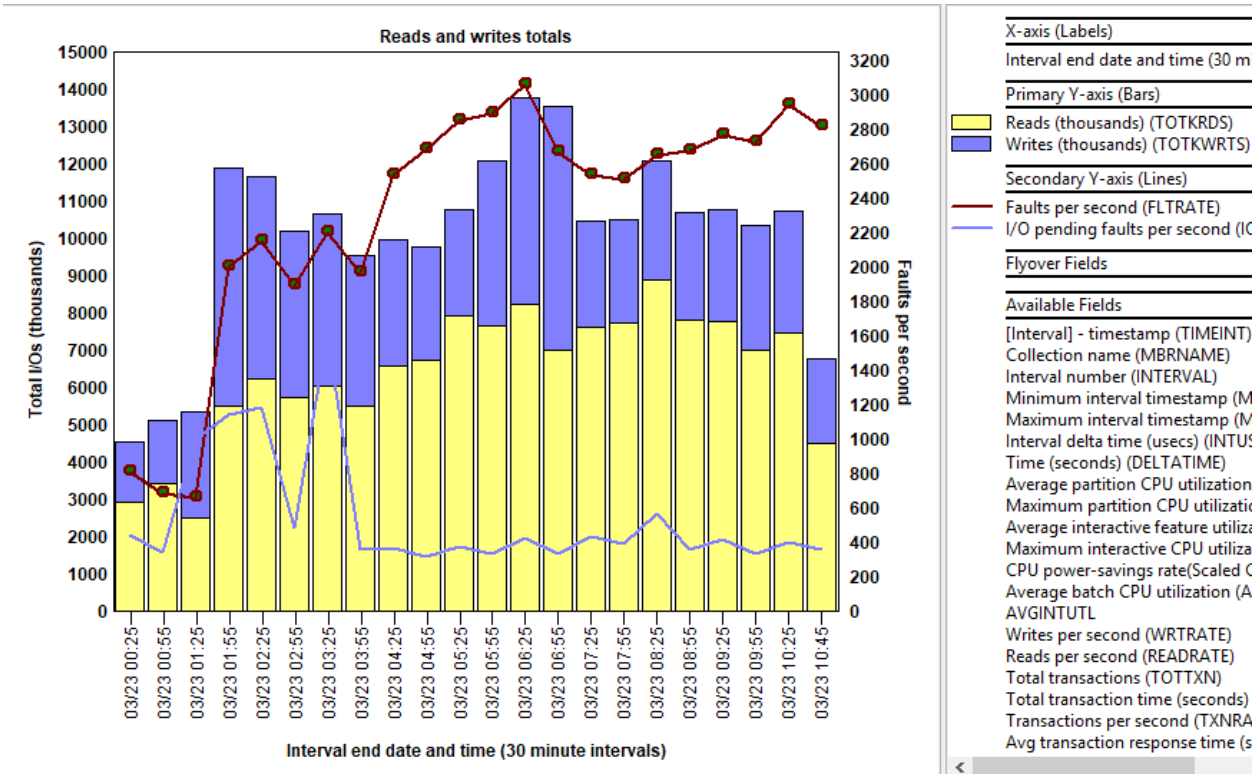
This graph displays cumulative net pages allocated over time.



Cumulative net pages allocated

9.20.7 Reads and writes totals

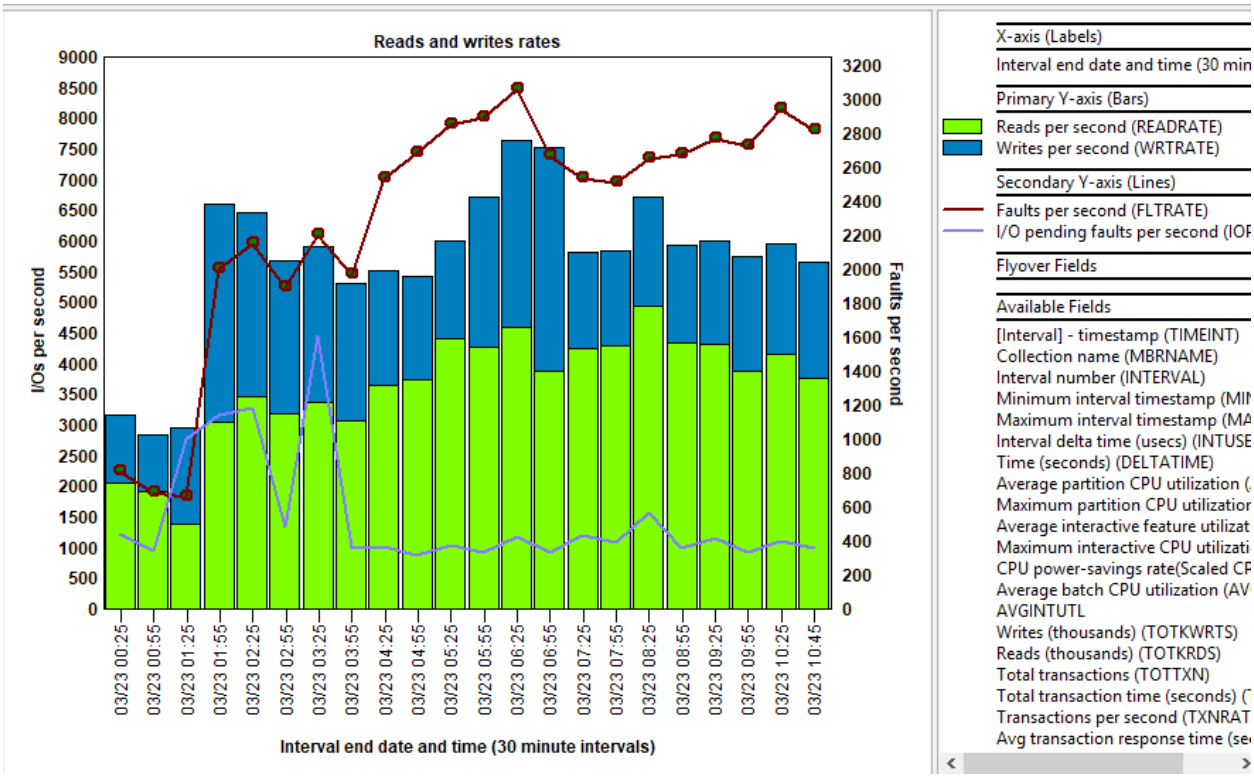
This graph displays the reads and writes total over time. The secondary Y-axis displays the faults per second along with the I/O pending page faults.



Reads and writes totals

9.20.8 Reads and writes rates

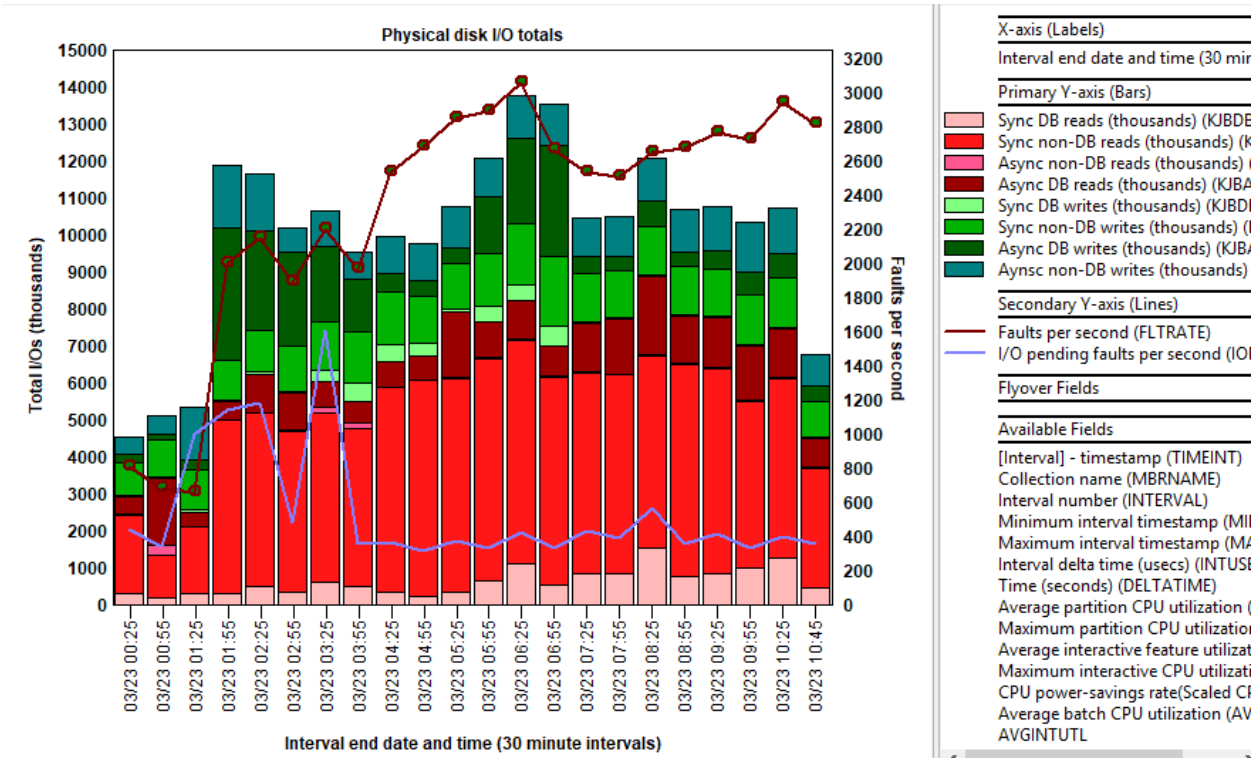
This graph displays the reads and writes rate per second over time. The secondary Y-Axis displays the faults per second along with the I/O pending page faults.



Reads and writes rates

9.20.9 Physical disk I/O totals

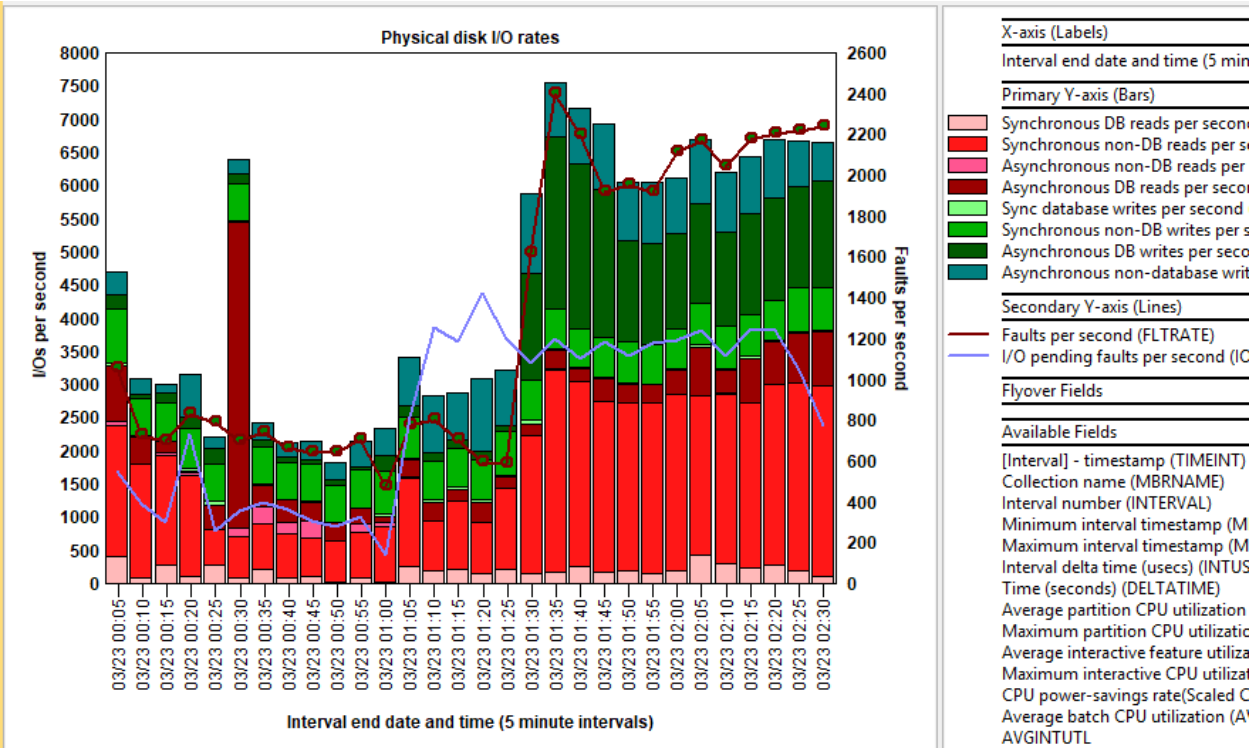
This graph displays the physical disk I/O reads and writes total over time. The secondary Y-axis displays the faults per second along with the I/O pending page faults. This graph contains 8 metrics on the primary Y-axis that show both synchronous and asynchronous DB and non-DB IOs.



Physical disk I/O totals

9.20.10 Physical disk I/O rates

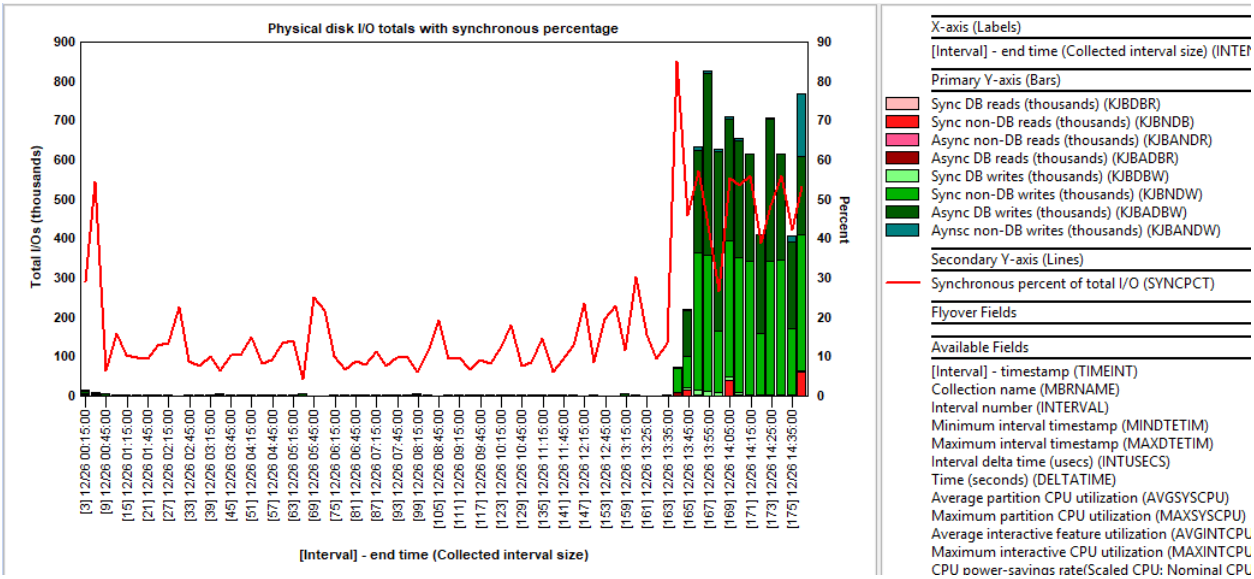
This graph displays the physical disk I/O reads and writes rate per second over time. The secondary Y-Axis displays the faults per second along with the I/O pending page faults. This graph contains 8 metrics on the primary Y-axis that show both synchronous and asynchronous DB and non-DB IOs.



Physical disk I/O rates

9.20.11 Physical disk I/O totals with synchronous percentage

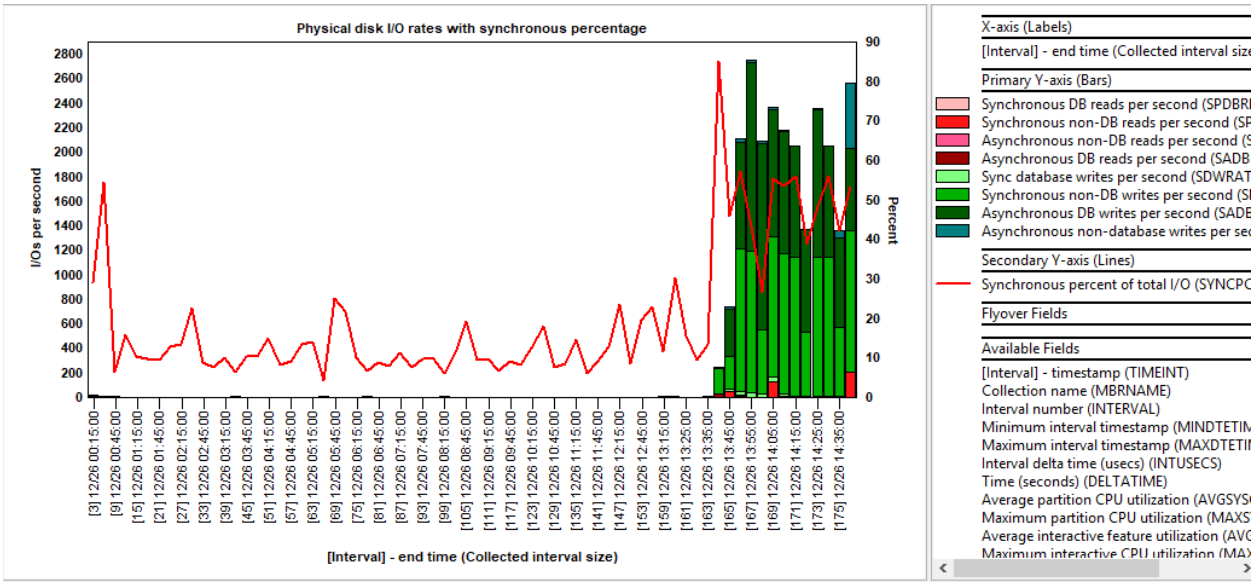
This graph is the same as the physical disk I/O totals graph except the secondary Y-axis displays the percentage of synchronous I/Os of the total I/Os. This graph contains 8 metrics on the primary Y-axis that show both synchronous and asynchronous DB and non-DB IOs.



Physical disk I/O totals with synchronous percentage

9.20.12 Physical disk I/O rates with synchronous percentage

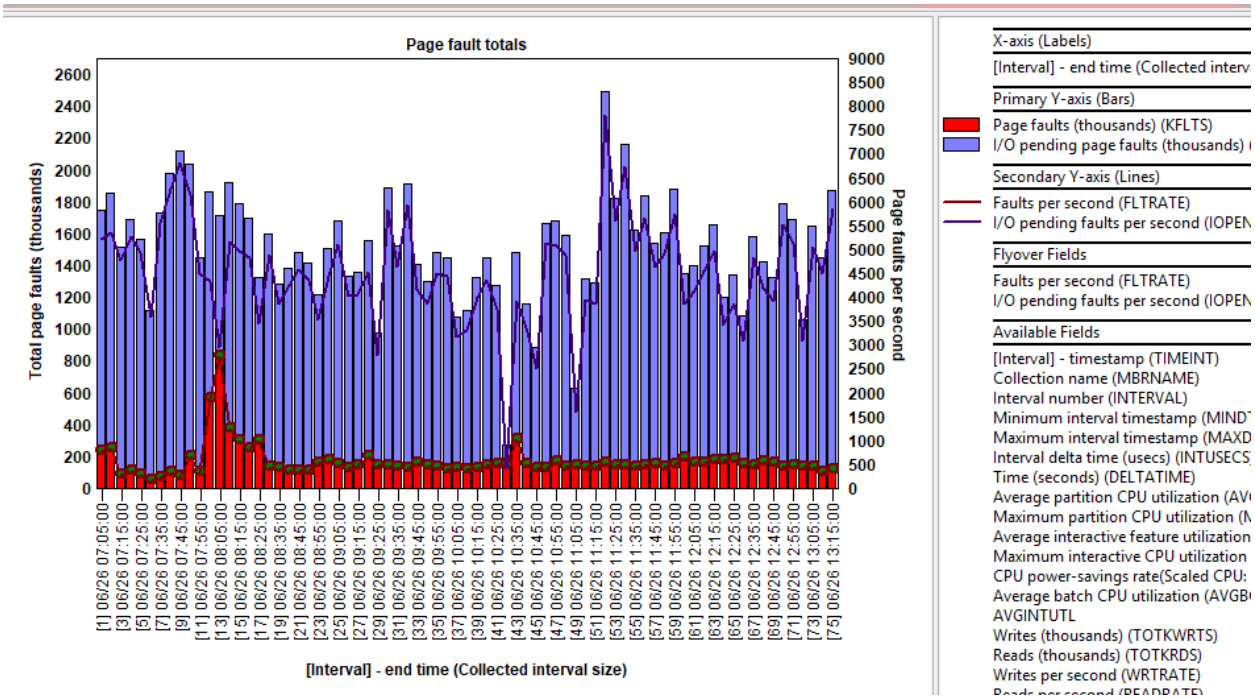
This graph is the same as the physical disk I/O rates graph except the secondary Y-axis displays the percentage of synchronous I/Os of the total I/Os. This graph contains 8 metrics on the primary Y-axis that show both synchronous and asynchronous DB and non-DB IOs.



Physical disk I/O rates with synchronous percentage

9.20.13 Page fault totals

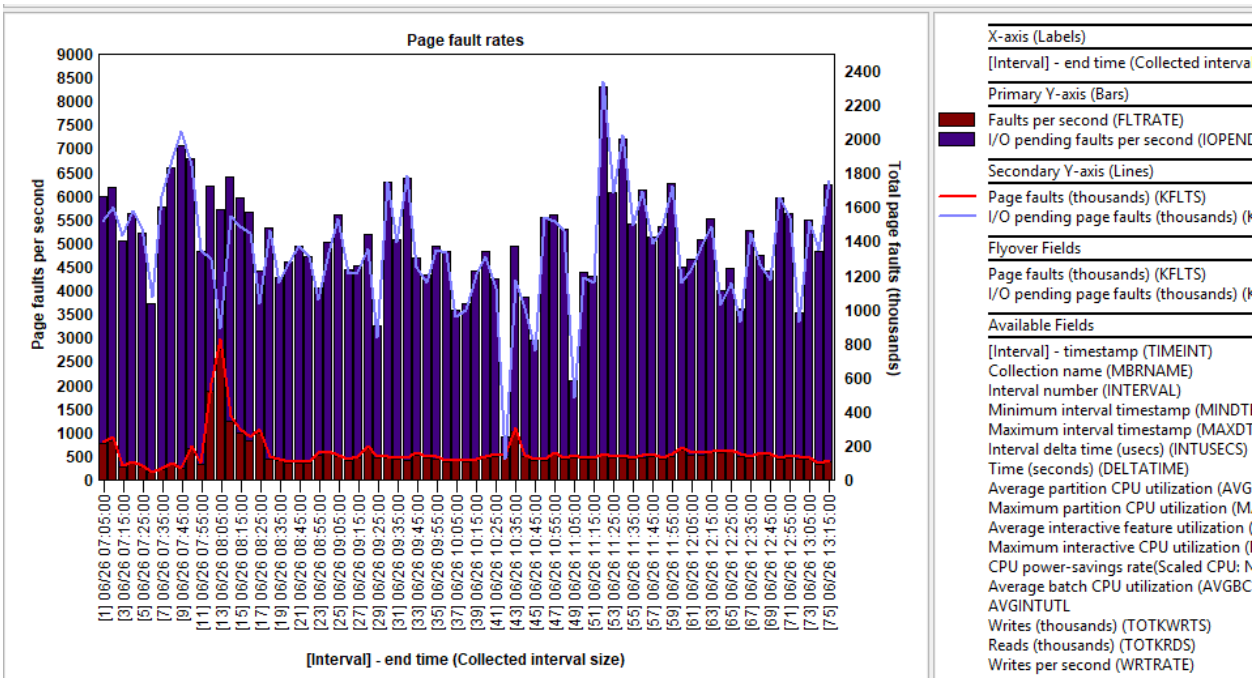
This graph displays the page fault totals (in thousands) on the primary Y-axis with the secondary Y-axis showing page fault rates per second.



Page fault totals

9.20.14 Page fault rates

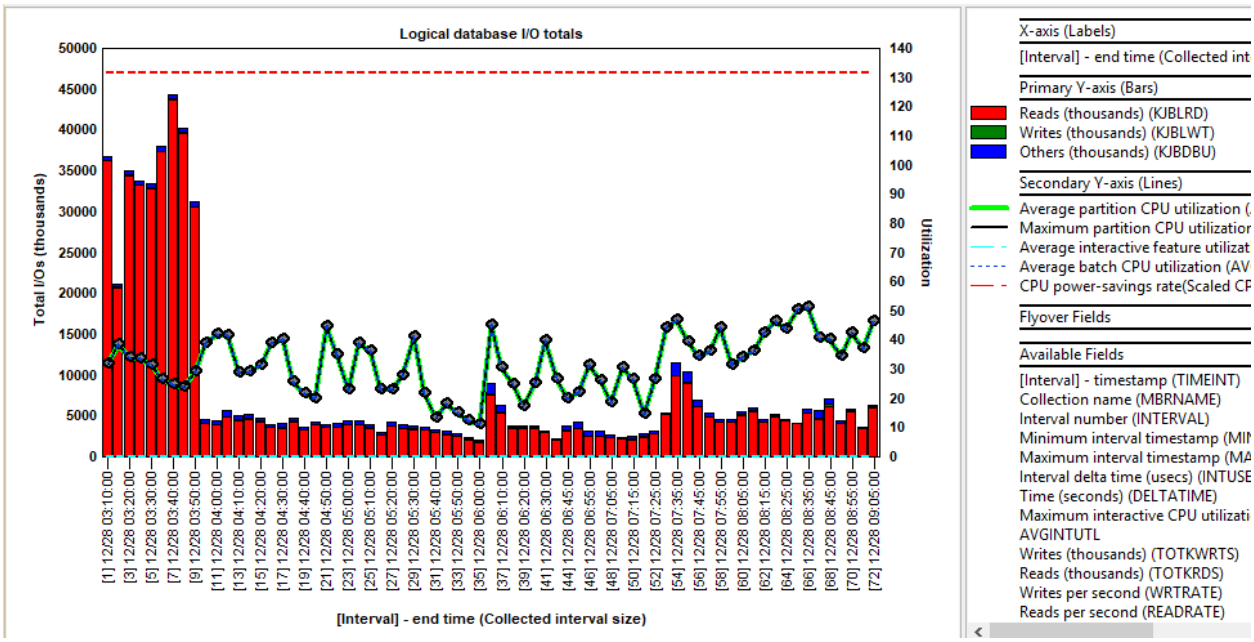
This graph displays the page fault rate per second on the primary Y-axis with the secondary Y-axis showing page fault totals (in thousands.)



Page fault rates

9.20.15 Logical database I/O totals

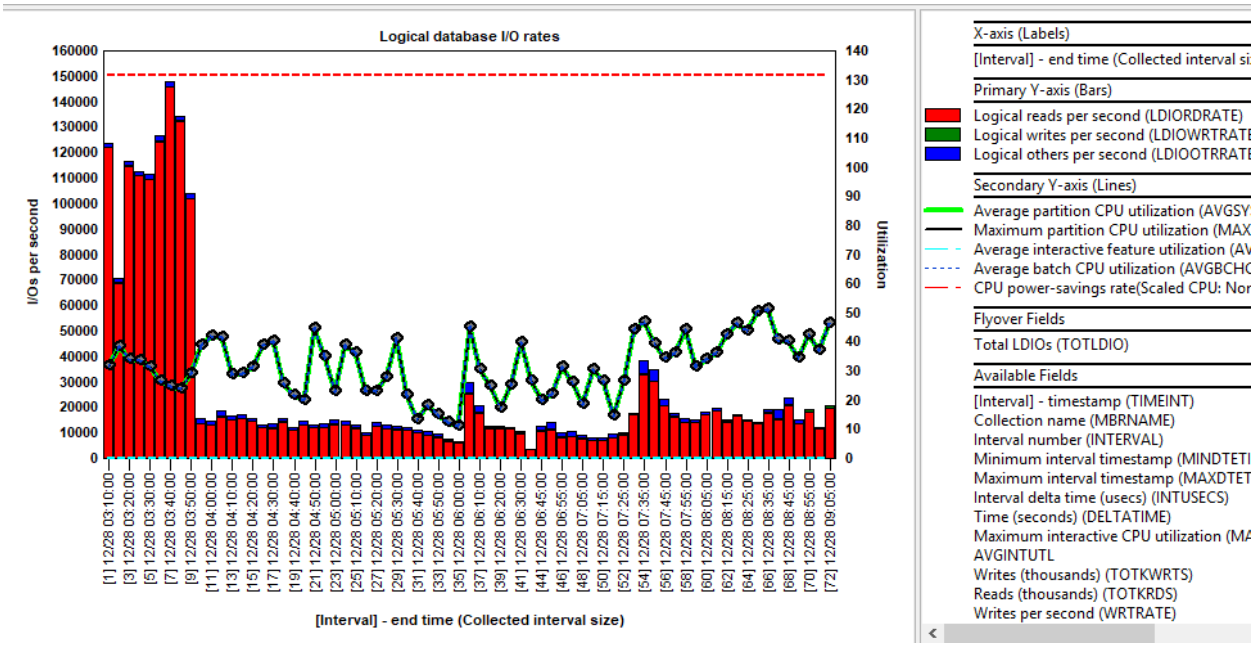
This graph shows the logical database I/O totals over time. This includes reads, writes and others (updates + deletes.)



Logical database I/O totals

9.20.16 Logical database I/O rates

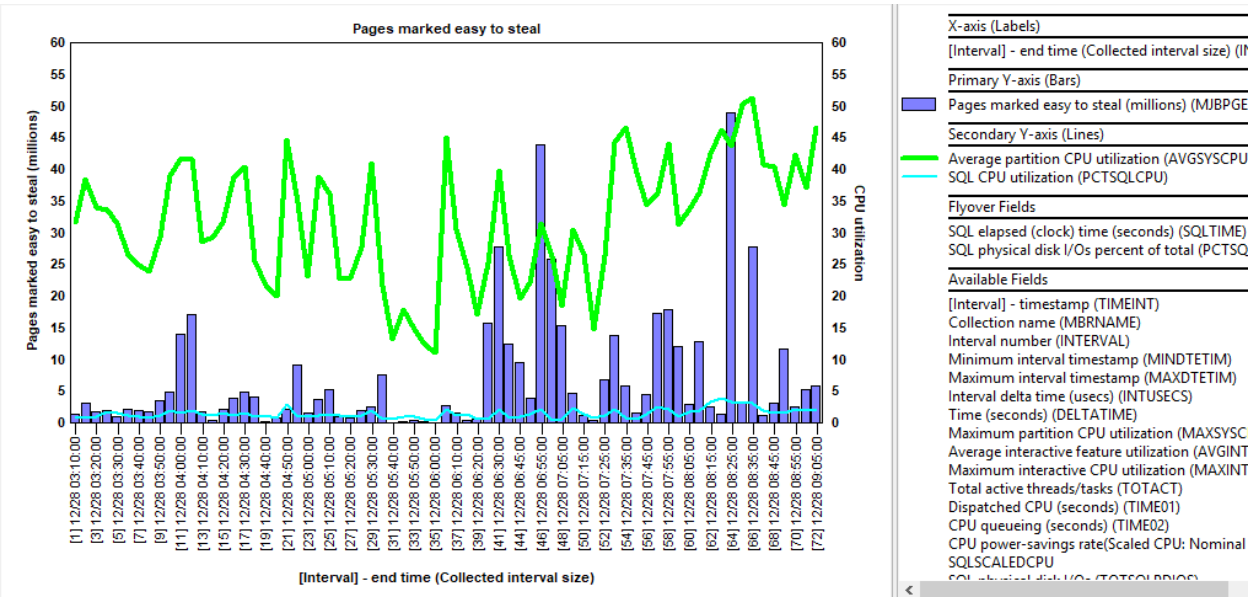
This graph shows the logical database I/O rates per second over time. This includes reads, writes and others (updates + deletes.)



Logical database I/O rates

9.20.17 Pages marked easy to steal

This graph displays the number of pages marked easy to steal (in millions) over time.

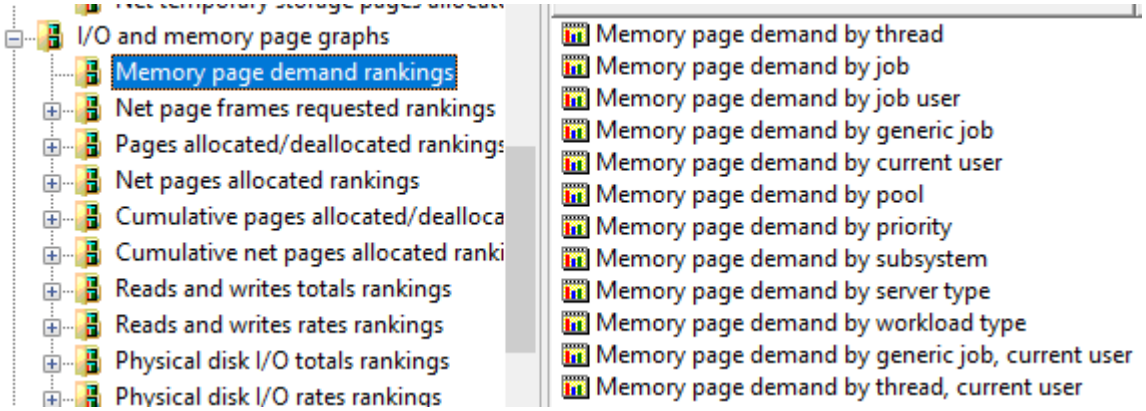


Pages marked easy to steal

9.20.18 Memory page demand rankings

These rankings graphs displays the memory pages requested and released which gives a sense of how much memory is being utilized by the jobs captured by Collection Services.

The bars display the total of the page frames requested and released. The secondary Y-axis displays the 2 values as separate lines instead. The available types of job rankings are shown below:

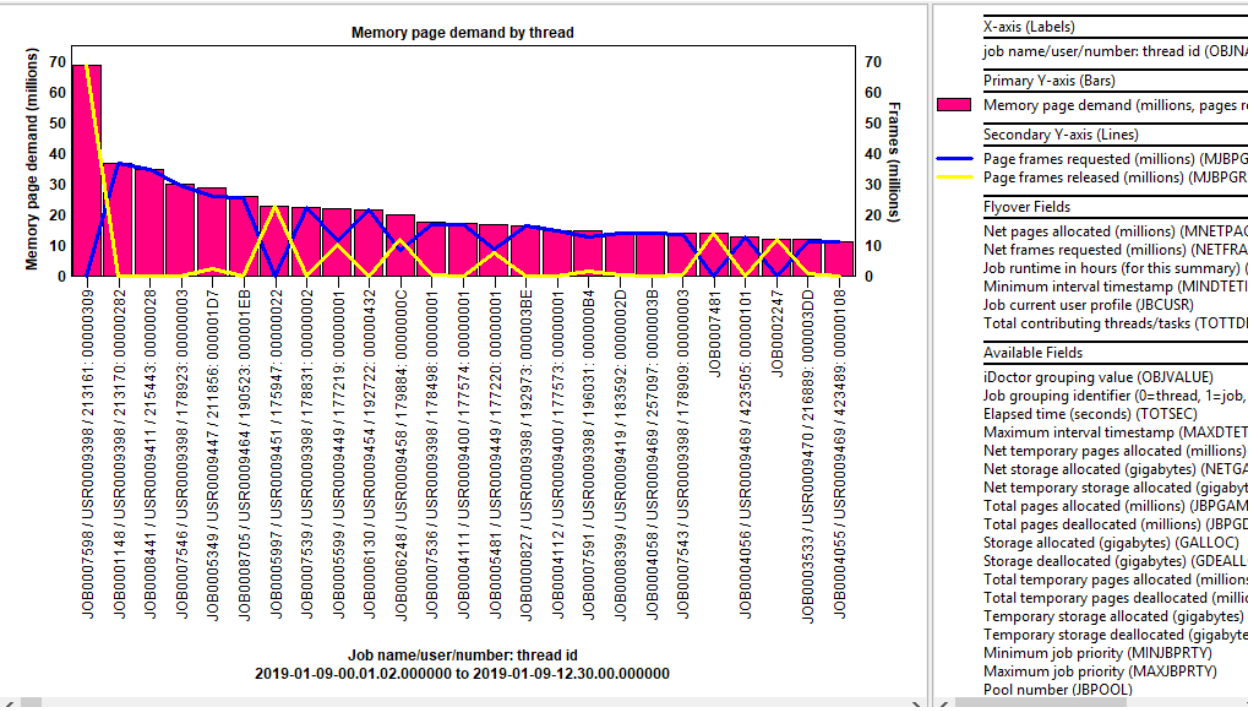


Memory page demand rankings

Tip: These graphs show vertical bar graphs which make the labels harder to read. You can use the Toggle graph format button  on the toolbar to flip these to horizontal bar graphs if desired.

9.20.18.1 Memory page demand by thread

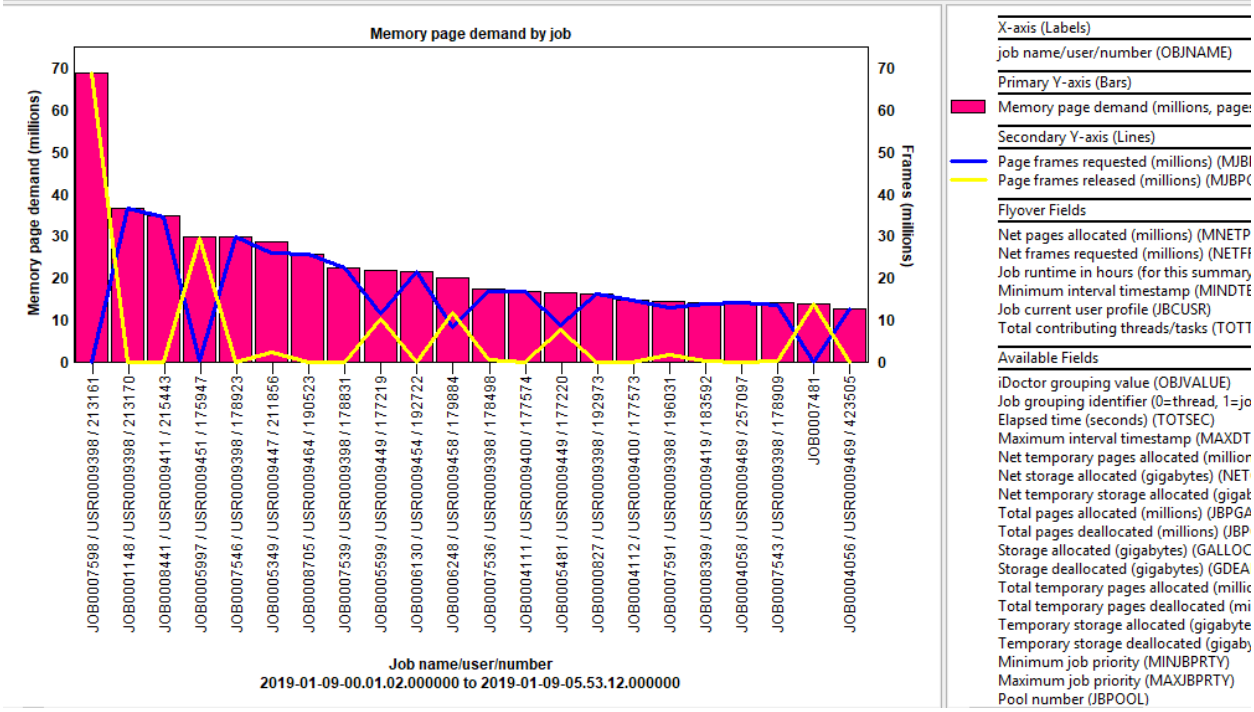
This graph shows the memory page demand (pages requested and released) ranked by thread or system task.



Memory page demand by thread

9.20.18.2 Memory page demand by job

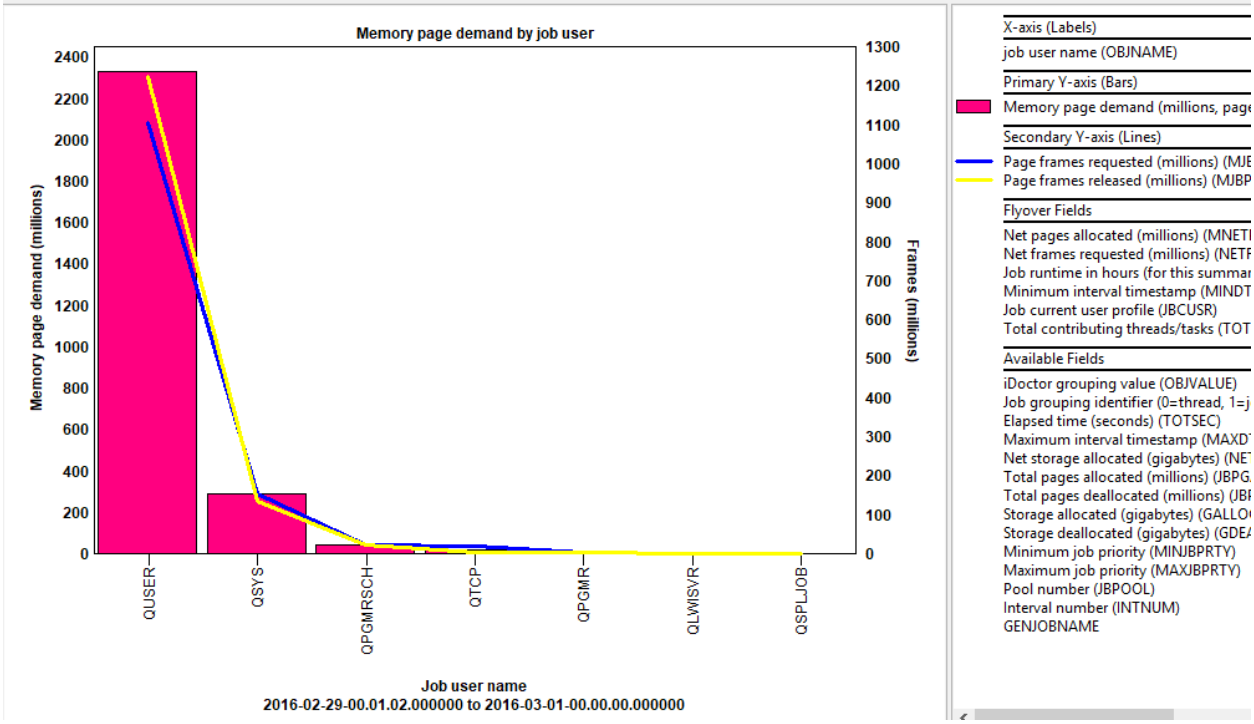
This graph shows the memory page demand (pages requested and released) ranked by job. This is all threads added together or system task.



Memory page demand by job

9.20.18.3 Memory page demand by job user

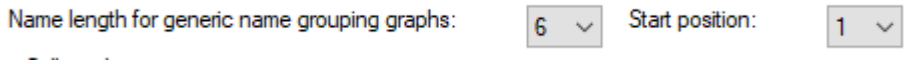
This graph shows the memory page demand (pages requested and released) ranked by job user. All jobs having the same job user name will be added together per bar.

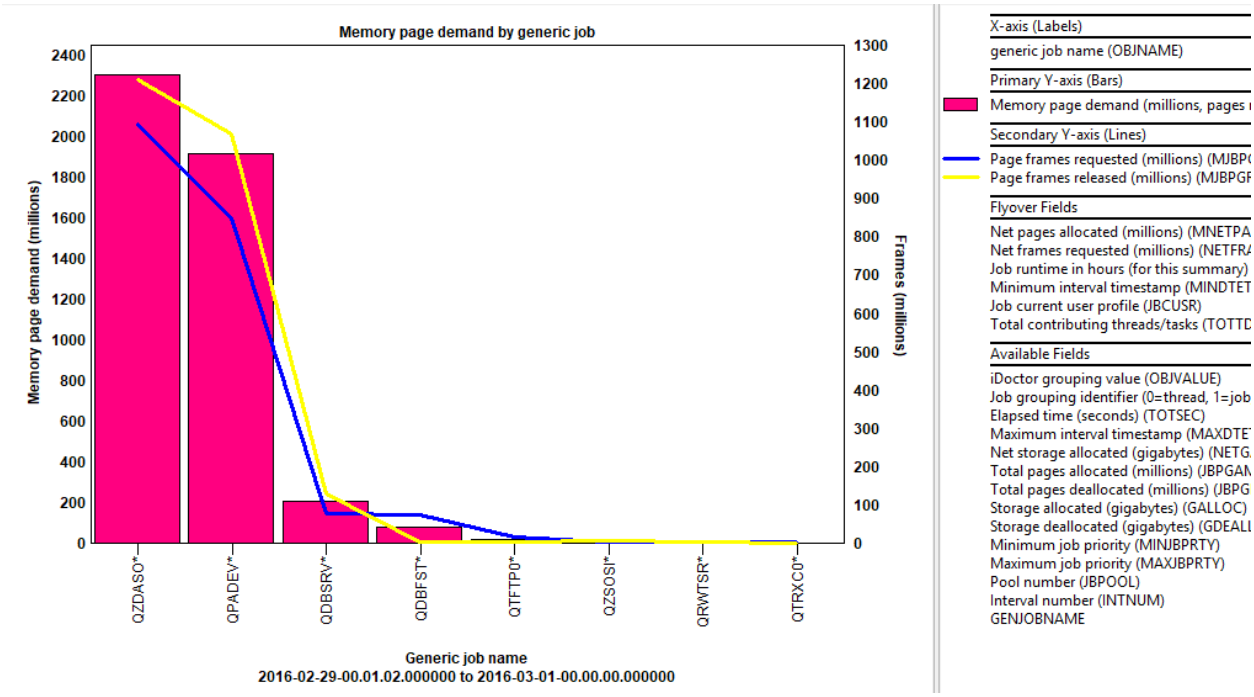


Memory page demand by job user

9.20.18.4 Memory page demand by generic job

This graph shows the memory page demand (pages requested and released) ranked by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:

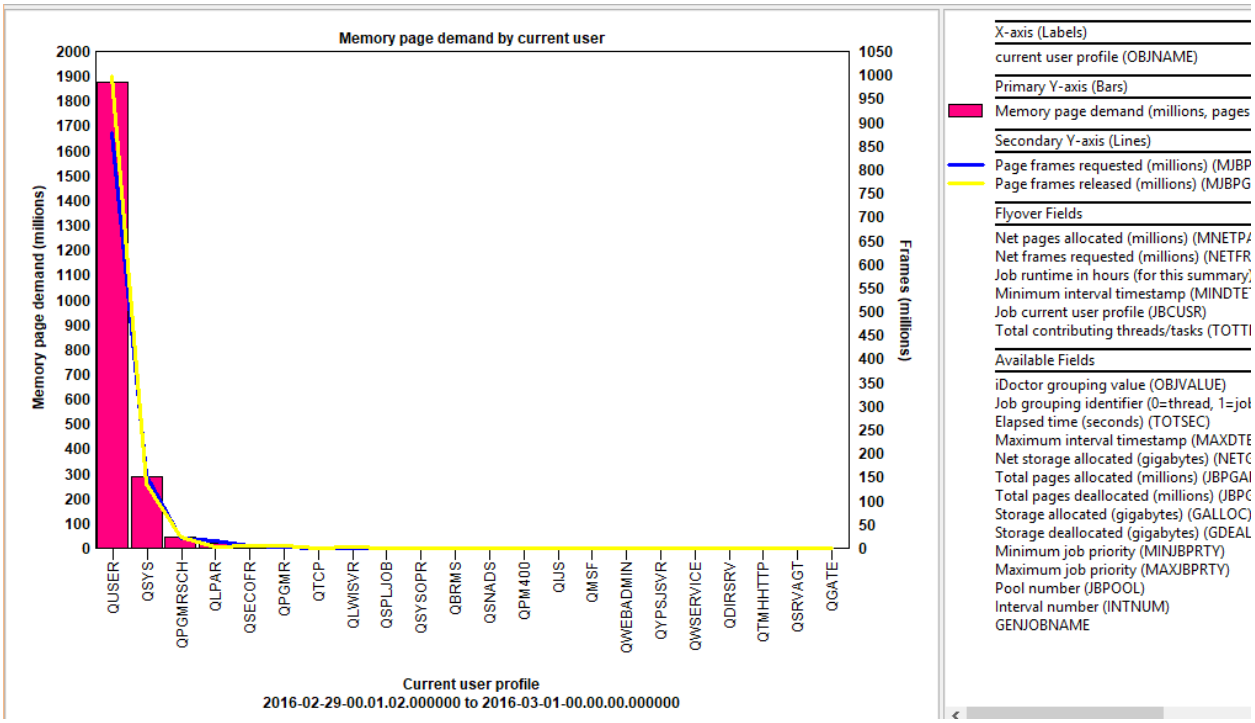




Memory page demand by generic job

9.20.18.5 Memory page demand by current user

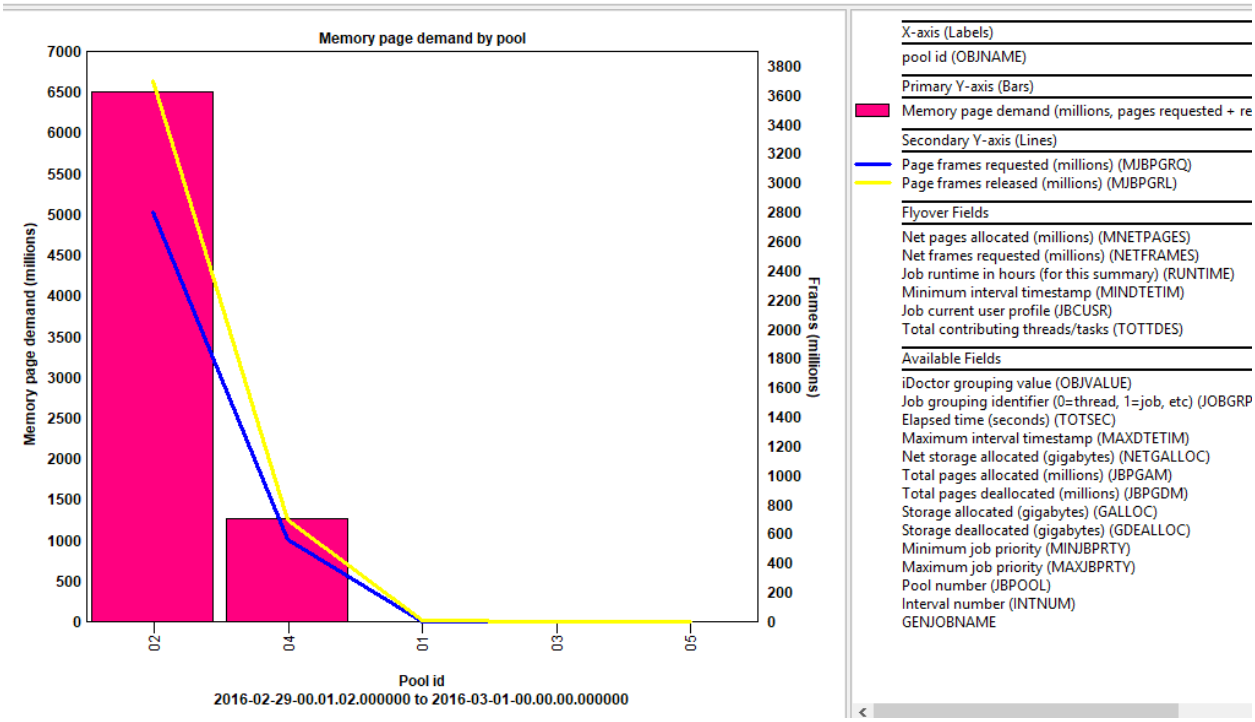
This graph shows the memory page demand (pages requested and released) ranked by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



Memory page demand by current user

9.20.18.6 Memory page demand by pool

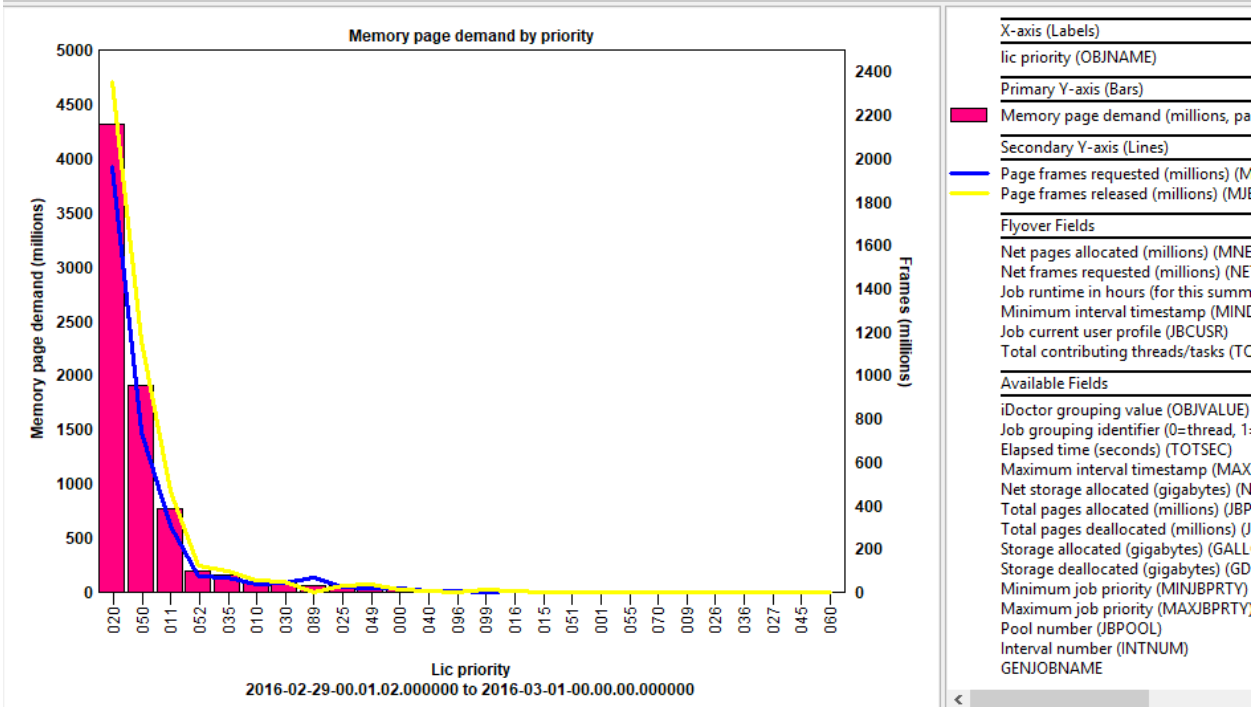
This graph shows the memory page demand (pages requested and released) ranked by memory pool.



Memory page demand by pool

9.20.18.7 Memory page demand by priority

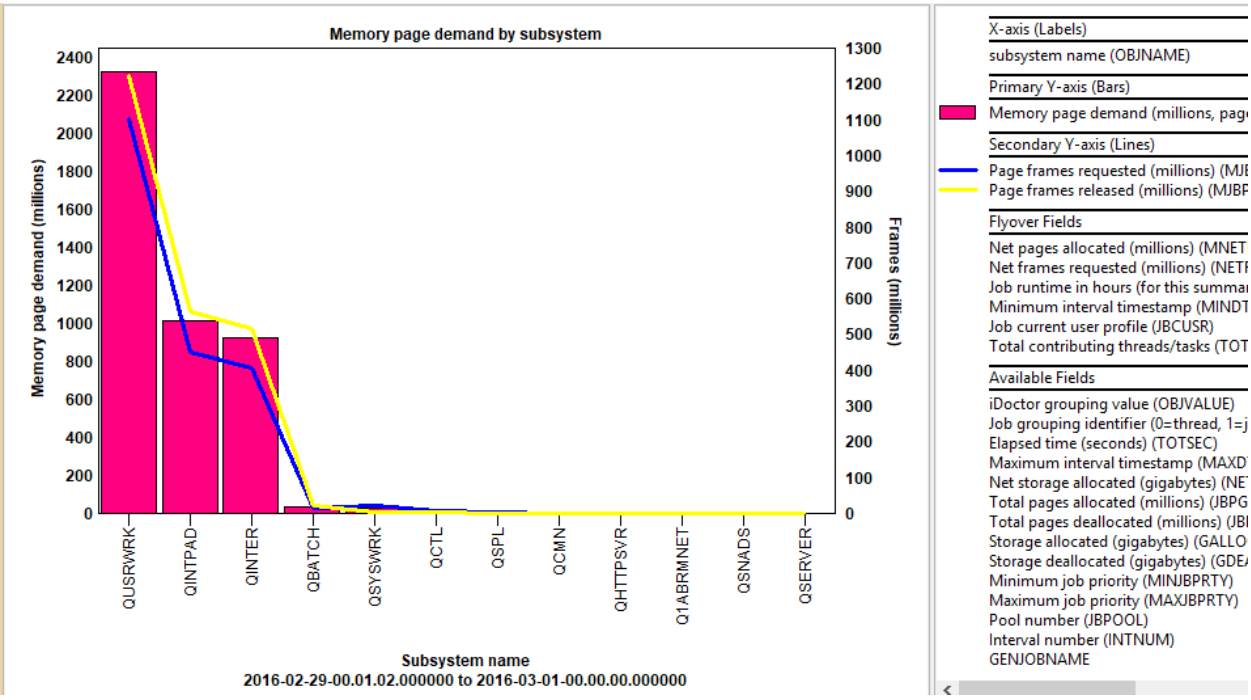
This graph shows the memory page demand (pages requested and released) ranked by LIC priority.



Memory page demand by priority

9.20.18.8 Memory page demand by subsystem

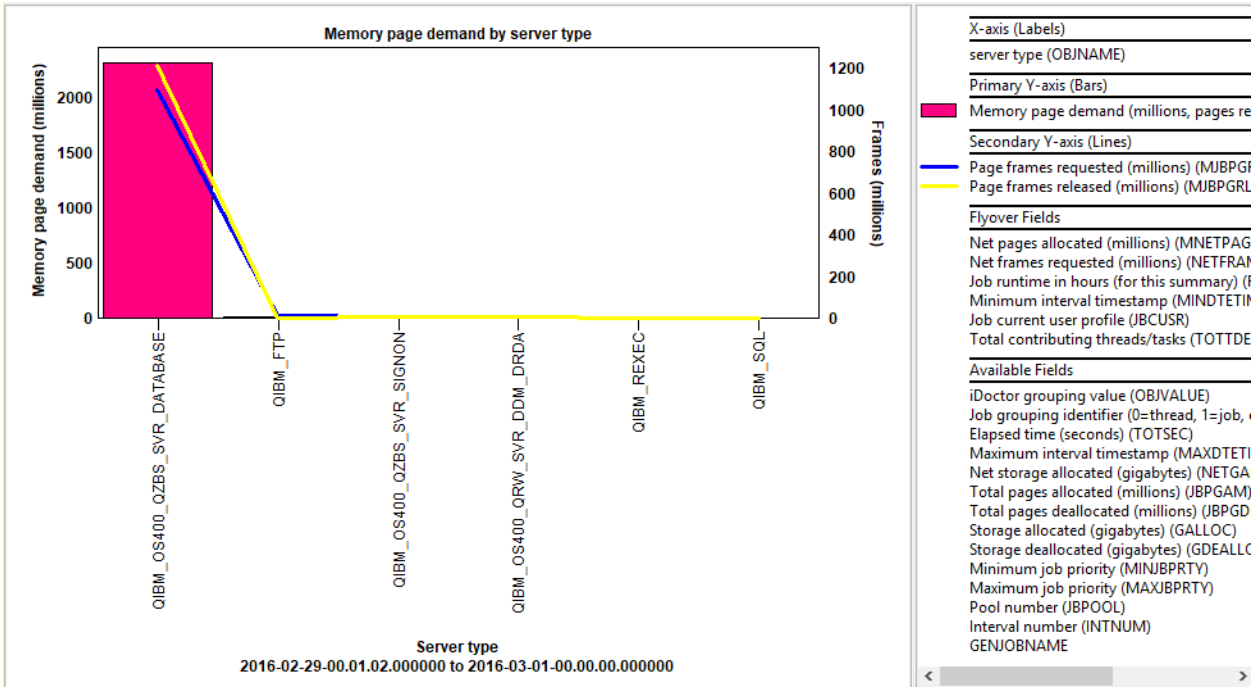
This graph shows the memory page demand (pages requested and released) ranked by subsystem.



Memory page demand by subsystem

9.20.18.9 Memory page demand by server type

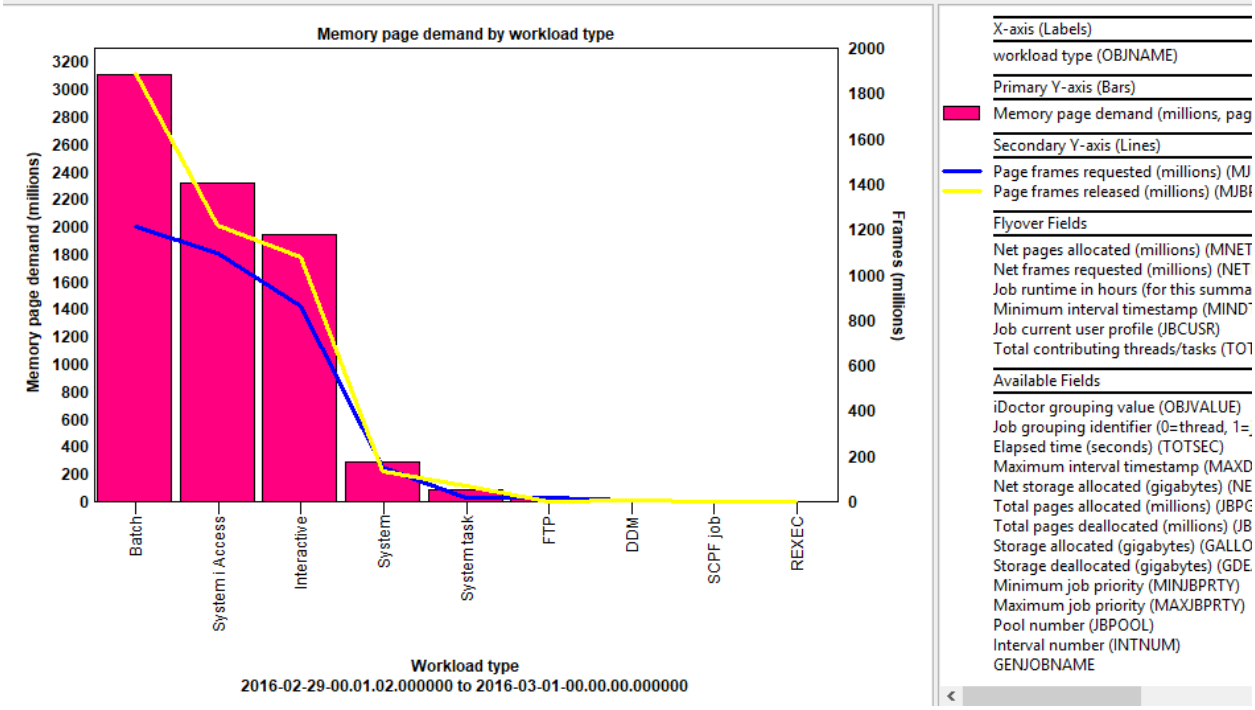
This graph shows the memory page demand (pages requested and released) ranked by server type.



Memory page demand by server type

9.20.18.10 Memory page demand by workload type

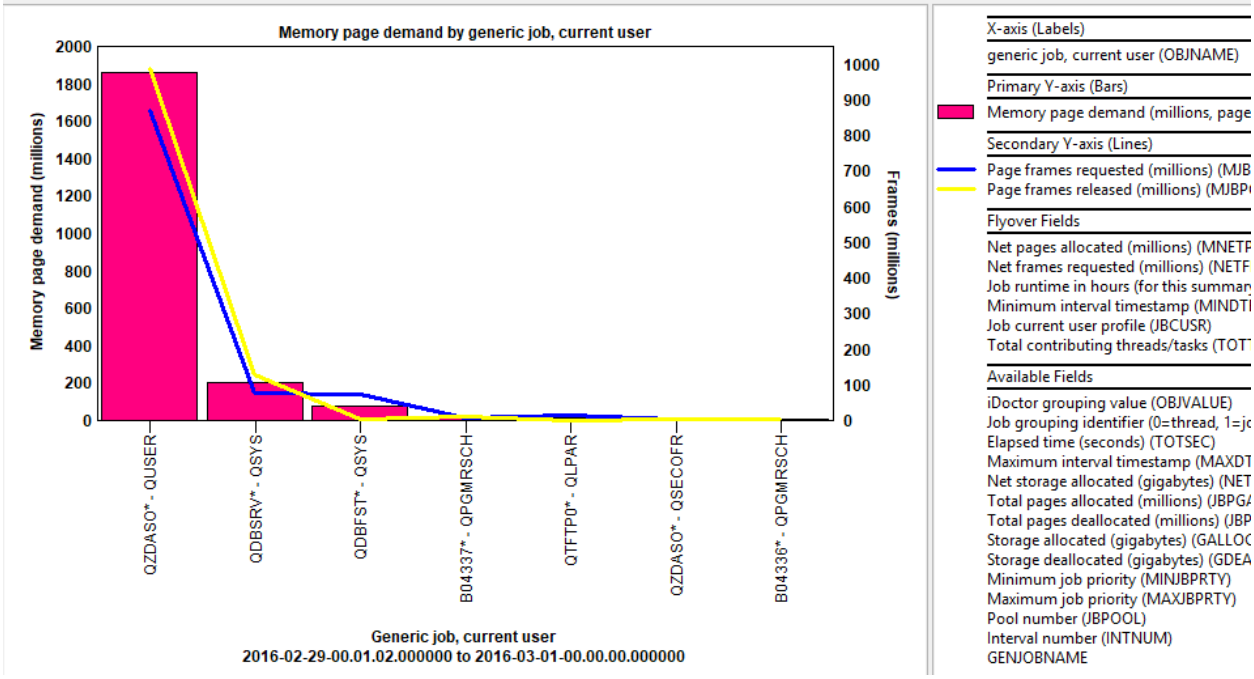
This graph shows the memory page demand (pages requested and released) ranked by workload type.



Memory page demand by workload type

9.20.18.11 Memory page demand by generic job, current user

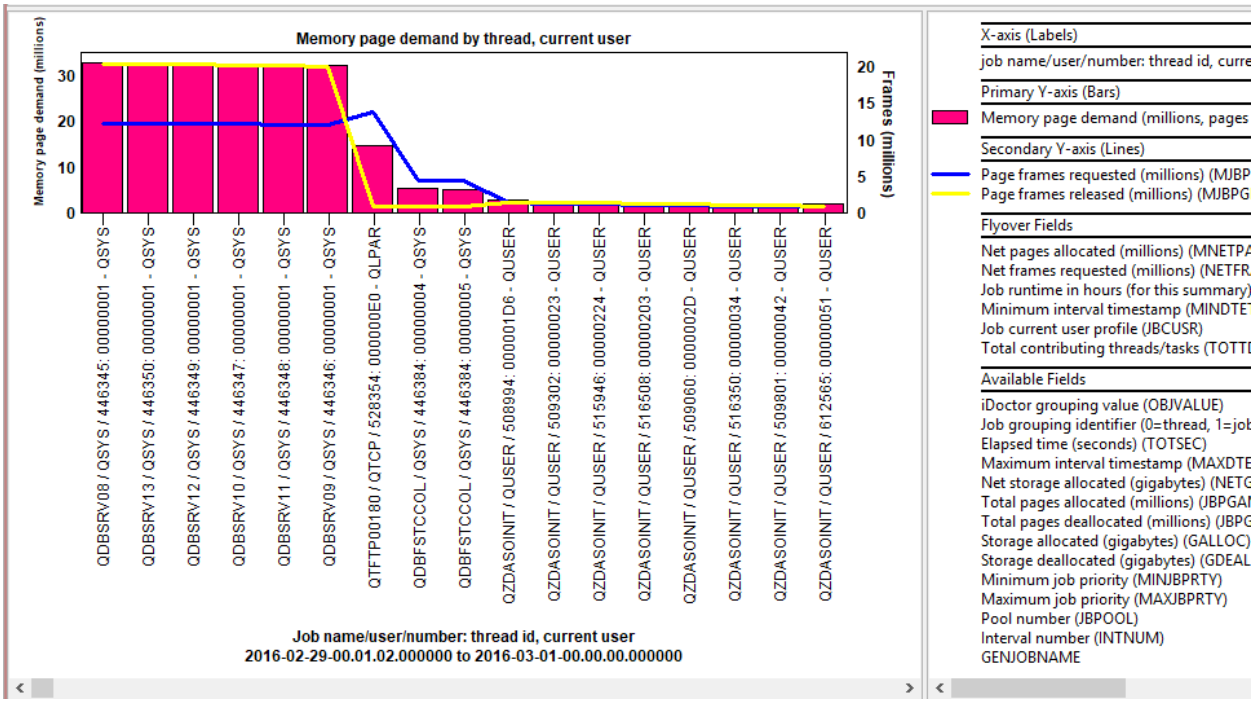
This graph shows the memory page demand (pages requested and released) ranked by generic job and current user combination.



Memory page demand by generic job, current user

9.20.18.12 Memory page demand by thread, current user

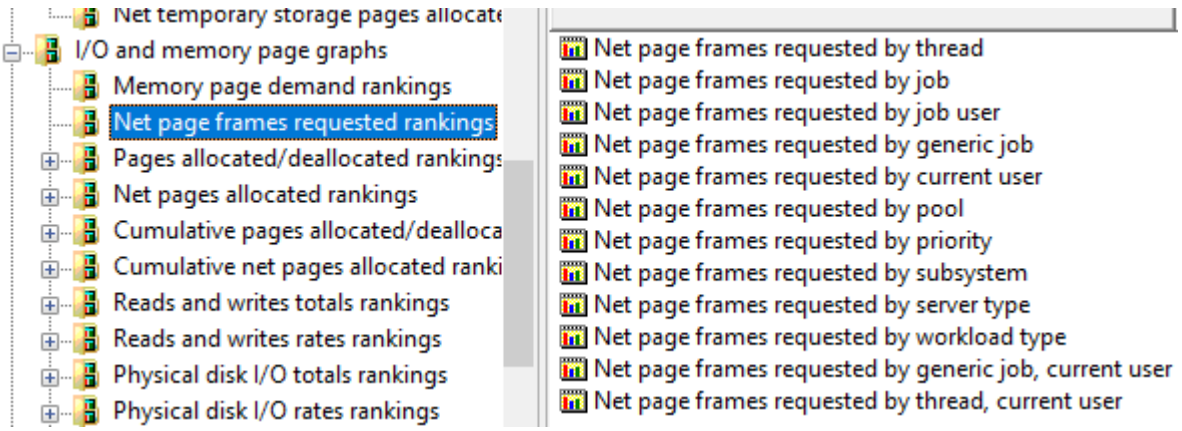
This graph shows the memory page demand (pages requested and released) ranked by thread/task and current user combination.



Memory page demand by thread, current user

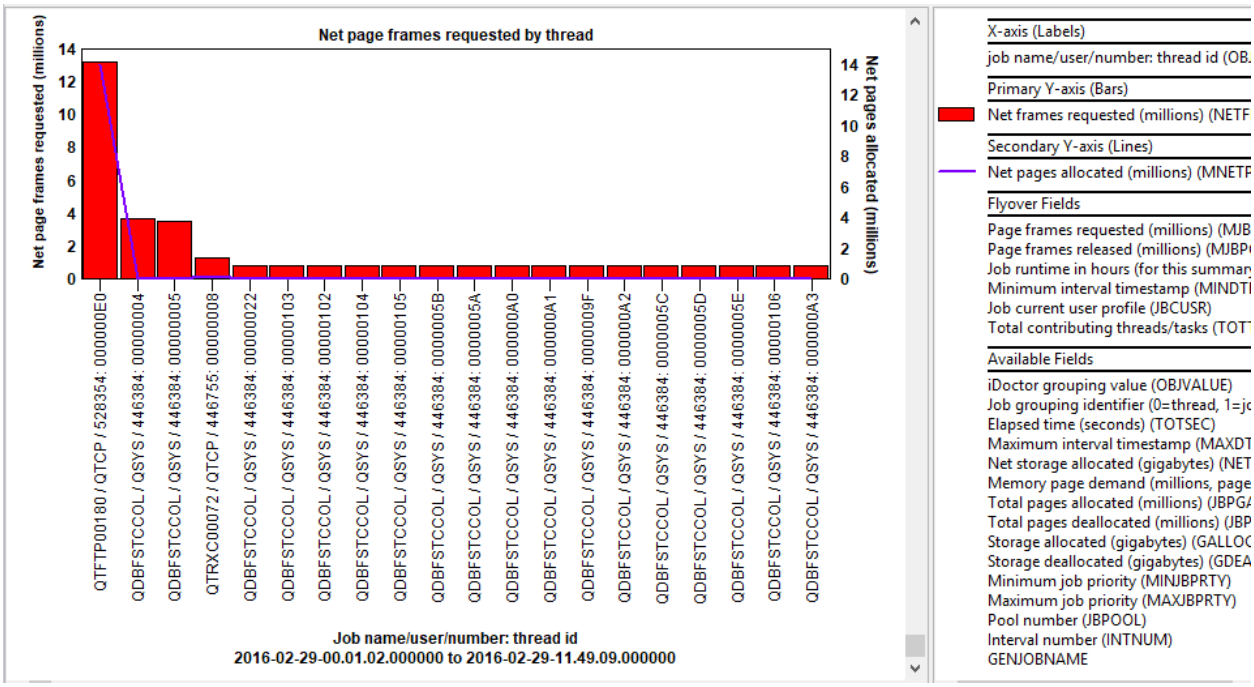
9.20.19 Net page frames requested rankings

These rankings graphs display the net page frames requested. Negative values indicate that more pages were released than requested in that job grouping. The data is ranked by the various job groupings shown below:



Net page frame requested rankings

An example is shown below:



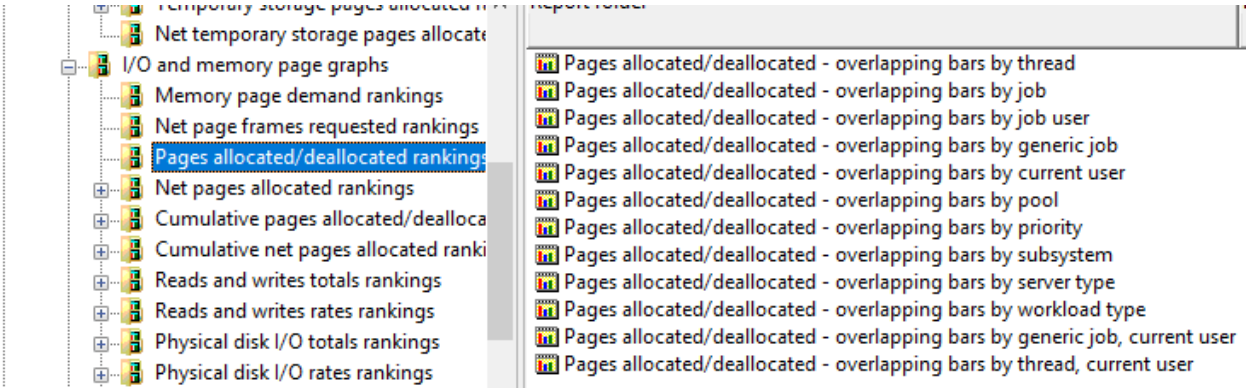
Net page frames requested by thread

For examples showing the other job groupings see the Memory page demand rankings section.

9.20.20 Pages allocated/deallocated rankings

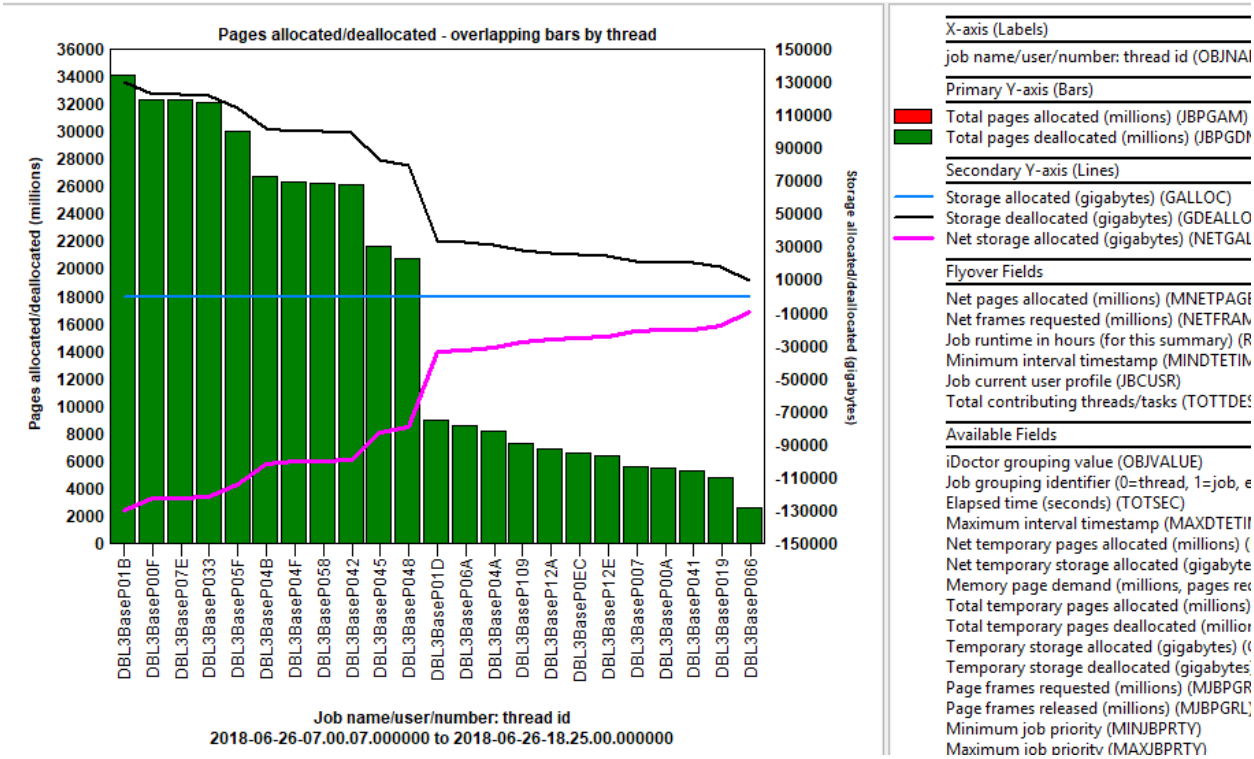
This graph displays the total pages allocated and deallocated (in millions) shown with overlapping bars. The larger value will be on top for each bar. The total size of these allocations or deallocations is shown on the secondary Y-axis in gigabytes.

The data is ranked by the various job groupings shown below:



Pages allocated/deallocated rankings

An example is shown below:



Pages allocated/deallocated – overlapping bars by thread

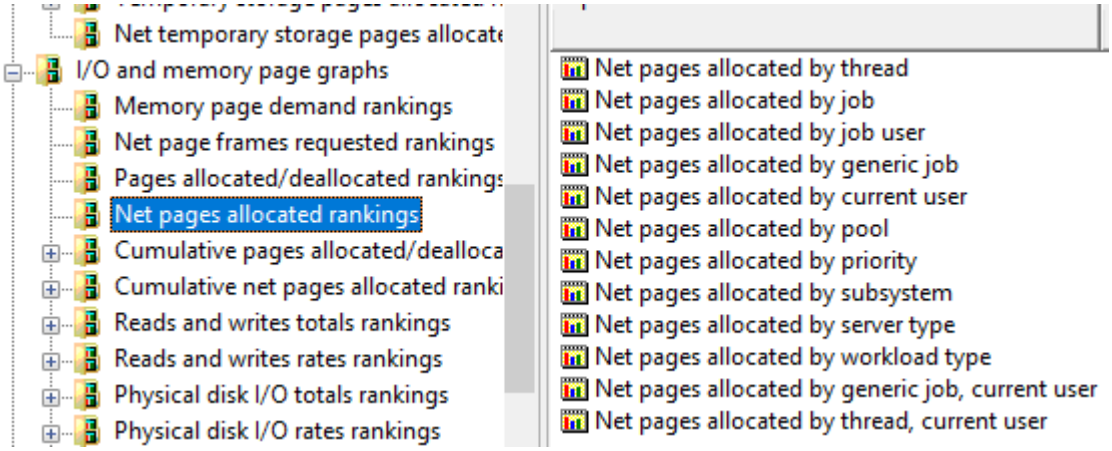
Note: The data shown above is not accurate and is shown for illustrative purposes only. An apparent bug exists in the IBM i for Collection Services file QAPMJOBMI for these values where DBL3 system tasks and other jobs do not show correct numbers of pages deallocated. As of January 2019, no fix is available so be aware of this if you find similar results with huge (impossible) numbers.

Tip: The toggle graph format option does not work for overlapping bar graphs. You would need to convert the graph type to stacked vertical bars (or stacked horizontal bars) in order to use that option.

For examples showing the other job groupings see the Memory page demand rankings section.

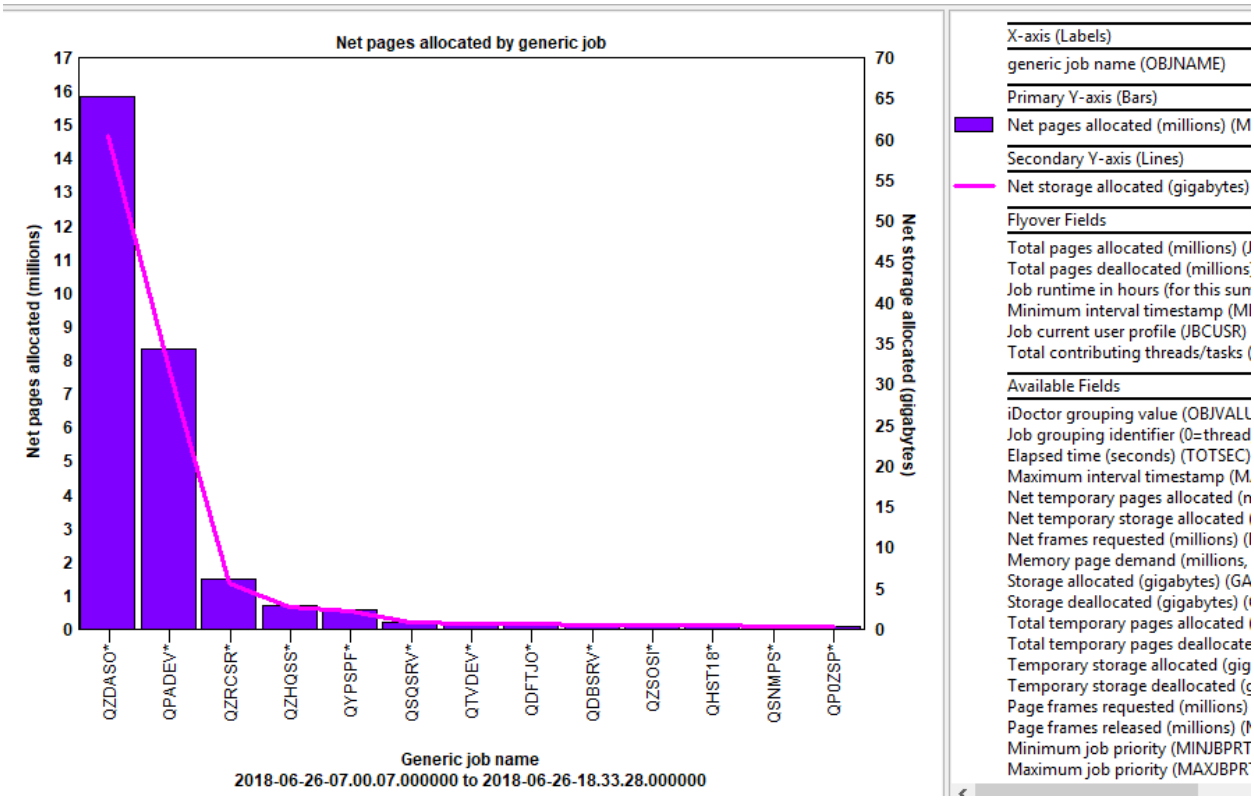
9.20.21 Net pages allocated rankings

These rankings graphs display the net pages allocated. Negative values indicate that more pages were deallocated than allocated in that job grouping. The data is ranked by the various job groupings shown below:



I/O and memory page graphs -> Net pages allocated rankings

An example is shown below:



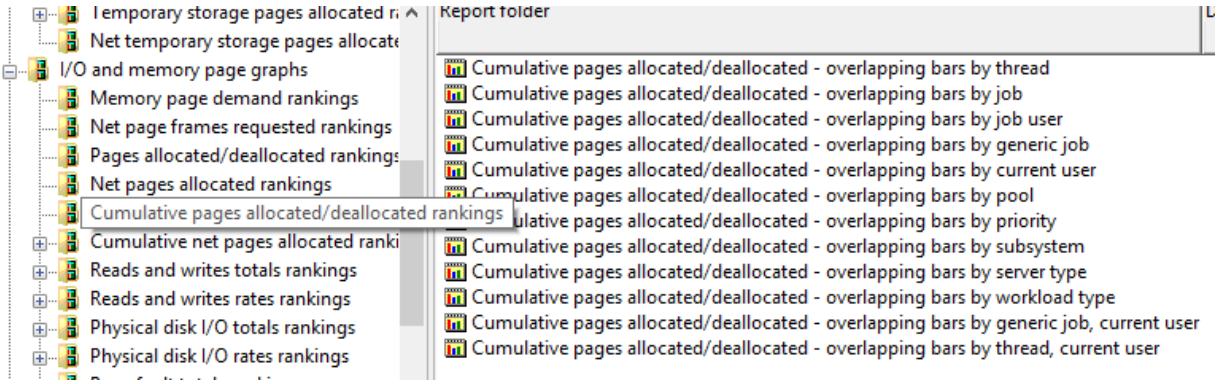
Net pages allocated by generic job

For examples showing the other job groupings see the Memory page demand rankings section.

9.20.22 Cumulative pages allocated/deallocated rankings

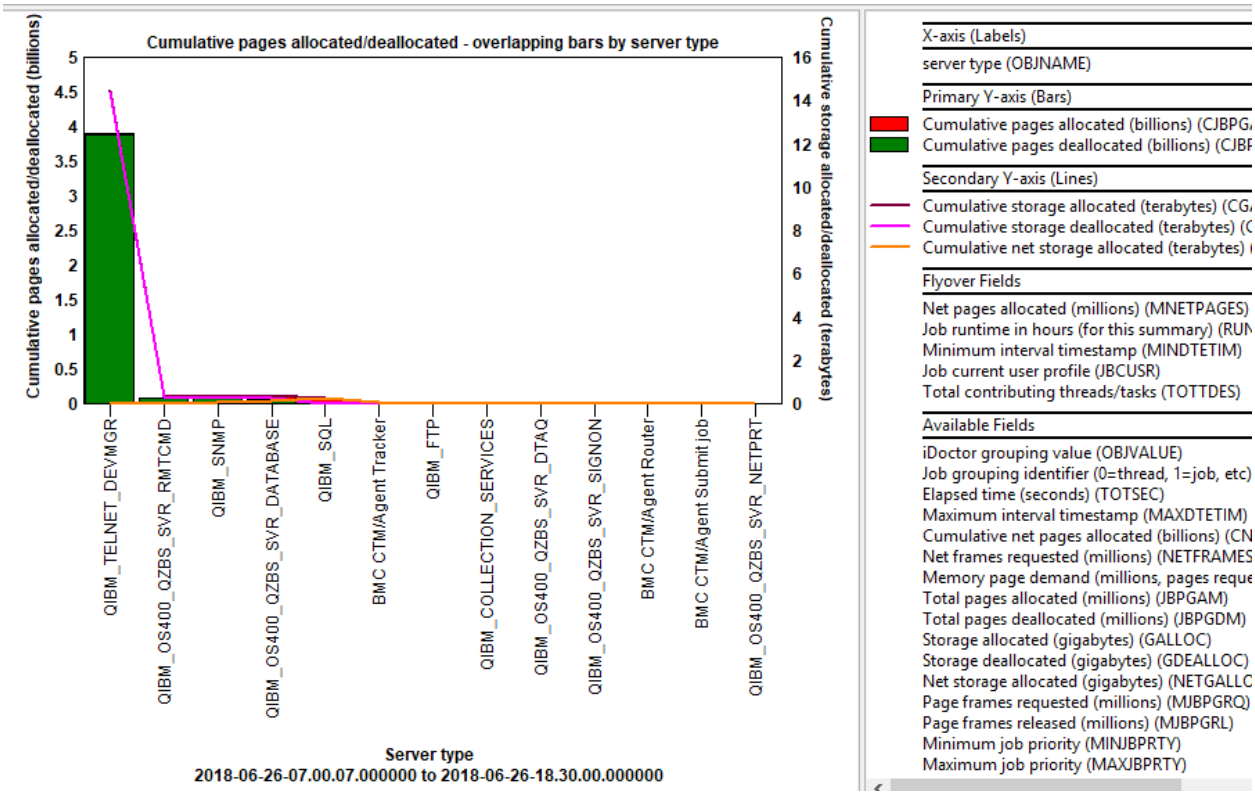
These graph displays the cumulative pages allocated/deallocated with overlapping bars. The larger value will be on top for each bar. Cumulative values means that we are graphing the number of pages allocated/deallocated since each job started and adding that value together. Because of the large numbers encountered the primary Y-axis is in billions and the secondary Y-axis is in terrabytes.

The data is ranked by the various job groupings shown below:



I/O and memory page graphs -> Cumulative pages allocated/deallocated rankings

An example is shown below:



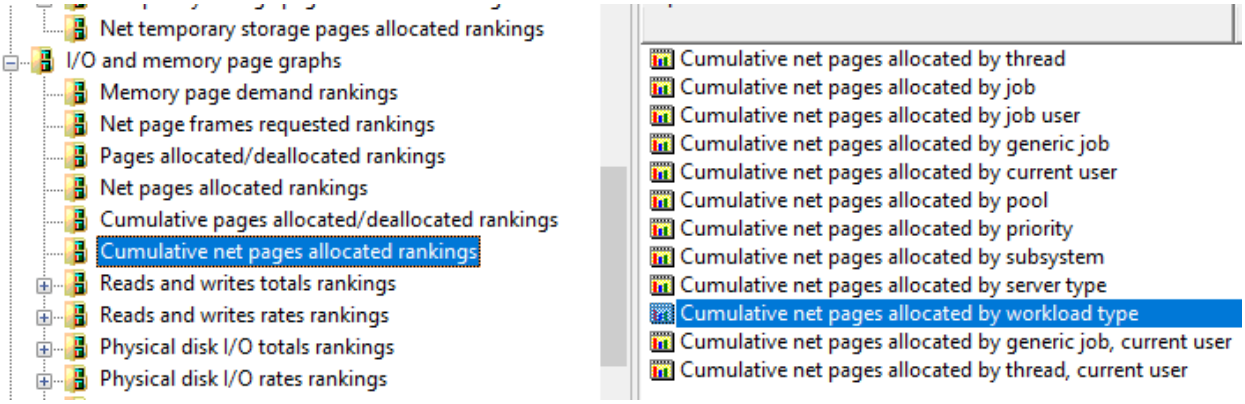
Cumulative pages allocated/deallocated – overlapping bars by server type

For examples showing the other job groupings see the Memory page demand rankings section.

9.20.23 Cumulative net pages allocated rankings

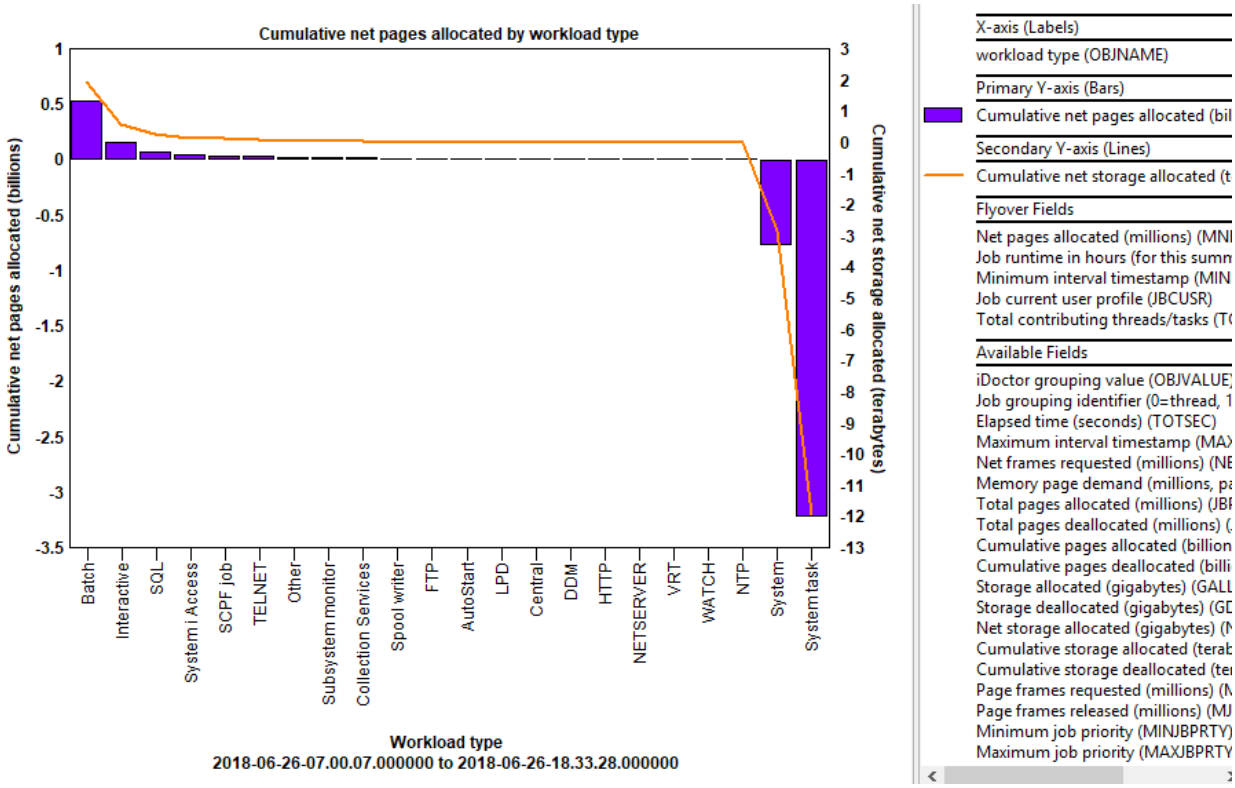
These graphs display cumulative net pages allocated ranked by one of several job groupings. The larger value will be on top for each bar. Because of the large numbers encountered the primary Y-axis is in billions and the secondary Y-axis is in terrabytes.

The data is ranked by the various job groupings shown below:



I/O and memory page graphs -> Cumulative net pages allocated rankings

An example is shown below:

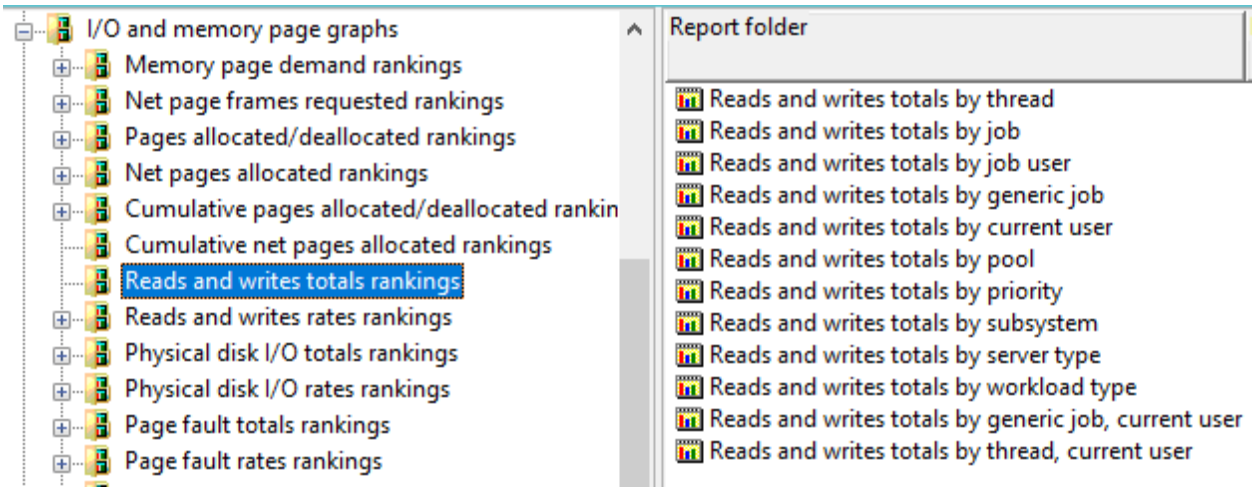


Cumulative net pages allocated by workload type

For examples showing the other job groupings see the Memory page demand rankings section.

9.20.24 Reads and writes totals rankings

These rankings graphs displays the reads and writes totals ranked by several types of job groupings as shown below:

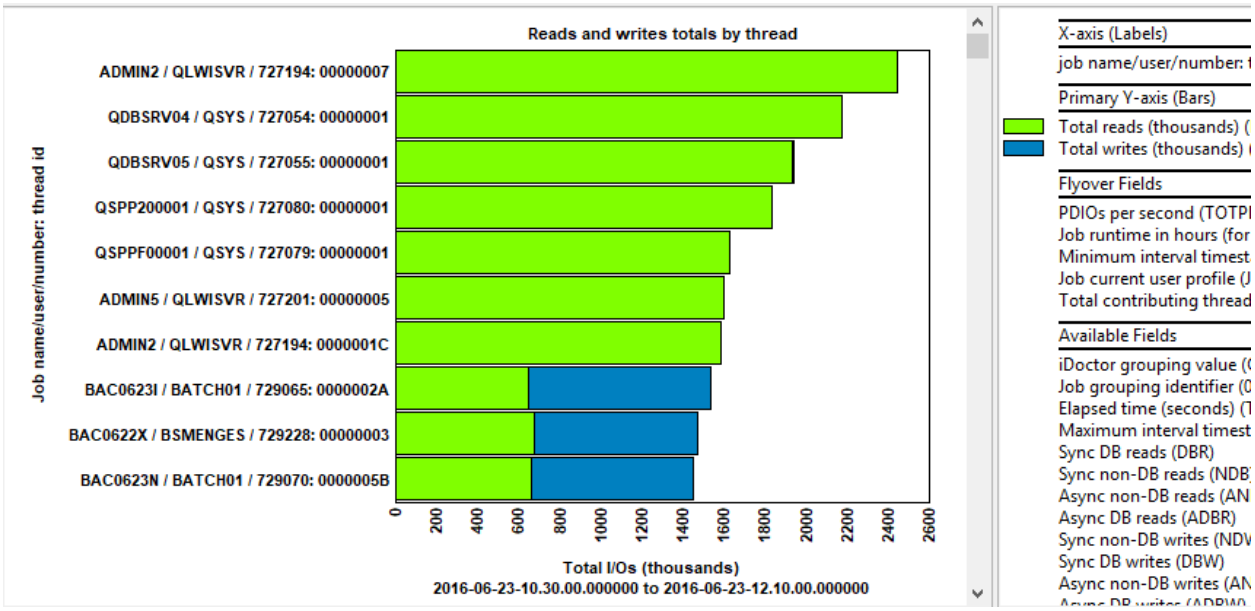


I/O and memory page graphs -> Reads and writes totals rankings

Tip: These graphs show horizontal bar graphs which make the labels easier to read. You can use the Toggle graph format button  on the toolbar to flip these to vertical bar graphs if desired.

9.20.24.1 Reads and writes totals by thread

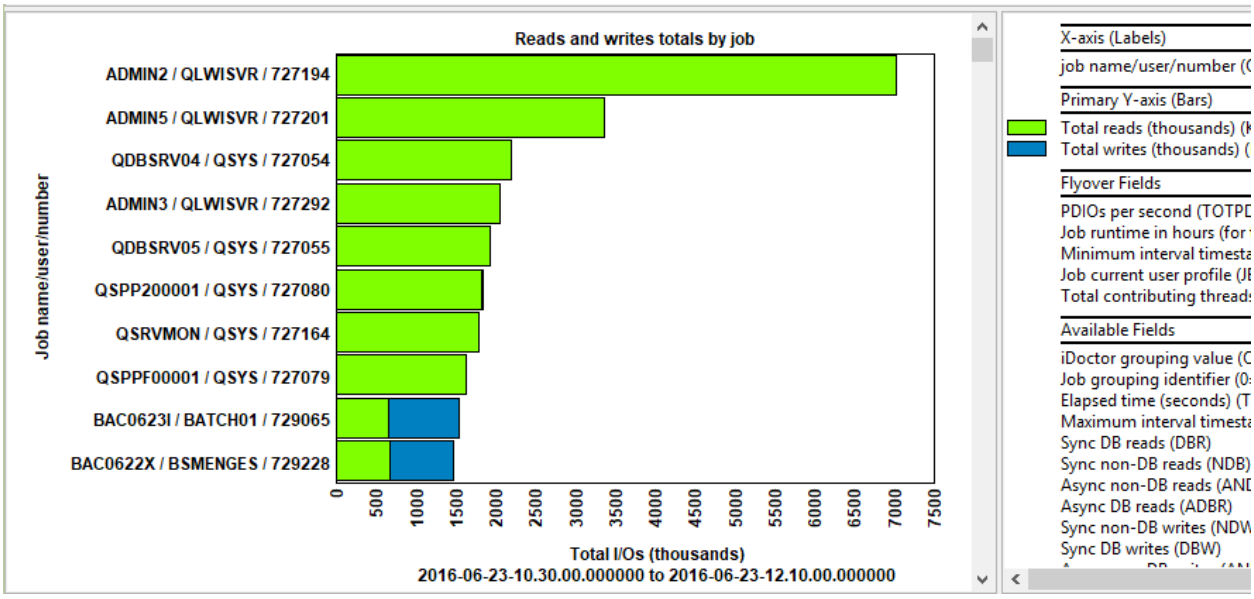
This graph shows the reads and writes totals ranked by thread or system task.



Reads and writes totals by thread

9.20.24.2 Reads and writes totals by job

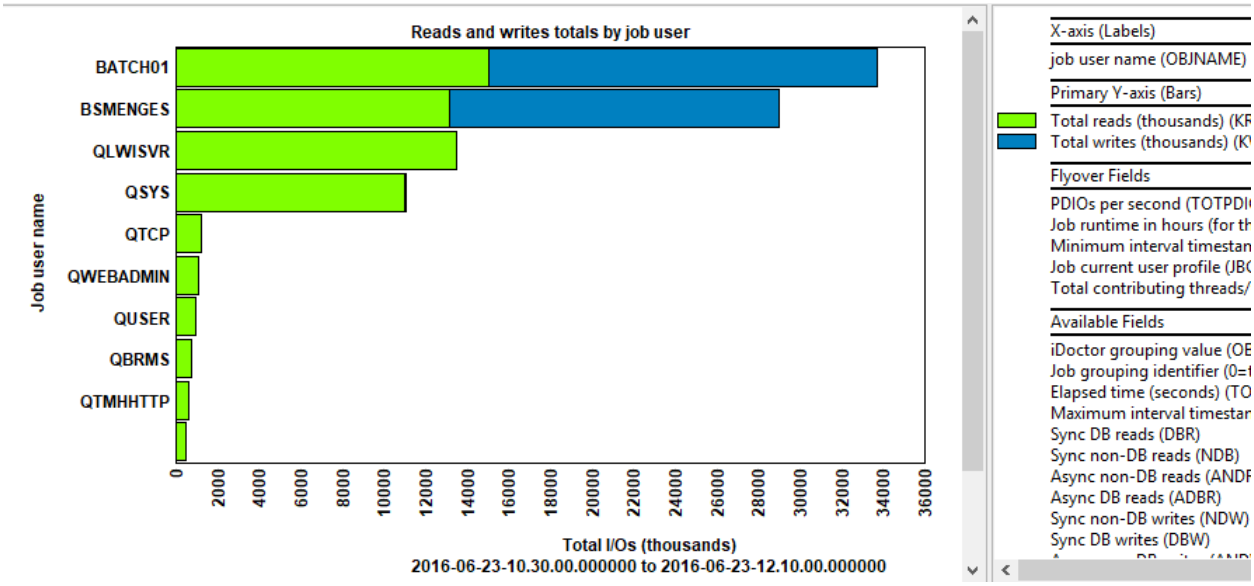
This graph shows the reads and writes totals ranked by job. This is all threads added together or system task.



Reads and writes totals by job

9.20.24.3 Reads and writes totals by job user

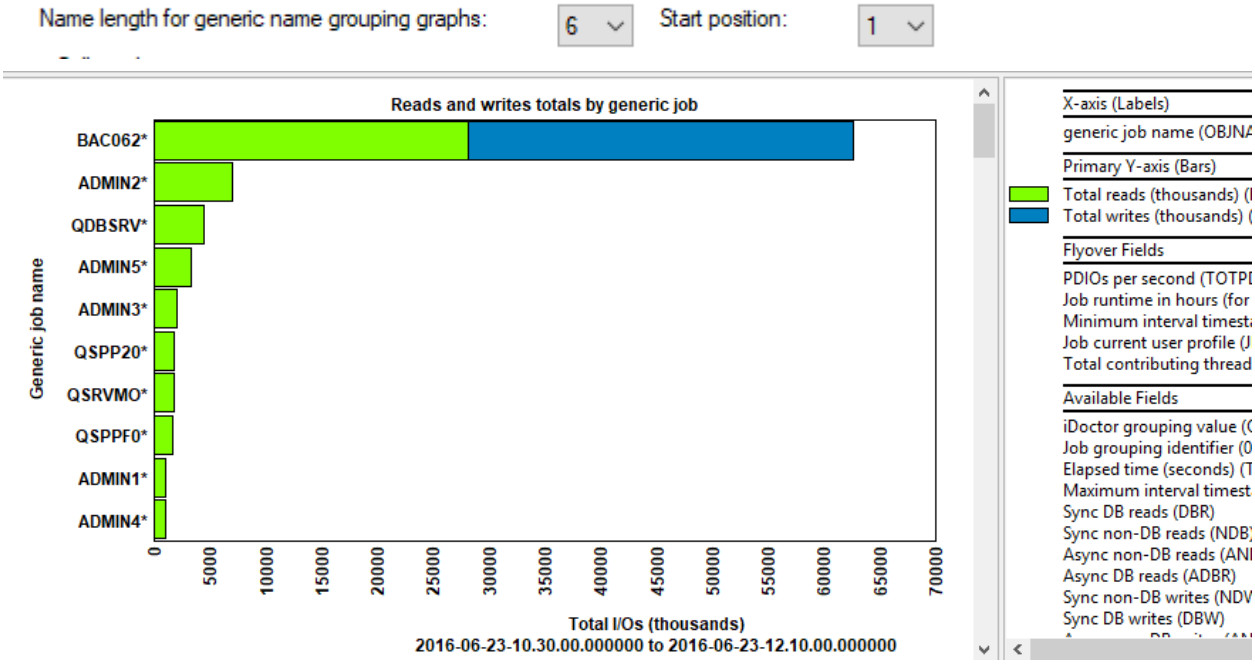
This graph shows the reads and writes totals ranked by job user. All jobs having the same job user name will be added together per bar.



Reads and writes totals by job user

9.20.24.4 Reads and writes totals by generic job

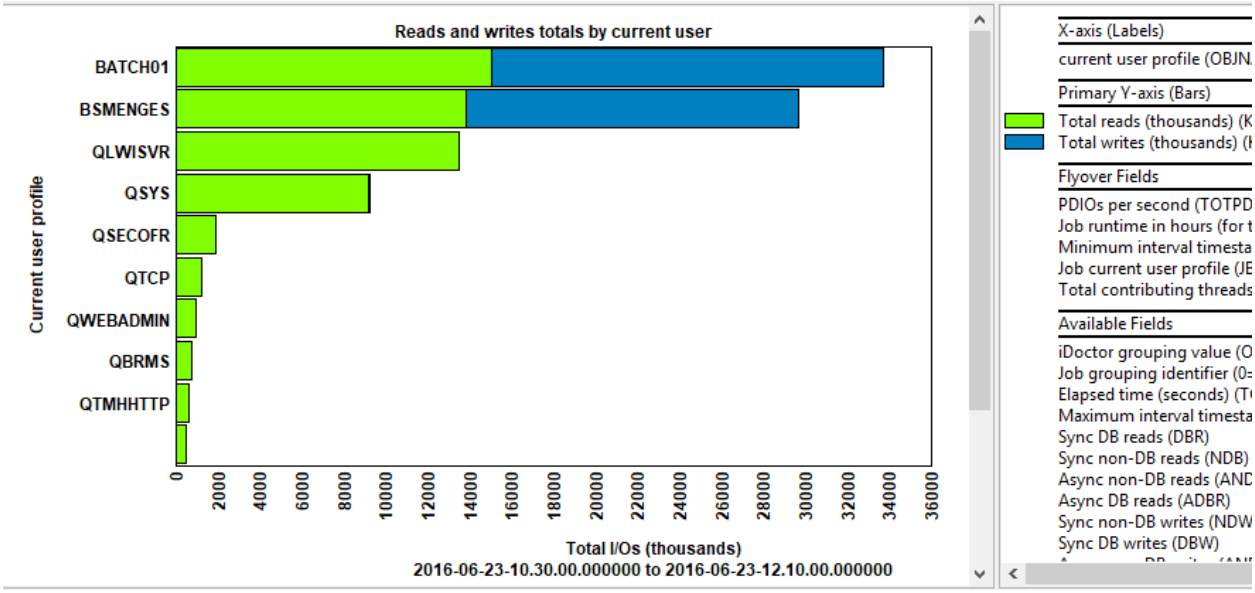
This graph shows the reads and writes totals ranked by generic job or system task name. The length (and start position) of the generic job name is controlled via the **Preferences -> Data Viewer** options shown below:



Reads and writes totals by generic job

9.20.24.5 Reads and writes totals by current user

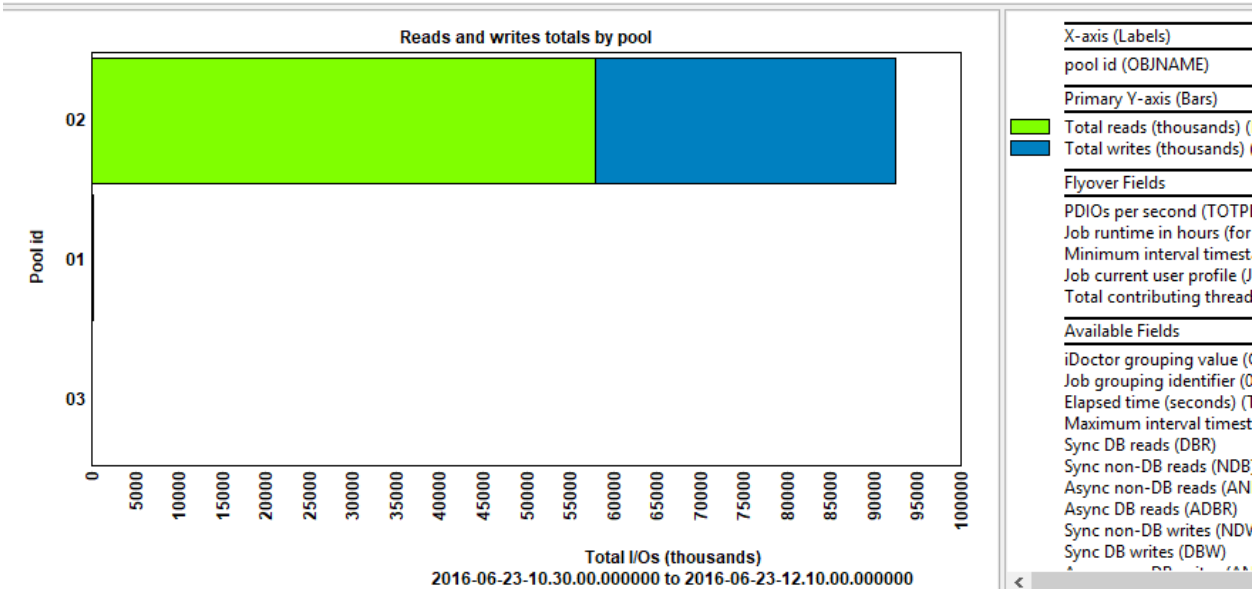
This graph shows the reads and writes totals ranked by current user profile. **Note:** All system tasks are grouped together into one "**System tasks"** record within this report.



Reads and writes totals by current user

9.20.24.6 Reads and writes totals by pool

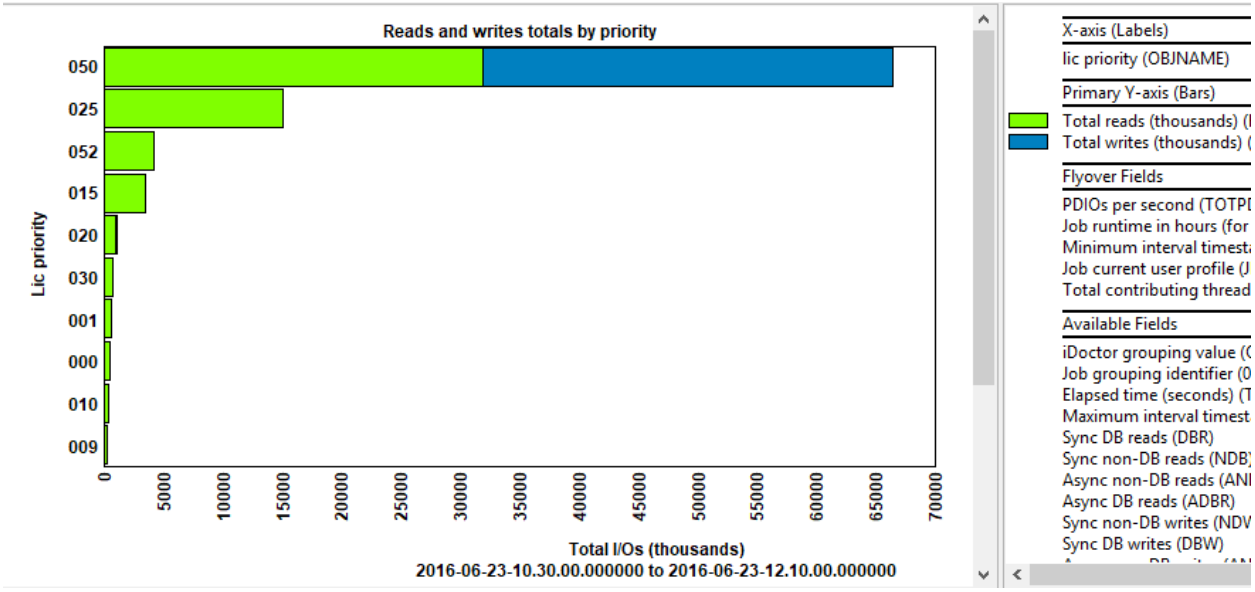
This graph shows the reads and writes totals ranked by memory pool.



Reads and writes totals by pool

9.20.24.7 Reads and writes totals by priority

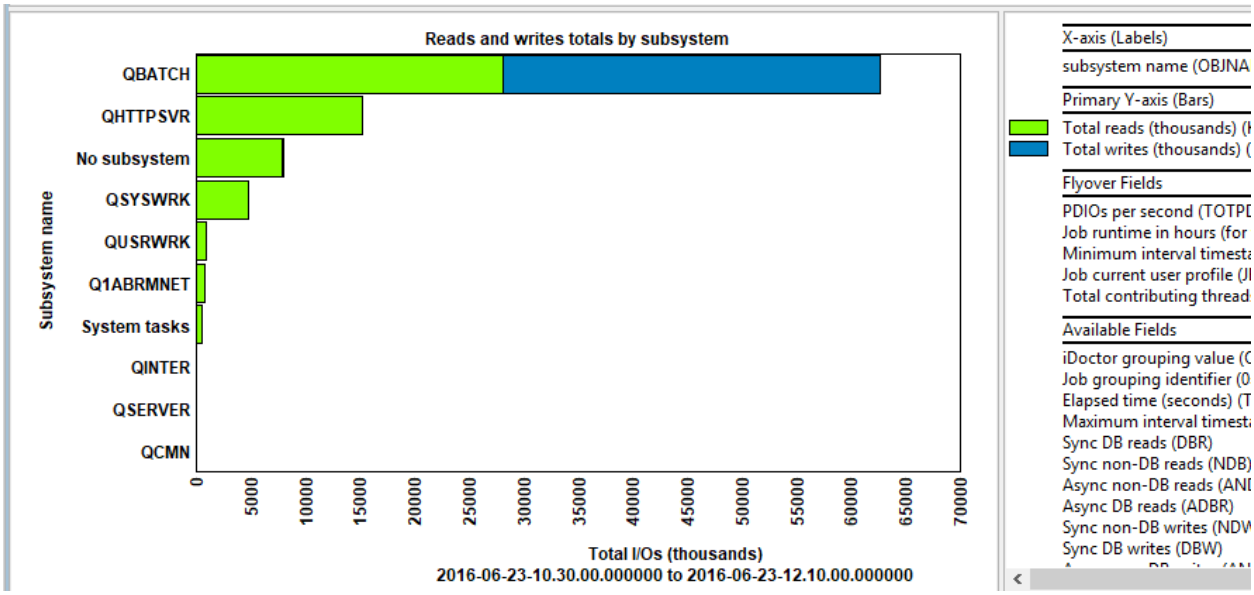
This graph shows the reads and writes totals ranked by LIC priority.



Reads and writes totals by priority

9.20.24.8 Reads and writes totals by subsystem

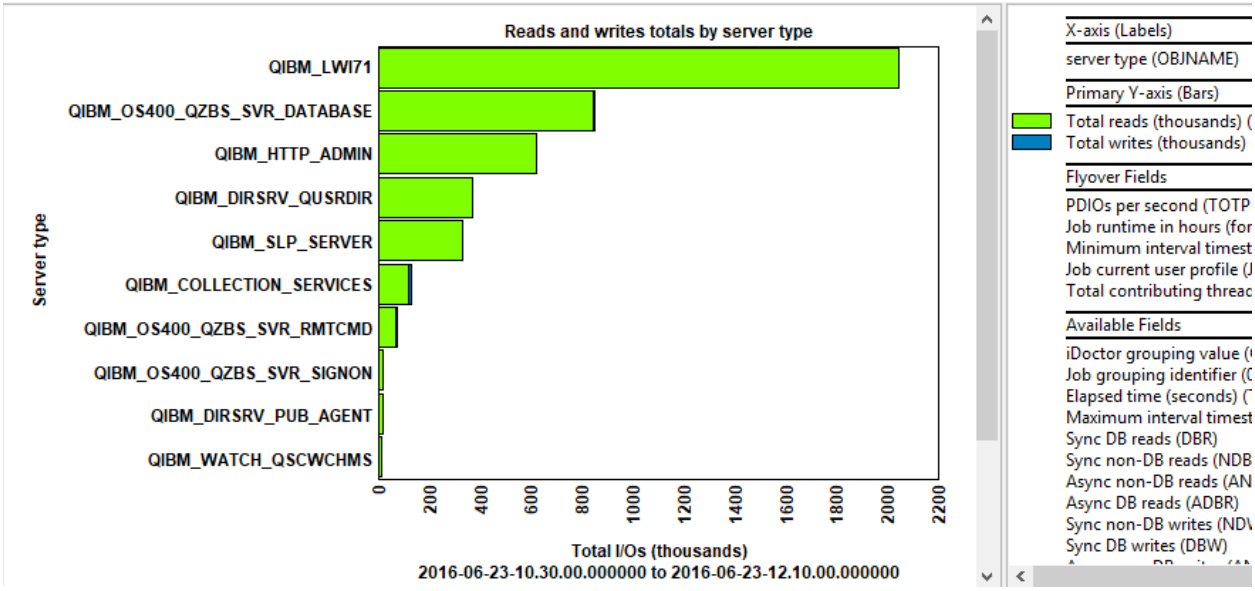
This graph shows the reads and writes totals ranked by subsystem. Typically the graph will contain 1 bar called “No subsystem” for work that did not run within a subsystem and another bar for all system tasks called “System tasks”.



Reads and writes totals by subsystem

9.20.24.9 Reads and writes totals by server type

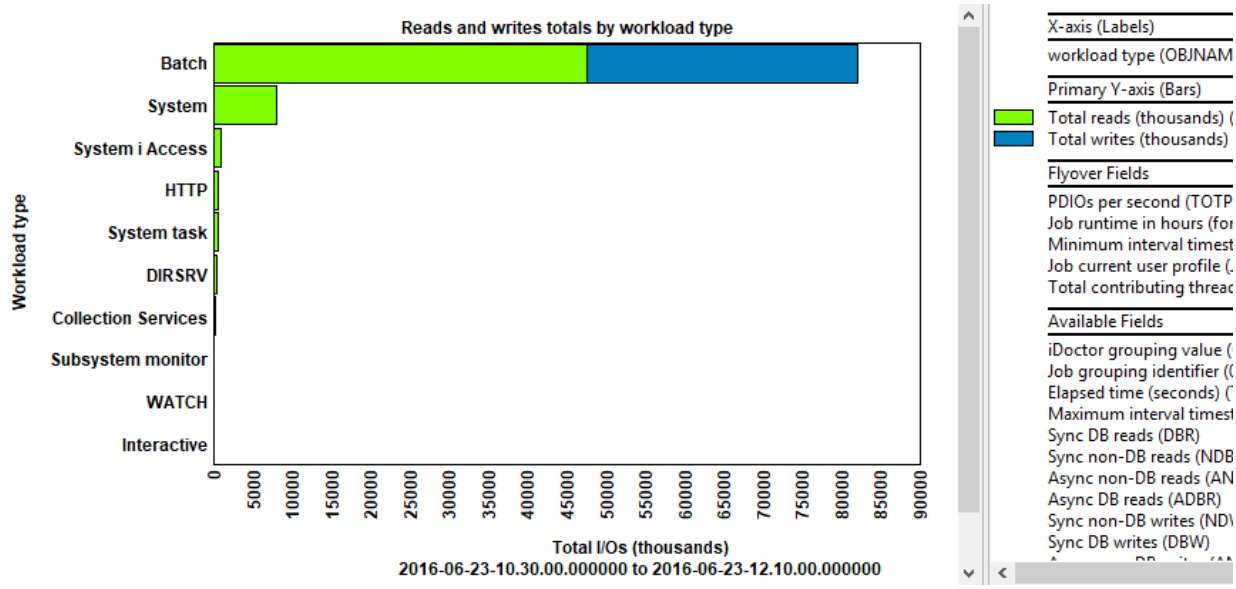
This graph shows the reads and writes totals ranked by server type.



Reads and writes totals by server type

9.20.24.10 Reads and writes totals by workload type

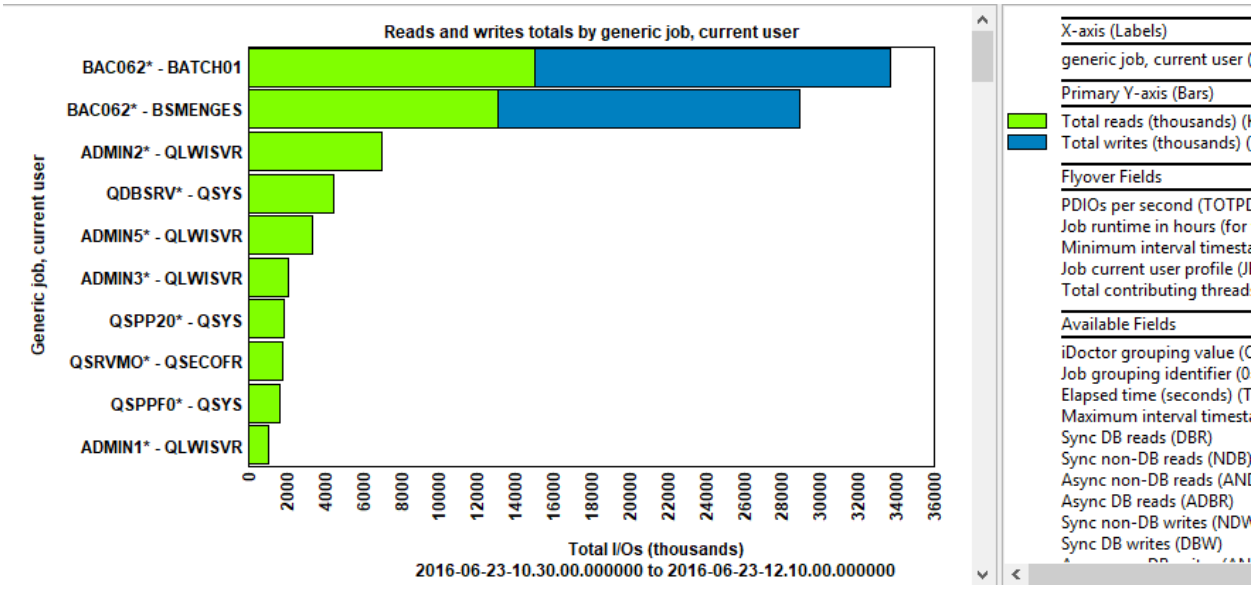
This graph shows the reads and writes totals ranked by workload type.



Reads and writes totals by workload type

9.20.24.11 Reads and writes totals by generic job, current user

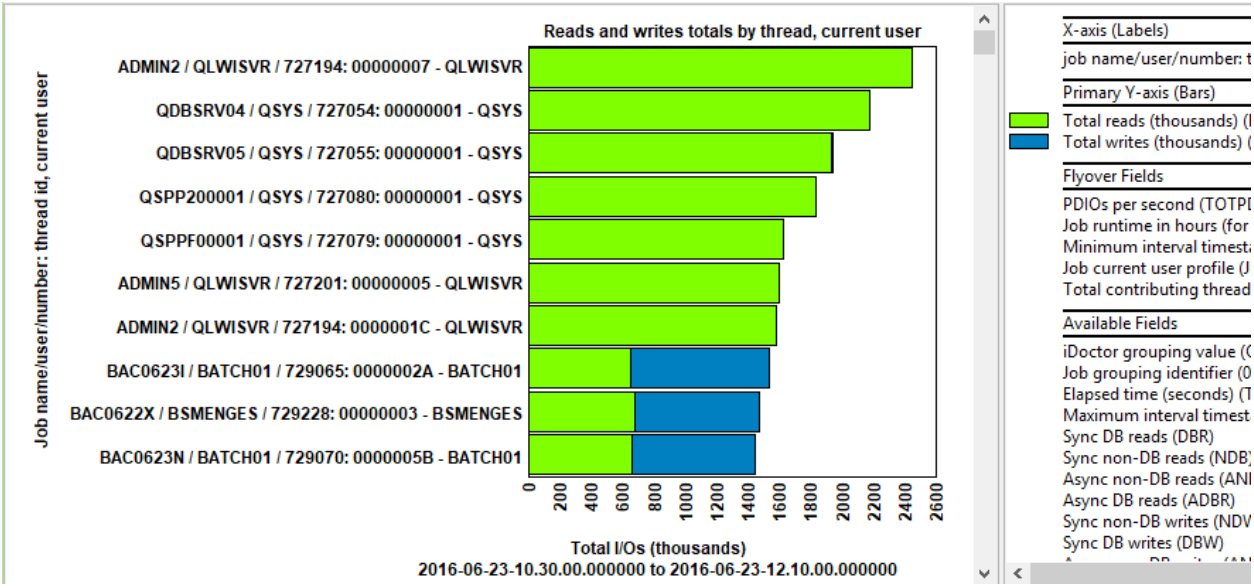
This graph shows the reads and writes totals ranked by generic job and current user combination.



Reads and writes totals by generic job, current user

9.20.24.12 Reads and writes totals by thread, current user

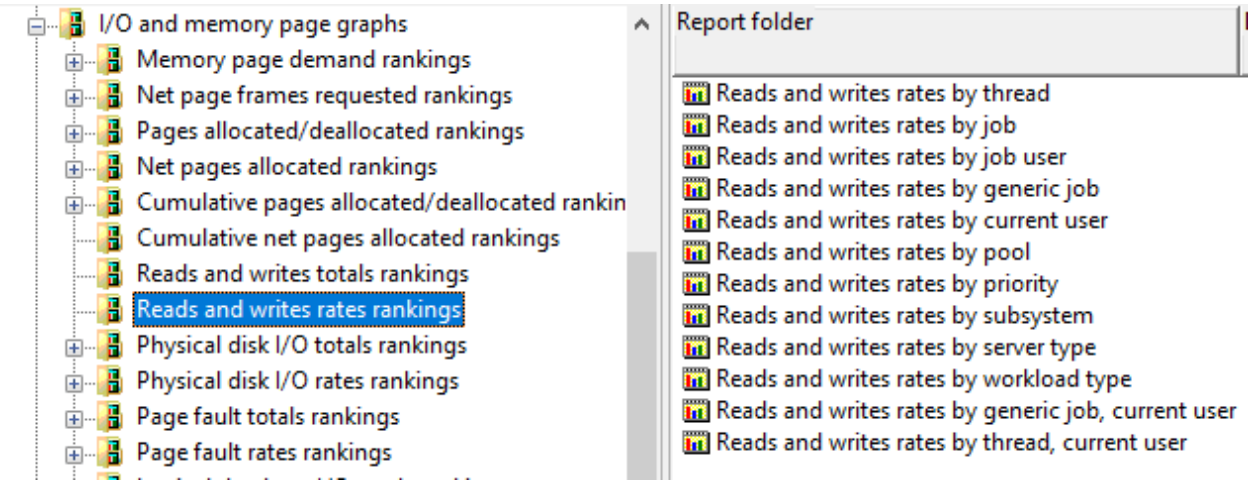
This graph shows the reads and writes totals ranked by thread/task and current user combination.



Reads and writes totals by thread, current user

9.20.25 Reads and writes rates rankings

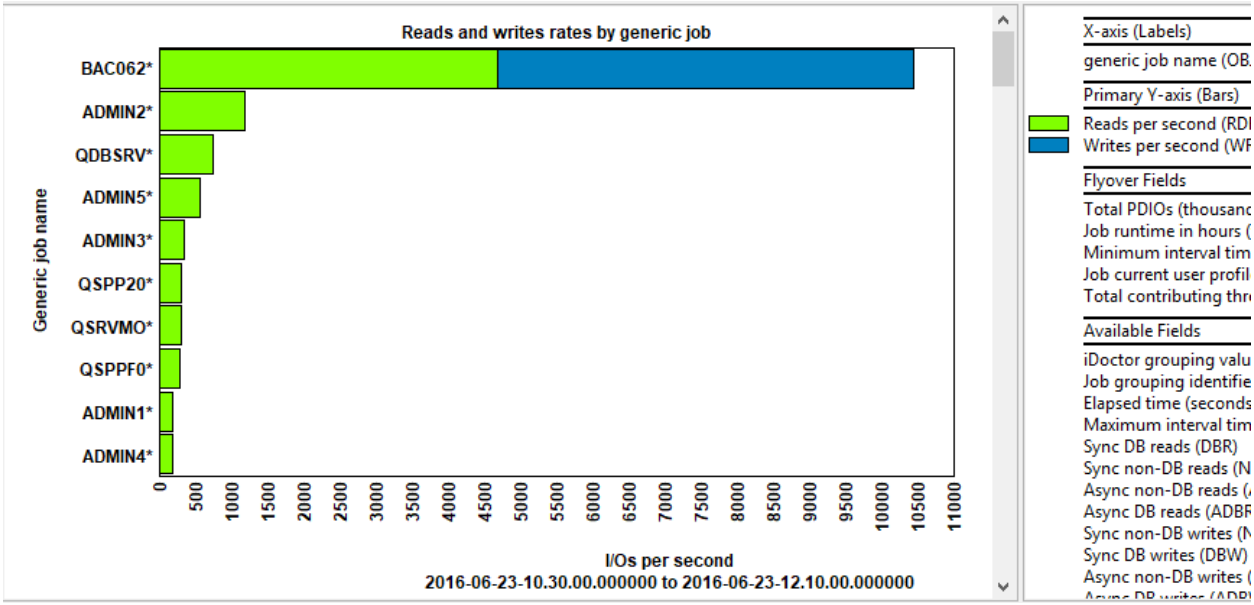
These rankings graphs displays the reads and writes rate per second ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Reads and writes totals rankings

Tip: These graphs show horizontal bar graphs which make the labels easier to read. You can use the Toggle graph format button  on the toolbar to flip these to vertical bar graphs if desired.

An example of this type of graph is:

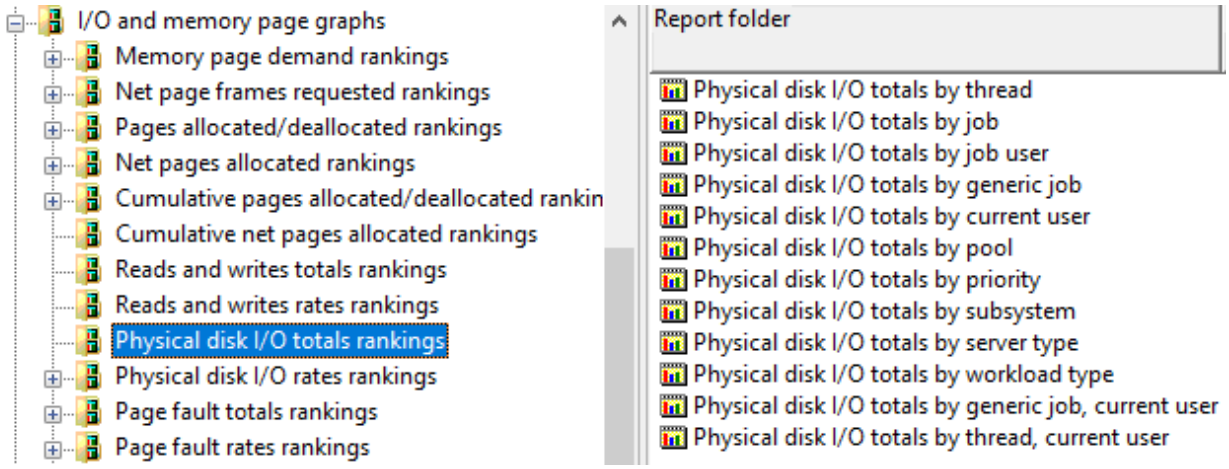


Reads and writes rates by generic job

For more examples, please see the Read and writes totals rankings section.

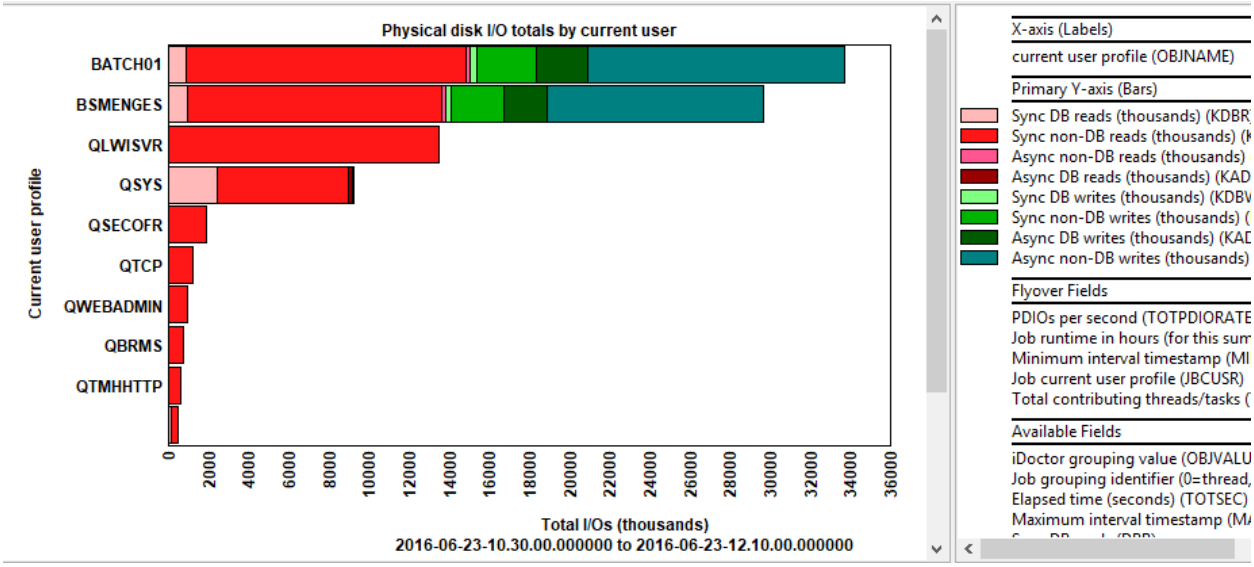
9.20.26 Physical disk I/O totals rankings

These rankings graphs displays the physical disk I/O totals ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Physical disk I/O totals rankings

An example of this type of graph is:

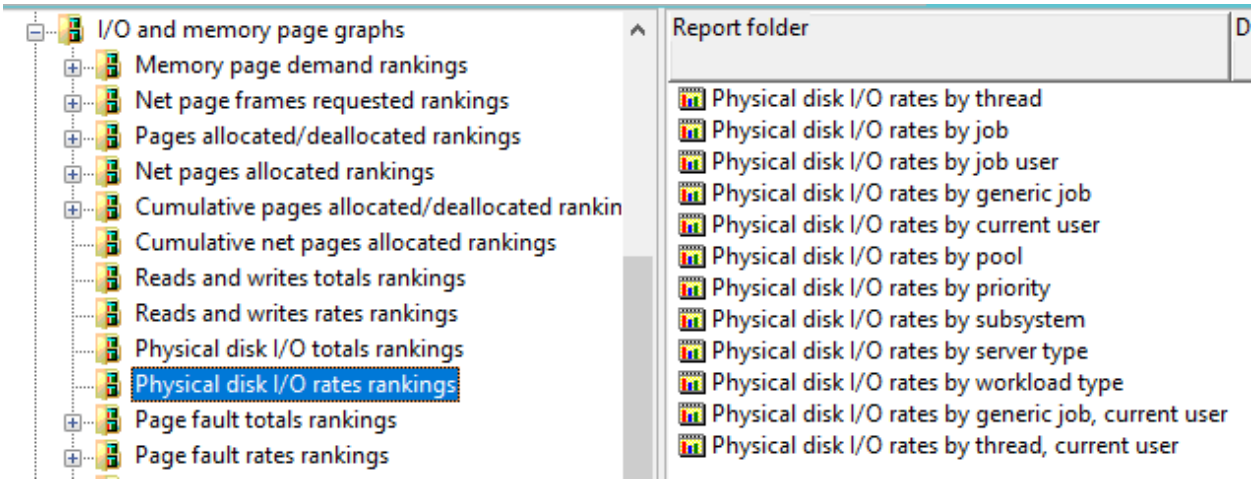


Physical disk I/O totals by current user

For more examples, please see the Read and writes totals rankings section.

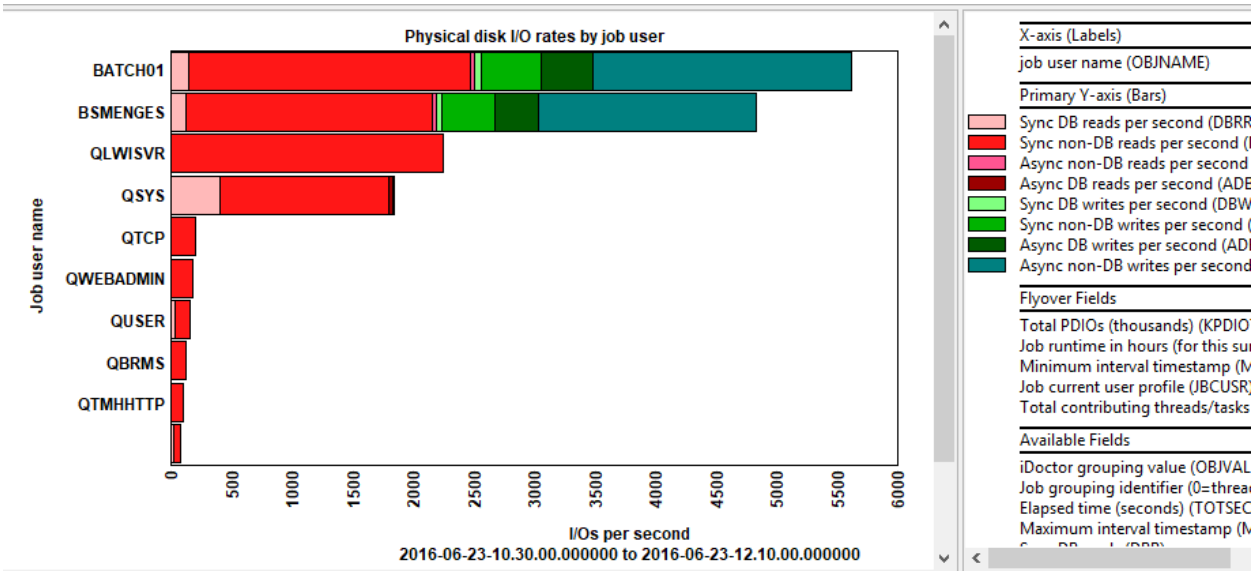
9.20.27 Physical disk I/O rates rankings

These rankings graphs displays the physical disk I/O rate per second ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Physical disk I/O rates rankings

An example of this type of graph is:

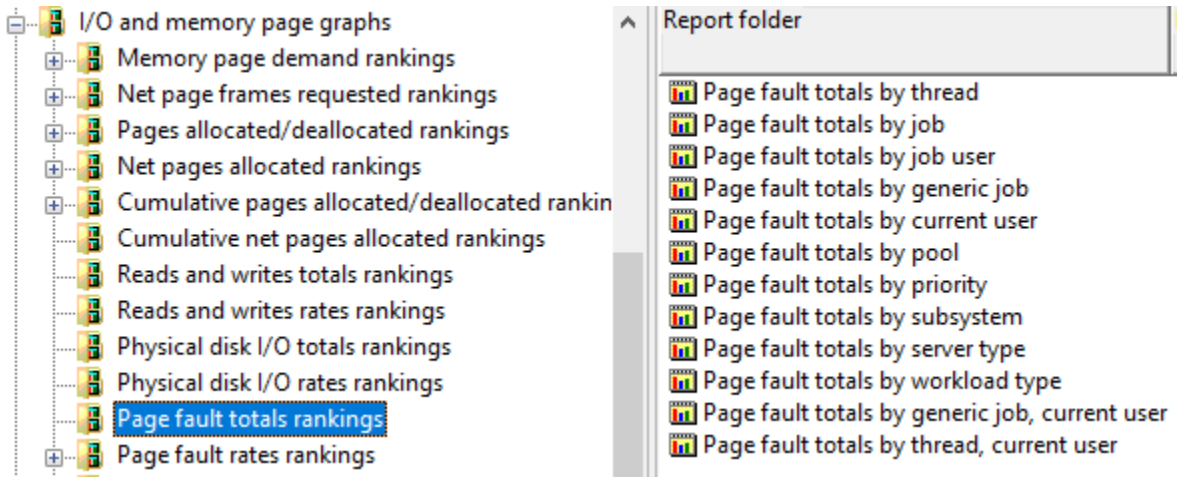


Physical disk I/O rates by job user

For more examples, please see the Read and writes totals rankings section.

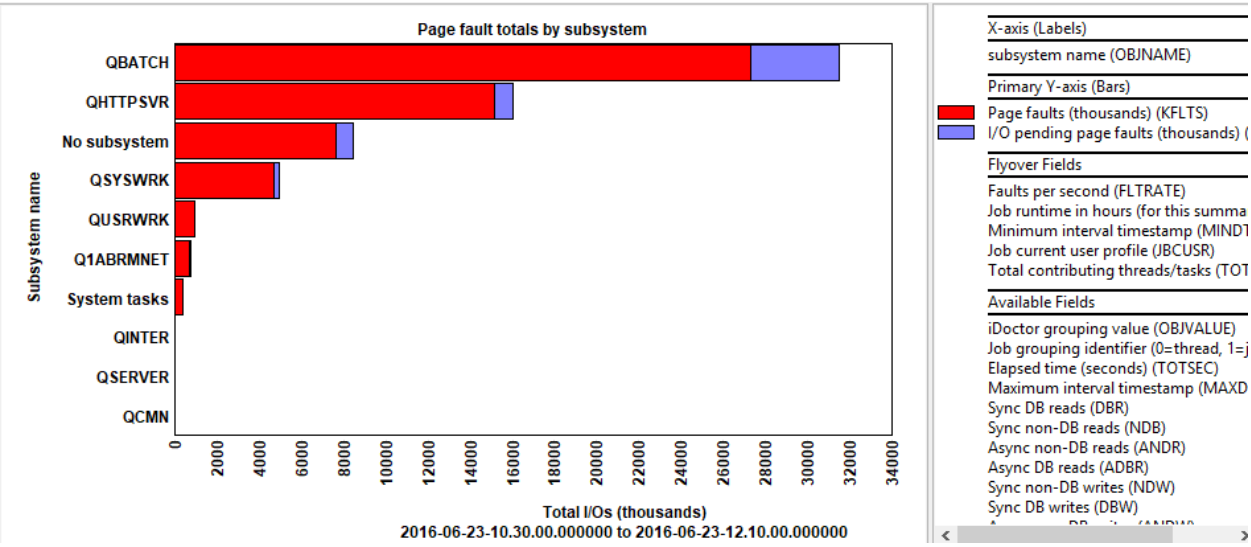
9.20.28 Page fault totals rankings

These rankings graphs displays the page fault totals. Both page faults and IO pending page fault are shown. These graphs are ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Page fault totals rankings

An example of this type of graph is:

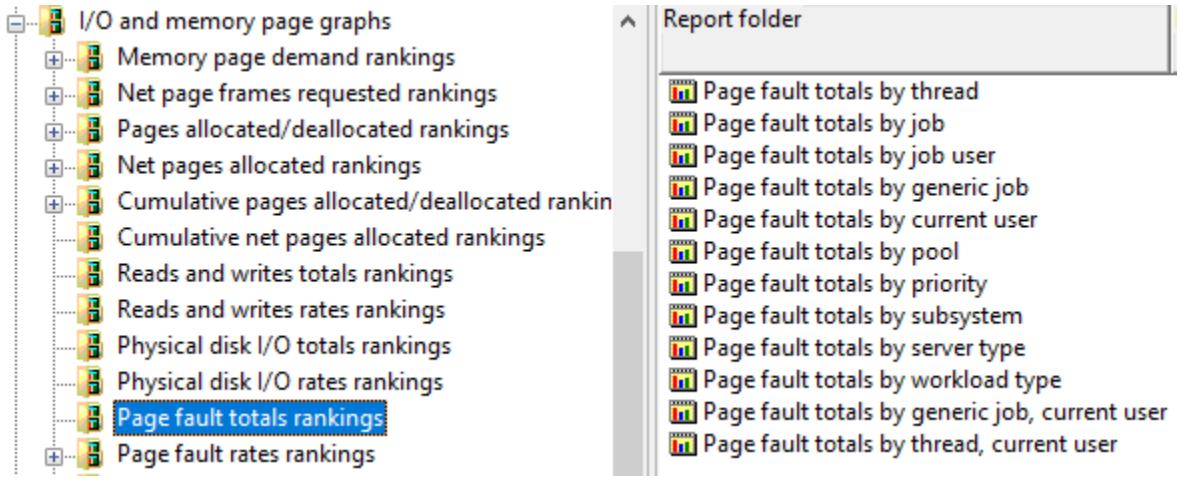


Page fault totals by subsystem

For more examples, please see the Read and writes totals rankings section.

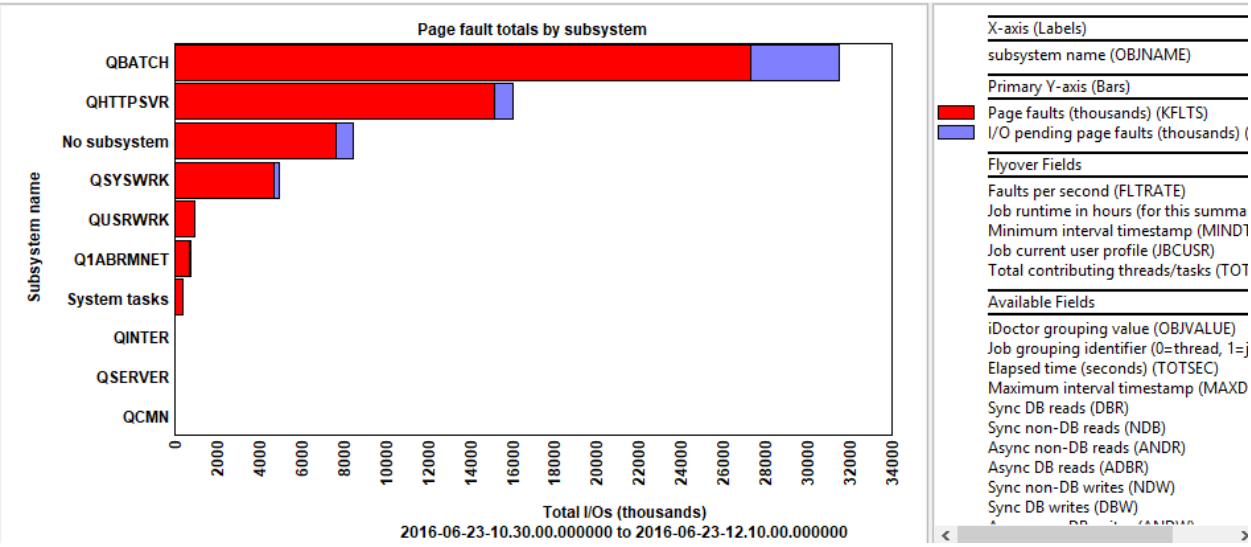
9.20.29 Page fault totals rankings

These rankings graphs displays the page fault totals. Both page faults and IO pending page fault are shown. These graphs are ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Page fault totals rankings

An example of this type of graph is:

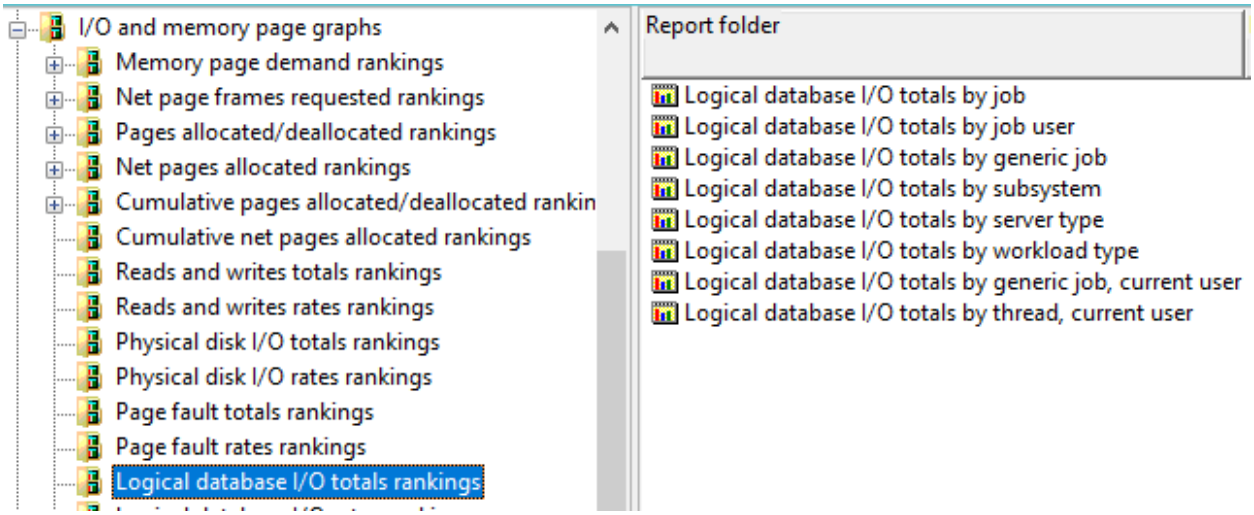


Page fault totals by subsystem

For more examples, please see the Read and writes totals rankings section.

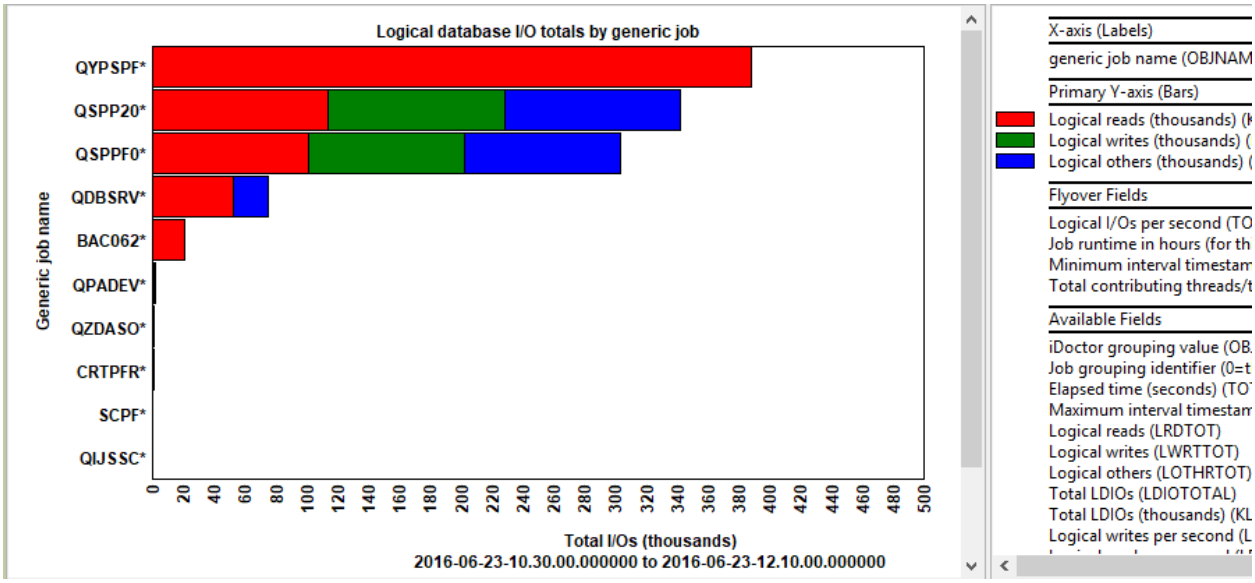
9.20.30 Logical database I/O totals rankings

These rankings graphs displays the logical database I/O totals. These graphs are ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Logical database I/O totals rankings

An example of this type of graph is:

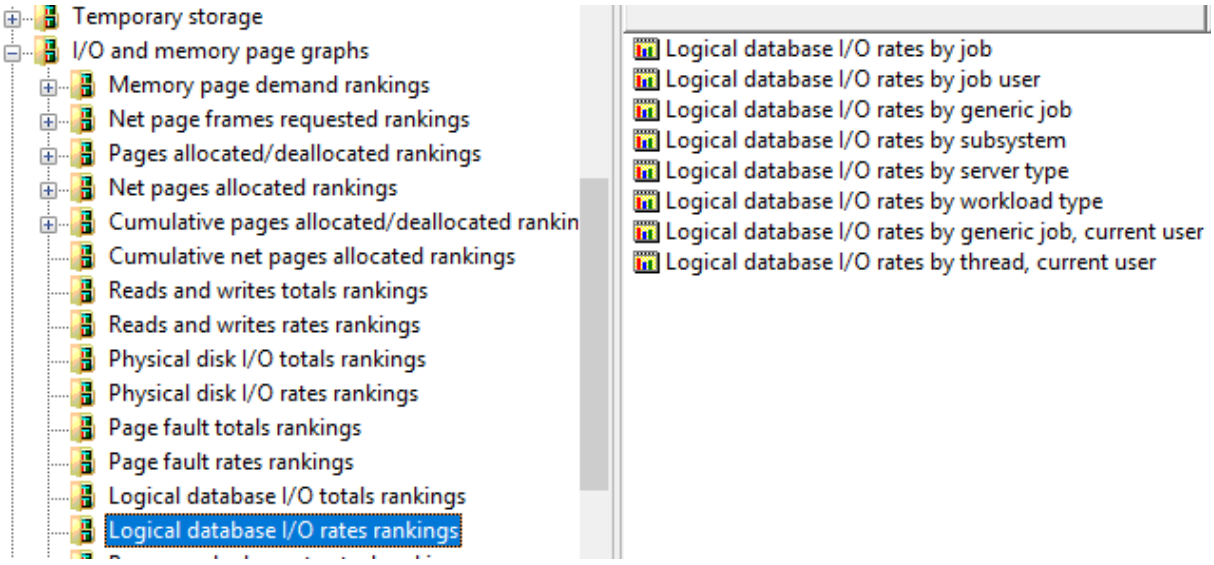


Logical database I/O totals by generic job

For more examples, please see the Read and writes totals rankings section.

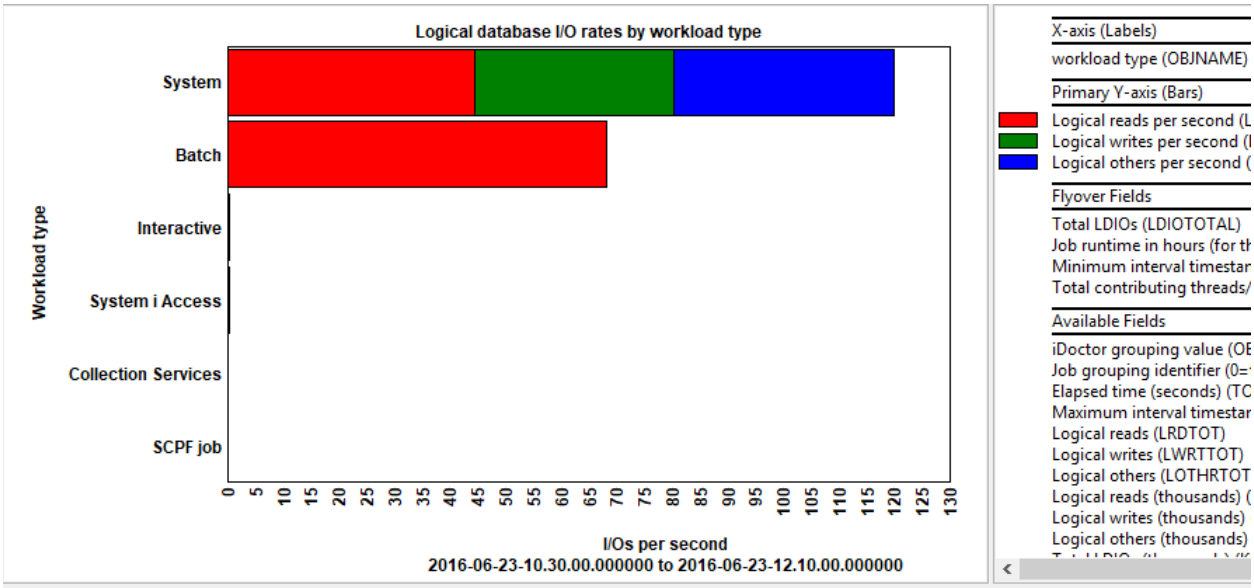
9.20.31 Logical database I/O rates rankings

These rankings graphs displays the logical database I/O rate per second. These graphs are ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Logical database I/O rates rankings

An example of this type of graph is:

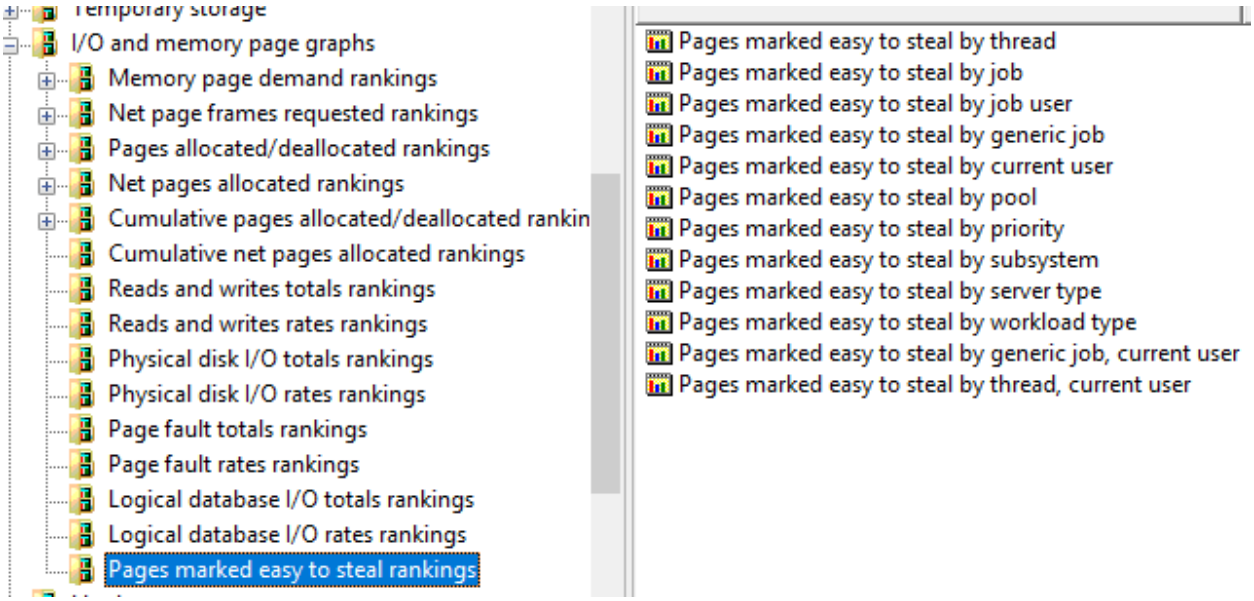


Logical database I/O rates by workload type

For more examples, please see the Read and writes totals rankings section.

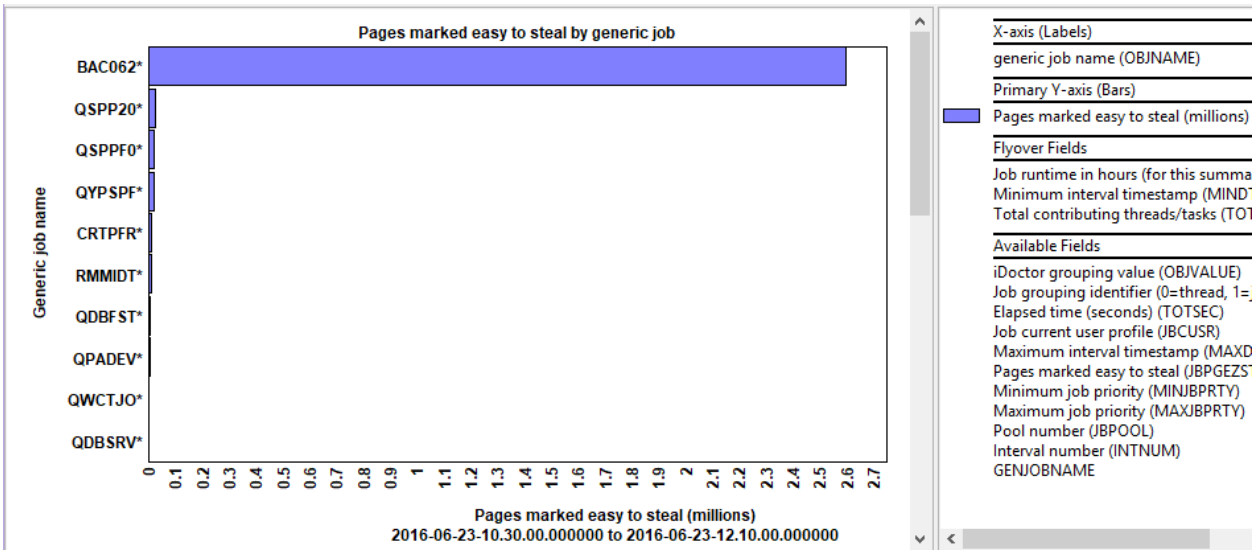
9.20.32 Pages marked easy to steal rankings

These rankings graphs display the pages marked easy to steal. These values are large and therefore shown in millions. These graphs are ranked by several types of job groupings as shown below:



I/O and memory page graphs -> Pages marked easy to steal rankings

An example of this type of graph is:



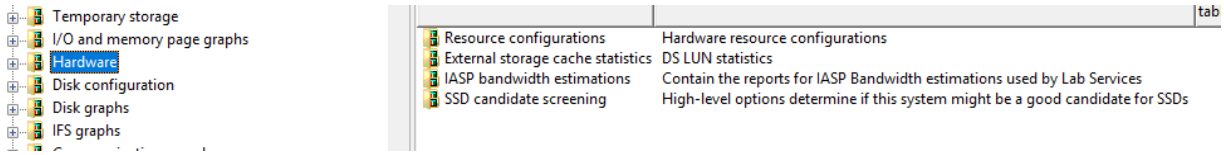
Pages marked easy to steal by generic job

For more examples, please see the Read and writes totals rankings section.

9.21 Hardware

This folder contains graphs and reports related to the system hardware such as disk configuration, resource configuration, Tape, SSDs, IASPs bandwidth estimations, external storage and more.

Note: The external storage graphs will only appear in this folder if you have external storage data AND the external storage analyses have been executed.

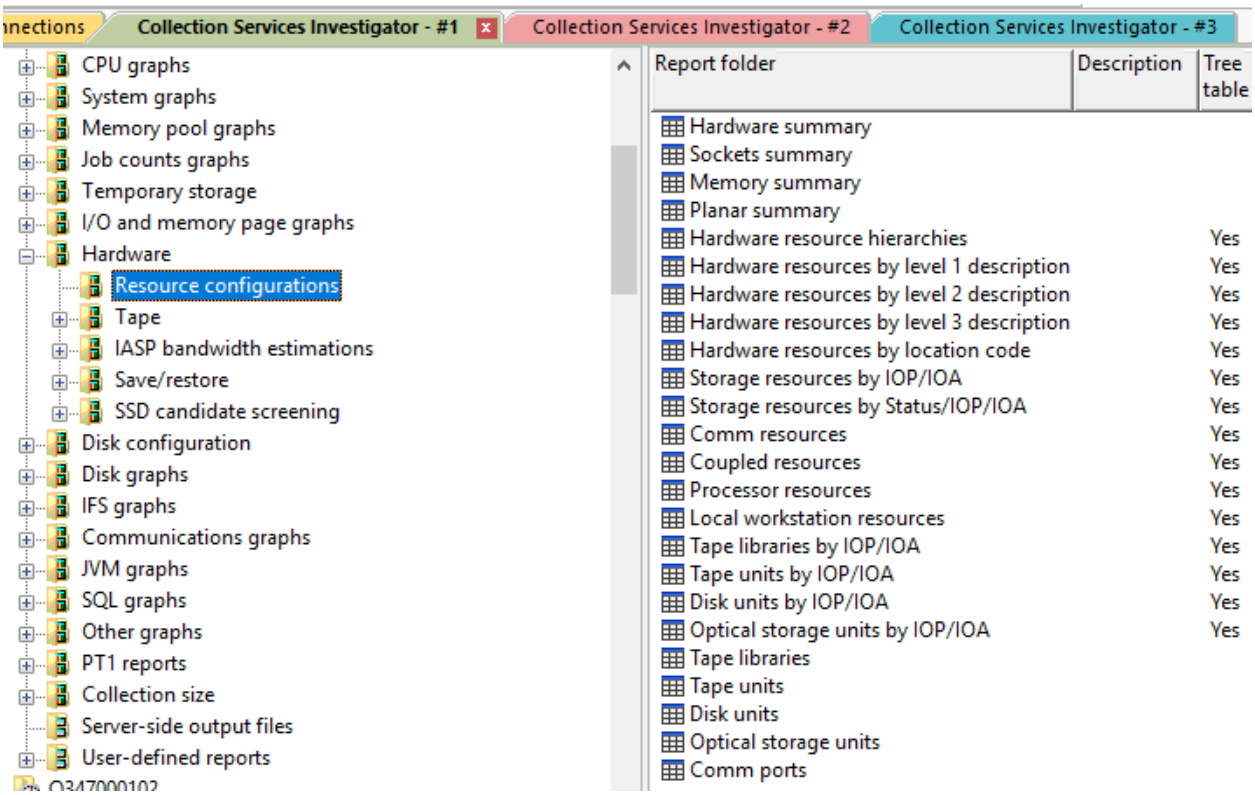


Hardware

9.21.1 Resource configurations

This folder contains several resource configuration reports some of which were provided by IBM Lab Services and are intended primarily for advanced users.

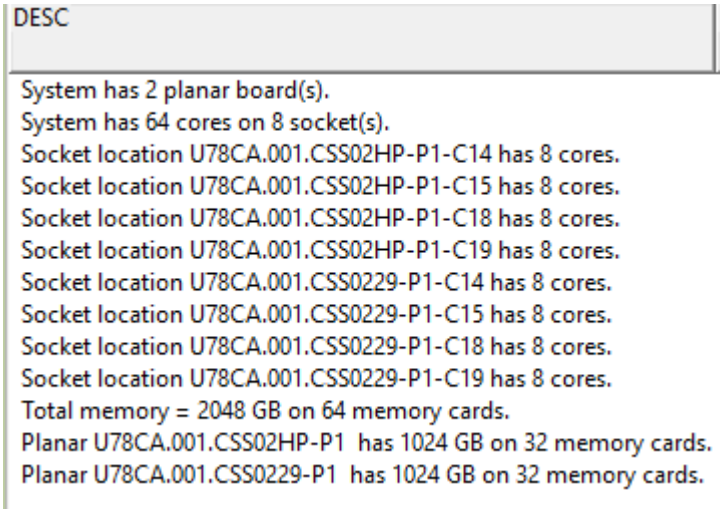
Note: These reports require file QAPMHDWR to have been included in the collection or you will have no data within them.



Hardware -> Resource configurations

9.21.1.1 Hardware summary

This report provides a high-level look at the planar boards, sockets and the amount of memory and CPU cores assigned to each.



Hardware summary

9.21.1.2 Sockets summary

This report is like the previous one but just contains information about the number of cores for each socket location. The memory data is not available/reliable.

/PMR00250AD/Q348000102/Sockets summary - #1

Location code (LOCCOD)	Cores
U78CA.001.CSS02HP-P1-C14	8
U78CA.001.CSS02HP-P1-C15	8
U78CA.001.CSS02HP-P1-C18	8
U78CA.001.CSS02HP-P1-C19	8
U78CA.001.CSS0229-P1-C14	8
U78CA.001.CSS0229-P1-C15	8
U78CA.001.CSS0229-P1-C18	8
U78CA.001.CSS0229-P1-C19	8

Sockets summary

9.21.1.3 Memory summary

This report lists the total memory and the total memory cards on the system and whether there are any mismatched memory DIMMs.

Memory (GBs) (TOTMEMGB)	Memory cards (CARDS)	Memory sizes are misn (1) or all the same (0) (MISMATCHDIMM)
2048	64	0

Memory summary

9.21.1.4 Planar summary

This report simply lists each planar board found and the total memory and memory cards for each one.

Planar location code (PLANAR)	Memory (GBs) (TOTMEMGB)	Memory cards (CARDS)
U78CA.001.CSS02HP-P1	1,024	32
U78CA.001.CSS0229-P1	1,024	32

Planar summary

9.21.1.5 Hardware resource hierarchies

This report is a tree report that allows you to expand the various hardware pieces of the system starting with the main card enclosure and includes the storage controllers and processors. This view is a bit like using the WRKHDWRSC command.

/PMR00250AD/Q348000102/Hardware resource hierarchies - #1

Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total									100% - 648
CEC01	9119-MME	Operational	Main card enclosure		10-3709C			U9119.MME.103709C	20.52% - 133
CMB07	577F-001	Operational	Storage controller		0A-0351658	0000000E9266	C4	U78CD.001.FZH3589-P1-C4	17.28% - 112
CMB11	577F-001	Operational	Storage controller		0A-0351478	0000000E9266	C4	U78CD.001.FZH3497-P1-C4	17.28% - 112
CMB01	577F-001	Operational	Storage controller		0A-0351694	0000000E9266	C4	U78CD.001.FZH3500-P1-C4	17.28% - 112
CMB10	577F-001	Operational	Storage controller		0A-0351475	0000000E9266	C4	U78CD.001.FZH3499-P1-C4	11.73% - 76
CMB09	577F-001	Operational	Storage controller		0A-0351474	0000000E9266	C1	U78CD.001.FZH3498-P2-C1	11.73% - 76
CMB06	5774-001	Operational	Storage controller		3C-8923311	0000010N7255	C2	U78CD.001.FZH3499-P2-C2	1.39% - 9
DC04	5774-001	Operational	Storage controller		3C-8923311	0000010N7255	C2	U78CD.001.FZH3499-P2-C2	1.39% - 9
CMB04	268C-001	Operational	Comm processor		00-00000			U9119.MME.103709C-V18	.46% - 3
CMB17	5767-001	Operational	Comm processor		02-E77A10E	0000046K6601	C6	U78CD.001.FZH3589-P1-C6	.31% - 2
CMB05	5774-001	Operational	Storage controller		3C-89340C3	0000010N7255	C3	U78CD.001.FZH3801-P1-C3	.15% - 1
CMB03	6B03-001	Operational	Comm processor		00-00000		C1	U9119.MME.103709C-V18-C1	.15% - 1
CMB02	6B03-001	Operational	Comm processor		00-00000		C0	U9119.MME.103709C-V18-C0	.15% - 1
CMB12	576E-001	Operational	Comm processor		02-43B8FA6	0000010N9034	C4	U78CD.001.FZH3499-P2-C4	.15% - 1

Hardware resource hierarchies

9.21.1.6 Hardware resource hierarchies by level 1 descriptions

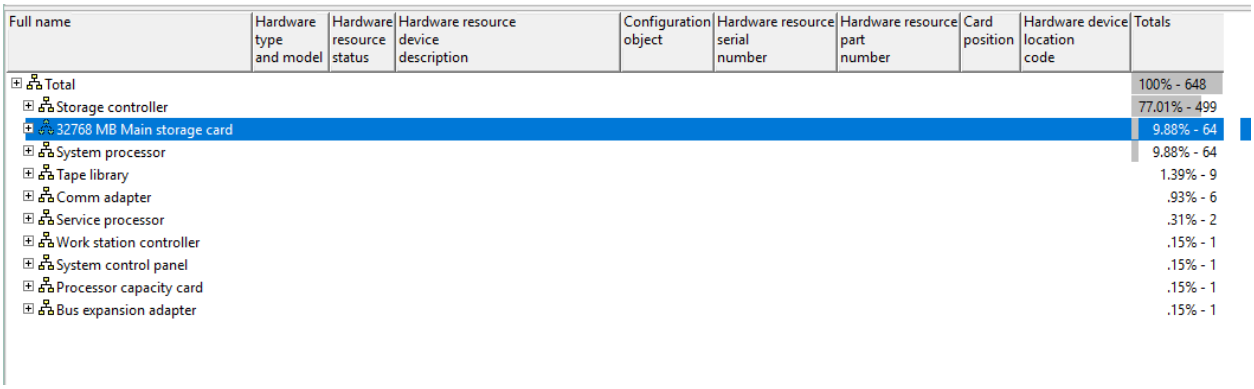
This report is another way of looking at the data in the hardware resource file and groups up and counts the number of resources by level 1 description. This includes storage controllers, main card enclosures and comm processor resources typically.

Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total									100% - 651
Storage controller									77.88% - 507
CMB11	577F-001	Operat>	Storage controll>		0A-0351478	0000000E9266	C4	U78CD.001.FZ>	17.20% - 112
CMB07	577F-001	Operat>	Storage controll>		0A-0351658	0000000E9266	C4	U78CD.001.FZ>	17.20% - 112
CMB01	577F-001	Operat>	Storage controll>		0A-0351694	0000000E9266	C4	U78CD.001.FZ>	17.20% - 112
CMB10	577F-001	Operat>	Storage controll>		0A-0351475	0000000E9266	C4	U78CD.001.FZ>	11.67% - 76
CMB09	577F-001	Operat>	Storage controll>		0A-0351474	0000000E9266	C1	U78CD.001.FZ>	11.67% - 76
DC04	5774-001	Operat>	Storage controll>		3C-8923311	0000010N7255	C2	U78CD.001.FZ>	1.38% - 9
CMB06	5774-001	Operat>	Storage controll>		3C-8923311	0000010N7255	C2	U78CD.001.FZ>	1.38% - 9
CMB05	5774-001	Operat>	Storage controll>		3C-89340C3	0000010N7255	C3	U78CD.001.FZ>	.15% - 1
Main card enclosure									20.43% - 133
Comm processor									1.23% - 8
Combined function IO processor									.46% - 3

Hardware resource hierarchies by level 1 descriptions

9.21.1.7 Hardware resource hierarchies by level 2 descriptions

This report is another way of looking at the data in the hardware resource file and groups up and counts the number of resources by level 2 description. This includes storage controllers, main storage cards, system processors and more.

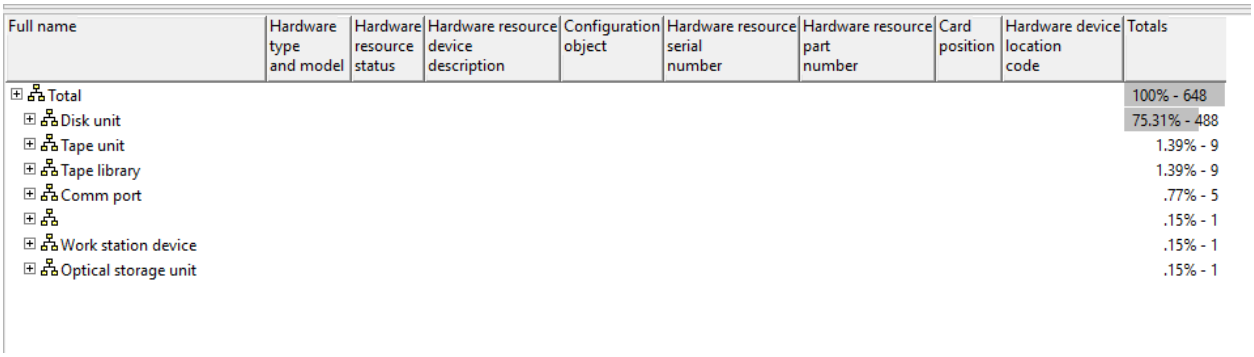


Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total									100% - 648
Storage controller									77.01% - 499
32768 MB Main storage card									9.88% - 64
System processor									9.88% - 64
Tape library									1.39% - 9
Comm adapter									.93% - 6
Service processor									.31% - 2
Work station controller									.15% - 1
System control panel									.15% - 1
Processor capacity card									.15% - 1
Bus expansion adapter									.15% - 1

Hardware resource hierarchies by level 2 descriptions

9.21.1.8 Hardware resource hierarchies by level 3 descriptions

This report is another way of looking at the data in the hardware resource file and groups up and counts the number of resources by level 3 description. This includes things like disk units and tape units.

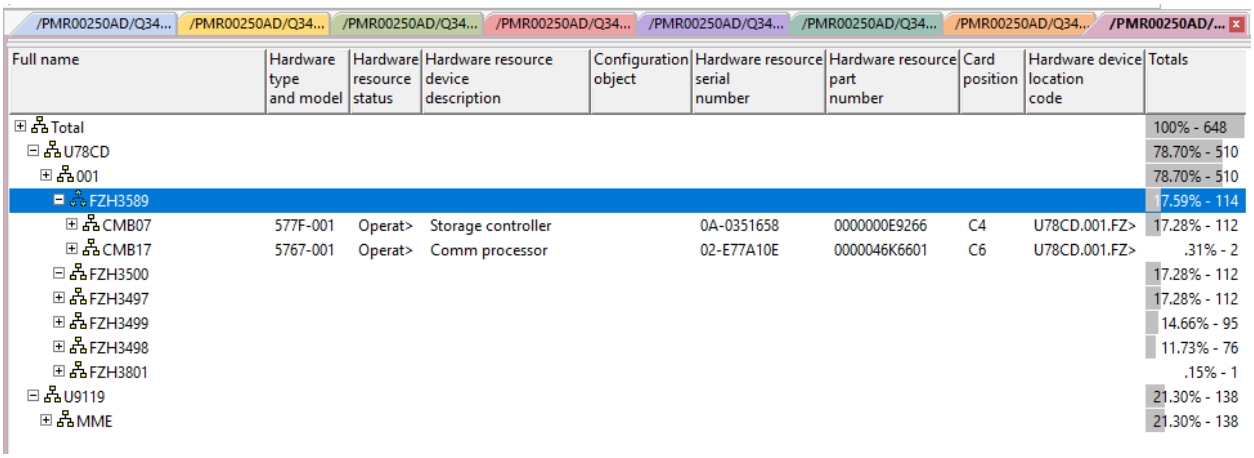


Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total									100% - 648
Disk unit									75.31% - 488
Tape unit									1.39% - 9
Tape library									1.39% - 9
Comm port									.77% - 5
									.15% - 1
Work station device									.15% - 1
Optical storage unit									.15% - 1

Hardware resource hierarchies by level 3 descriptions

9.21.1.9 Hardware resources by location code

This report is another way of looking at the data in the hardware resource file and groups up and counts the data by the various sections of the resource location code.



/PMR00250AD/Q34... /PMR00250AD/Q34... /PMR00250AD/Q34... /PMR00250AD/Q34... /PMR00250AD/Q34... /PMR00250AD/Q34... /PMR00250AD/Q34... /PMR00250AD/...

Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total									100% - 648
U78CD									78.70% - 510
001									78.70% - 510
FZH3589									17.59% - 114
CMB07	577F-001	Operat>	Storage controller		0A-0351658	0000000E9266	C4	U78CD.001.FZ>	17.28% - 112
CMB17	5767-001	Operat>	Comm processor		02-E77A10E	0000046K6601	C6	U78CD.001.FZ>	.31% - 2
FZH3500									17.28% - 112
FZH3497									17.28% - 112
FZH3499									14.66% - 95
FZH3498									11.73% - 76
FZH3801									.15% - 1
U9119									21.30% - 138
MME									21.30% - 138

Hardware resources by location code

9.21.1.10 Storage resources by IOP/IOA

This report displays the IOPs and IOAs and the disk units found within them.

Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total									100% - 491
CMB11	577F-001	Operational	Storage controller		0A-0351478	0000000E9266	C4	U78CD.001.FZH3497-P1-C4	22.81% - 112
CMB07	577F-001	Operational	Storage controller		0A-0351658	0000000E9266	C4	U78CD.001.FZH3589-P1-C4	22.81% - 112
CMB01	577F-001	Operational	Storage controller		0A-0351694	0000000E9266	C4	U78CD.001.FZH3500-P1-C4	22.81% - 112
CMB10	577F-001	Operational	Storage controller		0A-0351475	0000000E9266	C4	U78CD.001.FZH3499-P1-C4	15.48% - 76
CMB09	577F-001	Operational	Storage controller		0A-0351474	0000000E9266	C1	U78CD.001.FZH3498-P2-C1	15.48% - 76
CMB06	5774-001	Operational	Storage controller		3C-8923311	0000010N7255	C2	U78CD.001.FZH3499-P2-C2	.41% - 2
CMB04	268C-001	Operational	Comm processor		00-00000			U9119.MME.103709C-V18	.20% - 1
CMB05	5774-001	Operational	Storage controller		3C-89340C3	0000010N7255	C3	U78CD.001.FZH3801-P1-C3	0% - 0

Storage resources by IOP/IOA

9.21.1.11 Storage resources by Status/IOP/IOA

This report is like the previous one except it includes the hardware status as the 1st level of the report.

Full name	Hardware type and model	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware resource part number	Card position	Hardware device location code	Totals
Total								100% - 491
Operational								100% - 491
								0% - 0

Storage resources by Status/IOP/IOA

9.21.1.12 Comm resources

Full name	Hardware type and model	Hardware resource status	Hardware resource device description	Configuration object	Hardware resource serial number	Hardware device location code
Total						
CMB02	6B03-001	Operational	Comm processor		00-00000	U9119.MME.103709C-V18-C0
CMB03	6B03-001	Operational	Comm processor		00-00000	U9119.MME.103709C-V18-C1
CMB04	268C-001	Operational	Comm processor		00-00000	U9119.MME.103709C-V18
CMB12	576E-001	Operational	Comm processor		02-43B8FA6	U78CD.001.FZH3499-P2-C4
CMB17	5767-001	Operational	Comm processor		02-E77A10E	U78CD.001.FZH3589-P1-C6

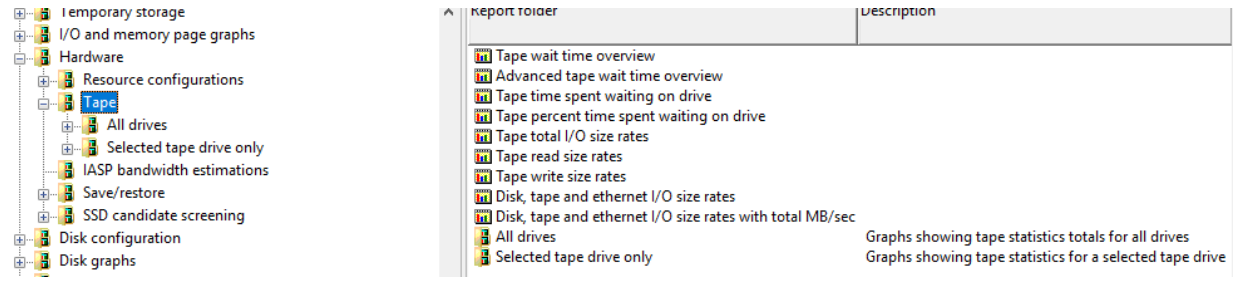
9.21.2 Tape

These graphs show statistics for the tape devices connected to the system and working. 3 types of graphs are available:

Overview: These show each tape device together on the same graph over time.

All drives: These graphs summarize all the tape device metrics together over time.

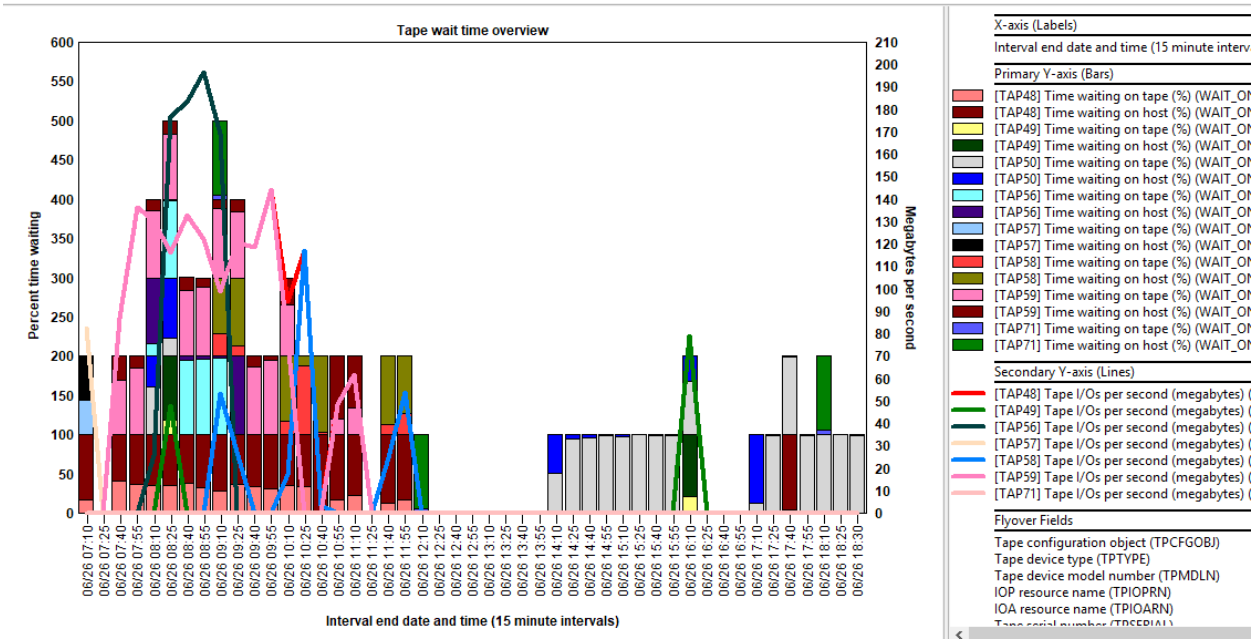
Selected tape drive only: When using these graph you will be prompted for the desired tape to graph over time.



Hardware -> Tape

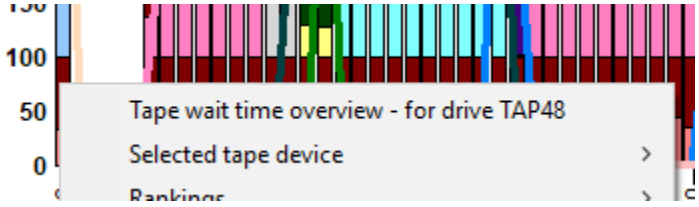
9.21.2.1 Tape wait time overview

This graph displays each tape device used over time during the collection. The primary Y-axis displays time waiting on the tape device and time waiting on the host. The secondary Y-axis displays the amount data transferred per second in megabytes.

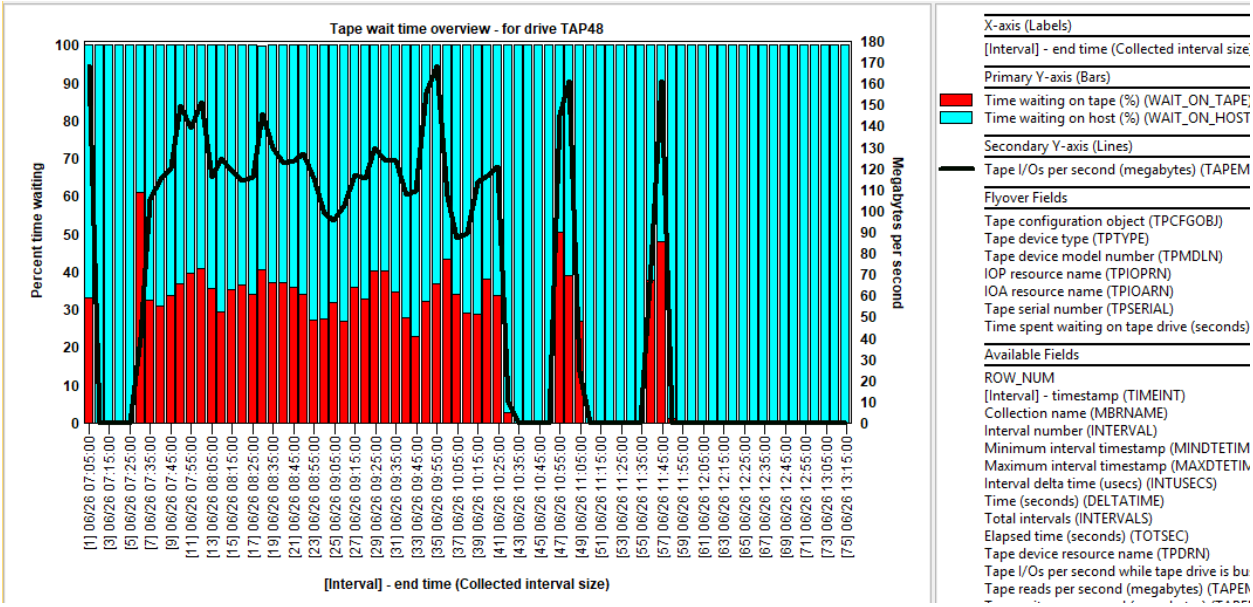


Tape wait time overview

Tip: If you wish to graph a single tape as a drill down, simply right-click the graph and pick the 1st option.



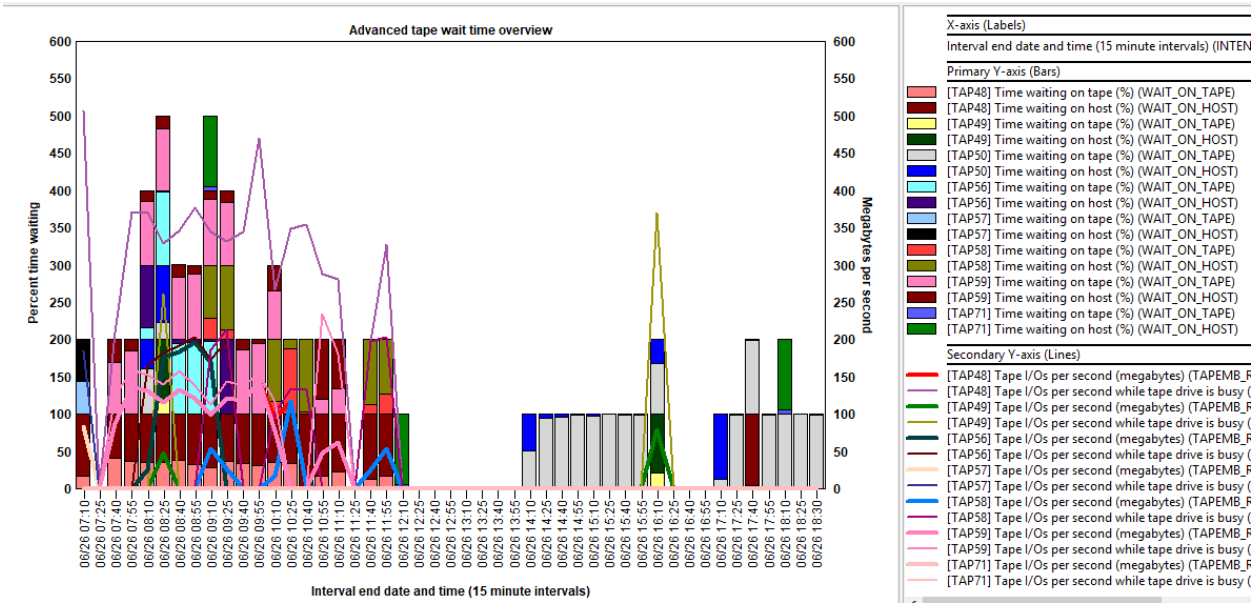
Tape wait time overview – for drive X drill down option



Tape wait time overview – for drive TAP48

9.21.2.2 Advanced tape wait time overview

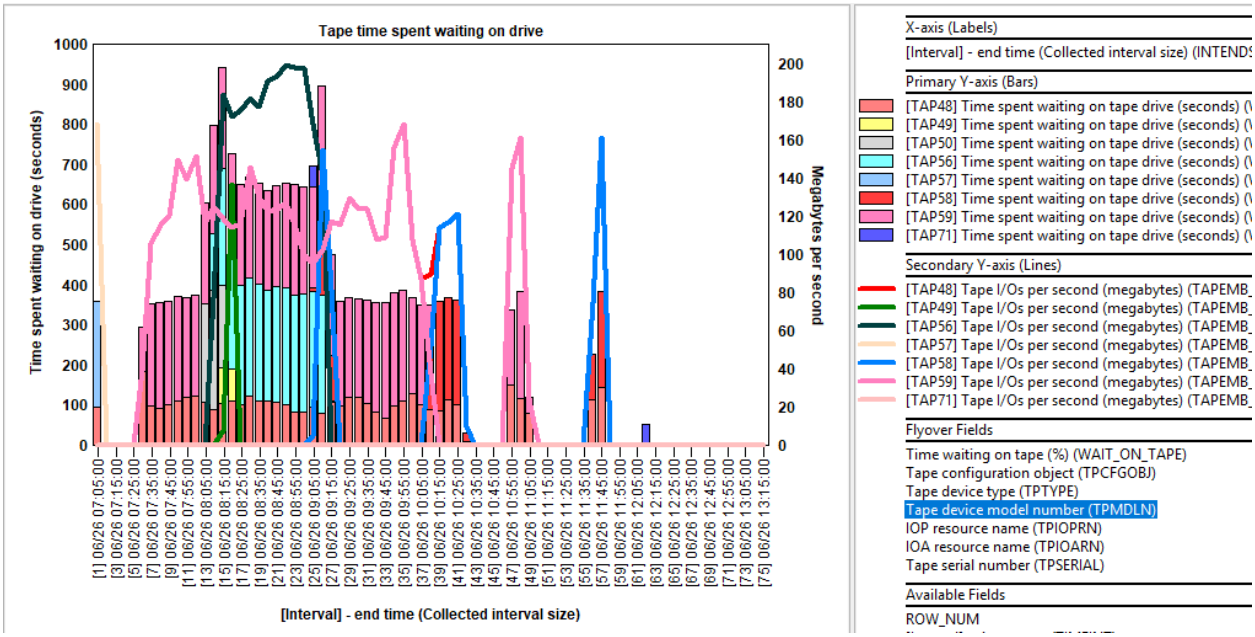
This graph is the same as the previous but includes a second line on the secondary Y-axis to show tape I/Os per second while the tape drive was busy.



Advanced tape wait time overview

9.21.2.3 Tape time spent waiting on drive

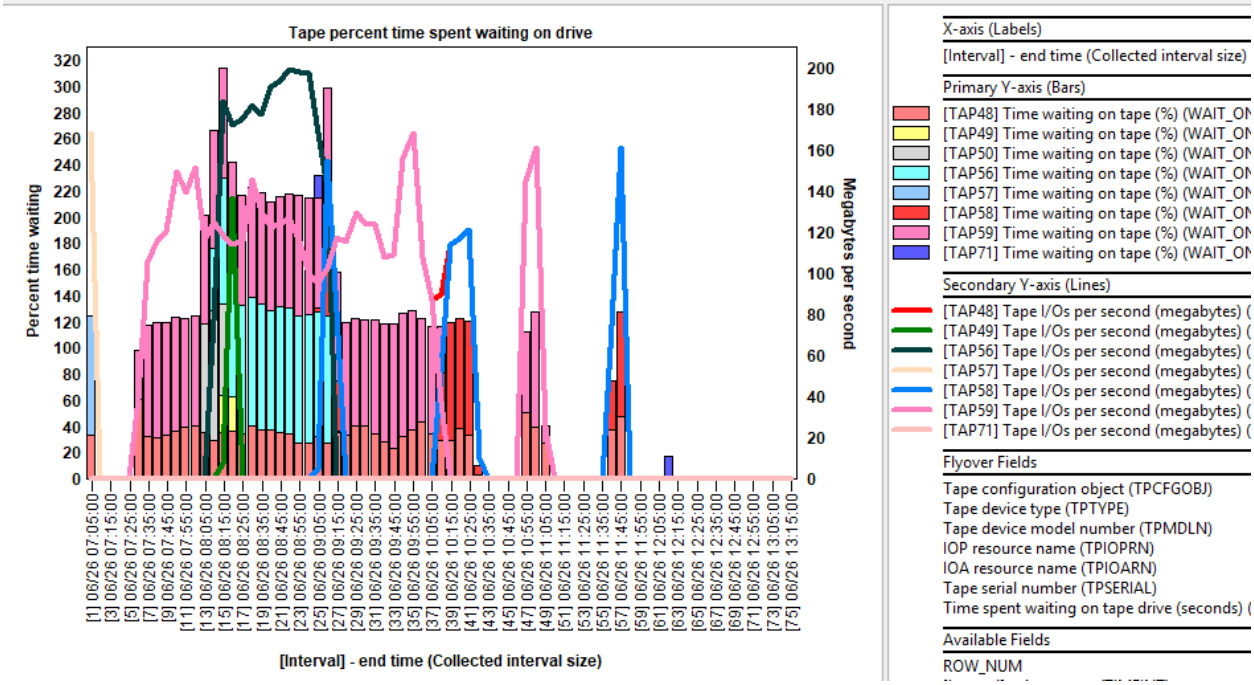
This graph just shows the total time spent waiting on the tape drives over time on the primary Y-axis. The secondary Y-axis displays the tape I/Os per second.



Tape time spent waiting on drive

9.21.2.4 Tape percent time spent waiting on drive

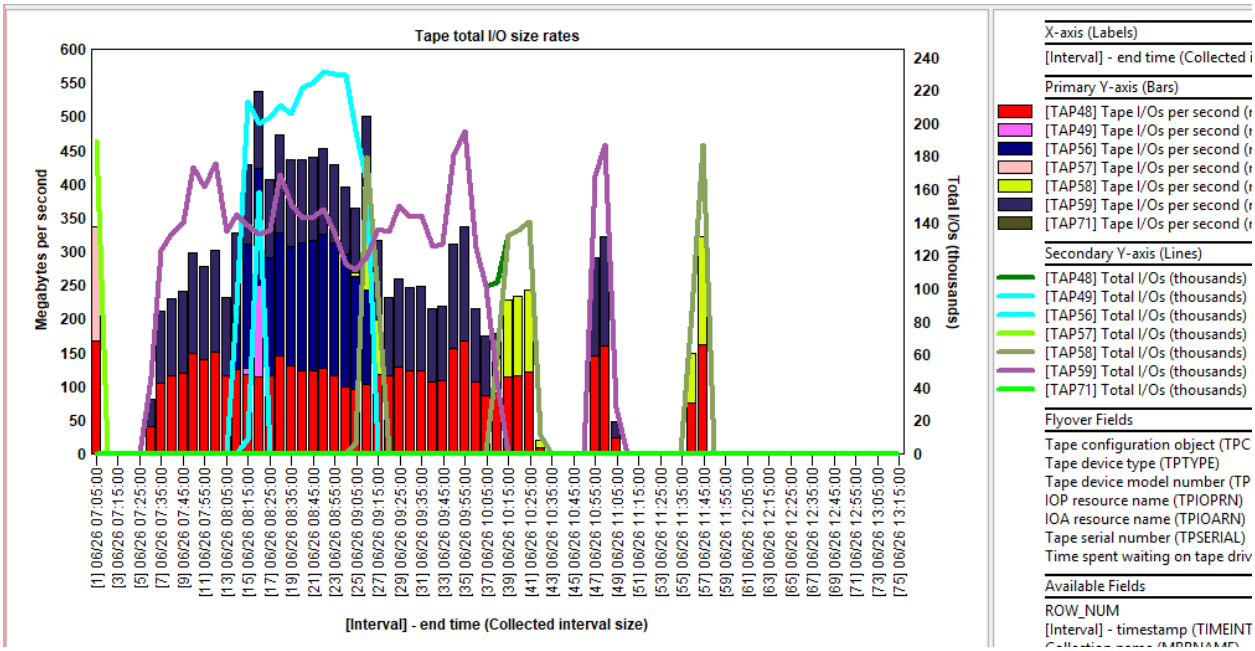
This graph displays the percentage of time spent waiting on each tape drive over time.



Tape percent time spent waiting on drive

9.21.2.5 Tape total I/O size rates

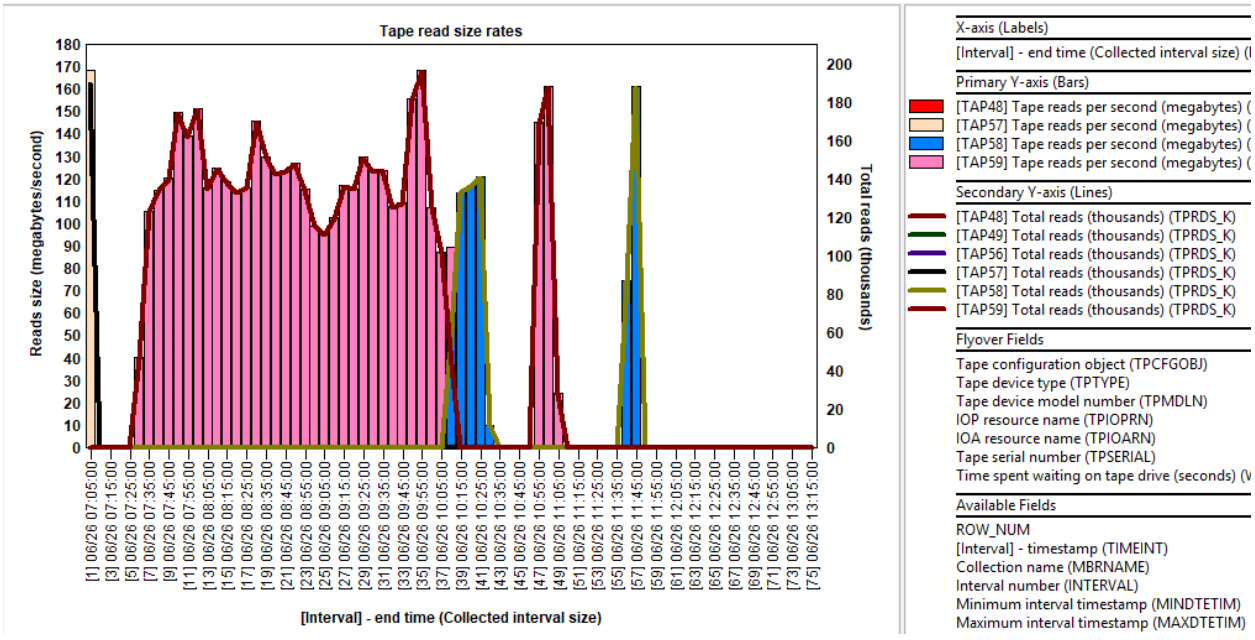
This graph displays each tape drives I/O per second in megabytes on the primary Y-axis and the total I/Os (in thousands) on the secondary Y-axis.



Tape total I/O size rates

9.21.2.6 Tape read size rates

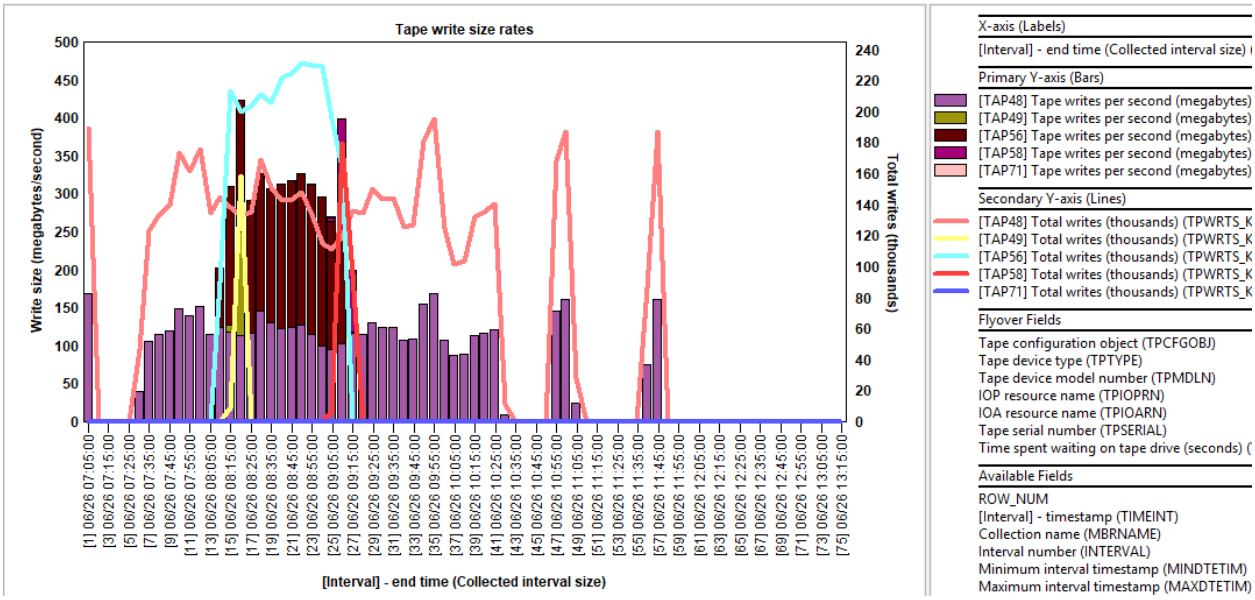
This graph shows the amount of data read in megabytes per second from the tape drives over time. The secondary Y-axis displays the total reads (in thousands.)



Tape read size rates

9.21.2.7 Tape write size rates

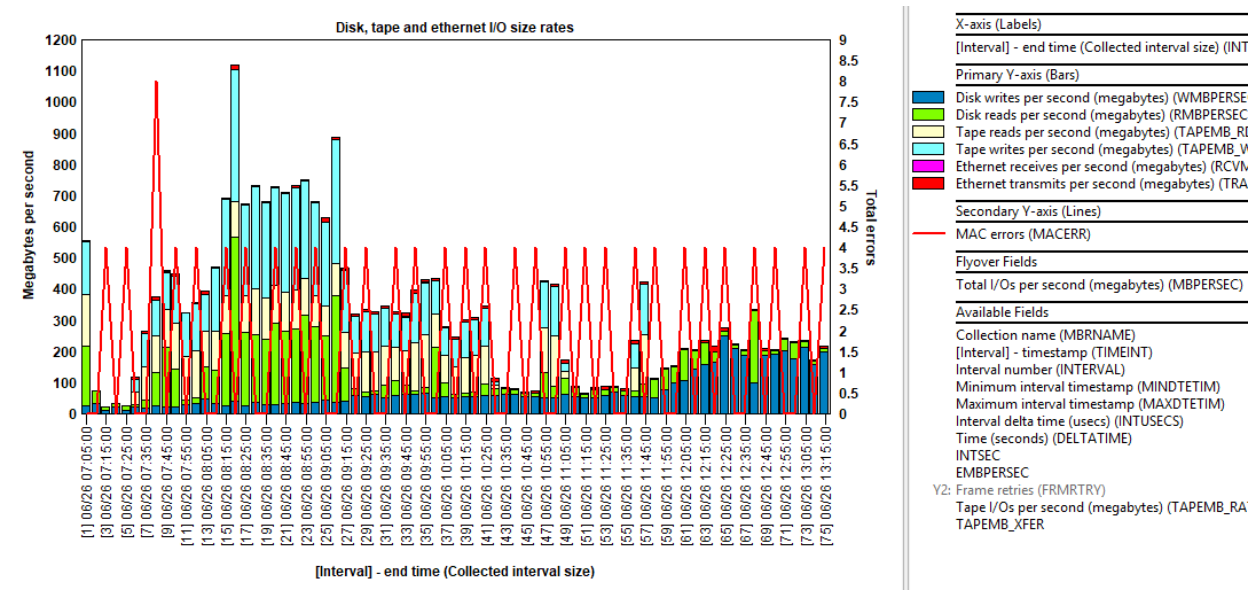
This graph shows the amount of data written in megabytes per second from the tape drives over time. The secondary Y-axis displays the total writes (in thousands.)



Tape write size rates

9.21.2.8 Disk, tape and ethernet I/O size rates

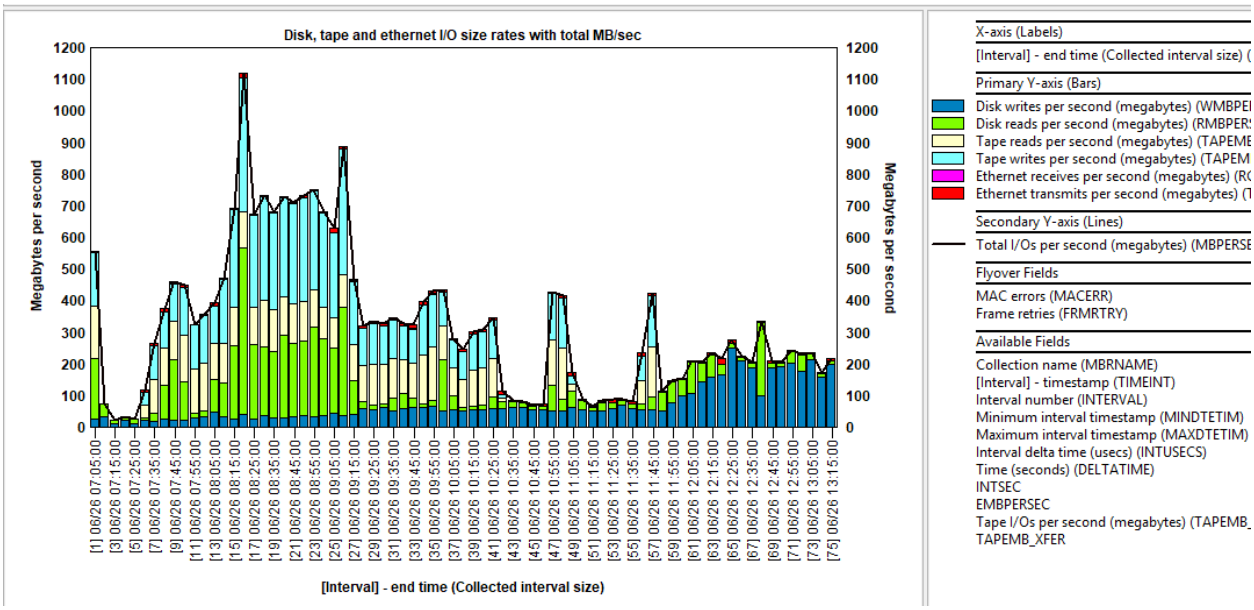
This graph combines disk, tape and ethernet I/O size rates per second in megabytes over time.



Disk, tape and ethernet I/O size rates

9.21.2.9 Disk, tape and ethernet I/O size rates with total MB/sec

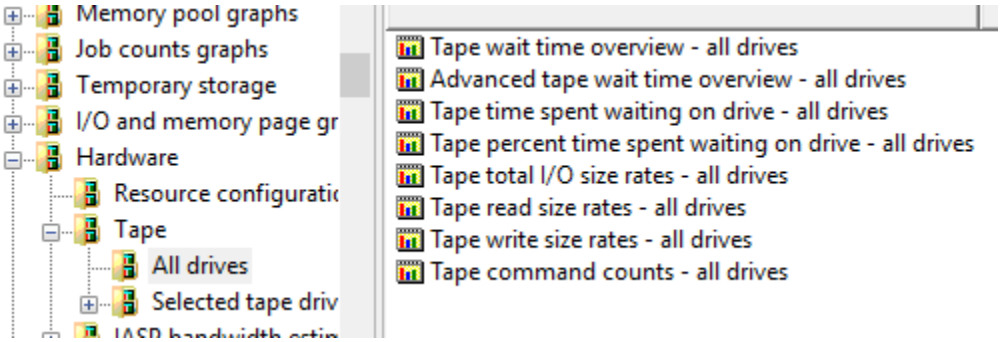
This graph is the same as the previous one except combines the data from all the metrics on the primary Y-axis as a total on the secondary Y-axis.



Disk, tape and ethernet I/O size rates with total MB/sec

9.21.2.10 All drives

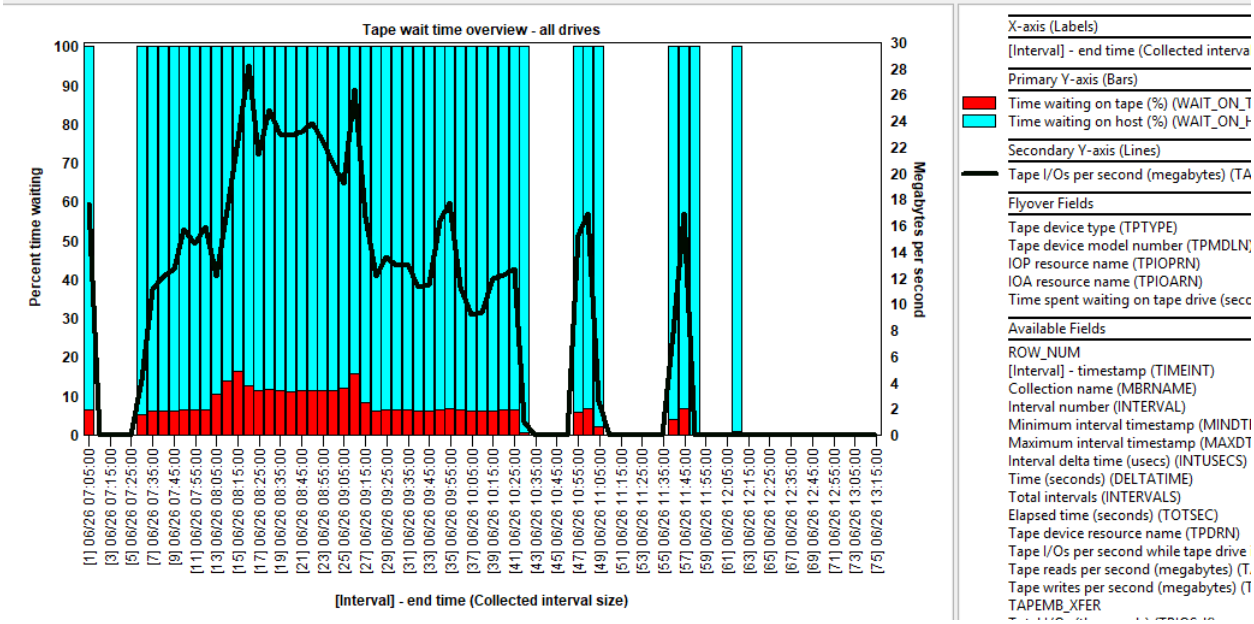
These graphs combine all tape drives together into a set of graphs over time.



Hardware -> Tape -> All drives

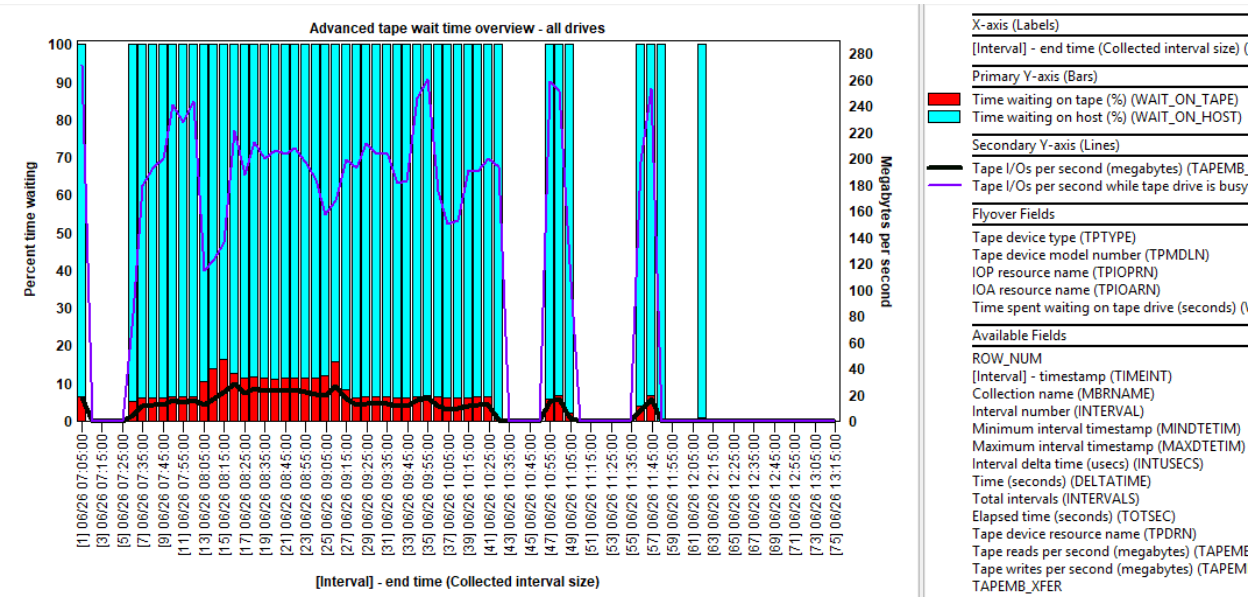
Note: Examples are shown below but see the Tape section for more details about the metrics provided on these graphs.

9.21.2.10.1 Tape wait time overview - all drives



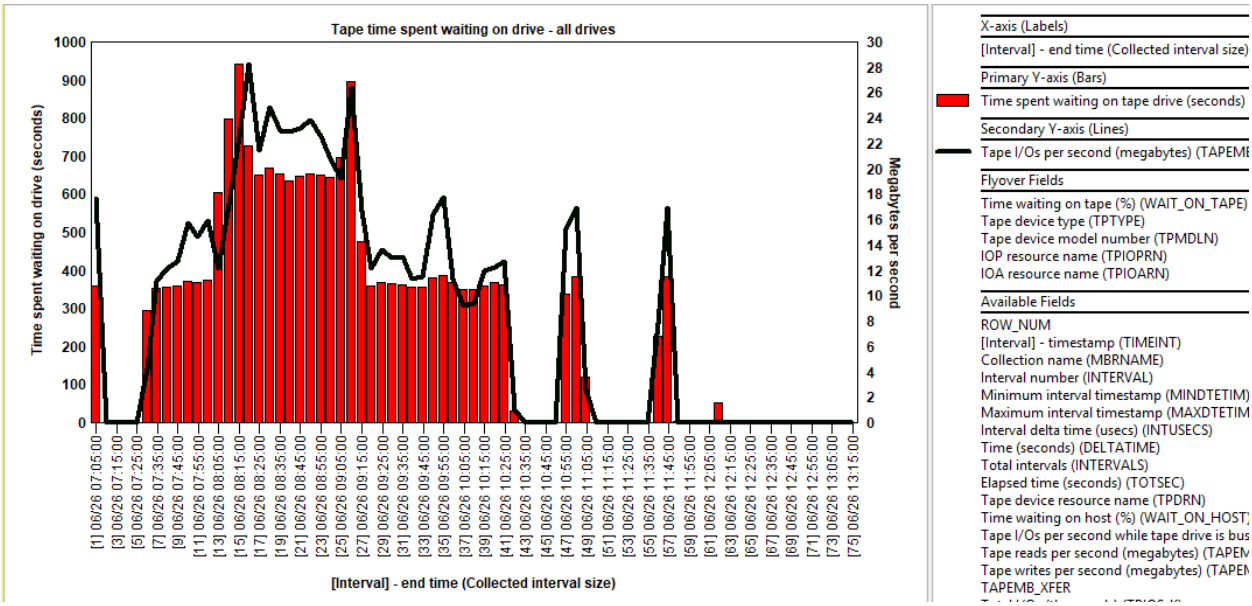
Tape wait time overview – all drives

9.21.2.10.2 Advanced tape wait time overview - all drives



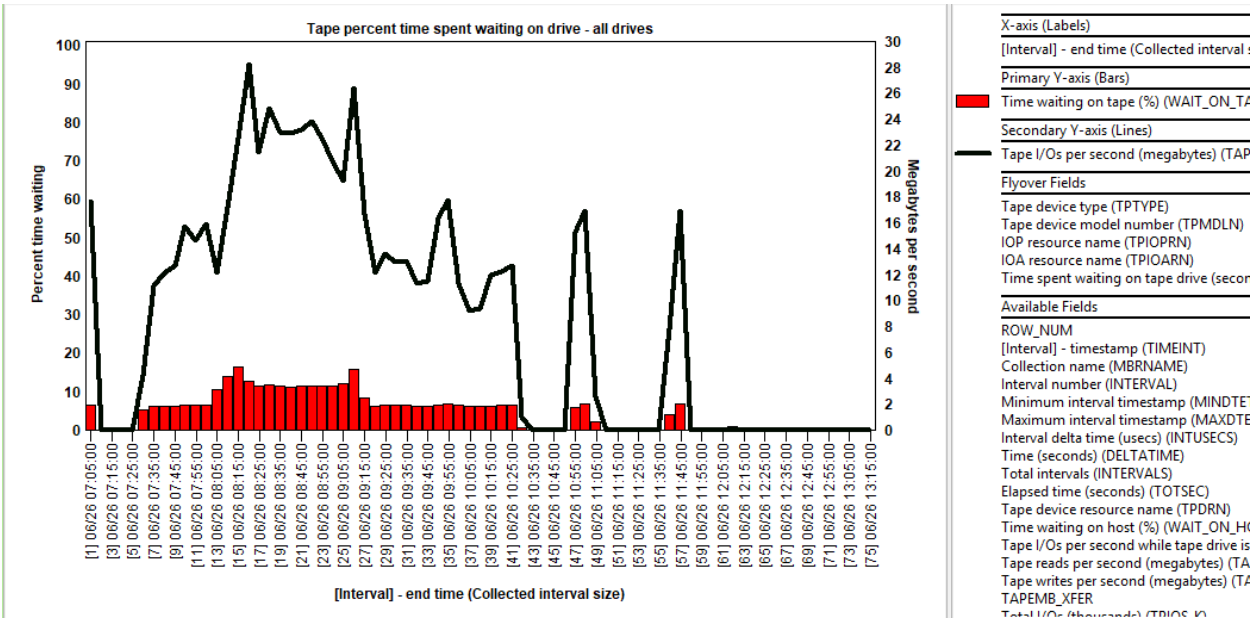
Advanced tape wait time overview - all drives

9.21.2.10.3 Tape time spent waiting on drive - all drives



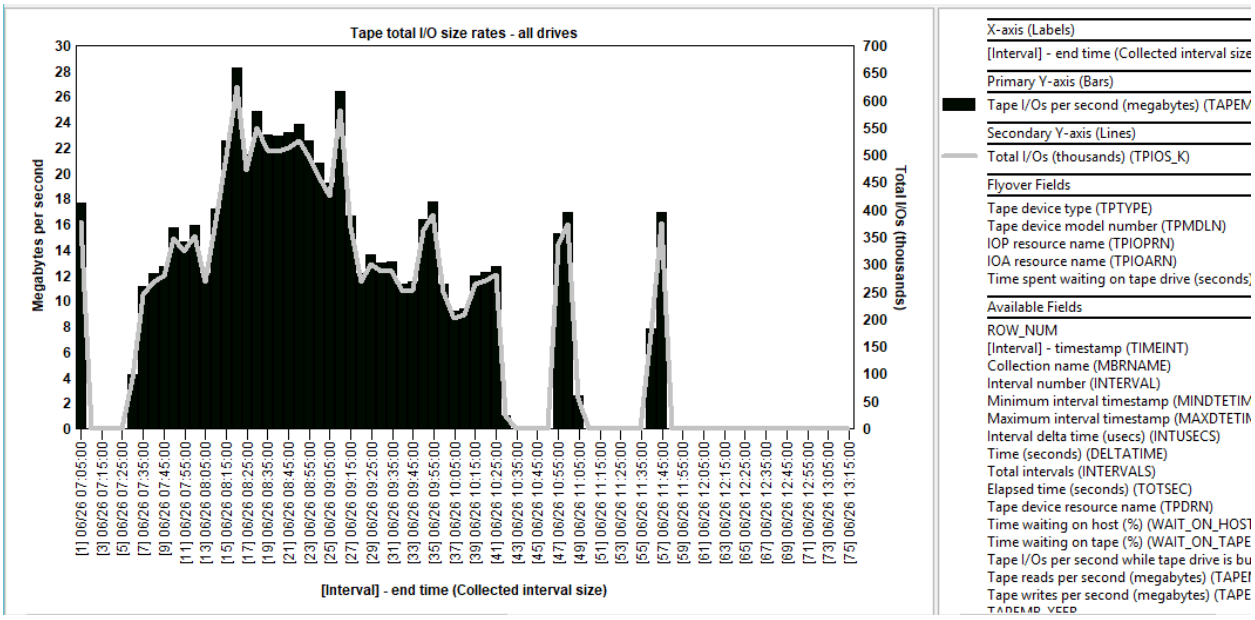
Tape time spent waiting on drive - all drives

9.21.2.10.4 Tape percent time spent waiting on drive - all drives



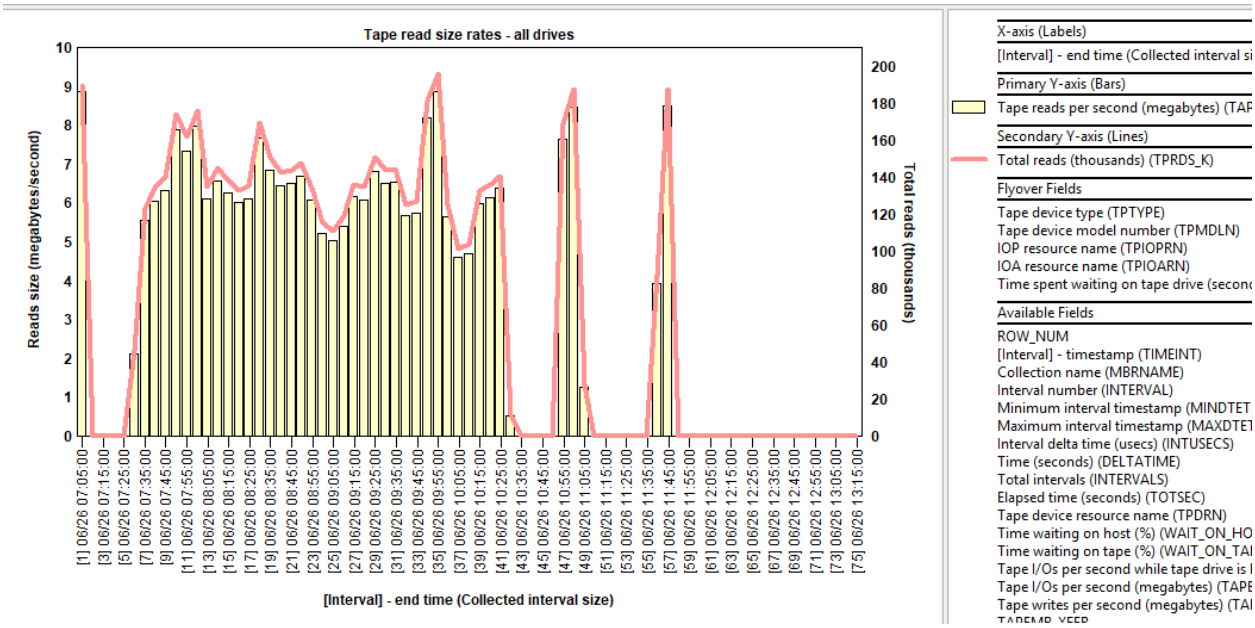
Tape percent time spent waiting on drive - all drives

9.21.2.10.5 Tape total I/O size rates - all drives



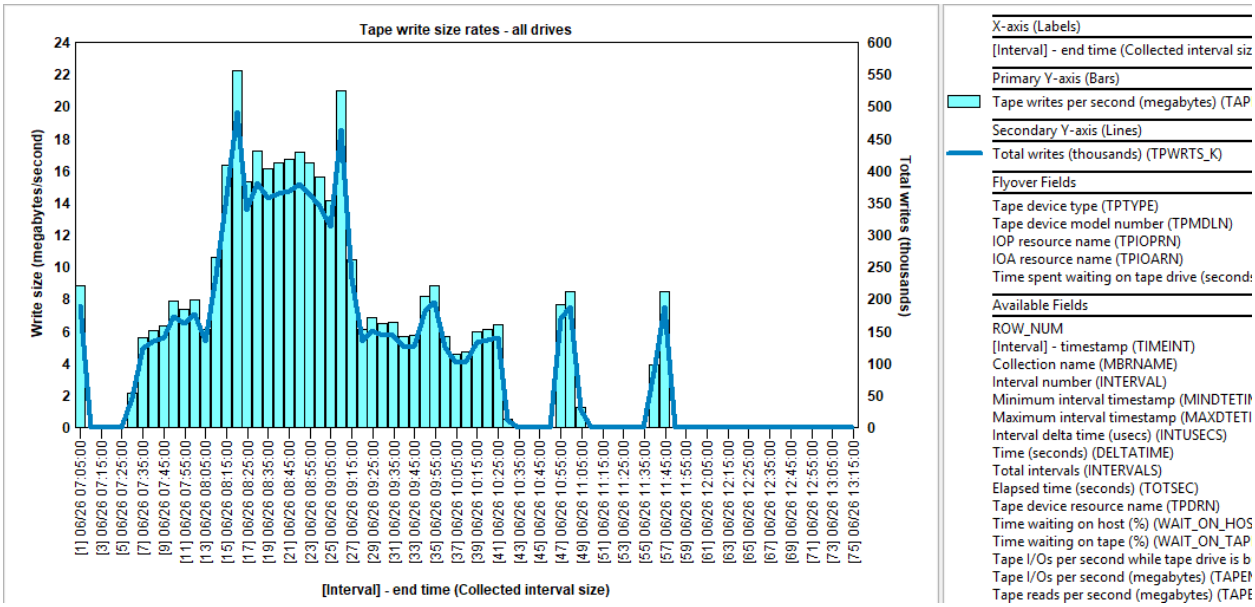
Tape total I/O size rates - all drives

9.21.2.10.6 Tape read size rates - all drives



Tape read size rates - all drives

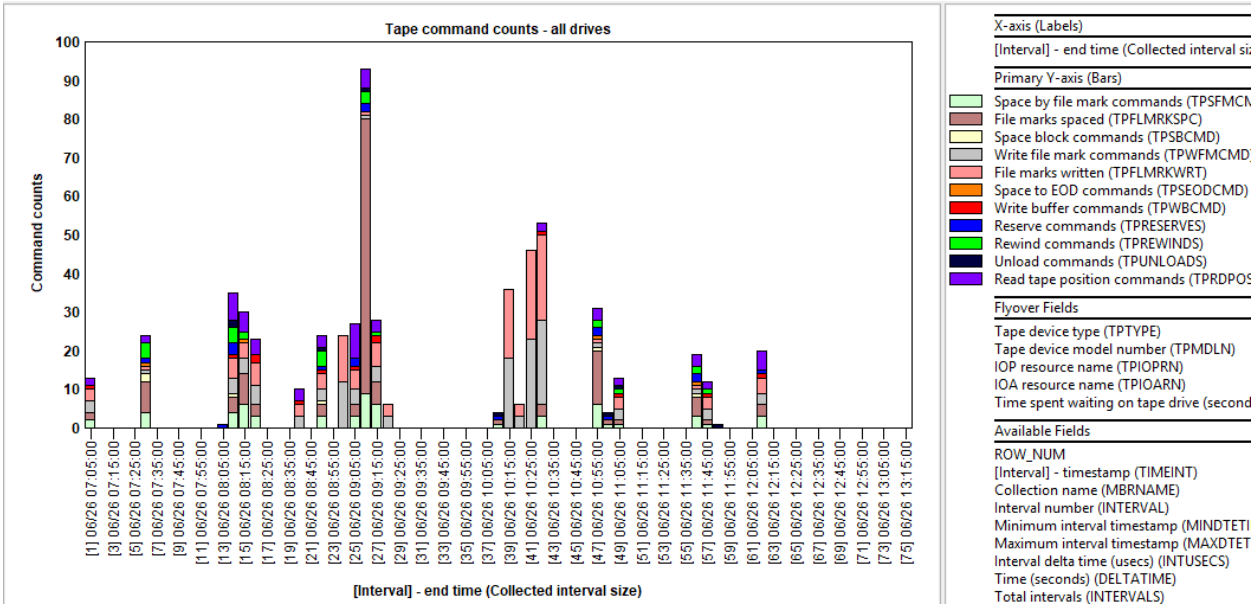
9.21.2.10.7 Tape write size rates - all drives



Tape write size rates - all drives

9.21.2.10.8 Tape command counts - all drives

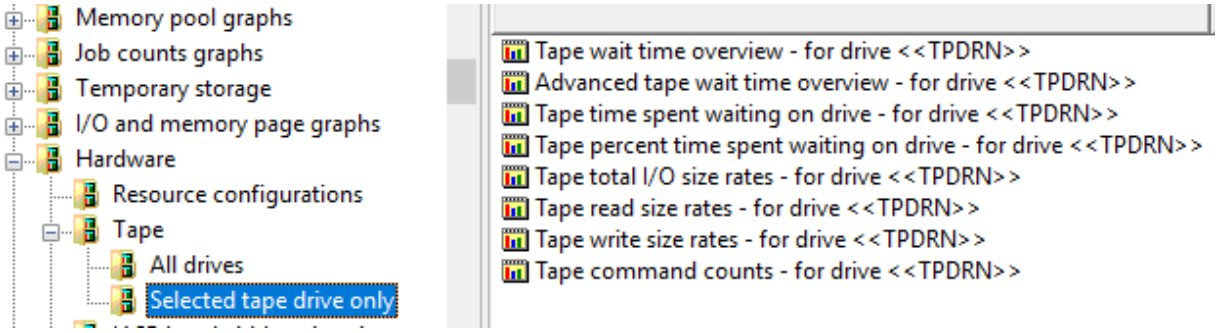
This graph displays the total counts for various types of tape drive commands (added together for all drives) over time.



Tape command counts - all drives

9.21.2.11 Selected tape drive only

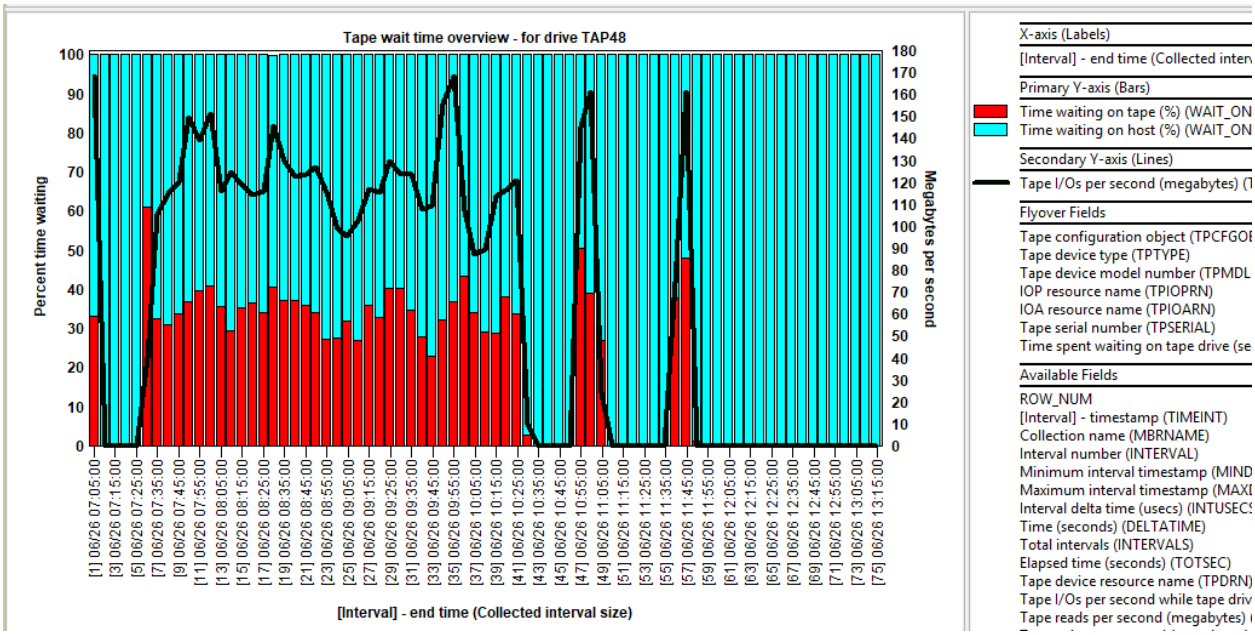
These graphs are the same as the previous set but only for a specific drive selected. You must fill in the tape drive name when opening any of these graphs.



Hardware -> Tape -> Selected tape drive only

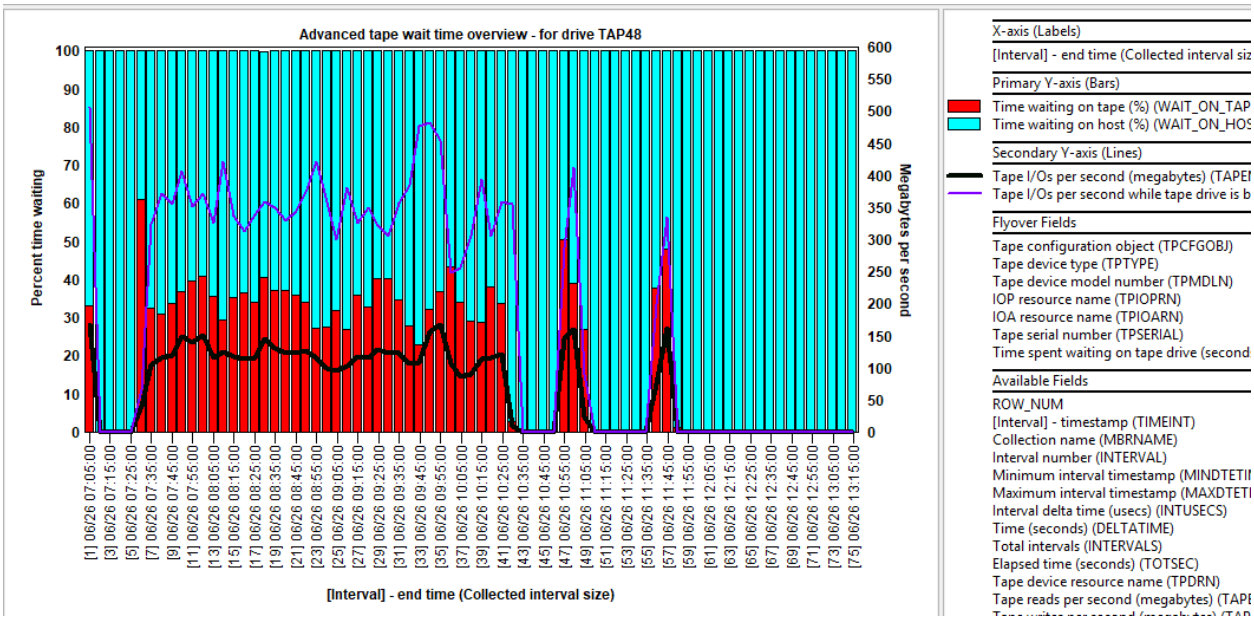
Note: Examples are shown below but see the Tape section for more details about the metrics provided on these graphs.

9.21.2.11.1 Tape wait time overview - for drive <<TPDRN>>



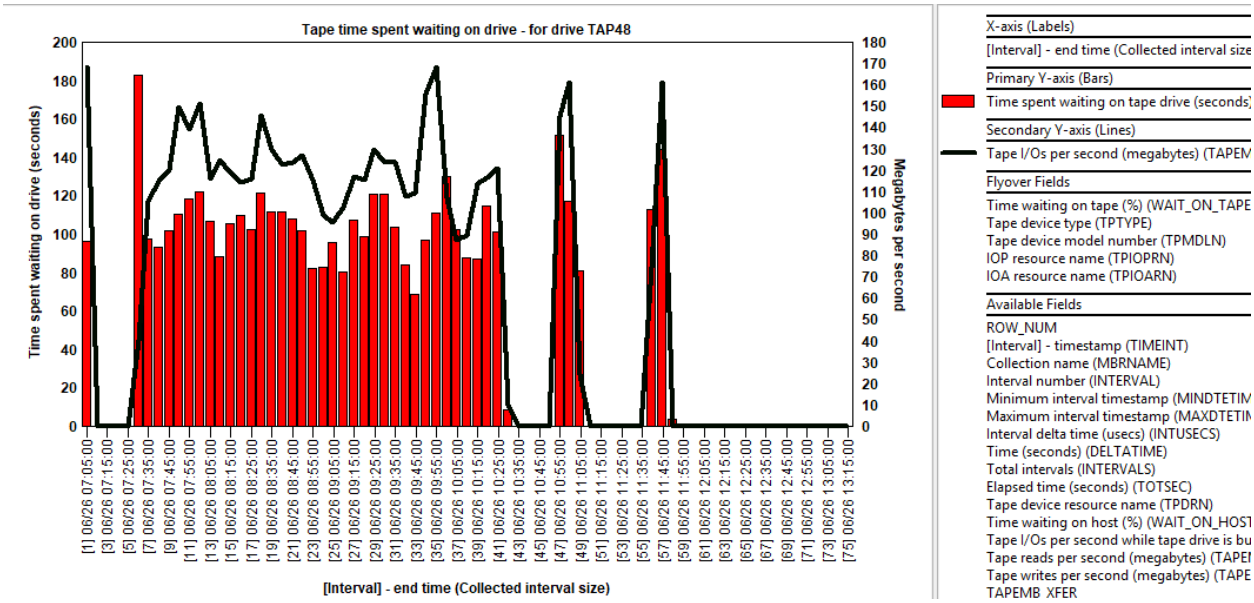
Tape wait time overview – for drive <<TPDRN>>

9.21.2.11.2 Advanced tape wait time overview - for drive <<TPDRN>>



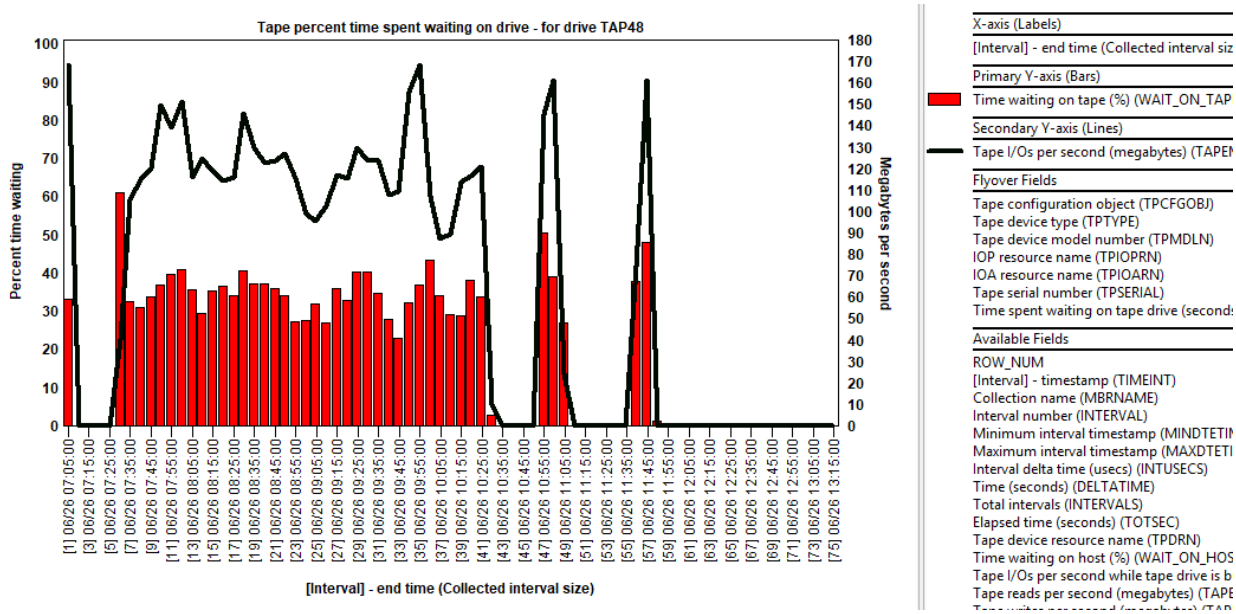
Advanced tape wait time overview - for drive <<TPDRN>>

9.21.2.11.3 Tape time spent waiting on drive - for drive <<TPDRN>>



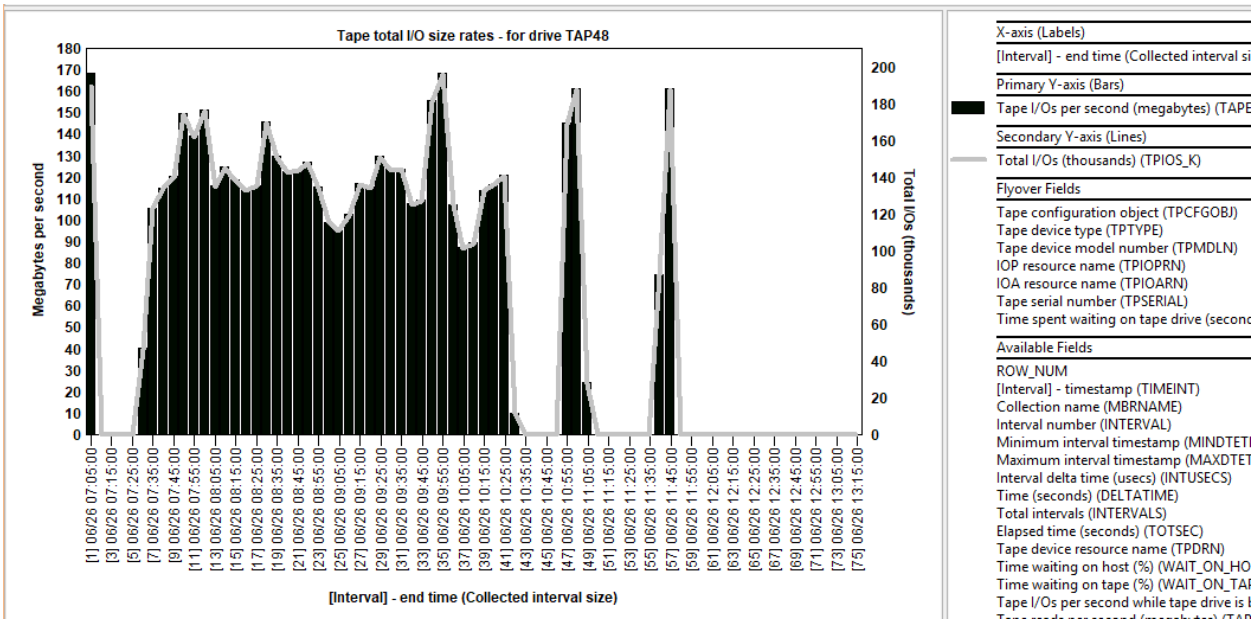
Tape time spent waiting on drive - for drive <<TPDRN>>

9.21.2.11.4 Tape percent time spent waiting on drive - for drive <<TPDRN>>



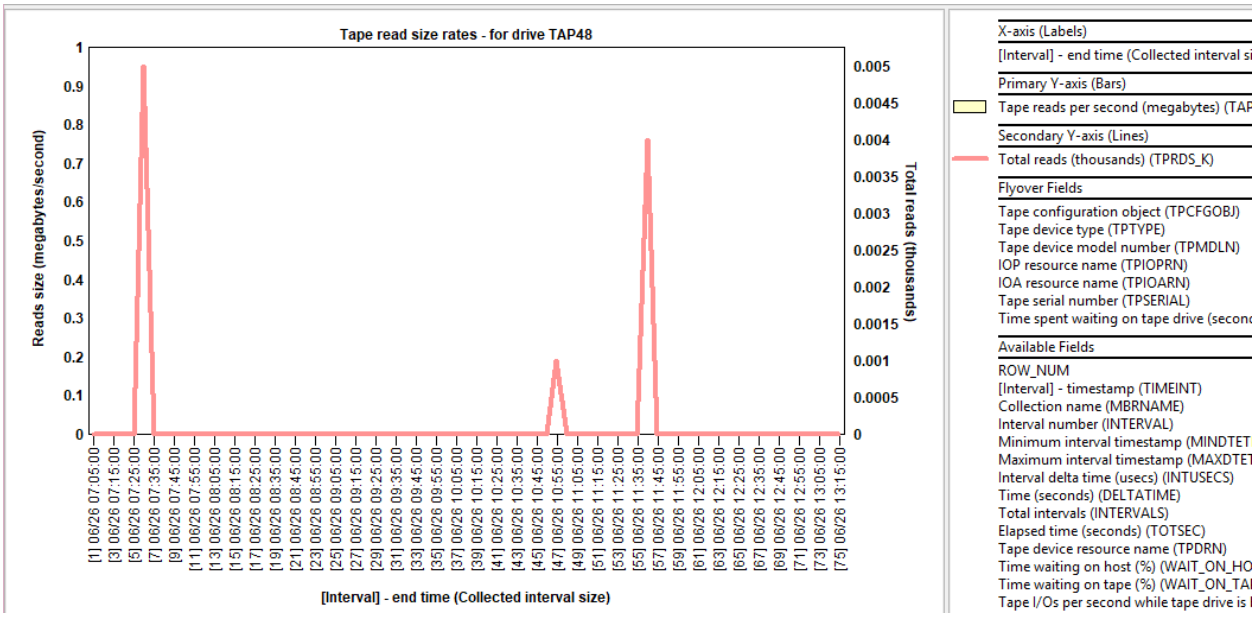
Tape percent time spent waiting on drive - for drive <<TPDRN>>

9.21.2.11.5 Tape total I/O size rates - for drive <<TPDRN>>



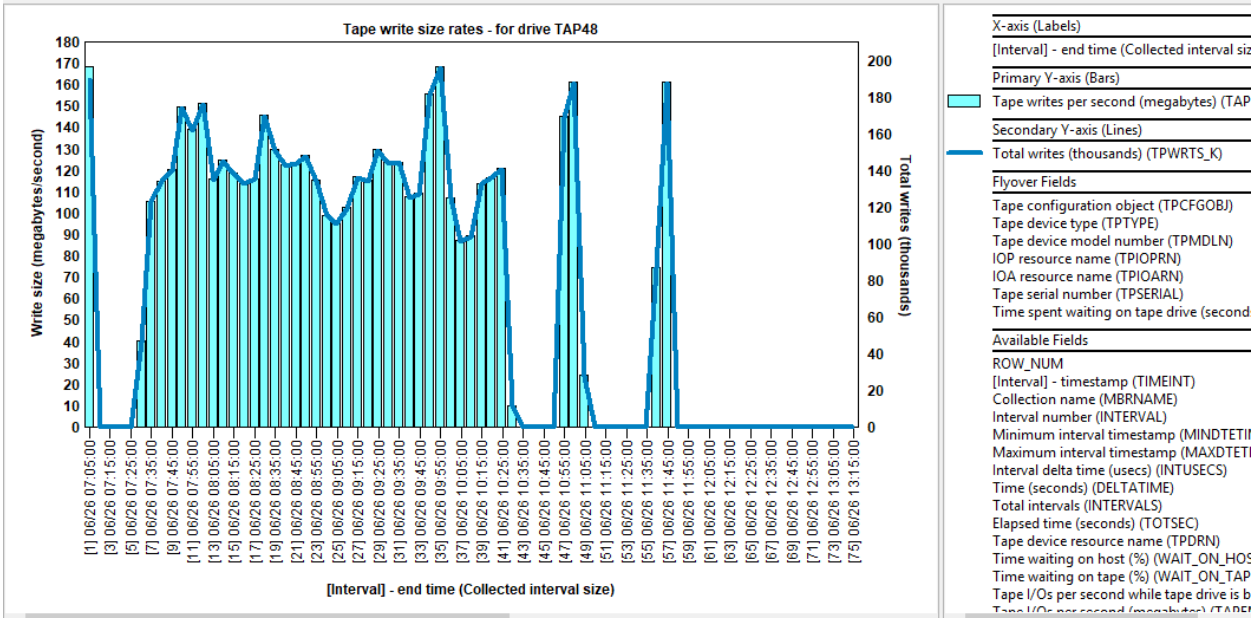
Tape total I/O size rates - for drive <<TPDRN>>

9.21.2.11.6 Tape read size rates - for drive <<TPDRN>>



Tape read size rates - for drive <<TPDRN>>

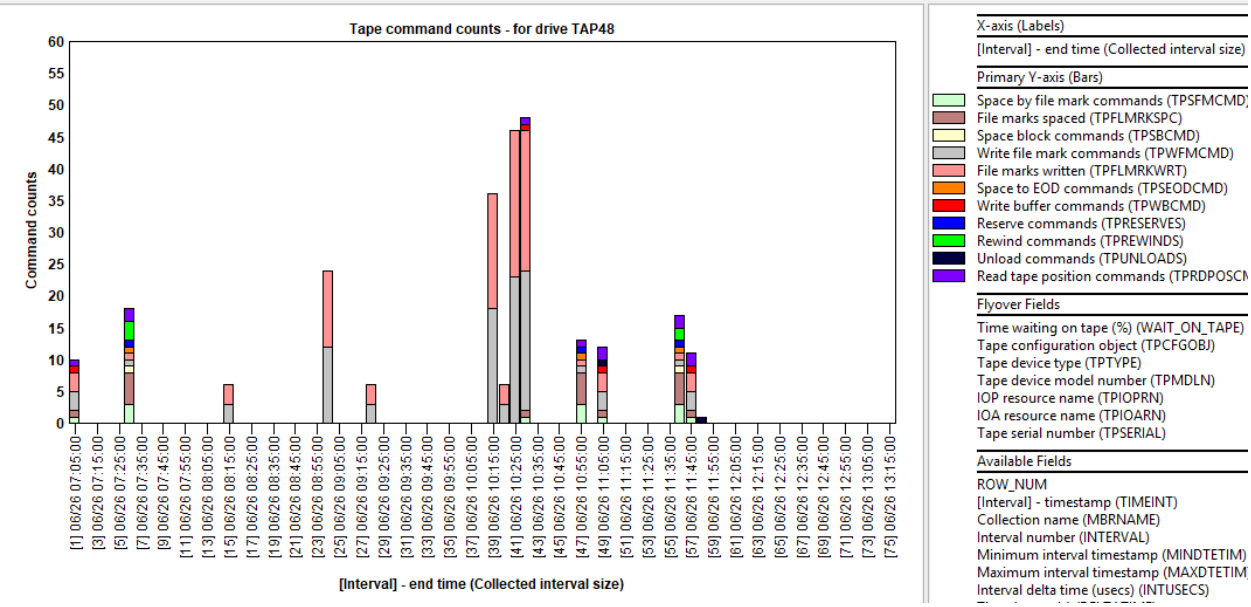
9.21.2.11.7 Tape write size rates - for drive <<TPDRN>>



Tape write size rates - for drive <<TPDRN>>

9.21.2.11.8 Tape command counts - for drive <<TPDRN>>

This graph displays the total counts for various types of tape drive commands (added together for all drives) over time.

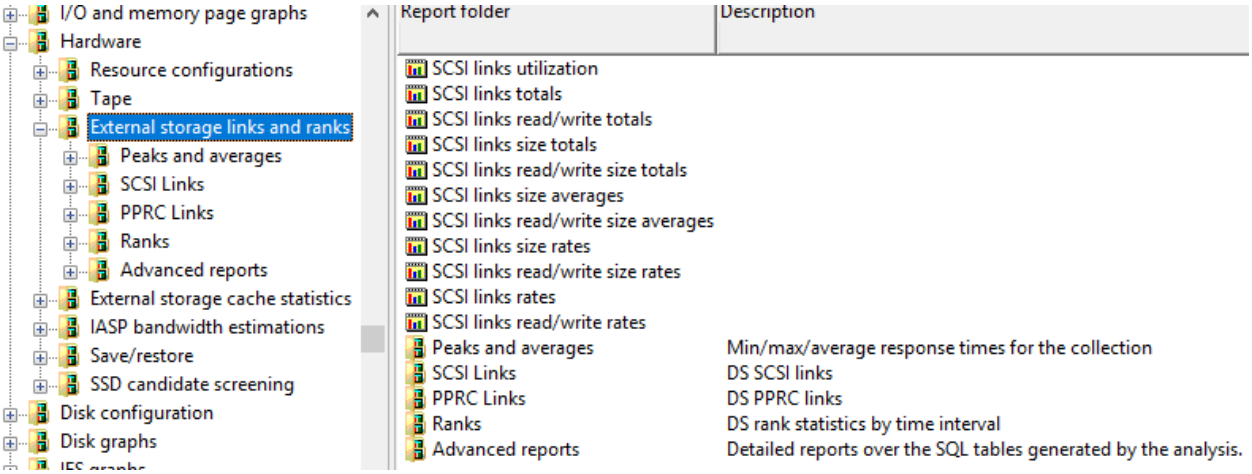


Tape command counts - for drive <<TPDRN>>

9.21.3 External storage links and ranks

These graphs show metrics for external storage systems (such as DS8000) and provides metrics for both SCSI/PPRC links as well as the ranks found on the disk storage system.

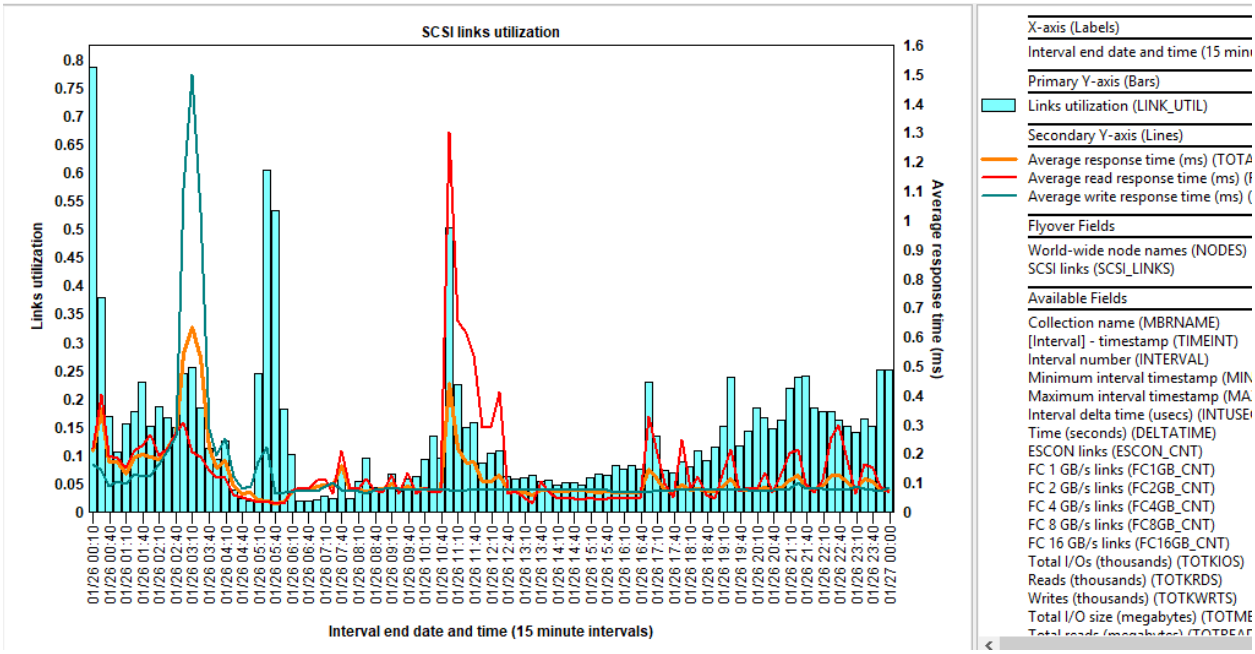
Note: The Collection Services file QAPMXSTGV must exist in the collection AND External storage links and ranks analysis must be ran first in order for these graphs to be available.



Hardware -> External storage links and ranks

9.21.3.1 SCSI links utilization

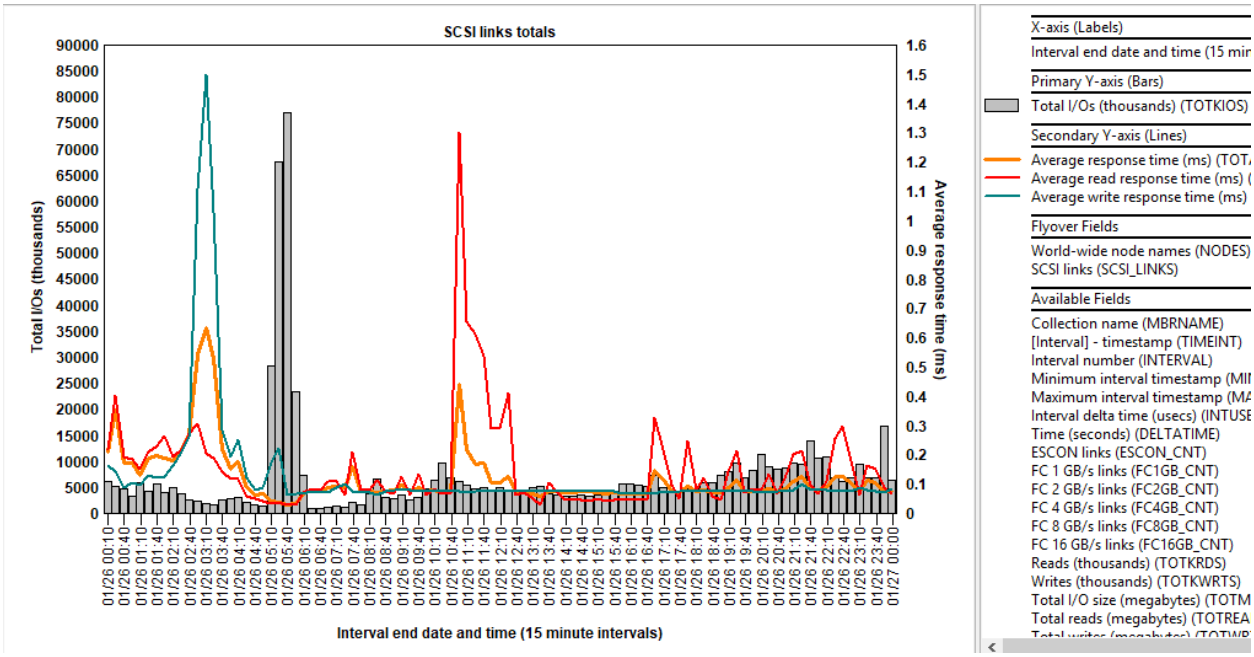
This graph displays the average SCSI link utilization for the external storage metrics found in the collection. The secondary Y-axis displays the average response times (in milliseconds.)



SCSI links utilization

9.21.3.2 SCSI links totals

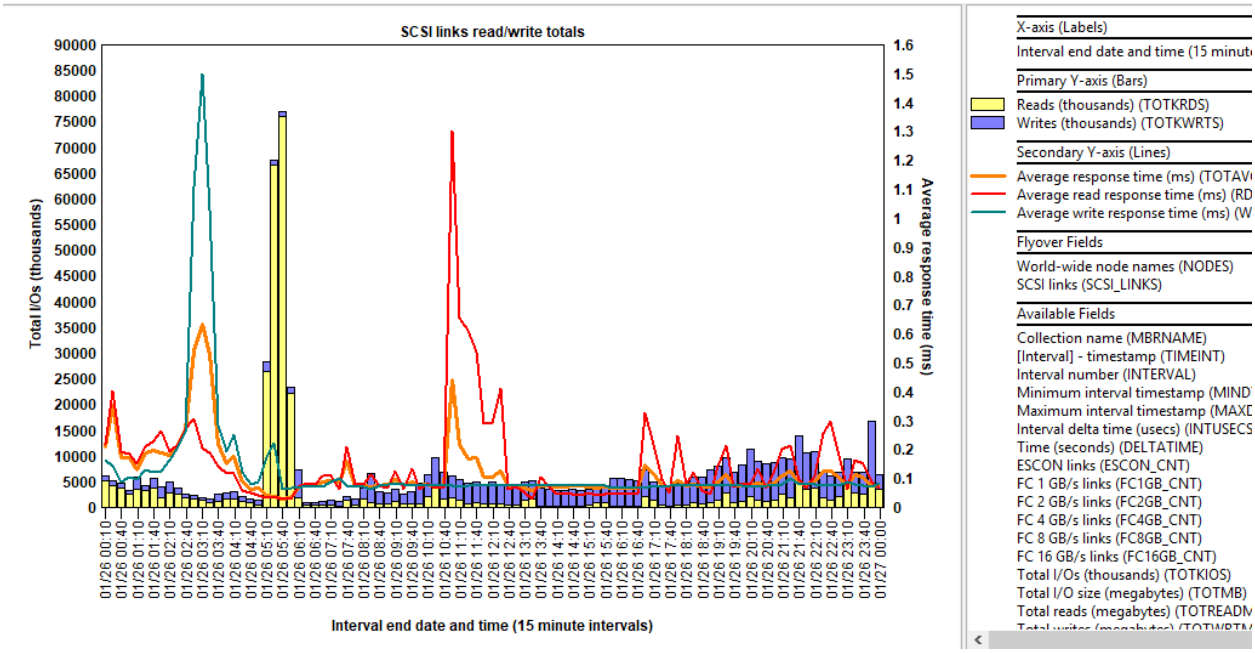
This graph displays the total I/Os (in thousands) for the SCSI links found in the collection. The secondary Y-axis displays the average response times (in milliseconds.)



SCSI links totals

9.21.3.3 SCSI links read/write totals

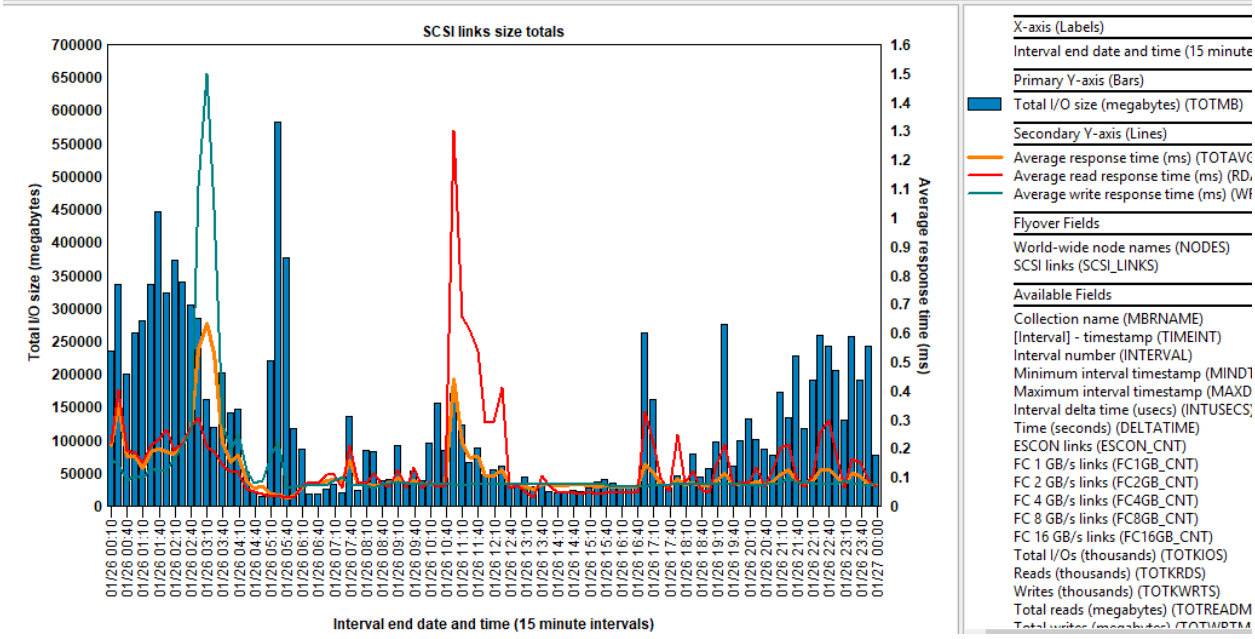
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



SCSI links read/write totals

9.21.3.4 SCSI links size totals

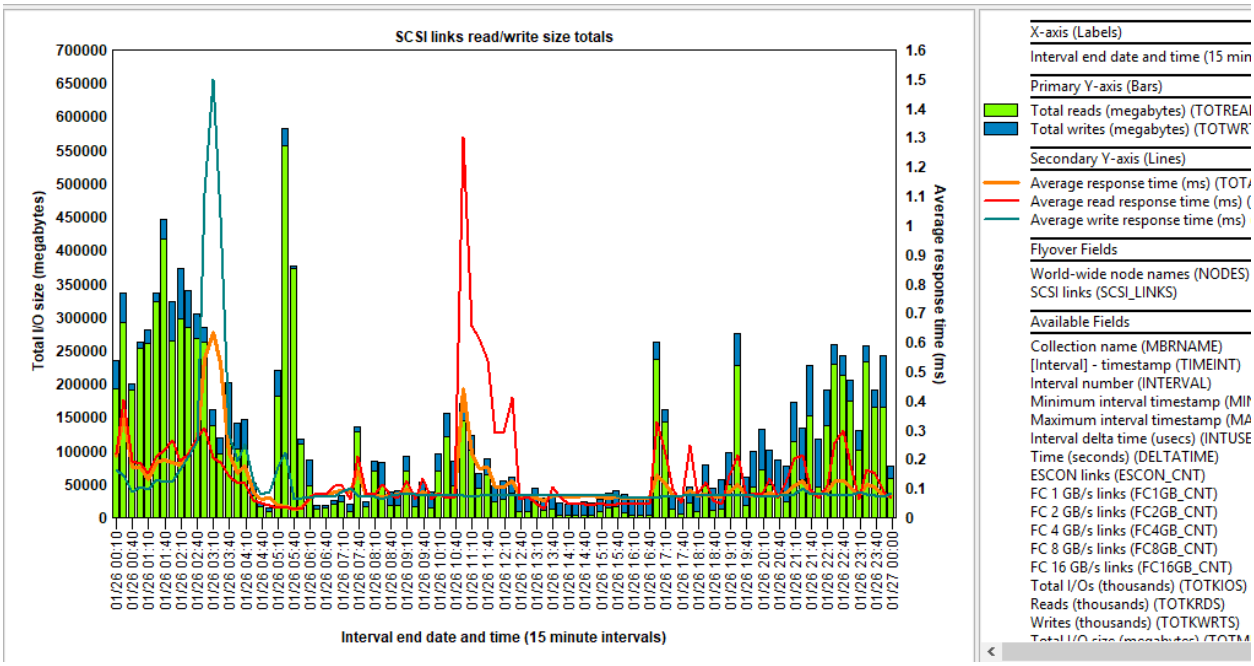
This graph displays the total size of all read and writes SCSI link IO operations over time (in megabytes.) The secondary Y-axis displays the average response times (in milliseconds.)



SCSI links size totals

9.21.3.5 SCSI links read/write size totals

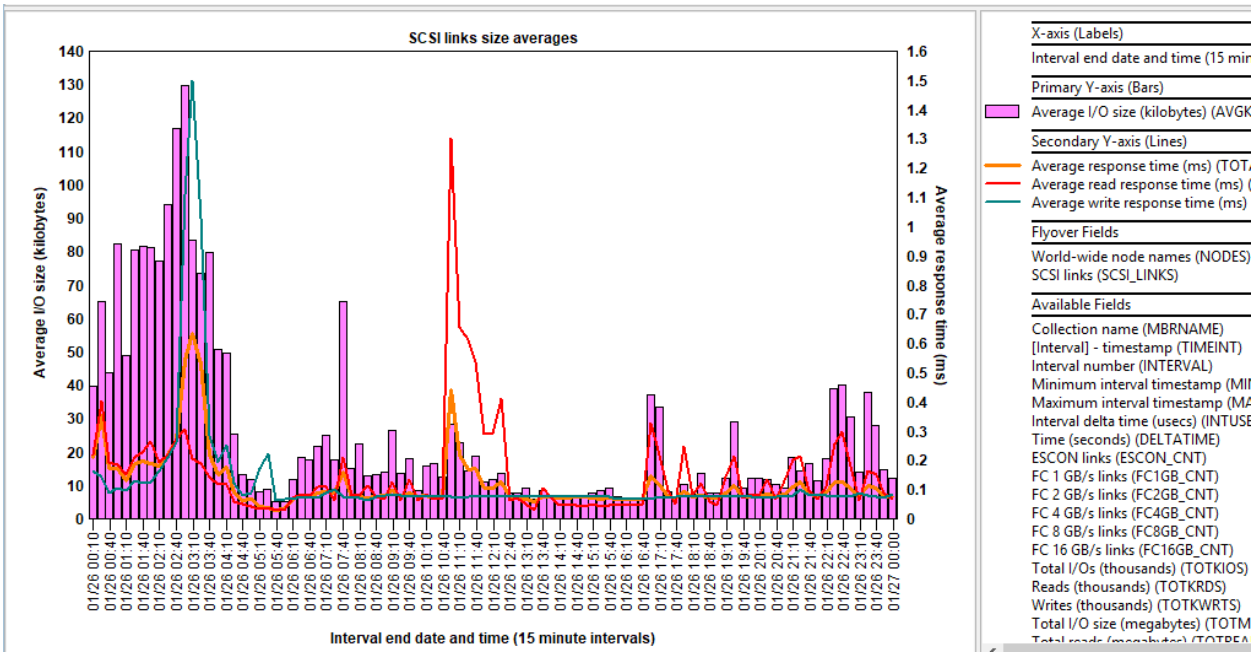
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



SCSI links read/write size totals

9.21.3.6 SCSI links size averages

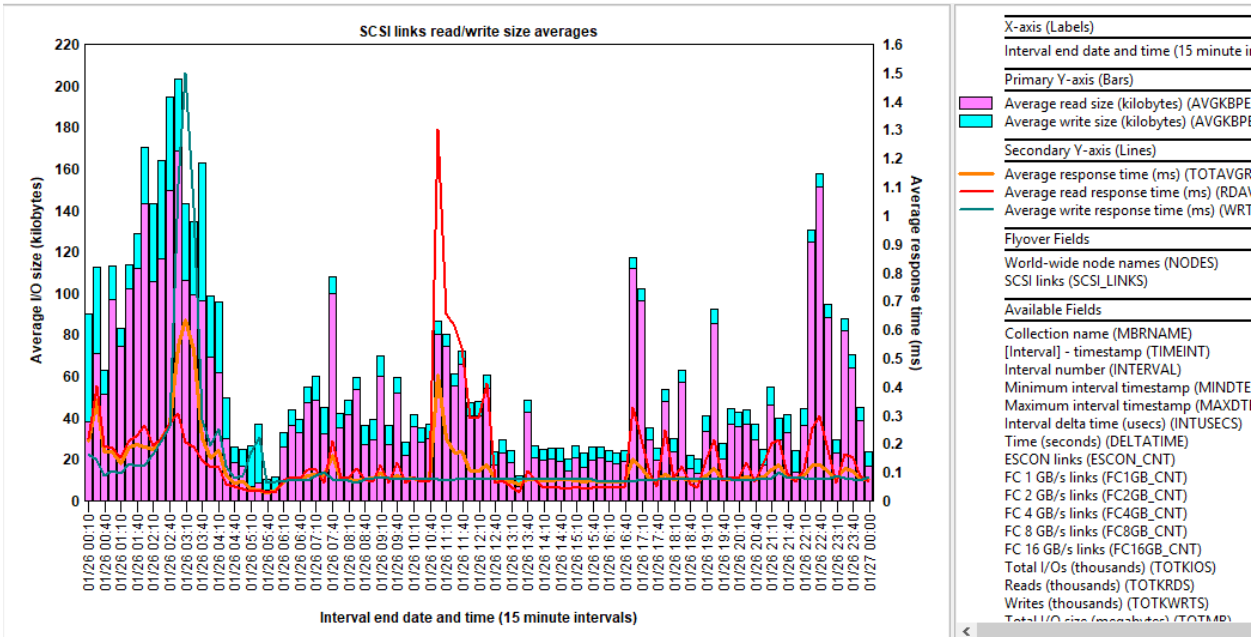
This graph shows the average I/O size for both reads and writes together over time.



SCSI links size averages

9.21.3.7 SCSI links read/write size averages

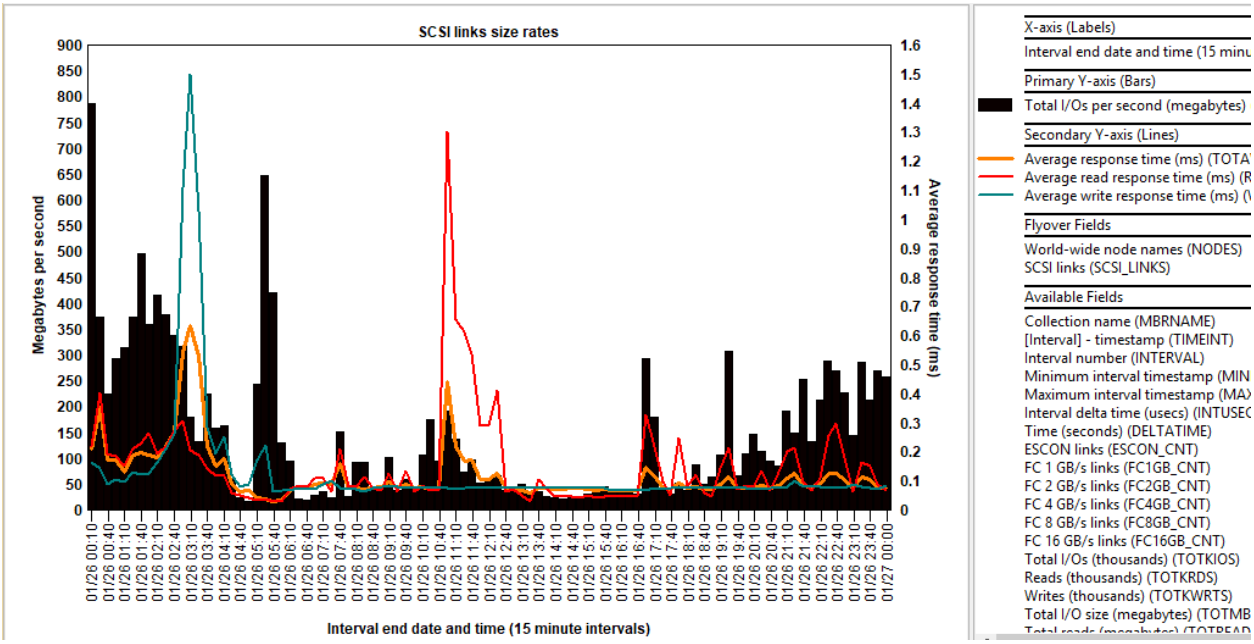
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



SCSI links read/write size averages

9.21.3.8 SCSI links size rates

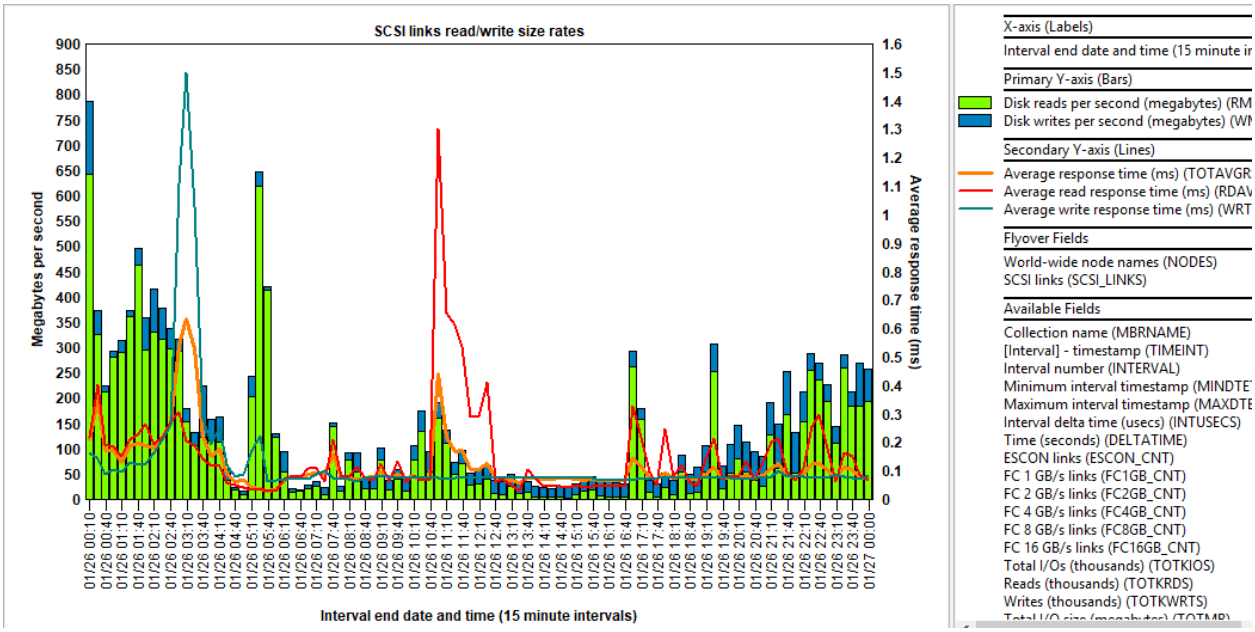
This graph displays the total I/O per second in megabytes over time for the SCSI links found in the collection.



SCSI links size rates

9.21.3.9 SCSI links read/write size rates

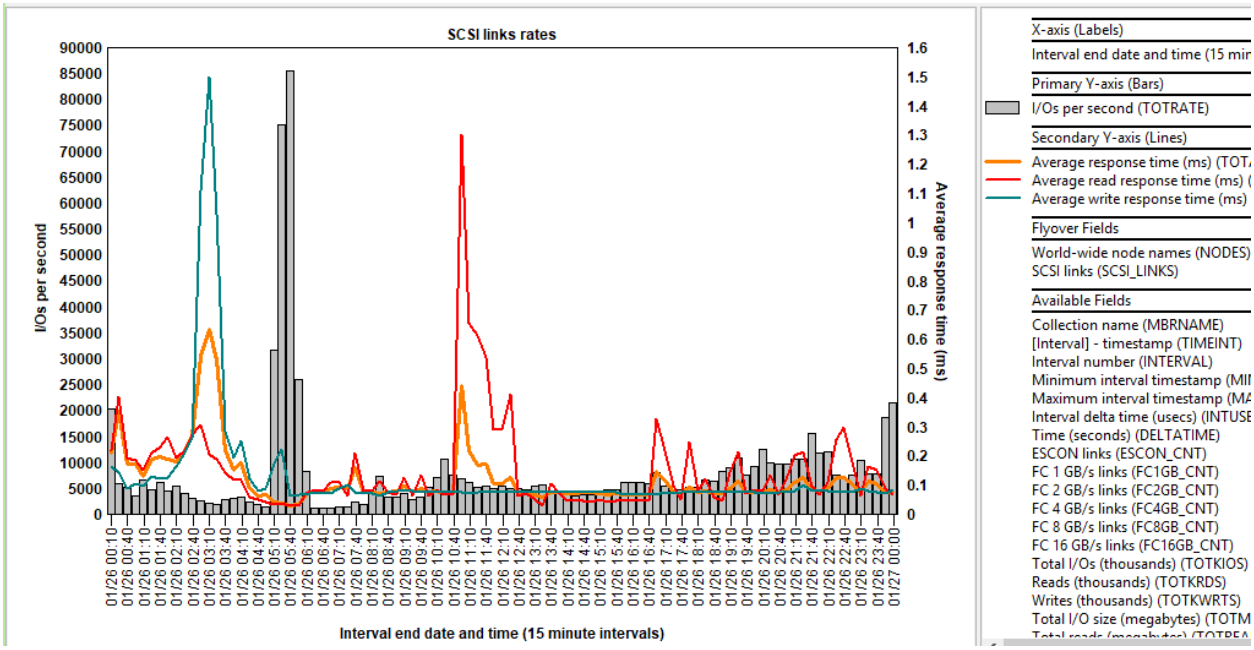
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



SCSI links read/write size rates

9.21.3.10 SCSI links rates

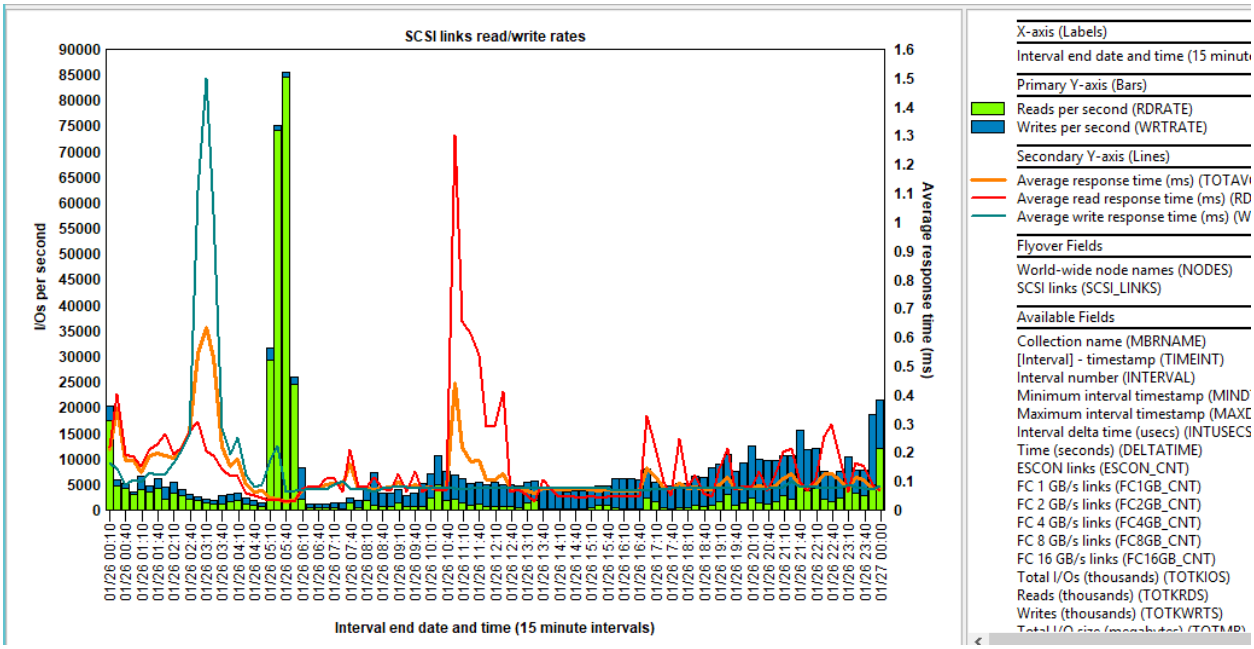
This graph displays the I/Os per second over time for the SCSI links found in the collection.



SCSI links rates

9.21.3.11 SCSI links read/write rates

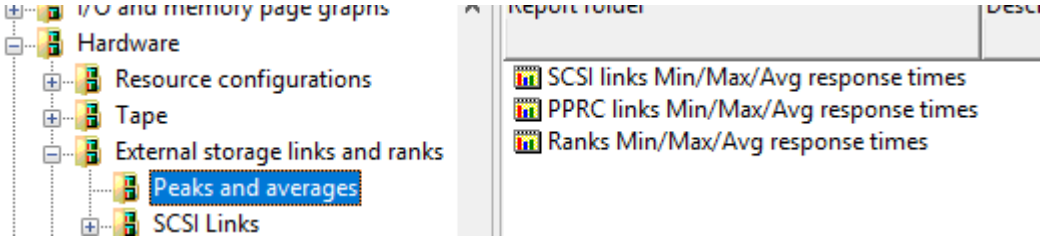
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



SCSI links read/write rates

9.21.3.12 Peaks and averages

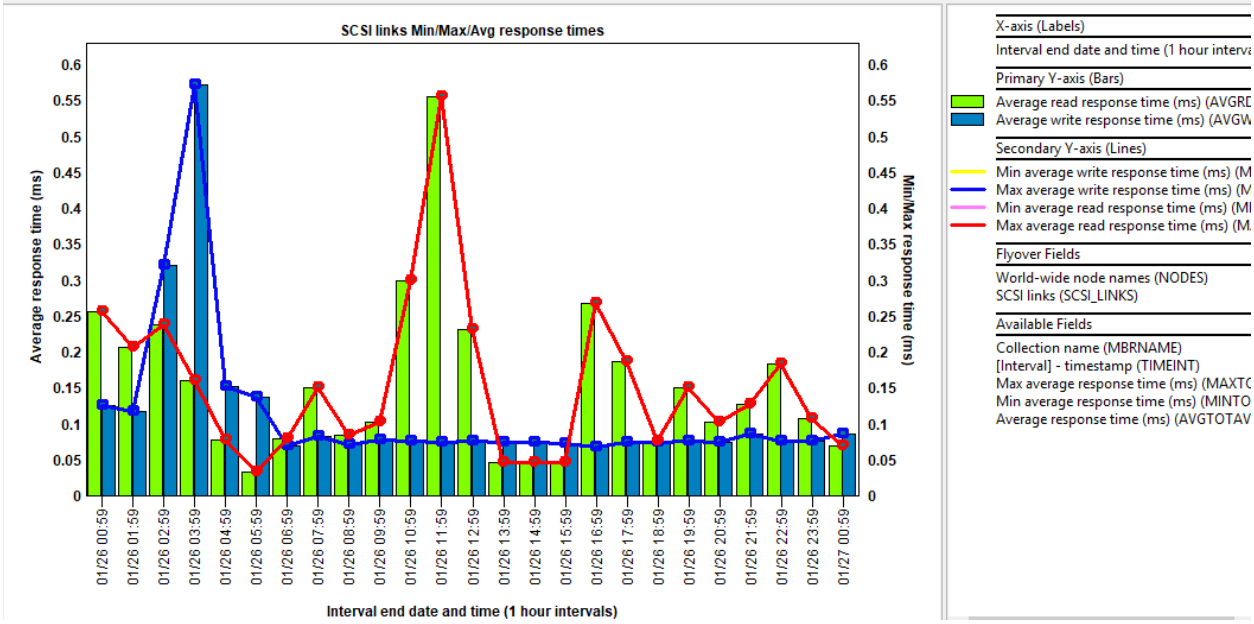
These graphs display average, min and max response times for the links and ranks found in the data collection. These are meant to be high-level graphs so by default the time groupings are shown with 1 hour intervals.



Hardware -> External storage links and ranks -> Peaks and averages

9.21.3.12.1 SCSI links Min/Max/Avg response times

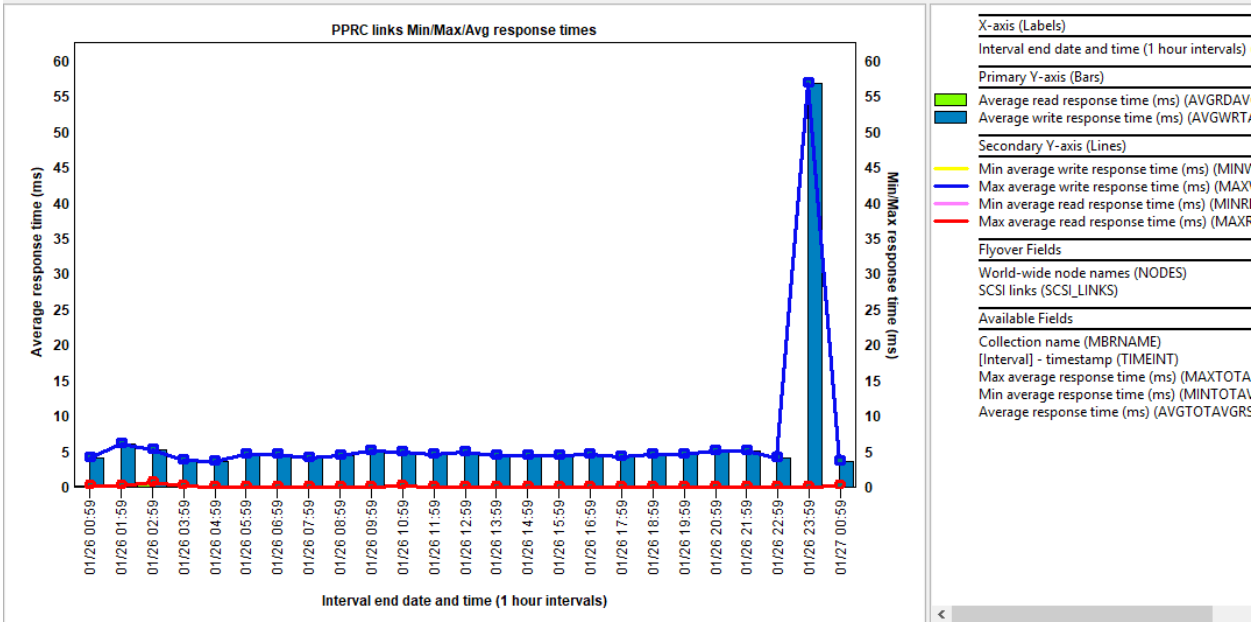
The primary Y-axis displays the SCSI links average read and write response times. The secondary Y-axis contains the min and max response times for each.



SCSI links Min/Max/Avg response times

9.21.3.12.2 PPRC links Min/Max/Avg response times

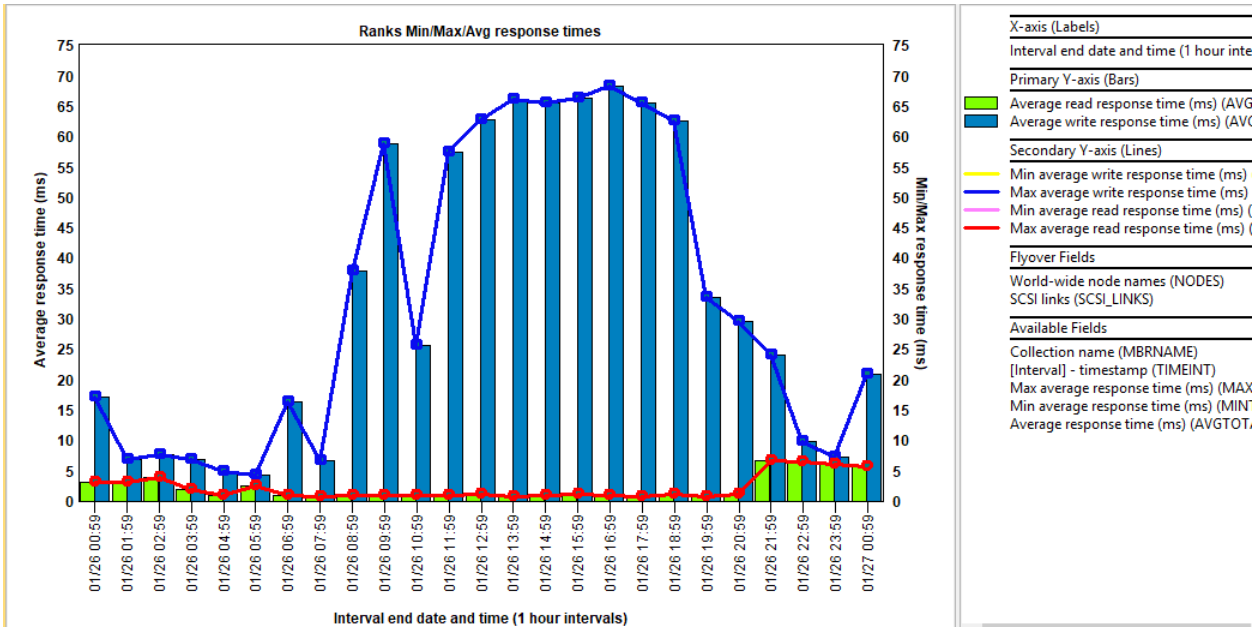
The primary Y-axis displays the PPRC links average read and write response times. The secondary Y-axis contains the min and max response times for each.



PPRC links Min/Max/Avg response times

9.21.3.12.3 Ranks Min/Max/Avg response times

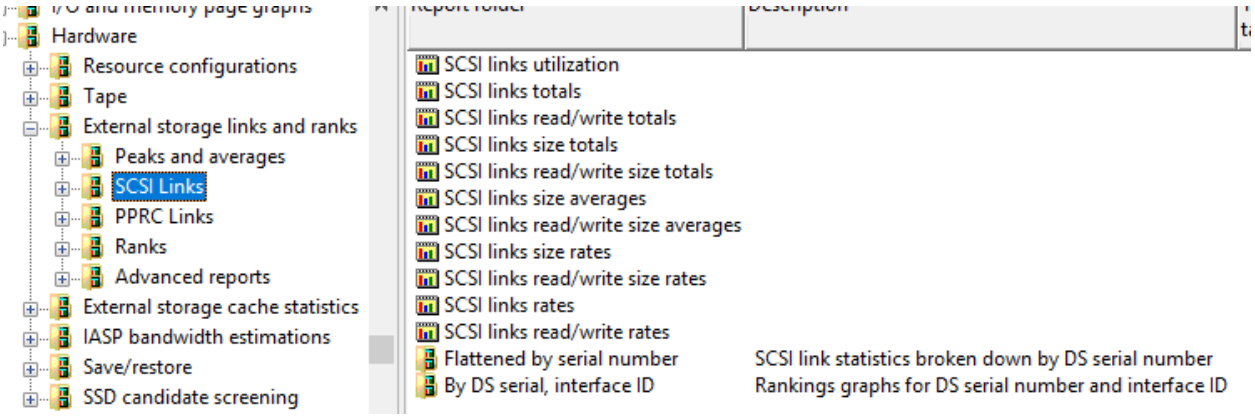
The primary Y-axis displays the ranks average read and write response times. The secondary Y-axis contains the min and max response times for each.



Ranks Min/Max/Avg response times

9.21.3.13 SCSI Links

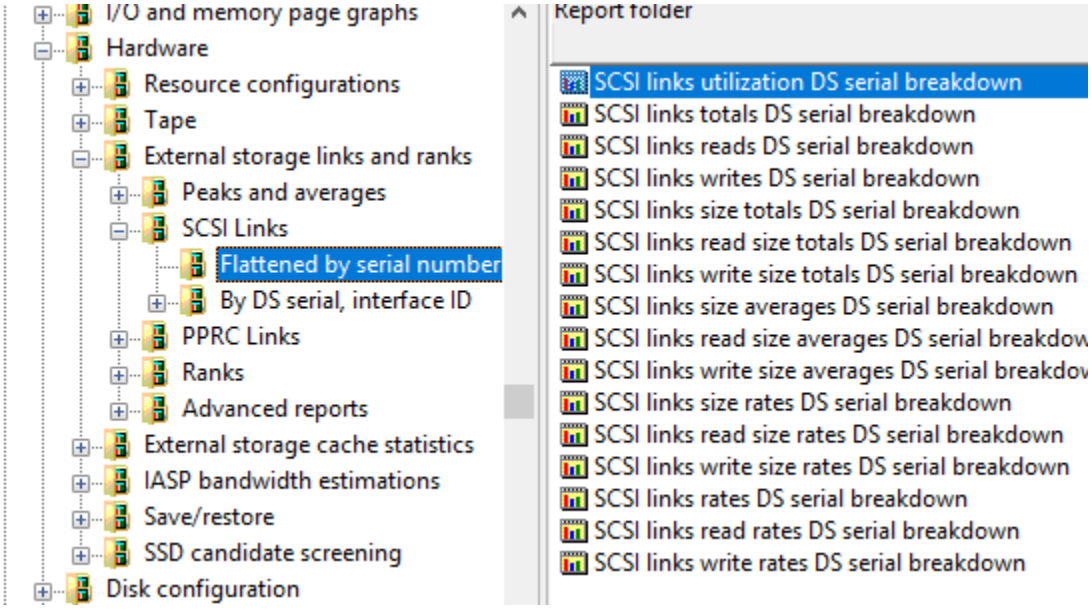
These graphs are duplicates of the initial graphs appearing in the External storage links and ranks folder. See that section for more information on them.



Hardware -> SCSI Links

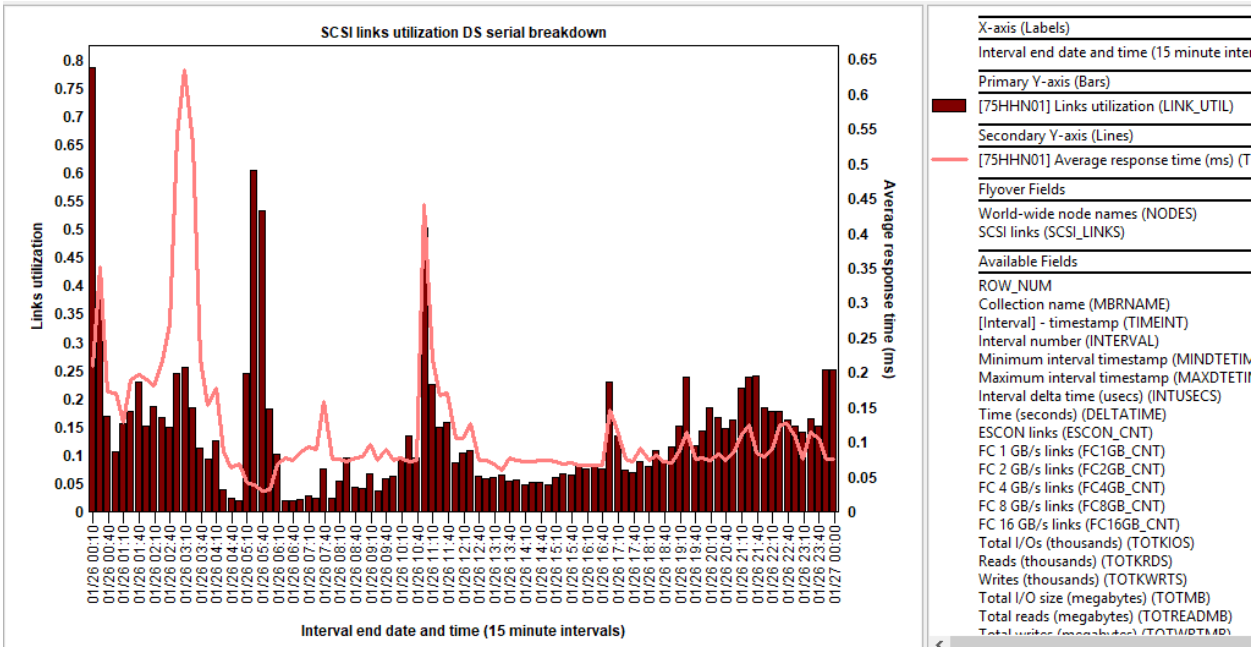
9.21.3.13.1 Flattened by serial number

This folder displays the SCSI links graphs but flattened by serial number. This means instead of summarizing all links together over time each link while be shown individually over time typically where each link has its own color.



Hardware -> External storage links and ranks -> SCSI Links -> Flattened by serial number

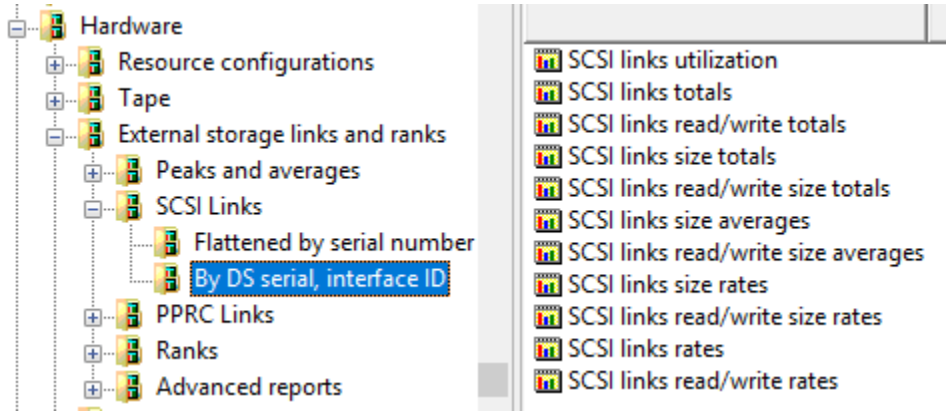
An example of this type of graph is shown below:



SCSI links utilization DS serial breakdown

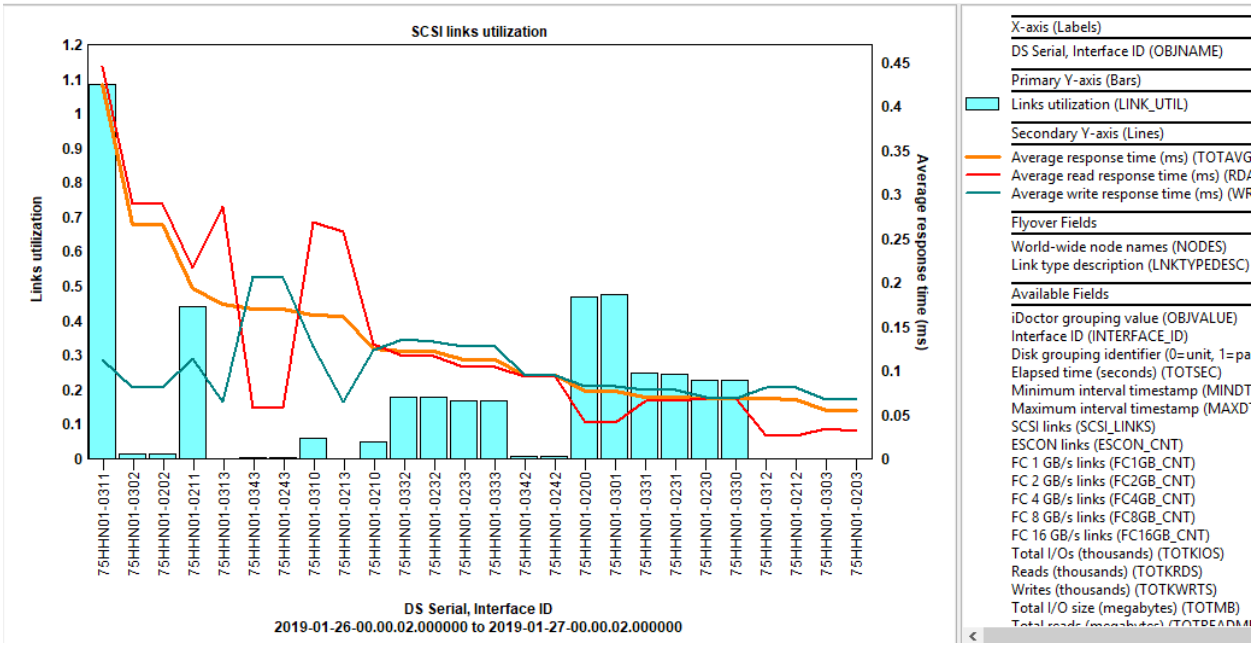
9.21.3.13.2 By DS serial, interface ID

These graphs rank the SCSI links by serial number and interface ID combination.



Hardware -> External storage links and ranks -> SCSI Links -> By DS serial, interface ID

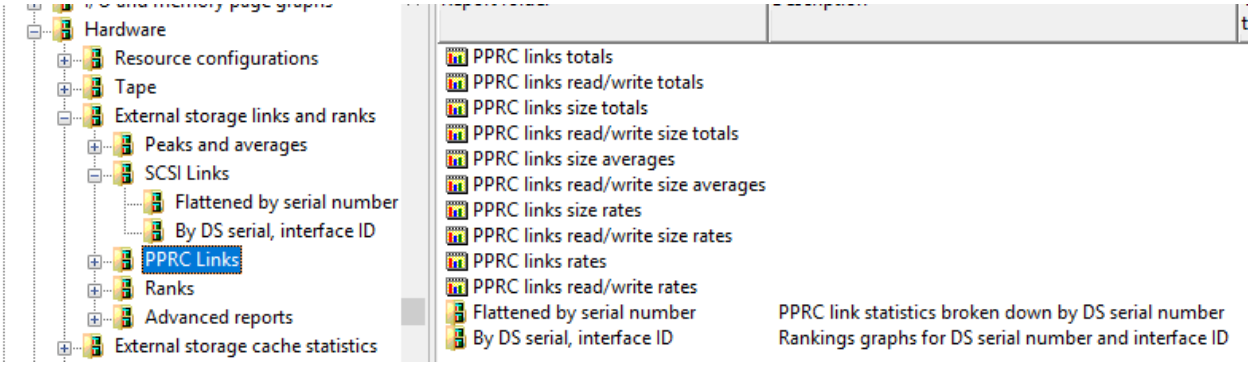
An example of this type of graph is shown below:



SCSI links utilization

9.21.3.14 PPRC Links

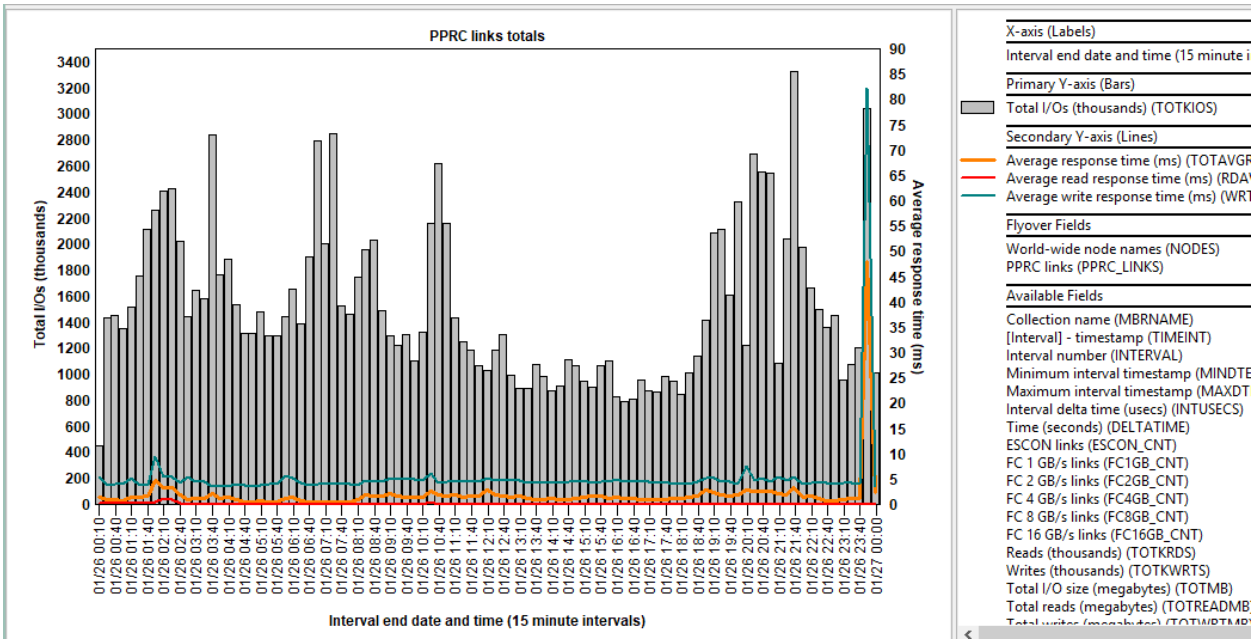
These graphs are just like the graphs for SCSI links but instead they only shown metrics for PPRC Links found in the data collection. See the SCSI Links section for more information on the metrics shown on these graphs.



Hardware -> PPRC Links

9.21.3.14.1 PPRC links totals

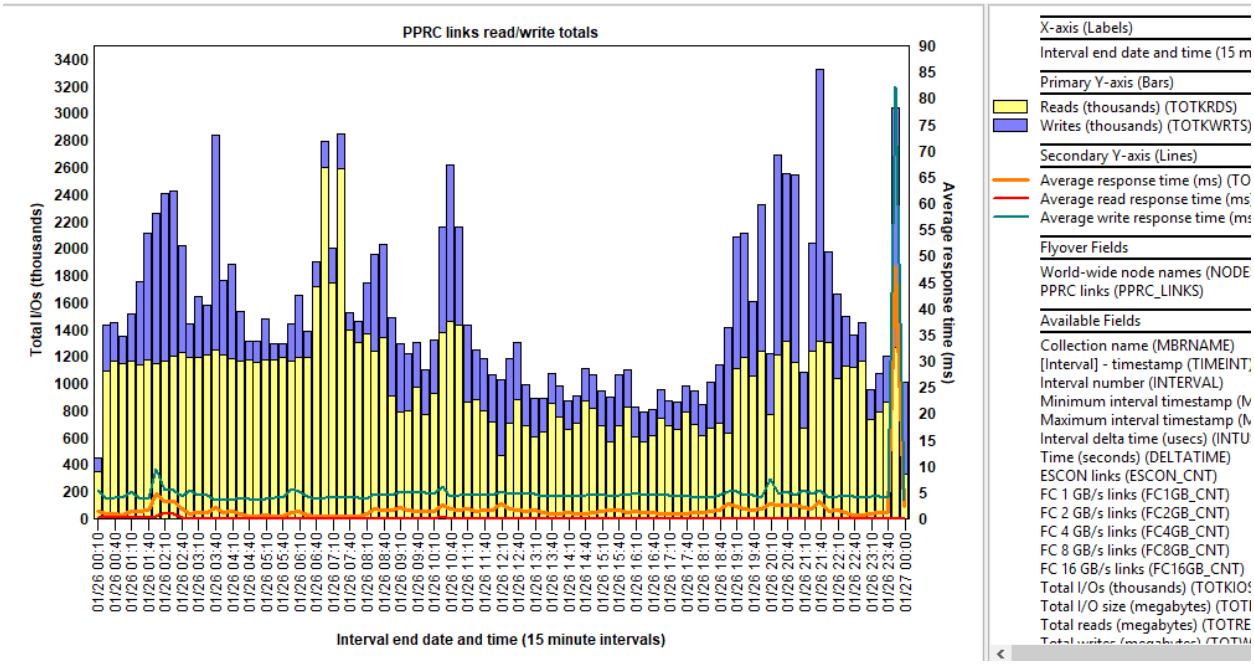
This graph displays the total I/Os (in thousands) for the PPRC links found in the collection. The secondary Y-axis displays the average response times (in milliseconds.)



PPRC links totals

9.21.3.14.2 PPRC links read/write totals

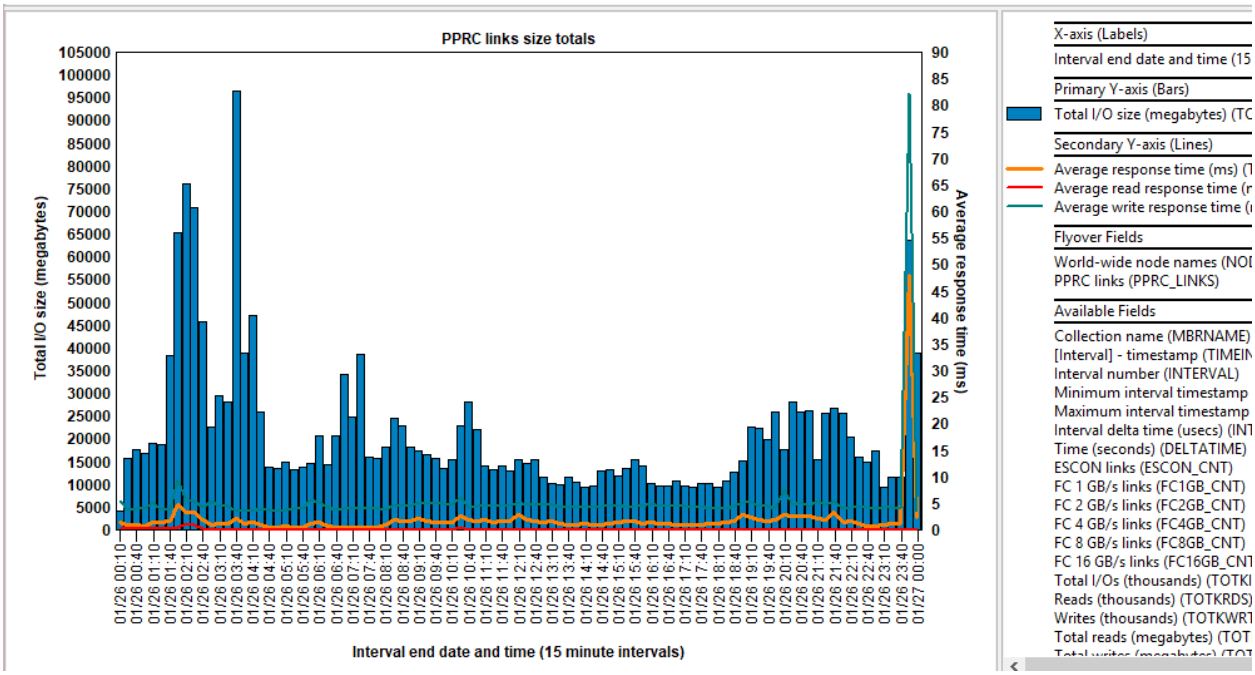
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



PPRC links read/write totals

9.21.3.14.3 PPRC links size totals

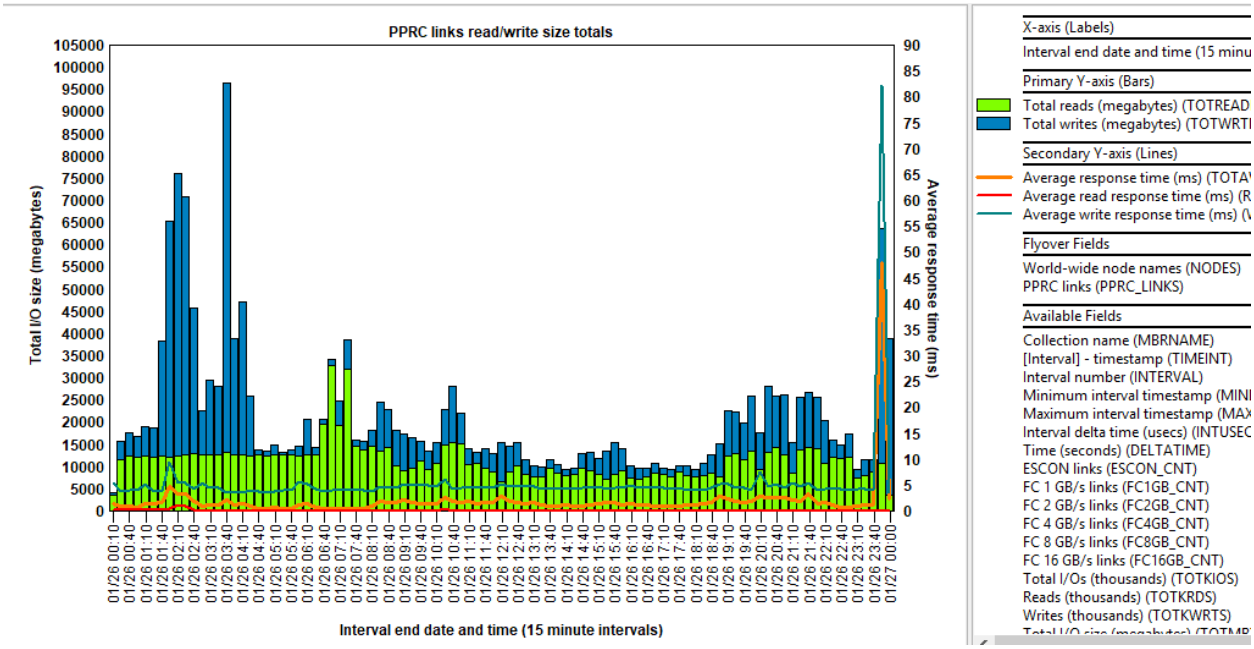
This graph displays the total size of all read and writes PPRC link IO operations over time (in megabytes.) The secondary Y-axis displays the average response times (in milliseconds.)



PPRC links size totals

9.21.3.14.4 PPRC links read/write size totals

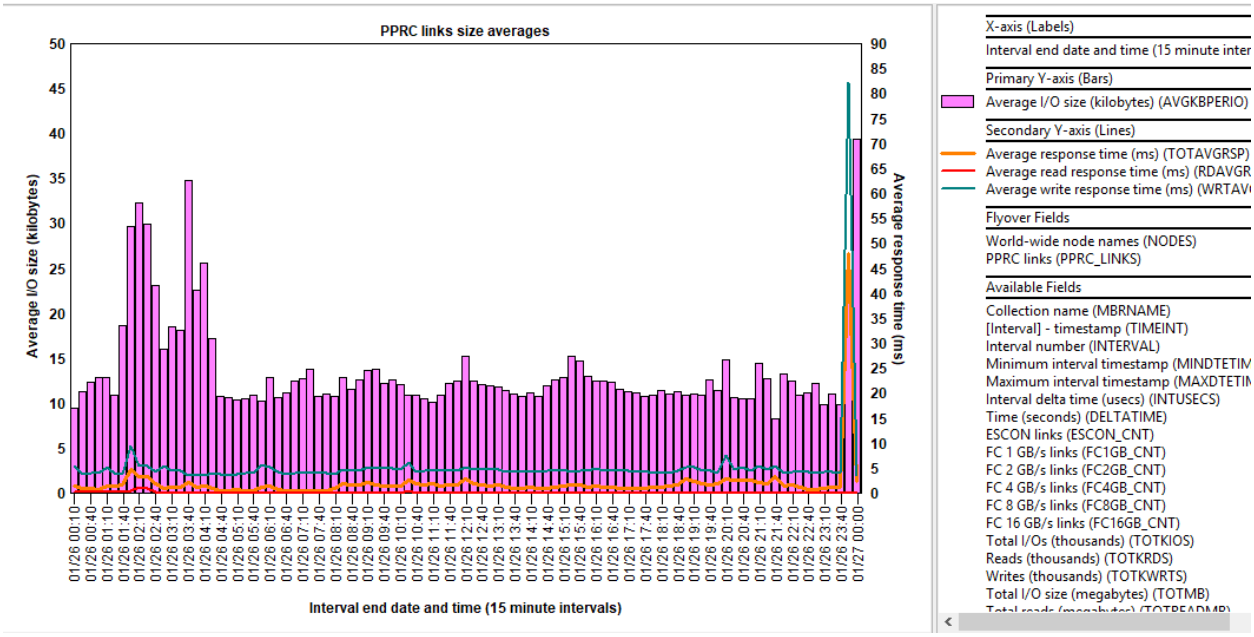
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



PPRC links read/write size totals

9.21.3.14.5 PPRC links size averages

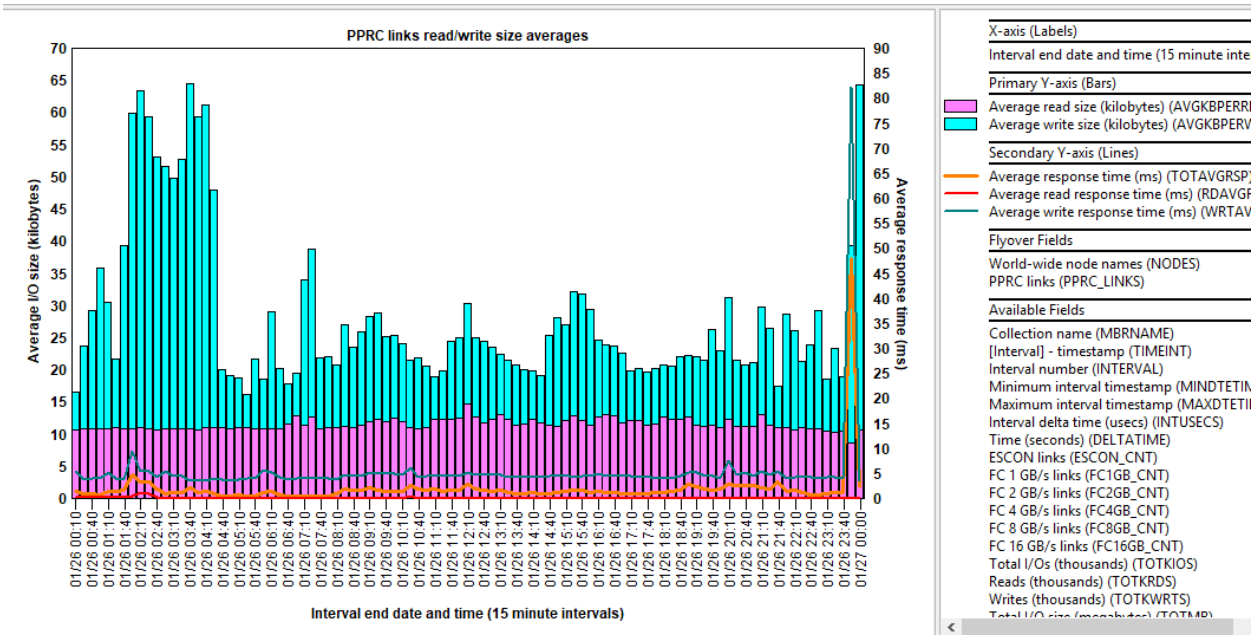
This graph shows the average I/O size for both reads and writes together over time.



PPRC links size averages

9.21.3.14.6 PPRC links read/write size averages

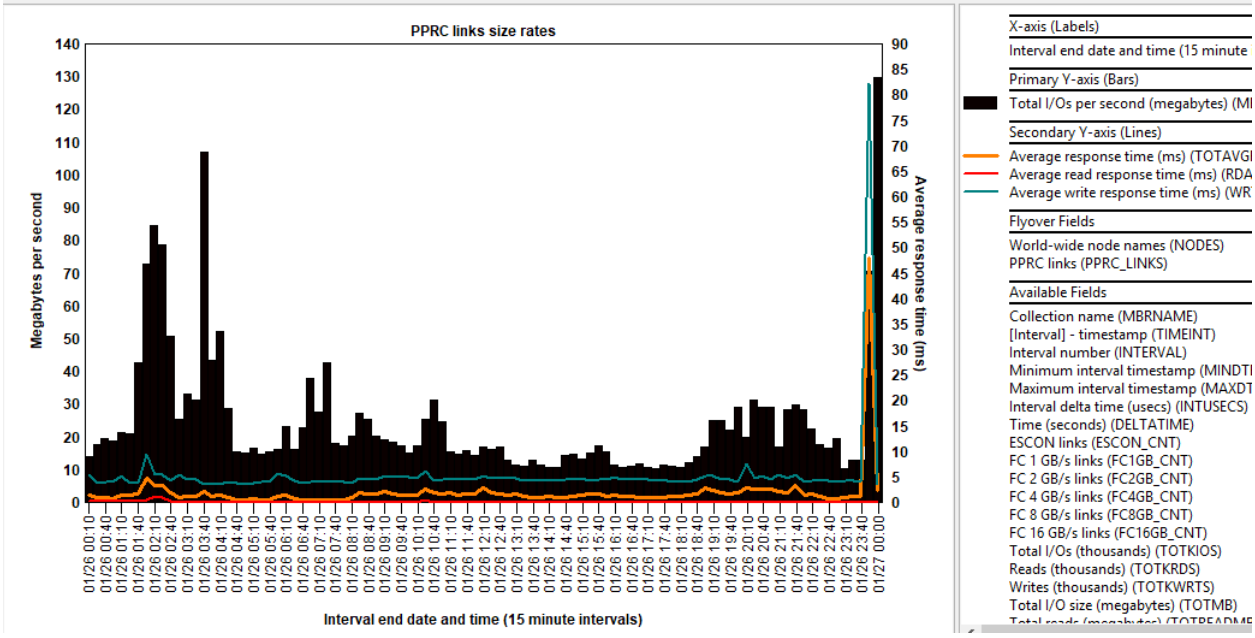
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



PPRC links read/write size averages

9.21.3.14.7 PPRC links size rates

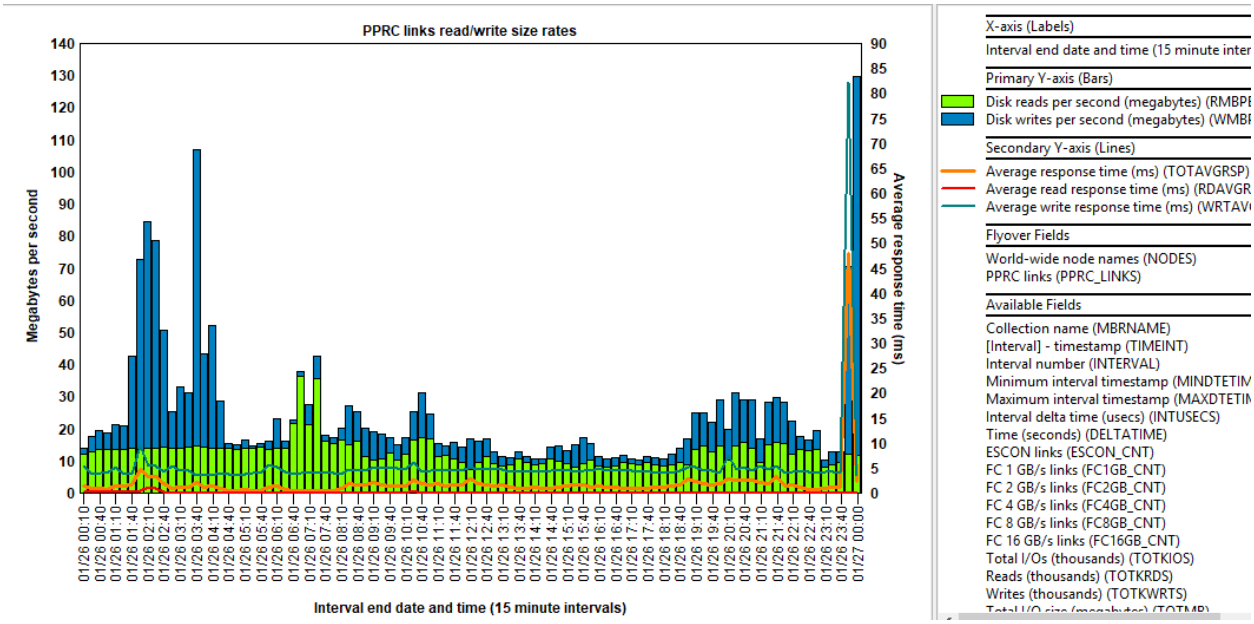
This graph displays the total I/O per second in megabytes over time for the SCSI links found in the collection.



PPRC links size rates

9.21.3.14.8 PPRC links read/write size rates

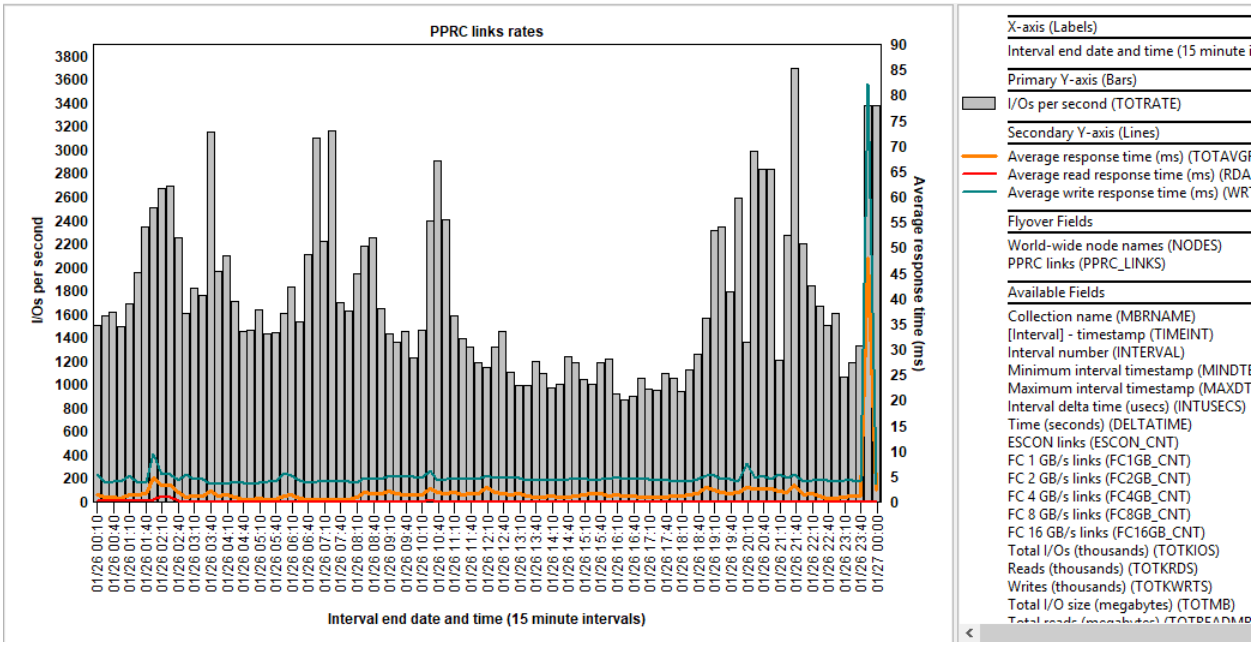
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



PPRC links read/write size rates

9.21.3.14.9 PPRC links rates

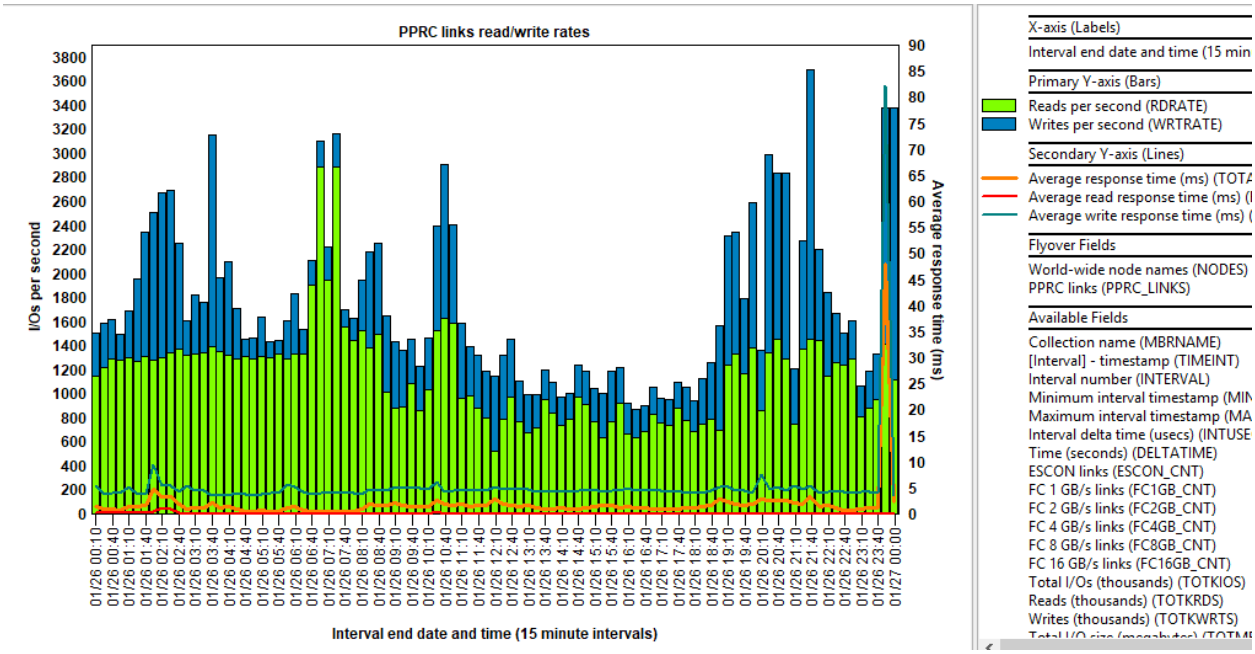
This graph displays the I/Os per second over time for the PPRC links found in the collection.



PPRC links rates

9.21.3.14.10 PPRC links read/write rates

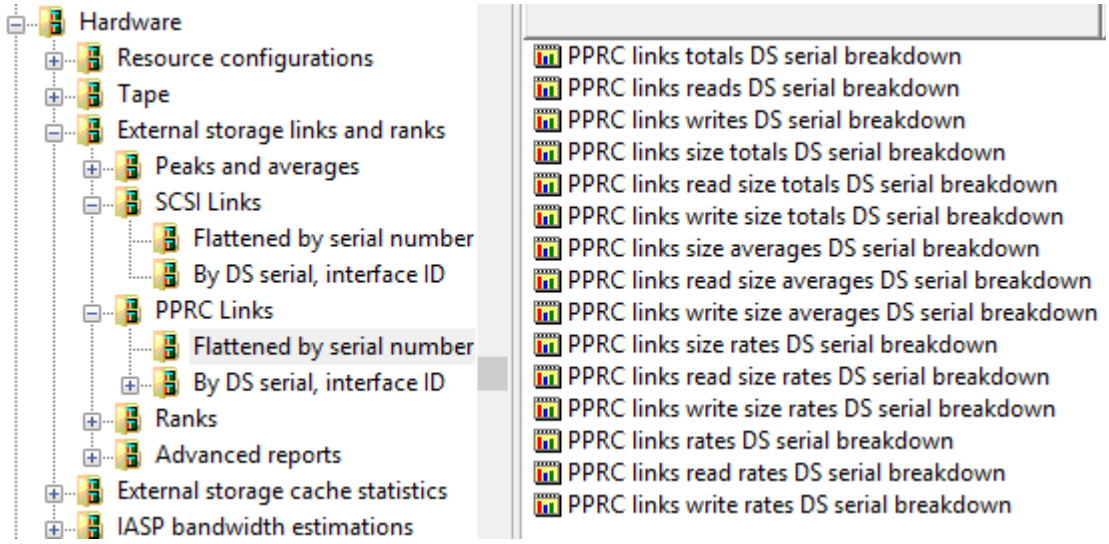
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



PPRC links read/write rates

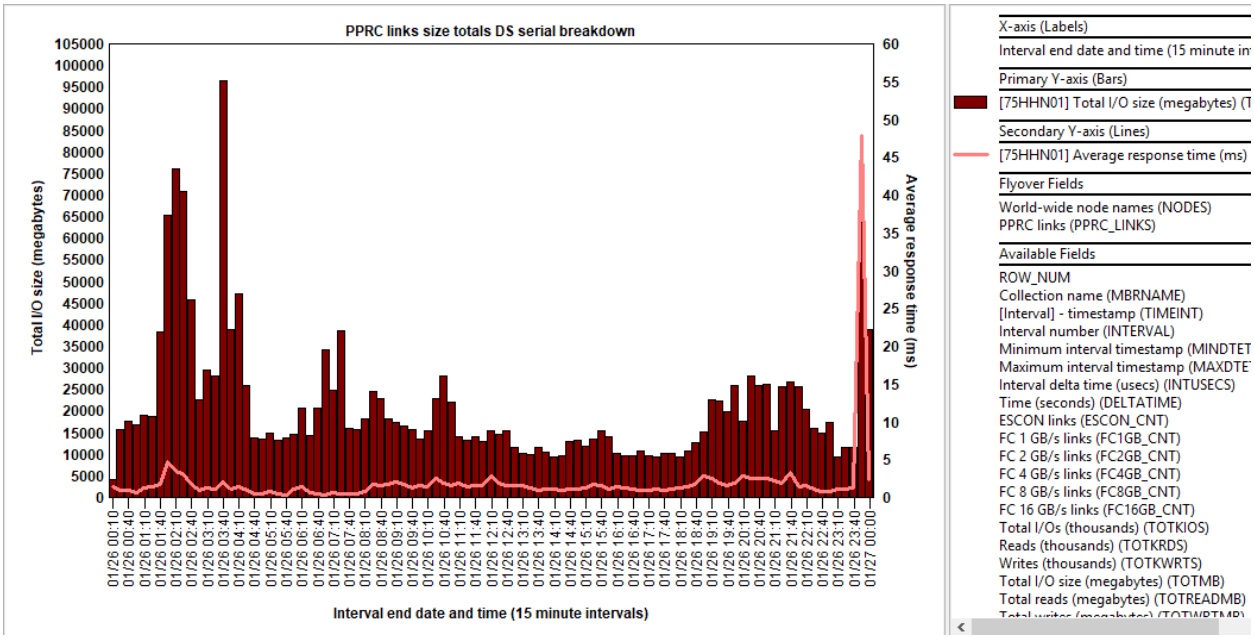
9.21.3.14.11 Flattened by serial number

This folder displays the PPRC links graphs but flattened by serial number. This means instead of summarizing all links together over time each link while be shown individually over time typically where each link has its own color.



Hardware -> External storage links and ranks -> PPRC Links -> Flattened by serial number

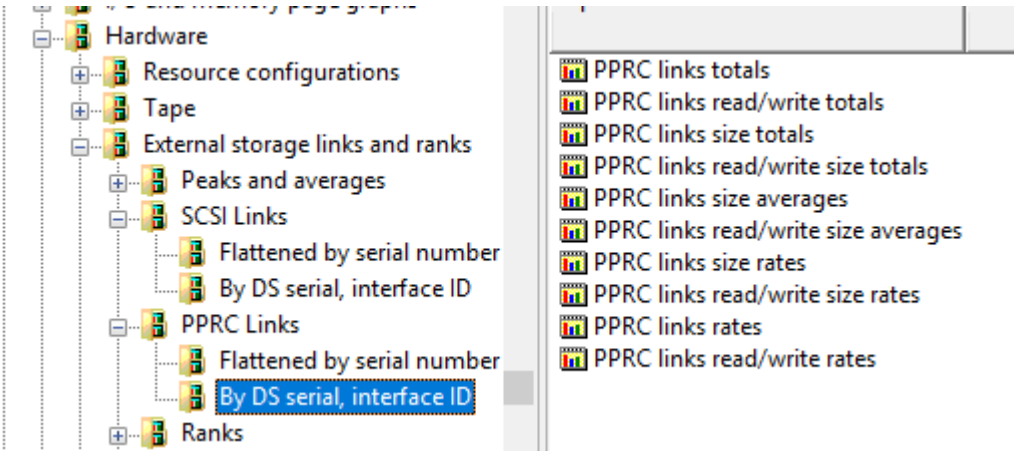
An example of this type of graph is shown below:



PPRC links size totals DS serial breakdown

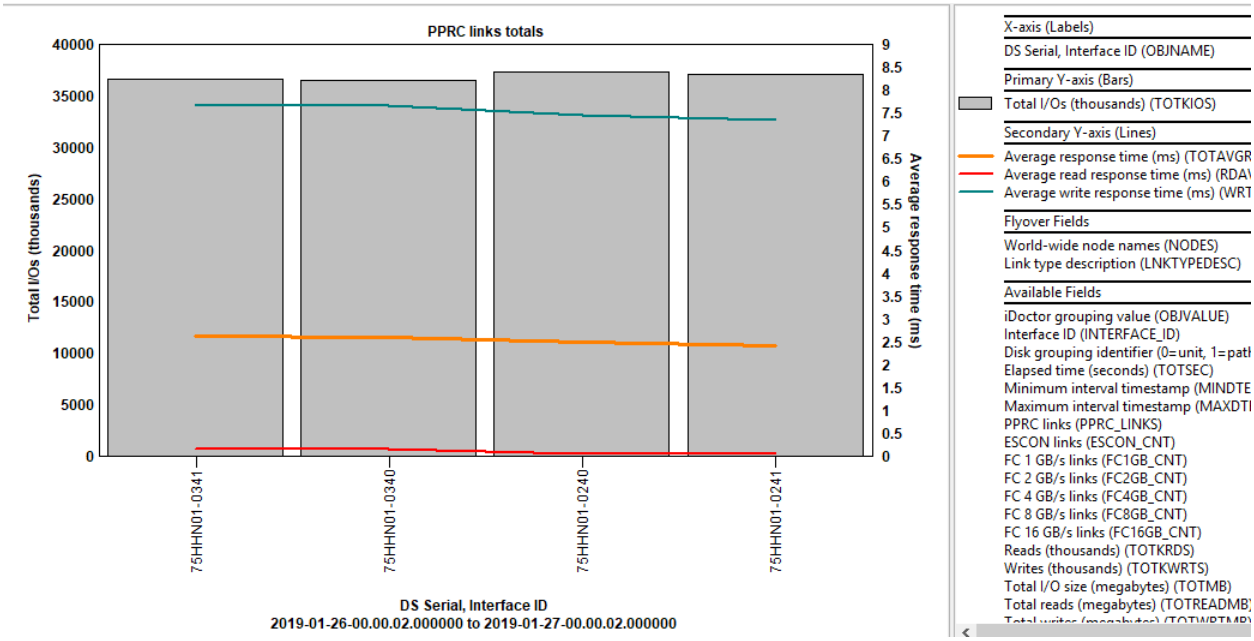
9.21.3.14.12 By DS serial, interface ID

These graphs rank the PPRC links by serial number and interface ID combination.



Hardware -> External storage links and ranks -> PPRC Links -> By DS serial, interface ID

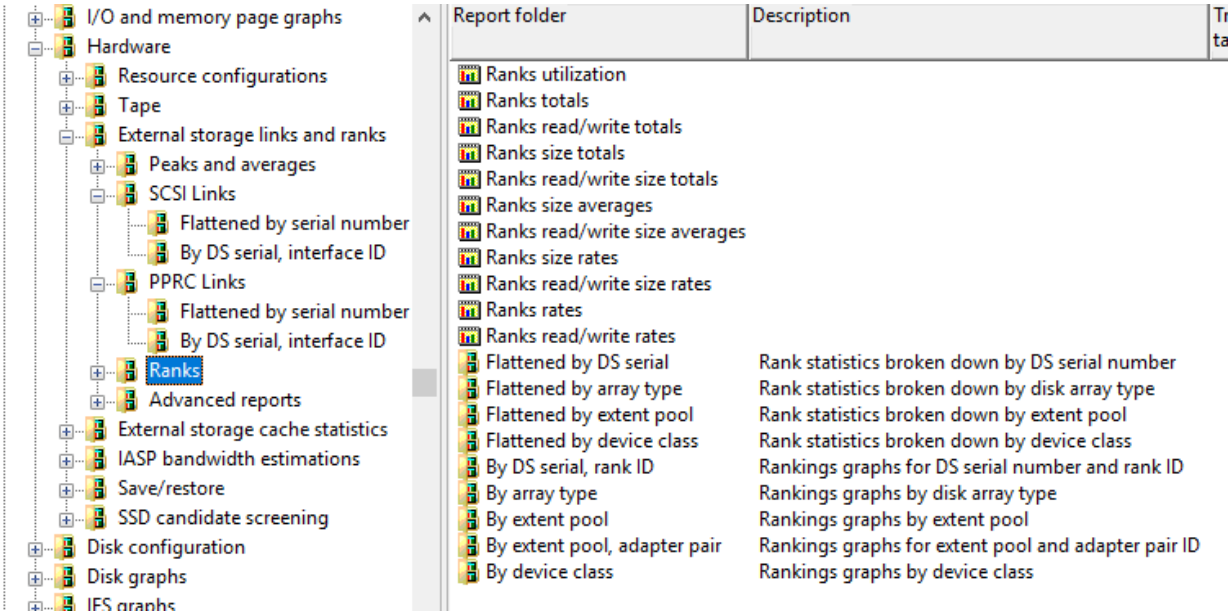
An example of this type of graph is shown below:



PPRC links totals

9.21.3.15 Ranks

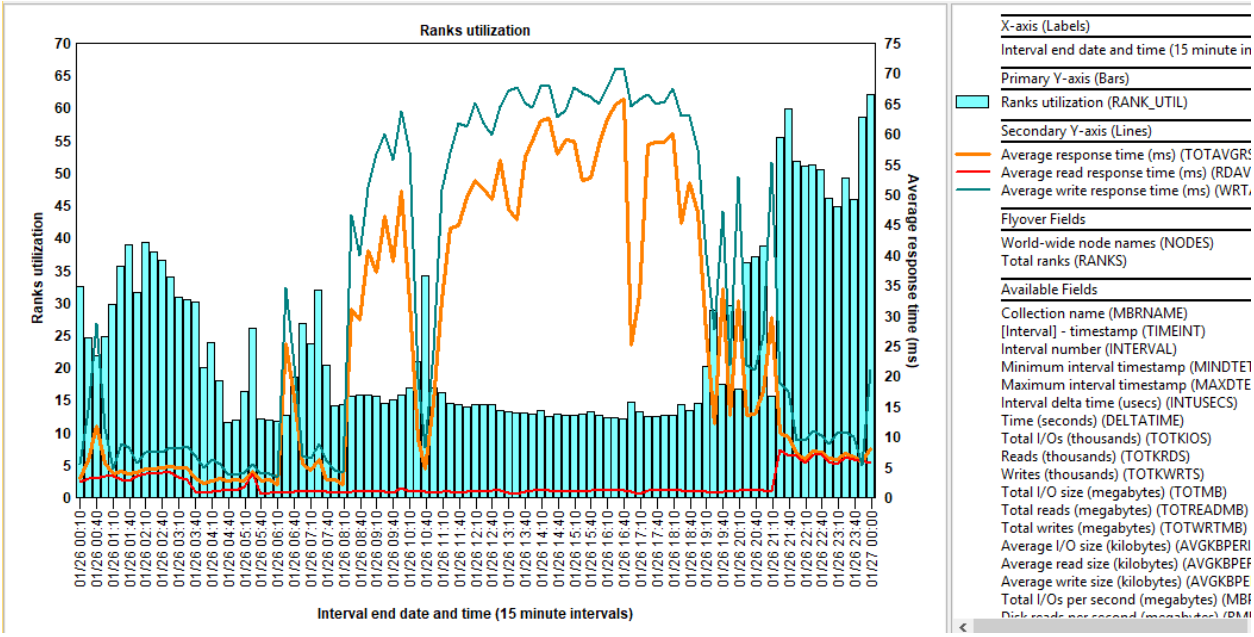
This folder contains graphs over the Ranks found on the external storage systems found in the data collection. Several types of graphs to either flatten the data over time or rank the ranks in the collection are available.



Hardware -> External storage links and ranks -> Ranks

9.21.3.15.1 Ranks utilization

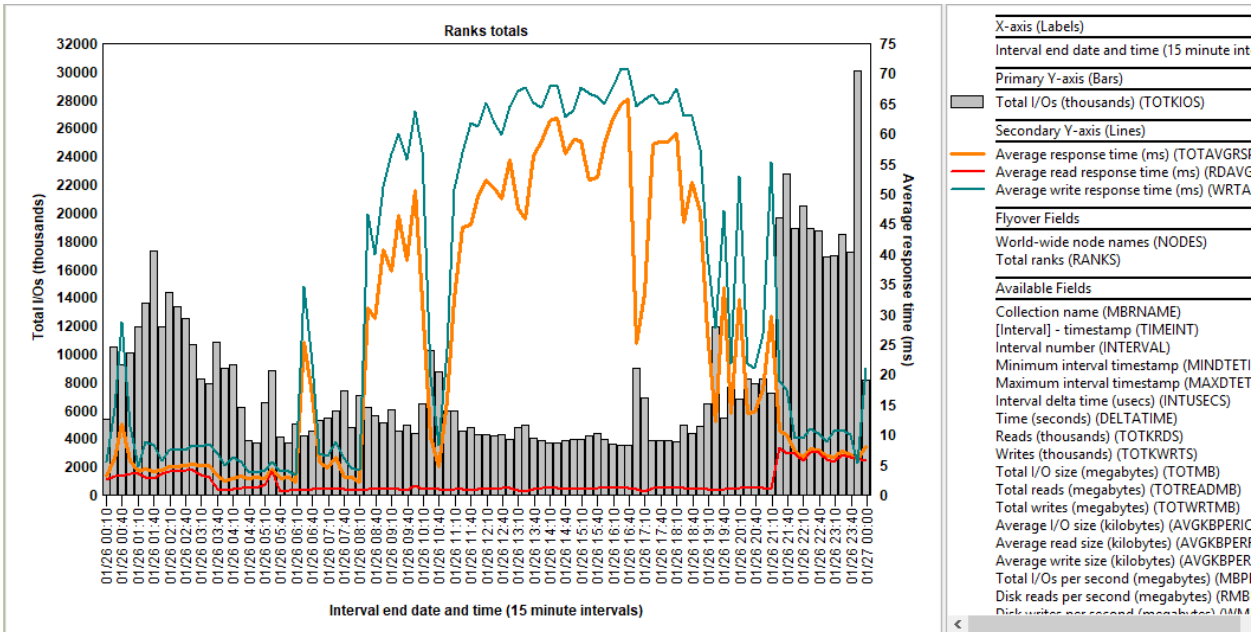
This graph displays the average rank utilization for the external storage devices (ranks) found in the collection. The secondary Y-axis displays the average response times (in milliseconds.)



Ranks utilization

9.21.3.15.2 Ranks totals

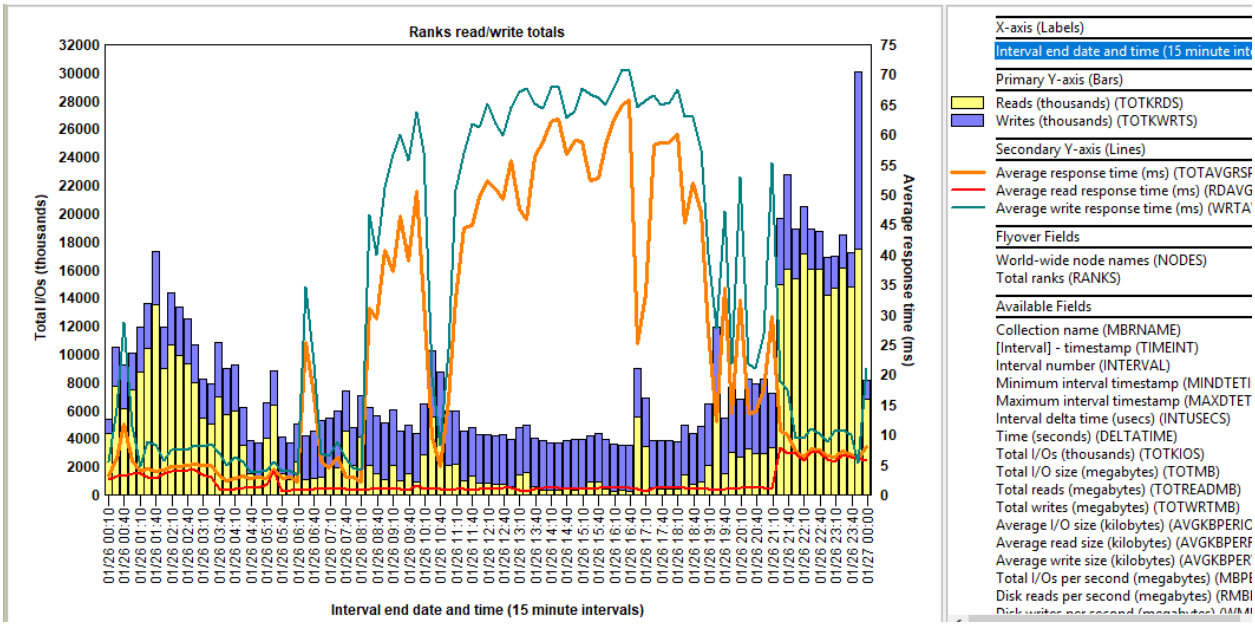
This graph displays the total I/Os (in thousands) for the ranks found in the collection. The secondary Y-axis displays the average response times (in milliseconds.)



Ranks totals

9.21.3.15.3 Ranks read/write totals

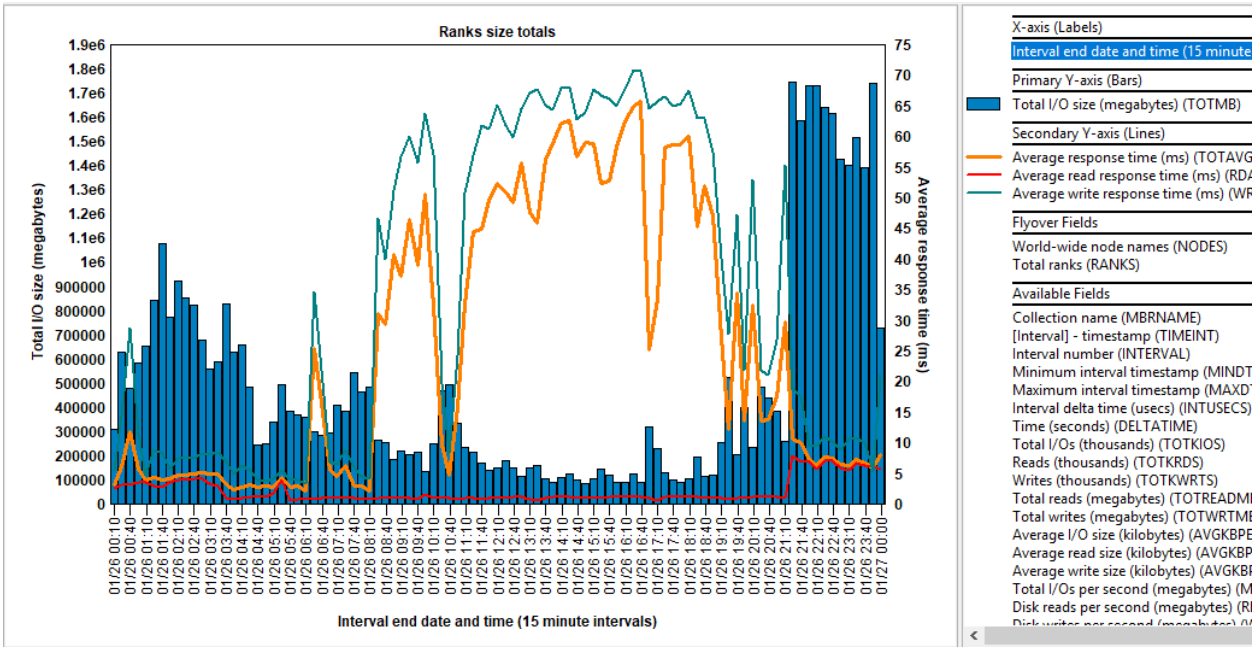
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



Ranks read/write totals

9.21.3.15.4 Ranks size totals

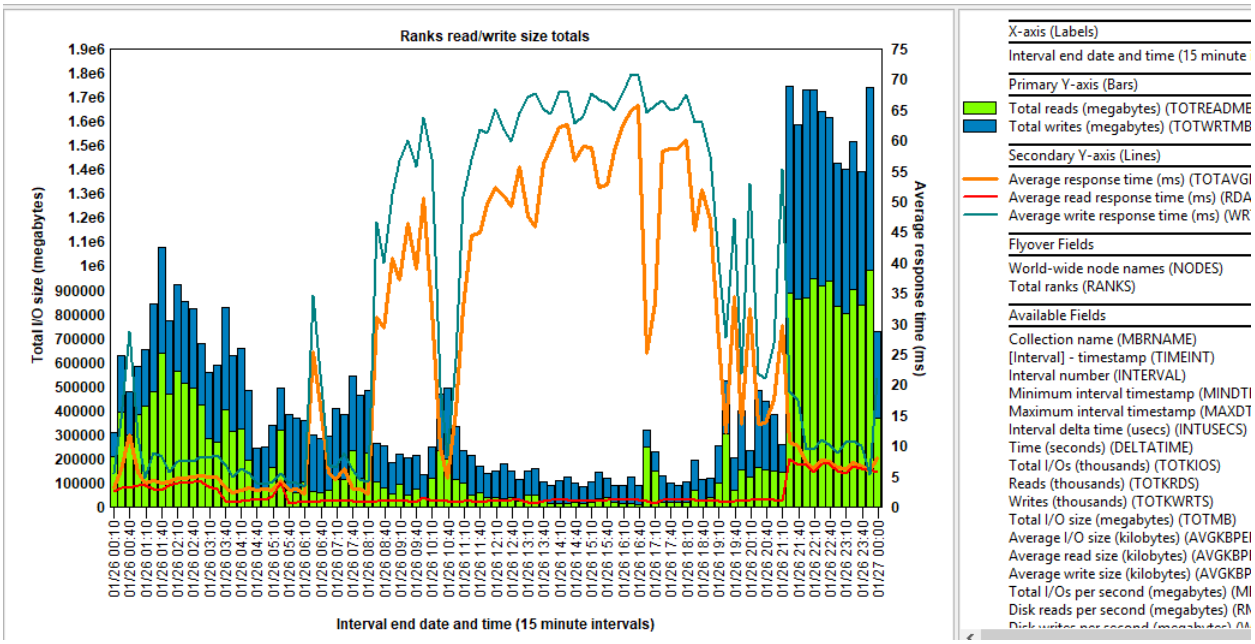
This graph displays the total size of all read and writes rank IO operations over time (in megabytes.) The secondary Y-axis displays the average response times (in milliseconds.)



Ranks size totals

9.21.3.15.5 Ranks read/write size totals

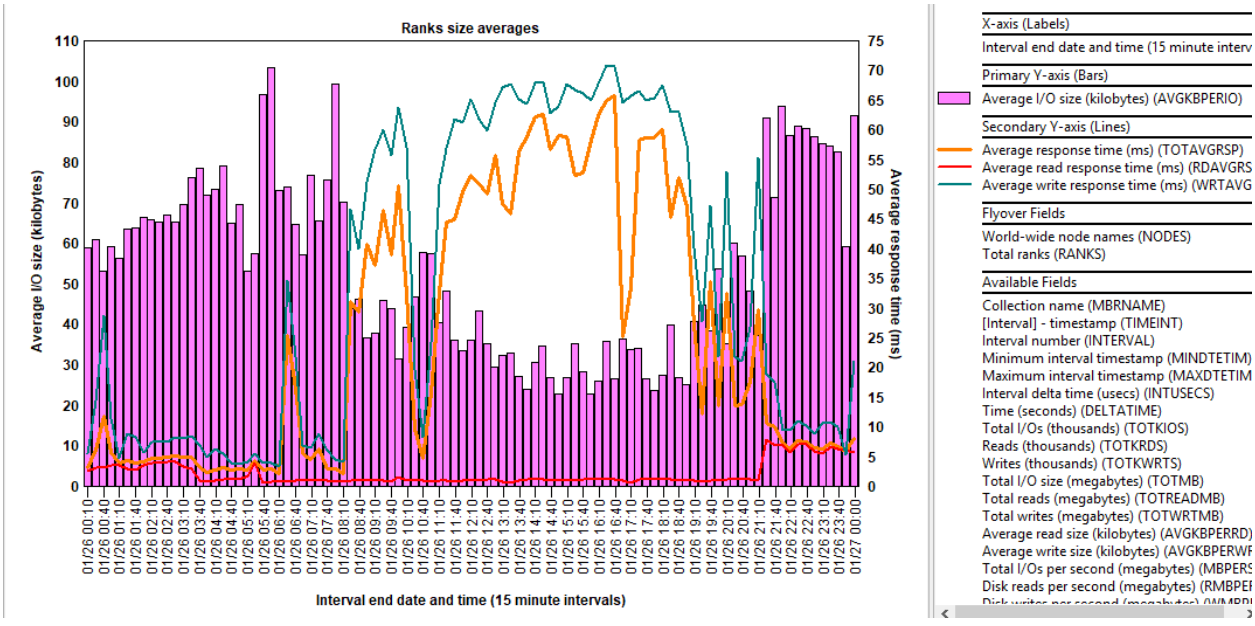
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



Ranks read/write size totals

9.21.3.15.6 Ranks size averages

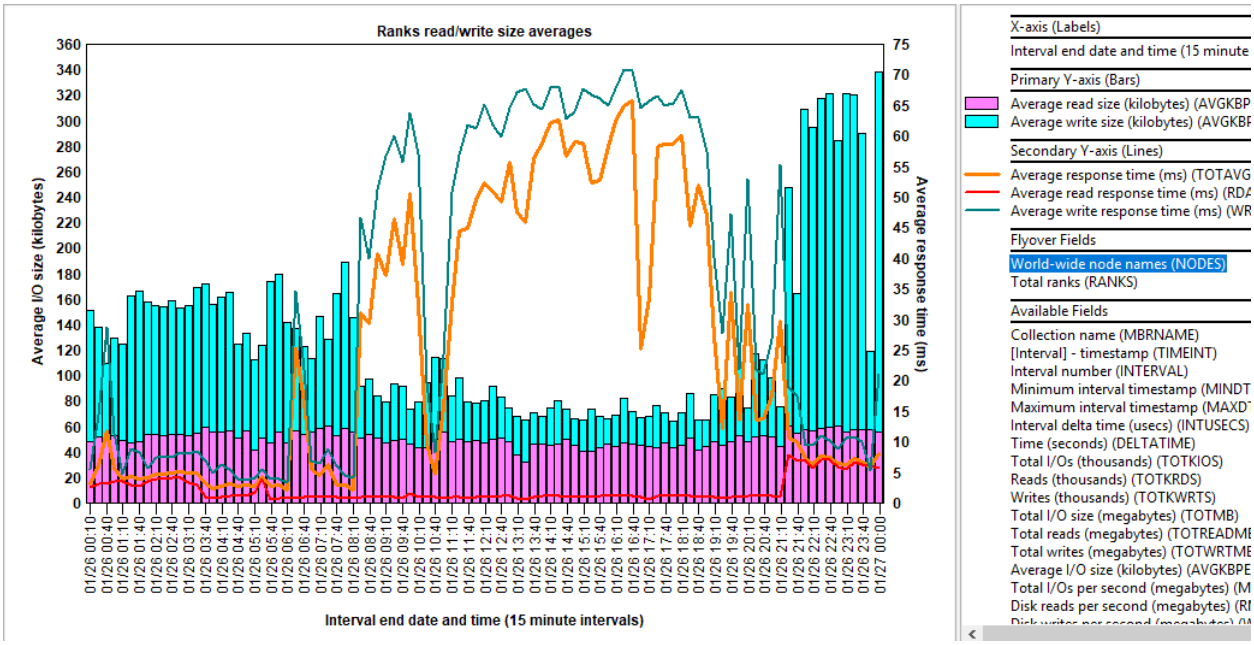
This graph shows the average I/O size for both reads and writes together over time.



Ranks size averages

9.21.3.15.7 Ranks read/write size averages

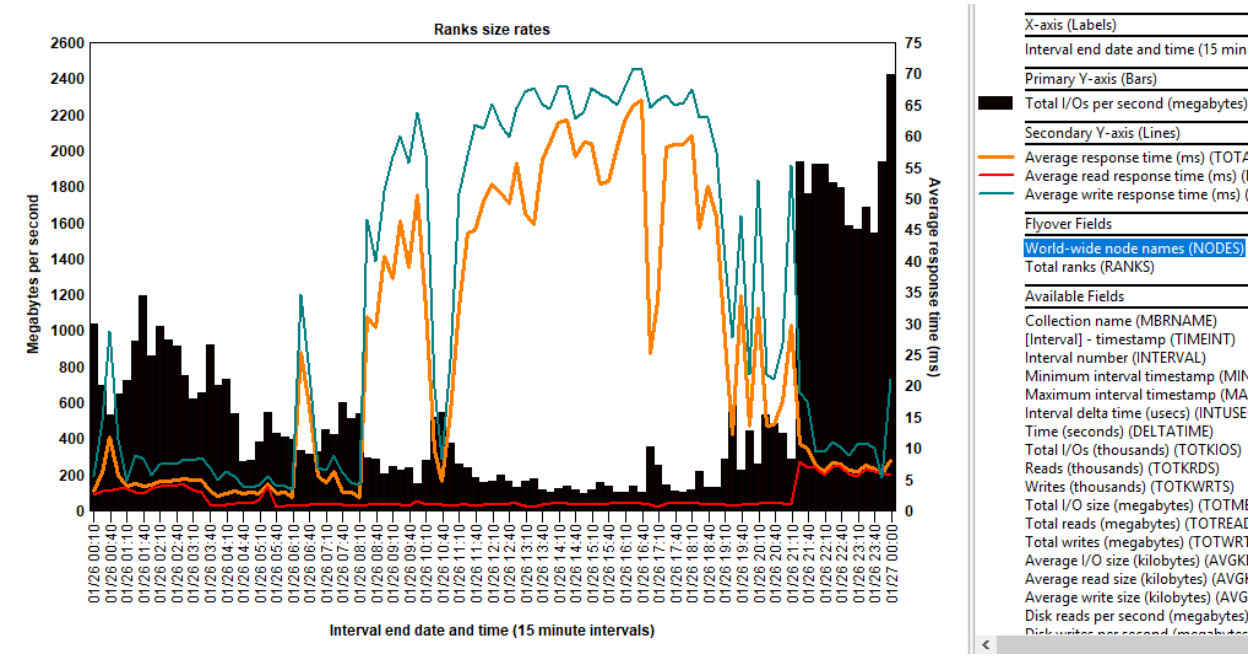
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



Ranks read/write size averages

9.21.3.15.8 Ranks size rates

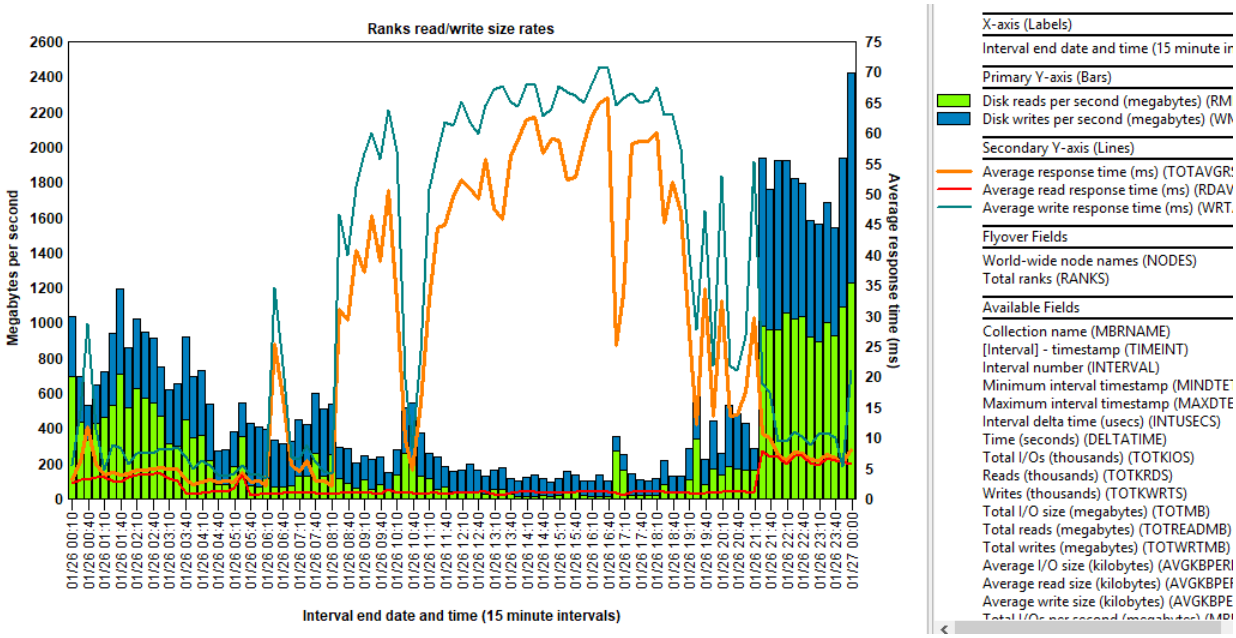
This graph displays the total I/O per second in megabytes over time for the ranks found in the collection.



Ranks size rates

9.21.3.15.9 Ranks read/write size rates

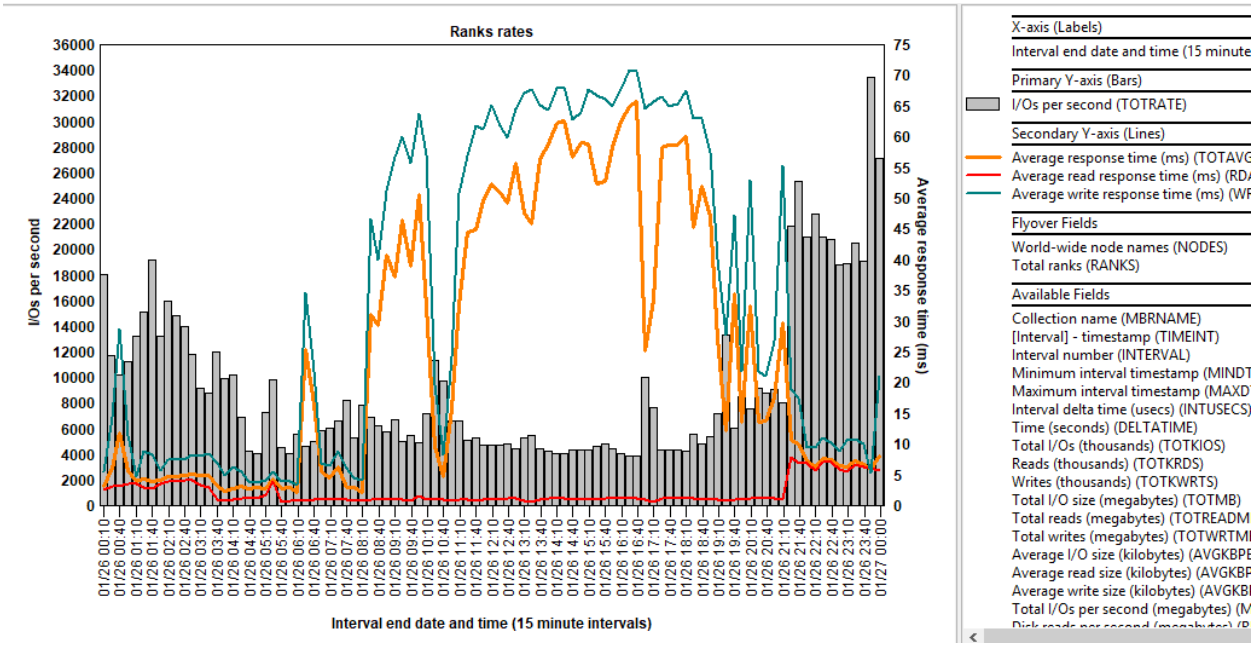
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



Ranks read/write size rates

9.21.3.15.10 Ranks rates

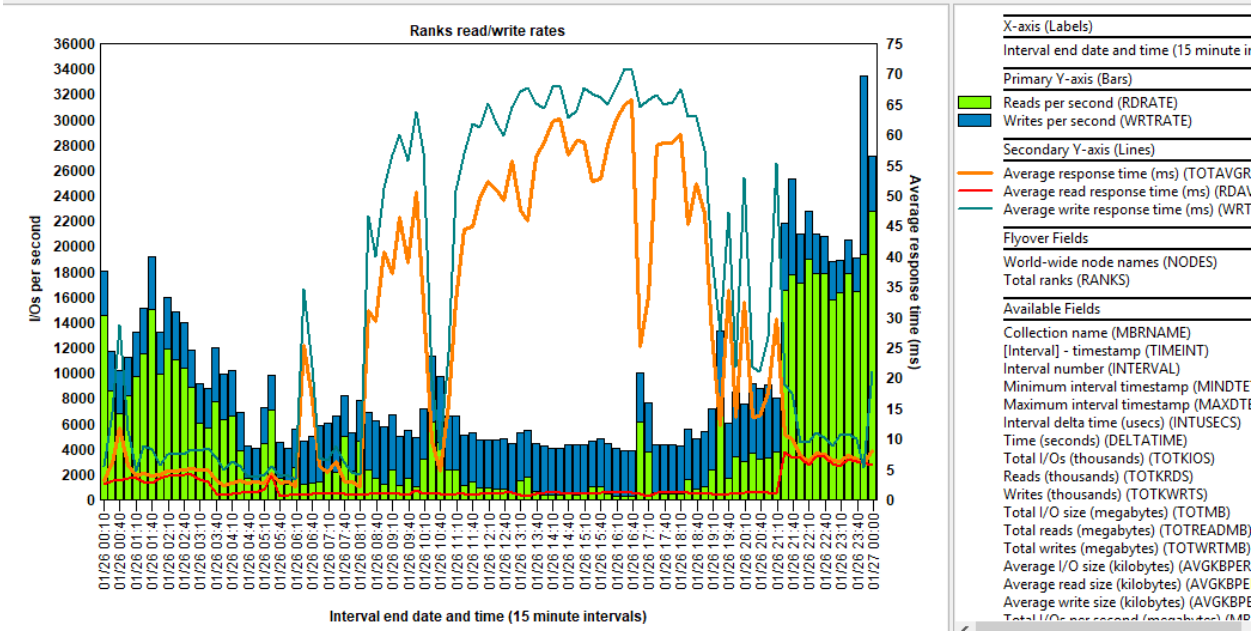
This graph displays the I/Os per second over time for the ranks found in the collection.



Ranks rates

9.21.3.15.11 Ranks read/write rates

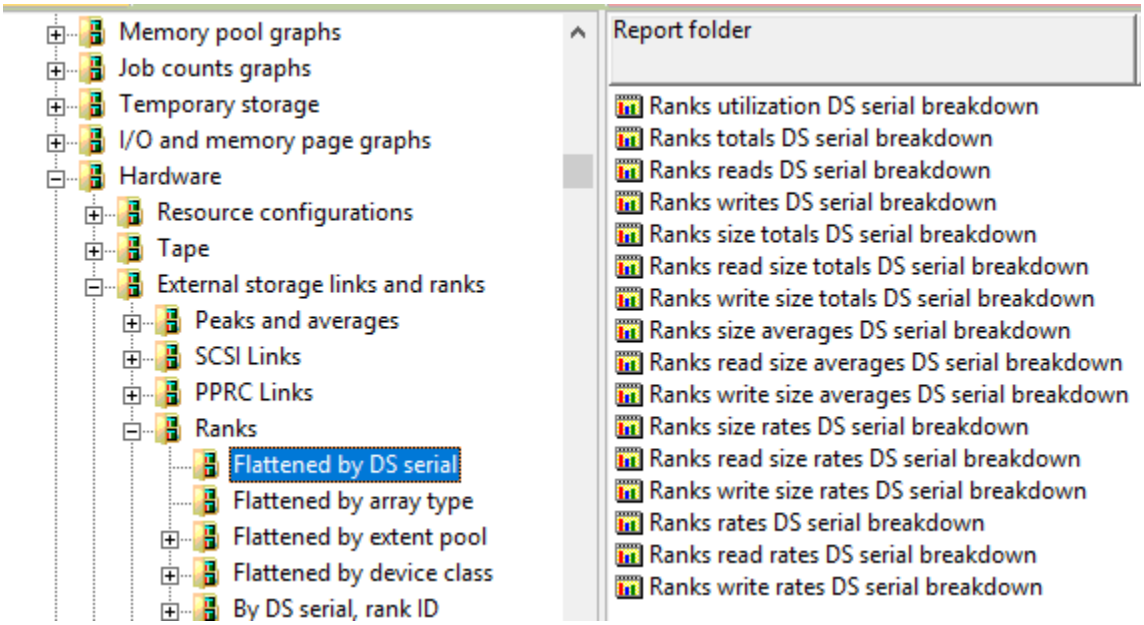
This graph is the same as the previous one except breaks out the reads and writes on the primary Y-axis.



Ranks read/write rates

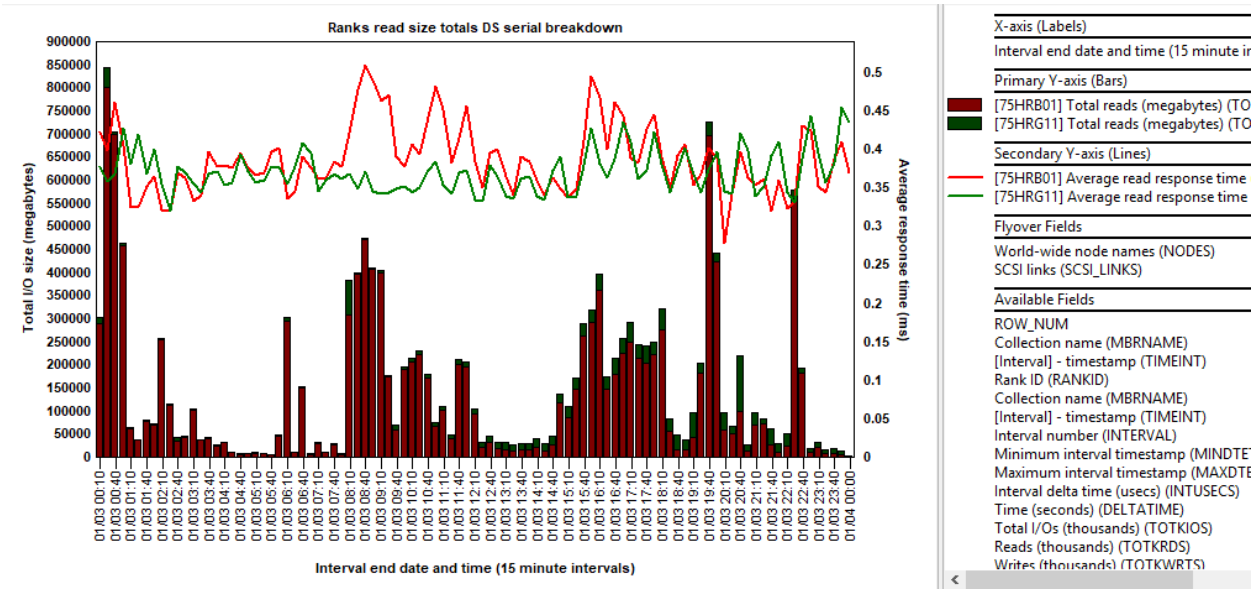
9.21.3.15.12 Flattened by DS serial

These graphs are used to graph the rank statistics by serial number over time indvidually. Each ds serial number wil have a different color and typically only show 1 metric per Y-axis on each graph.



Hardware -> External storage links and ranks -> Ranks -> Flattened by DS serial

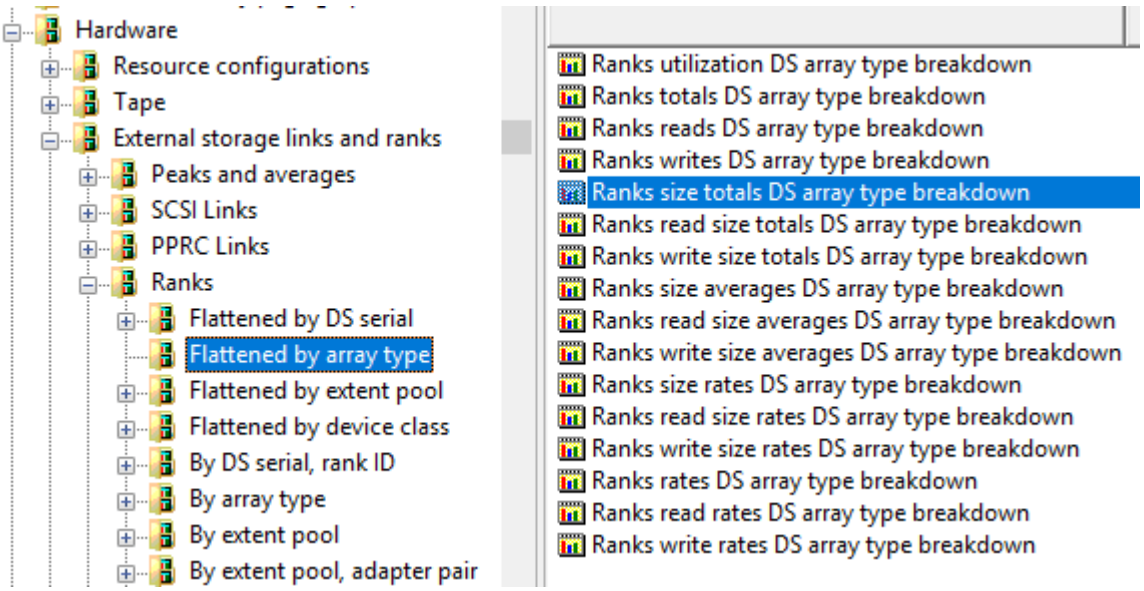
An example of this type of graph is shown below:



Ranks read size total DS serial breakdown

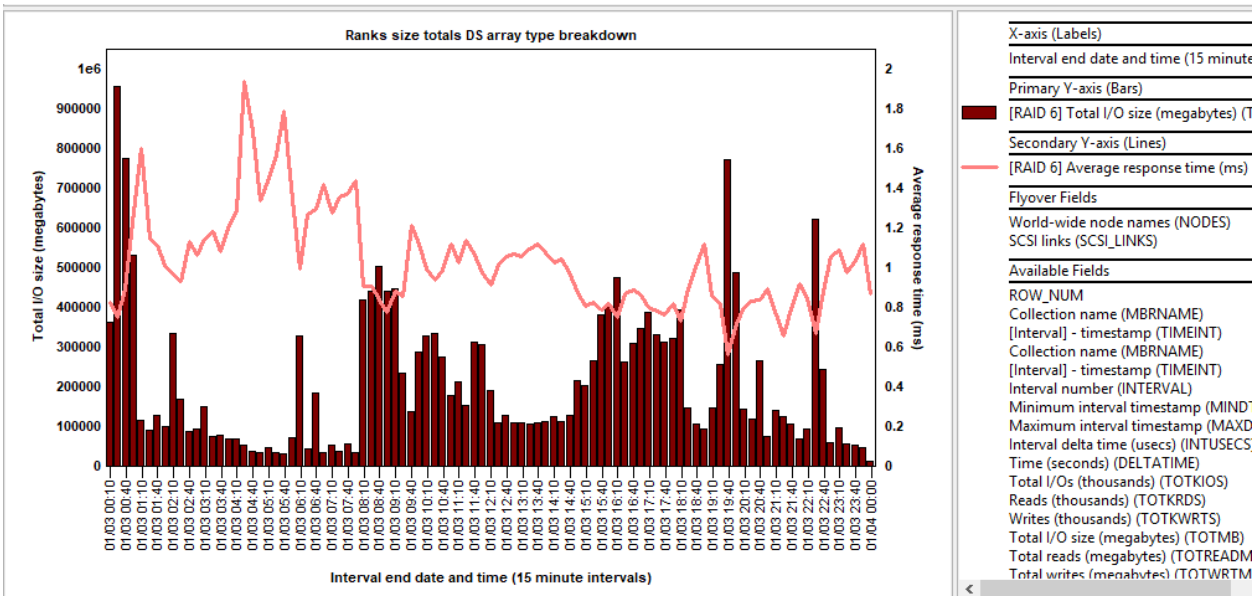
9.21.3.15.13 Flattened by array type

These graphs are used to graph the rank statistics by array type over time indvidually. Each array type will have a different color and typically only show 1 metric per Y-axis on each graph.



Hardware -> External storage links and ranks -> Ranks -> Flattened by array type

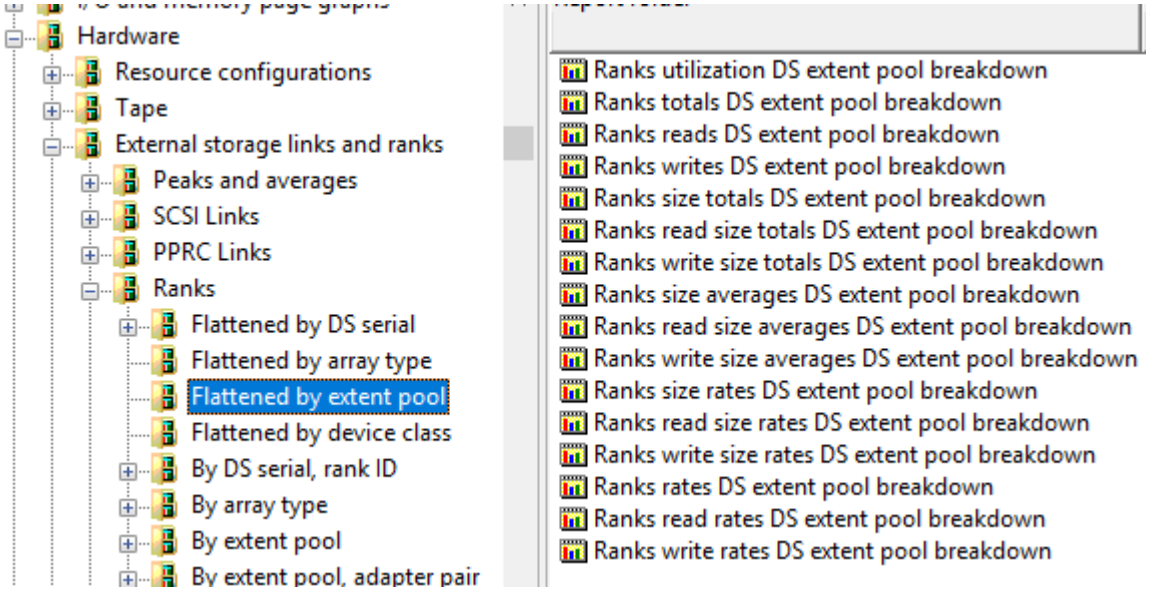
An example of this type of graph is shown below:



Ranks size totals DS array type breakdown

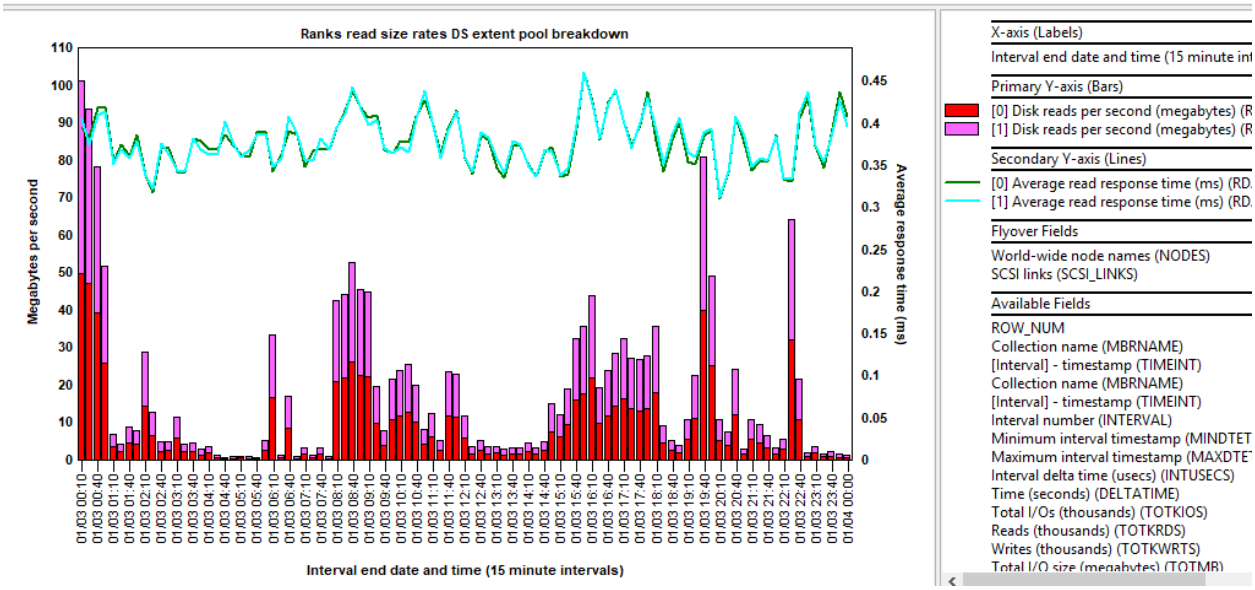
9.21.3.15.14 Flattened by extent pool

These graphs are used to graph the rank statistics by extent pool over time indvidually. Each extent pool will have a different color and typically only show 1 metric per Y-axis on each graph.



Hardware -> External storage links and ranks -> Ranks -> Flattened by extent pool

An example of this type of graph is shown below:



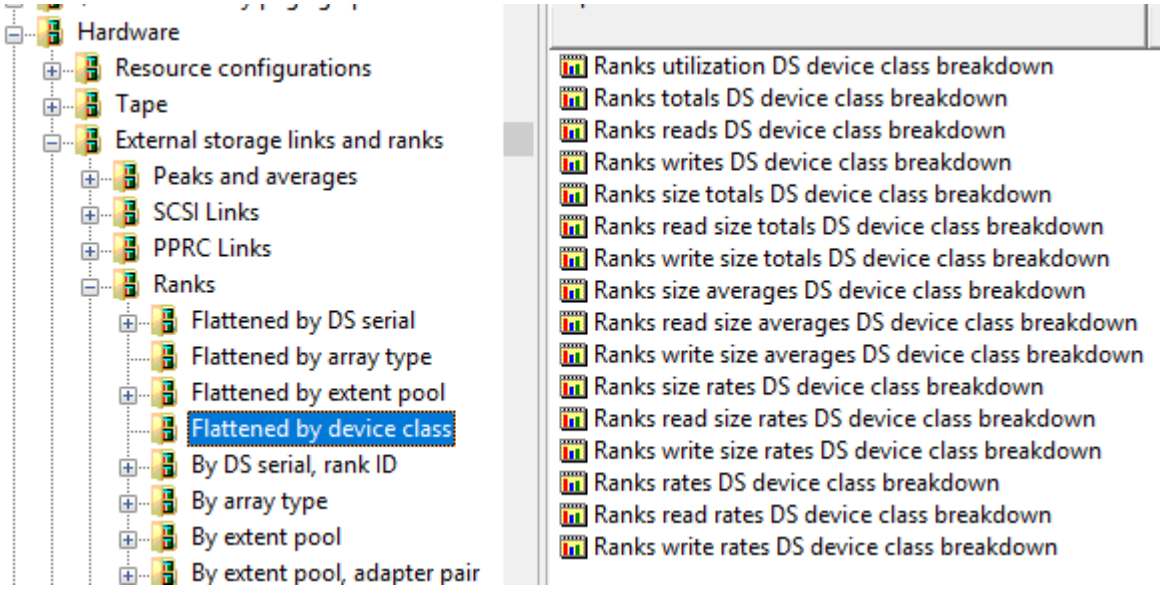
Ranks read size rates DS extent pool breakdown

9.21.3.15.15 Flattened by device class

These graphs are used to graph the rank statistics by device class over time indvidually. Each device class will have a different color and typically only show 1 metric per Y-axis on each graph.

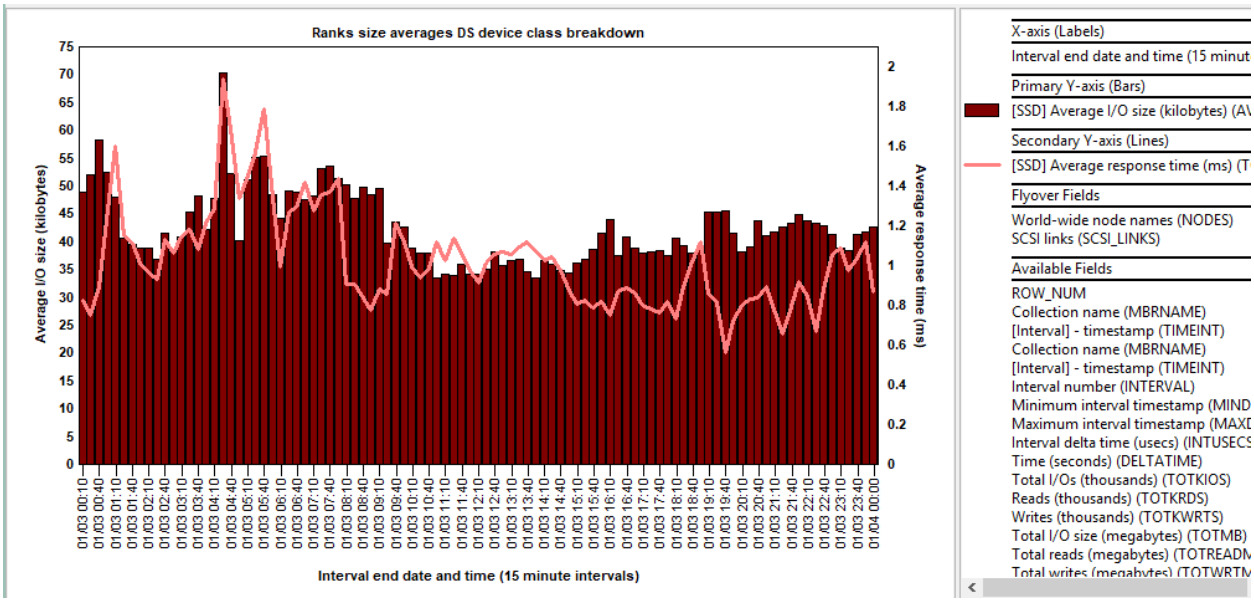
The possible device classes in this data are:

1) Enterprise
2) Near-line (ATA)
3) SATA
4) SSD



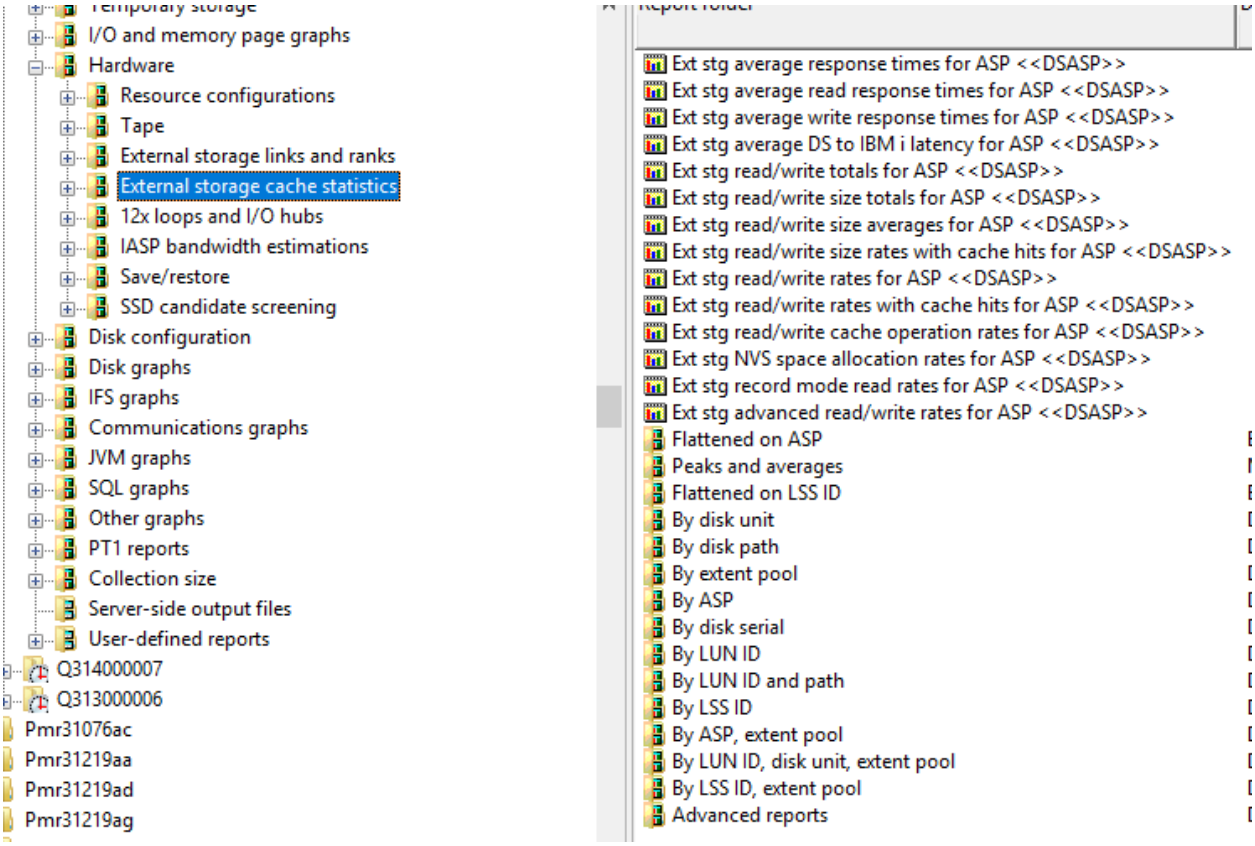
Hardware -> External storage links and ranks -> Ranks -> Flattened by device class

An example of this type of graph is shown below:



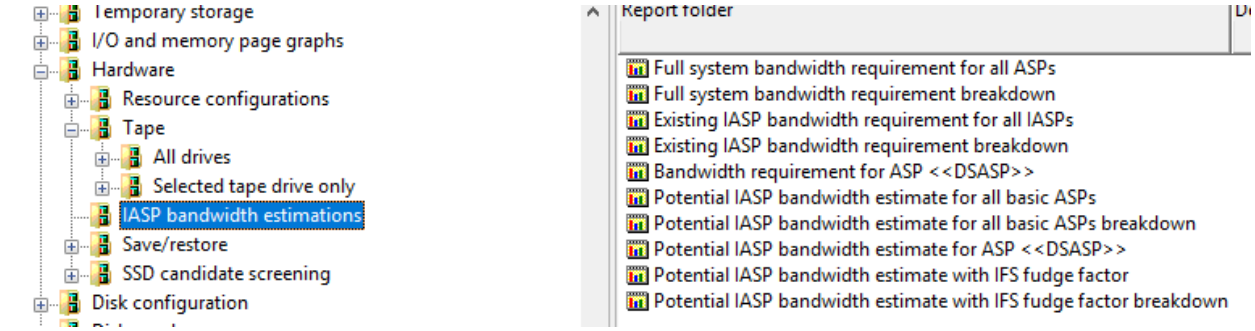
Ranks size averages DS device class breakdown

9.21.4 External storage cache statistics



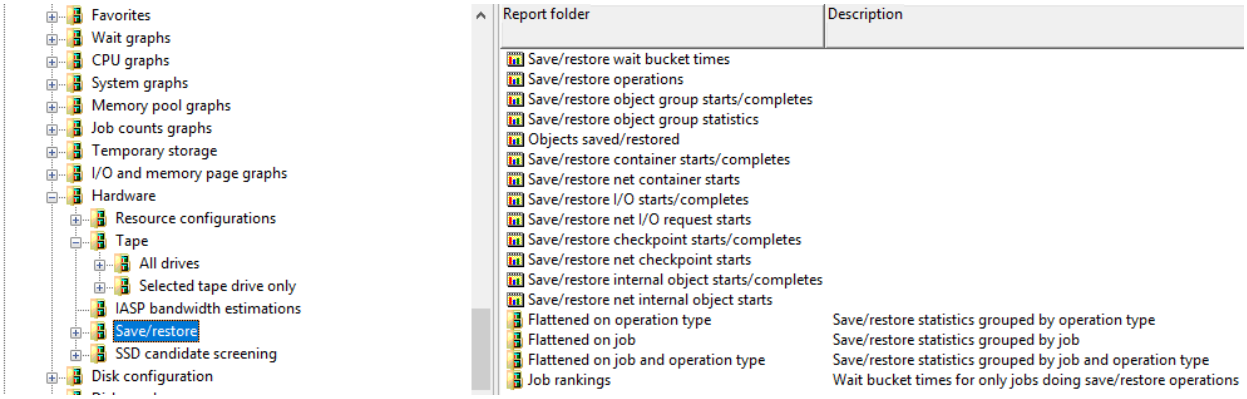
Hardware -> External storage cache statistics

9.21.5 IASP Bandwidth estimations



Hardware -> IASP Bandwidth estimations

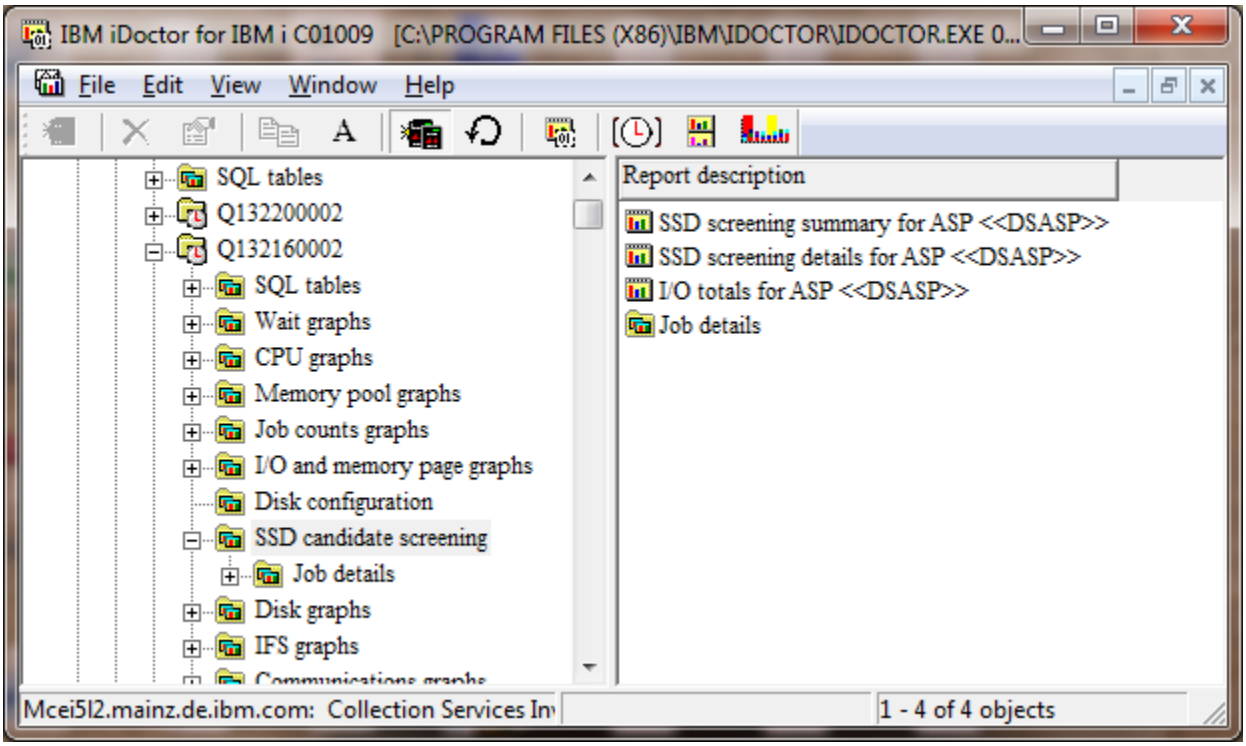
9.21.6 Save/restore



Hardware -> Save/restore

9.21.7 SSD candidate screening

These reporting options are designed to help a user determine if the system would be a good candidate for installing SSDs. The graphing options are filtered by the desired ASP, which lets you focus on a single ASP at a time.



SSD candidate screening options

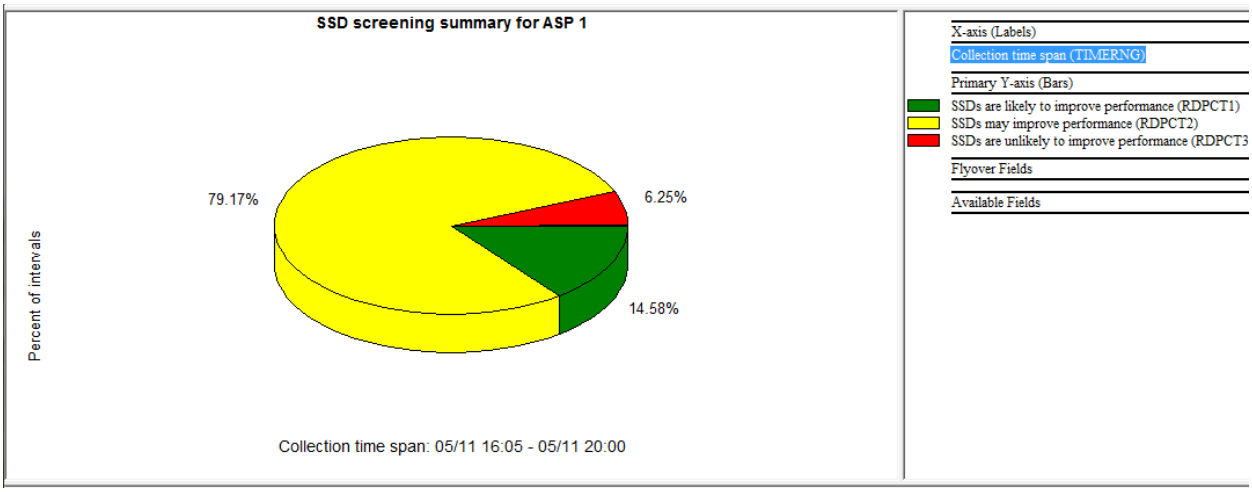
9.21.7.1 SSD screening summary for ASP <<DSASP>>

When opening this report you may be prompted for the ASP desired to analyze if more than 1 is detected.

It simply indicates an overview of how likely it is that SSDs would improve performance. This is tested by looking at the average read response time for each Collection Services interval and categorizing the intervals into 3 groups:

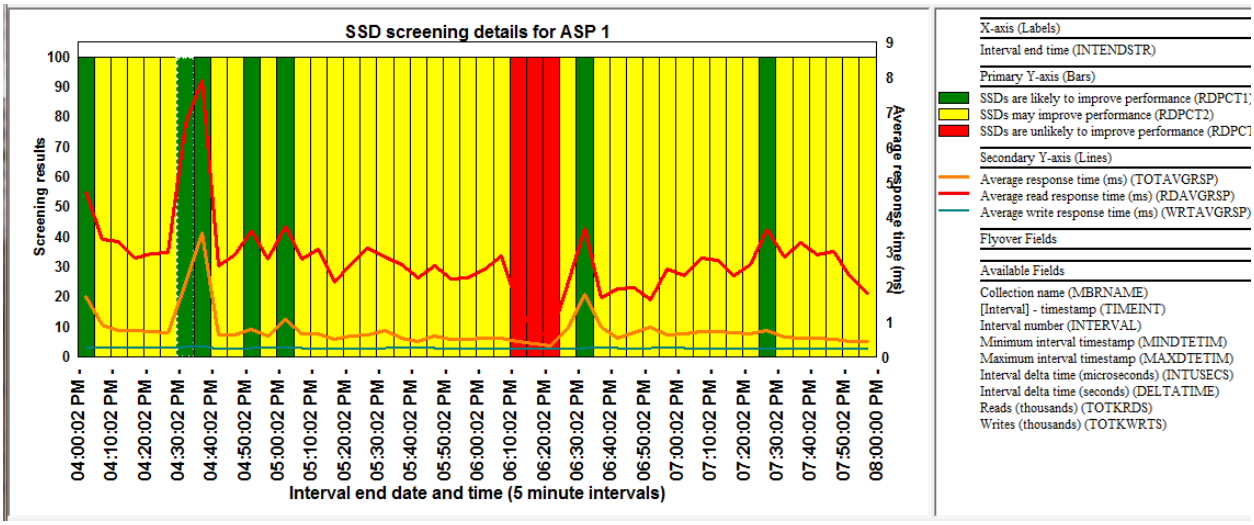
1. Read response time > 3.5 ms = “likely to improve performance”

2. Read response time between 1.5 and 3.5 ms = “may improve performance”
3. Read response time < 1.5 ms = “unlikely to improve performance”



SSD screening summary for ASP 1

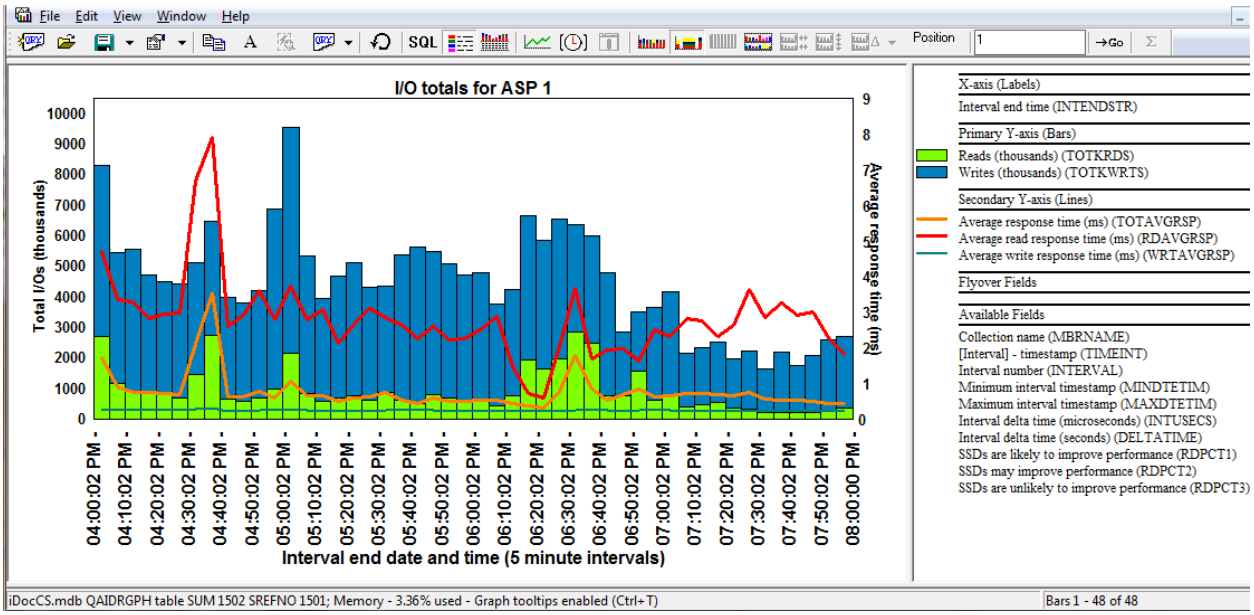
9.21.7.2 SSD screening details for ASP <<DSASP>>



SSD screening details for ASP 1

This graph is similar to the previous graph except it shows the data over time for each interval and also displays the average response time and average read and write response times on the 2nd Y-axis for more granularity.

9.21.7.3 I/O totals for ASP <<DSASP>>



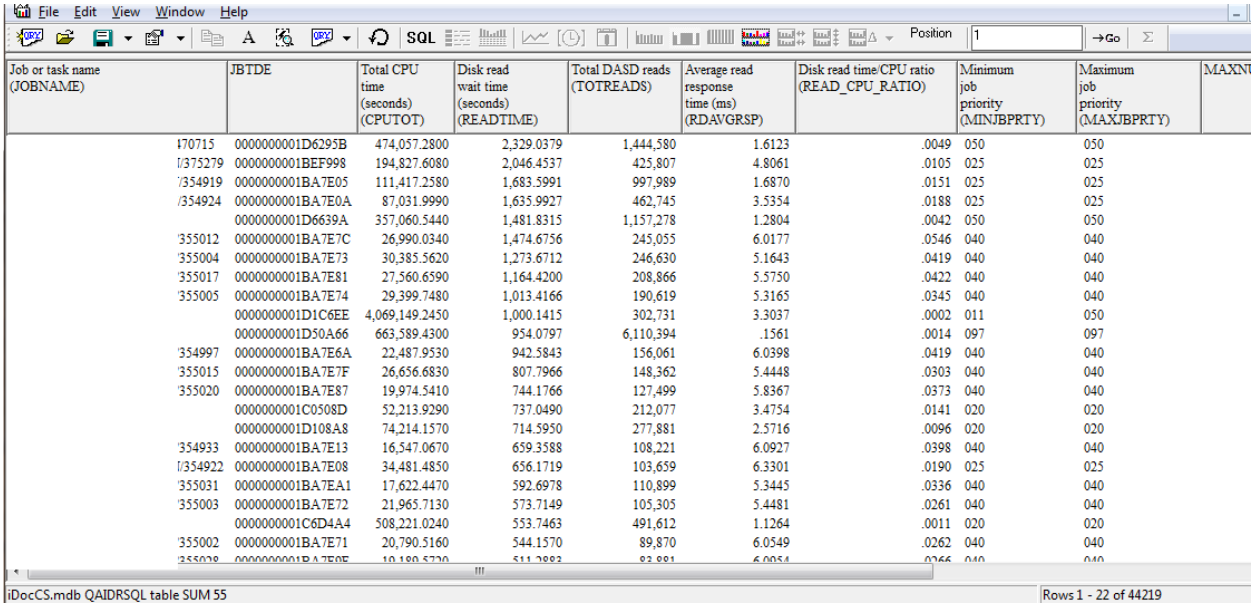
I/O totals for ASP 1

This graph shows the total reads and writes along with the response times on the 2nd Y-axis.

9.21.7.4 Job details

This folder contains a single report that looks at the total reads and average read response time on a per job basis.

9.21.7.5 SSD screening details by job



Job or task name (JOBNAME)	JBTDE	Total CPU time (seconds) (CPUTOT)	Disk read wait time (seconds) (READTIME)	Total DASD reads (TOTREADS)	Average read response time (ms) (RDAVGRSP)	Disk read time/CPU ratio (READ_CPU_RATIO)	Minimum job priority (MINJBPRTY)	Maximum job priority (MAXJBPRTY)	MAXN
170715	0000000001D6295B	474,057.2800	2,329.0379	1,444,580	1.6123	.0049	050	050	
375279	0000000001BEF998	194,827.6080	2,046.4537	425,807	4.8061	.0105	025	025	
354919	0000000001BA7E05	111,417.2580	1,683.5991	997,989	1.6870	.0151	025	025	
354924	0000000001BA7E0A	87,031.9990	1,635.9927	462,745	3.5354	.0188	025	025	
	0000000001D6639A	357,060.5440	1,481.8315	1,157,278	1.2804	.0042	050	050	
355012	0000000001BA7E7C	26,990.0340	1,474.6756	245,055	6.0177	.0546	040	040	
355004	0000000001BA7E73	30,385.5620	1,273.6712	246,630	5.1643	.0419	040	040	
355017	0000000001BA7E81	27,560.6590	1,164.4200	208,866	5.5750	.0422	040	040	
355005	0000000001BA7E74	29,399.7480	1,013.4166	190,619	5.3165	.0345	040	040	
	0000000001D1C6EE	4,069,149.2450	1,000.1415	302,731	3.3037	.0002	011	050	
	0000000001D50A66	663,589.4300	954.0797	6,110,394	.1561	.0014	097	097	
354997	0000000001BA7E6A	22,487.9530	942.5843	156,061	6.0398	.0419	040	040	
355015	0000000001BA7E7F	26,656.6830	807.7966	148,362	5.4448	.0303	040	040	
355020	0000000001BA7E87	19,974.5410	744.1766	127,499	5.8367	.0373	040	040	
	0000000001C0508D	52,213.9290	737.0490	212,077	3.4754	.0141	020	020	
	0000000001D108A8	74,214.1570	714.5950	277,881	2.5716	.0096	020	020	
354933	0000000001BA7E13	16,547.0670	659.3588	108,221	6.0927	.0398	040	040	
354922	0000000001BA7E08	34,481.4850	656.1719	103,659	6.3301	.0190	025	025	
355031	0000000001BA7EA1	17,622.4470	592.6978	110,899	5.3445	.0336	040	040	
355003	0000000001BA7E72	21,965.7130	573.7149	105,305	5.4481	.0261	040	040	
	0000000001C6D4A4	508,221.0240	553.7463	491,612	1.1264	.0011	020	020	
355002	0000000001BA7E71	20,790.5160	544.1570	89,870	6.0549	.0262	040	040	

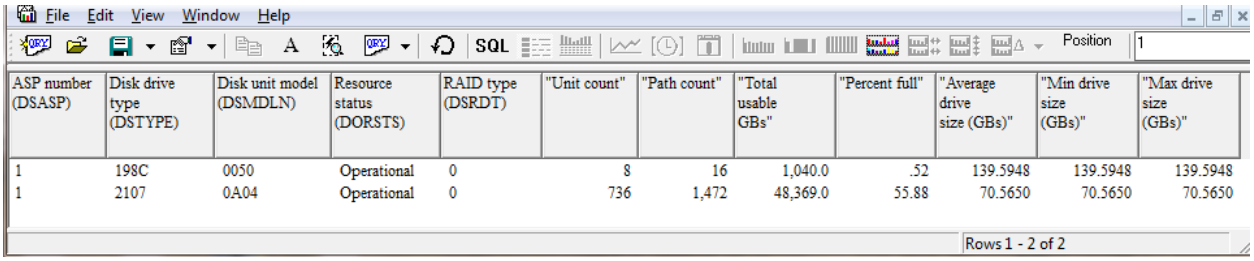
SSD screening details by job

This report lists the jobs in the collection and sorts the data by total disk read wait time (in seconds.) It helps to isolate which jobs are performing disk reads and might be helped by installing SSDs in order to improve disk read response time.

9.22 Disk configuration

These reports display information about the system's disk configuration including the ASPs, IOPs and IOAs.

9.22.1 Capacity (in GBs) by ASP with paths



ASP number (DSASP)	Disk drive type (DSTYPE)	Disk unit model (DSMDLN)	Resource status (DORSTS)	RAID type (DSRDT)	"Unit count"	"Path count"	"Total usable GBs"	"Percent full"	"Average drive size (GBs)"	"Min drive size (GBs)"	"Max drive size (GBs)"
1	198C	0050	Operational	0	8	16	1,040.0	.52	139.5948	139.5948	139.5948
1	2107	0A04	Operational	0	736	1,472	48,369.0	55.88	70.5650	70.5650	70.5650

Capacity (in GBs) by ASP with paths

This report displays each ASP and type of disk with the total disk units and total paths for each provided along with additional information.

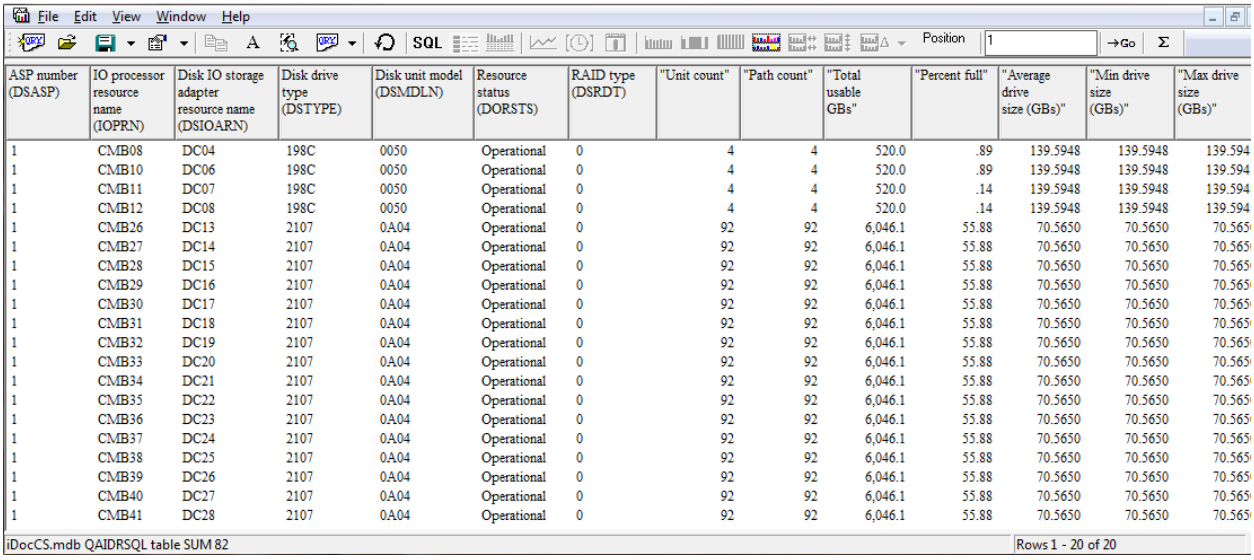
9.22.2 Capacity (in GBs) by ASP/IOP with paths

ASP number (DSASP)	IO processor resource name (IOPRN)	Disk drive type (DSTYPE)	Disk unit model (DSMDLN)	Resource status (DORSTS)	RAID type (DSRDT)	"Unit count"	"Path count"	"Total usable GBs"	"Percent full"	"Average drive size (GBs)"	"Min drive size (GBs)"	"Max drive size (GBs)"
1	CMB08	198C	0050	Operational	0	4	4	520.0	.89	139.5948	139.5948	139.5948
1	CMB10	198C	0050	Operational	0	4	4	520.0	.89	139.5948	139.5948	139.5948
1	CMB11	198C	0050	Operational	0	4	4	520.0	.14	139.5948	139.5948	139.5948
1	CMB12	198C	0050	Operational	0	4	4	520.0	.14	139.5948	139.5948	139.5948
1	CMB26	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB27	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB28	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB29	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB30	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB31	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB32	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB33	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB34	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB35	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB36	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB37	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB38	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB39	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB40	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650
1	CMB41	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.5650

Capacity (in GBs) by ASP/IOP with paths

This report is similar to the previous report except it also includes the data for each IOP.

9.22.3 Capacity (in GBs) by ASP/IOP/IOA with paths



File Edit View Window Help

SQL Position 1 →Go Σ

ASP number (DSASP)	IO processor resource name (IOPRN)	Disk IO storage adapter resource name (DSIOARN)	Disk drive type (DSTYPE)	Disk unit model (DSMDLN)	Resource status (DORSTS)	RAID type (DSRDT)	"Unit count"	"Path count"	"Total usable GBs"	"Percent full"	"Average drive size (GBs)"	"Min drive size (GBs)"	"Max drive size (GBs)"
1	CMB08	DC04	198C	0050	Operational	0	4	4	520.0	.89	139.5948	139.5948	139.594
1	CMB10	DC06	198C	0050	Operational	0	4	4	520.0	.89	139.5948	139.5948	139.594
1	CMB11	DC07	198C	0050	Operational	0	4	4	520.0	.14	139.5948	139.5948	139.594
1	CMB12	DC08	198C	0050	Operational	0	4	4	520.0	.14	139.5948	139.5948	139.594
1	CMB26	DC13	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB27	DC14	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB28	DC15	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB29	DC16	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB30	DC17	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB31	DC18	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB32	DC19	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB33	DC20	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB34	DC21	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB35	DC22	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB36	DC23	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB37	DC24	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB38	DC25	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB39	DC26	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB40	DC27	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565
1	CMB41	DC28	2107	0A04	Operational	0	92	92	6,046.1	55.88	70.5650	70.5650	70.565

iDocCS.mdb QAIDRSQL table SUM 82 Rows 1 - 20 of 20

Capacity (in GBs) by ASP/IOP with paths

This report is similar to the first report except it also includes the data for each IOP and IOA.

9.22.4 Disk configuration

Disk arm number (DSARM)	Device resource name (DSDRN)	Disk drive type (DSTYPE)	Resource model number (DORMOD)	Resource part number (DORPRT)	Disk type description (DISKGRP)	ASP number (DSASP)	IO processor resource name (IOPRN)	Disk IO storage adapter resource name (DSIOARN)	Disk drive size (GBs) (CAPACITY)	Resource status (DORSTS)	IOA CCIN (IOACCIN)	IO (IC
0001	DMP007	198C	050	44V6841	15K SAS	1	CMB08	DC04	139.5948	Operational	57B5	
0003	DMP019	198C	050	44V6841	15K SAS	1	CMB08	DC04	139.5948	Operational	57B5	
0006	DMP023	198C	050	44V6841	15K SAS	1	CMB08	DC04	139.5948	Operational	57B5	
0002	DMP027	198C	050	44V6841	15K SAS	1	CMB08	DC04	139.5948	Operational	57B5	
0001	DMP005	198C	050	44V6841	15K SAS	1	CMB10	DC06	139.5948	Operational	57B5	
0003	DMP003	198C	050	44V6841	15K SAS	1	CMB10	DC06	139.5948	Operational	57B5	
0005	DMP011	198C	050	44V6841	15K SAS	1	CMB10	DC06	139.5948	Operational	57B5	
0007	DMP015	198C	050	44V6841	15K SAS	1	CMB10	DC06	139.5948	Operational	57B5	
0002	DMP002	198C	050	44V6841	15K SAS	1	CMB11	DC07	139.5948	Operational	57B5	
0004	DMP010	198C	050	44V6841	15K SAS	1	CMB11	DC07	139.5948	Operational	57B5	
0006	DMP014	198C	050	44V6841	15K SAS	1	CMB11	DC07	139.5948	Operational	57B5	
0008	DMP032	198C	050	44V6841	15K SAS	1	CMB11	DC07	139.5948	Operational	57B5	
0008	DMP018	198C	050	44V6841	15K SAS	1	CMB12	DC08	139.5948	Operational	57B5	
0005	DMP022	198C	050	44V6841	15K SAS	1	CMB12	DC08	139.5948	Operational	57B5	
0007	DMP026	198C	050	44V6841	15K SAS	1	CMB12	DC08	139.5948	Operational	57B5	
0004	DMP030	198C	050	44V6841	15K SAS	1	CMB12	DC08	139.5948	Operational	57B5	
0101	DMP035	2107	A04		DS8000	1	CMB26	DC13	70.5650	Operational	577D	
0092	DMP033	2107	A04		DS8000	1	CMB26	DC13	70.5650	Operational	577D	
0110	DMP037	2107	A04		DS8000	1	CMB26	DC13	70.5650	Operational	577D	
0119	DMP039	2107	A04		DS8000	1	CMB26	DC13	70.5650	Operational	577D	
0131	DMP041	2107	A04		DS8000	1	CMB26	DC13	70.5650	Operational	577D	

iDocCS.mdb QAIDRSQL table SUM 85 Rows 1 - 21 of 1488

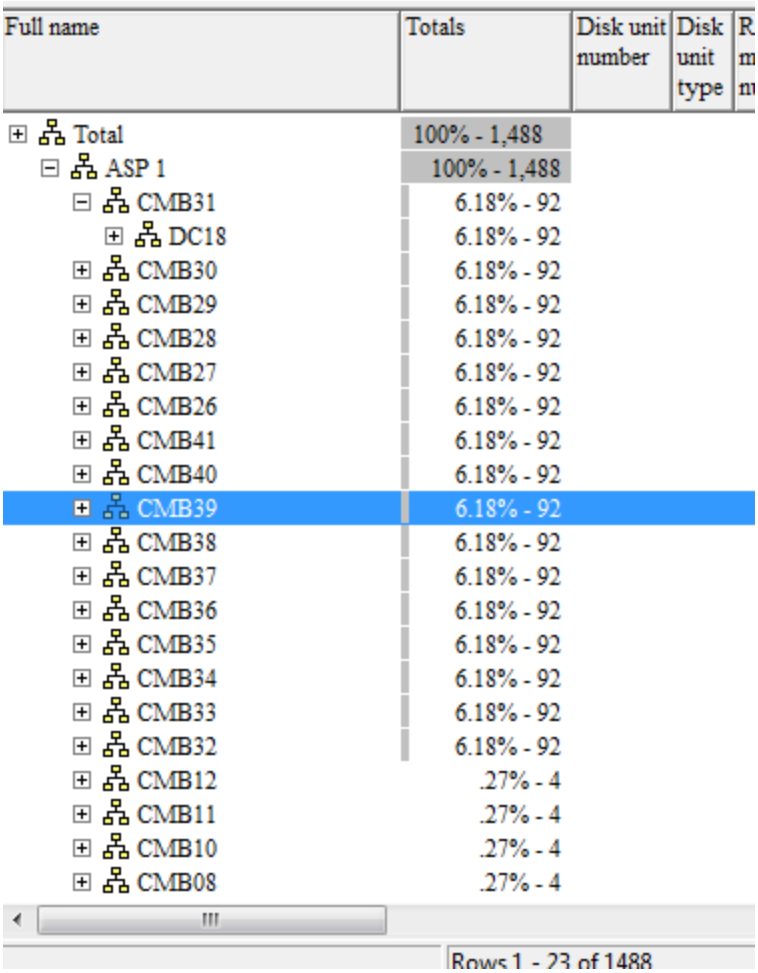
Disk configuration

This report lists the complete disk configuration of a partition.

9.22.5 Disk configuration (non-operational disks only)

This report is the same as the previous one, except it only lists disks that are detected as not operational when the collection was taken.

9.22.6 Disk configuration (tree) by ASP/IOP/IOA/Unit

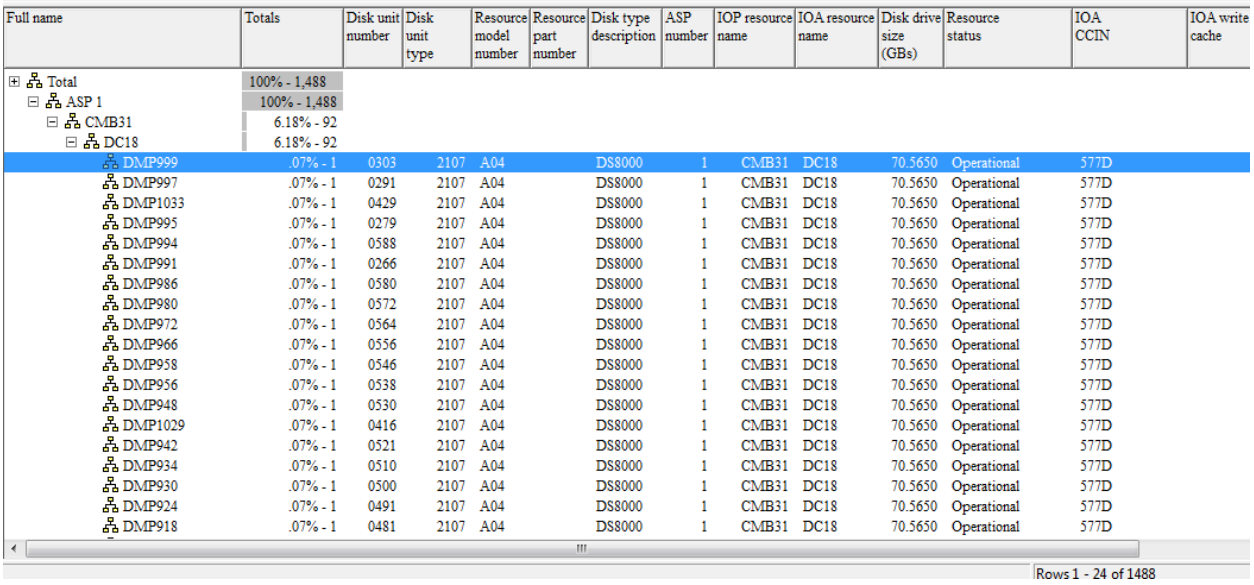


Full name	Totals	Disk unit number	Disk unit type	R m n
Total	100% - 1,488			
ASP 1	100% - 1,488			
CMB31	6.18% - 92			
DC18	6.18% - 92			
CMB30	6.18% - 92			
CMB29	6.18% - 92			
CMB28	6.18% - 92			
CMB27	6.18% - 92			
CMB26	6.18% - 92			
CMB41	6.18% - 92			
CMB40	6.18% - 92			
CMB39	6.18% - 92			
CMB38	6.18% - 92			
CMB37	6.18% - 92			
CMB36	6.18% - 92			
CMB35	6.18% - 92			
CMB34	6.18% - 92			
CMB33	6.18% - 92			
CMB32	6.18% - 92			
CMB12	.27% - 4			
CMB11	.27% - 4			
CMB10	.27% - 4			
CMB08	.27% - 4			

Rows 1 - 23 of 1488

Disk configuration (tree) without disk units shown

This report lets you expand/collapse individual sections of the disk configuration as desired.



Full name	Totals	Disk unit number	Disk unit type	Resource model number	Resource part number	Disk type description	ASP number	IOP resource name	IOA resource name	Disk drive size (GBs)	Resource status	IOA CCIN	IOA write cache
Total	100% - 1,488												
ASP 1	100% - 1,488												
CMB31	6.18% - 92												
DC18	6.18% - 92												
DMP999	.07% - 1	0303	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP997	.07% - 1	0291	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP1033	.07% - 1	0429	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP995	.07% - 1	0279	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP994	.07% - 1	0588	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP991	.07% - 1	0266	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP986	.07% - 1	0580	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP980	.07% - 1	0572	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP972	.07% - 1	0564	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP966	.07% - 1	0556	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP958	.07% - 1	0546	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP956	.07% - 1	0538	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP948	.07% - 1	0530	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP1029	.07% - 1	0416	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP942	.07% - 1	0521	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP934	.07% - 1	0510	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP930	.07% - 1	0500	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP924	.07% - 1	0491	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	
DMP918	.07% - 1	0481	2107	A04		DS8000	1	CMB31	DC18	70.5650	Operational	577D	

Rows 1 - 24 of 1488

Disk configuration (tree) with disk units shown

9.23 Disk graphs

A large number of disk reporting options are available within this folder. This data is based primarily on data found within Collection Services file QAPMDISK. Several different groupings are available as well, which behave similarly in design to the job groupings. Each of these grouping (rankings) folders contain the same set as graphs from the “over-time” set. You can drill down from the over-time graphs into these rankings graphs. These groupings are:

1. By disk path
2. By disk unit
3. By I/O processor
4. By ASP
5. By disk type
6. By I/O adapter (6.1+)

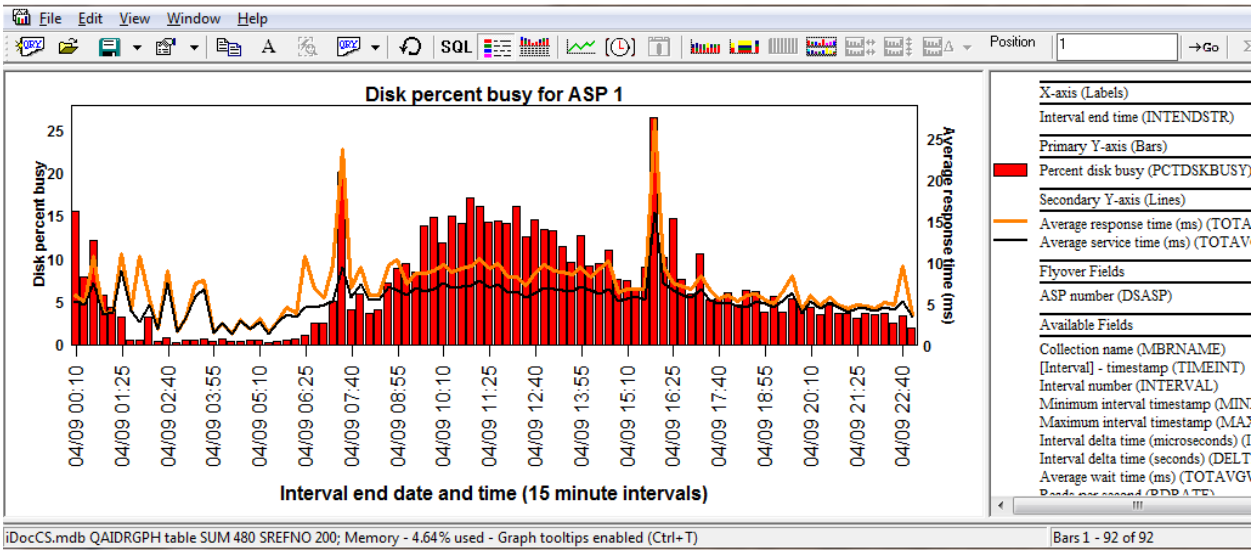
One limitation of file QAPMDISK however is it does not separate out read response times from write response times. At 7.1 however file QAPMDISKRB was added to the system in order to accomplish this and to also provide response time buckets for both reads and writes. These options are included under the Disk graphs -> Advanced folder.

Note: At 5.4, the latest version of file QAPMDISK should be used, otherwise a much smaller set of graphs will be shown. In most cases this will not be a problem however because the PTFs for this update were released many years ago.

For more information about the metrics in these graphs visit the following page:

http://pic.dhe.ibm.com/infocenter/iseries/v7r1m0/topic/rzahx/rzahxqapmdisk.htm?resultof=%22%51%41%50%4d%44%49%53%4b%22%20%22%71%61%70%6d%64%69%73%6b%22%20

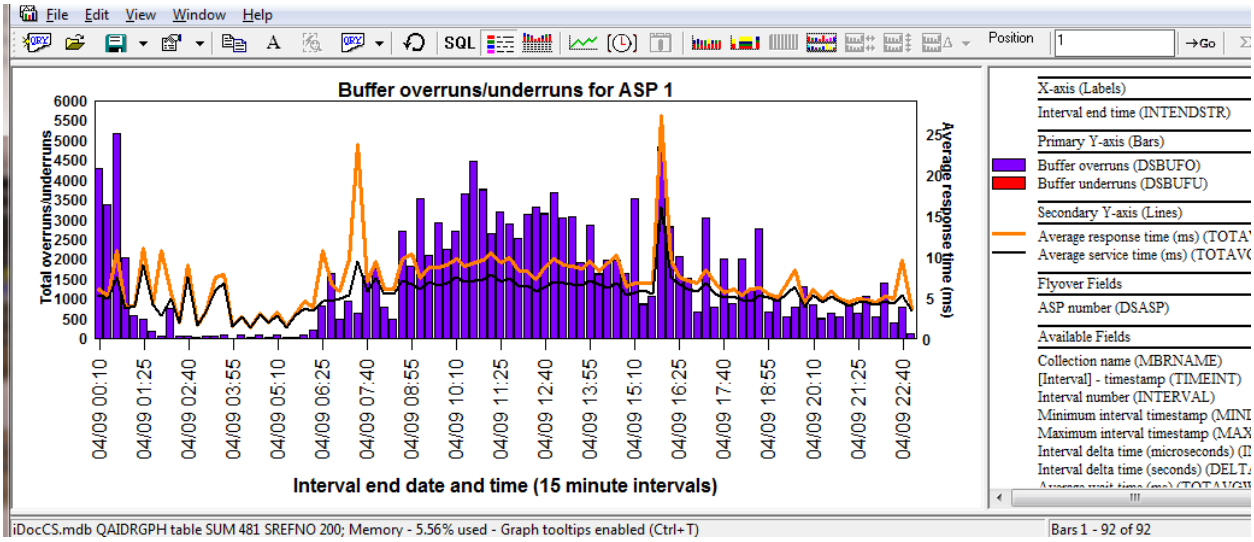
9.23.1 Disk percent busy for ASP <<DSASP>>



Disk percent busy for ASP 1

This graph displays the (average) disk percent busy for the desired ASP(s), along with the average response times on the 2nd Y-axis. Both the average response time and average service time are provided.

9.23.2 Buffer overruns/underruns for ASP <<DSASP>>



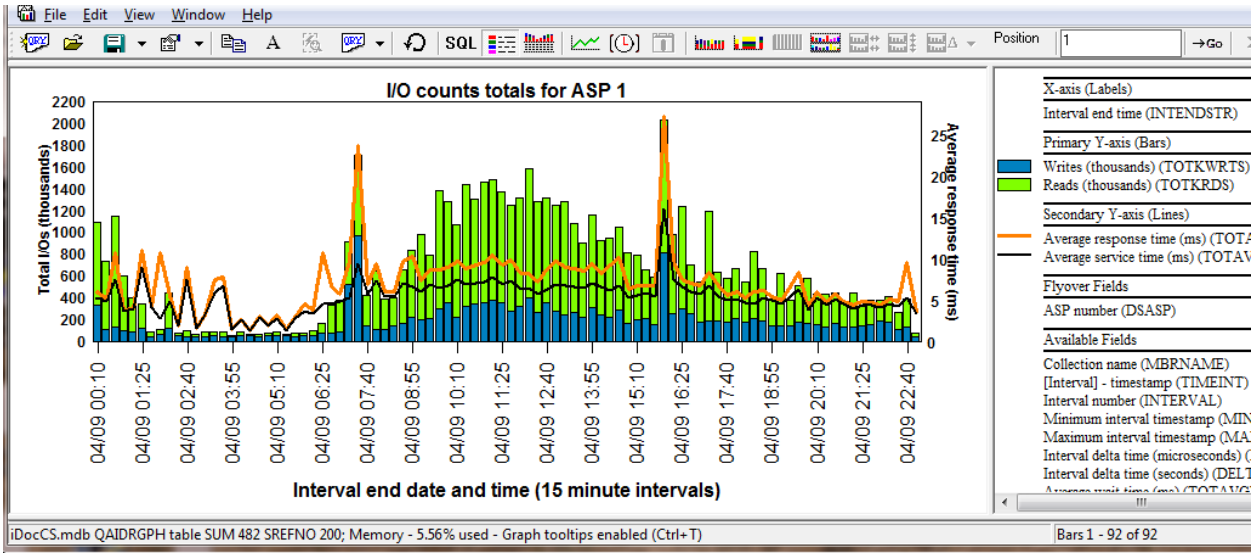
Buffer overruns/underruns for ASP 1

Buffer overruns are the number of times that data was available to be read into the disk controller buffer from the disk, but the disk controller buffer still contained valid data that was not retrieved by the storage device controller. Consequently, the disk had to take an additional revolution until the buffer was available to accept data.

Buffer underruns are the number of times that the disk controller was ready to transfer data to the disk on a write, but the disk controller buffer was empty. The data was not transferred in time by the disk IOP to the disk controller buffer. The disk was forced to take an extra revolution awaiting the data.

This graph also displays the average response times on the 2nd Y-axis.

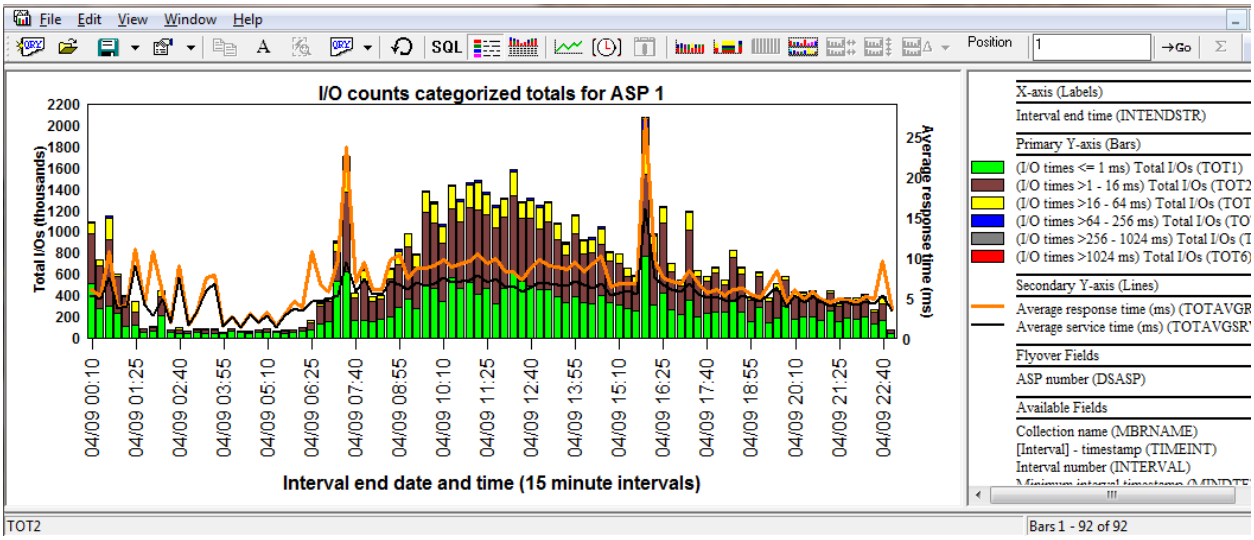
9.23.3 I/O counts totals for ASP <<DSASP>>



I/O counts totals for ASP 1

This graph displays the total number of reads and writes along with the average response times on the 2nd Y-axis.

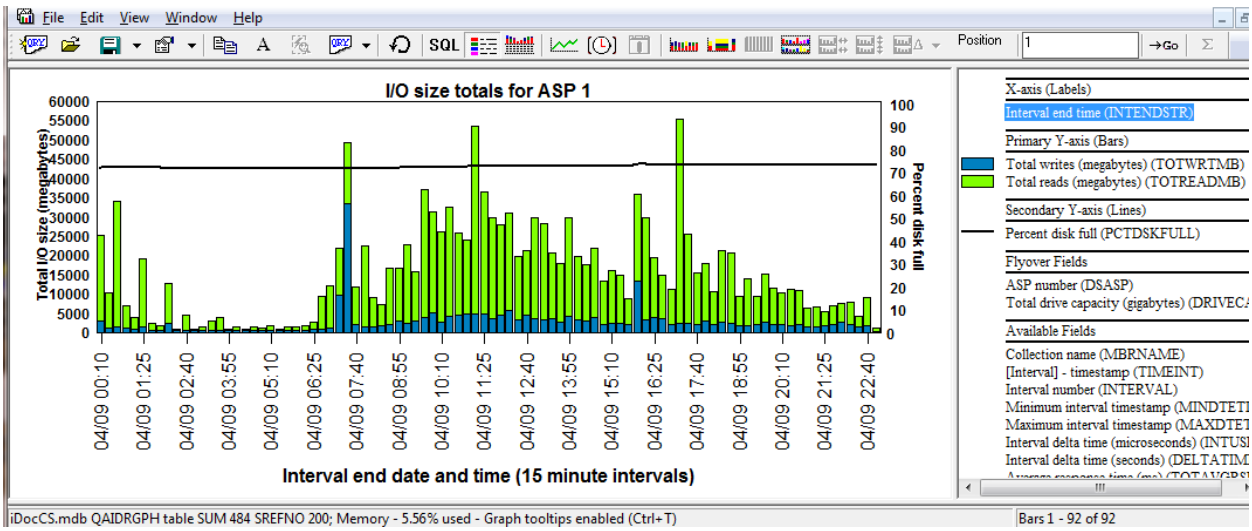
9.23.4 I/O counts categorized totals for ASP <<DSASP>>



I/O counts categorized totals for ASP 1

This graph displays the number of I/Os that occurred based on their response times. Six response time buckets are used each with a different color. The overall average response times are shown on the 2nd Y-axis.

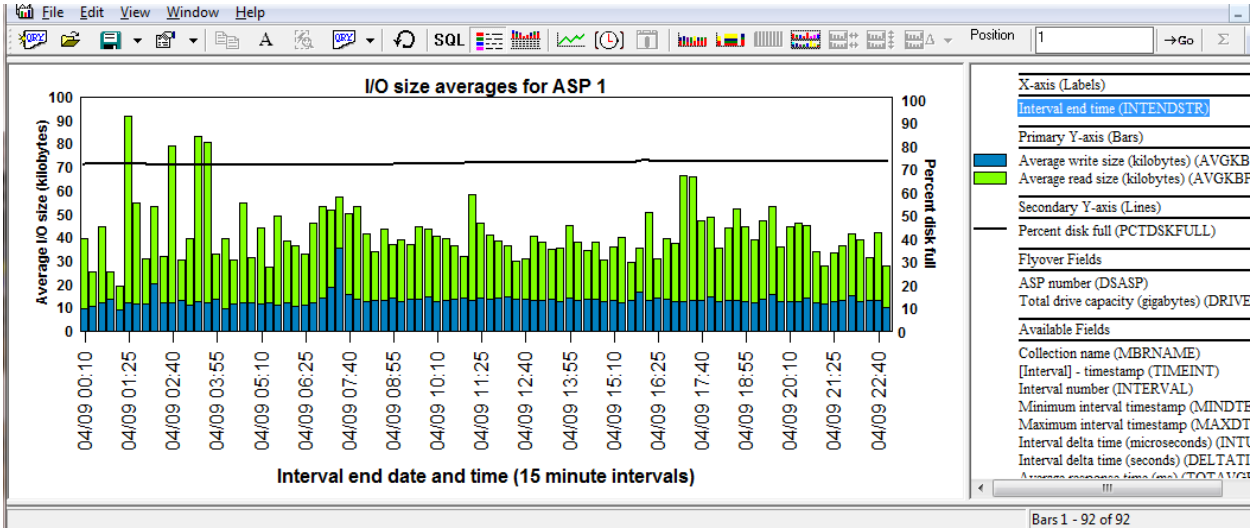
9.23.5 I/O size totals for ASP <<DSASP>>



I/O size totals for ASP 1

This graph displays the total I/O size (in megabytes) for both reads and writes that occurred over time. The 2nd Y-axis displays the overall average disk percent full (from each disk unit) for the ASPs selected.

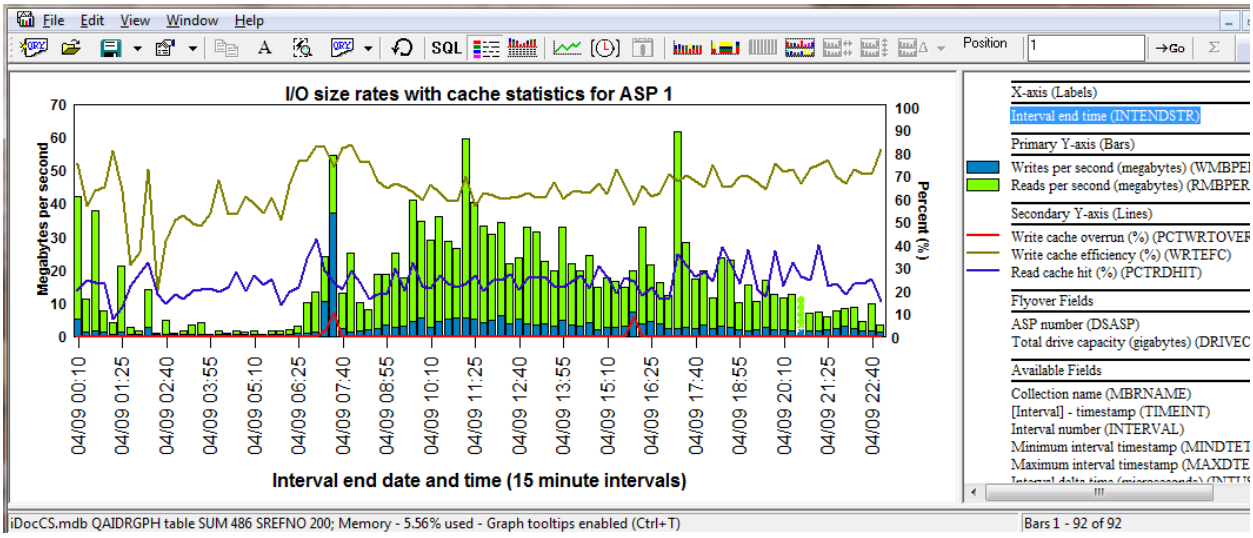
9.23.6 I/O size averages for ASP <<DSASP>>



I/O size averages for ASP <<DSASP>>

This graph shows the average I/O size (in kilobytes) for both reads and writes. The 2nd Y-axis displays the overall average disk percent full (from each disk unit) for the ASPs selected.

9.23.7 I/O size rates with cache statistics for ASP <<DSASP>>

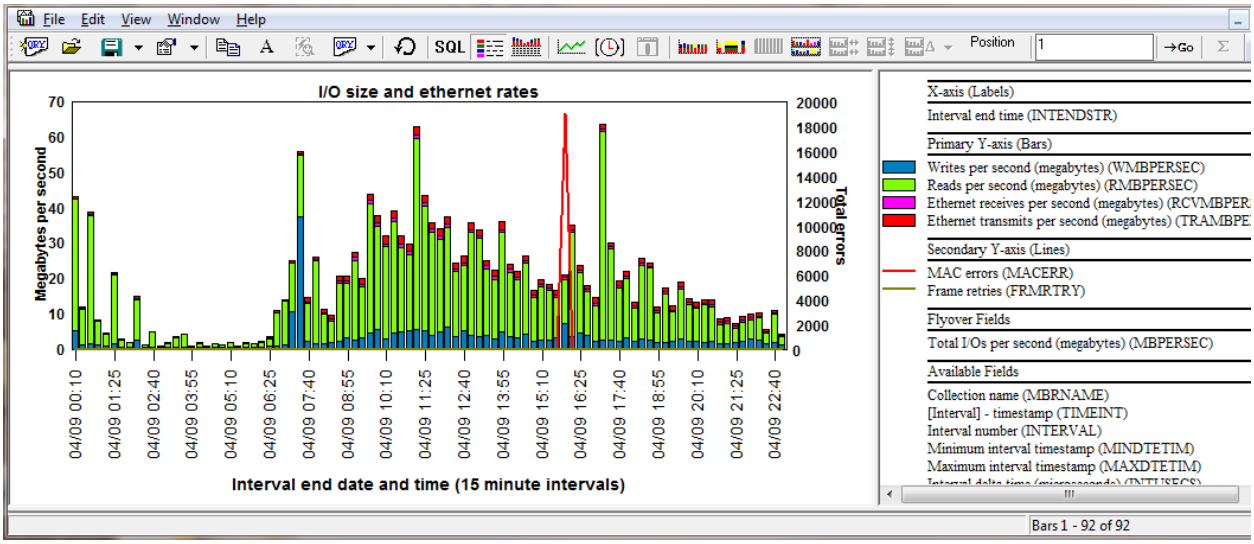


I/O size rates with cache statistics for ASP 1

This report displays the size of all disk reads and disk writes as megabytes per second.

The cache statistics shown on the 2nd Y-axis includes write cache overruns, write cache efficiency and read cache hits.

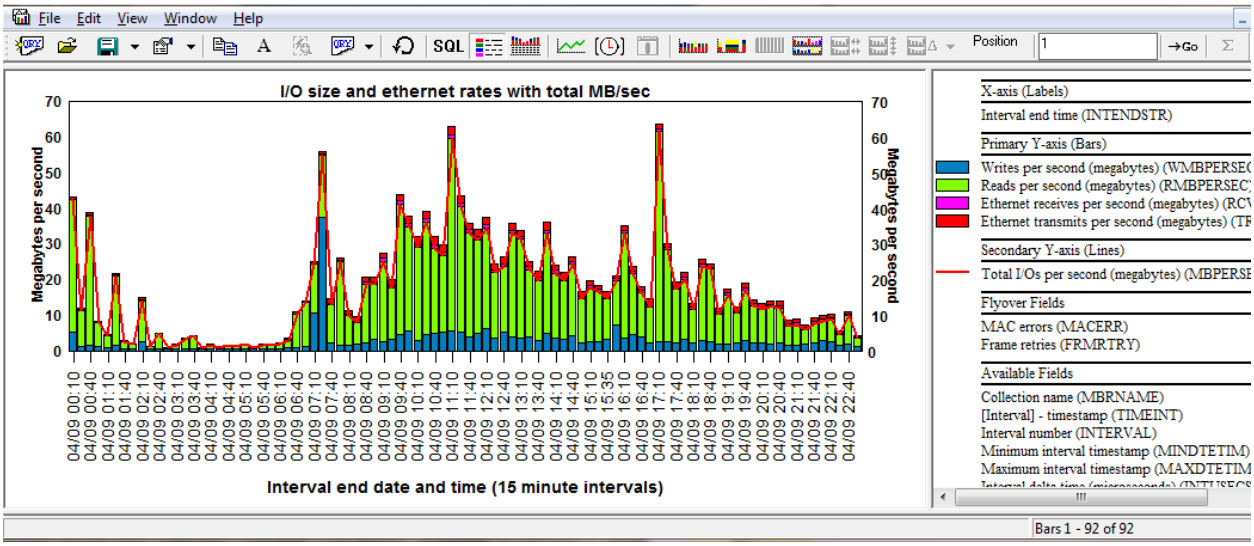
9.23.8 I/O size and Ethernet rates



I/O size and Ethernet rates

This graph combines I/O size rates (in megabytes per second) with Ethernet transmission rates. The 2nd Y-axis shows Ethernet MAC errors and frame retries.

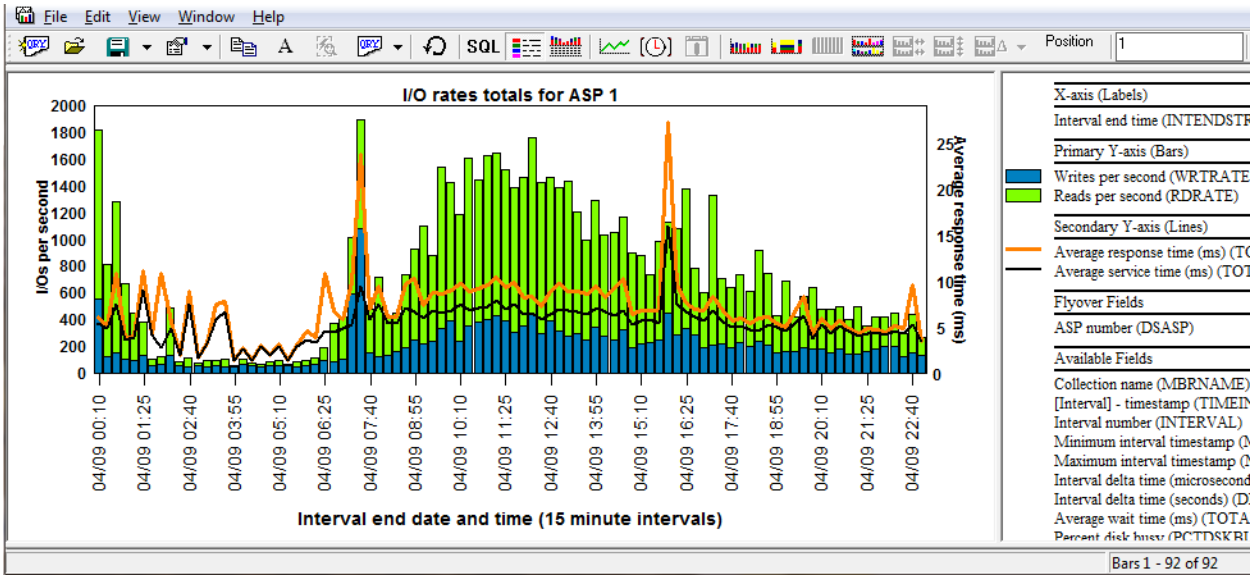
9.23.9 I/O size and Ethernet rates with total MB/sec



I/O size and ethernet rates with total MB/sec

This graph is the same as the previous one except the total I/Os per second for both the disk and Ethernet portion are added up on the 2nd Y-axis.

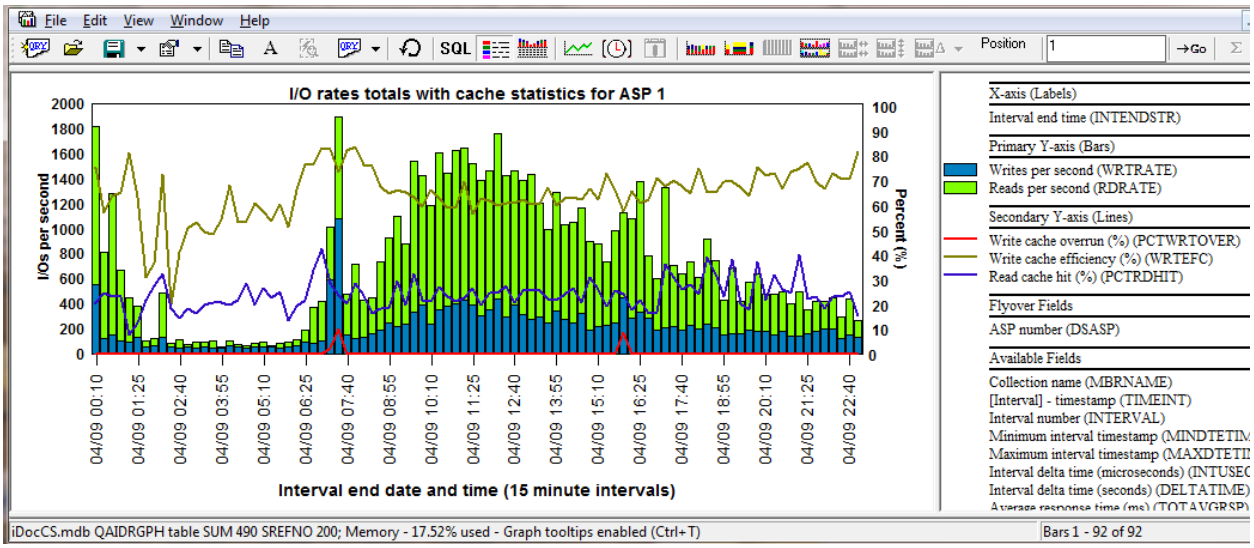
9.23.10 I/O rates totals for ASP <<DSASP>>



I/O rates totals for ASP 1

This graph shows the reads per second and writes per second over time. The 2nd Y-axis displays the average response times.

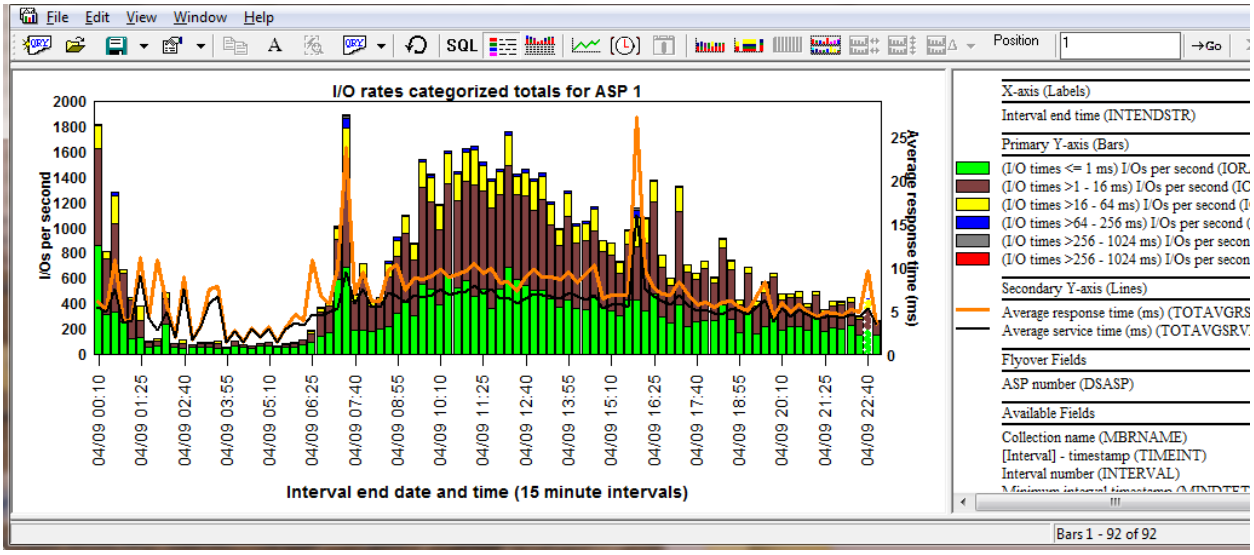
9.23.11 I/O rates totals with cache statistics for ASP <<DSASP>>



I/O rates totals with cache statistics for ASP 1

This graph is the same as previous graph except it shows the cache statistics on the 2nd Y-axis.

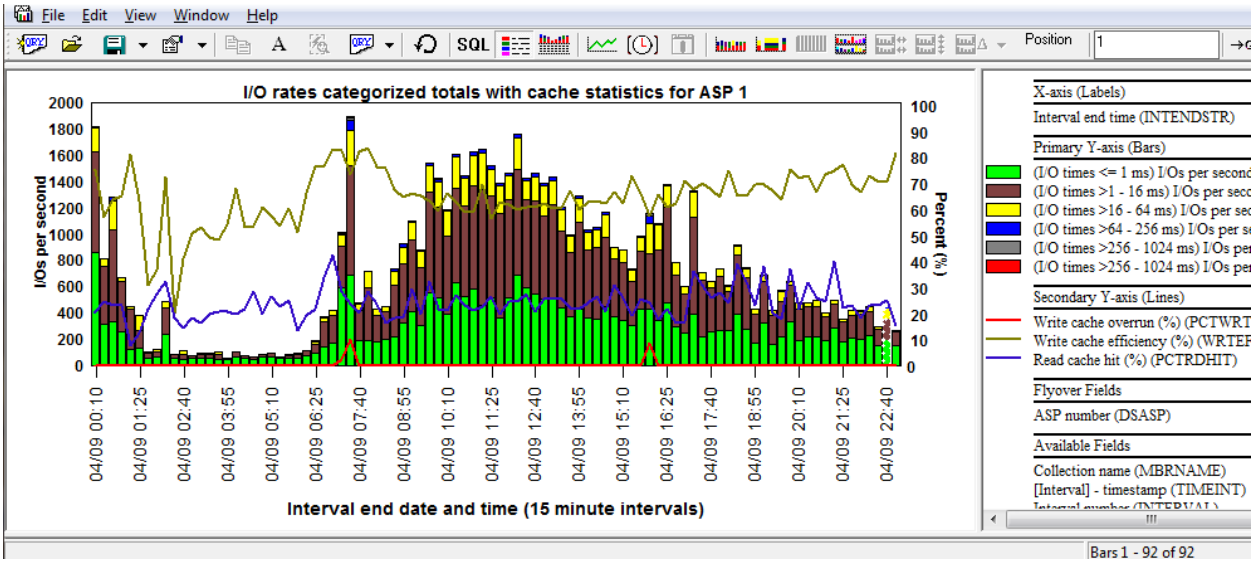
9.23.12 I/O rates categorized totals for ASP <<DSASP>>



I/O rates categorized totals for ASP 1

This graph displays the I/Os per second that occurred based on their response times. Six response time buckets are used each with a different color. The overall average response times are shown on the 2nd Y-axis.

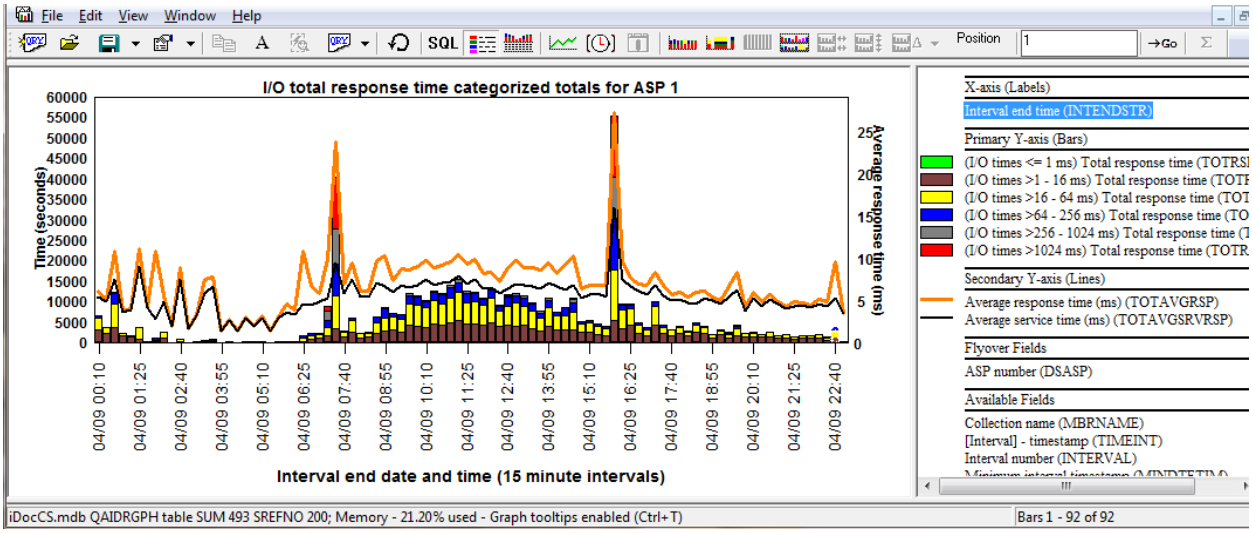
9.23.13 I/O rates categorized totals with cache statistics for ASP <<DSASP>>



I/O rates categorized totals with cache statistics for ASP 1

This graph is the same as the previous one except it shows the cache statistics on the 2nd Y-axis.

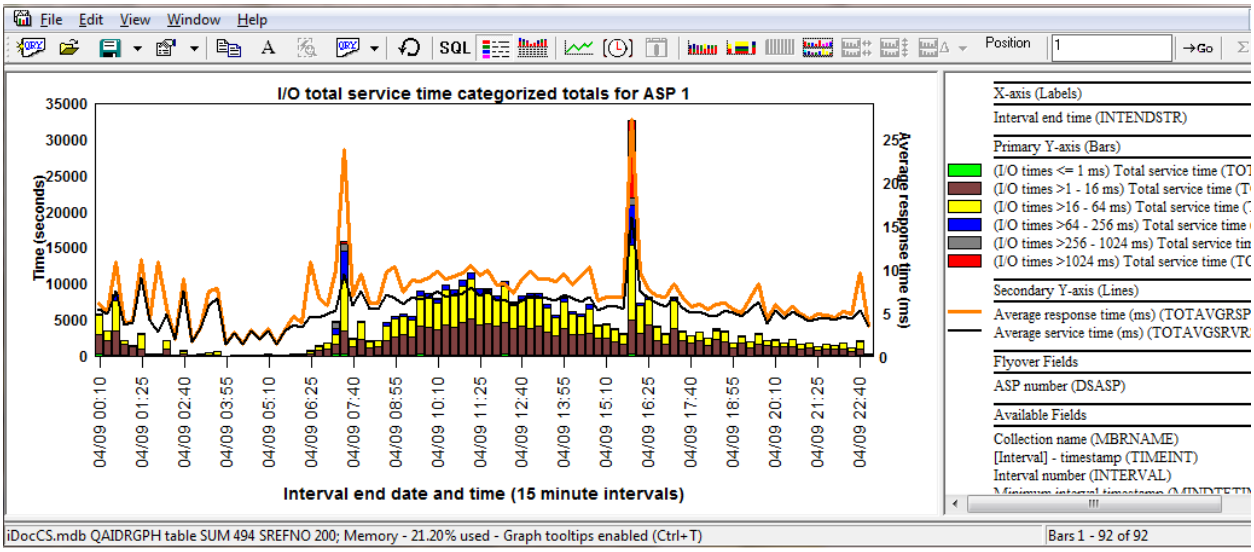
9.23.14 I/O total response time categorized totals for ASP <<DSASP>>



I/O total response time categorized totals for ASP 1

This graph adds up the total response times for all I/Os and shows them with different colors based on the six response time buckets. The 2nd Y-axis displays the overall average response times.

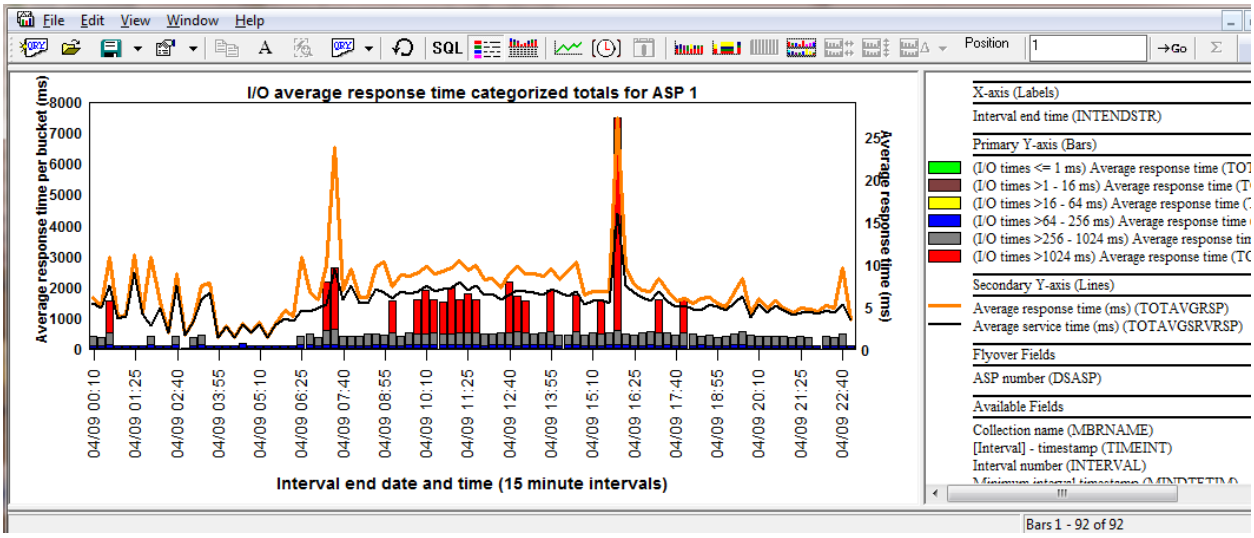
9.23.15 I/O total service time categorized totals for ASP <<DSASP>>



I/O total service time categorized totals for ASP 1

This graph adds up the total service (hardware) times for all I/Os and shows them with different colors based on the six response time buckets. The 2nd Y-axis displays the overall average response times.

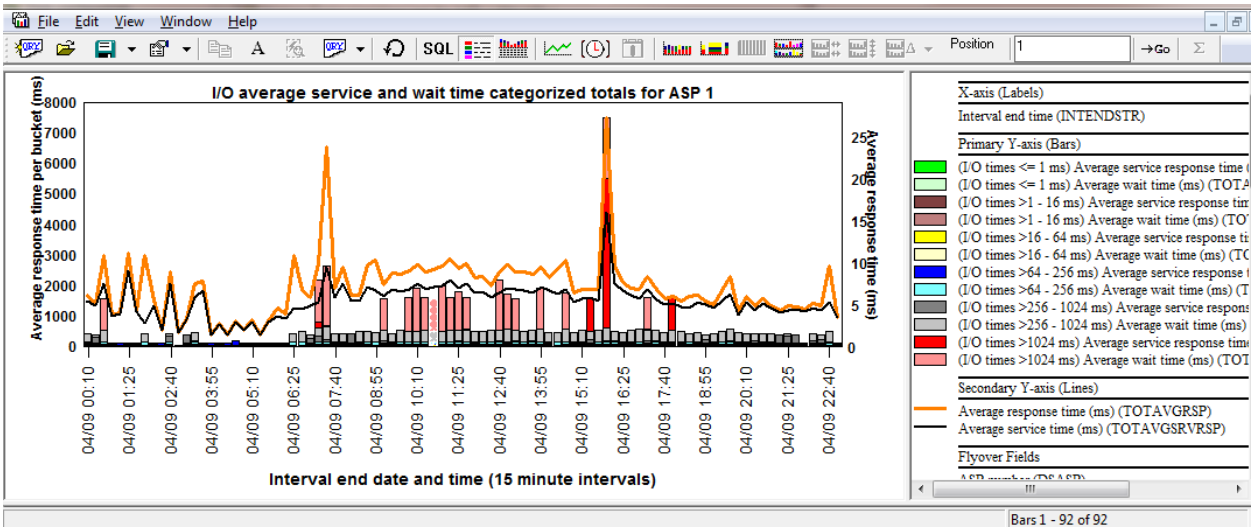
9.23.16 I/O average response time categorized totals for ASP <<DSASP>>



I/O average response time categorized totals for ASP 1

This graph displays the average response time on a per response time bucket basis. For that reason the smaller 3 buckets will rarely be visible on the graph.

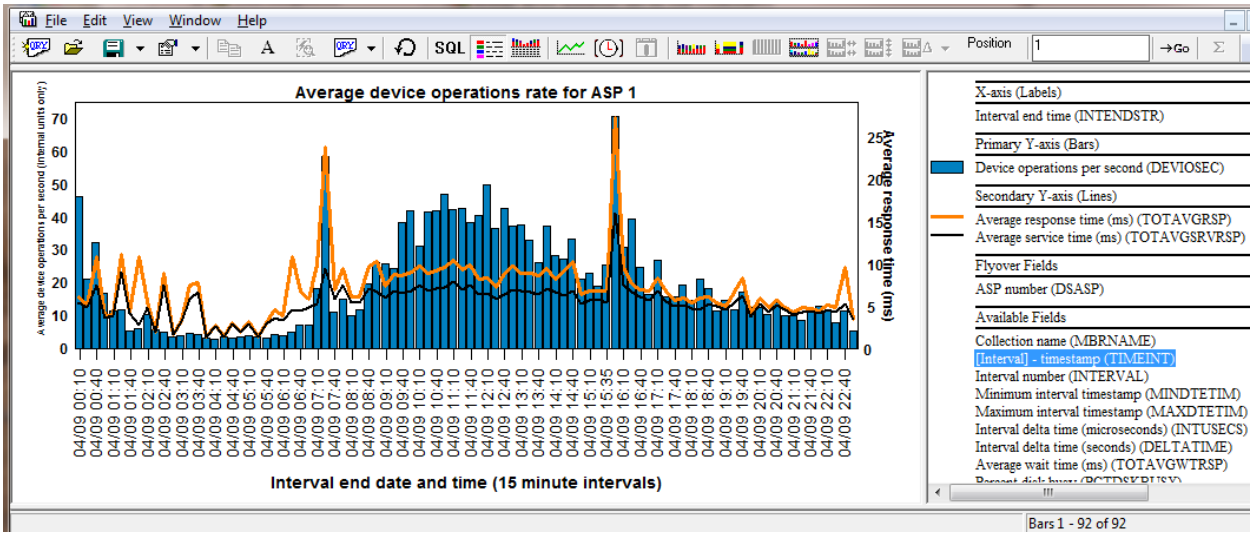
9.23.17 I/O avg service and wait time categorized totals for ASP <<DSASP>>



I/O avg service and wait time categorized totals for ASP 1

This graph displays the average response times and average service times on a per response time bucket basis. For that reason the smaller 3 buckets for both the service and response times will rarely be visible on the graph.

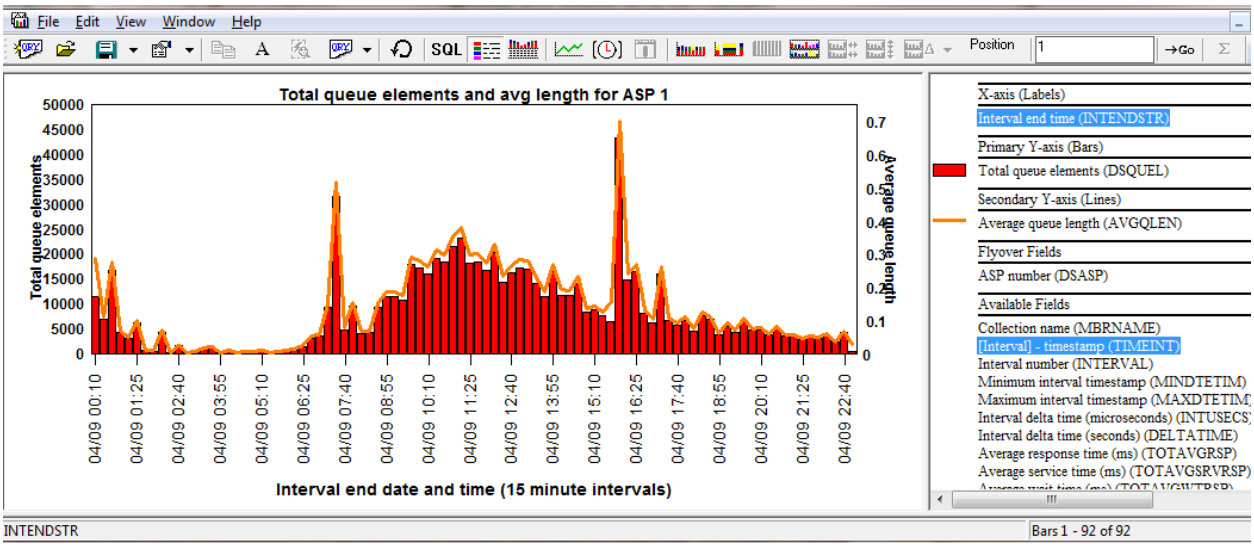
9.23.18 Average device operations rate for ASP <<DSASP>>



Average device operations rate for ASP 1

This graph shows the average device operations rate for all disk units within the specified ASPs.

9.23.19 Total queue elements and avg length for ASP <<DSASP>>



Total queue elements and avg length for ASP 1

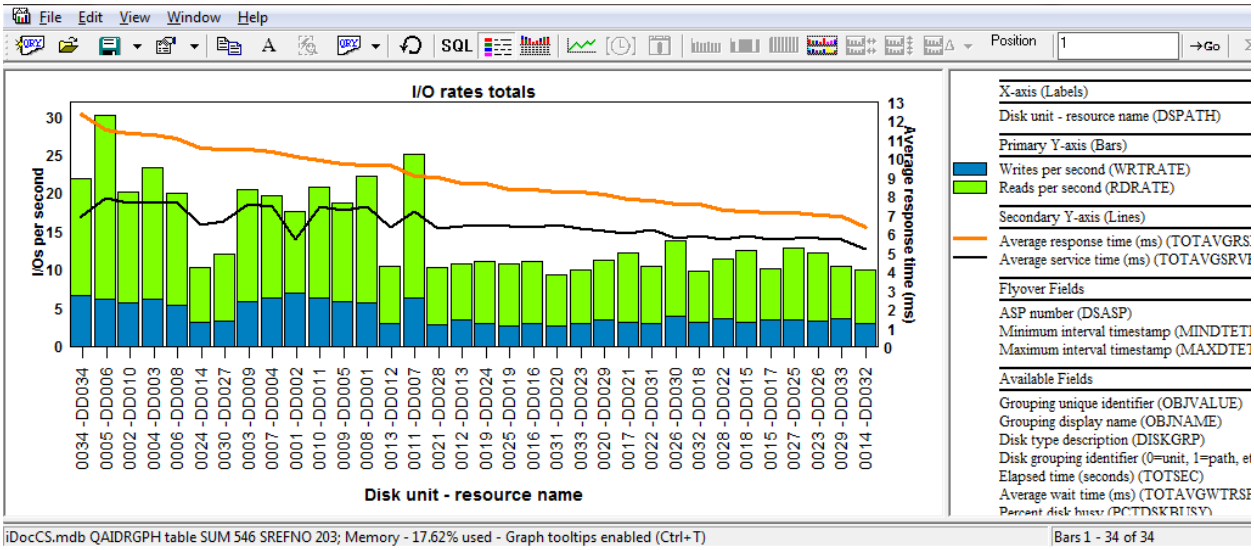
This graph shows the total queue elements and average queue length for all the disk units within the selected ASPs.

9.23.20 By disk path

These graphs are the same as the ones within the disk graphs folder over time, except these are ranking graphs grouped by the disk path (or disk unit and device resource name.)

Tip: You can drill down into any of these ranking graphs from the disk overview graphs by selecting the desired time period and right-clicking and choosing the 1st menu.

An example of this type of graph is shown below:



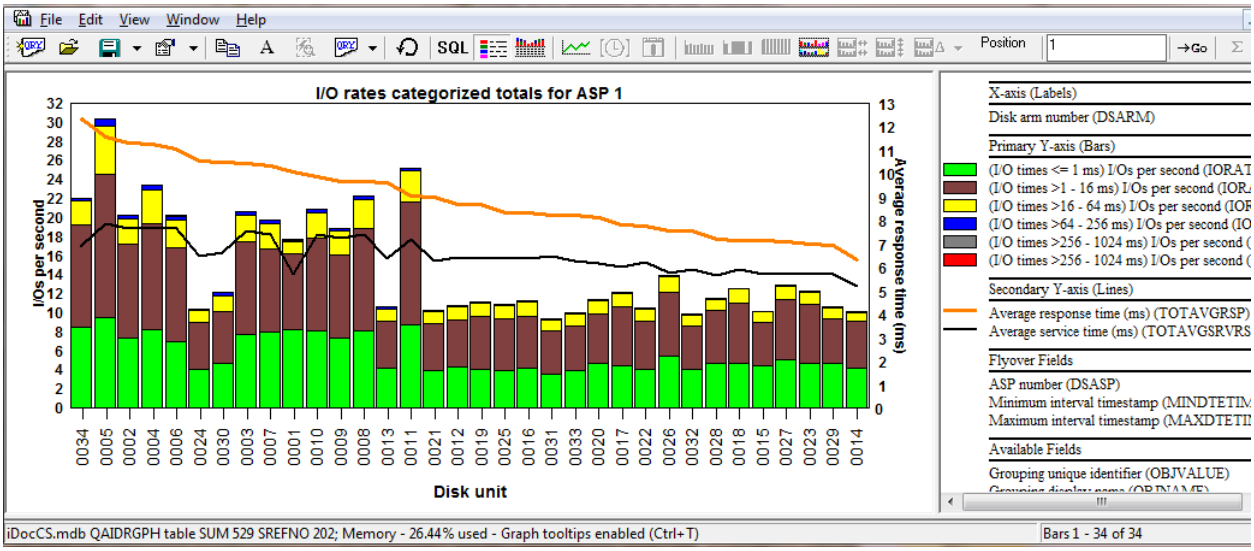
(By disk path) – I/O rates totals

9.23.21 By disk unit

These graphs are the same as the ones within the disk graphs folder over time, except these are ranking graphs grouped by the disk unit.

Tip: You can drill down into any of these ranking graphs from the disk overview graphs by selecting the desired time period and right-clicking then Rankings -> Disk graphs -> by disk unit.

An example of this type of graph is shown below:



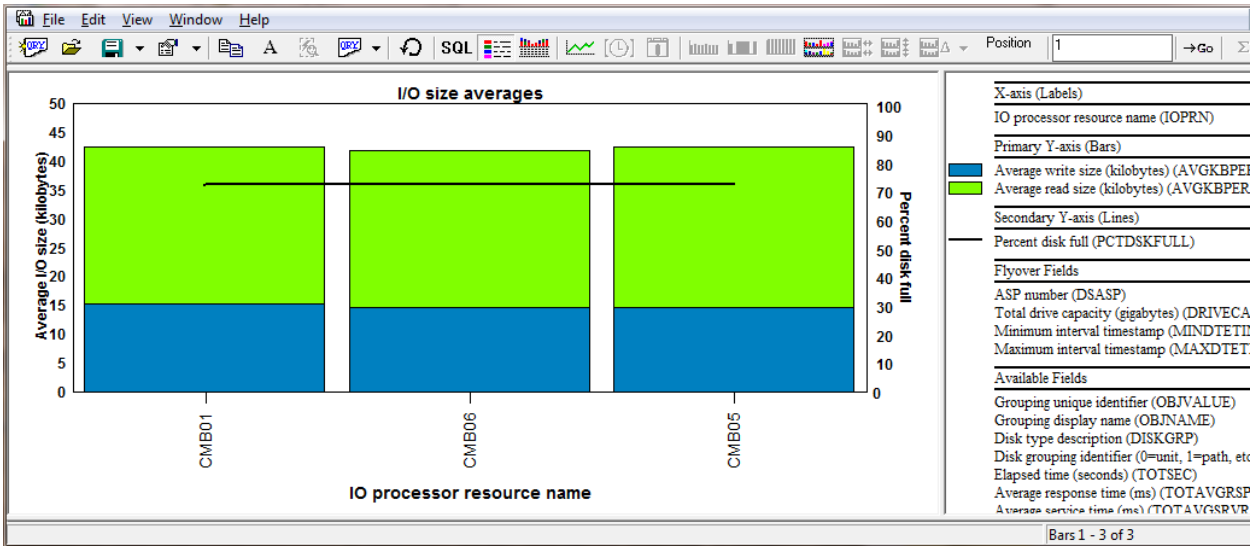
(By disk unit) – I/O rates categorized totals for ASP 1

9.23.22 By I/O processor

These graphs are the same as the ones within the disk graphs folder over time, except these are ranking graphs grouped by I/O processor.

Tip: You can drill down into any of these ranking graphs from the disk overview graphs by selecting the desired time period and right-clicking then Rankings -> Disk graphs -> by I/O processor.

An example of this type of graph is shown below:



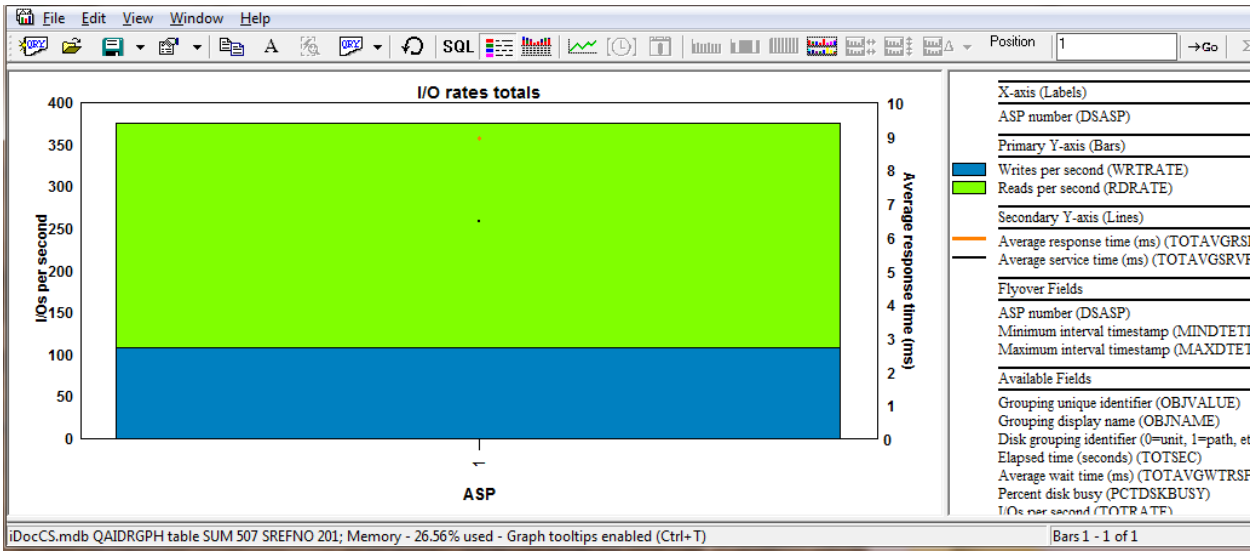
(By I/O processor) – I/O size averages

9.23.23 By ASP

These graphs are the same as the ones within the disk graphs folder over time, except these are ranking graphs grouped by ASP.

Tip: You can drill down into any of these ranking graphs from the disk overview graphs by selecting the desired time period and right-clicking then Rankings -> Disk graphs -> by ASP.

An example of this type of graph is shown below:



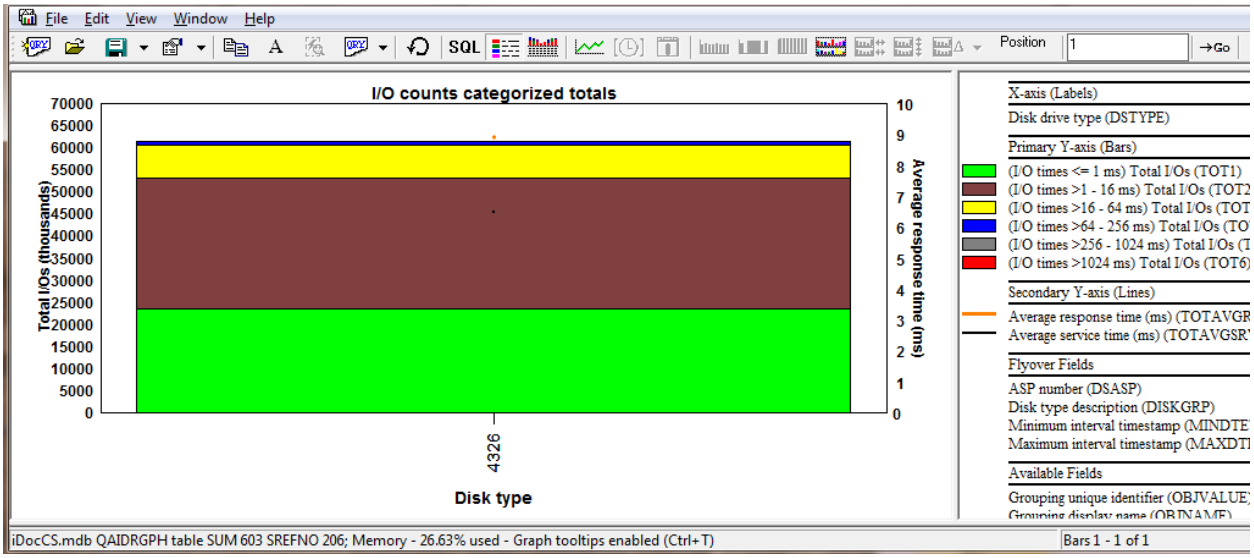
(By ASP) – I/O rates totals

9.23.24 By disk type

These graphs are the same as the ones within the disk graphs folder over time, except these are ranking graphs grouped by disk type.

Tip: You can drill down into any of these ranking graphs from the disk overview graphs by selecting the desired time period and right-clicking then Rankings -> Disk graphs -> by disk type.

An example of this type of graph is shown below:



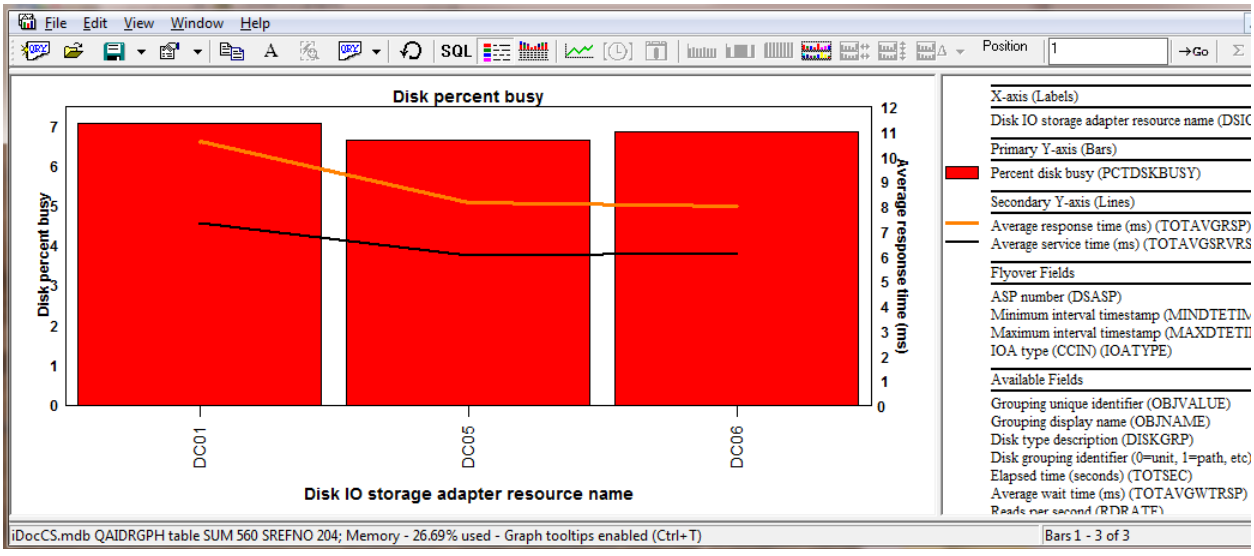
(By disk type) – I/O counts categorized totals

9.23.25 By I/O adapter (6.1+)

These graphs are the same as the ones within the disk graphs folder over time, except these are ranking graphs grouped by I/O adapter (IOA.)

Tip: You can drill down into any of these ranking graphs from the disk overview graphs by selecting the desired time period and right-clicking then Rankings -> Disk graphs -> by I/O adapter

An example of this type of graph is shown below:



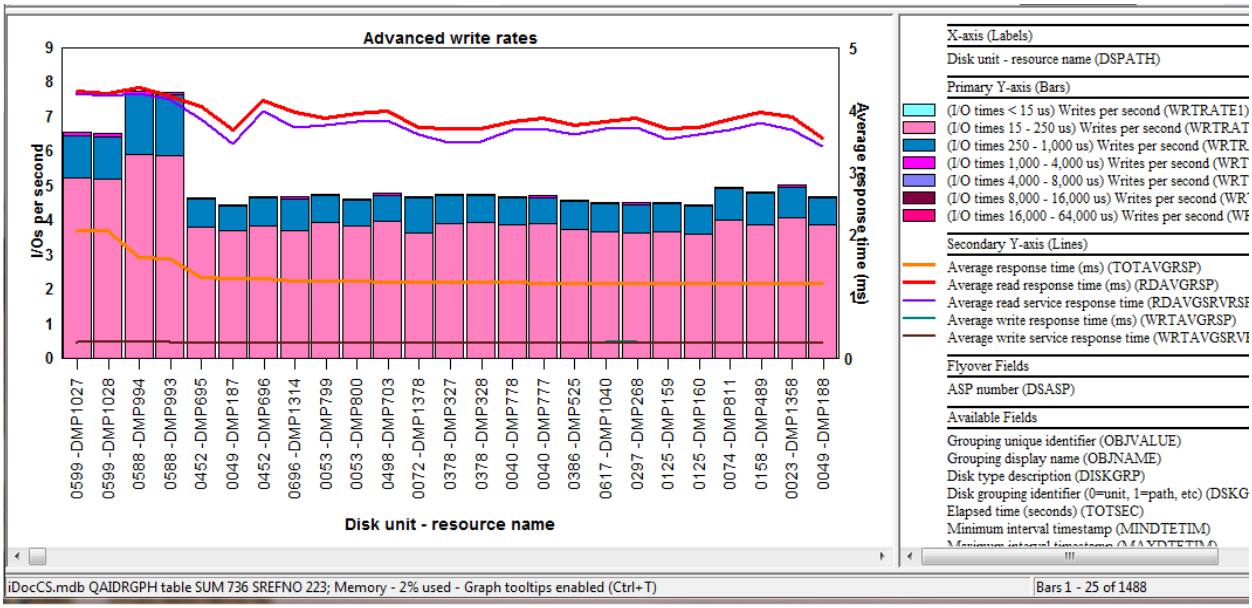
(By I/O adapter) – Disk percent busy

9.23.26 By disk path

These graphs are the same as the set of graphs within the Disk graphs -> advanced folder except they are grouped by disk path (or disk unit and device resource name.)

Tip: You can drill down into any of these ranking graphs from any of the disk graphs -> advanced folder's graphs by selecting the desired time period and right-clicking and choosing the 1st menu.

An example of this type of graph is shown below:



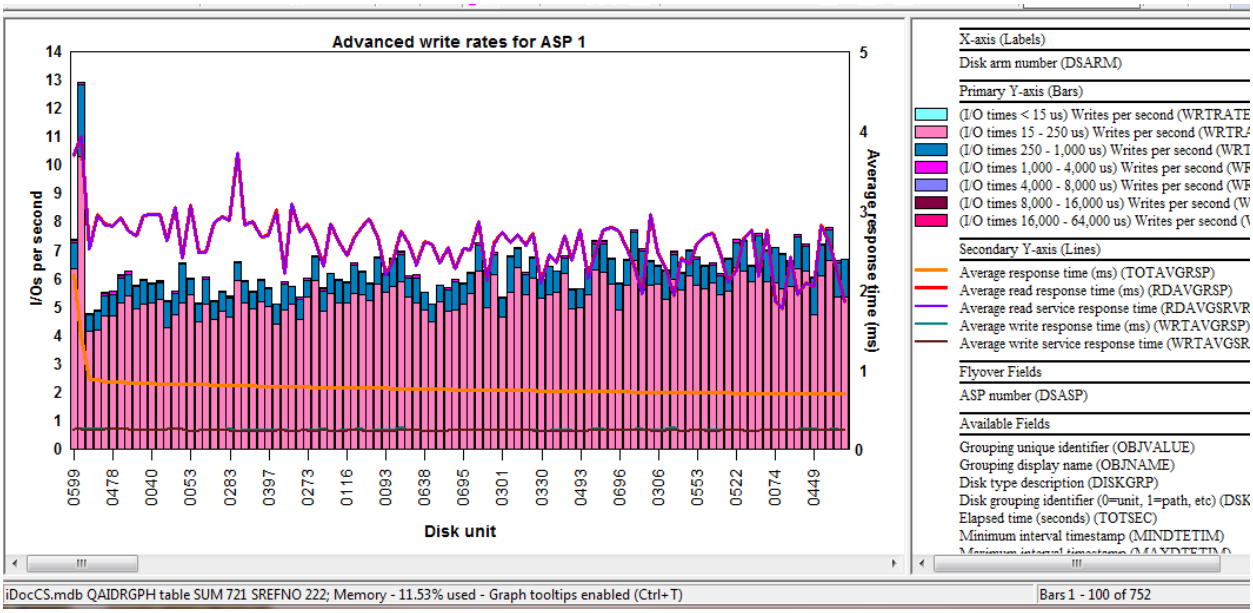
(By disk path) – Advanced write rates

9.23.27 By disk unit

These graphs are the same as the set of graphs within the Disk graphs -> advanced folder except they are grouped by disk unit.

Tip: You can drill down into any of these ranking graphs from any of the disk graphs -> advanced folder's graphs by selecting the desired time period and right-clicking and choosing the Rankings -> disk graphs -> advanced -> by disk unit menu.

An example of this type of graph is shown below:



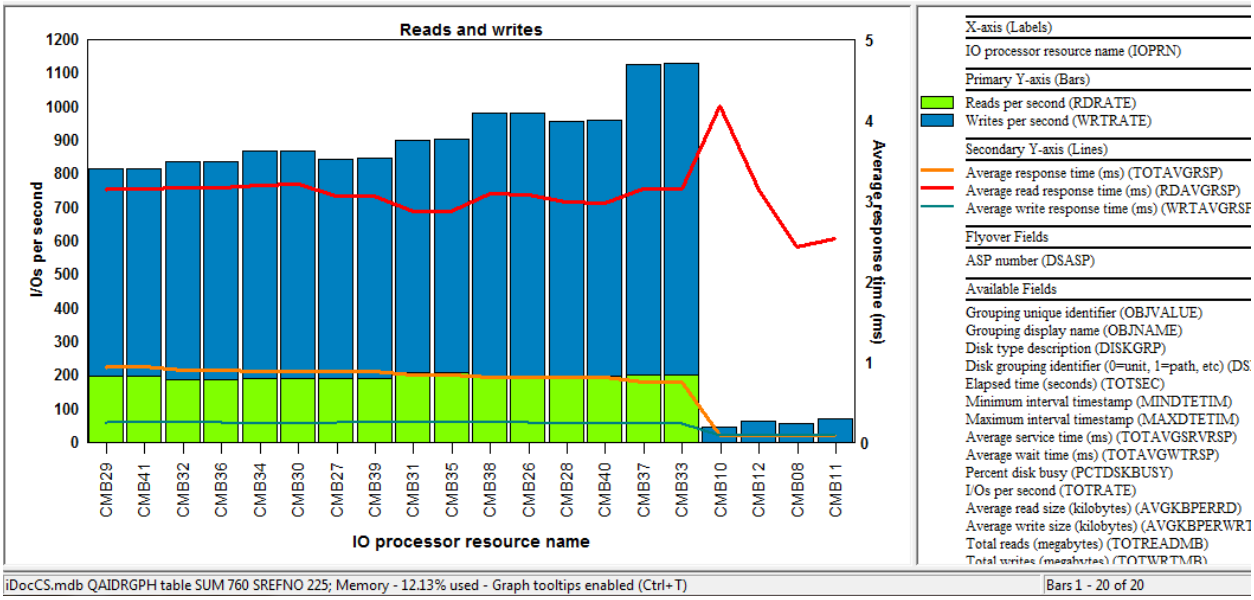
(By disk unit) – Advanced write rates

9.23.28 By I/O processor

These graphs are the same as the set of graphs within the Disk graphs -> advanced folder except they are grouped by I/O processor (IOP.)

Tip: You can drill down into any of these ranking graphs from any of the disk graphs -> advanced folder's graphs by selecting the desired time period and right-clicking and choosing the Rankings -> disk graphs -> advanced -> by I/O processor menu.

An example of this type of graph is shown below:



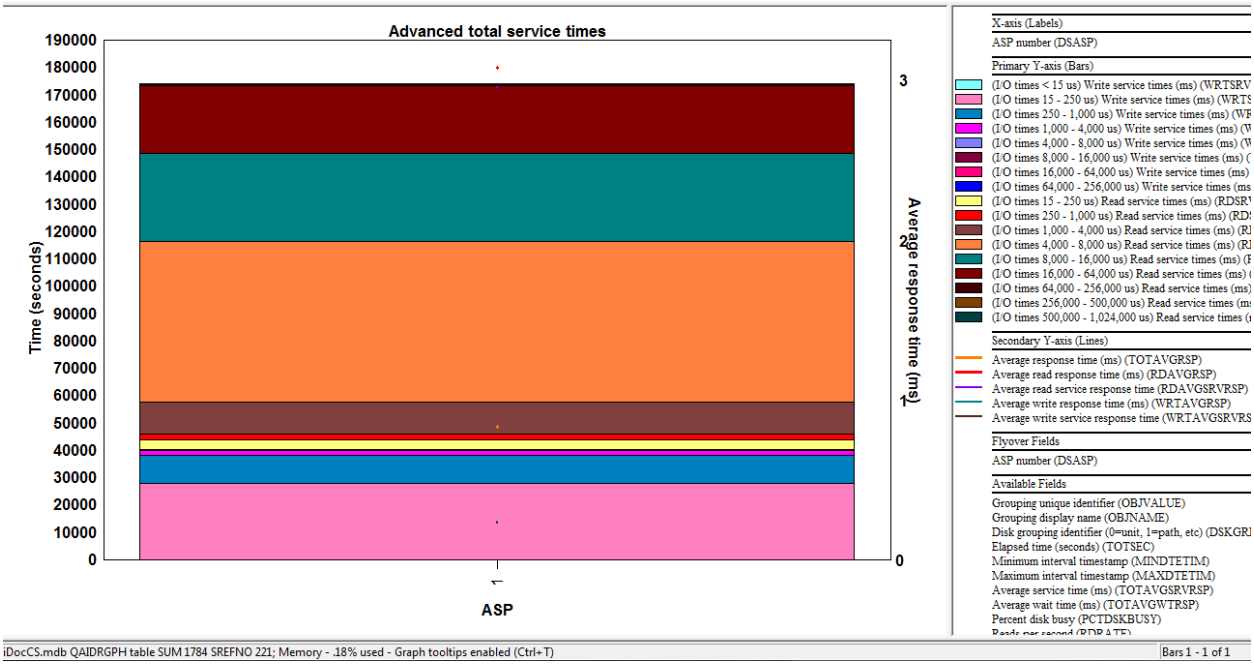
(By I/O processor) – Reads and writes

9.23.29 By ASP

These graphs are the same as the set of graphs within the Disk graphs -> advanced folder except they are grouped by ASP.

Tip: You can drill down into any of these ranking graphs from any of the disk graphs -> advanced folder's graphs by selecting the desired time period and right-clicking and choosing the Rankings -> disk graphs -> advanced -> by ASP menu.

An example of this type of graph is shown below:



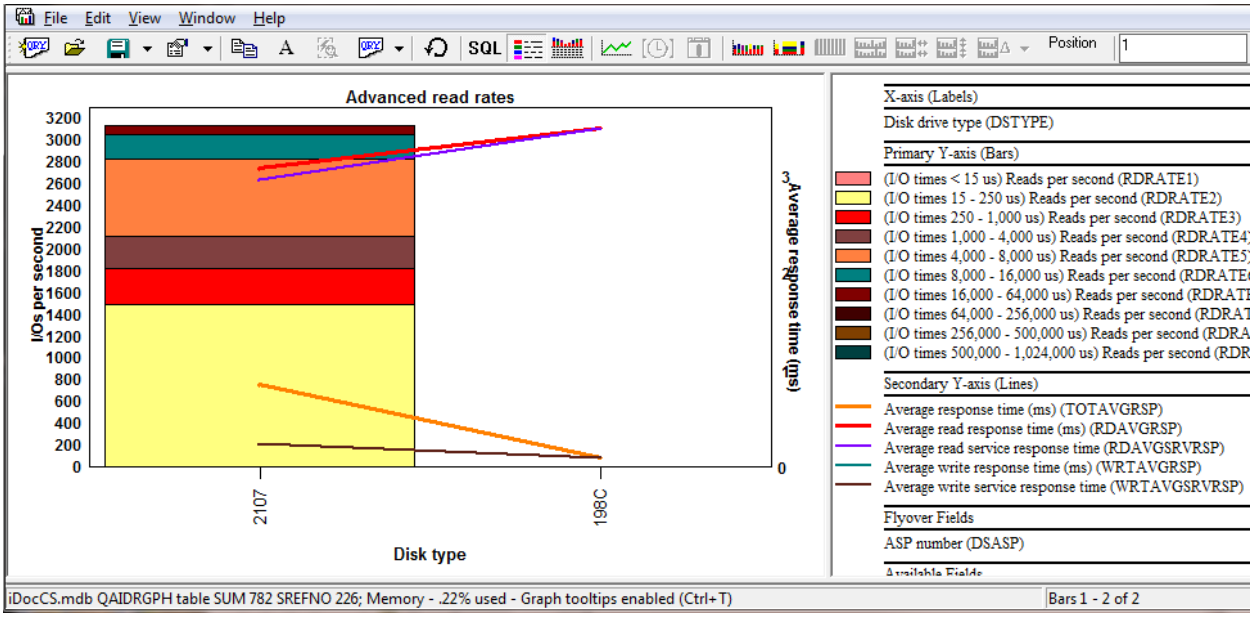
(By ASP) – Advanced total service times

9.23.30 By disk type

These graphs are the same as the set of graphs within the Disk graphs -> advanced folder except they are grouped by disk type.

Tip: You can drill down into any of these ranking graphs from any of the disk graphs -> advanced folder's graphs by selecting the desired time period and right-clicking and choosing the Rankings -> disk graphs -> advanced -> by disk type menu.

An example of this type of graph is shown below:



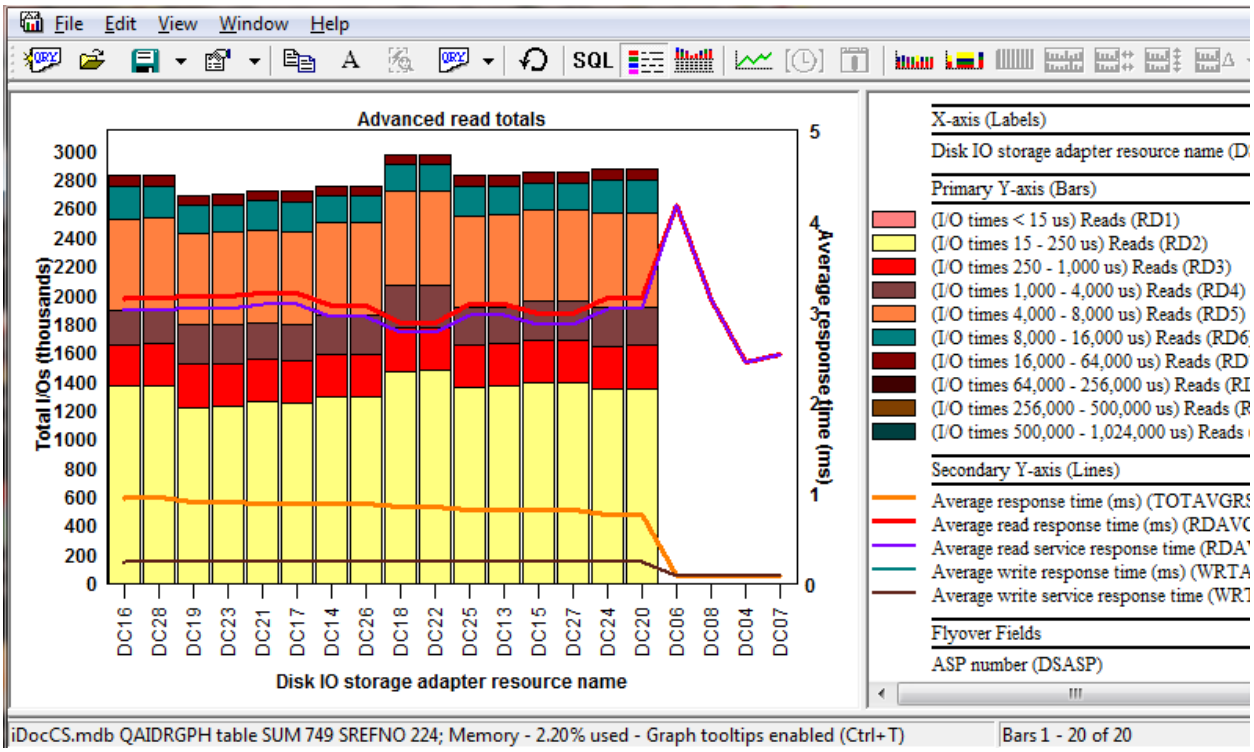
(By disk type) – Advanced read rates

9.23.31 By I/O adapter

These graphs are the same as the set of graphs within the Disk graphs -> advanced folder except they are grouped by I/O adapter (IOA.)

Tip: You can drill down into any of these ranking graphs from any of the disk graphs -> advanced folder's graphs by selecting the desired time period and right-clicking and choosing the Rankings -> disk graphs -> advanced -> by I/O adapter menu.

An example of this type of graph is shown below:

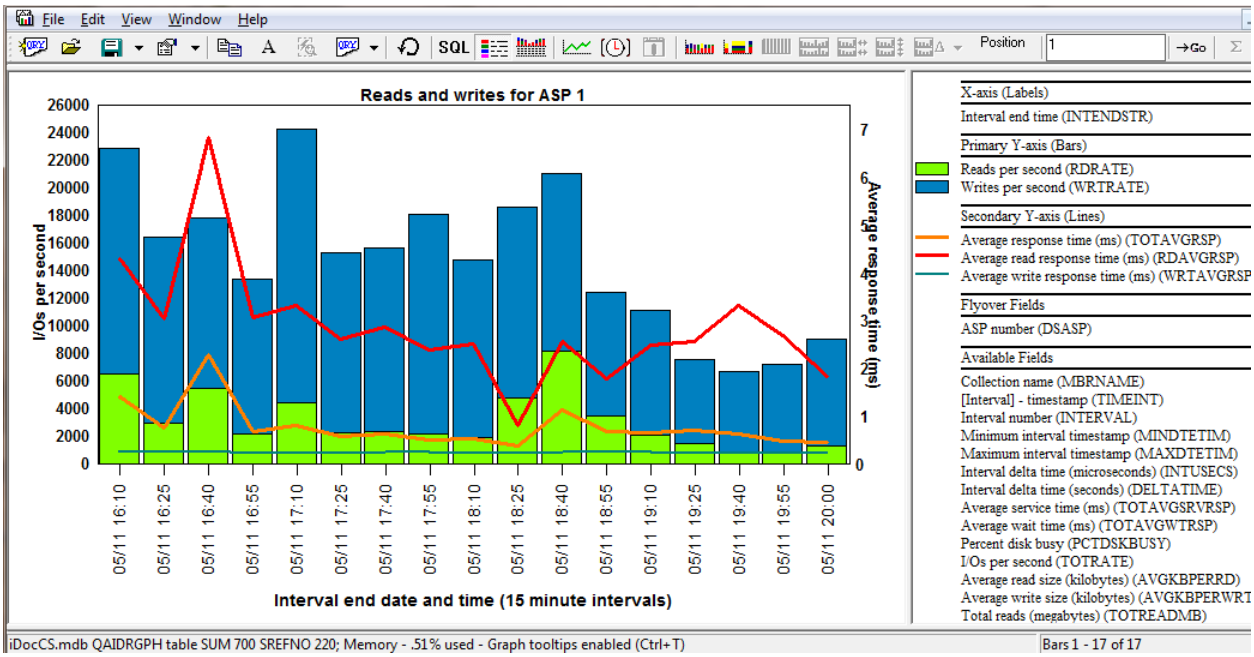


(By I/O adapter) – Advanced read totals

9.23.32 Advanced

These graphs are only available if the 7.1 Collection Services file QAPMDISKRB has been captured. The focus on these graphs is to show response times for both reads and writes. Some of the graphs (those that start with the word "advanced" display 11 response time buckets for reads and/or writes.

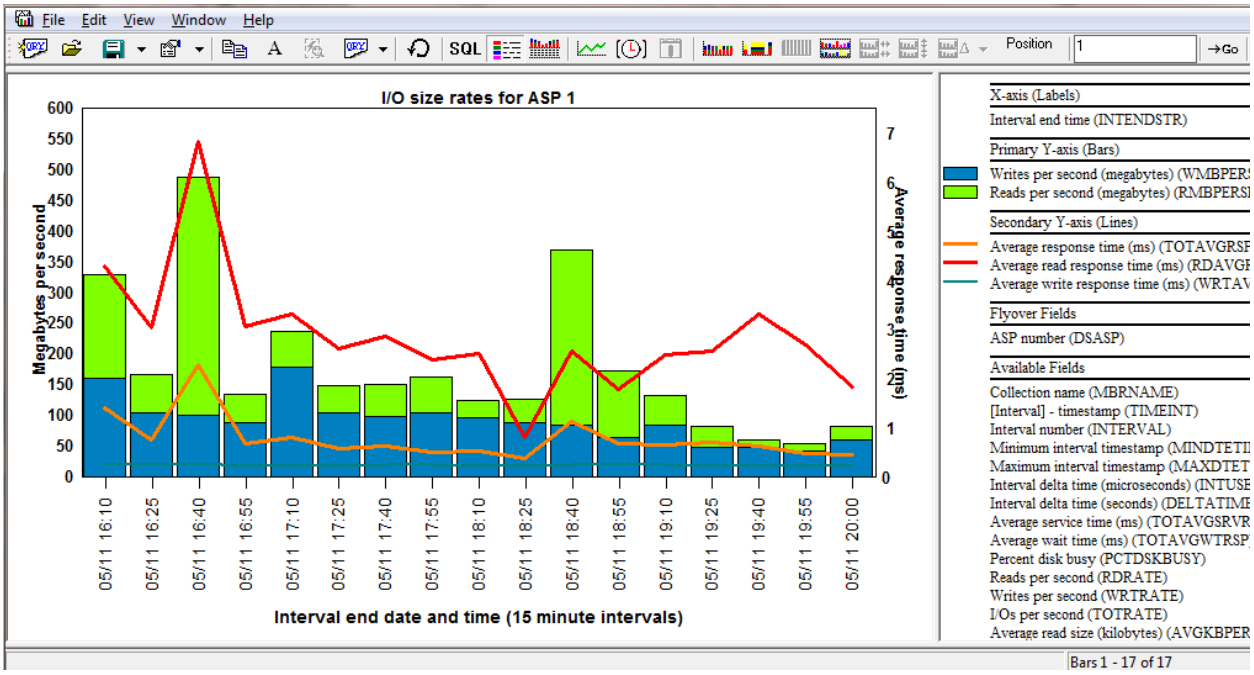
9.23.32.1 Reads and writes for ASP <<DSASP>>



Reads and writes for ASP 1

This graph displays the reads per second, writes per second along with read and write response times on the 2nd Y-axis. The average of both is also shown as the orange line.

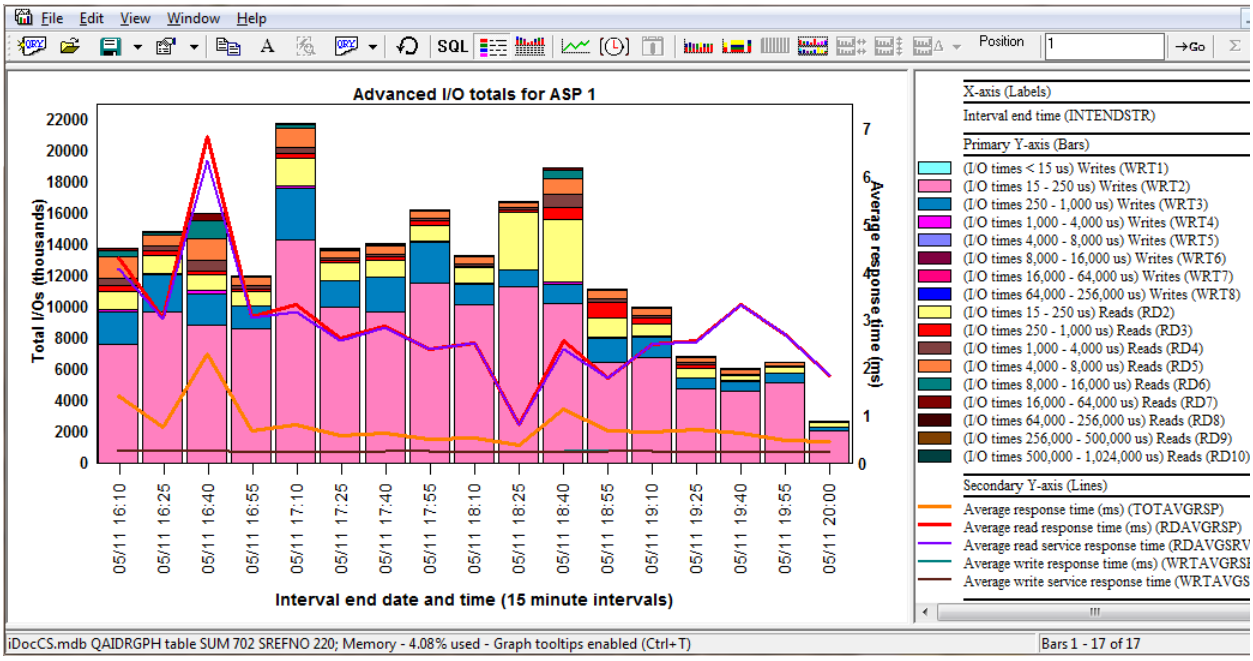
9.23.32.2 I/O size rates for ASP <<DSASP>>



I/O size rates for ASP 1

This graph shows the writes per second and reads per second (in megabytes) along with the response times on the 2nd Y-axis.

9.23.32.3 Advanced I/O totals for ASP <<DSASP>>



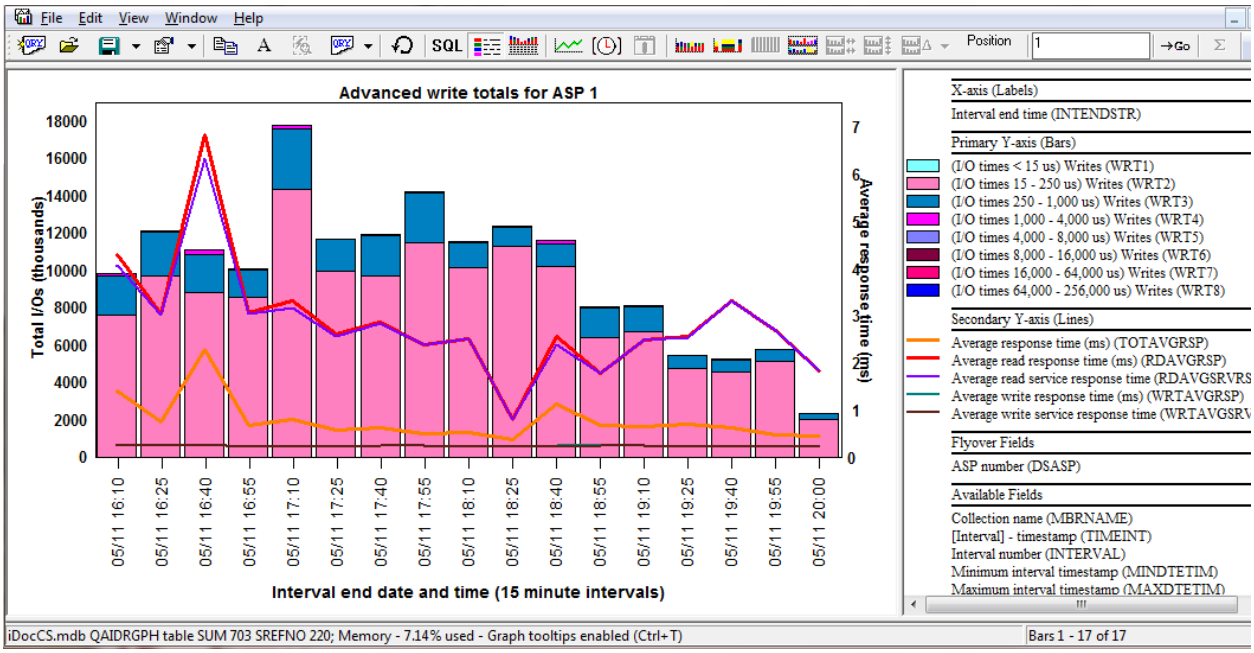
Advanced I/O totals for ASP 1

This graph displays total I/O counts by grouping the counts into 11 response time buckets for both reads and writes. The 2nd Y-axis displays the following 5 response time values:

1. Average response time (overall)
2. Average read response time
3. Average read service response time
4. Average write response time
5. Average write service response time

Tip: The legend will most likely not show all 11 buckets for both reads and writes. By design the graph legend on the primary Y-axis only shows fields with non-zero values.

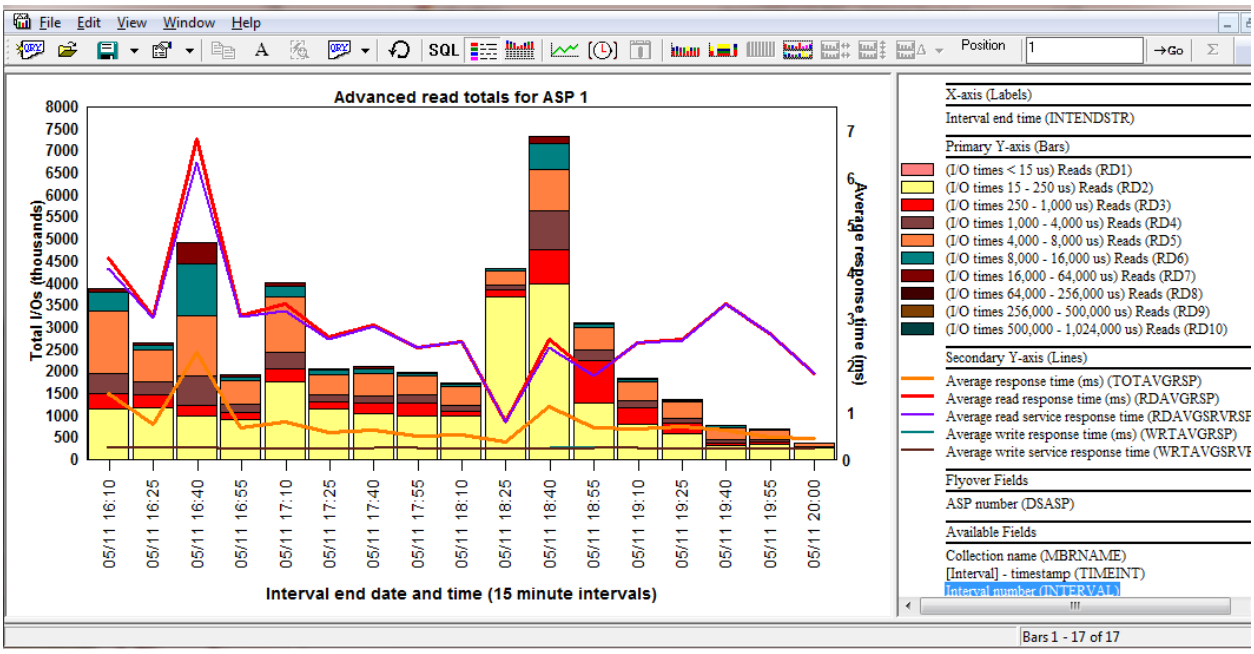
9.23.32.4 Advanced write totals for ASP <<DSASP>>



Advanced write totals for ASP 1

This graph is the same as the previous one except it only shows the write buckets on the Primary Y-axis.

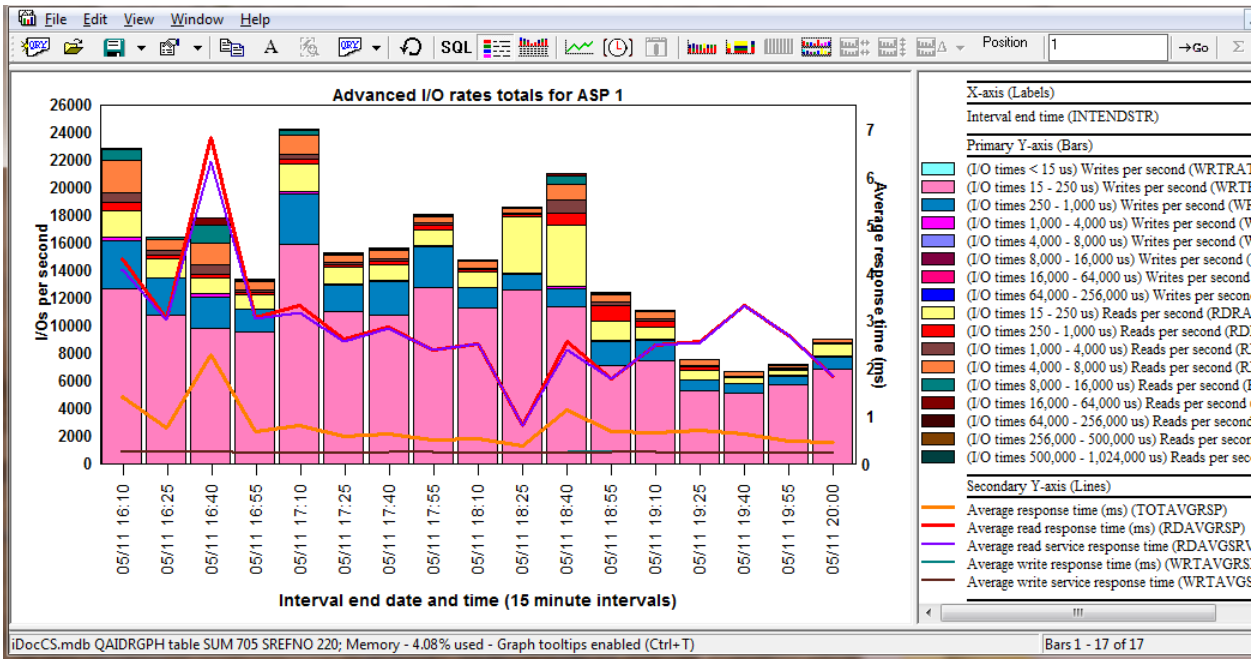
9.23.32.5 Advanced read totals for ASP <<DSASP>>



Advanced I/O reads for ASP <<DSASP>>

This graph is the same as the Advanced I/O totals graph except it only shows the read buckets.

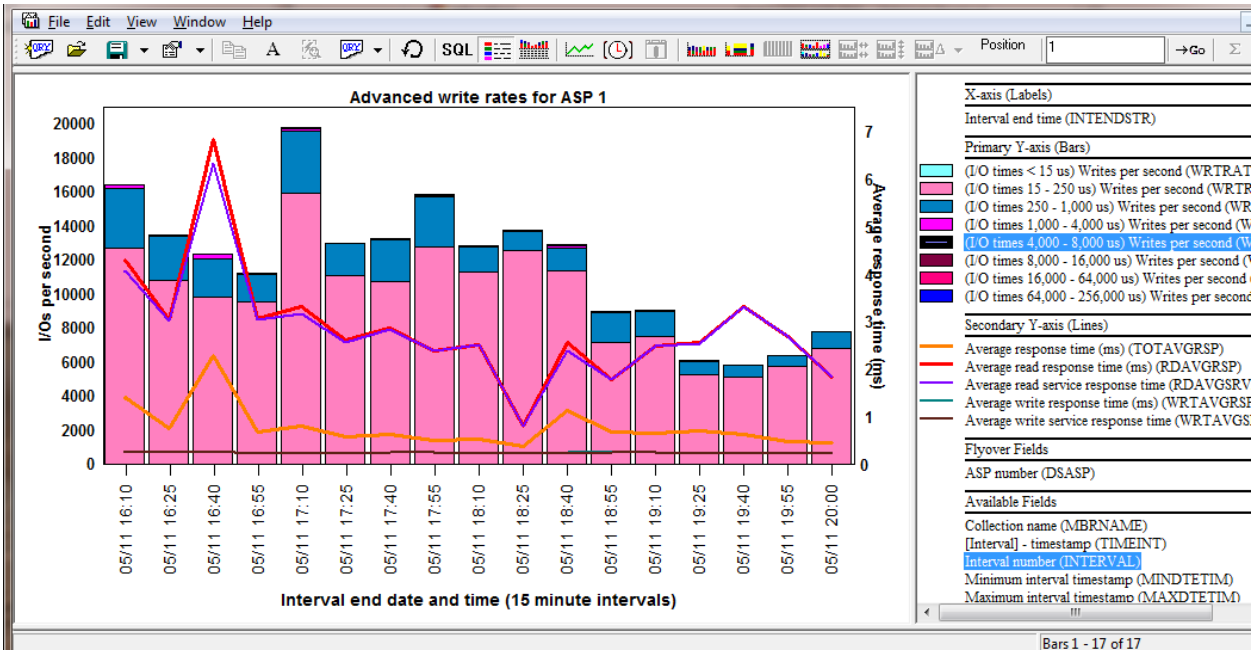
9.23.32.6 Advanced I/O rates totals for ASP <<DSASP>>



Advanced I/O rates totals for ASP 1

This graph displays the I/Os per second for both reads and writes using 11 response time buckets.

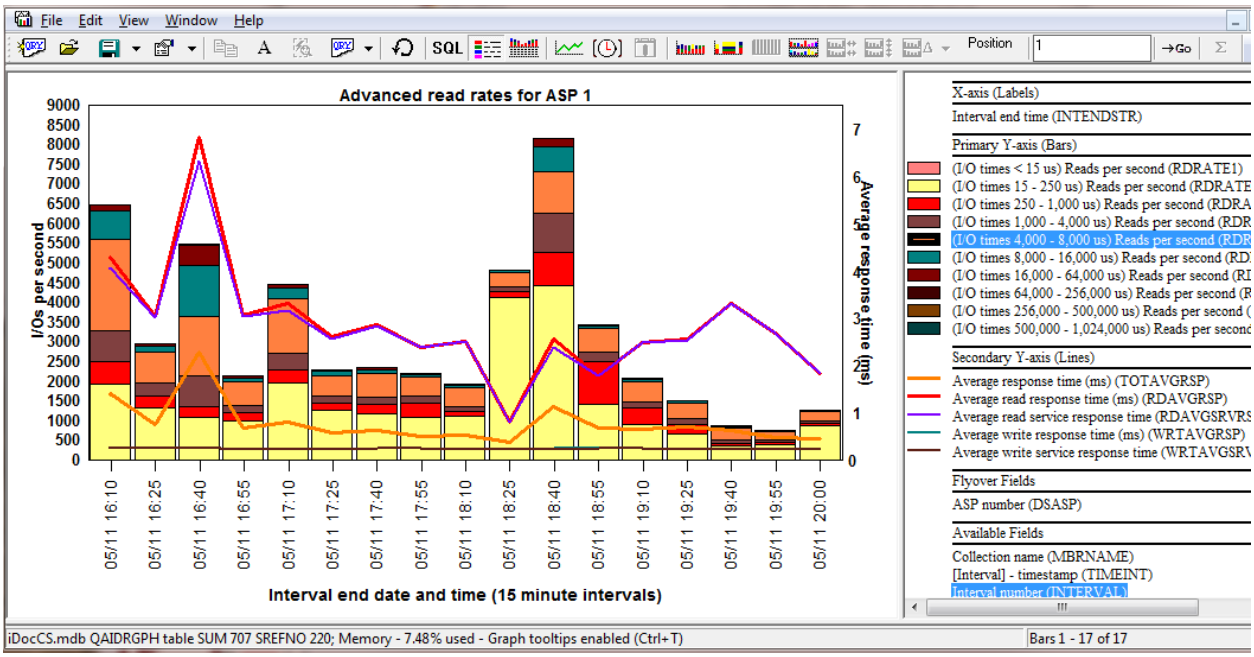
9.23.32.7 Advanced write rates for ASP <<DSASP>>



Advanced write rates for ASP <<DSASP>>

This graph is the same as the previous one except it only shows the write response time buckets.

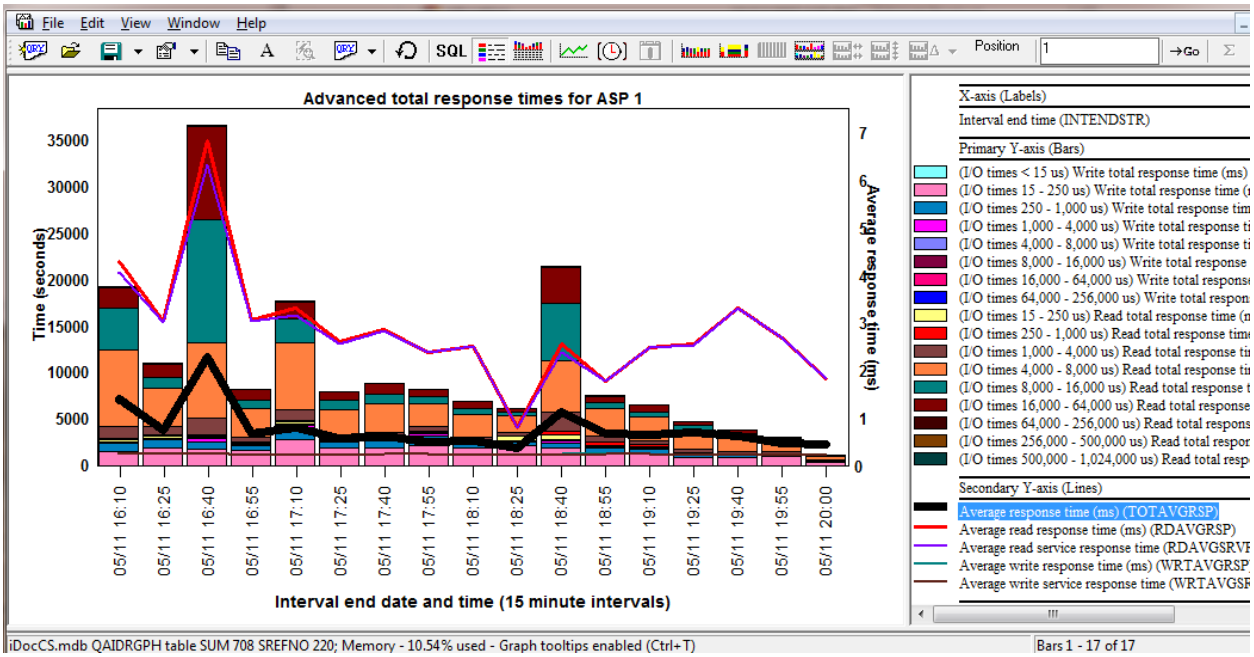
9.23.32.8 Advanced read rates for ASP <<DSASP>>



Advanced read rates for ASP

This graph is the same as the Advanced I/O rates totals graph except it only shows the read buckets.

9.23.32.9 Advanced total response times for ASP <<DSASP>>



Advanced total response times for ASP 1

This graph adds up the total response time for all 11 buckets for both reads and writes.

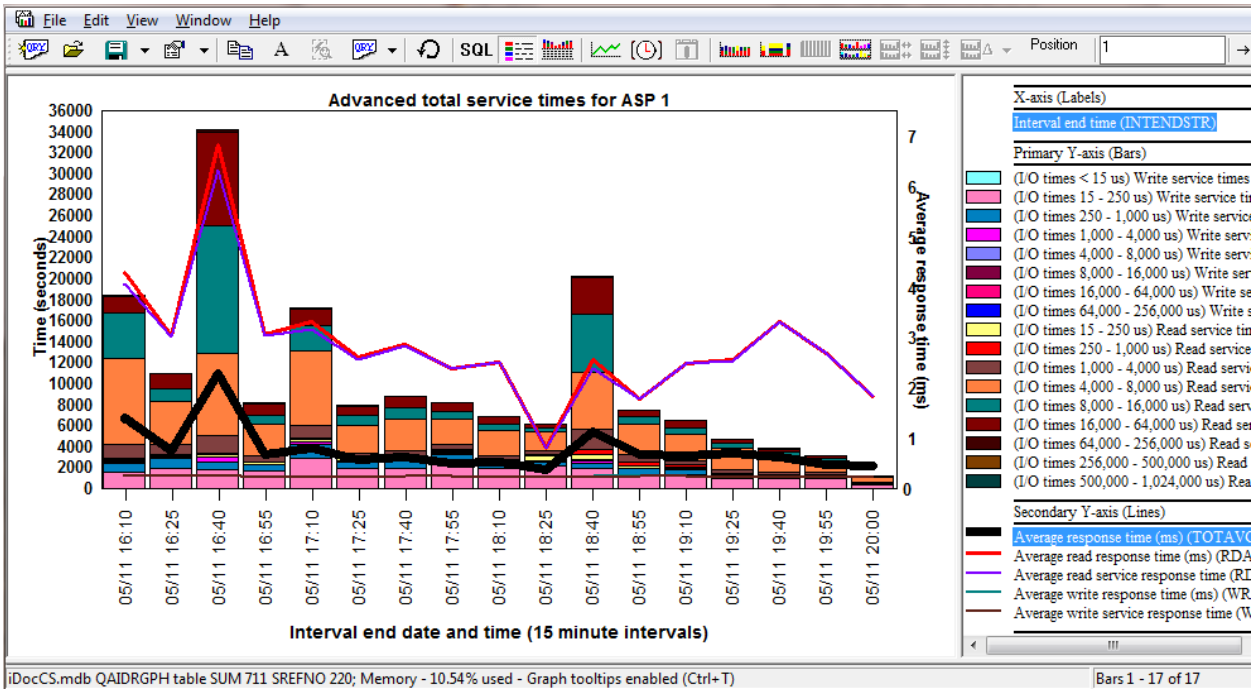
9.23.32.10 Advanced write response times for ASP <<DSASP>>

This graph is the same as the previous graph except only shows the write response times.

9.23.32.11 Advanced read response times for ASP <<DSASP>>

This graph is the same as the previous graph except only shows the read response times.

9.23.32.12 Advanced total service times for ASP <<DSASP>>



Advanced total service times for ASP 1

This graph displays the total service time all 11 response time buckets for both reads and writes.

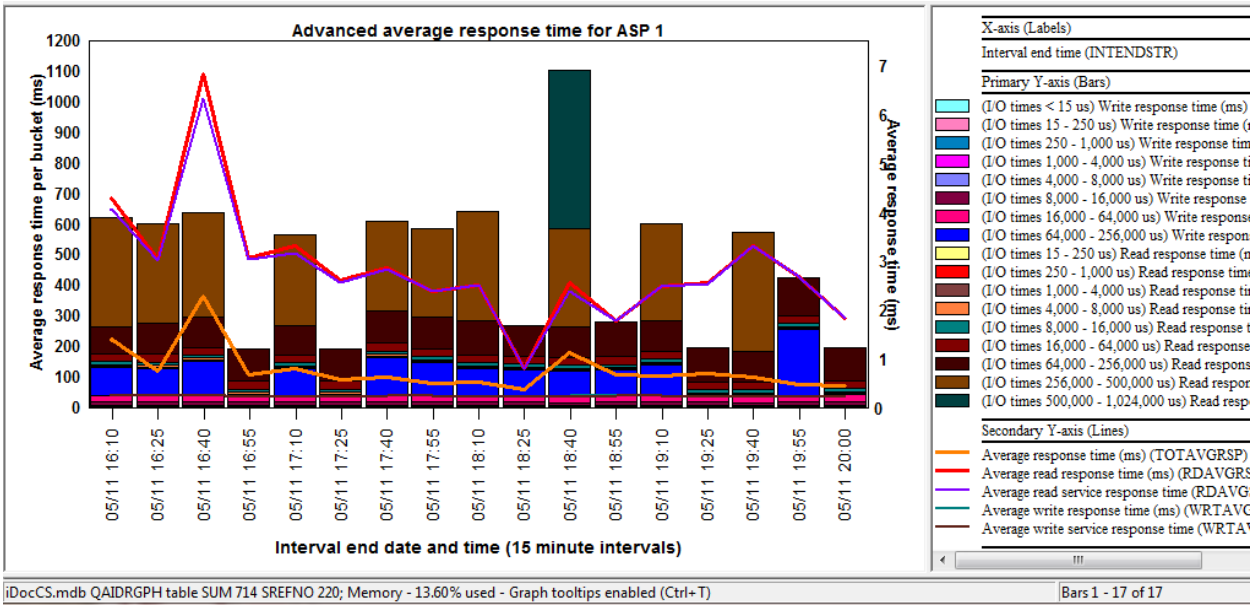
9.23.32.13 Advanced write service times for ASP <<DSASP>>

This graph is the same as the previous graph except only shows the write service times.

9.23.32.14 Advanced read service times for ASP <<DSASP>>

This graph is the same as the previous graph except only shows the read service times.

9.23.32.15 Advanced average response time for ASP <<DSASP>>



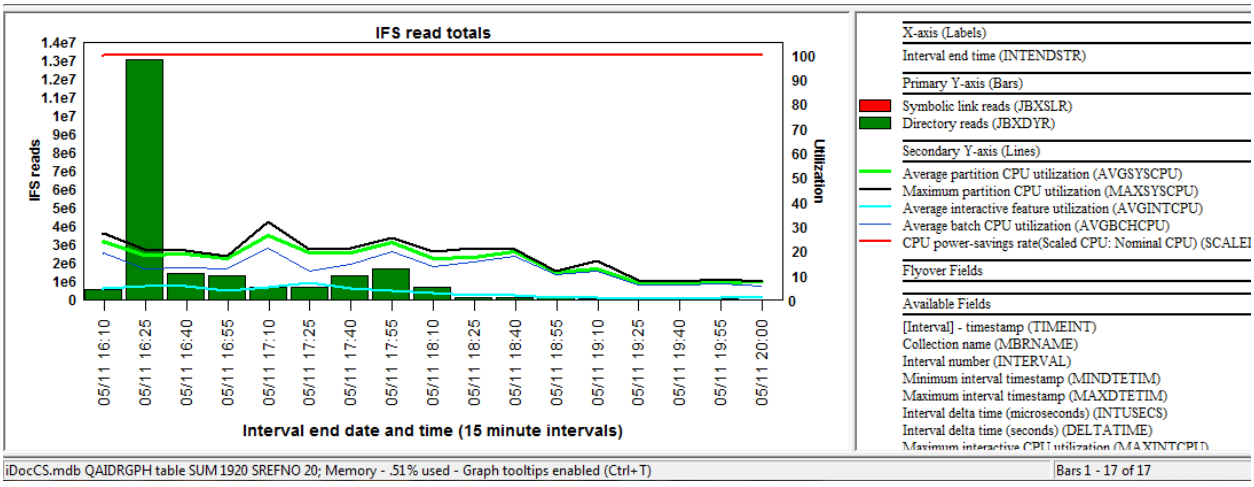
Advanced average response time for ASP 1

This graph displays the average response times within each of the 22 response time buckets. 11 buckets for reads and 11 for writes.

9.23.33 IFS graphs

The IFS graphs are essentially the same as the set of IFS graphs available in Job Watcher. The provide IFS-related statistics for reads, lookup cache hits/misses, opens and directory creates/deletes.

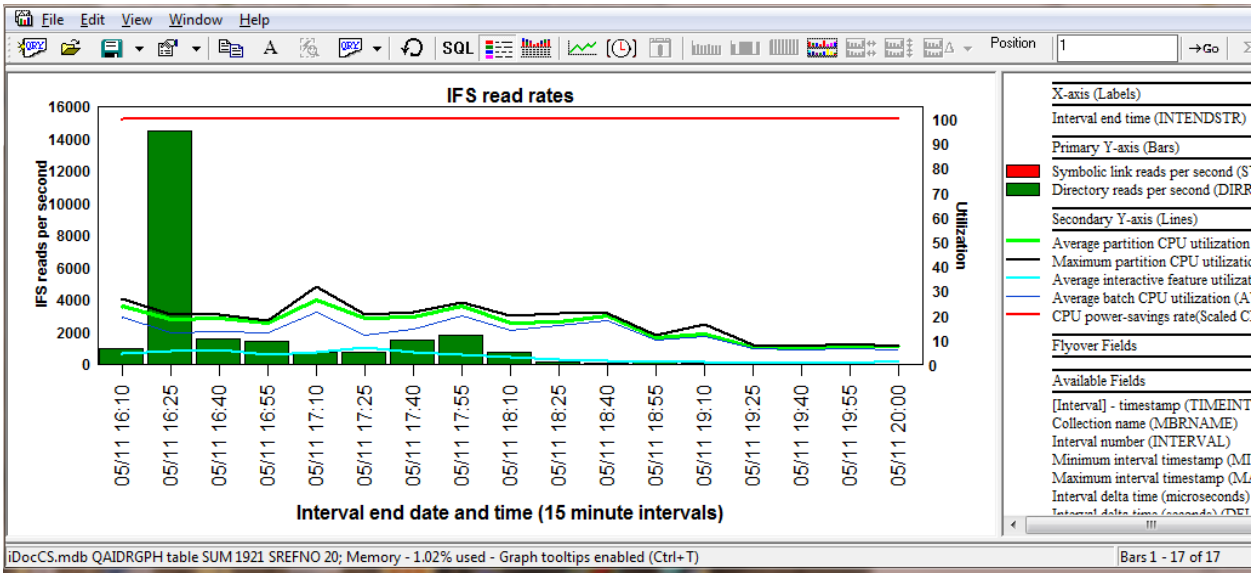
9.23.33.1 IFS read totals



IFS read totals

This graph displays the total IFS symbolic link reads and directory reads on the primary Y-axis and the CPU utilization fields on the 2nd Y-axis.

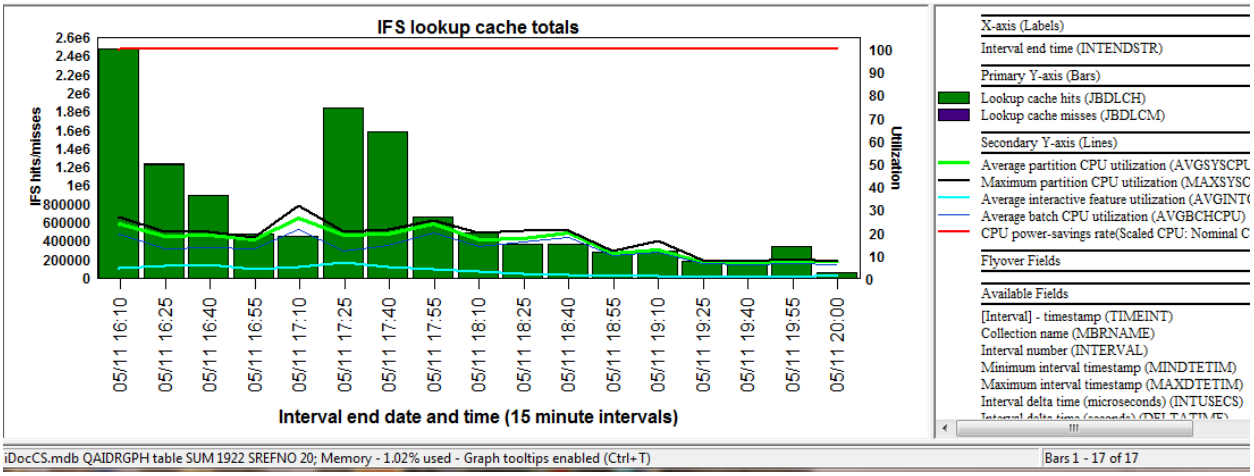
9.23.33.2 IFS read rates



IFS read rates

This graph is the same as the previous one except the values are shown as a rate per second.

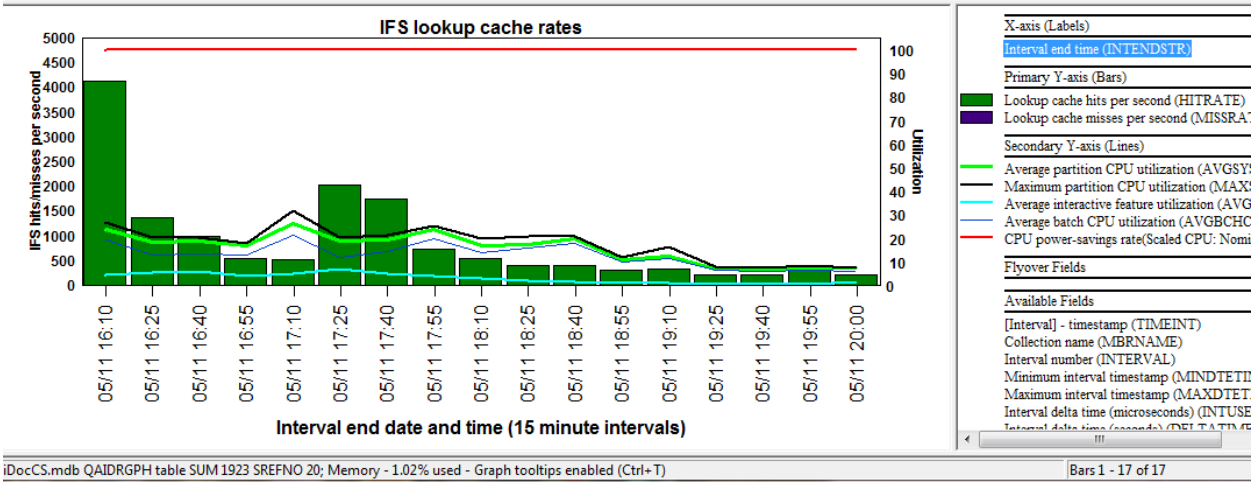
9.23.33.3 IFS lookup cache totals



IFS lookup cache totals

This graph displays the total IFS lookup cache hits and misses over time. The 2nd Y-axis displays CPU utilization.

9.23.33.4 IFS lookup cache rates



IFS lookup cache rates

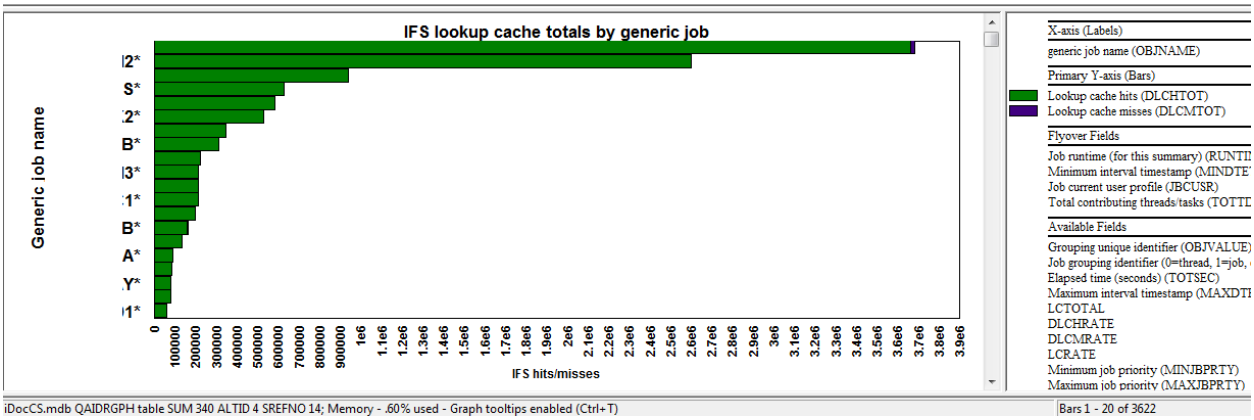
This graph is the same as the previous one except the values on the Primary Y-axis are shown as a rate per second.

9.23.34 IFS ranking graphs

A set of IFS rankings graphs are provided within 8 subfolders of the IFS graphs folder. Each folder contains the 8 possible job groupings to choose from. The following ranking graph folders are available:

- IFS read totals rankings
- IFS read rates rankings
- IFS lookup cache totals rankings
- IFS lookup cache rates rankings
- IFS open totals rankings
- IFS open rates rankings
- IFS create/delete totals rankings
- IFS create/delete rates rankings

An example is shown below:

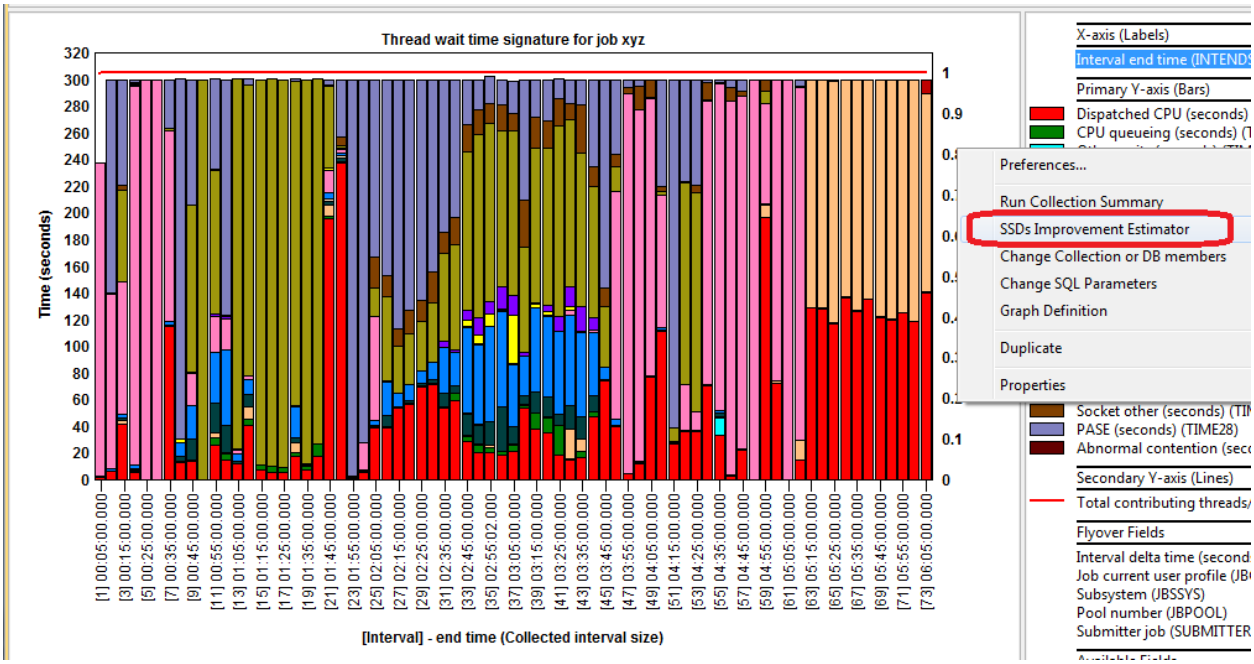


IFS lookup cache totals by generic job

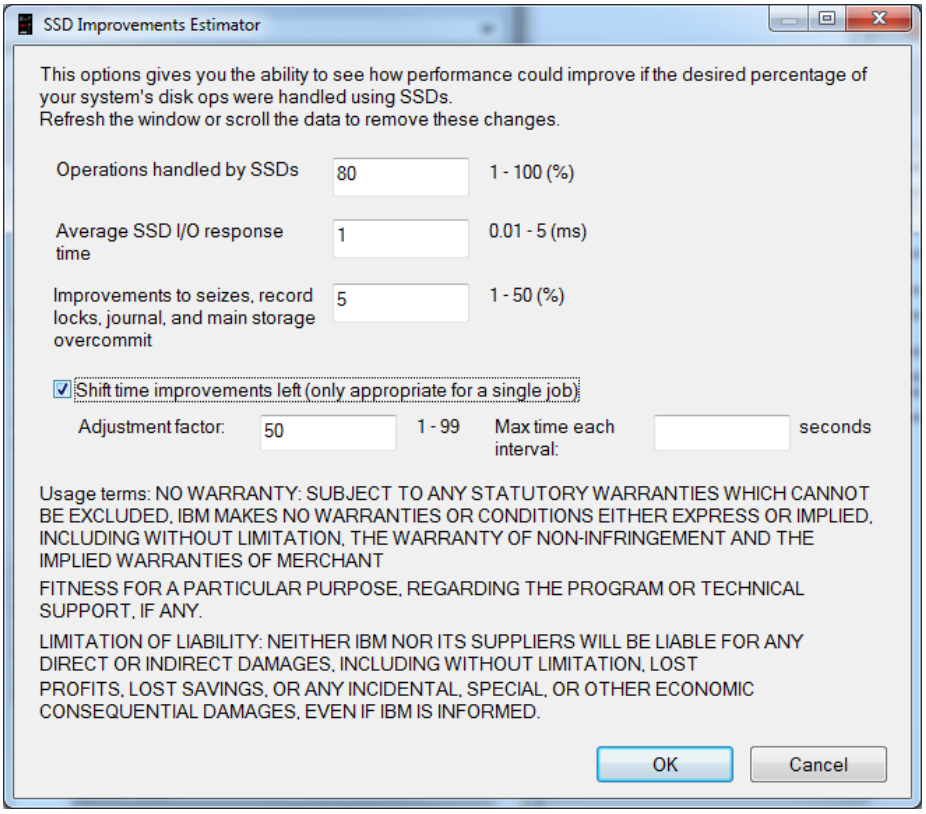
9.24 SSD Improvements Estimator

The SSD Improvements Estimator is an option in Collection Services Investigator that can be used to see how wait bucket graphs might look if SSDs were added to the system. This option can be used on any wait bucket time-based graph in Collection Services Investigator, but is typically used on either the Collection Overview Time Signature or on the Thread Wait Time Signature for job xyz graphs.

To access this function, when the desired graph is opened, right-click on the graph and use the SSDs Improvements Estimator menu.



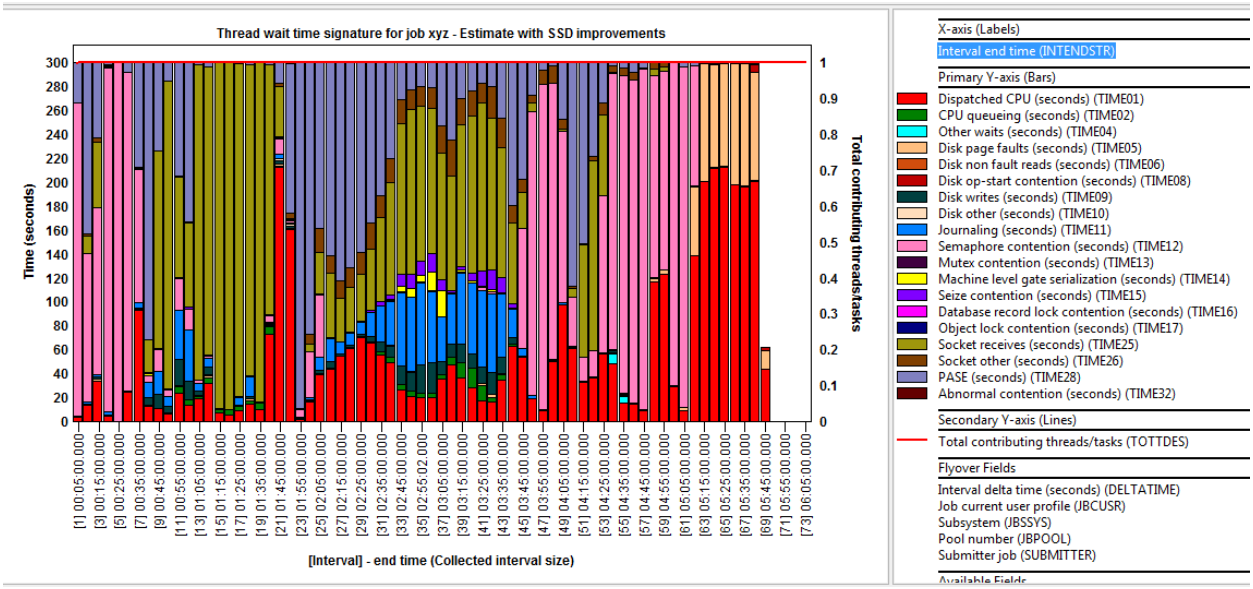
Initiating the SSDs Improvements Estimator option



SSD Improvements Estimator Window

Field	Description
Operations handled by SSDs	This value should be the estimated percentage of disk operations that will be handled by SSDs.
Average SSD I/O response time	Indicate the estimated SSD I/O average response time.
Improvements to other wait buckets	Indicate how much other wait buckets will improve based on the installation of SSDs. A modest 5% is the default value for this.
Shift time improvements left	This is typically only used when viewing a graph for a single job. When activated the time values will move to the left in the graph filing all available space for each collection interval.
Adjustment factor	This is the number of passes through the data iDoctor will do looking for gaps to fill. Typically several passes are needed to make the graph look good, using too small a number for this value will cause the height of some bars to not be consistent.
Max time each interval	Optional value for the maximum number of seconds each interval should contain.

After using the option the graph title will show the results of the estimated improvements. You can see several minutes of time were reduced in the job mostly in the disk page faults bucket times.



Graph example displaying estimated SSD improvements

9.25 Analyzing Collection Services Data

Analyzing Collection Services data effectively consists of the use of many graphs and reports available within the collections. Some information on how to analyze and use the iDoctor (Job Watcher) graphs is provided in demos on the website and additional ones will be added. The Job Watcher graphs are very similar to the CSI graphs.

Visit the following page to access the current iDoctor demos:
https://www-912.ibm.com/i_dir/idoctor.nsf/videos.html

Tip: Use the graph compare function by clicking the graph compare icon on the toolbar of the main window. This will allow you to view two graphs at once with synchronized scrolling. The graph compare function is either on (if pressed in the toolbar on Main Window) or off. Any graph opened while the compare mode is on will produce a split view two areas used to analyze graphs. Either an alternate view graph can be used as the comparison graph or the clock icon can be pressed to compare graphs of different interval sizes.

10Disk Watcher

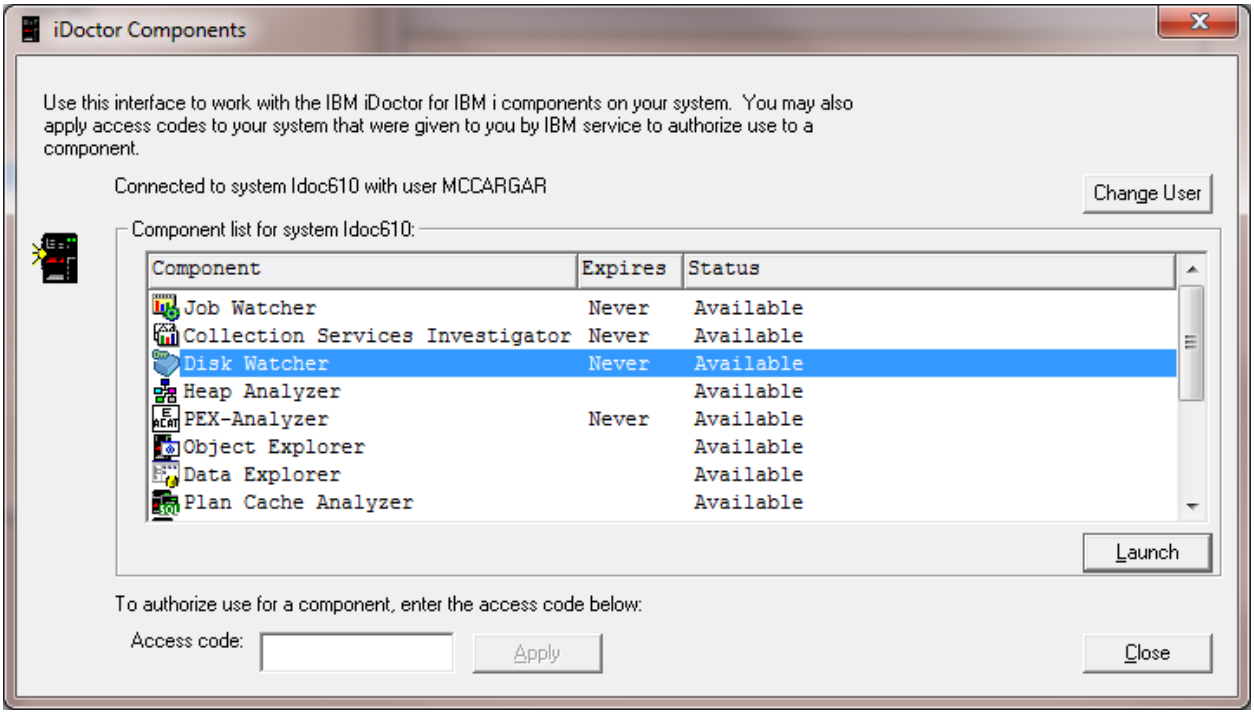
This chapter provides an overviw of the interfaces within the IBM iDoctor for IBM i - Disk Watcher component.

The Disk Watcher component provides a number of interfaces designed to help the user analyze disk performance problems on IBM i.

10.1 Starting Disk Watcher

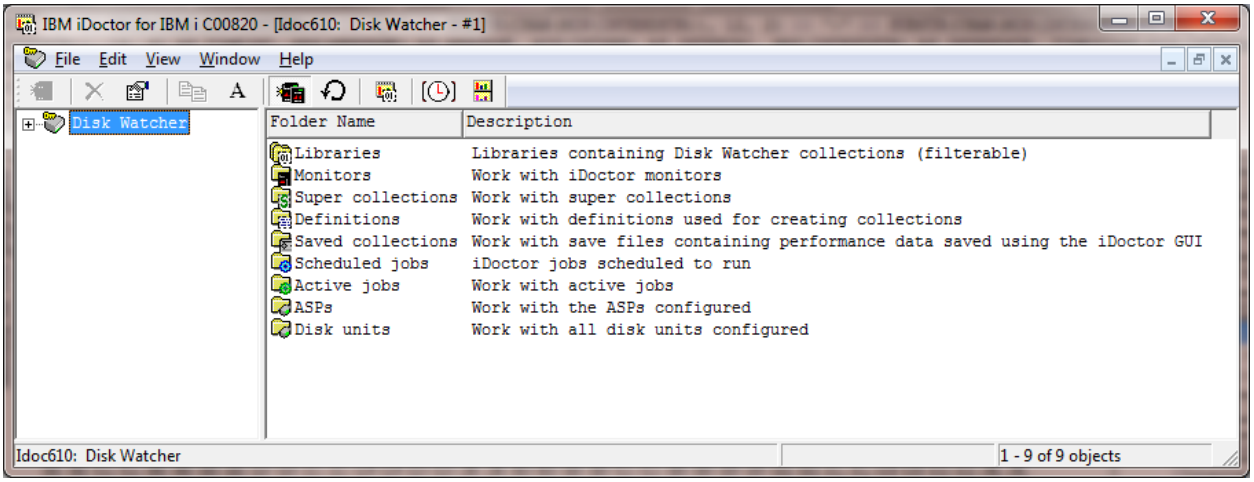
Disk Watcher is a component of the iDoctor suite of tools. iDoctor can be started using the Start menu: Start->Programs->IBM iDoctor for IBM i. Once the IBM iDoctor for IBM i application appears, the Disk Watcher component is started from the Connection List View by double-clicking on the desired system.

A list of available components will appear on the next window. Double-click on the Disk Watcher component or select Disk Watcher and click the Launch button in order to continue



iDoctor Components View

10.2 Disk Watcher Component View



Disk Watcher Component View

The 'Disk Watcher' folder contains a list of folders, each providing different features available. Collections can be displayed in various ways, either under the Libraries folder on a per library basis, or under the Monitors or Super Collections folders for Disk Watcher collections that exist within a monitor or Super Collection.

10.2.1 Menu Options

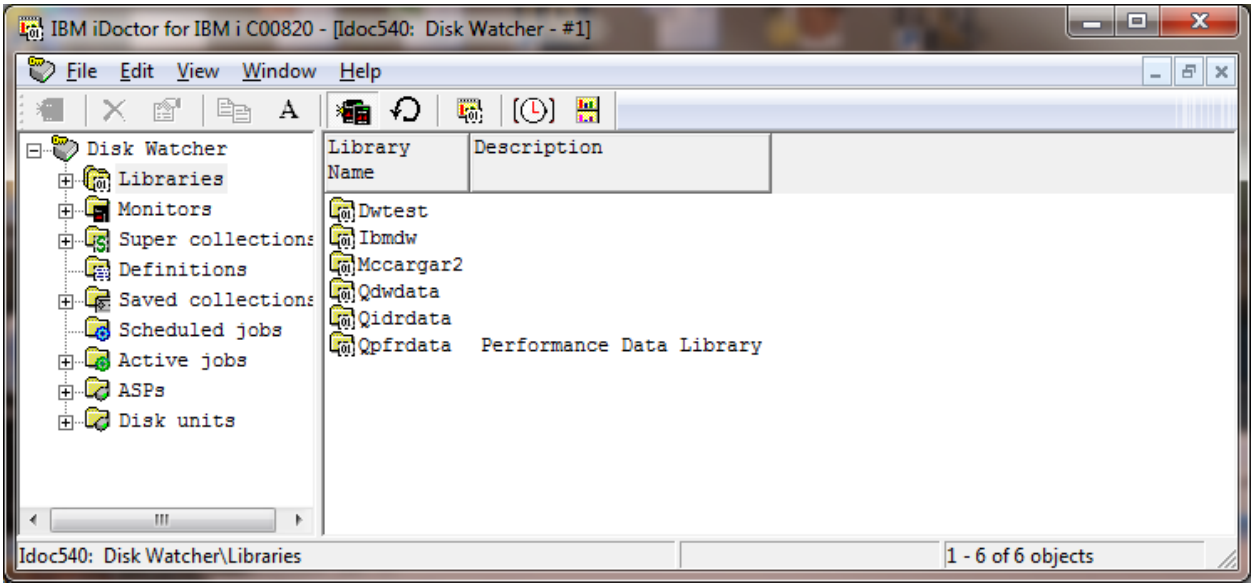
The following Disk Watcher specific menu options are available by right clicking on the 'Disk Watcher' icon in the component view above:

Menu Item	Description
Filter libraries…	This option allows you to filter the libraries shown in the Libraries and SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of the list if the system contains many libraries containing collections (and/or SQL tables).
Add Definition	This option displays the Add Disk Watcher Definition Wizard.. The definition defines characteristics about the collection such as which data options to collect. At 6.1 or higher a definition is required when starting a collection.
Start Collection	This menu will open the Start Disk Watcher Wizard where the user can define and run a collection. At 6.1 or higher a definition must be defined first before creating a collection unless you wish to use an IBM-supplied definition.
Start Monitor	This menu will open the Start Monitor Wizard for iDoctor where the user can start a Job Watcher, PEX Analyzer or Disk Watcher monitor. Monitors are designed to provide 24x7 collection of performance data.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table form.
Set User-Defined Reports Database	This option allows the user to view/modify the currently used user-defined reports database. The database can either be an MS Access file or a library on an IBM i. The database stores the information needed to build the user-defined tables and graphs shown in iDoctor.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Work with iDoctor scheduled jobs	This option is a shortcut to the General functions –> Work management -> Scheduled jobs folder. It shows all the iDoctor created scheduled jobs that exist on the current IBM i system.
Collections database	The iDoctor collection database identifies all collections on the system and can be used to facilitate the drill down from one component to another in some situations.

Descriptions for additional menu options that are common to all components can be found here.

10.3 Libraries

This folder contains the libraries on the system that contain Job Watcher data (specifically the libraries containing file QAPDJWRUNI). The list displays each library's name and description. By clicking on a library in the tree you will see its contents (the collection(s) that exist in the library)



Libraries in the Disk Watcher Component View

10.3.1 Menu Options

The following menu options are available by right clicking on a library in the component view.

Menu Item	Description
Start Collection...	This menu will open the collection wizard for Disk Watcher where the user can define and run a collection. At 6.1+, a definition must be defined first before creating a collection unless you wish to use an IBM-supplied definition.

Additional menu options that are common to all library folders in iDoctor are discussed here.

10.4 Monitors

Disk Watcher monitors allow for 24x7 collection of Disk Watcher data on a system. They run continuously storing only the most recent collections desired. Disk Watcher monitors will run until ended manually by the user. Monitors can be held and released if the user wishes to stop collecting data for now and then continue collection again later. Monitors can also be scheduled to start and end at the desired times.

Once a monitor has been started and ended, it must be restarted using the Restart Monitor option. You cannot use the Start New Monitor option to restart an existing monitor.

A Monitors folder is provided in Disk Watcher to allow the user to work with the monitors that exist on the current system. For more information about monitors, see the section on Monitors in chapter 4.

10.5 SQL Tables

This folder contains all the SQL tables that exist on the system generated by Disk Watcher analyses.

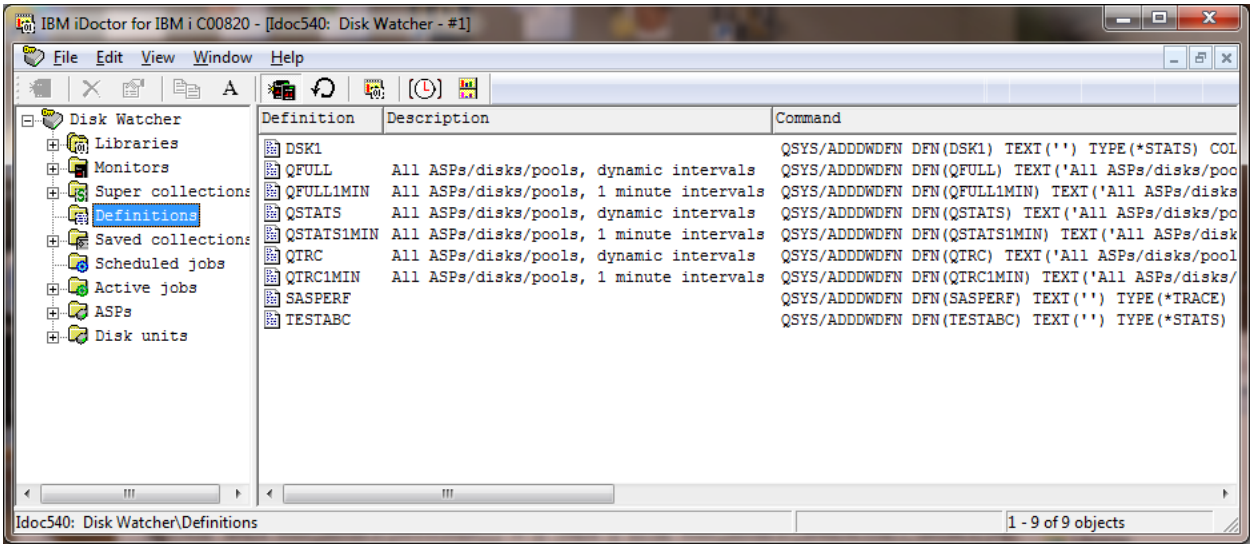
The folder exists in 3 places and each will filter the contents appropriately based on where it is located:

Location	Description
Under Disk Watcher	Entire system
Under a library	All collections in the library
Under a collection	Only this collection

For more information see the SQL Tables section in chapter 4.

10.6 Definitions

A Definitions folder is provided in Disk Watcher to allow the user to work with the Disk Watcher definitions that exist on the current system. An example of this interface is:



Disk Watcher Definitions Folder

The fields shown in this view are as follows:

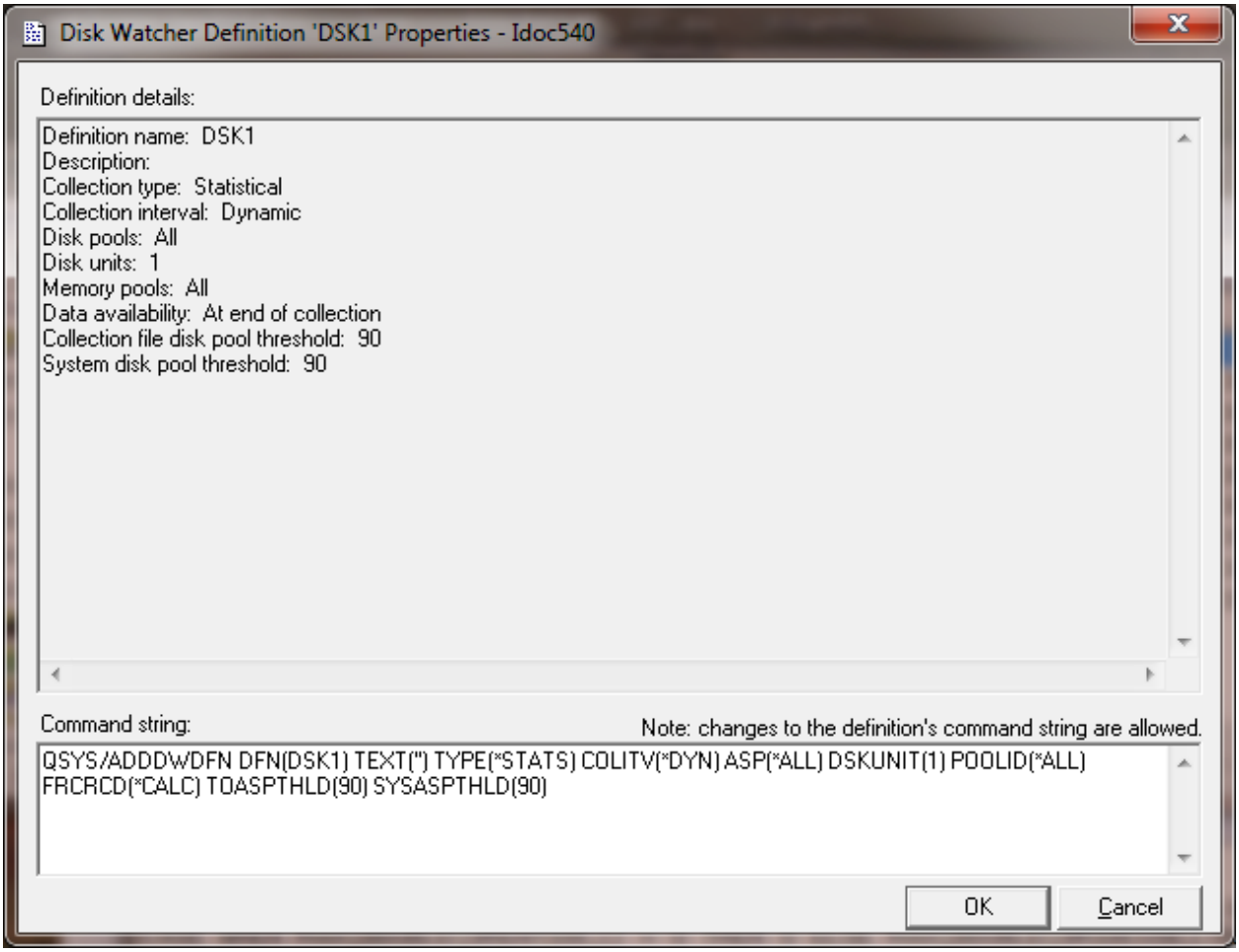
Field	Description
Definition	Name of the definition. IBM-supplied definitions begin with Q.
Description	An optional description given to the Disk Watcher definition
Command	The command string used to create the definition.

The following options are available when right clicking on one or more definitions in the list:

Field	Description
Change Definition	Opens the Add Disk Watcher Definition Wizard and loads the selected definition into it so it can be changed.
Add Definition	Opens the Add Disk Watcher Definition Wizard in order to create a new definition.
Start Collection	Opens the Start Disk Watcher Collection Wizard using the selected definition.
Start Monitor	Opens the Start iDoctor Monitor Wizard using the selected definition.
Delete	Removes the selected definitions from the system.
Properties	Displays the properties for the selected Disk Watcher definition.

10.6.1 Properties

Double-clicking on a definition or using the Properties menu from the Disk Watcher Definitions View displays all of the parameters that were used when creating the definition. An example of this interface is:



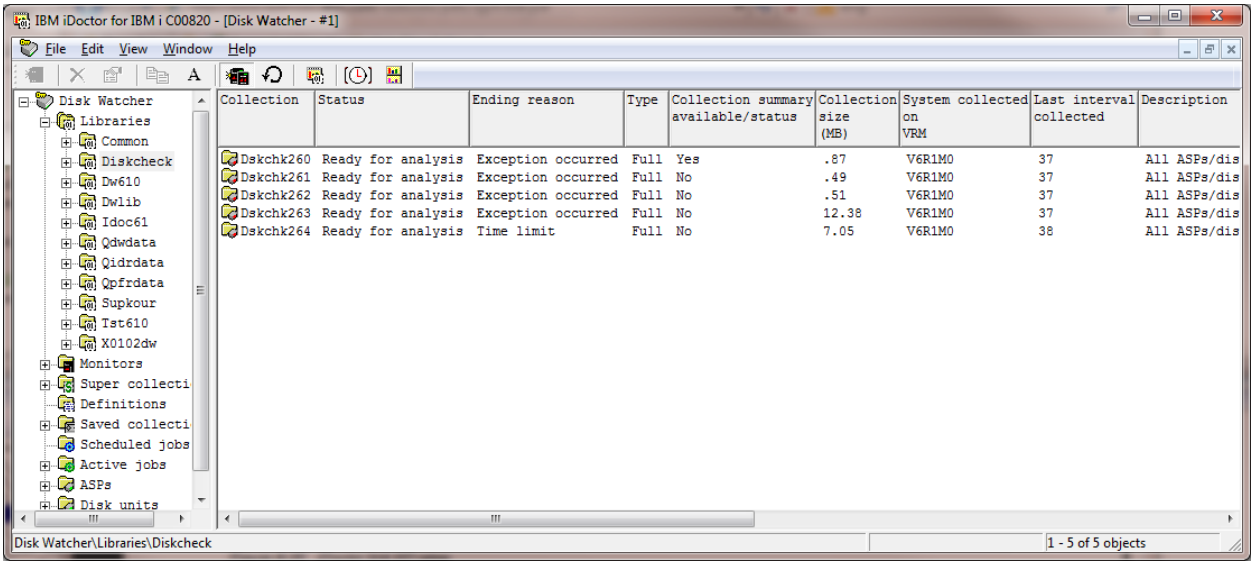
Disk Watcher Definition Properties

Advanced users can change the command string that defines the definition if desired. If changes have been made to the command string, pressing the OK button will remove the existing definition from the system and replace it using the command string specified.

Note: IBM-supplied definitions cannot be changed.

10.7 Collections

Moving down the tree within each Library folder are one or more collections that have been created or are currently being created within the current library.



Disk Watcher Collections in a Library

Tip: Collections that contain Trace data can be summarized at V5R3 and higher to produce several Trace only graphs and reports. If a collection has been summarized the Collection summary available/status field will display 'Yes'.

10.7.1 Collection Fields

The list of collections displays the collection name, description, status as well as several additional fields.

Each collection in the list has a set of fields available which can be optionally reordered and displayed. To change the current field selections for the collection list, use the Select fields... menu from the library folder. A listing of the available fields and a short description is provided in the table below:

Field	Description
Collection	Name of the collection. This name matches the member name used in the database files named QAPYDW* that exist in the current library.
Status	The status field indicates the status of the job on the system running the collection (if active) or if not active the status indicates whether or not the collection is ready for use.
Ending reason	This field indicates what caused the collection to end. There are several possible reasons a collection may end as described below: Size limit – The collection exceeded the maximum disk space allowed as described in the definition. Interval limit – The collection stopped when the maximum intervals to collect was met. Time limit – The collection stopped when the maximum time limit to collect was met. ASP limit – The system ASP limit as defined in SLIC service tools has been exceeded causing the collection to end. Ended by user – Disk Watcher detected that the user ended the collection manually.
Type	The type of collection created. The valid types are: Full – includes both Trace and Statistical data Statistical – does not include information about all I/Os, only higher-level summary data. Trace – includes information about each I/O that occurred and optionally the causing program, object or memory pool.
Collection summary available/status	Indicates if the collection has been summarized or not. If this is No, then the trace graphs folder is not available under the collection. Use the summarize menu for a Trace or Full type collection to create the required summary files.
Collection size (MB)	Displays the approximate size of the collection in megabytes. This size does not include the size of the summary files.
System collected on VRM	The version of IBM i that was used to create this collection. It is possible to view and analyze collections from a previous (or even later) release using the GUI. Versions V5R3 through V6R1 are currently supported.
Last interval collected	This value shows the last interval collected. If the collection is not running, this value indicates the total number of intervals that were collected.
Description	A description for the collection specified at creation time.
Definition	The name of the definition used to create the collection
Start time	The date/time the collection started.
End time	The date/time the collection ended.
Job creating collection	The fully qualified job that created (or is currently creating) the collection.

10.7.2 Menu Options

The table below outlines the different types of operations that may be performed by right clicking on a collection within the Disk Watcher component view.

Menu Item	Description
Explore	Displays the contents of the collection folder in the right pane of the Collection Services Investigator component view.
Statistical graphs	Provides several graphs over the statistical data generated by Disk Watcher
Trace graphs	Provides several graphs over the trace data generated by Disk Watcher. This option is only available if trace data has been collected and the collection has been summarized.
Collection files	Displays a table view for the desired file included in the collection.
Generate Reports…	This option can be used to build a report of the desired set of Disk Watcher tables and graphs. The report consists of a screenshot of each graph along with its title and collection information. The reports are built into a HTML page and displayed in the web browser when completed.
Summarize	Runs the summary for the collection. Running the summary is recommended before analyzing data. This action will make available the Trace graphs if trace data has been captured.
Copy...	Copies one or more collections to another library. Selecting multiples is only available from the list side of the component view.
Delete...	Deletes a collection. Select multiple collections in order to delete more than one at a time. Selecting multiples is only available from the list side of the component view.
Save	Saves the selected collections to a save file on the system. The save file will be added to the list under the Saved collections folder.
Transfer to...	FTP one or more collections to another system. Selecting multiples is only available from the list side of the component view.
Stop	Stops an active collection.
Properties	Use this menu to display the property pages for the collection. The property pages provide quick access to additional summary information about the collection.

10.7.3 Generate Reports

See the ***Generate*** Reports section in chapter 4.

10.7.4 Copy

See the Copy collection section in chapter 4.

10.7.5 Delete

See the Delete collection section in chapter 4.

10.7.6 Save

See the Save collection section in Chapter 4.

10.7.7 Transfer to…

See the Transfer to section in chapter 4.

10.7.8 Stop

An active collection can be stopped by using the Stop menu found by right clicking on a collection within the Disk Watcher component view.

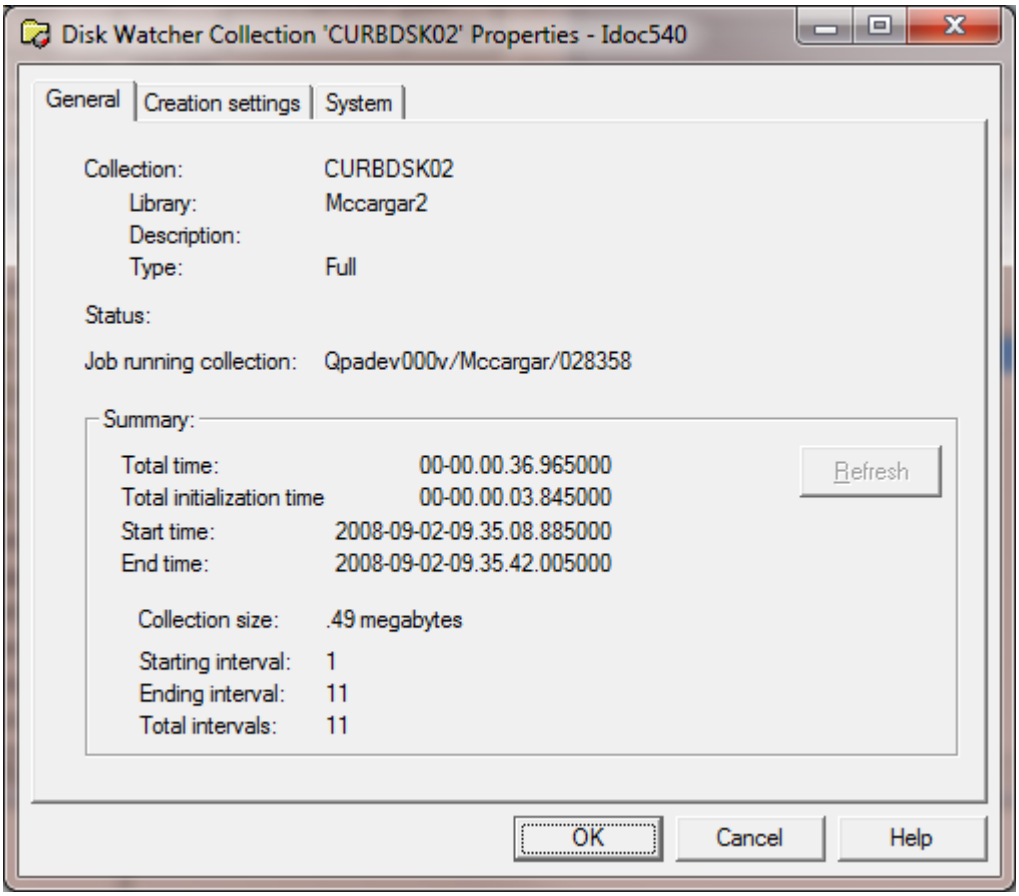
At 5.4 and 5.3, this option will issue an ENDJOB command for the job running the collection.
At 6.1 and higher this will issue an ENDDW command.

10.7.9 Properties

This section covers the property pages for a collection. Access the property pages by right clicking on a collection and choosing the Properties menu.

10.7.9.1 General

The General property page provides basic information about the collection such as when it was created.



Collection Properties - General

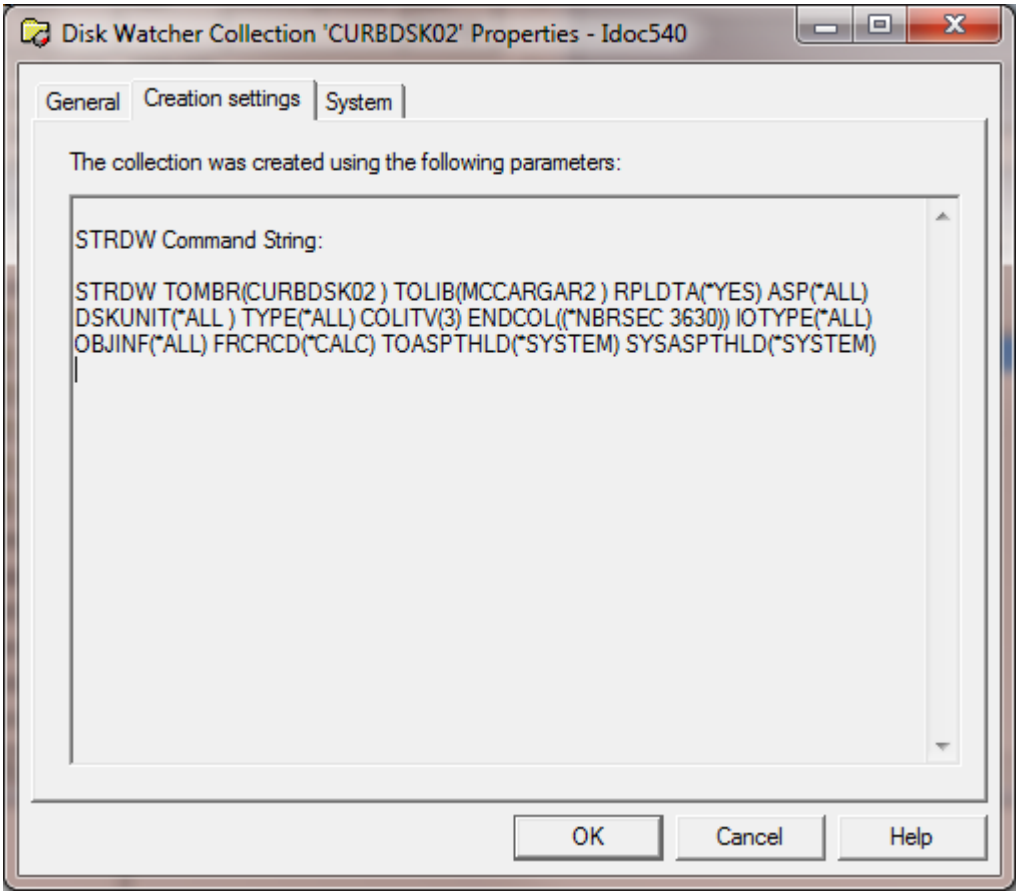
The following information is displayed on the General property page:

GUI Element	Field Description
Collection	Name of the collection. This matches the member name used in the QAPYDW* files on the server in the library specified.
Library	Library the collection resides in.
Description	Description of the collection.
Type	The type of collection indicating the type(s) of data captured: Statistical, Trace or Full (which is both Statistical and Trace)
Status	The status of the collection. This could indicate if the job running the collection failed or that the collection is ready for analysis.
Job running collection	Displays the name of the job that created or is currently creating the collection. If the job log is available a button will be shown to display it.

GUI Element	Field Description
Total time	Displays the total run time of the collection in timestamp format.
Total initialization time	Displays the estimated initialization time for the collection in timestamp format. This is an estimate of the amount of time it took between the collection being started and the 1st interval of data being collected
Start time	The time the collection was started.
End time	The time the collection ended (if it has ended).
Collection size	The total size of the collection. This number does not include any summary files generated.
Starting interval	The 1st interval number detected in the collection.
Ending interval	The last interval number detected in the collection.
Total interval	The total number of intervals found in the collection.

10.7.9.2 Creation Settings

The Creation settings property page provides details about the parameters that were used when creating the collection.

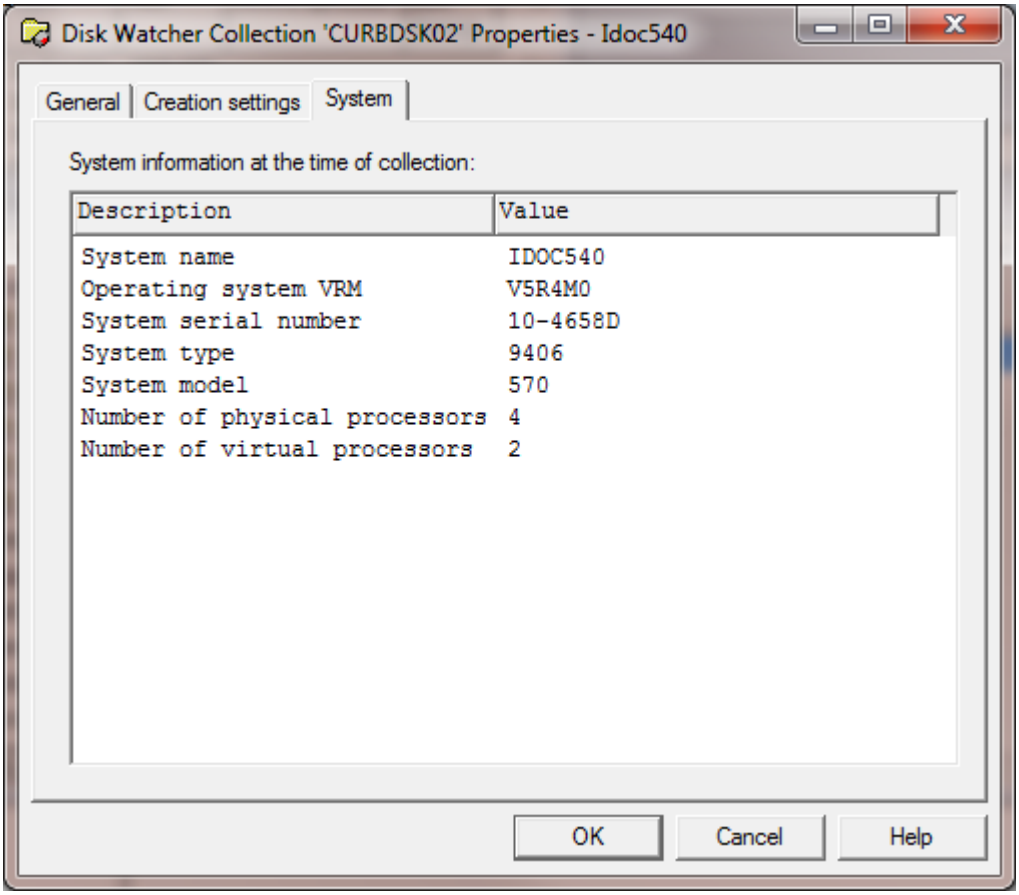


Collection Properties – Creation Settings

The information shown on this window matches the summary page of the Start Disk Watcher Wizard when the collection was created.

10.7.9.3 System

The system property page displays details about the system the collection was created on. This information includes the type, model, operating system VRM and the number of processors.



Collection Properties – System

10.8 Analyzing Disk Watcher Data

Analyzing Disk Watcher data effectively consists of the use of many graphs and reports available within the collections. Some information on how to analyze and use iDoctor graphs is provided in demos on the website and additional ones will be added in the future.

Visit the following page to access the current iDoctor demos:

http://www-912.ibm.com/I_dir/idoctor.nsf/downloadsDemos.html

Tip: Use the graph compare function by clicking the graph compare icon on the toolbar of the main window. This will allow you to view two graphs at once with synchronized scrolling. The graph compare function is either on (if pressed in the toolbar on Main Window) or off. Any graph opened while the compare mode is on will produce a split view two areas used to analyze graphs. Either an alternate view graph can be used as the comparison graph or the clock icon can be pressed to compare graphs of different interval sizes.

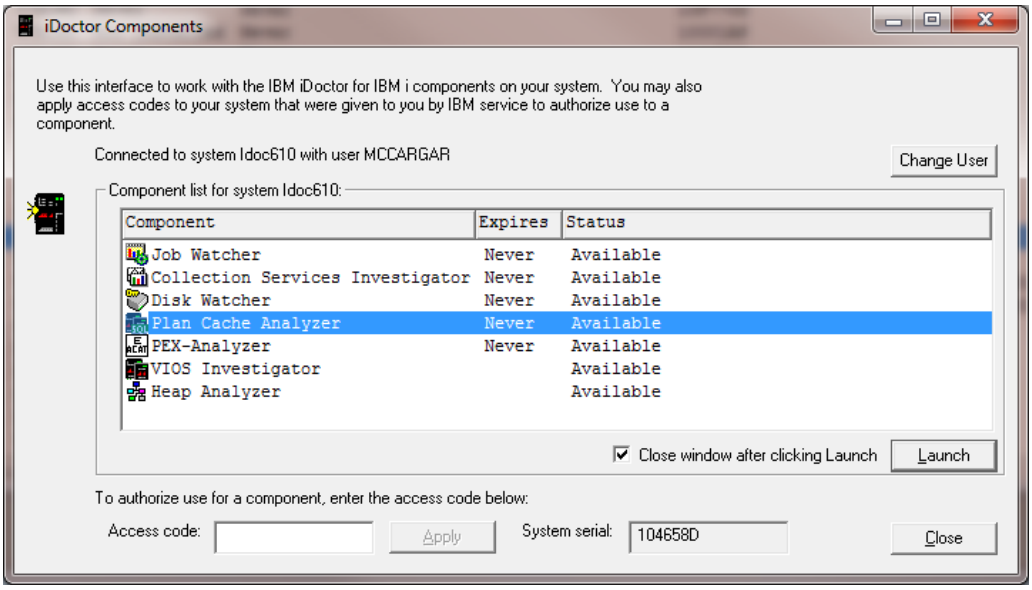
11Plan Cache Analyzer

This chapter provides an overview of the interfaces within the Plan Cache Analyzer component.

11.1 Starting Plan Cache Analyzer

Plan Cache Analyzer is a component of the iDoctor suite of tools. iDoctor can be started using the Start menu: Start->Programs->IBM iDoctor for IBM i. Once the IBM iDoctor for IBM i application appears, the Plan Cache Analyzer component is started from the Connection List View by double-clicking on the desired system.

A list of available components will appear on the next window. Double-click on the Plan Cache Analyzer component or select Plan Cache Analyzer and click the Launch button in order to continue

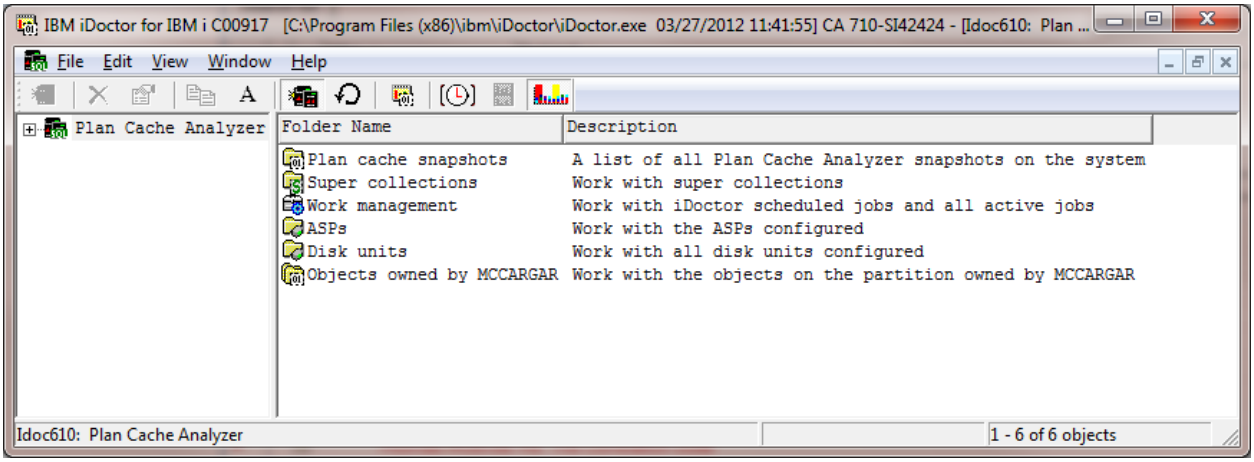


iDoctor Components Window

Note: Plan Cache Analyzer is a subcomponent of Job Watcher and will only be available if Job Watcher is installed correctly and a valid access code for Job Watcher has been applied. This component is included with the Job Watcher license.

11.2 Plan Cache Analyzer Component View

The Plan Cache Analyzer view is the interface used to create SQL plan cache snapshots or work with and analyze existing data.

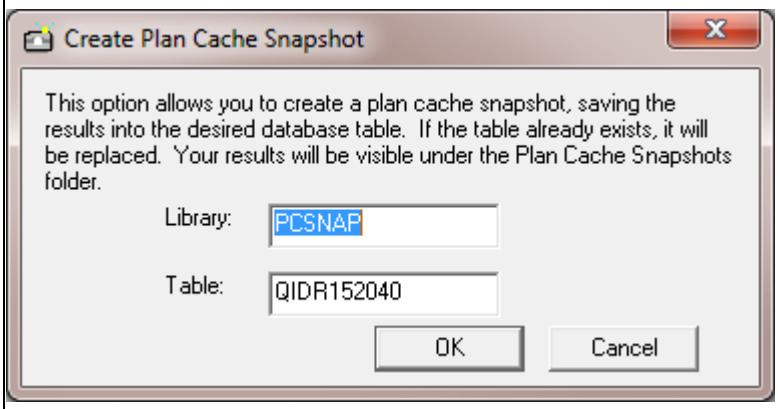
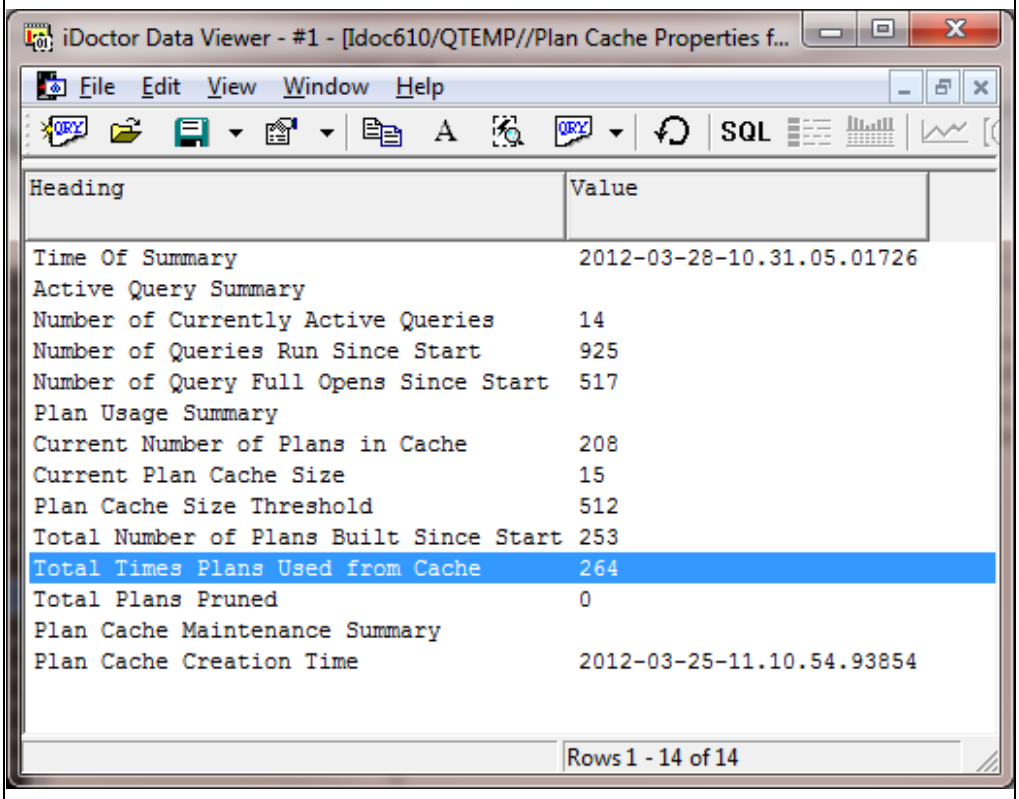


Plan Cache Analyzer Component View

The 'Plan Cache Analyzer' folder contains a list of folders, each providing different features available. The snapshots are displayed under the Plan cache snapshots folder.

11.2.1 Menu Options

The following Plan Cache Analyzer specific menu options are available by right clicking on the 'Plan Cache Analyzer' icon in the component view above:

Menu Item	Description
Filter libraries…	This option allows you to filter the libraries shown in the Plan cache snapshots, Plan cache dumps, SQL performance monitors and SQL Tables folders by a generic library name or library owner. This is useful for speeding up the display of the list if the system contains many libraries containing collections (and/or SQL tables).
Create Plan Cache Snapshot	This option displays a window where the user can specify the library and table to create the new plan cache snapshot into.  After it has been created the Plan cache snapshots folder can be refreshed to display the new snapshot.
View Plan Cache Properties	This option runs a stored procedure (`qsys2/dump_plan_cache_properties`) that displays information about the system's plan cache.  **Note:** If you wish to run this report multiple times in the same window, open up the SQL editor and read the comments about the option to clear the previous data.
Open New Data Viewer	Opens a new Data Viewer window. This window is used to display tables and graphs on the system. You can open iDoctor-defined reports into this window or you can also open any database file or SQL table and display the results in graph or table

	form.
Set User-Defined Reports Database	This option allows the user to view/modify the currently used user-defined reports database. The database can either be an MS Access file or a library on an IBM i. The database stores the information needed to build the user-defined tables and graphs shown in iDoctor.
Clear iDoctor cache	Mainly intended for IBM use, this option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Work with iDoctor scheduled jobs	This option is a shortcut to the General functions –> Work management -> Scheduled jobs folder. It shows all the iDoctor created scheduled jobs that exist on the current IBM i system.
Collections database	The iDoctor collection database identifies all collections on the system and can be used to facilitate the drill down from one component to another in some situations.
Properties	Use this menu to display version information for the current component installed on the current system. The build level of the GUI is also displayed here.

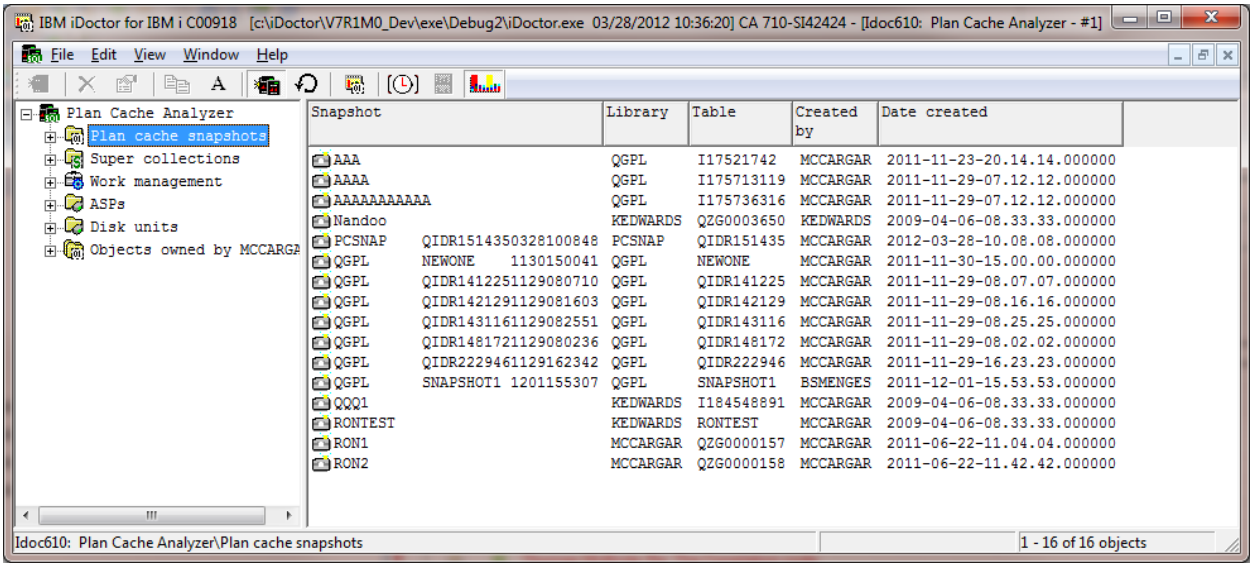
Descriptions for additional menu options that are common to all components can be found here.

11.3 Plan Cache Snapshots

This folder contains the list of Plan Cache Snapshots found on the system. This list can also be found in the IBM i Navigator GUI. Both lists are built from the same repository.

Each snapshot contains a detailed set of reporting options (graphs or tables).

Note: You can also use the IBM i Navigator to create snapshots (their GUI provides additional options for creating snapshots not found in the iDoctor GUI).



Plan cache snapshots folder in the Plan Cache Analyzer Component View

11.3.1 Snapshot Fields

The list of snaphots contains several columns which are described in this section.
A listing of the available fields and a short description is provided in the table below:

Field	Description
Snapshot	Name of the snapshot. In most cases this will contain the full library, table and member name information for the snapshot.
Library	The name of the library the snapshot was created in.
Table	The table name that contains the snapshot data.
Created by	The IBM i user profile that created the snapshot.
Date created	Date and time when the snapshot was created.

11.3.2 Menu Options

The following Plan Cache Analyzer specific menu options are available by right clicking on a snapshot in the component view.

Menu Item	Description
Statement graphs	Provides a set of graphing options that rank the data in various ways by statement (technically the Plan Hash ID)
Plan graphs	Provides a set of graphing options that rank the data in various ways by plan ID.
Generate Reports	Launches the Report Generator function that lets you create multiple reports at once.
Delete	Removes the selected plan cache snapshots from the system.

Additional menu options that are common to all library folders in iDoctor are discussed here.

11.4 Super Collections

For more information, see the Super Collections section in chapter 4.

11.5 Work management

See the Work management section in chapter 4.

11.6 ASPs

See the ASPs section in chapter 4.

11.7 Disk units

See the Disk units section in chapter 4.

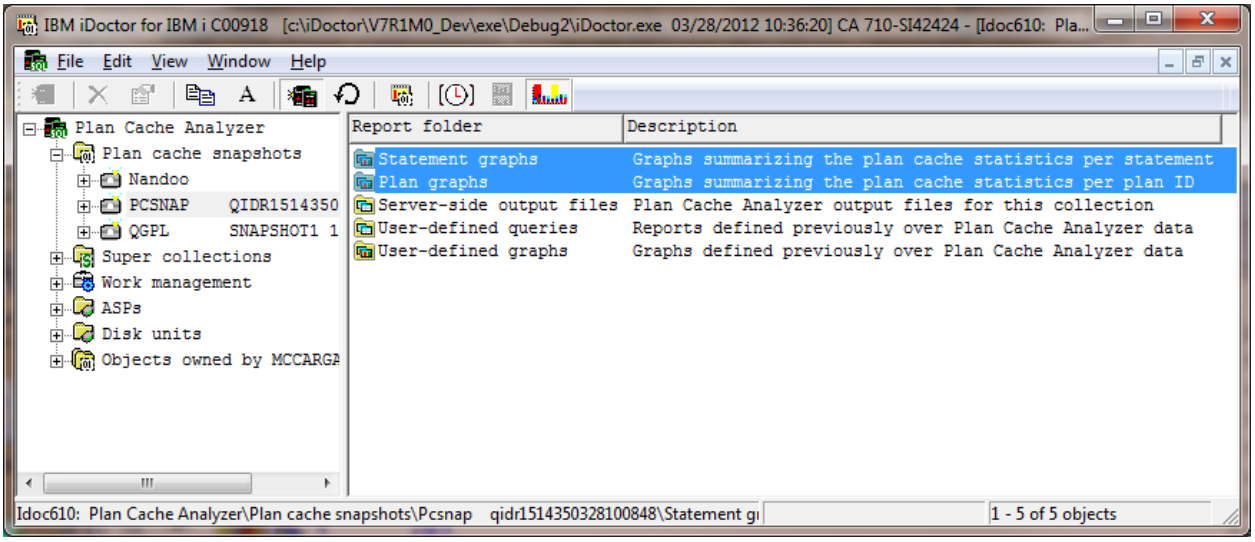
11.8 Analyses

Currently Plan Cache Analyzer does not provide any iDoctor analyses that produce additional SQL tables. Instead, graphs are immediately available from the snapshot data.

11.9 Snapshot Graphs

This section discusses the graphs directly underneath a snapshot and how to use them.

These graphs are contained within 2 folders under the collection. You can also access this same set of graphs by right-clicking the collection and picking the appropriate menu.



Graphing options in Plan Cache Analyzer

Each folder contains a series of graphs. You can open one by expanding the folder and double-clicking on the desired graph name. You can also open graphs by right-clicking them and choosing the desired menu option to either open the graph in a new Data Viewer or into an existing one.

Often most of these graphs in a folder will have several alternate views available. This allows you to quickly toggle between one graph and a different one. You can also use the Graph Compare icon on the toolbar of the Main Window in order to perform comparisons between graphs.

Tip: Use the graph compare function by clicking the Graph Compare icon on the toolbar of the main window. This will allow you to view two graphs at once with synchronized scrolling. The graph compare function is either on (if pressed in the toolbar on Main Window) or off. Any graph opened while the compare mode is on will produce a split view two areas used to analyze graphs. Either an alternate view graph can be used as the comparison graph or the clock icon can be pressed to compare graphs of different interval sizes.

Note: Unlike the other components, Plan Cache Analyzer does not contain time intervals.
Note: Graphing multiple snapshots at once is not currently supported

11.9.1 Graph Menu options

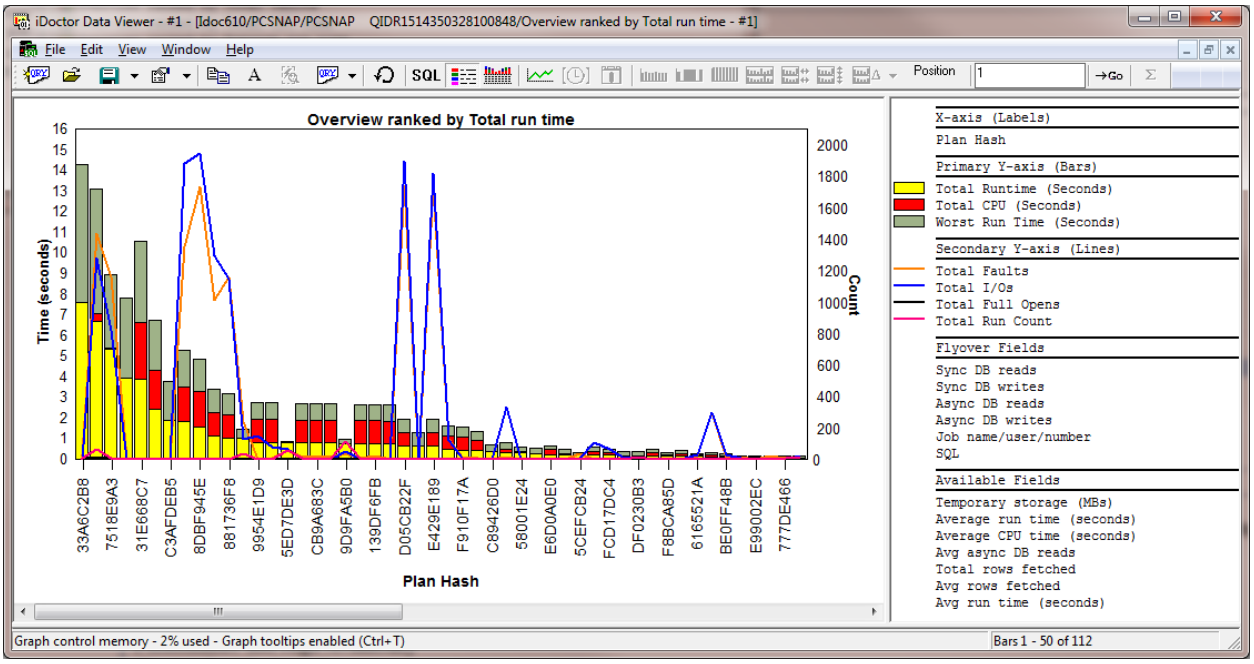
Right-clicking a graph gives a menu with the following options:

Menu	Field Description
Open graph(s)	Opens the selected graphs into a new Data Viewer or an existing one depending on the submenu available that shows the list of Data Viewers (if any are open).
Edit	This option will open the graph without running the SQL statement. The SQL Editor will be opened allowing the user to modify the query before running the SQL.

11.9.2 Statement Graphs

These graphs rank the data in the plan cache snapshot by statement (technically the Plan Hash ID) in various ways to get an idea of the relative performance contributions in terms of CPU, run time, I/Os and more.

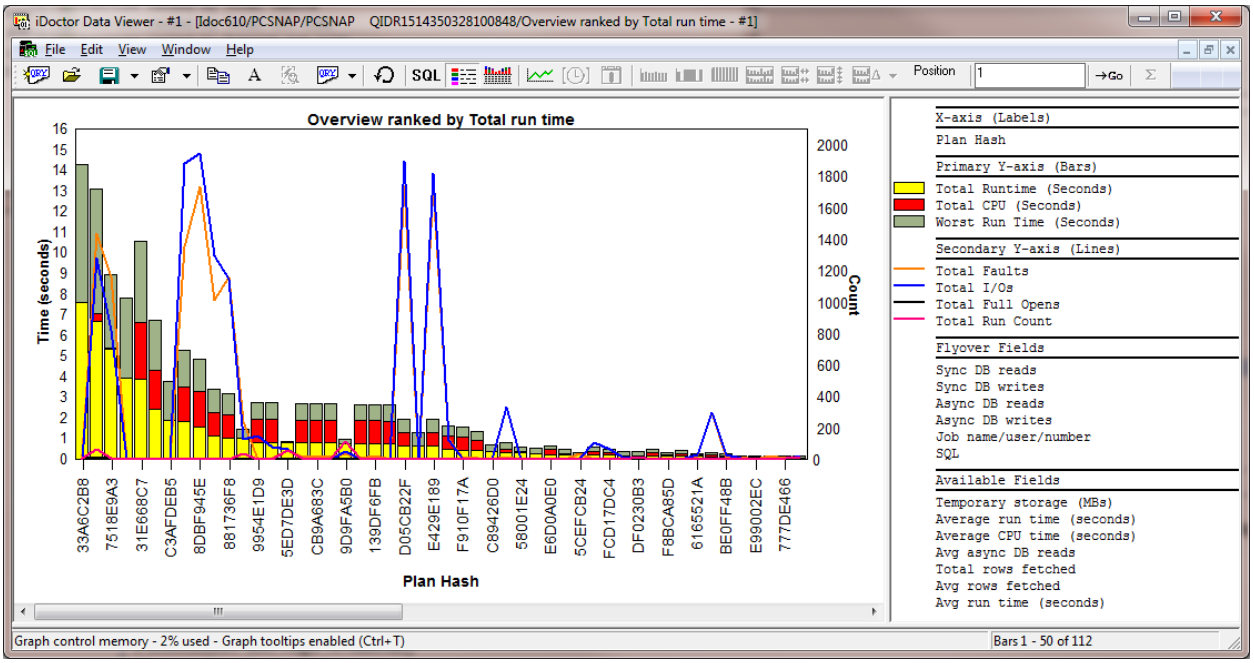
11.9.2.1 Overview ranked by Total run time



Overview ranked by Total run time

The statements are ranked by Total run time (yellow bar) in this graph. Faults, I/Os, opens, run count, CPU and run times are shown.

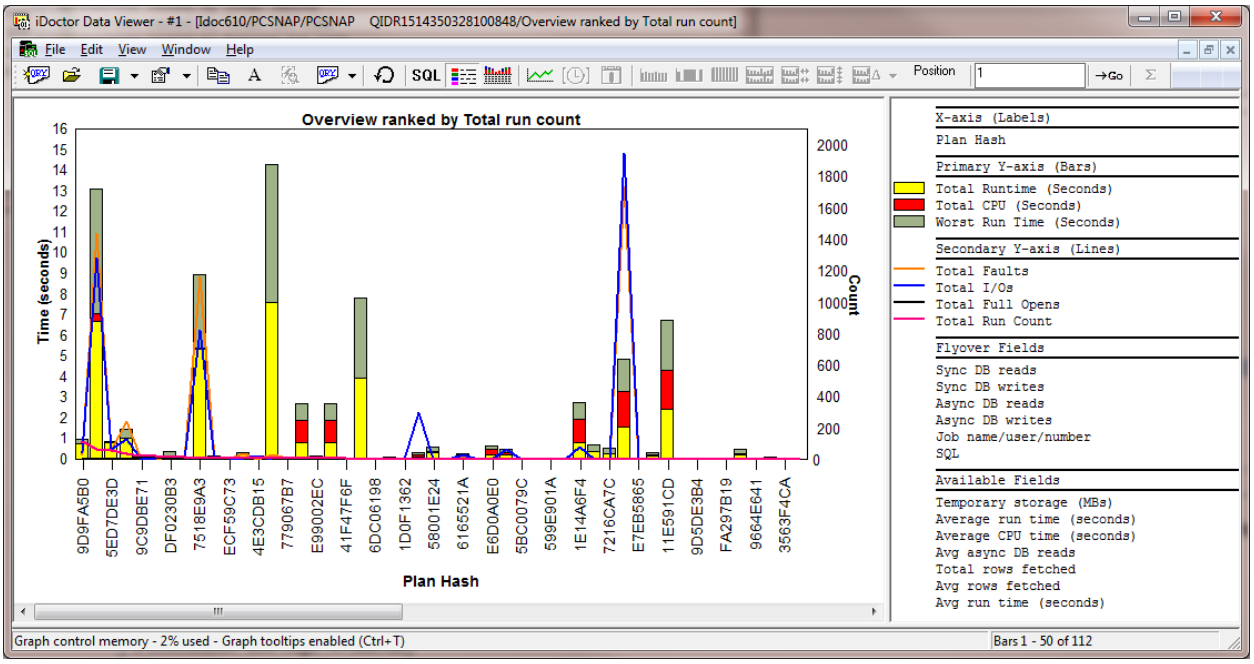
11.9.2.2 Overview ranked by Worst run time



Overview ranked by Worst run time

The statements are ranked by the worst run time (grey bar) in this graph. Faults, I/Os, opens, run count, CPU and run times are shown.

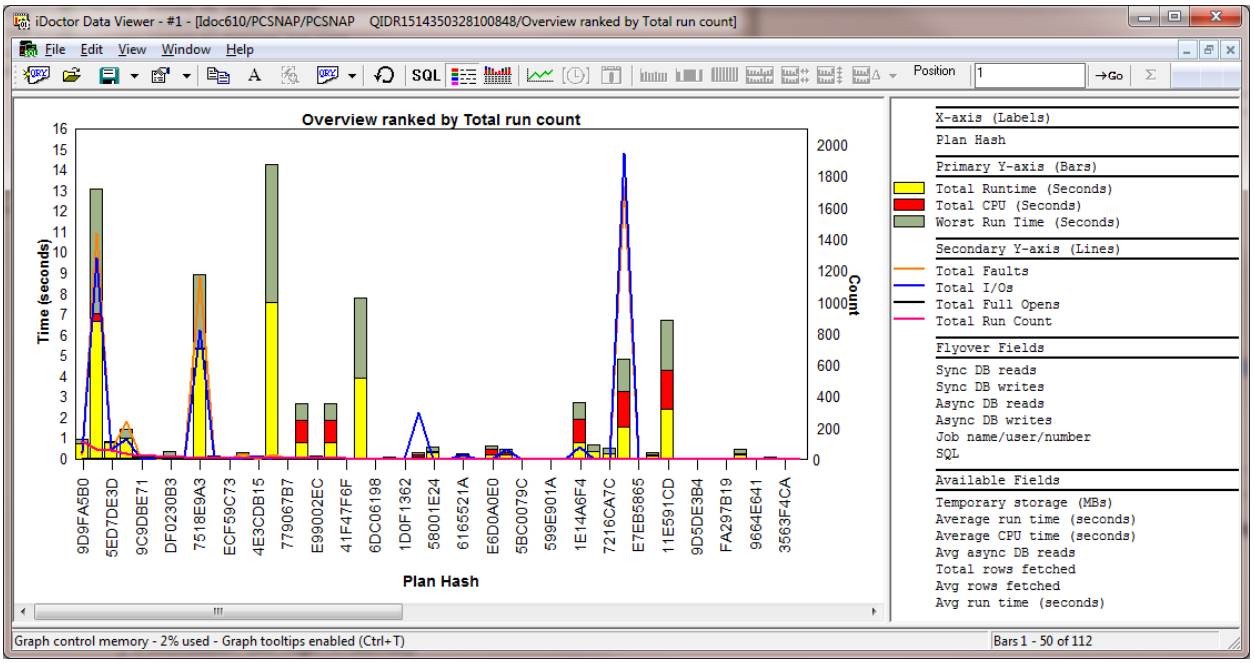
11.9.2.3 Overview ranked by Total run count



Overview ranked by Total run count

The statements are ranked by the one that ran the most times. Faults, I/Os, opens, run count, CPU and run times are shown.

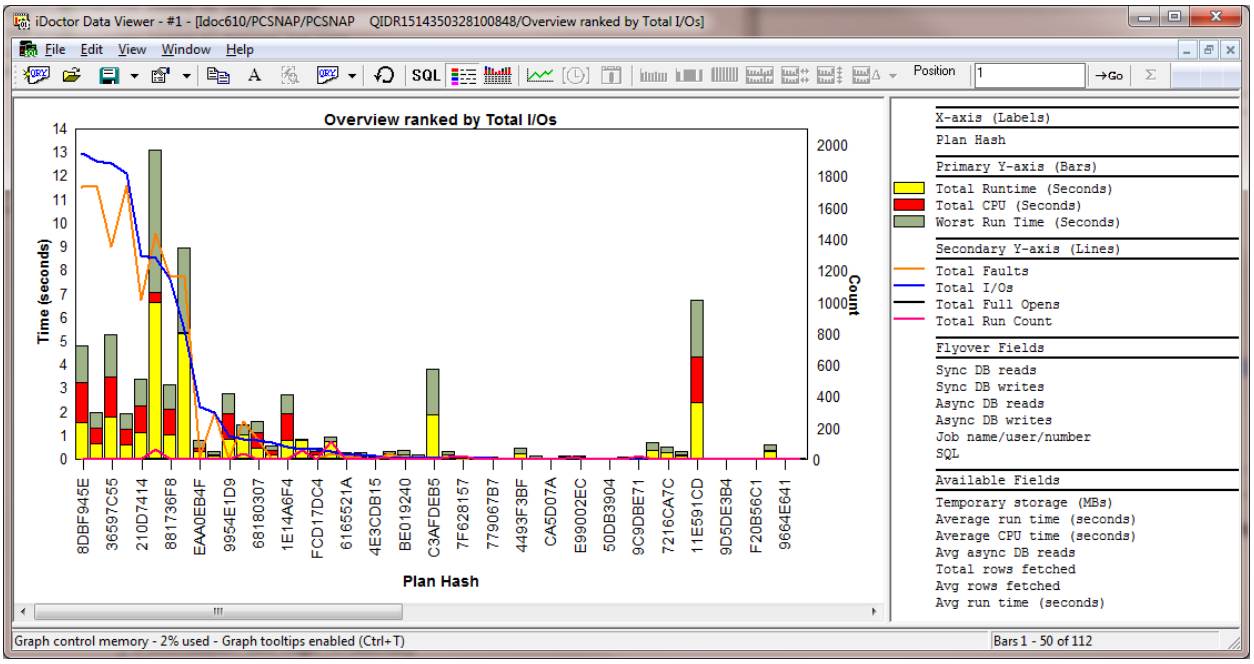
11.9.2.4 Overview ranked by Total full opens



Overview ranked by Total full opens

The statements are ranked by total full opens (black line). Faults, I/Os, opens, run count, CPU and run times are shown.

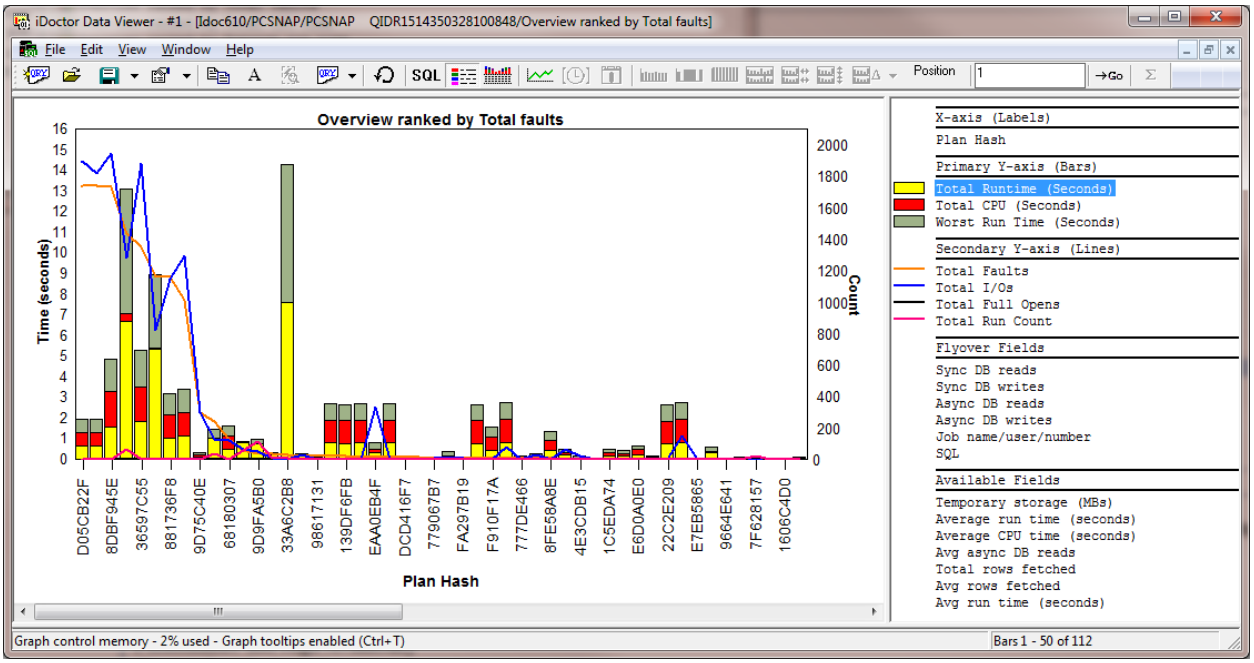
11.9.2.5 Overview ranked by Total I/Os



Overview ranked by Total I/Os

The statements are ranked by total I/Os (blue line). Faults, I/Os, opens, run count, CPU and run times are shown.

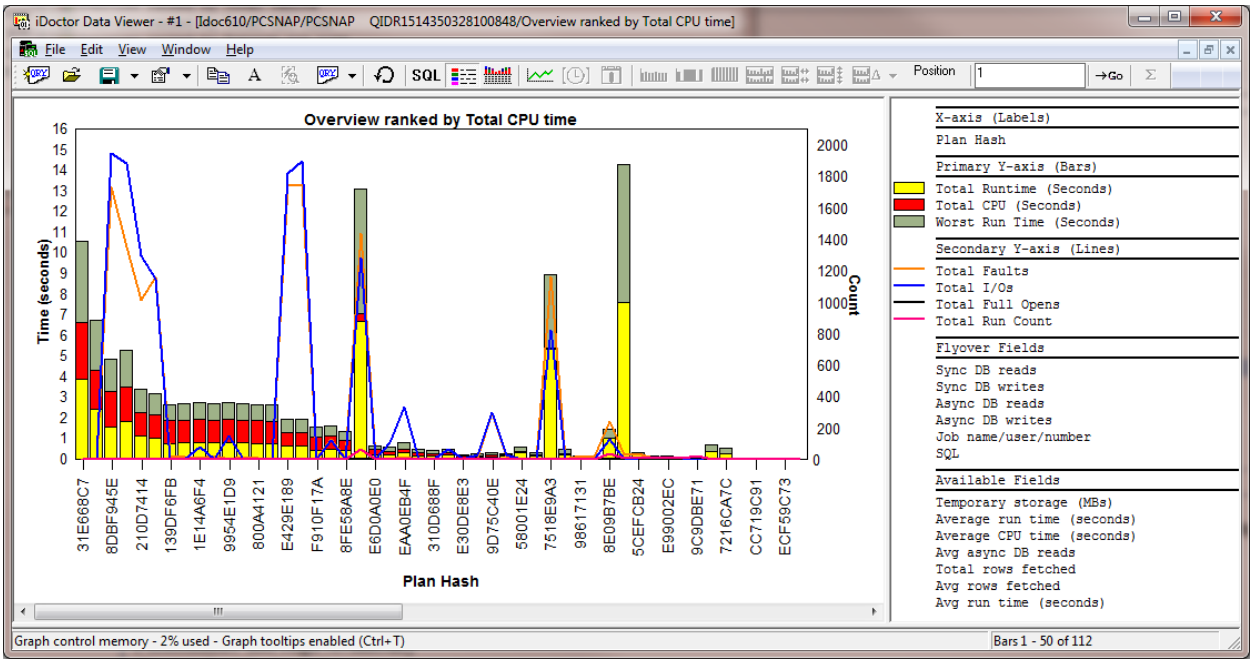
11.9.2.6 Overview ranked by Total faults



Overview ranked by Total faults

The statements are ranked by total faults (orange line). Faults, I/Os, opens, run count, CPU and run times are shown.

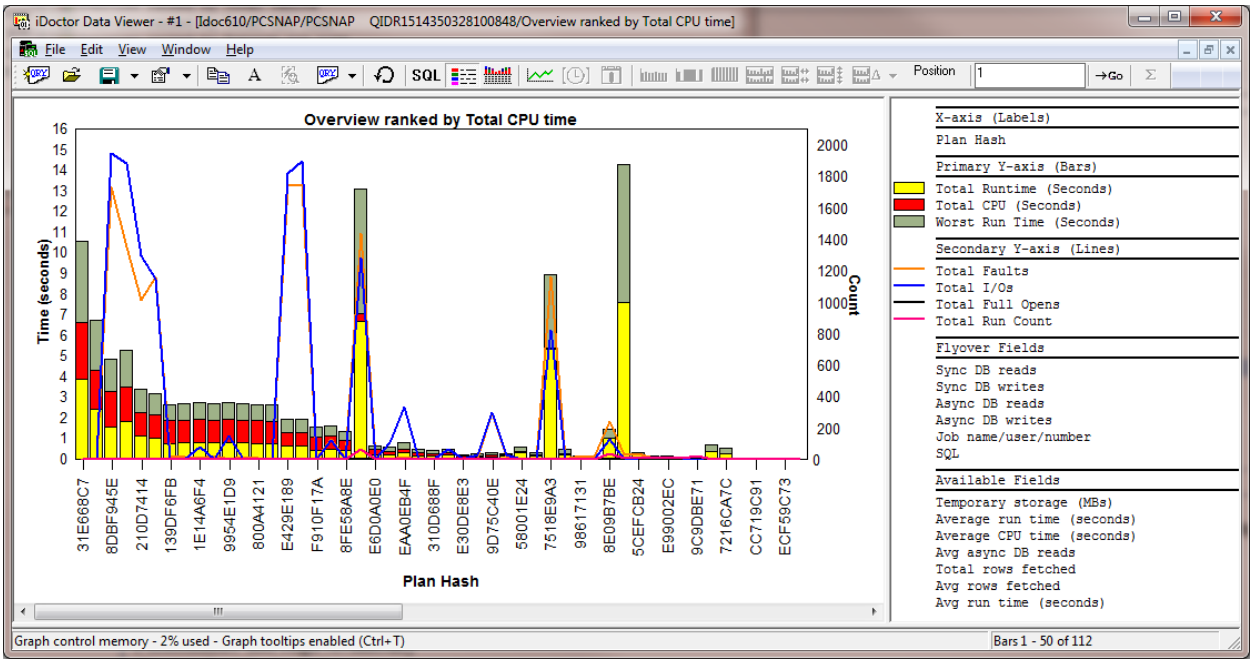
11.9.2.7 Overview ranked by Total CPU time



Overview ranked by Total CPU time

The statements are ranked by total CPU time (red bar). Faults, I/Os, opens, run count, CPU and run times are shown.

11.9.2.8 Overview ranked by Average run time

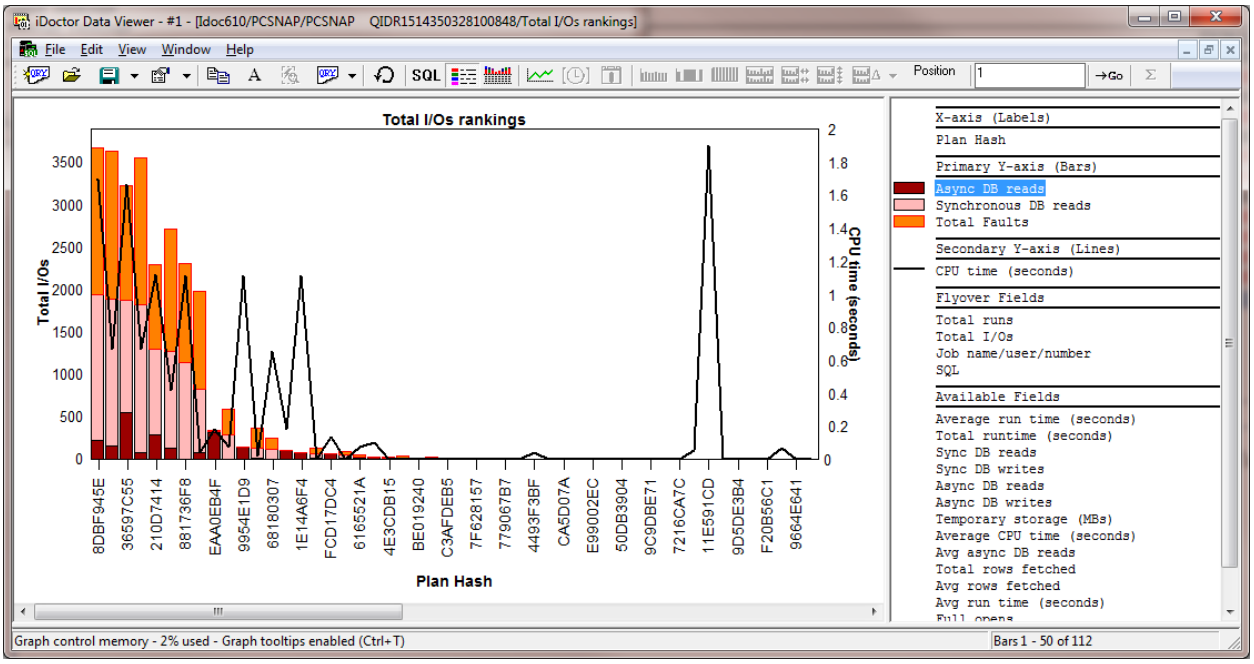


Overview ranked by Average run time

The statements are ranked by average run time (not shown but in available fields within the graph legend if desired)

Faults, I/Os, opens, run count, CPU and run times are shown.

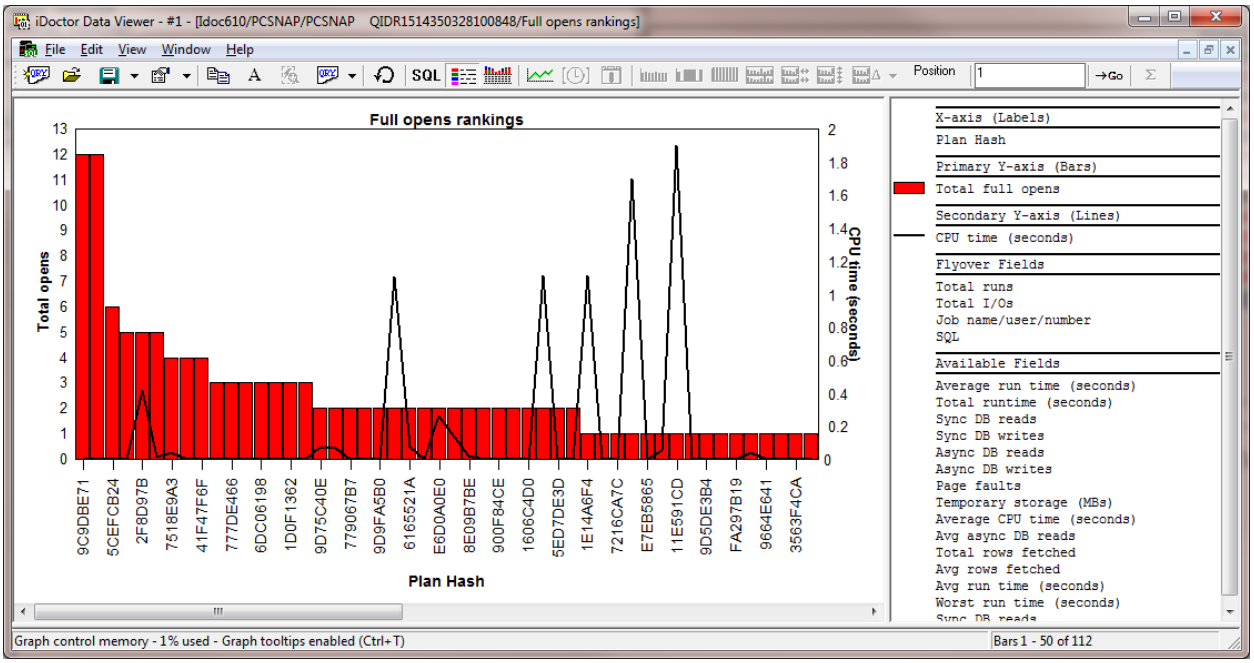
11.9.2.9 Total I/Os rankings



The statements are ranked by the total reads and faults added together.

Async DB reads, Sync DB reads total faults and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

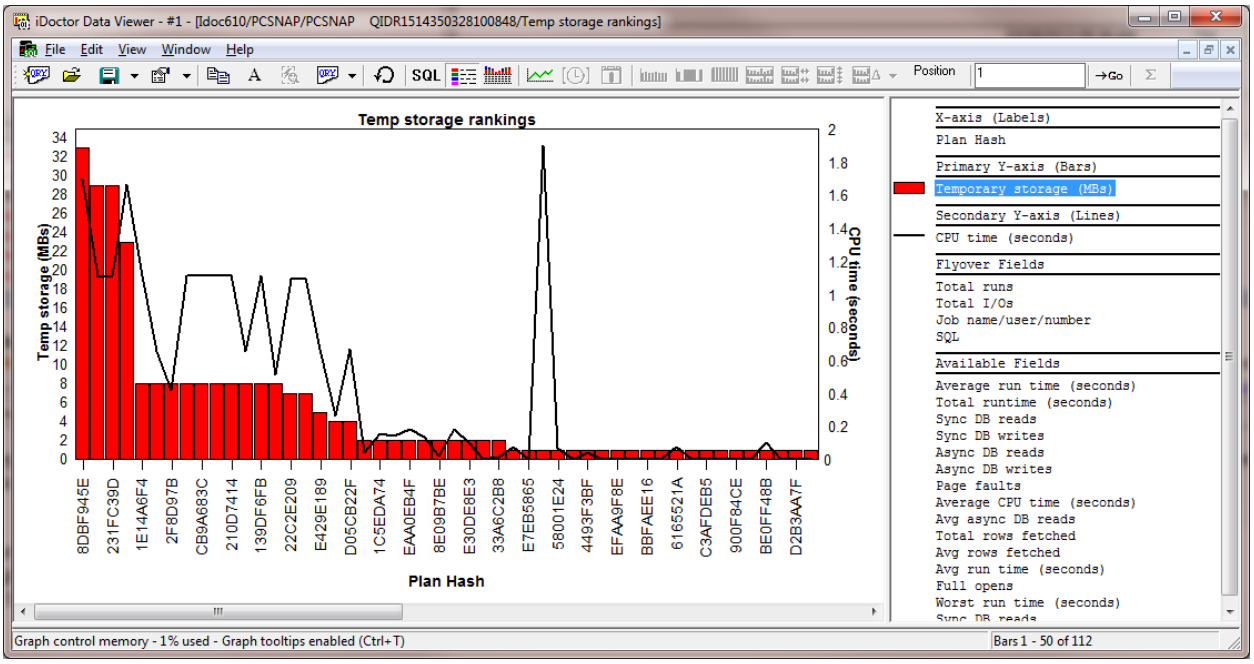
11.9.2.10 Full opens rankings



The statements are ranked by the total number of full opens.

Total full opens and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

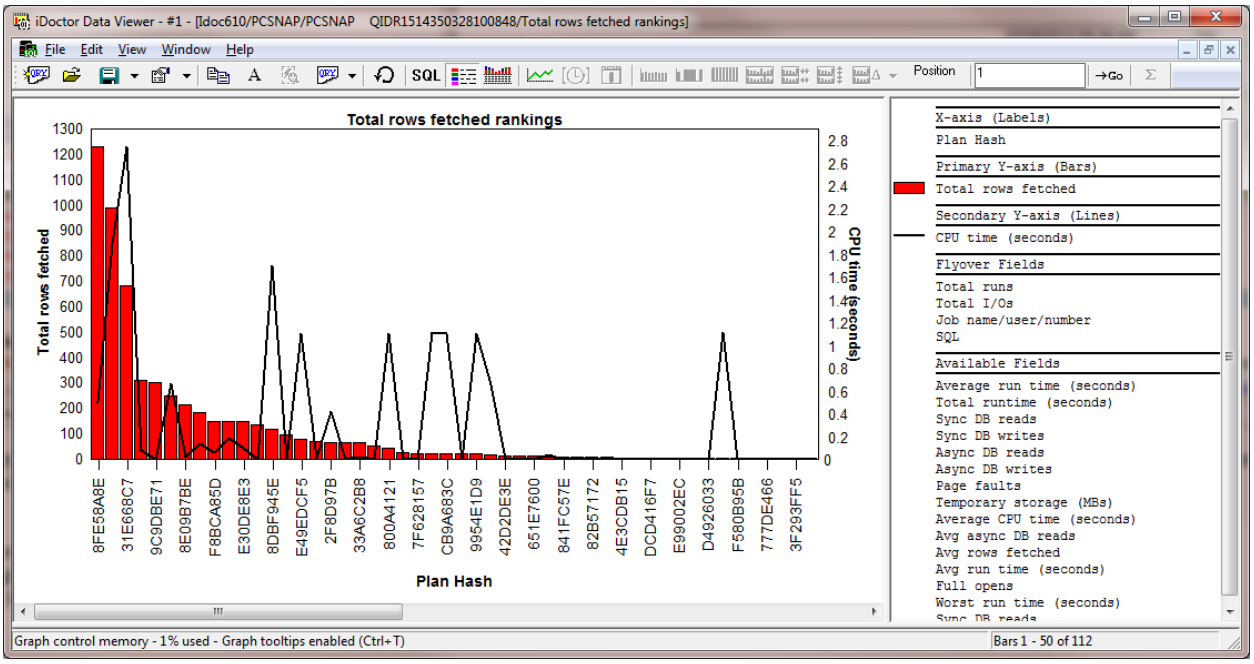
11.9.2.11 Temp storage rankings



The statements are ranked by the total temporary storage used (in megabytes.)

Temporary storage and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

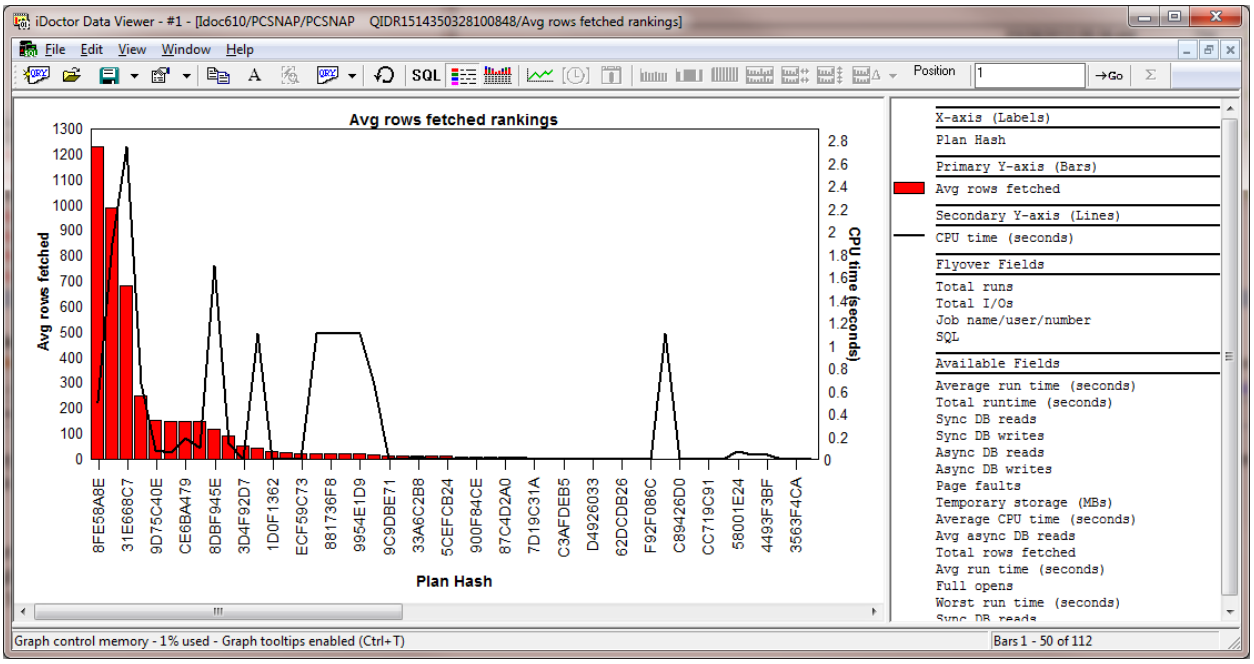
11.9.2.12 Total rows fetched rankings



The statements are ranked by the total number of rows fetched.

Total rows fetched and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

11.9.2.13 Average rows fetched rankings



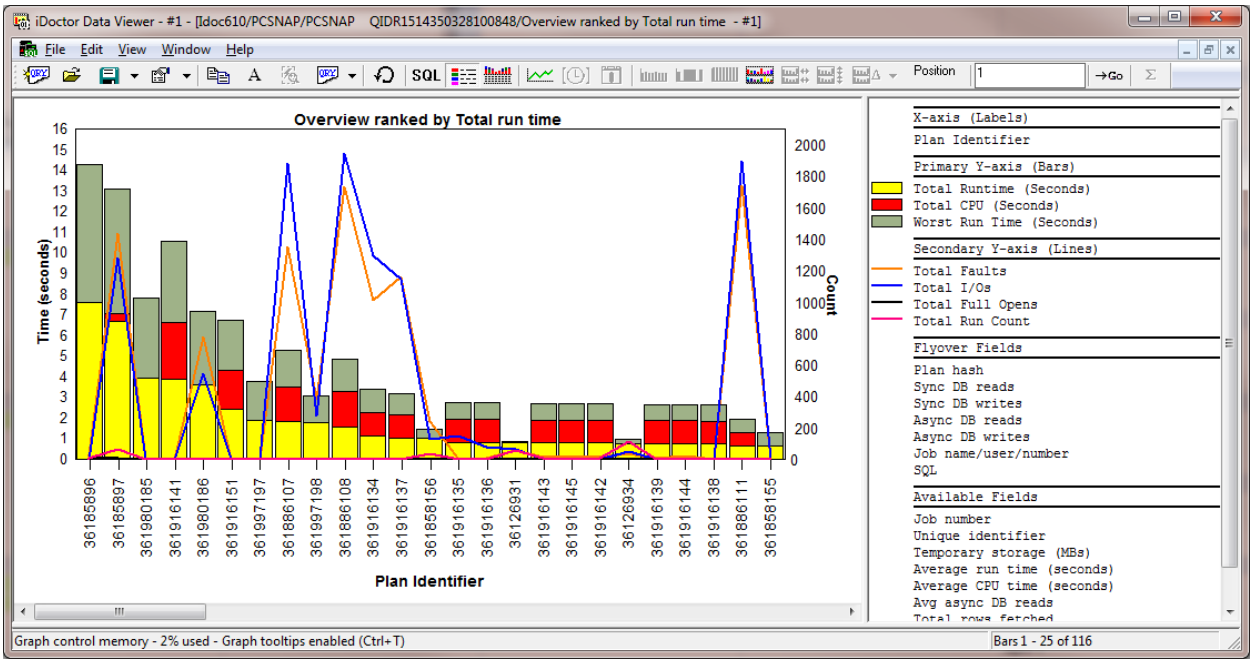
The statements are ranked by the average number of rows fetched.

Average rows fetched and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

11.9.3 Plan Graphs

These graphs rank the data in the plan cache snapshot by Plan ID in various ways to get an idea of the relative performance contributions in terms of CPU, run time, I/Os and more.

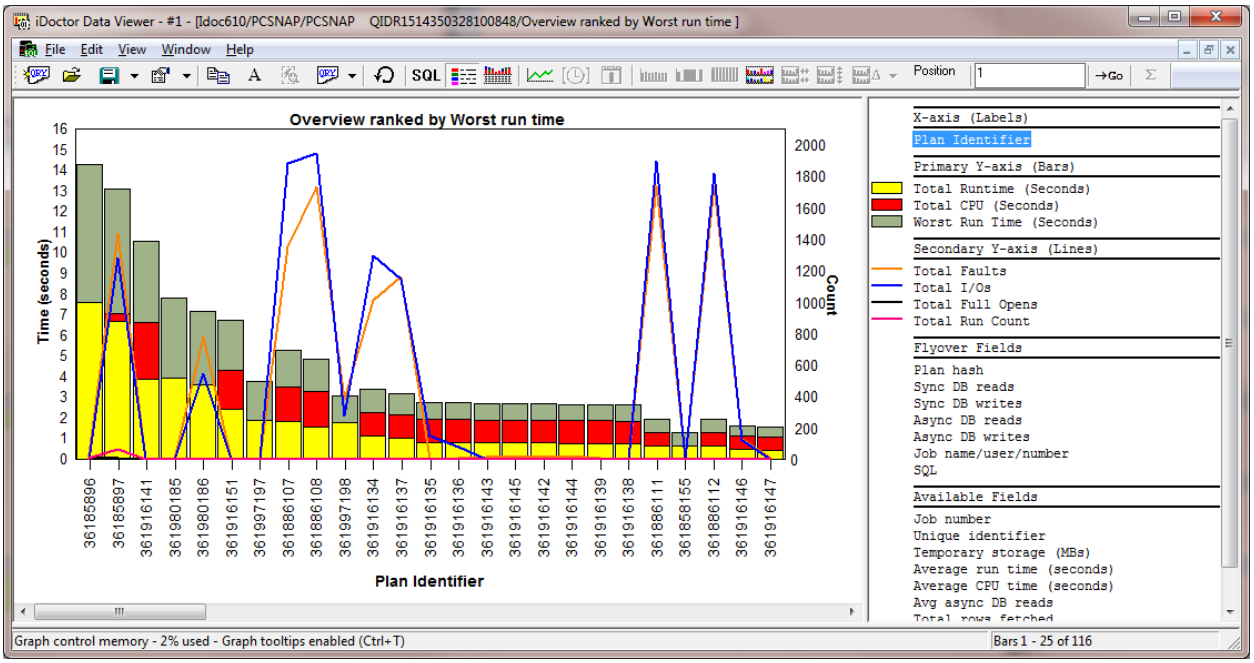
11.9.3.1 Overview ranked by Total run time



Overview ranked by Total run time

The plans are ranked by Total run time (yellow bar) in this graph. Faults, I/Os, opens, run count, CPU and run times are shown.

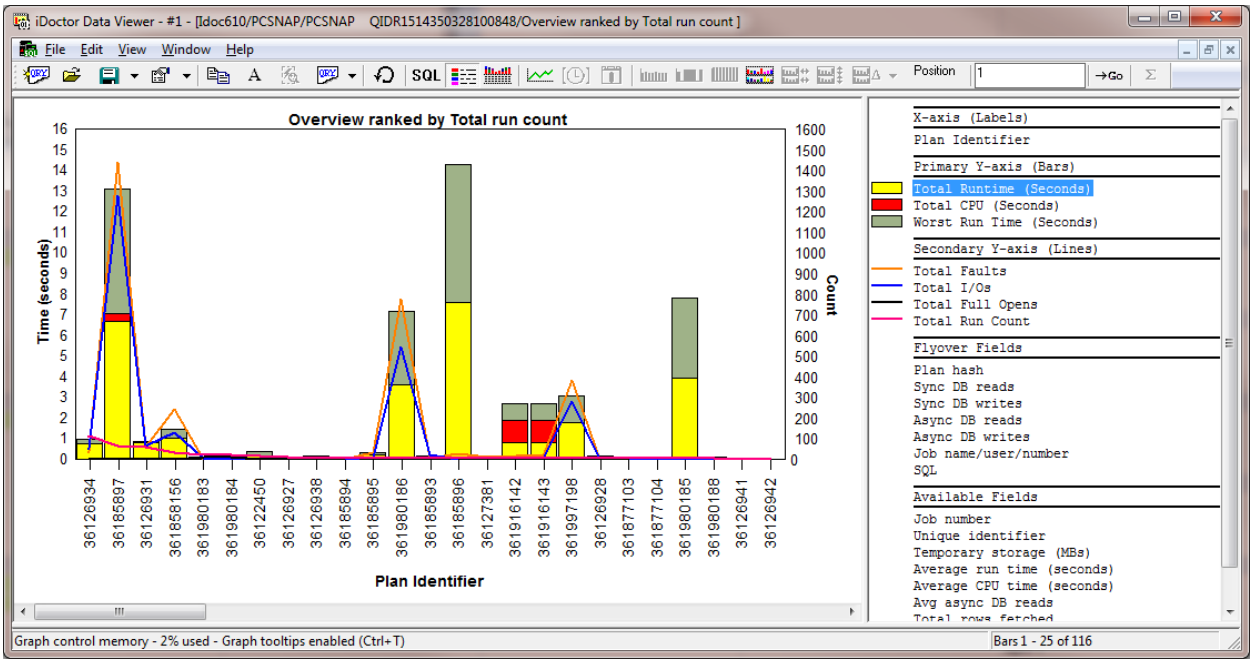
11.9.3.2 Overview ranked by Worst run time



Overview ranked by Worst run time

The plans are ranked by the worst run time (grey bar) in this graph. Faults, I/Os, opens, run count, CPU and run times are shown.

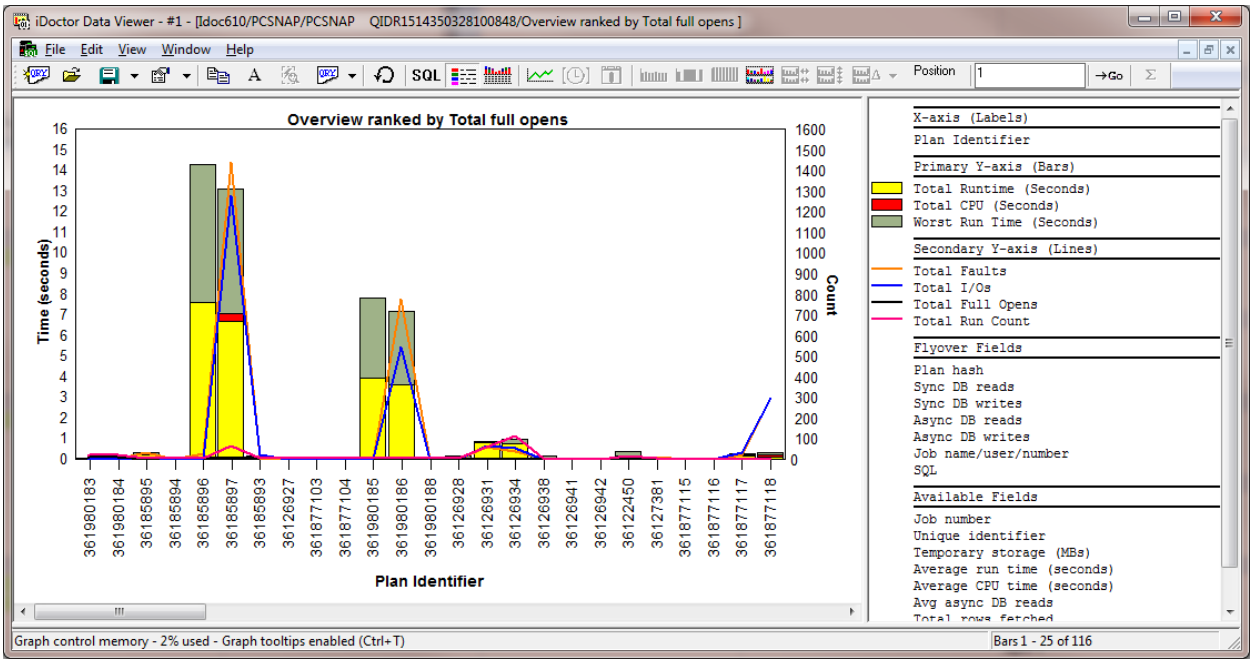
11.9.3.3 Overview ranked by Total run count



Overview ranked by Total run count

The plans are ranked by the one that ran the most times. Faults, I/Os, opens, run count, CPU and run times are shown.

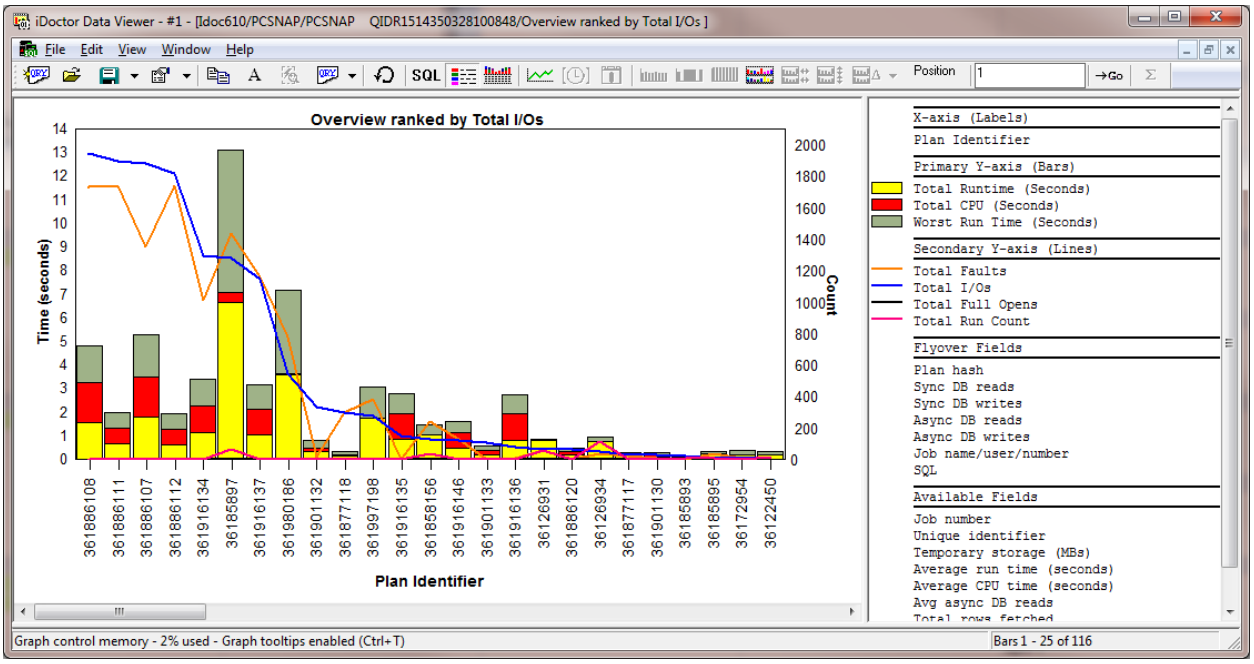
11.9.3.4 Overview ranked by Total full opens



Overview ranked by Total full opens

The plans are ranked by total full opens (black line). Faults, I/Os, opens, run count, CPU and run times are shown.

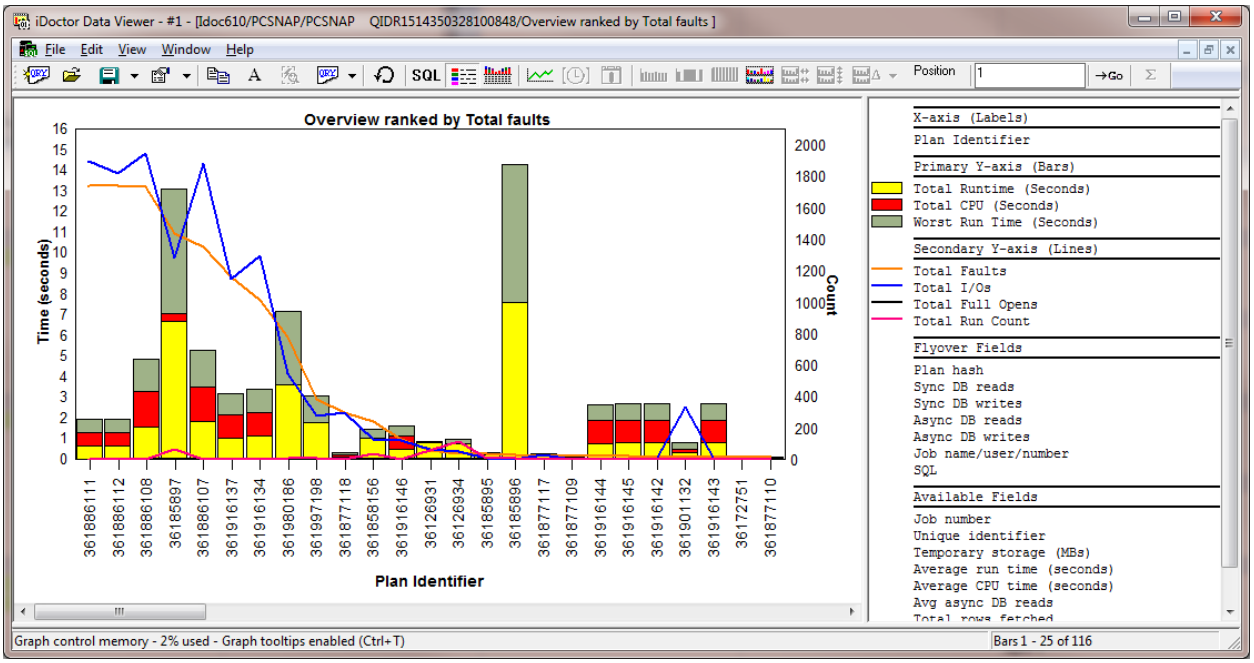
11.9.3.5 Overview ranked by Total I/Os



Overview ranked by Total I/Os

The plans are ranked by total I/Os (blue line). Faults, I/Os, opens, run count, CPU and run times are shown.

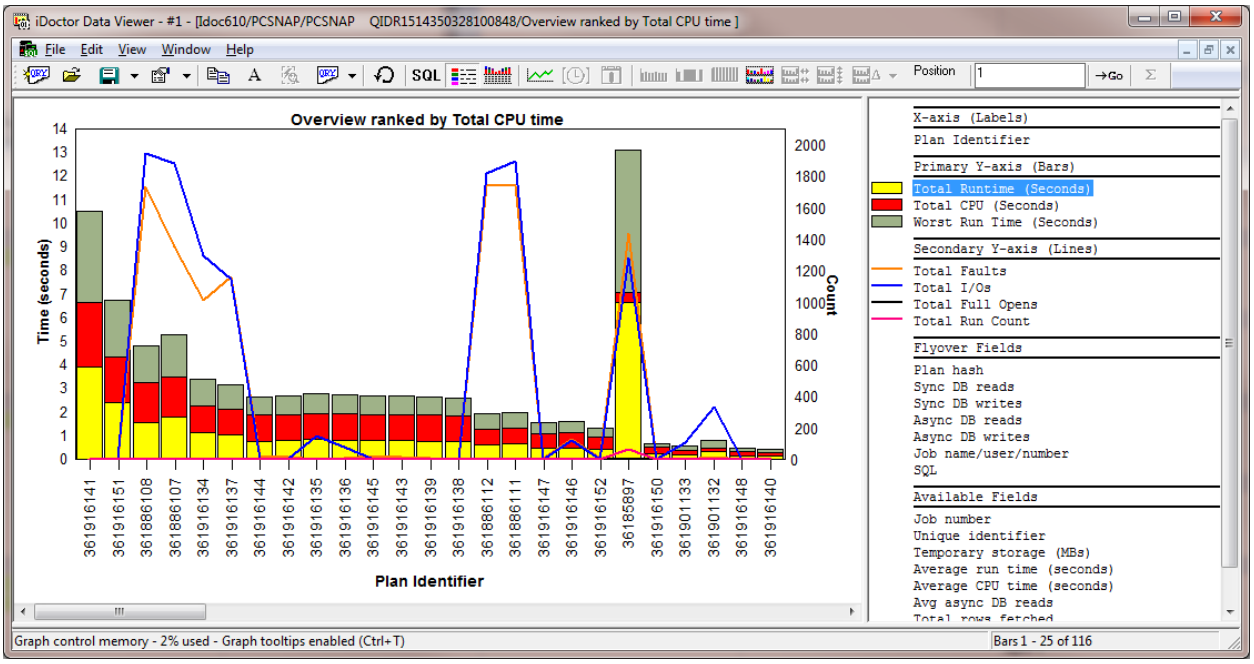
11.9.3.6 Overview ranked by Total faults



Overview ranked by Total faults

The plans are ranked by total faults (orange line). Faults, I/Os, opens, run count, CPU and run times are shown.

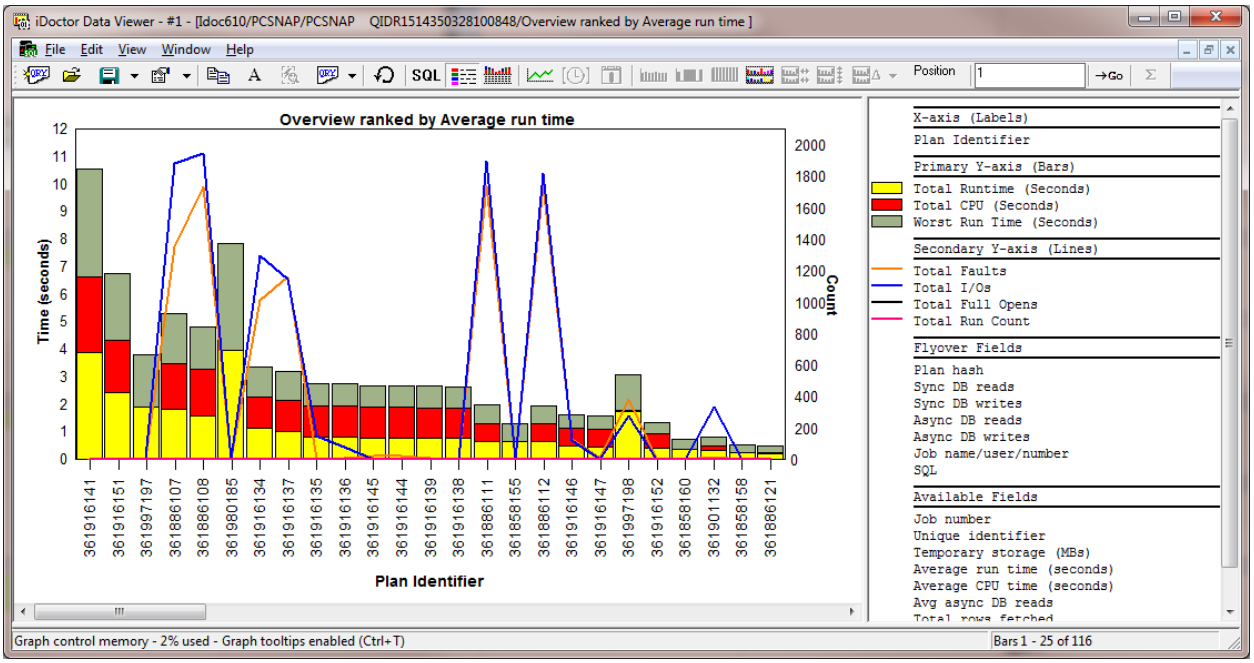
11.9.3.7 Overview ranked by Total CPU time



Overview ranked by Total CPU time

The plans are ranked by total CPU time (red bar). Faults, I/Os, opens, run count, CPU and run times are shown.

11.9.3.8 Overview ranked by Average run time

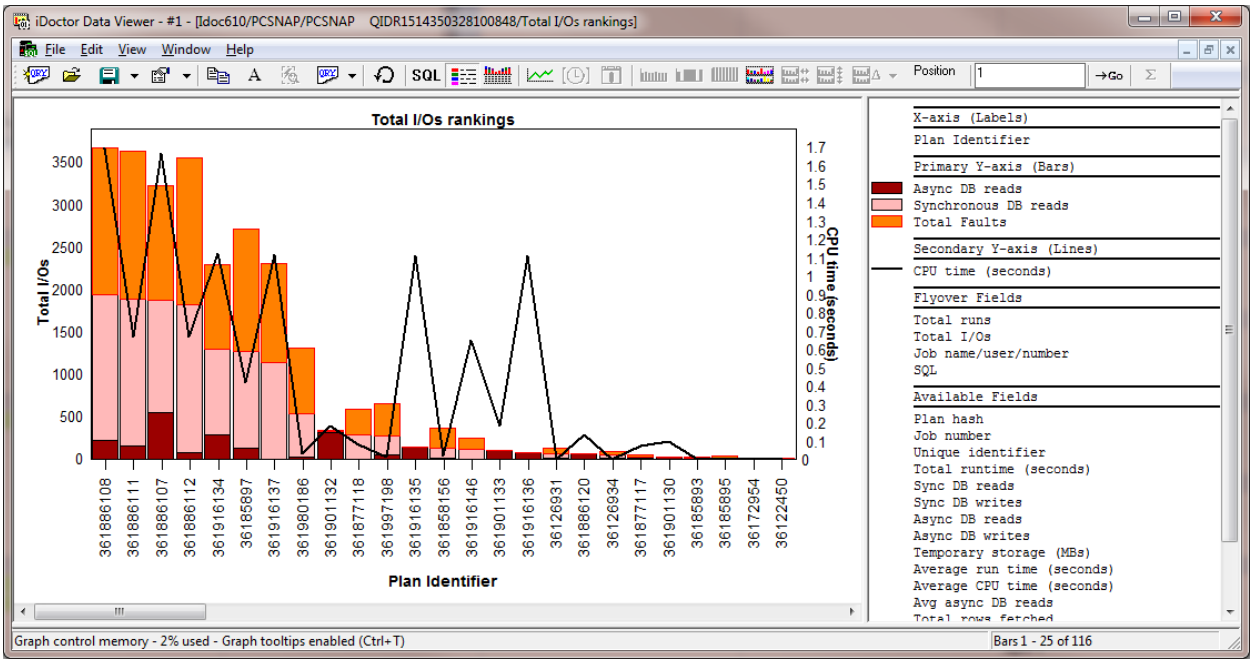


Overview ranked by Average run time

The plans are ranked by average run time (not shown but in available fields within the graph legend if desired)

Faults, I/Os, opens, run count, CPU and run times are shown.

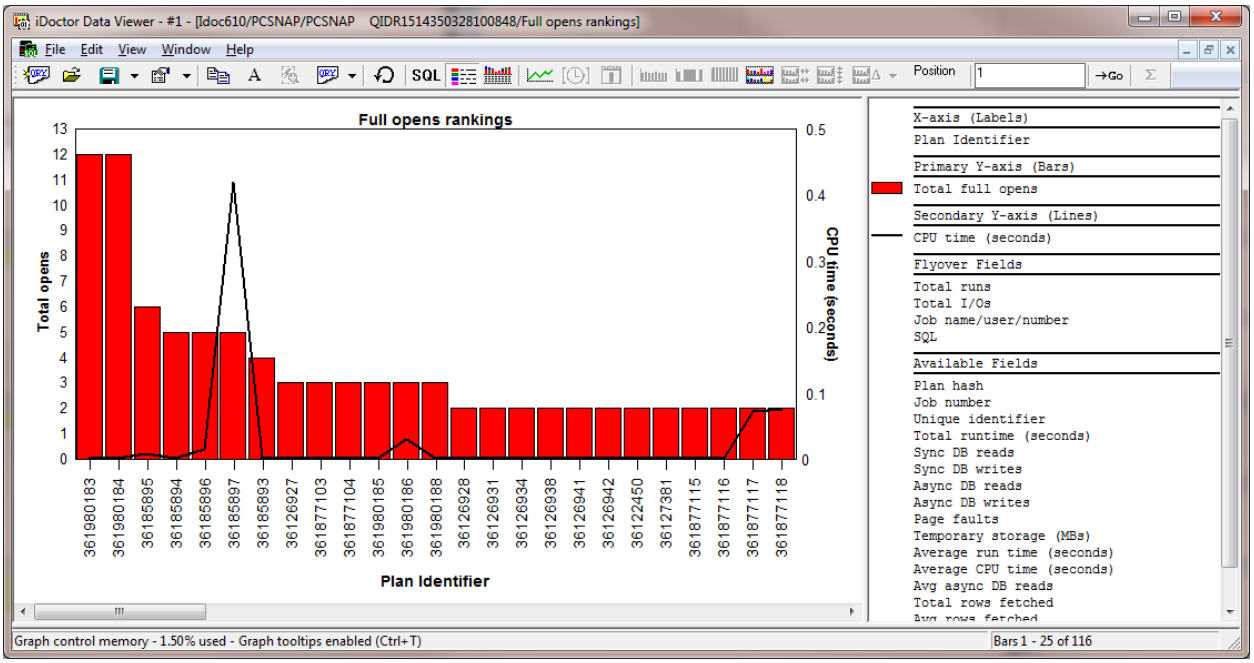
11.9.3.9 Total I/Os rankings



The plans are ranked by the total reads and faults added together.

Async DB reads, Sync DB reads total faults and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

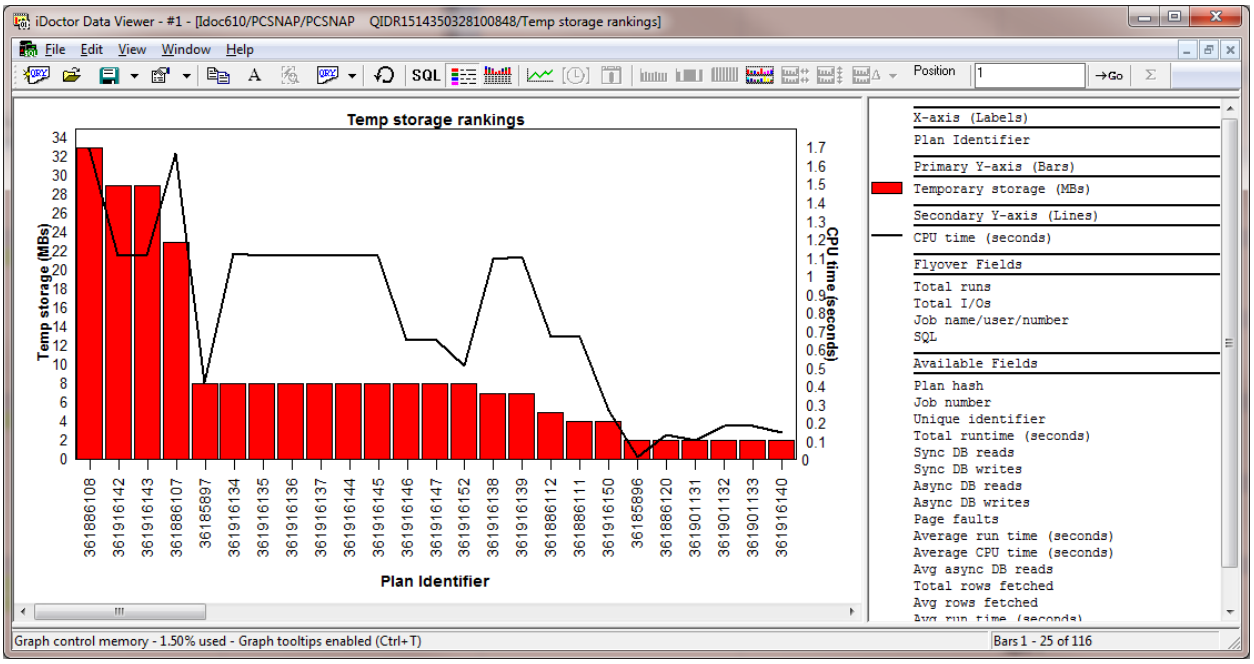
11.9.3.10 Full opens rankings



The plans are ranked by the total number of full opens.

Total full opens and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

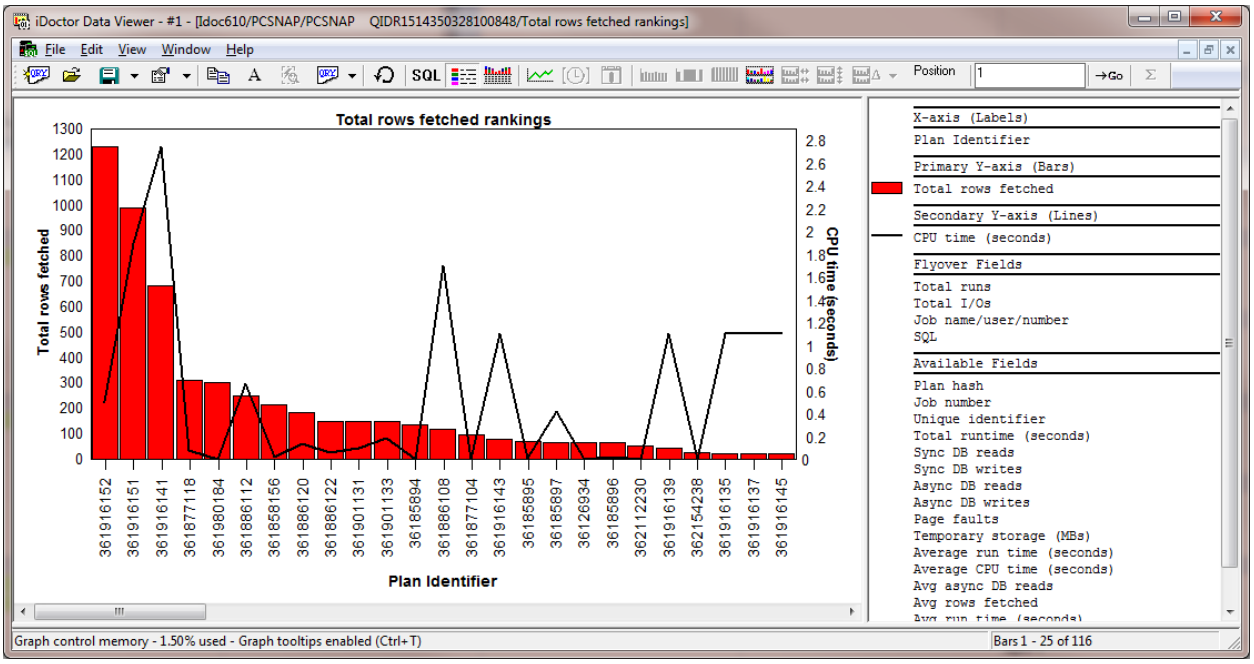
11.9.3.11 Temp storage rankings



The plans are ranked by the total temporary storage used (in megabytes.)

Temporary storage and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

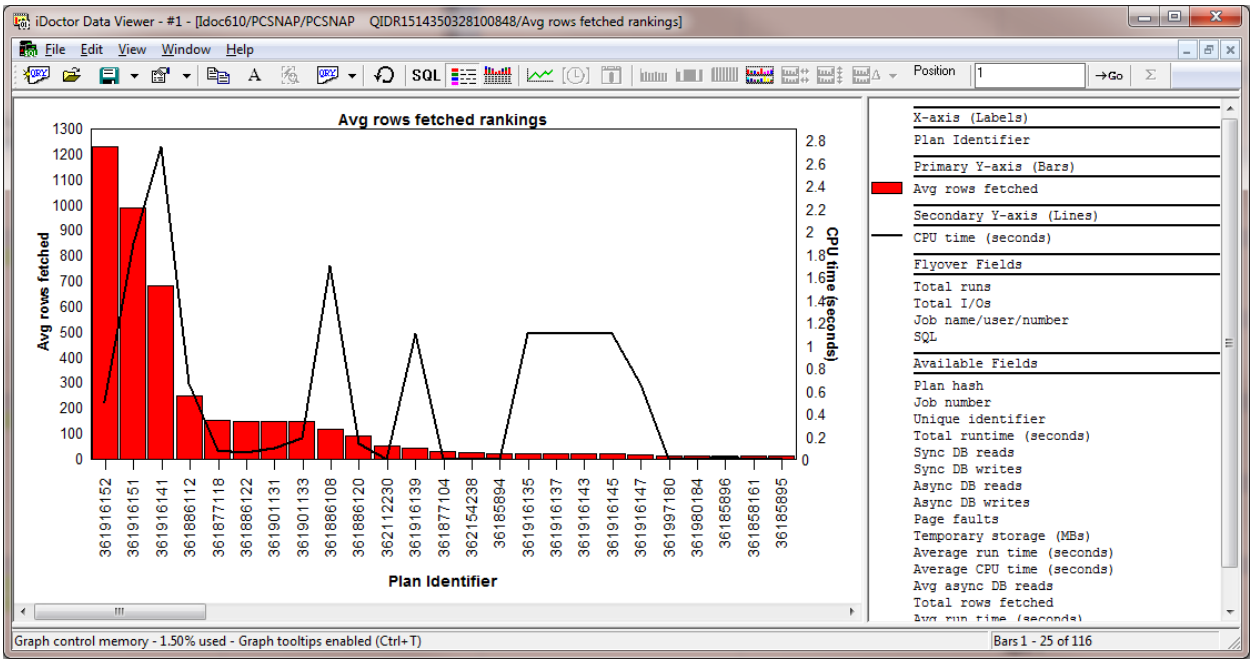
11.9.3.12 Total rows fetched rankings



The plans are ranked by the total number of rows fetched.

Total rows fetched and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

11.9.3.13 Average rows fetched rankings

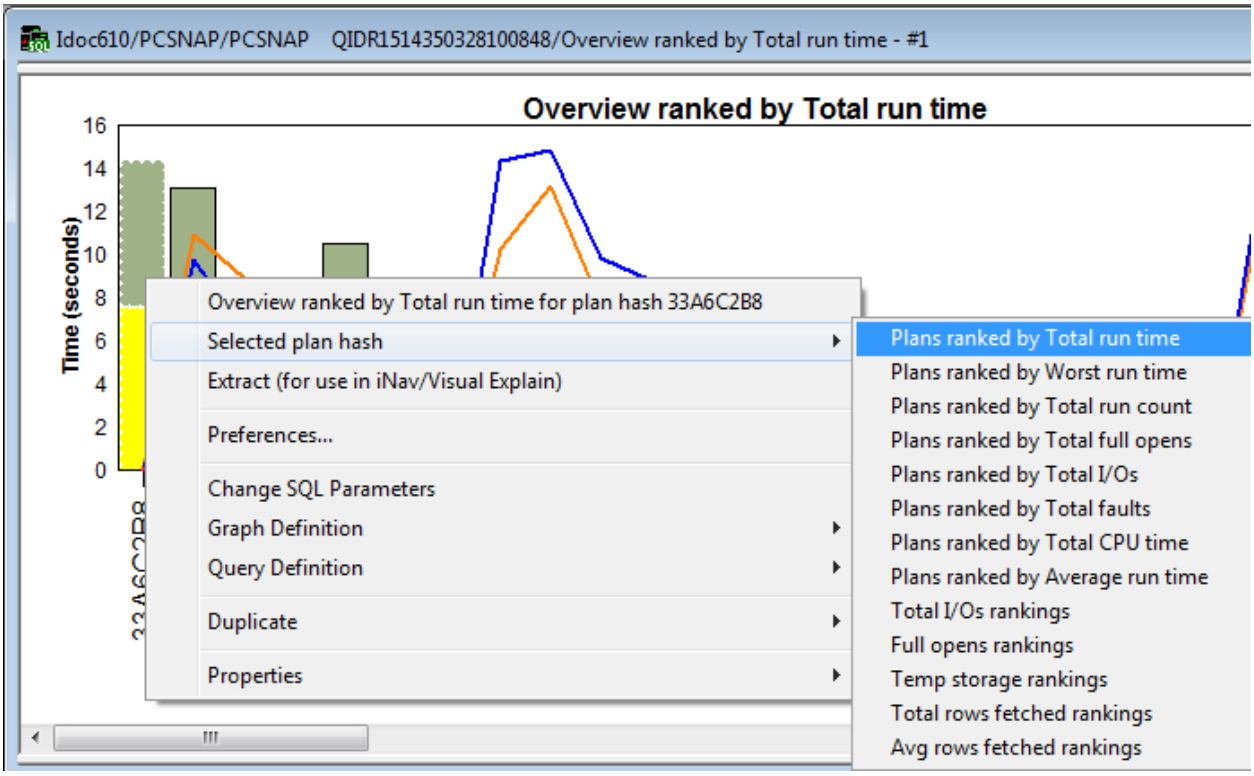


The plans are ranked by the average number of rows fetched.

Average rows fetched and CPU time are shown on this graph. Additional fields are available within the graph legend and can be added/removed as desired.

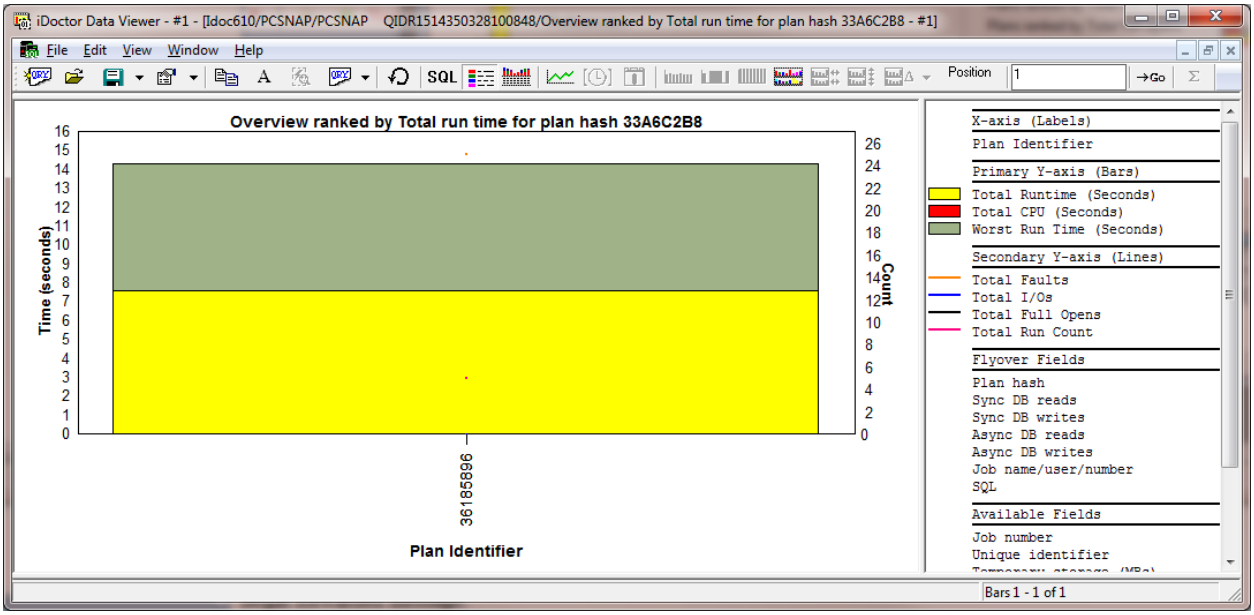
11.9.4 Statement graphs -> Selected plan hash drill down

From any of the statement graphs you can right-click the desired plan has and perform a drill down option to see the Plans associated with the plan hash.



Selected plan hash drill down options

Opening one of these drill down options will open one of the Plan graphs but only for the selected plan hash ID.



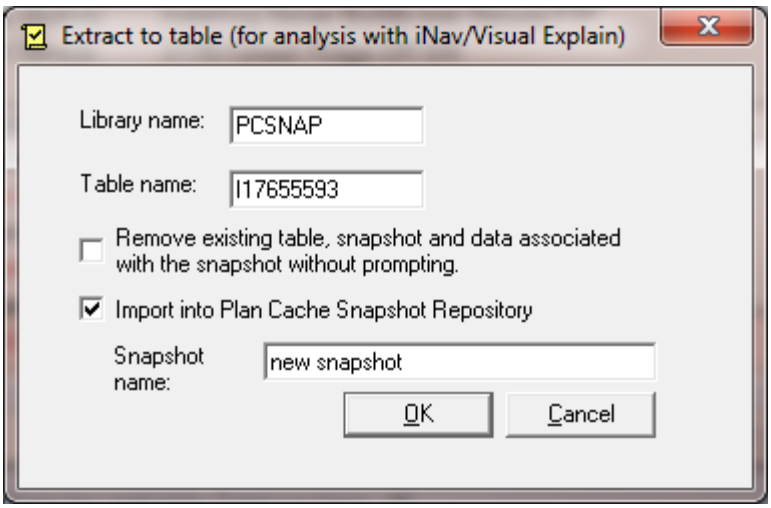
Selected plan hash drill down example graph

11.9.5 Extract function

From any of the statement or plan graphs you can select a desired plan hash or plan ID and right-click and use the Extract menu option to extract just the data for that selection into a new snapshot.

This new snapshot can be used within IBM i Navigator's Visual Explain for performance tuning and optimization of the query.

An example of the Extract window is as follows:



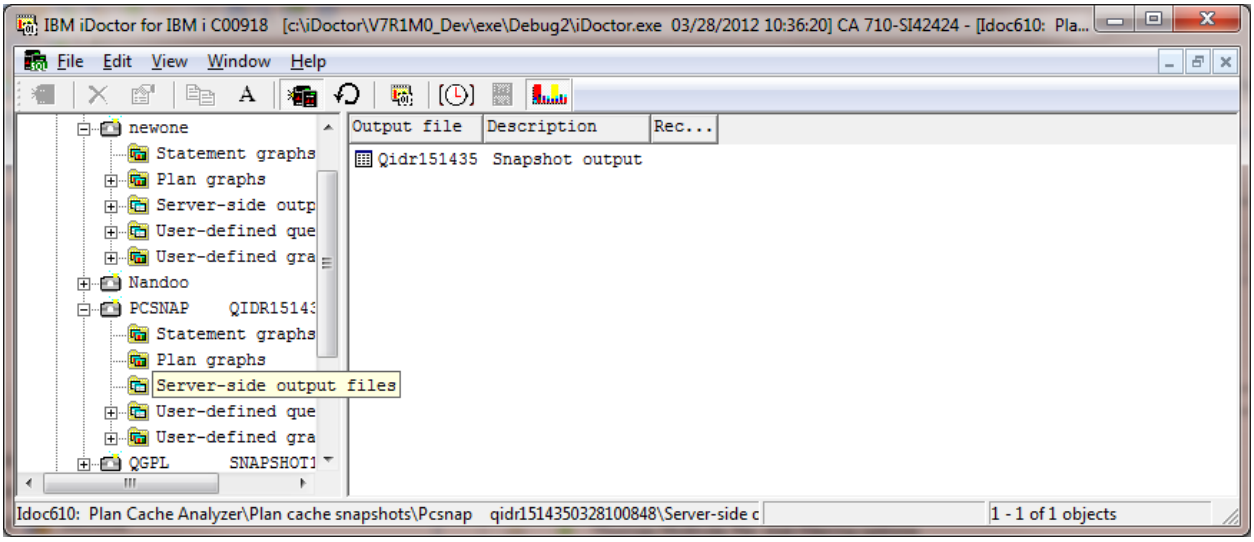
Extract to table window

Field	Value
Library name	The name of the library to store the snapshot data
Table name	The table name that will contain the new snapshot data.
Remove existing table	If checked and the table specified already exists, any existing data will be deleted without any prompting.
Import into Plan Cache Snapshot Repository	This option indicates if the plan cache snapshot data should be added to the repository. **Note:** Not selecting this option would mean the snapshot would not be visible within the Plan cache snapshots folder.
Snapshot name	This is the snapshot name or description for the new snapshot to be created.

11.10 Server-side output files

This folder contains a list of tables associated with the current snapshot.

Currently this list will always just contain 1 table (the table specified at creation time) that contains the plan cache snapshot data.



Server-side output files folder

12Server-side components

This chapter describes the libraries and important commands, programs and files that are either included with iDoctor or come with IBM i. Unless otherwise noted, this information applies to IBM i 7.2, 7.1 and 6.1 only.

12.1 Base iDoctor support (Library QIDRGUI)

QIDRGUI library contains functions/programs/commands needed in order for the GUI to function properly. Library QIDRGUI must be installed in order to use any of the iDoctor components with the GUI. In some cases the library is also necessary when running iDoctor commands in other libraries (like QIDRPA/STRPACOL) because it contains several common objects.

12.1.1 Commands

The following commands are included in library QIDRGUI:

12.1.1.1 ADDIDRUSR

Use this command to grant the desired user profile the authority to do most tasks required by iDoctor.

This command will add the specified user to the QIBM_SERVICE_TRACE, QIBM_SERVICE_JOB_WATCHER and QIBM_SERVICE_DISK_WATCHER function groups. The collection commands (STRJW, STRDW, STRPACOL) perform checks against these function groups to grant a user without *SERVICE authority the ability to run a collection.

This command will also grant authority to the following objects:

- *ALL to file QAPEXDFN in library QUSRSYS.
- *ALL to file QAPYJWDFN in library QUSRSYS.
- *ALL to file QAPYDWDFN in library QUSRSYS.
- *USE to commands ADDPEXDFN/RMVPEXDFN in library QSYS.
- *USE to commands ADDPEXFTR/RMVPEXFTR in library QSYS.
- *USE to commands ADDJWDFN/RMVJWDFN/STRJW in library QSYS.
- *USE to commands ADDDWDFN/RMVDWDFN/STRDW in library QSYS.
- *USE to commands SAVPFRCOL/STRPFRCOL/DLTPFRDTA in library QSYS.
- *USE to CS API QYPSRSCA in library QSYS.

These authorities are necessary in order for some interfaces in the iDoctor GUI to function properly.

12.1.1.2 ADDPRDACS

Add iDoctor access codes to the system.

12.1.1.3 CHKPTFS

This command can be used to check if the given list of MF (SLIC) and SI (IBM i) PTFs exist on a system. An example of calling this command on a 6.1 system is:

QIDRGUI/CHKPTFS PTFDATA('002006MF53206 MF51454 SI44916 SI42955 SI41500 SIA2726 SI3AA78 SI28986')
FOUND RELEASE LEVEL V6R1M0 FOR PRODUCT 5761999
FOUND RELEASE LEVEL V6R1M0 FOR PRODUCT 5761SS1
PTF *ONLY-SIA2726 V6R1M0 not found.
PTF SIA2726 is not loaded and applied.
PTF *ONLY-SI3AA78 V6R1M0 not found.
PTF SI3AA78 is not loaded and applied.

The first 3 characters are the number of MF PTFs.
The next 3 characters are the number of SI PTFs.
After that each 10 characters should be a PTF name (MF list first, then the SI list of PTFs.)

12.1.1.4 CRTTCPRPT

This command is used by PEX Analyzer to optionally process and analyze PEX data that contains communication events.

12.1.1.5 DLTOLDSUM

This command removes all obsolete iDoctor summaries and analyses files from a system.
The types of data that can be deleted with this command are:

*ALL: All collection types.

*JW: Job Watcher (CRTWCHSUM command output)

*PA: PEX Analyzer (All classic analyses except Taskswitch)

*CS: Collection Services (CRTCSSUM command output)

12.1.1.6 ENDJOBLCK

This command will end all jobs that have a lock on a library. This is used by the IDRINSTALL command.

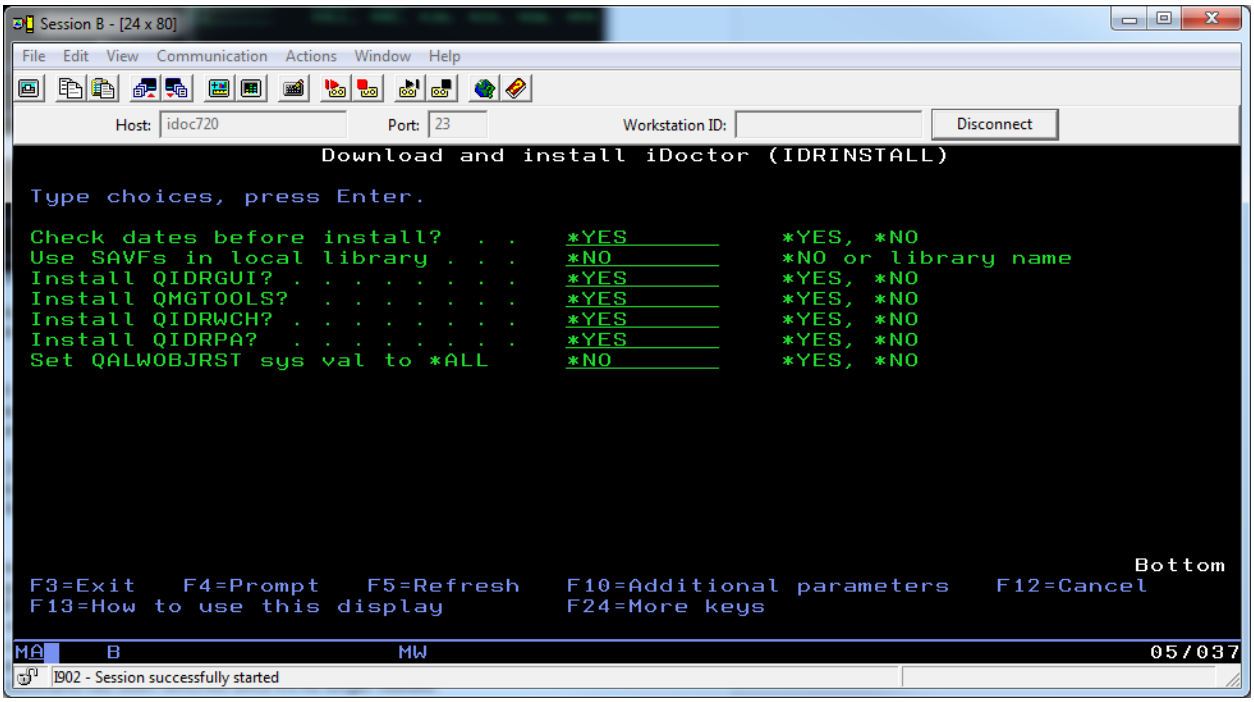
12.1.1.7 FTPFILE

This command can be used to send a file from one system to another. This is used by the IDRINSTALL command and when the GUI transfers save files.

12.1.1.8 GETJOBINFO

Returns the current job name/user/number.

12.1.1.9 IDRINSTALL



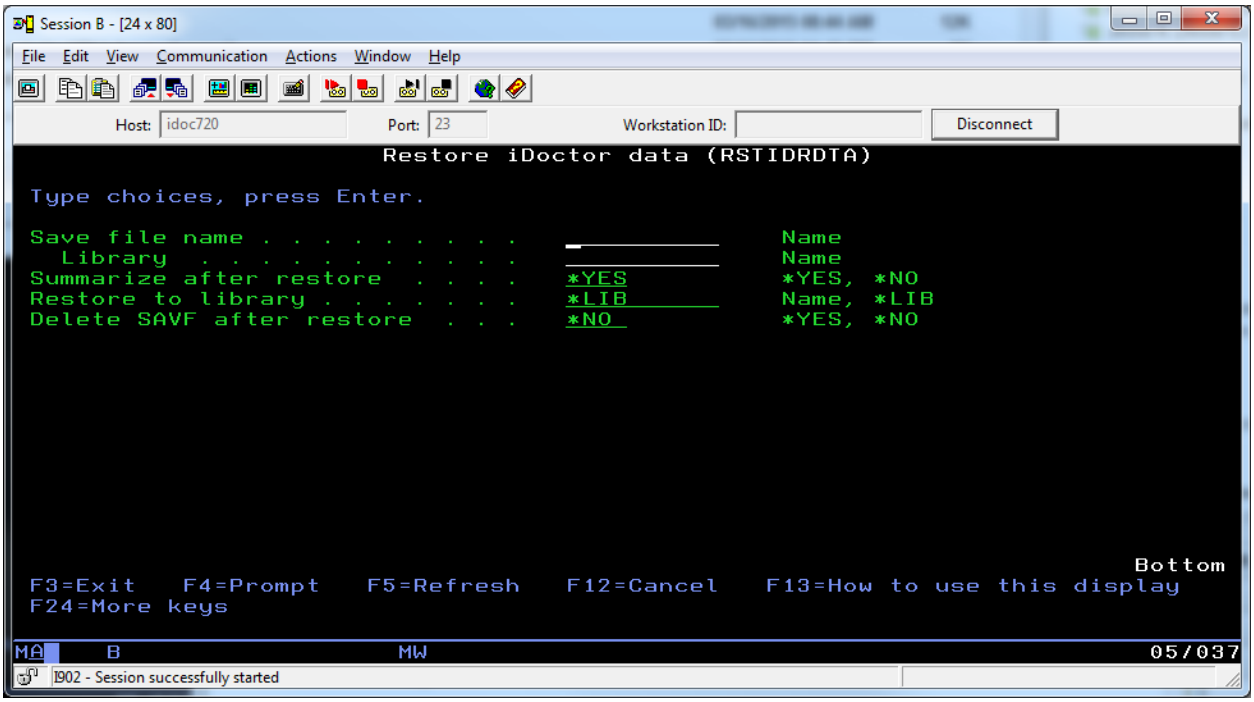
This command allows you to download and install the latest iDoctor server builds from ftp://public.dhe.ibm.com/services/us/igsc/idoctor/web/

If your system does not have access to the Internet then you may copy the SAVFs from this server to your own FTP server in order to install the server builds on partitions within your network.

12.1.1.10 RMVIDRUSR

Revokes authorities granted by ADDIDRUSR.

12.1.1.11 RSTIDRDTA



This command restores the iDoctor data found in the given save file to the desired library and afterwards will optionally run the STRIDRSUM command. You can also delete the SAVF after the restore if you have an automated process using this command.

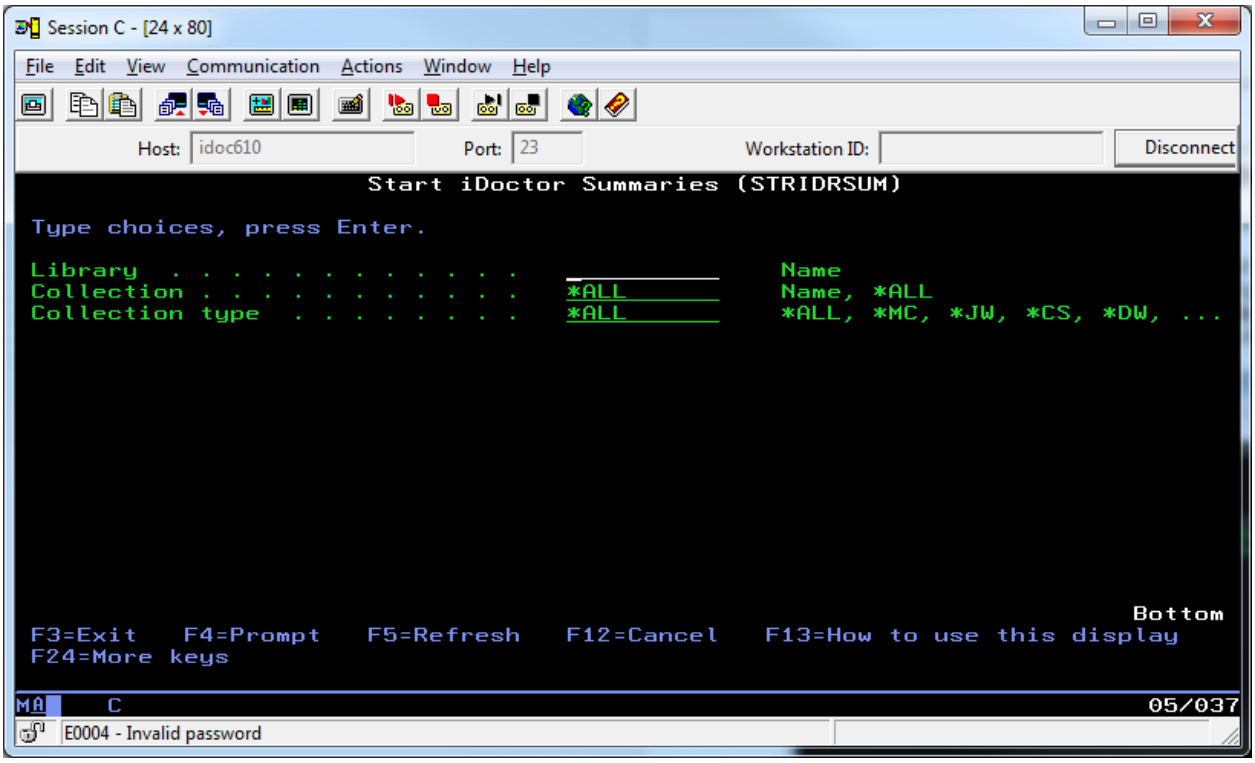
12.1.1.12 RTVSTKDTA

This command calls an RPG program to retrieve call stack data into a user space. This is used by the GUI for showing call stacks.

12.1.1.13 RUNMYSQL

Executes the supplied SQL statement in the current job. This command is now obsolete since IBM i now provides QSYS/RUNSQL that does the same thing at 6.1 and higher.

12.1.1.14 STRIDRSUM



This command will start iDoctor summary and analysis processes for the desired collection (or all of them) in the specified library.

All summarizations and analyses will be ran for each component with the normal defaults during iDoctor GUI use specified. If management collection objects (with attribute *PEX or *PFR) are found they will be expanded in the library if this has not been done yet.

Note: This command requires that the stored procedures created by the GUI on the system exist in QIDRGUI. The installation should normally create all of these however at 6.1 and higher.

12.1.2 Programs

Library QIDRGUI contains a number of important programs needed by iDoctor.

12.1.2.1 iDoctor Stored Procedures

All iDoctor SQL stored procedures and SQL functions are created in library QIDRGUI. Many of these are created by the installation at install time. At startup of the GUI, when connecting to a system the GUI will check if the stored procedures are at the required version and update it from SQL source saved with the client if necessary.

Note: The version is identified by the last 3 characters of the description on the *PGM or *SRVPGM object.

12.1.2.2 CHKEXPDATE

This program can be used to determine the current access code expiration dates from the green screen. This information is given in the GUI on the iDoctor components screen when connected to a specific IBM i.

If you call this program like this:

CALL PGM(QIDRGUI/CHKEXPDATE)

The PEX Analyzer expiration date will be found in QIDRGUI/CHKCODE20 data area.

The Job Watcher expiration date will be found in QIDRGUI/CHKCODE21 data area.

12.1.3 Files

Some of the physical files provided by iDoctor in library QIDRGUI are described in this section.

12.1.3.1 QAIDRCCINS

This file contains hardware resource information that is grouped by CCIN ID. iDoctor uses the information in this file to better describe some of the hardware configuration reports shown in Collection Services Investigator.

12.1.3.2 QAIDROT

This file provides a list of IBM i object types and descriptions.

12.1.3.3 QAIDRST

This file provides a list of IBM I segment types and descriptions.

12.2 Job Watcher (Library QIDRWCH and QSYS)

This section describes green screen commands and functionality related to Job Watcher.

12.2.1 IBM i Job Watcher Commands

At 6.1 the Job Watcher support that was only in iDoctor was added to IBM i. The following commands are part of the Job Watcher support included with IBM i in the QSYS library:

12.2.1.1 ADDJWDFN

Adds a Job Watcher definition to the system.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/addjwdfn.htm

12.2.1.2 STRJW

Starts a Job Watcher collection.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/strjw.htm

12.2.1.3 RMVJWDFN

Deletes a Job Watcher definition.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rmvjwdfn.htm

12.2.1.4 ENDJW

Ends a Job Watcher collection.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/endjw.htm

12.2.1.5 CPYPFRCOL

Copy a collection to another location.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cpypfrcol.htm

12.2.1.6 CVTPFRCOL

This command will convert a collection created at a previous release to the current release.
At 6.1 this command only supports Collection Services collections. At 7.1 and higher it also supports Disk Watcher, Job Watcher and PEX.

Note: Using this command is not recommended for use with iDoctor. iDoctor will handle your data regardless of the release it was created on (5.4 and up.)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cvtpfrcol.htm

12.2.1.7 DLTPFRCOL

Deletes one or more collections (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/dltpfrcol.htm

Note: This command will NOT remove iDoctor created analysis files (SQL Tables.).

12.2.1.8 SAVPFRCOL

Saves one or more collections to a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/savpfrcol.htm

12.2.1.9 RSTPFRCOL

Restore one or more collections from a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rstpfrcol.htm

---

12.2.2 iDoctor Job Watcher Commands

The following commands are included with Job Watcher in library QIDRWCH:

12.2.2.1 CPYJWCOL

This command is used to copy a Job Watcher collection. This action will copy every member matching the collection name from the "from" library to the "to" library. If desired the collection can be renamed by setting the TOCOL parameter with a new name.

Note: CPYPFRCOL can also perform this action, but this command is still provided to avoid possible problems where CPYPFRCOL will not allow a collection to be copied based on VRM differences of the database files.

This command does NOT also copy iDoctor created SQL Tables.

12.2.2.2 DLTJWCOL

This command is used to delete a Job Watcher collection from a user's library on a system. This action will remove the member matching the collection name from every QAPYJW* file found in the library specified.

This command also removes all iDoctor created SQL Tables.

12.2.2.3 DLTJWMON

This command is used to delete a job watcher monitor and all collections it contains on the current system. If the monitor is still running, the monitor will first be ended using the ENDJWMON command.

The record in file QGPL/QAIDRJWM1 that identifies the existence of this monitor will also be removed by this command.

12.2.2.4 DMPLONGSQL

Looks for long running SQL in an active collection and dumps plan cache

12.2.2.5 ENDJWMON

This command is used to end the job running a Job Watcher monitor and any jobs currently running collections within the monitor.

12.2.2.6 HLDJWMON

This command is used to hold a job watcher monitor. A held monitor will not delete any old collections or create any new collections until it is released using the RLSJWMON command.

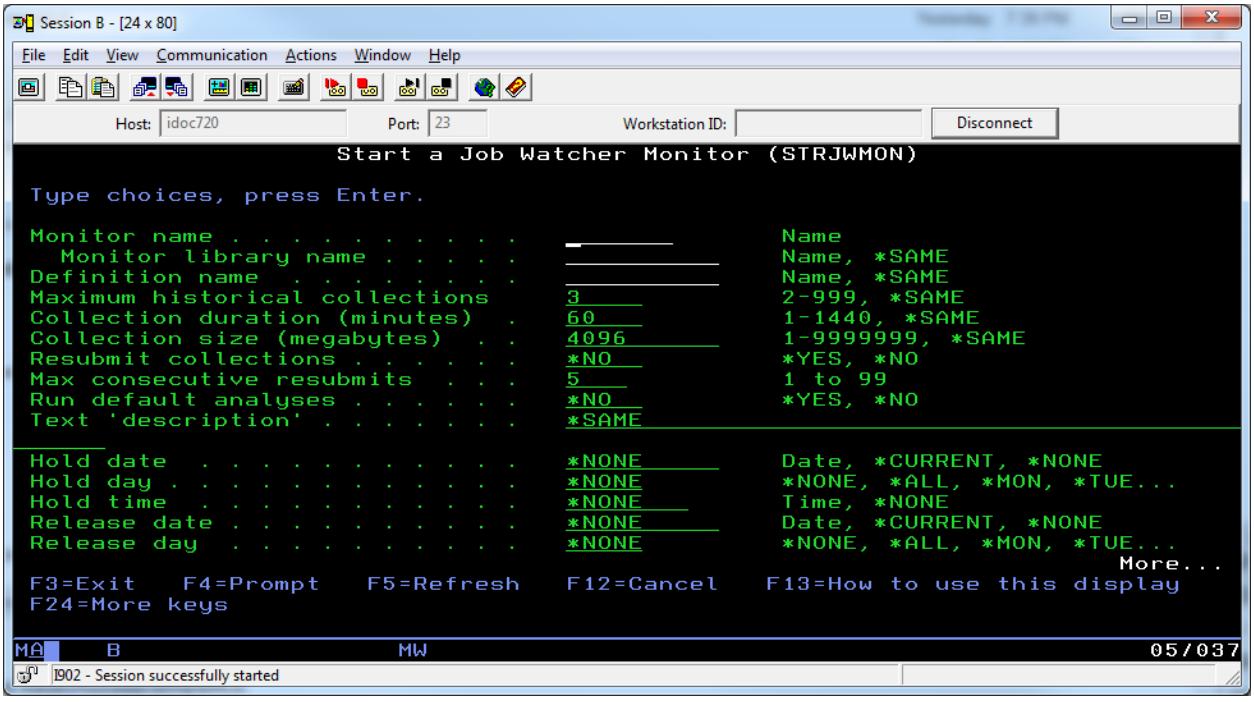
12.2.2.7 RLSJWMON

This command is used to release a monitor that is currently in a held state. Once released the monitor will continue to create new collections and delete old collections normally.

12.2.2.8 RUNPXSTATS

Runs PEX stats flat for the top N CPU jobs in a JW collection

12.2.2.9 STRJWMON



This command is used to start or restart a Job Watcher monitor. Use DLTJWMON to delete an existing monitor.

Note: This command should be submitted to batch using the SBMJOB command and not be ran interactively.

A monitor is a set of collections that continuously collect data over a system overwriting the oldest collection when it creates a new collection.

Monitors are built from a Job Watcher definition which are stored in file QUSRSYS/QAPYJWDFN. A definition indicates the parameters the monitor should use in its collections. Definitions can be created using the iDoctor client or by using the ADDJWDFN command.

The maximum historical collections parameter (COLNS) determines how many collections should be saved at one time.

A record in file QUSRSYS/QAIDRJWM2 that identifies the existence and status of the monitor is created and updated by this command.

See the command's help text for more information on this command.

12.2.3 IBM i Job Watcher Files

This section describes the Job Watcher database files that come with IBM i and is applicable to release 7.2 only. Fields that are new in release 7.2 will have >>> <<< identifiers around them.

These files are not currently documented by IBM in the IBM Knowledge Center so are provided here for your convenience.

12.2.3.1 Terminology

A few comments about terminology in this section:

Term	Description
TDE	The TDE (task dispatching element) uniquely identifies a job/task/thread running on a system. Also known as task count. In Job Watcher this is reported as a long (8 byte) integer. In Collection Services and PEX this is reported in HEX and must be converted to decimal in order to compare values with Job Watcher.
Task count	The task count uniquely identifies a job/task/thread running on a system. Also known as TDE. In Job Watcher this is reported as a long (8 byte) integer. In Collection Services and PEX this is reported in HEX and must be converted to decimal in order to compare values with Job Watcher.
Interval	An interval of data in Job Watcher is produced by producing 2 snapshots and comparing the changes that occurred in terms of metrics between the 2 snapshots. Many statistics on the system are stored in cumulative counters that go up over time, so delta calculations are done by Job Watcher and these values are placed in the DB files for each interval. Also some data produced by Job Watcher is produced by looking only at the end of the interval (or the 2nd snapshot). This includes call stacks, wait objects and holders. When snapshots occur each task count is examined one at a time by Job

	Watcher. Because of this there will be very slight timing differences involved in the data depending on how many jobs/tasks/threads are running and how well the system is running. If the system and Job Watcher is not running well then Job Watcher itself can slow down and be unable to complete the requested snapshots in the desired time frame. In those cases this can be visualized in the graphs by using the variable-width bar graphing option in the Data Viewer.
Initial thread task count	This refers to the task count for the primary thread for a job.

12.2.3.2 QAPYJWAIGP

This file provides activation group information applicable for each job/task and time interval. This data will only be collected if the ADDJWDFN, ADDDTACGY parameter includes value *ACTGRPDTL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8, 0)	1	4
ACTGRPKEY	Activation group key	B (8, 0)	5	4
TASKCOUNT	Initial thread task count This value uniquely identifies the primary thread of the job that created this activation group. Also known as TDE.	B (18, 0)	9	8
AGRESERVE	Reserved	B (8, 0)	17	4
ACTGRP	Activation group name	C (30)	21	30
AGROOTNAME	Activation group root program name	C (30)	51	30
AGROOTTYPE	Activation group root program type	H (2)	81	2
AGROOTLIB	Activation group root program library name	C (10)	83	10
AGTYPE	Activation group type	H (2)	93	2
AGSTATE	Activation group state	B (4, 0)	95	2
AGSTGMODL	Activation group storage model	B (4, 0)	97	2
AGSHARED	Activation group shared flag	B (4, 0)	99	2
AGMARK	Activation group mark	B (18, 0)	101	8
AGDFTHSIZ	Activation group default heap size	B (18, 0)	109	8
AGDFTHBLKS	Activation group default heap blocks	B (8, 0)	117	4
AGOTHERHS	Activation group other heaps	B (8, 0)	121	4

12.2.3.3 QAPYJWAIHP

This file provides activation group heap information applicable for each job/task and time interval. This data will only be collected if the ADDJWDFN, ADDDTACGY parameter includes value *ACTGRPDTL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8, 0)	1	4
ACTGRPKEY	Activation group key	B (8, 0)	5	4
TASKCOUNT	Initial thread task count This value uniquely identifies the primary thread of the job that created this activation group. Also known as TDE.	B (18, 0)	9	8
AGOTSIZE	Activation group other heaps heap size	B (18, 0)	17	8
AGOTID	Activation group other heaps heap ID	B (18, 0)	25	8
AGOTBLKS	Activation group other heaps heap block count	B (8, 0)	33	4
AGHRESERVE	Reserved	C (4)	37	4

12.2.3.4 QAPYJWAIPA

This file provides activation group program information applicable for each job/task and time interval. This data will only be collected if the ADDJWDFN, ADDDTACGY parameter includes value *ACTGRPDTL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8, 0)	1	4
ACTGRPKEY	Activation group key	B (8, 0)	5	4
TASKCOUNT	Initial thread task count This value uniquely identifies the primary thread of the job that created this activation group. Also known as TDE.	B (18, 0)	9	8
PACTNAME	Program activation program name	C (30)	17	30
PACTPGMTYP	Program activation program type	H (2)	47	2
PACTLIB	Program activation program library name	C (10)	49	10
PACTRSVD	Reserved	C (20)	59	20
PACTLICTYP	Program activation LIC activation type	H (2)	79	2

PACTRES	Reserved	H (2)	81	2
PACTFRAMES	Program activation static frame count	B (8,0)	83	4
PACTFRAMSZ	Program activation total static frame size	B (18,0)	87	8

12.2.3.5 QAPYJWBKT

This file lists the wait bucket mapping of bucket numbers and enums at the time of data collection. Only on rare occasions would this mapping/file ever change (via PTFs) within the same IBM i release.

The wait bucket mapping applies to Collection Services, Job Watcher and also PEX Taskswitch events.

Field Name	Description	Attribute	Buffer Position	Buffer Length
BUCKETNUM	Bucket number IBM i maps each type of wait (as well as CPU and CPU queuing) into a specific wait bucket. At 6.1 and higher 32 wait buckets exist on the system.	B (8, 0)	1	4
BUCKETDESC	Bucket description	C (50)	5	50
BKRESERVED	Reserved	C (2)	55	2
ENUM	Specific type of wait ID number within the IBM i OS. This is also known as the ENUM.	B (8, 0)	57	4
EYE	Eye catcher. This is special 3 character code also assigned to specific types of waits in order to more easily identify them. This code will be seen in some types of PEX events such as Taskswitch.	C (3)	61	3

12.2.3.6 QAPYJWDFN

This file is used to store the Job Watcher definitions on the system. After the ADDJWDFN command is ran for the first time a copy of this file will exist in QUSRSYS to store each definition created by user. A list of IBM-defined definitions (named Q*) will also be created after ADDJWDFN is first used.

Note: Unlike the other files in this section, this file/member will not be created along with each Job Watcher collection created.

Field Name	Description	Attribute	Buffer Position	Buffer Length
JWDFNNAME	Definition name	C (10)	1	10
JWDFNDESCR	Definition description	C (50)	11	50
JWDFNVERS	Definition version	B (4, 0)	61	2
JWDFN	Definition	C (2800)	63	2800

	This field stores a binary version of the definition.			
JWDFNCMD	Definition command string This value contains all the parameters used on the ADDJWDFN command string when this definition was created.	C (5002)	2863	5002

12.2.3.7 QAPYJWIJVM

This file captures IBM Technology for Java (J9) JVM statistics for each job/task running a JVM in each time interval. This data can be used to monitor JVM heap growth over time.

This data will only be collected if the ADDJWDFN, ADDDTACGY parameter includes value *JAVA.

Field Name	Description	**Attribute**	**Buffer Position**	**Buffer Length**
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (9, 0)	1	4
JMPID	Process identifier	B (9, 0)	5	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18, 0)	9	8
JMVRSN	JVM version	G (20)	17	20
JMTYPE	JVM type	C (1)	37	1
JMPOLICY	Garbage collection policy	G (30)	38	30
JMRESERVE2	Reserved	C (3)	68	3
JMSTRTIM	JVM start time	Z	71	26
JMINITSZ	Initial heap size (KB)	B (18, 0)	97	8
JMMAXSZ	Maximum heap size (KB)	B (18, 0)	105	8
JMHEAPC	Current heap allocated (KB)	B (18, 0)	113	8
JMHEAPU	Heap in use (KB)	B (18, 0)	121	8
JMMLCMEM	Malloc memory size (KB)	B (18, 0)	129	8
JMINTMEM	Internal memory size (KB)	B (18, 0)	137	8
JMJITMEM	JIT memory size (KB)	B (18, 0)	145	8
JMSCLMEM	Shared class size (KB)	B (18, 0)	153	8
JMGCLNBR	GC cycle number	B (9, 0)	161	4
JMGCREASON	GC reason	B (9, 0)	165	4
JMGCAREA	GC area	B (9, 0)	169	4

JMGCCMPRSN	GC compaction reason	B (9, 0)	173	4
JMTGCTTME	Total GC time (ms)	B (18, 0)	177	8
JMGCLTME	GC time last cycle (ms)	B (18, 0)	185	8
JMGCMRKDUR	GC mark duration (ms)	B (18, 0)	193	8
JMGCSWPDUR	GC sweep duration (ms)	B (18, 0)	201	8
JMGCCMPDUR	GC compaction duration (ms)	B (18, 0)	209	8
JMGCALCH	GC allocated heap space start (bytes)	B (18, 0)	217	8
JMGCALCHP	GC allocated heap space end (bytes)	B (18, 0)	225	8
JMGCHPSTR	GC total heap space start (bytes)	B (18, 0)	233	8
JMGCHPEND	GC total heap space end (bytes)	B (18, 0)	241	8
JMGCHPFSTR	GC free heap space start (bytes)	B (18, 0)	249	8
JMGCHPFEND	GC free heap space end (bytes)	B (18, 0)	257	8
JMGCSFRCLR	GC soft references cleared	B (9, 0)	265	4
JMGCWKRCLR	GC weak references cleared	B (9, 0)	269	4
JMGCFNRCLR	GC finalizer refs cleared	B (9, 0)	273	4
JMGCPHRCLR	GC phantom refs cleared	B (9, 0)	277	4

12.2.3.8 QAPYJWIJVS

This file captures IBM Technology for Java (J9) call stacks for each job/task running a JVM in each time interval. Multiple records will be created in this file per job per interval with each record representing 1 level/frame of the call stack within the PASE portion of the call stack for the job.

This data will only be collected if the ADDJWDFN, ADDDTACGY parameter includes value *JAVASTACK.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (9, 0)	1	4
JSFRAMENBR	Frame number	B (9, 0)	5	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18, 0)	9	8
JSMMIF	MMI interpreted Java method	C (1)	17	1
JSJITF	JIT compiled Java method	C (1)	18	1

JSITRF	Interpreted Java method	C (1)	19	1
JSDIREXF	Direct execution Java method	C (1)	20	1
JSJVMETHOD	Java method name	G (8002)	21	8002

12.2.3.9 QAPYJWIJVT

This file captures IBM Technology for Java (J9) thread information for each job/task running a JVM in each time interval.

This data will only be collected if the ADDJWDFN, ADDDTACGY parameter includes value *JAVA.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (9, 0)	1	4
JTOBJHCNT	Java objects held count	B (9, 0)	5	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18, 0)	9	8
JTHKTID	Holder kernel thread ID	B (18, 0)	17	8
JTSTATE	Java thread state	B (9,0)	25	4
JTTHDNAME	Java thread name	G (12002)	29	12002
JTOBJWT	Java object currently waited on name	G (12002)	12031	12002

12.2.3.10 QAPYJWINTI

This file is used to identify each interval of data captured by Job Watcher in the collection.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8, 0)	1	4
ISTARTTOD	Ending snapshot start time of day	Z	5	26
IENDTOD	Ending snapshot end time of day	Z	31	26
SYSTDECNT	System TDE count Reflects the total number of jobs/tasks/threads that were running on the system at the time the sample was taken.	B (8, 0)	57	4
SELTDECNT	Selected TDE count The number of jobs/tasks/threads that matched the thread	B (8, 0)	61	4

	selection criteria on the ADDJWDFN and/or STRJW command.			
ASELTDECNT	Active selected TDE count The number of jobs/tasks/threads that matched the thread selection criteria and consumed CPU during the interval. These TDEs will be reported in the QAPYJWTDE file.	B (8, 0)	65	4
EXMTDECNT	Maximum system task count	B (8, 0)	69	4
ICRITSTAT	Conditional criteria status 1 = condition met 0 = condition not met This field indicates if the conditions defined in the definition on the CONDCTLF parameter (Condition control file) were met during this interval.	C (1)	73	1
IRESERVED	Reserved	C (7)	74	7
INTUSECS	Elapsed interval time in microseconds This value is the actual duration of the interval. Even if collecting data at a specified duration (such as 5 seconds), the value shown here only rarely will be exactly that value.	B (18, 0)	81	8

12.2.3.11 QAPYJWJVM

This file contained classic JVM data at releases 6.1 and earlier. It is no longer used at releases 7.1 and higher.

12.2.3.12 QAPYJWJVTH

This file contained classic JVM thread data at releases 6.1 and earlier. It is no longer used at releases 7.1 and higher.

12.2.3.13 QAPYJWPRC

This file contains 1 record for each primary (initial) thread of each job in every interval of the collection.

Certain statistics here are only present in this file and apply to all threads of the job unlike the statistics found in the QAPYJWTDE file.

Field Name	Description	Type	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
PRESERVE1	Reserved	C (4)	5	4
TASKCOUNT	Initial thread task count This number uniquely identifies the primary job running on the system. Also known as TDE.	B (18, 0)	9	8
JOBSBS	Job subsystem	C (10)	17	10

JOBTYPE	Job type	C (1)	27	1
JOBFNCTN	Job function	C (14)	28	14
JOBSTATUS	Job status	C (4)	42	4
PRESERVE2	Reserved	C (3)	46	3
DELTAPRCPU	Job CPU in microseconds	B (8, 0)	49	8
ACTTHREADD	Threads active	B (8, 0)	57	4
ACTTHREADC	Total threads active since job start	B (8, 0)	61	4
CRTTHREADD	Threads created	B (8, 0)	65	4
CRTTHREADC	Total threads created since job start	B (8, 0)	69	4
LDIOWRT	LDIO writes	B (8, 0)	73	4
LDIORD	LDIO reads	B (8, 0)	77	4
LDIOOTHR	LDIO other non reads/writes	B (8, 0)	81	4
LDIOUPD	LDIO updates **Note:** This field is believed to not be implemented and will always be zero.	B (8, 0)	85	4
LDIODEL	LDIO deletes **Note:** This field is believed to not be implemented and will always be zero.	B (8, 0)	89	4
LDIOFEOD	LDIO FEODs	B (8, 0)	93	4
LDIOCOMIT	LDIO commits	B (8, 0)	97	4
LDIOROLLB	LDIO rollbacks	B (8, 0)	101	4
LDIOOPEN	LDIO opens **Note: As of this writing it is unknown if this value refers to native or SQL opens. Will update this once this becomes known.**	B (8, 0)	105	4
LDIOCLOSE	LDIO closes **Note:** This field is believed to not be implemented and will always be zero.	B (8, 0)	109	4
LDIOIXBLD	LDIO index builds	B (8, 0)	113	4
LDIOSORT	LDIO sorts	B (8, 0)	117	4
CMNWRT	Communication file writes **Note:** This field is believed to not be implemented and will always be zero.	B (8, 0)	121	4
CMNRD	Communication file reads **Note:** This field is believed to not be implemented and	B (8, 0)	125	4

	will always be zero.			
LDTAQSND	Data queue sends **Note:** This field is believed to not be implemented and will always be zero.	B (8, 0)	129	4
LDTAQRCV	Data queue receive **Note:** This field is believed to not be implemented and will always be zero. s	B (8, 0)	133	4
LDTAAOP	Data area operations **Note:** This field is believed to not be implemented and will always be zero.	B (8, 0)	137	4
LUSRSPCIOP	User space/index operations	B (8, 0)	141	4
TXAPPIQT	Application input queueing time in microseconds.	B (18, 0)	145	8
TXRSCUT	Resource usage time in microseconds.	B (18, 0)	153	8
TXDSPLRT	Display I/O response time in microseconds.	B (18, 0)	161	8
TXINQTRAN	Application input queueing transactions	B (8, 0)	169	4
TXRSCUTRAN	Resource usage transactions	B (8, 0)	173	4
TXDSPLTRAN	Display I/O transactions	B (8, 0)	177	4
IFSSYMLRD	IFS symbolic link reads	B (8, 0)	181	4
IFSDIRRD	IFS directory reads	B (8, 0)	185	4
IFSLUCHIT	IFS lookup cache hits	B (8, 0)	189	4
IFSLUCMIS	IFS lookup cache misses	B (8, 0)	193	4
IFSOPENS	IFS opens	B (8, 0)	197	4
IFSDIRCRT	IFS directory creates	B (8, 0)	201	4
IFSNDIRCRT	IFS non directory creates	B (8, 0)	205	4
IFSDIRDLT	IFS directory deletes	B (8, 0)	209	4
IFSNDIRDLT	IFS non directory deletes	B (8, 0)	213	4
SOCKRD	Socket reads	B (8, 0)	217	4
SOCKWRT	Socket writes	B (8, 0)	221	4
SOCKBRD	Socket bytes read	B (18, 0)	225	8
SOCKBWRT	Socket bytes written	B (18, 0)	233	8
OPENCURSOR	Fully opened SQL cursors	B (8, 0)	241	4
PSUCLOCURS	Pseudo closed SQL cursors	B (8, 0)	245	4
CURNUMACTG	Current activation groups	B (8, 0)	249	4

	Note: This value will only be provided if on the ADDJWDFN command parameter ADDDTACGY includes value *ACTGRPSUM. Otherwise the value will be zero.			
CURNUMACT	Current activations **Note:** This value will only be provided if on the ADDJWDFN command parameter ADDDTACGY includes value *ACTGRPSUM. Otherwise the value will be zero.	B (8, 0)	253	4
PRJVMF	JVM started	C (1)	257	1
PRJVMT	JVM type	C (1)	258	1
PRCAS	Reserved	C (255)	259	255
PRCAN	Reserved	C (255)	514	255
PRCPI	Reserved	C (255)	769	255
PRCUI	Reserved	C (255)	1024	255
PRCWN	Reserved	C (255)	1279	255
PRITFN	Reserved	C (127)	1534	127
PRITFT	Reserved	C (63)	1661	63
PRITFL	Reserved	C (63)	1724	63
PRIPSJT	Reserved	C (1)	1787	1
PRIPSJ	Reserved	C (45)	1788	45
PRLPNS	Reserved	B (8, 0)	1833	4
PRSSSTR	Reserved	Z	1837	26
PRESERVE3	Reserved	C (10)	1863	10
PRESERVE4	Reserved	B (18, 0)	1873	8
PRESERVE5	Reserved	B (18, 0)	1881	8
CLIENTJOBN	Client job name	C (28)	1889	28
CLIENTTHD	Client thread task count	B (18, 0)	1917	8
>>> PRESERVE6	Reserved	B (9, 0)	1925	4 <<<
>>> JBSPLFC	Spooled files created	B (18, 0)	1929	8 <<<
>>> JBSBMJOBS	Jobs submitted	B (18, 0)	1937	8 <<<
>>> JBSQLSTMT	SQL statements	B (18, 0)	1945	8 <<<
>>> JBPASCMP	SQL PAS compressions	B (18, 0)	1953	8 <<<
>>> JBPKGCMP	*SQLPKG compressions	B (18, 0)	1961	8 <<<

>>> JBLRDSQL	Logical SQL related reads	B (18, 0)	1969	8 <<<
>>> JBLWTSQL	Logical SQL related writes	B (18, 0)	1977	8 <<<
>>> JBDBUSQL	Miscellaneous SQL related operations	B (18, 0)	1985	8 <<<
>>> CURTMPSTG	Current temporary storage allocated	B (18, 0)	1993	8 <<<
>>> PEAKTMPSTG	Peak temporary storage allocated	B (18, 0)	2001	8 <<<
>>> MAXTMPSTG	Maximum temporary storage allowed	B (18, 0)	2009	8 <<<
>>> JOBINSCNT	Job instructions charged **Note:** This field might not be implemented. It appears to always be zero.	B (18, 0)	2017	8 <<<
>>> PRESERVE7	Reserved	B (9, 0)	2025	4 <<<
>>> PRESERVE8	Reserved	B (9, 0)	2029	4 <<<

12.2.3.14 QAPYJWPROC

This file identifies each program, module and procedure found in the call stacks of jobs captured by Job Watcher. Each call level of a call stack is identified by a trace back table address identifier. These identifiers are found throughout each call stack returned in the QAPYJWSTK file (1 per call level.)

In order to piece together a call stack in Job Watcher, a join from each call stack level returned in the QAPYJWSTK file (STACK field) must be made to this file by resolving the program, procedure associated with each trace back table address.

Up to a max of 1000 levels of the call stack can be returned.

Field Name	Description	Attribute	Buffer Position	Buffer Length
TBTADDR	TBT address	H (8)	1	8
PGMLIB	Program library	C (10)	9	10
PGMNAME	Program name	C (10)	19	10
MODNAME	Module name	C (10)	29	10
PROCTYPE	Procedure type 0 = SLIC 1 = NMI (i.e. “New MI”) 2 = OMI (i.e. “Original Machine Interface”) http://www.mcpressonline.com/rpg/a-more-complete-view-of-the-machine-interface-of-ibm-i.html	B (4, 0)	39	2
PROCSTRADR	Procedure start address	H (8)	41	8
PROCENDADR	Procedure end address	H (8)	49	8
PROCNAME	Procedure name	C (5002)	57	5002

12.2.3.15 QAPYJWRUNI

This file contains 1 record per collection and identifies high level information about the system and collection and the time the data collection was performed.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number This is the most recent interval of data captured (or the last interval if the collection has ended.)	B (8, 0)	1	4
STARTTOD	Start time of day	Z	5	26
ENDTOD	End time of day	Z	31	26
COLLSIZE	Data written to file size in KB	B (8, 0)	57	4
TDERCDCNT	Previous interval TDE count This is the total number of CPU using jobs/tasks/threads captured to file QAPYJWTDE in the last interval captured (see INTERVAL field.).	B (8, 0)	61	4
CYCUSEC	Cycles per microsecond This is number of system clock cycles per microsecond.	B (8, 0)	65	4
FILELEVEL	File level 0 = V5R3 < 3 = V5R4 3 = 6.1 6 = 7.1 8 = 7.2	B (4, 0)	69	2
COLLSTAT	Collector status	C (1)	71	1
CRITSTAT	Conditional criteria status	C (1)	72	1
SYSTNAME	System name	C (8)	73	8
SYSTSERIAL	System serial number	C (8)	81	8
SYSTTYPE	System type	C (4)	89	4
SYSTMODEL	System model	C (4)	93	4
NUMPROC	Number of processors	B (8, 0)	97	4
OSVRM	Operating system VRM	C (6)	101	6
CALLJOB	Calling job name	C (28)	107	28
ENDRSN	Collection end reason	C (1)	135	1
COLLNAME	Collection name	C (10)	136	10

COLLDESC	Collection description	C (50)	146	50
STRCMD	STRJW Command string	C (1002)	196	1002
DFNCMD	ADDJWDFN Command string	C (5002)	1198	5002

12.2.3.16 QAPYJWSKJB

This file identifies the jobs associated with sockets in file QAPYJWSKTC over time.

It will only be produced if the ADDJWDFN, ADDDTACGY parameter includes value *SOCKETJOBS.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
SKJBKEY	Key into QAPYJWSKTC file	B (8,0)	5	4
SKJOBNAME	Job or task using socket	C (28)	9	28

12.2.3.17 QAPYJWSKTC

This file provides socket and TCP endpoint information for each job on the system using socket related APIs over time during the collection.

It will only be produced if the ADDJWDFN, ADDDTACGY parameter includes value *SOCKETJOBS or *SOCKETTCP.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
SKJBKEY	Key into QAPYJWSKJB file	B (8,0)	5	4
TASKCOUNT	Initial thread task count This number uniquely identifies the primary job/thread running on the system. Also known as TDE.	B (18,0)	9	8
SOCRESRVAD	Reserved	C (8)	17	8
SOCRESRVDR	Reserved	B (4,0)	25	2
SOCKOBJT	Socket object type	B (4,0)	27	2
SOCRESRV1	Reserved	B (8,0)	29	4
SOCRESRVVL	Reserved	B (4,0)	33	2
SOCRESRVJC	Job count	B (4,0)	35	2
SOCKFAM	Socket family	B (8,0)	37	4

SOCKTYPE	Socket type	B (8,0)	41	4
SOCKSTATE	Socket state	B (8,0)	45	4
SOCKTPIST	Socket TPI state	B (8,0)	49	4
SOCKERR	Socket error	B (8,0)	53	4
SOCKRBUF	Socket receive buffer size in bytes	B (8,0)	57	4
SOCKRLWAT	Socket receive lowat size in bytes	B (8,0)	61	4
SOCKSBUF	Socket send buffer size in bytes	B (8,0)	65	4
SOCKRCBQ	Socket receive bytes queued	B (8,0)	69	4
SOCKLING	Socket linger time in seconds	B (8,0)	73	4
SOCKLINGO	Socket linger on/off	C (1)	77	1
SOCKRFC	Socket receive flow controlled	C (1)	78	1
SOCKEOF	Socket end of file	C (1)	79	1
SOCKSFC	Socket send flow controlled	C (1)	80	1
SOCKSECUR	Socket secure	C (1)	81	1
SOCKNONBLK	Socket non blocking	C (1)	82	1
SOCKKA	Socket keep alive	C (1)	83	1
SOCKDBG	Socket debug	C (1)	84	1
SOCKURCV	Socket receive timeout in microsecs	B (18,0)	85	8
SOCKUSND	Socket send timeout in microsecs	B (18,0)	93	8
TCPSTATE	TCP state ID	B (4,0)	101	2
TCPTPIST	TCP TPI state ID	B (4,0)	103	2
TCPUPROF	TCP user profile	C (10)	105	10
TCPRESRV1	Reserved	C (2)	115	2
TCPSRCADR	TCP source address	B (8,0)	117	4
TCPDSTADR	TCP destination address	B (8,0)	121	4
TCPSRBUF	TCP socket receive buffer size in bytes	B (8,0)	125	4
TCPRBUF	TCP receive buffer size in bytes	B (8,0)	129	4
TCPSSBUF	TCP socket send buffer size in bytes	B (8,0)	133	4
TCPSBUF	TCP send buffer size in bytes	B (8,0)	137	4
TCPFLAGS	TCP flags	H (16)	141	16

TCPCWND	TCP congestion window in bytes	B (8,0)	157	4
TCPSQLEN	TCP send queue length in bytes	B (8,0)	161	4
TCPSUNA	TCP suna in bytes	B (8,0)	165	4
TCPSNEXT	TCP bytes sent	B (8,0)	169	4
TCPUSNEXT	Application bytes sent to TCP	B (8,0)	173	4
TCPSENDWIN	TCP send window in bytes	B (8,0)	177	4
TCPSENDMAX	TCP send maximum in bytes	B (8,0)	181	4
TCPRQLEN	TCP receive queue length in bytes	B (8,0)	185	4
TCPRECVWIN	TCP receive window in bytes	B (8,0)	189	4
TCPCURRXMT	TCP current re-transmits	B (8,0)	193	4
TCPRXMTCNT	TCP re-transmits	B (4,0)	197	2
TCPRXMTTOT	TCP re-transmit total	B (4,0)	199	2
TCPRXMTFST	TCP fast re-transmits	B (4,0)	201	2
TCPRESRV2	Reserved	C (2)	203	2
TCPMAXBLOG	TCP maximum backlog	B (8,0)	205	4
TCPCURBLOG	TCP current backlog	B (8,0)	209	4
TCPLASTACK	TCP last ACK	B (8,0)	213	4
TCPSRCPT	TCP source port	B (4,0)	217	2
TCPDSTPT	TCP destination port	B (4,0)	219	2
TCPSEQNUM	TCP sequence number	B (8,0)	221	4
TCPACKNUM	TCP ACK number	B (8,0)	225	4
TCP6SRCADR	TCP IP6 source address	C (16)	229	16
TCP6DSTADR	TCP IP6 destination address	C (16)	245	16
SKTCUSECS	Elapsed interval time in microsecs	B (18,0)	261	8
SOCKFAMC	Socket family	C (13)	269	13
SOCKTYPEC	Socket type	C (9)	282	9
SOCKSTATEC	Socket state	C (13)	291	13
SOCKTPISTC	Socket TPI state	C (24)	304	24
SOCKERRC	Socket error	C (15)	328	15
SOCKDESCR	Socket descriptor	B (8,0)	343	4

SOCKHAND	Socket handle	H (8)	347	8
SOCKLCLPOR	Socket local port	B (8,0)	355	4
SOCKRMTPOR	Socket remote port	B (8,0)	359	4
SOCKLCLADR	Socket local address	C (258)	363	258
SOCKRMTADR	Socket remote address	C (258)	621	258
TCPSTATEC	TCP state	C (11)	879	11
TCPTPISTC	TCP TPI state	C (25)	890	25
TCPSRCADRC	TCP source address	C (48)	915	48
TCPDSTADRC	TCP destination address	C (48)	963	48
TCPAPPSBYD	Delta application bytes sent to TCP	B (8,0)	1011	4
TCPDBS	Delta TCP bytes sent	B (8,0)	1015	4
TCPDBR	Delta TCP bytes received	B (8,0)	1019	4
TCPDBA	Delta TCP bytes ACKed	B (8,0)	1023	4
TCPDSQL	Delta TCP send queue length	B (18,0)	1027	8
TCPRQLEND	Delta TCP receive queue length	B (18,0)	1035	8
TCPCURBLOD	Delta TCP current backlog	B (18,0)	1043	8
TCPDBSPS	TCP bytes sent per second	B (8,0)	1051	4
TCPDBRPS	TCP bytes received per second	B (8,0)	1055	4
TCPCWNDCSQ	TCP congestion window delta	B (8,0)	1059	4
TCPCWNDA	TCP congestion alert	C (1)	1063	1
TCPDSQLA	TCP delta send queue length alert	C (1)	1064	1
TCPAPPSBYA	TCP retransmits alert	C (1)	1065	1
TCPRMTRWIA	TCP remote receive window alert	C (1)	1066	1
TCPLCLRWID	TCP local receive window alert	C (1)	1067	1

12.2.3.18 QAPYJWSQL

This file contains the SQL statements detected by Job Watcher. The SQL statements found in this file will either be the currently running SQL statement or the last ran SQL statement depending on the options used in the ADDJWDFN.

Host variables parameter markers will sometimes be found within the SQL statement field. The values for the host variables are stored in a different file, QAPYJWSQLH.

WARNING: These SQL host variable can sometimes contain sensitive customer information, so use appropriate procedures to avoid compromising the sensitivity of this data. If necessary, you can use

iDoctor analysis the analysis called Destroy all host variable data in QAPYJWSQLH to permanently delete the host variable data.

This file will only be produced if the ADDJWDFN, ADDDTACGY parameter includes value *SQLSTMT, *SQLCURSTMT or *SQLDETAIL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
SQRESERVE1	Reserved	C (4)	5	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18,0)	9	8
SRCLIB	SQL package source library	C (10)	17	10
SRCFILE	SQL package source file	C (10)	27	10
SRCMBR	SQL package source member	C (10)	37	10
SRCDATE	SQL package source date	C (13)	47	13
PKGLIB	SQL package library/container	C (18)	60	18
PKGNAME	SQL package name	C (18)	78	18
RDBSNAME	Remote DBS name	C (18)	96	18
MORE	Another STMT also associated (1 = yes)	C (1)	114	1
SQRESERVE2	Reserved	C (3)	115	3
HOSTREAL	Actual number of host variables	B (4,0)	118	2
HOSTLOGGED	Number of host variables in QAPYJWSQLH	B (4,0)	120	2
STMTCSID	SQL statement CCSID	B (4,0)	122	2
SQRESERVE3	Reserved	B (4,0)	124	2
SQLSTMTLEN	SQL statement full length	B (8,0)	126	4
SQLSTMT	SQL statement	C (32002)	130	32002

12.2.3.19 QAPYJWSQLH

This file contains the host variable data for SQL statements detected by Job Watcher.

WARNING: These SQL host variable can sometimes contain sensitive customer information, so use appropriate procedures to avoid compromising the sensitivity of this data. If necessary, you can use iDoctor analysis the analysis called Destroy all host variable data in QAPYJWSQLH to permanently delete the host variable data.

This file will only be produced if the ADDJWDFN, ADDDTACGY parameter includes value *SQLSTMT, *SQLCURSTMT or *SQLDETAIL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8, 0)	1	4
HVARNUM	Number of SQL host variables returned	B (8, 0)	5	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18, 0)	9	8
HDATANUM	Host variable number	B (4, 0)	17	2
HVARTYPE	Host variable type	B (4, 0)	19	2
HDATALEN	Host variable full length	B (4, 0)	21	2
HDECIMAL	Number of decimals	B (4, 0)	23	2
HDATA	Host variable data	C (5002)	25	5002

12.2.3.20 QAPYJWSQLO

This file contains the SQL open cursor lists associated with the jobs running SQL statements in Job Watcher over time during the collection.

This file will only be produced if the ADDJWDFN, ADDDTACGY parameter includes value *SQLDETAIL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
SHRESERVE1	Reserved	C (4)	5	4
TASKCOUNT	Initial thread task count This number uniquely identifies the primary job/thread running on the system. Also known as TDE.	B (18,0)	9	8
OSQCCACT	Activation group for cursor	B (8,0)	17	4
OSQCINDX	Statement array index	B (4,0)	21	2
OSQCFILE	File name	C (10)	23	10
OSQCOAID	AuthId or library name	C (10)	33	10
OSQCCUSR	Current user at open time	C (10)	43	10
OSQCCMTLVL	Cursor commit level	C (1)	53	1
OSQCCCTXID	Context ID at open time	C (8)	54	8

OSQCPCLS	Psuedo close flag	C (1)	62	1
OSQCXTD	Extended dynamic flag	C (1)	63	1
OSQCTEMP	Space-constrained destroy at close flag	C (1)	64	1
OSQCHOLD	Cursor hold attribute flag	C (1)	65	1
OSQCRR	Repeatable read cursor flag	C (1)	66	1
OSQCCPRCED	Opened by QSQPRCED API flag	C (1)	67	1
OSQCLOBLOC	LOBs associated flag	C (1)	68	1
OSQCCMTLVE	Commit level escalated flag	C (1)	69	1
OSQCUDF	Uses UDFs flag	C (1)	70	1
OSQCMGLCKS	Managing locks flag	C (1)	71	1
WSQCHDRCLS	Always hard close flag	C (1)	72	1
WSQCSEQ	SQE processing flag	C (1)	73	1
WSQCCNTS	NTS use flag	C (1)	74	1
OSQCCSRL	Cursor name length	B (4,0)	75	2
OSQCCSR	Cursor name	C (130)	77	130
OSQCSTMTL	Prepared statement name length	B (4,0)	207	2
OSQCSTMT	Prepared statement name	C (130)	209	130
OCLSTCSID	SQL statement CCSID	B (5,0)	339	4
OCLSTMTLEN	SQL statement full length	B (8,0)	343	4
OCLSTMTTXT	SQL statement	C (32002)	347	32002

12.2.3.21 QAPYJWSQLP

This file contains the SQL prepared statement areas associated with the jobs running SQL statements in Job Watcher over time during the collection.

This file will only be produced if the ADDJWDFN, ADDDTACGY parameter includes value *SQLDETAIL.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
SPRESERVE1	Reserved	C (4)	5	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18,0)	9	8

PSQPSALC	Number of allocated entries	B (8,0)	17	4
PSQPSNUM	Number of in-use entries	B (8,0)	21	4
PSQPPSASZ	Size of prepared statement area	B (8,0)	25	4
PSQPSUBP	*ENDSQL option flag	C (1)	29	1
PSQPENDJ	*ENDJOB option flag	C (1)	30	1
PSQPNOCC	No CCSID for any host variables flag	C (1)	31	1
PSQPENDACT	*ENDACTGRP specified flag	C (1)	32	1
PSQPCMPTHR	Compression threshold	B (8,0)	33	4
PSQPDUMCNT	Dummy statement count	B (8,0)	37	4
PSQPRTNTYP	Routine type	B (4,0)	41	2
PSQPCMPCNT	Number of compresses	B (4,0)	43	2
PSQPLSTIDX	Last index in area	B (8,0)	45	4
SPRESERVE2	Reserved	C (1)	49	1
PSQP_SWC	In system-wide cache flag	C (1)	50	1
PSQPNUMV	Host variable count	B (4,0)	51	2
PSQPSQTL	QDT and access plan length	B (8,0)	53	4
PSQPSQL2	Second QDT and access plan length	B (8,0)	57	4
PSQPSUSES	Usage or Open count	B (8,0)	61	4
PSQPCMTUSE	Compresses since last used	B (4,0)	65	2
PSASTCSID	SQL statement CCSID	B (5,0)	67	4
PSQPSTML	Statement name length	B (4,0)	71	2
PSQPSNAM	Statement name	C (130)	73	130
PSASTMTLEN	SQL statement full length	B (8,0)	203	4
PSASTMTTXT	SQL statement	C (32002)	207	32002

12.2.3.22 QAPYJWSTK

This file contains the call stacks associated with each job/task/thread collected by Job Watcher over time.

The call stacks are sampled at the end of each Job Watcher interval and is done in a non-intrusive manner to not affect the jobs that are running. Because of this at times the data found in the call stacks could be in a state of flux and not always accurate.

This file does not actually contain program names, module names or procedure names. This file contains trace back table address entries for each level of the call stack. The TBT address at each call level can be mapped to file QAPYJWPROC in order to resolve the correct program, module and procedure names associated with each one.

In cases where a job is running a J9 JVM under Pase, then additional Java specific call stack entries may be available in file QAPYJWIJVS if the data has been collected. For examples of how the data in QAPYJWSTK can be merged with the data in QAPYJWIJVS to form a complete call stack, it is best to view the data in the GUI within a J9 job's interval details -> call stack panel in Job Watcher.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18,0)	5	8
NUMFRAMES	Number of stack frames	B (8,0)	13	4
REASON	Stack collection reason	C (1)	17	1
STKERROR	Error indicator	C (1)	18	1
STACK	Call stack	C (32002)	19	32002

12.2.3.23 QAPYJWSTS

This file is used to identify the type of wait each active job/task/thread is in throughout the collection.

Because file QAPYJWTDE only produces wait bucket statistics for jobs/tasks/threads that use CPU in each interval, this additional file is necessary to fill the gaps where jobs are still running but did not use any CPU in a Job Watcher interval.

Note: In some cases Job Watcher will detect jobs that are idle and not using CPU but they never produce a QAPYJWTDE record during the collection. The QAPYJWTDE record identifies the job name and thread ID associated with each task count. In instances where task counts are listed in this file as idle but do not exist in QAPYJWTDE you should be able to utilize Collection Services data collected at the same time and perform a check against file QAPMJOBMI. Field JBTDE (when converted from hex to an integer) will match the task count fields provided in Job Watcher for the same job/task/thread.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
TDESTATUS	TDE status This field identifies the state of the job/task/thread during the end of this interval. The possible values are: 'A' – The job is active and using CPU. This taskcount will also have a record in the QAPYJWTDE file. 'I' – The job is active but did not use any CPU. This taskcount will NOT normally have a record in the QAPYJWTDE file unless the "force 1st interval" has been turned on where interval 1 collects all jobs/tasks in the QAPYJWTDE file regardless of CPU usage.	C (1)	5	1

	'T' – The job is running but is in the process of being terminated.			
STRESERVE	Reserved	C (3)	6	3
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also known as TDE.	B (18,0)	9	8
CURWAITD	Current wait duration total in microseconds This value identifies how long the job/task/thread has been in the current wait identified by the CURWAIT (enum) field. This could range from a few microseconds to days.	B (18,0)	17	8
INTWAITD	Current wait duration in this interval in microseconds	B (18,0)	25	8
CURWAIT	Current or last wait This is the specific type of wait or enum that occurred.	B (4,0)	33	2
CURWAITBKT	Current or last wait bucket number. See file QAPYJWBKT for wait bucket descriptions.	B (4,0)	35	2

12.2.3.24 QAPYJWSYS

This file provides system-wide high-level CPU statistics over time that are independent of the normal Job Watcher collection mechanism. The normal way Job Watcher works can cause some CPU using jobs to be missed from the collection if they were very short lived and did not live at least 1 Job Watcher interval.

The CPU numbers available in this file can be compared to the total CPU numbers for all jobs detected by Job Watcher to get a feel for the amount of short-lived activity on a system that is not being captured. In those instances you may need to decrease the interval size to a smaller number in order to catch more of these short-lived jobs.

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
JSTIME	Time of day	Z	5	26
JSRESERVE1	Reserved	B (4,0)	31	2
JSTOTAL	Total CPU time used in milliseconds	B (18,0)	33	8
JSCFGAV	Configured CPU time available in milliseconds	B (18,0)	41	8
JSUNCAV	Uncapped CPU time available in milliseconds	B (18,0)	49	8
JSSECWRKL	Secondary workload CPU time used in milliseconds	B (18,0)	57	8
JSINTACT	Interactive CPU time used in milliseconds	B (18,0)	65	8
JSINTACTAV	Interactive CPU time available in milliseconds	B (18,0)	73	8

JSINTTHLD	Interactive CPU threshold	B (4,0)	81	2
JSINTLMT	Interactive CPU limit	B (4,0)	83	2
JSCPC	Current processing capacity	B (8,0)	85	4
JSPCNT	Current processor count	B (4,0)	89	2
JSRESERVE2	Reserved	B (4,0)	91	2
JSRESERVE3	Reserved	B (8,0)	93	4
JSRESERVE4	Reserved	B (18,0)	97	8
JSRESERVE5	Reserved	B (18,0)	105	8
SYPTWAIT	CPU thread event wait time in milliseconds	B (18,0)	113	8
SYPTREADY	CPU thread ready wait time in milliseconds	B (18,0)	121	8
SYPTLATEN	CPU thread latency time in milliseconds	B (18,0)	129	8
SYPTACT	CPU thread active time in milliseconds	B (18,0)	137	8
SYPTIDLE	CPU thread idle time in milliseconds	B (18,0)	145	8
SYPTINTR	CPU thread interrupt time in milliseconds	B (18,0)	153	8

12.2.3.25 QAPYJWTDE

This file contains job statistics including wait buckets for all jobs/tasks/threads on the system that used some CPU during each Job Watcher interval. In intervals where jobs did not use CPU, then a record will be added to file QAPYJWSTS instead to keep track of the type of wait each job is in.

Note: Because of the design of Job Watcher (i.e. the non-intrusive snapshot approach) it is possible it will never collect the job/task/thread name for jobs that never use CPU during the collection. However these can be collected if you wish to force a snapshot for all jobs/tasks/thread on the system by using the INCALLFST(*YES) parameter on the ADDJWDFN command when creating your Job Watcher definition.

For descriptions of the 32 wait buckets, see either file QAPYJWBKT or the collection properties of a Job Watcher collection (Wait Buckets tab.)

Field Name	Description	Attribute	Buffer Position	Buffer Length
INTERVAL	Interval number: the nth sample database interval based on the start time of the collection.	B (8,0)	1	4
STARTOD	Time of day at ending snapshot start	Z	5	26
TRESERVE1	Reserved	C (2)	31	2
TASKCOUNT	Task count This number uniquely identifies the job/task/thread running on the system. Also	B (18,0)	33	8

	known as TDE.			
TDEUSECS	Elapsed interval time in microseconds	B (18,0)	41	8
STARTUSECS	Microsecs since IPL at ending snapshot start	B (18,0)	49	8
ENDUSECS	Microsecs since IPL at ending snapshot end	B (18,0)	57	8
THREADID	Thread ID in hex	H (8)	65	8
ITASKCOUNT	Process initial thread task count	B (18,0)	73	8
TDEJOBNAME	Job/task name	C (32)	81	32
THRDSTATUS	Thread status	C (4)	113	4
CURRUP	Current user profile	C (10)	117	10
BIRTHDAY	Job/task birth time of day	Z	127	26
DELTACPU	Thread unscaled CPU charged time (microseconds)	B (18,0)	153	8
EXTENDER	Job name extender	C (2)	161	2
TDETYPE	Job or task flag	C (1)	163	1
TRESERVE2	Reserved	C (1)	164	1
ORIGPRI	Original priority	B (4,0)	165	2
PRIORITY	Current LIC priority	B (4,0)	167	2
THREADPRI	Current XPF priority	B (4,0)	169	2
PRICHG	Priority changed flag	C (1)	171	1
POOLCHG	Pool changed flag	C (1)	172	1
POOL	Pool ID	B (4,0)	173	2
TRESERVE3	Reserved	C (2)	175	2
TOTWRT	Total DASD writes	B (8,0)	177	4
SYNDBRD	Synchronous database reads	B (8,0)	181	4
SYNNDBRD	Synchronous non database reads	B (8,0)	185	4
SYNDBWRT	Synchronous database writes	B (8,0)	189	4
SYNNDBWRT	Synchronous non database writes	B (8,0)	193	4
ASYDBRD	Asynchronous database reads	B (8,0)	197	4
ASYNDBRD	Asynchronous non database reads	B (8,0)	201	4
ASYDBWRT	Asynchronous database writes	B (8,0)	205	4

ASYNDBWRT	Asynchronous non database writes	B (8,0)	209	4
IOPENDING	I/O pending page faults	B (8,0)	213	4
SMSYNCIO	Waits for asynchronous writes	B (8,0)	217	4
FLTS	Page faults resulting in DASD reads	B (8,0)	221	4
ALLOCATED	Allocated DASD pages	B (18,0)	225	8
DEALLOCED	Deallocated DASD pages	B (18,0)	233	8
ALLOCATEDT	Total allocated DASD pgs since thread/task start	B (18,0)	241	8
DEALLOCEDT	Total deallocated DASD pgs since thrd/task start	B (18,0)	249	8
SEIZE	Reserved	B (8,0)	257	4
BINOVER	Binary overflows	B (8,0)	261	4
DECOVER	Decimal overflows	B (8,0)	265	4
FLOATOVER	Float overflows	B (8,0)	269	4
STMFRD	Stream file reads	B (8,0)	273	4
STMFWRT	Stream file writes	B (8,0)	277	4
MUTEX	Mutex time in microseconds	B (18,0)	281	8
ACTWAIT	Active to wait transitions	B (8,0)	289	4
WAITINEL	Wait to ineligible transitions	B (8,0)	293	4
ACTINEL	Active to ineligible transitions	B (8,0)	297	4
TRESERVE4	Reserved	C (4)	301	4
QCOUNT01	Wait bucket 01 count	B (8,0)	305	4
QCOUNT02	Wait bucket 02 count	B (8,0)	309	4
QCOUNT03	Wait bucket 03 count	B (8,0)	313	4
QCOUNT04	Wait bucket 04 count	B (8,0)	317	4
QCOUNT05	Wait bucket 05 count	B (8,0)	321	4
QCOUNT06	Wait bucket 06 count	B (8,0)	325	4
QCOUNT07	Wait bucket 07 count	B (8,0)	329	4
QCOUNT08	Wait bucket 08 count	B (8,0)	333	4
QCOUNT09	Wait bucket 09 count	B (8,0)	337	4
QCOUNT10	Wait bucket 10 count	B (8,0)	341	4

QCOUNT11	Wait bucket 11 count	B (8,0)	345	4
QCOUNT12	Wait bucket 12 count	B (8,0)	349	4
QCOUNT13	Wait bucket 13 count	B (8,0)	353	4
QCOUNT14	Wait bucket 14 count	B (8,0)	357	4
QCOUNT15	Wait bucket 15 count	B (8,0)	361	4
QCOUNT16	Wait bucket 16 count	B (8,0)	365	4
QCOUNT17	Wait bucket 17 count	B (8,0)	369	4
QCOUNT18	Wait bucket 18 count	B (8,0)	373	4
QCOUNT19	Wait bucket 19 count	B (8,0)	377	4
QCOUNT20	Wait bucket 20 count	B (8,0)	381	4
QCOUNT21	Wait bucket 21 count	B (8,0)	385	4
QCOUNT22	Wait bucket 22 count	B (8,0)	389	4
QCOUNT23	Wait bucket 23 count	B (8,0)	393	4
QCOUNT24	Wait bucket 24 count	B (8,0)	397	4
QCOUNT25	Wait bucket 25 count	B (8,0)	401	4
QCOUNT26	Wait bucket 26 count	B (8,0)	405	4
QCOUNT27	Wait bucket 27 count	B (8,0)	409	4
QCOUNT28	Wait bucket 28 count	B (8,0)	413	4
QCOUNT29	Wait bucket 29 count	B (8,0)	417	4
QCOUNT30	Wait bucket 30 count	B (8,0)	421	4
QCOUNT31	Wait bucket 31 count	B (8,0)	425	4
QCOUNT32	Wait bucket 32 count	B (8,0)	429	4
QTIME01	Wait bucket 01 time in microseconds	B (18,0)	433	8
QTIME02	Wait bucket 02 time in microseconds	B (18,0)	441	8
QTIME03	Wait bucket 03 time in microseconds	B (18,0)	449	8
QTIME04	Wait bucket 04 time in microseconds	B (18,0)	457	8
QTIME05	Wait bucket 05 time in microseconds	B (18,0)	465	8
QTIME06	Wait bucket 06 time in microseconds	B (18,0)	473	8
QTIME07	Wait bucket 07 time in microseconds	B (18,0)	481	8
QTIME08	Wait bucket 08 time in microseconds	B (18,0)	489	8

QTIME09	Wait bucket 09 time in microseconds	B (18,0)	497	8
QTIME10	Wait bucket 10 time in microseconds	B (18,0)	505	8
QTIME11	Wait bucket 11 time in microseconds	B (18,0)	513	8
QTIME12	Wait bucket 12 time in microseconds	B (18,0)	521	8
QTIME13	Wait bucket 13 time in microseconds	B (18,0)	529	8
QTIME14	Wait bucket 14 time in microseconds	B (18,0)	537	8
QTIME15	Wait bucket 15 time in microseconds	B (18,0)	545	8
QTIME16	Wait bucket 16 time in microseconds	B (18,0)	553	8
QTIME17	Wait bucket 17 time in microseconds	B (18,0)	561	8
QTIME18	Wait bucket 18 time in microseconds	B (18,0)	569	8
QTIME19	Wait bucket 19 time in microseconds	B (18,0)	577	8
QTIME20	Wait bucket 20 time in microseconds	B (18,0)	585	8
QTIME21	Wait bucket 21 time in microseconds	B (18,0)	593	8
QTIME22	Wait bucket 22 time in microseconds	B (18,0)	601	8
QTIME23	Wait bucket 23 time in microseconds	B (18,0)	609	8
QTIME24	Wait bucket 24 time in microseconds	B (18,0)	617	8
QTIME25	Wait bucket 25 time in microseconds	B (18,0)	625	8
QTIME26	Wait bucket 26 time in microseconds	B (18,0)	633	8
QTIME27	Wait bucket 27 time in microseconds	B (18,0)	641	8
QTIME28	Wait bucket 28 time in microseconds	B (18,0)	649	8
QTIME29	Wait bucket 29 time in microseconds	B (18,0)	657	8
QTIME30	Wait bucket 30 time in microseconds	B (18,0)	665	8
QTIME31	Wait bucket 31 time in microseconds	B (18,0)	673	8
QTIME32	Wait bucket 32 time in microseconds	B (18,0)	681	8
TRESERVE5	Reserved	C (6)	689	6
CURRSTATE	Current or last state	C (4)	695	4
BLOCKBCKT	Current or last blocking bucket	B (4,0)	699	2
LICWO	Current or last LIC wait object	C (4)	701	4
LICWOHNDL	Current or last LIC wait object handle	H (8)	705	8
WOBASSEG	Wait object base segment address in hex	H (8)	713	8

WOSEGTYP	Wait object segment type in hex	H (4)	721	4
WOOBJTYP	Wait object object type in hex	H (4)	725	4
WOOBJNAM	Wait object name	C (30)	729	30
WOOBJTYPD	Wait object obj type description	C (35)	759	35
WOSEGTYPD	Wait object segment type description	C (35)	794	35
TRESERVE6	Reserved	C (3)	829	3
HTYPE	Holding thread/task type	C (1)	832	1
HTASKCNT	Holding thread/task task count	B (18,0)	833	8
HTASKNAME	Holding thread/task name	C (32)	841	32
CURRWTDUR	Total time in current wait in microsecs	B (18,0)	873	8
BLOCKENUM	Current or last blocking enum	B (4,0)	881	2
TRESERVE7	Reserved	C (2)	883	2
RECCNFLCT	Ordinal record number if db record lock conflict	B (8,0)	885	4
DFTSOCKD	Default socket descriptor	B (8,0)	889	4
DFTSOCKTOD	Default socket time of day	Z	893	26
DFTSOCKCLV	Default socket cache level	B (4,0)	919	2
DFTSOCKH	Default socket handle	H (8)	921	8
LISSOCKD	Listen socket descriptor	B (8,0)	929	4
LISSOCKTOD	Listen socket time of day	Z	933	26
LISSOCKCLV	Listen socket cache level	B (4,0)	959	2
LISSOCKH	Listen socket handle	H (8)	961	8
FRMESTOL	New mainstore frames stolen	B (8,0)	969	4
SREMOVE	Successful removes	B (8,0)	973	4
PSAINUSE	PSA entries inuse	B (8,0)	977	4
SQLINTHRD	SQL statement in progress	C (1)	981	1
TDPASE	PASE run time	C (1)	982	1
TDJTHDT	JVM thread type	BINCHAR 1	983	1
TRESERVE8	Reserved	C (1)	984	1
TDKTID	Kernel thread ID	B (18,0)	985	8

TRESERVE9	Reserved	B (8,0)	993	4
TRESERVE10	Workload group ID	B (8,0)	997	4
TRESERVE11	Workload group latency in microseconds	B (18,0)	1001	8
TINDCPU	Thread unscaled CPU used in microseconds	B (18,0)	1009	8
TSINDCPU	Thread scaled CPU used in microseconds	B (18,0)	1017	8
TCPUWC	Processor elapsed time in microseconds	B (18,0)	1025	8
TVPDLY	Virtual CPU delay in microseconds	B (18,0)	1033	8
>>> TPGEZSTL	Pages marked easy to steal	B (9,0)	1041	4 <<<
>>> TRESERVE12	Reserved	B (9,0)	1045	4 <<<
>>> SQLCLKTM	SQL clock time in microseconds	B (18,0)	1049	8 <<<
>>> SQLCPU	Thread unscaled SQL CPU used in microseconds	B (18,0)	1057	8 <<<
>>> SQLSCPU	Thread scaled SQL CPU used in microseconds	B (18,0)	1065	8 <<<
>>> SQLSDBRD	SQL synchronous database reads	B (18,0)	1073	8 <<<
>>> SQLSNDBRD	SQL synchronous non database reads	B (18,0)	1081	8 <<<
>>> SQLSDBWRT	SQL synchronous database writes	B (18,0)	1089	8 <<<
>>> SQLSNDBWRT	SQL synchronous non database writes	B (18,0)	1097	8 <<<
>>> SQLADBRD	SQL asynchronous database reads	B (18,0)	1105	8 <<<
>>> SQLANDBRD	SQL asynchronous non database reads	B (18,0)	1113	8 <<<
>>> SQLADBWRT	SQL asynchronous database writes	B (18,0)	1121	8 <<<
>>> SQLANDBWRT	SQL asynchronous non database writes	B (18,0)	1129	8 <<<
>>> TMPALLOC	Total allocated DASD pages since task/thd start	B (18,0)	1137	8 <<<
>>> TMPDEALLOC	Total deallocated DASD pages since task/thd start	B (18,0)	1145	8 <<<
>>> SQLHLSTMT	High level SQL statements	B (18,0)	1153	8 <<<
>>> THDINSCHRG	Thread instructions charged	B (18,0)	1161	8 <<<
>>> THDINSUSED	Thread instructions used	B (18,0)	1169	8 <<<
>>> THDTIMBASE	Thread time base used (ms)	B (18,0)	1177	8 <<<
>>> WOOBJLIB	Wait object library	C (10)	1185	10 <<<
>>> TRESERVE13	Reserved	B (18,0)	1195	8 <<<

>>> TRESERVE14	Reserved	B (18,0)	1203	8 <<<
>>> TRESERVE15	Reserved	B (18,0)	1211	8 <<<
>>> TRESERVE16	Reserved	B (18,0)	1219	8 <<<

12.2.4 Job Watcher Definitions

Job Watcher definitions are stored in file QUSRSYS/QAPYJWDFN. This file will not exist on the system until command QSYS/ADDJWDFN is ran on a system for the 1st time.

12.2.4.1 Possible problem with definition command strings blank

The QAPYJWDFN and QAPYDWDFN files in QUSRSYS keep track of each definition and the command string that was used to create each one. In rare cases it is possible for the command strings listed in these tables to be blank. This typically could happen if the exit programs used to add the command strings to the files are no longer registered properly.

To verify that the exit program is properly registered, do the following:

Run WRKREGINF from the command line. You will see something similar to this screen

```
                     Work with Registration Information

Type options, press Enter.
  5=Display exit point   8=Work with exit programs

                          Exit
     Exit                 Point
Opt  Point                Format    Registered  Text
     QIBM_A1A_RETR_INF    RTVI0100     *YES     BRM Services/400 object retri
     QIBM_A1A_TAPE_INF    MEDI0100     *YES     BRM Services/400 media inform
     QIBM_A1A_TAPE_MOVE   MEDM0100     *YES     BRM Services/400 media moveme
     QIBM_QCA_CHG_COMMAND CHGC0100     *YES     Change command exit programs
     QIBM_QCA_RTV_COMMAND RTVC0100     *YES     Retrieve command exit program
     QIBM_QCQ_AGENT       ENDE0100     *YES
     QIBM_QCQ_AGENT       STRE0100     *YES
     QIBM_QCST_ADMDMN     ADMN0100     *YES     Cluster administrative domain
     QIBM_QCST_ADMDMN     ADMN0200     *YES     Cluster administrative domain
     QIBM_QCST_CLU        QCLU0100     *YES     Cluster node support
     QIBM_QCST_CRG        QCLU0100     *YES     Cluster resource group suppor
                                                                      More...
Command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Type an 8 next to `QIBM_QCA_RTV_COMMAND` and hit enter. You should see something like this.

```
                           Work with Exit Programs

Exit point:   QIBM_QCA_RTV_COMMAND     Format:   RTVC0100

Type options, press Enter.
  1=Add   4=Remove   5=Display   10=Replace

             Exit
           Program     Exit
Opt         Number     Program        Library
```

```
                    1      QPYDJCSTR      QSYS
                    2      QPYDJCSTR      QSYS
                    3      QPYDJCSTR      QSYS
                    4      QPYDJCSTR      QSYS
                    5      QPYDJCSTR      QSYS
                    6      QPYDJCSTR      QSYS
                    7      QPYDJCSTR      QSYS
                    8      QPYDJCSTR      QSYS

Command
 ===>
 F3=Exit    F4=Prompt    F5=Refresh    F9=Retrieve    F12=Cancel
```

There should be at least four entries for program QPYDJCSTR, one for each command ADDDWDFN, ADDJWDFN, STRDW, STRJW. There may be more than four. That is due to a bug that adds new entries at every install, even if they are already there (ie, slip install). While not "optimal," this doesn't break anything to have more than four entries and is fixed at 7.2.

If you type 5 (display) next to each one, you can see which command it is for:

```
                            Display Exit Program
                                                              System:   ISZ1LP13
 Exit point  . . . . . . . . . . . . . . . :   QIBM_QCA_RTV_COMMAND
 Exit point format . . . . . . . . . . . . :   RTVC0100

 Exit program number . . . . . . . . . . . :   1
 Exit program  . . . . . . . . . . . . . . :   QPYDJCSTR
   Library . . . . . . . . . . . . . . . . :     QSYS
 Text description  . . . . . . . . . . . . :   *BLANK

 Exit program data CCSID . . . . . . . . . :   37
 Exit program data length  . . . . . . . . :   20
 Threadsafe  . . . . . . . . . . . . . . . :   *UNKNOWN
 Multithreaded job action  . . . . . . . . :   *MSG
 Exit program data . . . . . . . . . . . . :
 ADDDWDFN  QSYS



                                                                        Bottom
 Press Enter to continue.

 F3=Exit    F10=Display data    F12=Cancel
```

If the exit programs are not registered then the Work with Exit Programs screen will show nothing. If that is the case, you can add them back in by running this command from the green screen command line: CALL QPYIEREG

12.3 Collection Services (Library QIDRWCH and QSYS)

This section describes green screen commands and functionality related to Collection Services and iDoctor's Collection Services Investigator.

12.3.1 IBM i Collection Services Commands

The following commands are included with Collection Services in library QSYS:

12.3.1.1 CFGPFRCOL

Configure the parameters used for capturing Collection Services data.

12.3.1.2 CRTPFRDTA

Creates a set of QAPM* database files from a Collection Services *MGTCOL object.

12.3.1.3 CPYPFRCOL

Copy a collection to another location.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cpypfrcol.htm

12.3.1.4 CVTPFRCOL

This command will convert a collection created at a previous release to the current release.
At 6.1 this command only supports Collection Services collections. At 7.1 and higher it also supports Disk Watcher, Job Watcher and PEX.

Note: Using this command is not recommended for use with iDoctor. iDoctor will handle your data regardless of the release it was created on (5.4 and up.)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cvtpfrcol.htm

12.3.1.5 DLTPFRCOL

Deletes one or more collections (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/dltpfrcol.htm

12.3.1.6 SAVPFRCOL

Saves one or more collections to a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/savpfrcol.htm

12.3.1.7 RSTPFRCOL

Restore one or more collections from a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rstpfrcol.htm

12.3.2 Collection Services Investigator Commands

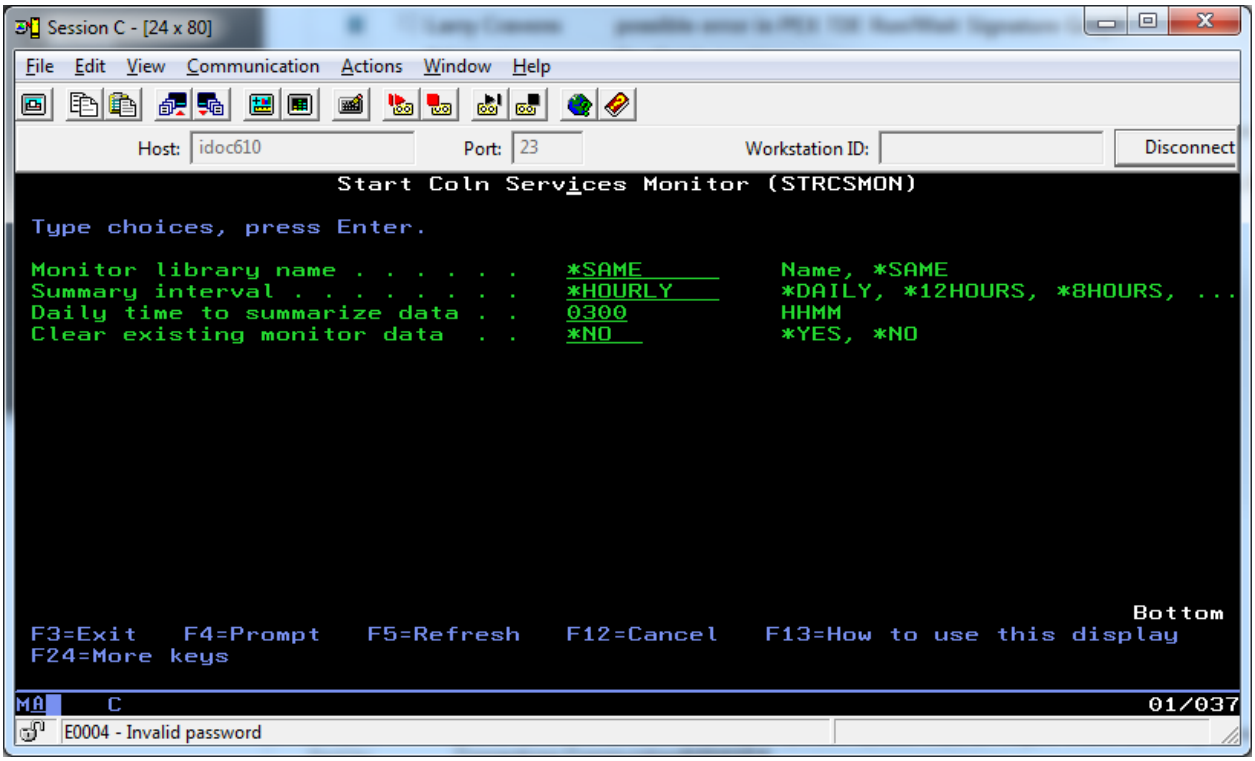
The following commands are included with Collection Services Investigator in library QIDRWCH:

12.3.2.1 ENDCSMON

This command is used to end the job running the Collection Services Monitor on the system (if it is currently active).

A Collection Services Monitor summarizes historical data for the purpose of graphing that data with iDoctor's Collection Services Investigator.

12.3.2.2 STRCSMON



This command is used to start or restart a Collection Services monitor in order to summarize and consolidate Collection Services data for historical analysis purposes.

Only 1 Collection Services monitor can be running on a system at a time.

Note: This command should be submitted to batch using the SBMJOB command and not be ran interactively.

12.3.3 IBM i Collection Services Files

For more information on this topic please visit the IBM Knowledge Center link below:

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzahx/rzahxperfdatafiles1.htm

.

12.4 Disk Watcher (Library QIDRWCH and QSYS)

This section describes green screen commands and functionality related to Disk Watcher.

12.4.1 IBM i Disk Watcher Commands

At 6.1 and higher the Disk Watcher commands were added to IBM i. The following commands are part of the Disk Watcher support included in library QSYS:

12.4.1.1 ADDDWDFN

Adds a Disk Watcher definition to the system.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/adddwdfn.htm

12.4.1.2 STRDW

Starts a Disk Watcher collection
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/strdw.htm

12.4.1.3 RMVDWDFN

Deletes a Disk Watcher definition.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rmvdwdfn.htm

12.4.1.4 ENDDW

Ends a Disk Watcher collection.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/enddw.htm

12.4.1.5 CPYPFRCOL

Copy a collection to another location.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cpypfrcol.htm

12.4.1.6 CVTPFRCOL

This command will convert a collection created at a previous release to the current release.
At 6.1 this command only supports Collection Services collections. At 7.1 and higher it also supports Disk Watcher, Job Watcher and PEX.

Note: Using this command is not recommended for use with iDoctor. iDoctor will handle your data regardless of the release it was created on (5.4 and up.)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cvtpfrcol.htm

12.4.1.7 DLTPFRCOL

Deletes one or more collections (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/dltpfrcol.htm

12.4.1.8 SAVPFRCOL

Saves one or more collections to a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/savpfrcol.htm

12.4.1.9 RSTPFRCOL

Restore one or more collections from a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rstpfrcol.htm

12.4.2 iDoctor Disk Watcher Commands

The following commands are included with Disk Watcher in library QIDRWCH:

12.4.2.1 CPYDWCOL

This command is used to copy a Disk Watcher collection. This action will copy every member matching the collection name from the "from" library to the "to" library. If desired the collection can be renamed by setting the TOCOL parameter with a new name.

Note: CPYPFRCOL can also perform this action, but this command is still provided to avoid possible problems where CPYPFRCOL will not allow a collection to be copied based on VRM differences of the database files.

This command will NOT copy the iDoctor created SQL Tables produced by iDoctor analyses.

12.4.2.2 DLTDWCOL

This command is used to delete a Disk Watcher collection from a user's library on a system. This action will remove the member matching the collection name from every QAPYDW* file found in the library specified.

This command also removes all iDoctor created SQL Tables.

12.4.2.3 DLTDWMON

This command is used to delete a Disk Watcher monitor and all collections it contains on the current system. If the monitor is still running, the monitor will first be ended using the ENDDWMON command.

The record in file QUSRSYS/QAIDRDWM1 that identifies the existence of this monitor will also be removed by this command.

12.4.2.4 ENDDWMON

This command is used to end the job running a Disk Watcher monitor and any jobs currently running collections within the monitor.

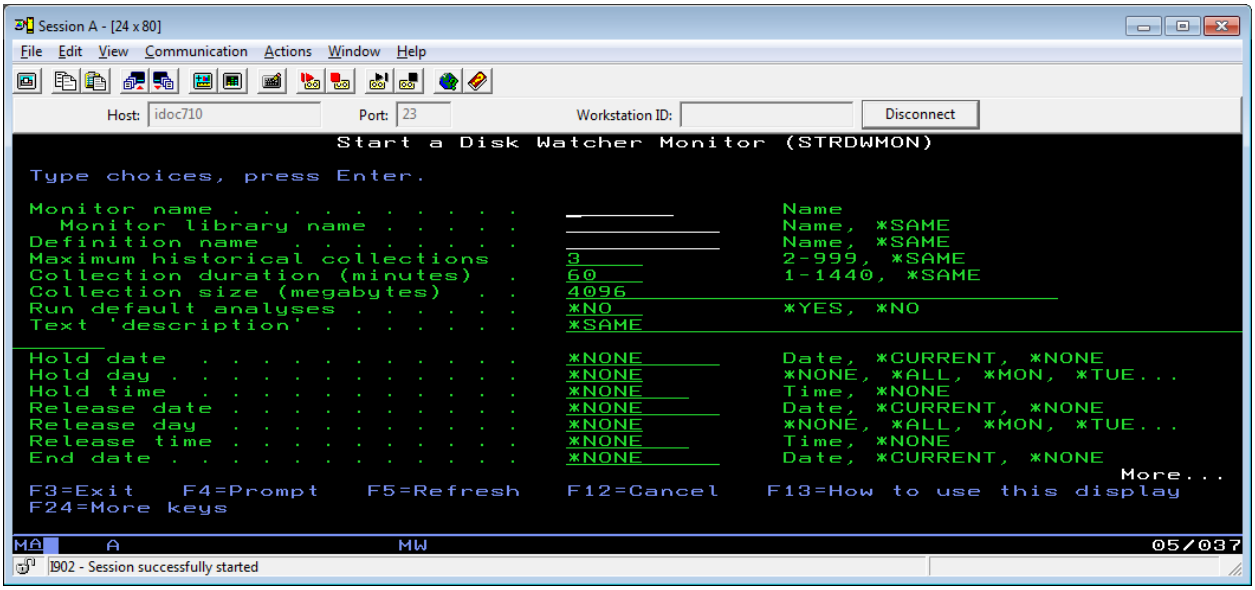
12.4.2.5 HLDDWMON

This command is used to hold a Disk Watcher monitor. A held monitor will not delete any old collections or create any new collections until it is released using the RLSDWMON command.

12.4.2.6 RLSDWMON

This command is used to release a monitor that is currently in a held state. Once released the monitor will continue to create new collections and delete old collections normally.

12.4.2.7 STRDWMON



This command is used to start or restart a Disk Watcher monitor. Use DLTDWMON to delete an existing monitor.

Note: This command should be submitted to batch using the SBMJOB command and not be ran interactively.

A monitor is a set of collections that continuously collect data over a system overwriting the oldest collection when it creates a new collection.

Monitors are built from a Disk Watcher definition which are stored in file QUSRSYS/QAPYDWDFN. A definition is a ADDDWDFN command string that indicates the parameters the monitor should use in its collections. Definitions can be created using the iDoctor client or by using the ADDDWDFN command.

The maximum historical collections parameter (COLNS) determines how many collections should be saved at one time.

A record in file QUSRSYS/QAIDRDWM2 that identifies the existence and status of the monitor is created and updated by this command.

12.4.3 IBM i Disk Watcher Files

For more information on this topic please visit the IBM Knowledge Center link below:

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzahx/rzahxdwdatafiles.htm

12.5 Plan Cache Analyzer (Library QPLANCACHE)

Plan Cache Analyzer contains several programs and commands which are used to analyze the system's Plan Cache. When library QIDRGUI is installed, iDoctor also restores library QPLANCACHE from a save file included with the install image.

12.5.1 OS Support for the SQL Plan Cache

Stored procedures are available to create SQL Plan Cache Snapshots or work with the output. Some of these are used under the covers by Plan Cache Analyzer.

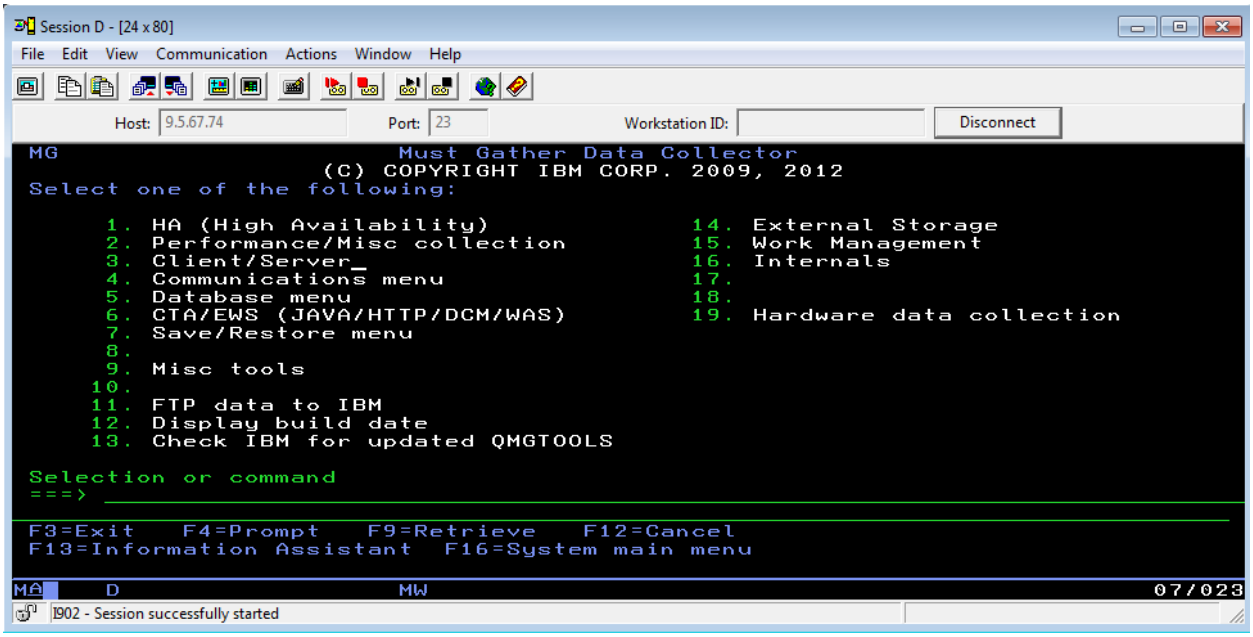
For more information on using these yourself, visit the section called Accessing the SQL plan cache with SQL stored procedures in the IBM i 7.1 Information Center.

12.5.2 Plan Cache Analyzer Commands

The library QPLANCACHE contains commands that are currently intended for IBM service use only.

12.6 Must Gather Tools (QMGTOOLS library)

Must Gather Tools (QMGTOOLS library) is a set of tools to assist individual IBM support teams collect data for issues concerning their products. For example, the PowerHA menu contains a set of programs that will collect specific items (job logs, SST macro output, VLOGs, and so on) to assist the technical representative in debugging a problem. A user will install this tool and use the menu shown below to collect data for the specific problem:



Must Gather Tools main menu

For more information on using Must Gather Tools please visit:
http://www-01.ibm.com/support/docview.wss?uid=nas8N1010441

12.7 PEX and PEX Analyzer (libraries QSYS and QIDRPA)

This section covers the server side for Performance Explorer (PEX) and the iDoctor component that analyses PEX data, PEX Analyzer.

12.7.1 IBM i PEX Commands

Here is a list of the PEX commands that are included in library QSYS:

12.7.1.1 ADDPEXDFN

Adds a PEX Definition to the system.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/addpexdfn.htm

12.7.1.2 ADDPEXFTR

Adds a PEX Filter to the system which are used to reduce the amount of data collected.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/addpexftr.htm

12.7.1.3 STRPEX

Starts a PEX session using a definition.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/strpex.htm

12.7.1.4 RMVPEXDFN

Deletes a PEX definition
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rmvpexdfn.htm

12.7.1.5 RMVPEXFTR

Deletes a PEX filter from the system.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rmvpexftr.htm

12.7.1.6 ENDPEX

Ends a PEX session (or view active ones)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/endpex.htm

12.7.1.7 CRTPEXDTA

Creates a PEX collection from a PEX *MGTCOL object.

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/crtpexdta.htm

12.7.1.8 PRTPEXRPT

The Print PEX Report (PRTPEXRPT) command prints a formatted listing of the data that was collected by PEX and saved across a set of physical files in a particular library.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/prtpexrpt.htm

12.7.1.9 CPYPFRCOL

Copy a collection to another location.
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cpypfrcol.htm

12.7.1.10 CVTPFRCOL

This command will convert a collection created at a previous release to the current release.
At 6.1 this command only supports Collection Services collections. At 7.1 and higher it also supports Disk Watcher, Job Watcher and PEX.

Note: Using this command is not recommended for use with iDoctor. iDoctor will handle your data regardless of the release it was created on (5.4 and up.)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/cvtpfrcol.htm

12.7.1.11 DLTPFRCOL

Deletes one or more collections (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/dltpfrcol.htm

12.7.1.12 SAVPFRCOL

Saves one or more collections to a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/savpfrcol.htm

12.7.1.13 RSTPFRCOL

Restore one or more collections from a SAVF (either PEX, CS, DW, JW, etc)
http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/cl/rstpfrcol.htm

---

12.7.2 QIDRWCH library PEX Analyzer Commands

The following commands are included with PEX Analyzer in library **QIDRWCH**:

12.7.2.1 DLTPAMON

This command is used to delete a PEX Analyzer monitor and all collections it contains on the current system. If the monitor is still running, the monitor will first be ended using the ENDPAMON command.

The record in file QUSRSYS/QAIDRPAM1 that identifies the existence of this monitor will also be removed by this command.

12.7.2.2 ENDPAMON

This command is used to end the job running a PEX Analyzer monitor and any jobs currently running collections within the monitor.

12.7.2.3 HLDPAMON

This command is used to hold a PEX Analyzer monitor. A held monitor will not delete any old collections or create any new collections until it is released using the RLSPAMON command.

12.7.2.4 RLSPAMON

This command is used to release a monitor that is currently in a held state. Once released the monitor will continue to create new collections and delete old collections normally.

12.7.2.5 STRPAMON

This command is used to start or restart a PEX monitor. Use DLTPAMON to delete an existing monitor.

Note: This command should be submitted to batch using the SBMJOB command and not be ran interactively.

A monitor is a set of collections that continuously collect data over a system overwriting the oldest collection when it creates a new collection.

Monitors are built from a PEX definition which are stored in file QUSRSYS/QAPEXDFN. A definition is a ADDPEXDFN command string that indicates the parameters the monitor should use in its collections. Definitions can be created using the iDoctor client or by using the ADDPEXDFN command.

The maximum historical collections parameter (COLNS) determines how many collections should be saved at one time.

A record in file QUSRSYS/QAIDRPAM2 that identifies the existence and status of the monitor is created and updated by this command.

12.7.3 QIDRPA library PEX Analyzer commands

The following commands are included with PEX Analyzer in library **QIDRPA**:

12.7.3.1 CPYPACOL

This command is used to copy a PEX collection. This action will copy every member matching the collection name from the "from" library to the "to" library. If desired the collection can be renamed by setting the TOCOL parameter with a new name.

Note: CPYPFRCOL can also perform this action, but this command is still provided to avoid possible problems where CPYPFRCOL will not allow a collection to be copied based on VRM differences of the database files.

This command will NOT copy the iDoctor created SQL Tables produced by iDoctor analyses.

12.7.3.2 DLTPACOL

This command is used to delete a PEX collection from a user's library on a system. This action will remove the member matching the collection name from every QAYPE* file found in the library specified.

This command also removes all iDoctor created SQL Tables.

12.7.3.3 ENDPACOL

This command is used to end a PEX collection that was started with the STRPACOL command, prior to the expiration of the initially provided time value.

This command has no effect once the time value provided on the STRPACOL has expired.

This command can also be used to override the option on the STRPACOL command for how the data is stored.

12.7.3.4 RSMPACOL

This command is used to resume a PEX collection started using STRPACOL with a STNDBY parameter value of 'Y'.

12.7.3.5 STRPACOL

This command is designed to simplify the collection of PEX data and primarily performs the functions of ADDPEXDFN, STRPEX and ENDPEX commands.

The command can be run in either interactive mode or in batch.

Execution of this command causes a job, QIDRPACOL, to be submitted to the jobq and library specified on this command. This job will be present for the entire life of the collection and any time required for the dumping of the collected data. Ending this job before it's normal completion will most likely invalidate any data that has been collected.

In addition to the functions listed above, the QIDRPACOL job also collects WRKSYSSTS and WRKDSKSTS information at predetermined intervals. This data is then copied into the following files in the collection library:

SMTRSTS, (WRKSYSSTS output),

SMTRDTS, (WRKDSKSTS output).

There will be a member placed in each file having the same name as the data collection.

12.7.4 IBM i PEX Files

This section describes the PEX (Performance Explorer) database files that come with IBM i and is applicable to release 7.2 only. Fields that are new in release 7.2 will have >>> <<< identifiers around them.

These files are not currently documented by IBM in the IBM Knowledge Center so are provided here for your convenience.

12.7.4.1 Terminology

A few comments about terminology in this section:

Term	Description
TDE	The TDE (task dispatching element) uniquely identifies a job/task/thread running on a system. Also known as task count. In Job Watcher this is reported as a long (8 byte) integer. In Collection Services and PEX this is reported in HEX and must be converted to decimal in order to compare values with Job Watcher.
Task count	The task count uniquely identifies a job/task/thread running on a system. Also known as TDE. In Job Watcher this is reported as a long (8 byte) integer. In Collection Services and PEX this is reported in HEX and must be converted to decimal in order to compare values with Job Watcher.

12.7.4.2 >>> QAYPEACT <<<

This file produces additional information about PEX program activations. This is a new file at 7.2 and is only produced if the base event *ACTDTA is included on the PEX definition.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QPGMADDR	Program address Address of the program being activated	H (8)	9	8
QRECTYPE	Record type	B (4,0)	17	2
QINVOKER	Activation invoker	B (4,0)	19	2
QINVTYPE	Invocation type	B (4,0)	21	2
QFLAGS	Program object flags	B (4,0)	23	2
QACTMODE	Activation mode	B (4,0)	25	2
QRQACTMODE	Required activation mode	B (4,0)	27	2
QPGMTYPE	Program type	B (4,0)	29	2
QAGSTGMDL	Group storage model	B (4,0)	31	2
QAGTYPE	Group type	B (4,0)	33	2
QWTSINITS	S-inits in cycle (deadlk)	B (9,0)	35	4
QGENCODE	Generation code	B (9,0)	39	4
QACTMARK	Activation mark	H (8)	43	8

QACTMARK2	Activation mark 2	H (8)	51	8
QGRPMARK	Activation group mark	H (8)	59	8
QTHREADID	Thread ID for s-init wait	H (8)	67	8
QNAME	Activation group name	C (12)	75	12
QRESERVED1	Reserved	B (4,0)	87	2
QRESERVED2	Reserved	B (4,0)	89	2
QRESERVED3	Reserved	B (4,0)	91	2
QRESERVED4	Reserved	B (4,0)	93	2
QRESERVED5	Reserved	B (9,0)	95	4
QRESERVED6	Reserved	B (9,0)	99	4
QRESERVED7	Reserved	B (18,0)	103	8
QRESERVED8	Reserved	B (18,0)	111	8

12.7.4.3 QAYPEAFN

This file contains resource affinity trace data. It will only be created if one or more resources affinity events (RSCAFNEVT) are included on the ADDPEXDFN command.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18, 0)	1	8
QPERIOD	Period	B (9,0)	9	4
QITVL	Interval	B (9,0)	13	4
QGRPID	Group identifier	B (9,0)	17	4
QTGTWTN	Target weight modification number	B (9,0)	21	4
QNEWRSC	New resource	B (4,0)	25	2
QOLDRSC	Old resource	B (4,0)	27	2
QGRPRSC	Group preferred resource	B (4,0)	29	2
QNEWSTT	New state	B (4,0)	31	2
QOLDSTT	Old state	B (4,0)	33	2
QNEWWT	New weight	B (4,0)	35	2
QOLDWT	Old weight	B (4,0)	37	2
QCURWT	Current weight	B (4,0)	39	2

QINITWT	Initiation weight	B (4,0)	41	2
QTRANWT	Transition weight	B (4,0)	43	2
QWTDFTH	Weight differential threshold	B (4,0)	45	2
QLODIDX	Load index	B (4,0)	47	2
QNEWPRC	New processors	B (4,0)	49	2
QOLDPRC	Old processors	B (4,0)	51	2
QNEWMEM	New memory pages	B (9,0)	53	4
QOLDMEM	Old memory pages	B (9,0)	57	4
QNBRTSK	Number of tasks	B (9,0)	61	4
QTASKCT	Task count	H (8)	65	8
QCYCMOV	Cycles moved	B (18, 0)	73	8
QCYCRCD	Cycles recorded	B (18, 0)	81	8
QCYCBY	Cumulative cycles run by this resource	B (18, 0)	89	8
QCYCFOR	Cumulative cycles run for this resource	B (18, 0)	97	8
QCYCBYFOR	Cumulative cycles run for/by this resource	B (18, 0)	105	8
QTGTCYC	Target cycles	B (18, 0)	113	8
QTHLDCYC	Threshold cycles	B (18, 0)	121	8
QPGSON	Cumulative pages allocated on this resource	B (18, 0)	129	8
QPGSFOR	Cumulative pages allocated for this resource	B (18, 0)	137	8
QPGSBYFOR	Cumulative pages allocated for/on this resource	B (18, 0)	145	8
QFLAGS	Controller flags	H (8)	153	8

12.7.4.4 QAYPEASM

This file contains auxiliary storage management trace event data. It will only be produced if one or more storage events (STGEVT parameter) are included on the ADDPEXDFN command.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QAMSPA	MI suspend point address	H (8)	9	8
QAMSPK	Key for MI suspend point	H (8)	17	8
QAMNIA	Next instruction address	H (8)	25	8

QAMNIK	Key for next instruction address	H (8)	33	8
QAMSAD	Starting address of segment	H (8)	41	8
QAMSOF	Segment start address offset	B (9,0)	49	4
QAMPGS	Pages in the segment	B (18,0)	53	8
QAMSTS	ASM status	B (4,0)	61	2
QAMNAC	ASM pages outside access group	B (4,0)	63	2
QAMASP	ASP number	B (9,0)	65	4
QAMSTYP	Segment type	B (4,0)	69	2
QOBJKEY	Object key	H (8)	71	8

12.7.4.5 QAYPEASPI

This file provides the ASP resource name for any independent ASPs found in the collection related to disk events in file QAYPEDASD.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QASPNO	Independent ASP number	B (9,0)	1	4
QASPNM	ASP resource name	C (10)	5	10

12.7.4.6 QAYPEBASE

This file is included in the collection whenever base events are captured in the PEX trace collection. The possible base events included PMCO, task switch, activation groups, CPU switch and more. See the BASEVT parameter of ADDPEXDFN for a complete list.

Note: Not all base events will add records to this file such as the taskswitch events. Taskswitch event data will appear in the QAYPETSKSW file.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QBSIAD	Instruction address	H (8)	9	8
QBSTBT	Traceback table address	H (8)	17	8
QBSTAD	Task address	H (8)	25	8
QBSMIN	Minor system reference code	B (4,0)	33	2
QBSIPL	IPL phase text	C (24)	35	24
QBSR03	General purpose register 3	H (8)	59	8

QBSR04	General purpose register 4	H (8)	67	8
QBSR05	General purpose register 5	H (8)	75	8
QBSR06	General purpose register 6	H (8)	83	8
QBSR07	General purpose register 7	H (8)	91	8
QBSR08	General purpose register 8	H (8)	99	8
QBSR09	General purpose register 9	H (8)	107	8
QBSR10	General purpose register 10	H (8)	115	8
QBSR11	General purpose register 11	H (8)	123	8
QBSR12	General purpose register 12	H (8)	131	8
QBSEXI	Exception Id or act group type	B (9,0)	139	4
QBSIXI	IMPI exception ID or act group stg model	B (9,0)	143	4
QBSETY	Exception type or act group state	B (4,0)	147	2
QBSFAD	Faulting address	H (8)	149	8
QBSEPA	Excepting program address	H (8)	157	8
QBSPIO	Offset of pgm instruction	B (9,0)	165	4
QBSIIN	Interrupt information	H (8)	169	8
QBSPAD	Program address	H (8)	177	8
QBSAGM	Activation group mark	H (8)	185	8
QBSTHI	Thread identifier or act grp creator pgm	H (8)	193	8
QBSITF	Initial thread flag Y/N	C (1)	201	1
QBSLNK	Link register or callers address	H (8)	202	8
QBSLTB	Link register traceback table	H (8)	210	8
QBSDAR	Sampled data address register	H (8)	218	8
QBSPLN	Statement number or PASE line number	B (9,0)	226	4
QBSPCL	PASE caller line number	B (9,0)	230	4
QBSPTI	Procedure type: 0 = ILE 1 = PASE	C (1)	234	1
QBSH01	PMC 1: snapshot of task PMC 1 (default is cycles)	B (18,0)	235	8
QBSH02	PMC 2: snapshot of task PMC 2	B (18,0)	243	8
QBSH03	PMC 3: snapshot of task PMC 3	B (18,0)	251	8
QBSH04	PMC 4: snapshot of PMC 4 (default = instructions)	B (18,0)	259	8

QBSH05	PMC 5: snapshot of PMC 5 (default = instructions)	B (18,0)	267	8
QBSH06	PMC 6: snapshot of PMC 6 (default = instructions)	B (18,0)	275	8
QBSH07	PMC 7: snapshot of PMC 7 (default = instructions)	B (18,0)	283	8
QBSH08	PMC 8: snapshot of PMC 8 (default = instructions)	B (18,0)	291	8
QBSCORR	SDAR/SIAR Correlated (Y/N)	C (1)	299	1
QBSHYPA	Hypervisor active flag (Y/N)	C (1)	300	1
QBSPRBST	Problem state flag (Y/N)	C (1)	301	1
QBSSLOT	Sampled slot number	B (4,0)	302	2
QBSTAGSA	Tags active flag (Y/N)	C (1)	304	1
QBSSDARRA	SDAR Real Address	H (8)	305	8
QBSPRFADR	Current user profile address	H (8)	313	8
QBSMISC	Miscellaneous	H (8)	321	8
QBSEVVEC	Event vector	B (18,0)	329	8
QOBJKEY	Object key	H (8)	337	8
QBSPSMMAT	Processor save mode or mobility action type	B (4,0)	345	2
QBSPPILSN	LPAR physical processor index or LPAR suspend number	B (4,0)	347	2
QBSHPIMRC	Dynamic HW processor id or LPAR migrate rtn code	B (9,0)	349	4

12.7.4.7 QAYPECFGI

This file provides basic configuration information about the definition used on the PEX collection.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QCFNM	Definition name	C (10)	1	10
QCFDSC	Definition description	C (72)	11	72
QCFMOD	Definition mode	C (10)	83	10
QCFUSR	User id of definition creator	C (8)	93	8
QCFSYS	System the definition was created	C (8)	101	8
QCFFNM	Filter name	C (10)	109	10
QCFFDS	Filter description	C (72)	119	72
QCFFUS	User id of filter creator	C (8)	191	8

QCFFSY	System the filter was created on	C (8)	199	8
QCFFTR	File containing detailed filter information	C (10)	207	10
QCFJOB	Jobs to collect on: *, *ALL, *NONE or list file	C (10)	217	10
QCFTNM	Tasks to collect on: *ALL, *NONE or list file	C (227)	10	227
QCFTNB	Tasks numbers to collect on: *NONE or list file	C (10)	237	10
QCFMI	MI programs collected on: *NONE or list file	C (10)	247	10
QCFLIC	LIC modules collected on: *NONE or list file	C (10)	257	10
QCFCPX	Machine instructions collected on: *NONE or file	C (10)	267	10
QCFMET	Metrics collected on: *NONE or list file	C (10)	277	10
QCFORG	Stats data organization: *FLAT or *HIER	C (10)	287	10
QCFCMB	Merge stats data from all jobs into one - Y or N	C (1)	297	1
QCFLBK	Enable LIC bracketing - Y or N	C (1)	298	1
QCFSZ	Maximum size for trace mode data (in KB)	B (9,0)	299	4
QCFWRP	Wrap trace mode data - Y or N	C (1)	303	1
QCFOPT	HW mode option: *HDWEVT or *INSTCNT	C (15)	304	15
QCFTSO	HW mode time slice: *NON/TIMESLICED	C (15)	319	15
QCFFET	HW mode first hardware event table entry	B (4,0)	334	2
QCFLET	HW mode last hardware event table entry	B (4,0)	336	2
QCFFIT	HW mode first instruction count table entry	B (4,0)	338	2
QCFLIT	HW mode last instruction count table entry	B (4,0)	340	2
QCFITS	HW mode interval time slice in milliseconds	B(9,0)	342	4
QCFRND	Random mode	B (2,0)	346	2
QCFLAJ	List All Jobs/Tasks	B (2,0)	348	2
QCFGLB	Reserved	B (2,0)	350	2
QCFSMP	Reserved	B (2,0)	352	2
QCFFRC	Reserved	B (2,0)	354	2
QCFADD	Add Threads/Tasks	B (2,0)	356	2
QCFTYP	Collection type	C (10)	358	10
QCFPRF	Profile type	C (10)	368	10

12.7.4.8 QAYPECMN

This file contains raw communication events trace data. Much of this data follows a specific format based on each type of packet used. Some of these packets are externally documented on Wikipedia.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QCMTYP	Type: 1=Stream 2=Dgram 3=Raw 5=SeqPkt	B (4,0)	9	2
QCMAID	API identifier	B (9,0)	11	4
QCMDEN	Socket descriptor number	B (18,0)	15	8
QCMNRC	Return code	B (18,0)	23	8
QCMERR	Error number	B (18,0)	31	8
QCMLAD	Local address, key to IP in QAYPERINF if INET/6	H (16)	39	16
QCMLAF	Local address flag 4 = IPV4, 6 = IPV6	C (1)	55	1
QCMRAD	Remote address, key to IP in QAYPERINF if INET/6	H (16)	56	16
QCMRAF	Remote address flag 4 = IPV4, 6 = IPV6	C (1)	72	1
QCMLPO	Local port	B (9,0)	73	4
QCMRPO	Remote port	B (9,0)	77	4
QCMADF	Family:1=Ux 2=Inet 24=Inet6 6=NetS 99=Tel 200=NetB	B (4,0)	81	2
QCMRAW	Raw data bytes	C (502)	83	502

12.7.4.9 QAYPEDASD

This file provides disk event trace statistics.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QDDVAD	Virtual seg/obj address	B (8,0)	9	8
QDDPGS	Page count	B (18,0)	17	8
QDDUNB	Dasd unit number	B (4,0)	25	2
QDDASP	ASP Number	B (9,0)	27	4
QDDMSU	Mirror sub unit	B (4,0)	31	2
QDDADR	Dasd address	B (8,0)	33	8
QDDARE	Dasd area	B (9,0)	41	4

QDDSKP	Skip operation - Y or N	C (1)	45	1
QDDLRC	Dasd logical return code	B (4,0)	46	2
QDDBLK	Block size in bytes	B (4,0)	48	2
QDDPID	MS pool id	B (4,0)	50	2
QDDATC	Asynch IO indicator 0 = False 1 = True	B (4,0)	52	2
QDDECT	Disk I/O event count	B (9,0)	54	4
QDDEIO	Exchange I/O flag Y/N, also 2 thru 8, *	C (1)	58	1
QDDSPN	Span length in pages	B (18,0)	59	8
QDDCSM	Compressed skip mask	H (4)	67	4
QDDPATHF	Path flag - S=Single path M=Multipath	C (1)	71	1
QDDBUSA	Bus address	B (4,0)	72	2
QDDBRDA	Board address	B (4,0)	74	2
QDDCRDA	Card address	B (4,0)	76	2
QDDIOA	IO adapter address	B (4,0)	78	2
QDDIOABA	IO bus address	B (4,0)	80	2
QDDCTLA	Controller address	B (4,0)	82	2
QDDDEVA	Device address	B (4,0)	84	2
QDDSMTOTT	Total I/O time	B (18,0)	86	8
QDDIODFRT	Defer queue time	B (18,0)	94	8
QDDSMHDWT	Hardware time	B (18,0)	102	8
QDDDFRQUE	Number of ops on Defer queue	B (9,0)	110	4
QOBJKEY	Object key	B (8,0)	114	8
QDDDESC	Operation description	C (4)	122	4
QDDHINT	Cache hint	B (4,0)	126	2
QDDPRTY	Reserved	B (4,0)	128	2
>>> QDDPGSZ	Page size	B (4,0)	130	2 <<<
>>> QDDLINKAGE	Linkage indicator	B (18,0)	132	8 <<<
>>> QDDRSVD01	Reserved	B (4,0)	140	2 <<<
>>> QDDRSVD02	Reserved	B (9,0)	142	4 <<<

12.7.4.10 >>> QAYPEDBDMT <<<

This file is new at 7.2 and is produced when ADDPEXDFN DBEVT parameter includes *DASDMETER.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (Unique)	B (18,0)	1	8
QKEY	Key	B (18,0)	9	8
QREADRESP	Average read response time (microseconds)	B (18,0)	17	8
QREADCOUNT	Read count	B (9,0)	25	4
QAVGTOTCYC	Average total cycles	B (18,0)	29	8
QAVGRUNCYC	Average run cycles	B (18,0)	37	8
QAVGRATIO	Average total to run ratio	B (18,0)	45	8
QMAXTOTCYC	Maximum total cycles	B (18,0)	53	8
QMAXRUNCYC	Maximum run cycles	B (18,0)	61	8
QMAXRATIO	Maximum total to run ratio	B (18,0)	69	8
QDASDCOUNT	Number of DASD units in this ASP	B (9,0)	77	4
QEVENTTYPE	Event type	B (4,0)	81	2
QSTATE	Current state	B (4,0)	83	2
QACTION	Recommended action	B (4,0)	85	2
QASPNBR	ASP Number	B (4,0)	87	2
QRESERVED1	Reserved	B (4,0)	89	2
QRESERVED2	Reserved	B (4,0)	91	2
QRESERVED3	Reserved	B (4,0)	93	2
QRESERVED4	Reserved	B (4,0)	95	2
QRESERVED5	Reserved	B (9,0)	97	4
QRESERVED6	Reserved	B (9,0)	101	4
QRESERVED7	Reserved	B (18,0)	105	8
QRESERVED8	Reserved	B (18,0)	113	8

12.7.4.11 >>> QAYPEDBIO <<<

This file is new at 7.2 and is only produced when the ADDPEXDFN DBEVT parameter includes *IO.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QKEY	Key	B (18,0)	9	8
QOBJPTR	Object pointer	H (8)	17	8
QJBSSHRRES	Jobs share space to reserve	B (18,0)	25	8
QJBSSHR	Current jobs share	B (18,0)	33	8
QBYTESUSED	Bytes used	B (18,0)	41	8
QRRN	Relative record number	B (18,0)	49	8
QJBSHRPCT	Job share percent	P (5,2)	57	3
QREADSIZE	Read size	B (9,0)	60	4
QBUFFSIZE	Read buffer size	B (9,0)	64	4
QBLOCKING	Expert cache recommended blocking factor	B (9,0)	68	4
QDELTA	Buffer delta (bytes or elements)	B (9,0)	72	4
QPOOLINDEX	Storage pool index	B (4,0)	76	2
QTYPE	Requested I/O type	B (4,0)	78	2
QASP	ASP number	B (4,0)	80	2
QCLASS	Expert cache recommended class	B (4,0)	82	2
QREADACTN	Read action	C (1)	84	1
QRESERVED1	Reserved	B (4,0)	85	2
QRESERVED2	Reserved	B (4,0)	87	2
QRESERVED3	Reserved	B (4,0)	89	2
QRESERVED4	Reserved	B (4,0)	91	2
QRESERVED5	Reserved	B (9,0)	93	4
QRESERVED6	Reserved	B (9,0)	97	4
QRESERVED7	Reserved	B (18,0)	101	8
QRESERVED8	Reserved	B (18,0)	109	8

12.7.4.12 >>> QAYPEDBOPT <<<

This file is new at 7.2 and is only produced when the ADDPEXDFN DBEVT parameter includes *OPTIMIZER.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QPLANID	Plan ID (Unique)	B (18,0)	9	8
QOPID	Operation ID	B (18,0)	17	8
QSUBOPID	Sub-operation ID	B (18,0)	25	8
QKEY	Key	B (9,0)	33	4
QTYPE	Type (0=general 1=strategy 2=statistics)	B (4,0)	37	2
QOBJNAME	Object name	C (30)	39	30
QACTION	Action ID (1=start 2=end)	B (4,0)	69	2
QRESERVED1	Reserved	B (4,0)	71	2
QRESERVED2	Reserved	B (4,0)	73	2
QRESERVED3	Reserved	B (9,0)	75	4
QRESERVED4	Reserved	B (18,0)	79	8

12.7.4.13 >>> QAYPEDBOP2 <<<

This file is new at 7.2 and is only produced when the ADDPEXDFN DBEVT parameter includes *OPTIMIZER2.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QPLANID	Plan ID (Unique)	B (18,0)	9	8
QKEY	Key	B (9,0)	17	4
QDATA	Data	C (116)	21	116
QRESERVED1	Reserved	B (4,0)	137	2
QRESERVED2	Reserved	B (4,0)	139	2
QRESERVED3	Reserved	B (9,0)	141	4
QRESERVED4	Reserved	B (18,0)	145	8

12.7.4.14 >>> QAYPEDBRT <<<

This file is new at 7.2 and is only produced when the ADDPEXDFN DBEVT parameter includes *RUNTIME.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8

12.7.4.15 >>> QAYPEDBSGC <<<

This file is new at 7.2 and is only produced when the ADDPEXDFN DBEVT parameter includes *SEGMENTCACHE.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (Unique)	B (18,0)	1	8
QSEGADDR	Segment address	H (8)	9	8
QLINKREG	Link register of caller	H (8)	17	8
QKEY	Key of caller	H (8)	25	8
QSTMTNBR	Statement number of caller	B (9,0)	33	4
QSIZE	Requested segment size	B (18,0)	37	8
QMAXSIZE	Maximum requested seg size	B (18,0)	45	8
QBYTES	Bytes allocated/deallocated	B (18,0)	53	8
QBLKXFERSZ	Block transfer size	B (9,0)	61	4
QPAGINGPL	ID of paging pool	B (4,0)	65	2
QSEGTYPE	Segment type	H (2)	67	2
QCACHESLOT	Cache slot	B (4,0)	69	2
QCACHEBKT	Cache bucket	B (4,0)	71	2
QCLEAR	Clear segment requested	B (4,0)	73	2
QCLEARED	Segment cleared	B (4,0)	75	2
QTYPE	Requested operation type	B (4,0)	77	2
QRESERVED1	Reserved	B (4,0)	79	2
QRESERVED2	Reserved	B (4,0)	81	2
QRESERVED3	Reserved	B (4,0)	83	2
QRESERVED4	Reserved	B (4,0)	85	2
QRESERVED5	Reserved	B (9,0)	87	4
QRESERVED6	Reserved	B (9,0)	91	4

QRESERVED7	Reserved	B (18,0)	95	8
QRESERVED8	Reserved	B (18,0)	103	8

12.7.4.16 >>> QAYPEDBSL <<<

This file is new at 7.2 and is only produced when the ADDPEXDFN DBEVT parameter includes *SEIZELOCK.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8

12.7.4.17 QAYPEDFN

This file exists only in QSYS, and is used to build the QAPEXDFN file in QUSRSYS when a user runs the ADDPEXDFN command for the 1st time.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QAYPEDFN	This contains a binary structure used for saving information about the PEX definition. The iDoctor GUI knows how to interpret most of this which allows you to use the Add/Change PEX Definition Wizard in the GUI to make changes to definitions. Some newer functionality on the ADDPEXDFN command is not yet supported.	C (80)	1	80

12.7.4.18 QAYPEDSRV

This file produces disk server event data and is only produced if ADDPEXDFN includes events from the Disk Server Events category (DSKSVREVT parameter.)

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QDSPRM	DS Parameters	C (96)	9	96
QDSEXT	DS External Info	C (24)	105	24
QDSMSG	Work Message Address	H (8)	129	8
QDSOBJ	Server Object Address	H (8)	137	8
QDSTSK	DS Task Id	C (2)	145	2

QDSCDE	DS Termination Code	B (4,0)	147	2

12.7.4.19 QAYPEEVENT

This file captures information about the event types that were collected during the PEX collection. It also lists every event available at the current release and how many occurred for each one.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QEVTY	Event type	B (4,0)	1	2
QEVSTY	Event subtype	B (4,0)	3	2
QEVSN	Event type description (short)	C (20)	5	20
QEVSSN	Event subtype description (short)	C (20)	25	20
QEVLN	Event type description	C (50)	45	50
QEVSLN	Event subtype description	C (50)	95	50
QEVCT	Number of occurrences of this event	B (18,0)	145	8

12.7.4.20 QAYPEFILSV

This file provides file server event details. It is only included in the collection if the definition includes file server events. This is specified using the ADDPEXDFN command FILSVREVT parameter.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QFSRRF	Operation indicator	C (6)	9	6
QFSSKY	iSeries NetServer session key	H (16)	15	16
QFSRQF	Requested function	B (18,0)	31	8
QFSERN	ERRNO from the function	B (18,0)	39	8
QFSPRT	Protocol type	B (9,0)	47	4
QFSWDH	Working directory handle	B (9,0)	51	4
QFSFIH	File handle	B (9,0)	55	4
QFSFFF	FFDC footprint	B (9,0)	59	4
QFSNFV	NFS version	B (4,0)	63	2
QFSNFR	NFS: RPC function number	B (4,0)	65	2

QFSNFU	NFS: UID	B (9,0)	67	4
QFSNFE	NFS: RPC Errno	B (9,0)	71	4
QFSIPF	IP address flag 4 = IPV4 6 = IPV6	C (1)	75	1
QFSIPA	IP address	H (16)	76	16
QFSFNM	File name (in unicode)	G (642)	92	642
QFSSMB	SMB data or NFS dependent data	C (130)	734	130

12.7.4.21 QAYPEFQCFG

This file provides hardware configuration frequency information.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QFQTE	Hardware table entry	B (9,0)	1	4
QFQCNT	Hardware counter	B (4,0)	5	2
QFQFR2	Interval in usecs if tbl entry 1, counter 1	B (9,0)	7	4

12.7.4.22 QAYPEFTRI

This file provides information about the PEX filter used when the collection was created.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QFTFTRVER	Filter version	B (4,0)	1	2
QFTUSERID	Created by	C (10)	3	10
QFTFTRTYP	Filter: BAS ASP DSK IFS IP JVA MEM OBJ PGM or USR	C (3)	13	3
QFTTRGTYP	Trigger: PGMENTEXIT PGMENT or JVAENTEXIT	C (10)	16	10
QFTCMF	Compare flag: *EQ *NE *GT or *LT	C (3)	26	3
QFTFVL	Filter length (in bytes)	B (9,0)	29	4
QFTILI	IP: Local IP address	C (63)	33	63
QFTIRI	IP: Remote IP address	C (63)	96	63
QFTILP	IP: Local port	B (9,0)	159	4
QFTIRP	IP: Remote port	B (9,0)	163	4
QFTIAF	Addr family: 2=INET 24=INET6 99=UNIX	B (4,0)	167	2
QFTICT	Type: 1=STREAM 2=DGRAM 3=RAW 5=SEQPACKET	B (4,0)	169	2

QFTPLB	PGM: Program library	C (10)	171	10
QFTPPG	Program or filter name	C (10)	181	10
QFTPMO	Module name System name	C (10)	191	10
QFTPTY	PGM: Program type *PGM or *SRVPGM	C (10)	201	10
QFTCF1	Procedure/method or generic or filter text	C (256)	211	256
QFTHF1	Generic hex bytes filter value 1	H (15)	467	15
QFTJVP	JVA: Package name	C (64)	482	64
QFTJVC	JVA:Class name	C (64)	546	64
QFTUFT	USR: Event type this filter for: 1 to 31	B (4,0)	610	2
QFTUFS	USR: Event subtype this filter for: 1 to 31	B (4,0)	612	2
QFTUFO	USR: Filter offset in the record	B (4,0)	614	2
QFTUFF	USR: Type: CHAR HEX INT1/2/4/8 UINT1/2/4/8	C (10)	616	10
QFTUFC	USR: Number of compare values specified	B (4,0)	626	2
QFTCF2	USR: Character filter value 2	C (30)	628	30
QFTCF3	USR: Character filter value 3	C (30)	658	30
QFTCF4	USR: Character filter value 4	C (30)	688	30
QFTCF5	USR: Character filter value 5	C (30)	718	30
QFTHF2	USR: Hex bytes filter value 2	H (15)	748	15
QFTHF3	USR: Hex bytes filter value 3	H (15)	763	15
QFTHF4	USR: Hex bytes filter value 4	H (15)	778	15
QFTHF5	USR: Hex bytes filter value 5	H (15)	793	15

12.7.4.23 QAYPEHEAP

This file is produced when the PEX trace collection includes heap events. This is specified using the ADDPEXDFN command STGEVT parameter with values that contain “HEAP”.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QHPNAM	Heap name	C (16)	9	16
QHPHCS	Heap control segment address	H (8)	25	8

QHPASA	Allocation start address	H (8)	33	8
QHPASZ	Allocation size in bytes	B (9,0)	41	4
QHPOPR	Op type: 0=new 1=delete 32,33=create,destroy heap	B (4,0)	45	2
QHPRET	Return code: 0 = ok	B (4,0)	47	2
QHPCA1	Caller instruction address 1	H (8)	49	8
QHPCK1	Key for caller instruction address 1	H (8)	57	8
QHPCA2	Caller instruction address 2	H (8)	65	8
QHPCK2	Key for caller instruction address 2	H (8)	73	8
QHPCA3	Caller instruction address 3	H (8)	81	8
QHPCK3	Key for caller instruction address 3	H (8)	89	8
QHPCA4	Caller instruction address 4	H (8)	97	8
QHPCK4	Key for caller instruction address 4	H (8)	105	8
QHPCA5	Caller instruction address 5	H (8)	113	8
QHPCK5	Key for caller instruction address 5	H (8)	121	8
QHPMSC	Miscellaneous text data	C (16)	129	16

12.7.4.24 QAYPEHMON

This file provides hardware data information.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record Number (UNIQUE)	B (18,0)	1	8
QHMPRC	Processor	B (4,0)	9	2
QHMNM	Event/Instruction Name	C (256)	11	256
QHMCNT	PMC count	B (18,0)	267	8
QHMMCR	MMCR0 register	H (8)	275	8
QHMMCR1	MMCR1 register	H (8)	283	8
QHMIMR	IMR register	H (8)	291	8
QHMTE	Table entry number	B (9,0)	299	4
QHMPMC	PMCS number	B (4,0)	303	2
QHMMCRA	MMCRA register	H (8)	305	8

QHMSHNM	Event/instruction short name	C (32)	313	32
QHMSHGRP	Group short name	C (32)	345	32
QHMGRPNM	Group Name	C (256)	377	256

12.7.4.25 QAYPEHTOT

This file includes hardware instruction totals on a per processor basis.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QHTPRC	Processor	B (4,0)	1	2
QHTINS	Total number of instructions	B (18,0)	3	8
QHTCYC	Total number of cycles	B (18,0)	11	8
QHTRNC	Total number of run cycles	B (18,0)	19	8

12.7.4.26 QAYPEIAD

This file produces instruction addresses, keys and statement numbers for call stacks produced by several format 2 events in PEX. Additionally the format 1 STGEVT parameter *HEAP* events will produce call stack addresses in this file.

To convert an instruction address to the readable program, procedure name use the QAYPEPROCI file.

Note: PEX call stacks using this file support a maximum of 16 call levels.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QIAIAD1	Instruction address 1	H (8)	9	8
QIAIAD2	Instruction address 2	H (8)	17	8
QIAIAD3	Instruction address 3	H (8)	25	8
QIAIAD4	Instruction address 4	H (8)	33	8
QIAIAD5	Instruction address 5	H (8)	41	8
QIAIAD6	Instruction address 6	H (8)	49	8
QIAIAD7	Instruction address 7	H (8)	57	8
QIAIAD8	Instruction address 8	H (8)	65	8
QIAIAD9	Instruction address 9	H (8)	73	8

QIAIAD10	Instruction address 10	H (8)	81	8
QIAIAD11	Instruction address 11	H (8)	89	8
QIAIAD12	Instruction address 12	H (8)	97	8
QIAIAD13	Instruction address 13	H (8)	105	8
QIAIAD14	Instruction address 14	H (8)	113	8
QIAIAD15	Instruction address 15	H (8)	121	8
QIAIAD16	Instruction address 16	H (8)	129	8
QIAKEY1	Address key 1	H (8)	137	8
QIAKEY2	Address key 2	H (8)	145	8
QIAKEY3	Address key 3	H (8)	153	8
QIAKEY4	Address key 4	H (8)	161	8
QIAKEY5	Address key 5	H (8)	169	8
QIAKEY6	Address key 6	H (8)	177	8
QIAKEY7	Address key 7	H (8)	185	8
QIAKEY8	Address key 8	H (8)	193	8
QIAKEY9	Address key 9	H (8)	201	8
QIAKEY10	Address key 10	H (8)	209	8
QIAKEY11	Address key 11	H (8)	217	8
QIAKEY12	Address key 12	H (8)	225	8
QIAKEY13	Address key 13	H (8)	233	8
QIAKEY14	Address key 14	H (8)	241	8
QIAKEY15	Address key 15	H (8)	249	8
QIAKEY16	Address key 16	H (8)	257	8
QIASTMT1	Statement 1	B (9,0)	265	4
QIASTMT2	Statement 2	B (9,0)	269	4
QIASTMT3	Statement 3	B (9,0)	273	4
QIASTMT4	Statement 4	B (9,0)	277	4
QIASTMT5	Statement 5	B (9,0)	281	4
QIASTMT6	Statement 6	B (9,0)	285	4
QIASTMT7	Statement 7	B (9,0)	289	4

QIASTMT8	Statement 8	B (9,0)	293	4
QIASTMT9	Statement 9	B (9,0)	297	4
QIASTMT10	Statement 10	B (9,0)	301	4
QIASTMT11	Statement 11	B (9,0)	305	4
QIASTMT12	Statement 12	B (9,0)	309	4
QIASTMT13	Statement 13	B (9,0)	313	4
QIASTMT14	Statement 14	B (9,0)	317	4
QIASTMT15	Statement 15	B (9,0)	321	4
QIASTMT16	Statement 16	B (9,0)	325	4

12.7.4.27 QAYPEJVA, QAYPEJVCI, QAYPEJVMI, QAYPEJVNI

These files are now obsolete at 7.1+ since they applied to classic Java only.

12.7.4.28 QAYPELBRKT

This file includes LIC bracket event data.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QLBTBT	Traceback table address for the procedure	H (8)	9	8
QLBIAD	Instruction address	H (8)	17	8
QLBR01	General purpose register 1(stack pointer)	H (8)	25	8
QLBCIA	Caller instruction address	H (8)	33	8
QLBCTB	Caller traceback table address for the procedure	H (8)	41	8

12.7.4.29 QAYPELCPLX

This file includes a list of MI complex instructions that were specified to be collected on the PEX definition used for this collection.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QLBTBT	Traceback table address for the procedure	H (8)	9	8
QLBIAD	Instruction address	H (8)	17	8

QLBR01	General purpose register 1(stack pointer)	H (8)	25	8
QLBCIA	Caller instruction address	H (8)	33	8
QLBCTB	Caller traceback table address for the procedure	H (8)	41	8

12.7.4.30 QAYPELJOB

This file includes the job details that were specified to be collected on the PEX definition used for this collection.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLJNM	Job name	C (10)	1	10
QLJUSR	Job user	C (10)	11	10
QLJNB	Job number	C (6)	21	6
QLJTHD1	Thread 1	C (8)	27	8
QLJTHD2	Thread 2	C (8)	35	8
QLJTHD3	Thread 3	C (8)	43	8
QLJTHD4	Thread 4	C (8)	51	8
QLJTHD5	Thread 5	C (8)	59	8
QLJTHD6	Thread 6	C (8)	67	8
QLJTHD7	Thread 7	C (8)	75	8
QLJTHD8	Thread 8	C (8)	83	8
QLJTHD9	Thread 9	C (8)	91	8
QLJTHD10	Thread 10	C (8)	99	8
QLJTHD11	Thread 11	C (8)	107	8
QLJTHD12	Thread 12	C (8)	115	8
QLJTHD13	Thread 13	C (8)	123	8
QLJTHD14	Thread 14	C (8)	131	8
QLJTHD15	Thread 15	C (8)	139	8
QLJTHD16	Thread 16	C (8)	147	8
QLJTHD17	Thread 17	C (8)	155	8
QLJTHD18	Thread 18	C (8)	163	8
QLJTHD19	Thread 19	C (8)	171	8
QLJTHD20	Thread 20	C (8)	179	8

QLSBSNM	Subsystem name	C (10)	187	10

12.7.4.31 QAYPELLIC

This file includes the LIC procedure list that were specified to be collected on the PEX definition used for this collection.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLLSEL	Selection method	B (9,0)	1	4
QLLMID	SLIC module Id	C (8)	5	8
QLLSAD	Start address of range	C (16)	13	16
QLLEAD	End address of range	C (16)	29	16
QLLPSZ	Pane size	B (9,0)	45	4

12.7.4.32 QAYPELMET

This file provides a list of event definitions that were specified on the definition to be collected.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLMCTR	Counter	B (4,0)	1	2
QLMCAT	Metric category	C (20)	3	20
QLMTYP	Metric type	C (20)	23	20
QLMFMT	Metric format	B (4,0)	43	2

12.7.4.33 QAYPELMI

This file provides a list of MI programs that were specified on PEX Profile type definition to be captured.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLMTYP	MI program type	C (10)	1	10
QLMNM	MI program name	C (10)	11	10
QLMLIB	MI program library name	C (10)	21	10
QLMMNM	MI module name	C (10)	31	10
QLMPNM	MI procedure name	C (258)	41	258
QLMPSZ	Pane size	B (9,0)	299	4

12.7.4.34 QAYPELNAMT

This file includes a list of task names that were specified to be included on the PEX definition.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLTTNM	Task name	C (16)	1	16

12.7.4.35 QAYPELNUMT

This file includes a list of task numbers that were specified to be included on the PEX definition.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLTTNB	Task number	H (16)	1	16

12.7.4.36 QAYPELTASK

Field Name	Description	Attribute	Buffer Position	Buffer Length
QLTFLG	Task flag: 0= Task Name 1= Task Number	C (1)	1	1
QLTTNB	Task value	C (16)	2	16

12.7.4.37 QAYPEMBRKT

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QMBTBT	Procedure traceback table address	H (8)	9	8
QMBIAD	Instruction address	H (8)	17	8
QMBIDX	Instruction index	B (4,0)	25	2
QMBHLL	HLL statement number	B (9,0)	27	4
QMBCIA	Caller instruction address	H (8)	31	8
QMBCTB	Caller traceback table address	H (8)	39	8
QMBJIM	Interpreted Java method table entry address	H (8)	47	8
QMBJII	Interpreted Java instruction offset	B (9,0)	55	4
QMBJCA	Interpreted Java class address	H (8)	59	8

12.7.4.38 QAYPEMICPX

This file provides a complete list of the IBM i MI complex instructions.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QCPIDX	Complex MI index	B (4,0)	1	2
QCPSNM	Short complex MI description	C (20)	3	20
QCPLNM	Long complex MI description	C (50)	23	50

12.7.4.39 QAYPEMIPTR

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QMIPTR	MI pointer address	H (8)	9	8

12.7.4.40 QAYPEMIUSR

This file is produced when either the APIs are used to start and end user-defined PEX transactions or when the operating system events (OSEVT parameter) are included on the PEX definition.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QMUCDE	User defined code	B (9,0)	9	4
QMUPTR	Number of MI pointers at start of QMUDTA	B (4,0)	13	2
QMUDTA	User defined data **For format information see the next section.**	C (4018)	15	4018
QMUSDBR	Synchronous DB Reads	B (18,0)	4033	8
QMUSNDBR	Synchronous NDB Reads	B (18,0)	4041	8
QMUADBR	Asynchronous DB Reads	B (18,0)	4049	8
QMUANDBR	Asynchronous NDB Reads	B (18,0)	4057	8
QMUSDBW	Synchronous DB Writes	B (18,0)	4065	8
QMUSNDBW	Synchronous NDB Writes	B (18,0)	4073	8
QMUADBW	Asynchronous DB Writes	B (18,0)	4081	8
QMUANDBW	Asynchronous NDB Writes	B (18,0)	4089	8
QMUAIOW	Waits on Asynchronous IO	B (18,0)	4097	8
QMUSPFW	Waits on Synchronous IO Pend Faults	B (18,0)	4105	8

QMUDBLK	DB Locks	B (18,0)	4113	8
QMUNDBLK	NDB Locks	B (18,0)	4121	8
QMUSZ	Seizes	B (18,0)	4129	8
QMUDBLKTM	DB Locks Microseconds	B (18,0)	4137	8
QMUNDBLKTM	NDB Locks Microseconds	B (18,0)	4145	8
QMUSZTM	Seizes Microseconds	B (18,0)	4153	8

12.7.4.40.1 QMUDTA format information

12.7.4.40.1.1 OSEVT (*DBIO, *DBOPEN)

Byte 1-3: event flag
'CLF'=close
'CLS'=sclose
'DEL'=delete
'GTD'=get direct
'GTK'=get keyed
'GTM'=get multiple
'GTS'=get sequential
'OPF'=open
'OPS'=sopen
'PTM'=put multiple
'PUT'=put
'PTD'=put direct
'RLS'=release
'UPD'=update
Byte 4: key flag
Byte 5-14: file name
Byte 15-24: library name
Byte 25-34: member name
Byte 35-38: option list
Byte low order bit in 38 = Data/no data flag bit
Byte 39-48: requested format name
Byte 49-50: # key fields or key length
Byte 51-54: # records retrieved
Byte 55-58: relative record #
Byte 59-60: member #
Byte 61-67: exception signaled / return code

12.7.4.40.1.2 OSEVT(*DTAARA)

Byte 1-3: event flag

'CDA'(CHGDTAARA)

'RDA'(RTVDTAARA)

Byte 4: Local/DDM flag = 'L' or 'D'

Byte 5-14: data area name

Byte 15-24 data area library

Byte 25 data type (i.e. char='C')

Byte 26-27: start

Byte 28-29: value length

Byte 30-49: value

12.7.4.40.1.3 OSEVT(*DTAQ and 1st 3 bytes = 'DQR')

Byte 1-3 event flag = 'DQR'

Byte 4 Local/DDM flag = 'L' or 'D'

Byte 5 msg found? 'Y' or 'N'

Byte 6-15 queue name

Byte 16-25 queue library

Byte 26-28 wait time

Byte 29-31 length

Byte 32-51 the msg

Byte 52-53 key length or 0 if non keyed (also end of non-keyed record)

Byte 54-73 key

Byte 74-75 key order

12.7.4.40.1.4 OSEVT(*DTAQ and 1st 3 bytes = 'DQS')

Byte 1- 3 event flag = 'DQS'

Byte 4- 4 Local/DDM flag = 'L' or 'D'

Byte 5-14 queue name

Byte 15-24 queue library

Byte 25-27 length

Byte 28-47 the msg

Byte 48-49 key len or 0 if non keyed (also end of non-keyed record)

Byte 50-69 key value

12.7.4.40.1.5 OSEVT (*IFSOPEN, *IFSIO)

For AddPexDfn OSEVT(*IFSOPEN and *IFSIO): Note: Opens, creates and closes are recorded if *ifsopen All others are recorded if *ifsio QAYPEMIUSR's QMUCDE field will contain a file descriptor number for successful opens and creates - and all subsequent i/o operations. This file descriptor can be used to group events for a given file, or to find the file name for a particular event, assuming the open/create was recorded (meaning addpexdfn osevt(*ifsopen *ifsio) specified and strpex was done prior to the file being opened or created).

The layout of the data in QMUDTA is:

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 IFS eyecatcher char(4) - always set to "IFS"
4 04 API identifier char(20) - API name as defined in the api_names array (currently in Qp0lUti2.C, CCSID 37)
24 18 errno signed 4-byte - zero if function returned successfully
28 1C API-specific info char(164) - See definitions below
192 C0 --- End ---

Layout of the data starting at offset Hex 1C:

open(), open64(), Qp0lOpen()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 oflag signed 4-byte

4 04 mode signed 4-byte (zero if not specified)

8 08 conversion ID signed 4-byte (zero if not specified)

12 0C path CCSID signed 4-byte - used to interpret the path - may be job CCSID

16 10 language char(3)

19 13 region or country char(2)
16 10 path char(143) - If path is longer than 143 bytes, the LAST 143 bytes will be stored here. The field will be blank padded.
164 A4 --- End ---

creat(), creat64()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 mode signed 4-byte (zero if not specified)

4 04 path CCSID signed 4-byte - used to interpret the path - may be jobCCSID

8 08 language char(3)

11 0B region or country char(2)
13 0D path char(151) - If path is longer than 151 bytes, the LAST 151 bytes will be stored here. The field will be blank-padded.
164 A4 --- End ---

read()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 nbyte signed 4-byte

4 04 bytes returned signed 4-byte

8 08 Reserved char(156), zeroes
164 A4 --- End ---

readv()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 Number of vectors signed 4-byte

4 04 bytes returned signed 4-byte

8 08 vector lengths array of 39 4-byte, containing all the vector lengths
164 A4 --- End ---

write()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 nbyte signed 4-byte

4 04 bytes written signed 4-byte

8 08 Reserved char(156), zeroes
164 A4 --- End ---

writev()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 Number of vectors signed 4-byte

4 04 bytes written signed 4-byte

8 08 vector lengths array of 39 4-byte, containing all the vector lengths
164 A4 --- End ---

close()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 Reserved char(164) - zeroes
164 A4 --- End ---

dup()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 new file descriptor signed 4-byte

4 04 Reserved char(160) - zeroes
164 A4 --- End ---

dup2()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 new file descriptor signed 4-byte

4 04 requested file descriptor signed 4-byte

8 08 Reserved char(156) - zeroes
164 A4 --- End ---

lseek()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 input offset signed 4-byte

4 04 output offset signed 4-byte

8 08 whence signed 4-byte

12 0C Reserved char(152) - zeroes
164 A4 --- End ---

lseek64()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 input offset signed 8-byte

8 08 output offset signed 8-byte

16 10 whence signed 4-byte

20 14 Reserved char(144) - zeroes
164 A4 --- End ---

ftruncate()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 length signed 4-byte

4 04 Reserved char(160) - zeroes
164 A4 --- End ---

ftruncate64()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 length signed 8-byte

8 08 Reserved char(156) - zeroes
164 A4 --- End ---

fsync()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 Reserved char(164) - zeroes
164 A4 --- End ---

fcntl()

Offset Dec Hex Field Name Data Type and Length

--- --- ---------- ----------

0 00 command signed 4-byte

4 04 return value signed 4-byte - This will have different meanings based on the command

8 08 third parameter length signed 4-byte - This will be the length of the following field.
12 0C third parameter variable - This will be the exact data specified as the third parameter on the API. It will normally be a 4-byte integer, an 8-byte integer, a flock structure, or an flock64 structure n n Reserved char(x) - Everything after the third parameter field will be zeroes.

12.7.4.40.1.6 DBSVRCNN and DBSVRREQ

For AddPexDfn OSEVT(*DBSVRCNN and *DBSVRREQ):

the layout of the data in QMUDTA is:
Layout for QMUDTA:

CONNECT (SVC)

WSQ_CLI_PEX_ID CHAR(4), /* eye catcher */

WSQ_CLI_PEX_THRD BIN(32), /* thread (or CLI handle) */

WSQ_CLI_PEX_J# CHAR(6), /* job number of the QSQSRVR job */

WSQ_CLI_PEX_JN CHAR(26), /* job name */
WSQ_CLI_PEX_USER CHAR(10); /* user */

DISCONNECT (SVD)

WSQ_CLI_PEX_ID CHAR(4), /* eye catcher */
WSQ_CLI_PEX_THRD BIN(32), /* thread (or CLI handle) */

CLI FUNCTION (CLI)

WSQ_CLI_PEX_ID CHAR(4), /* eye catcher */

WSQ_CLI_PEX_HNDL BIN(32), /* statement handle */

WSQ_CLI_PEX_J# CHAR(6), /* job number of the QSQSRVR job */

WSQ_CLI_PEX_JN CHAR(26), /* job name */
WSQ_CLI_PEX_STMT CHAR(3000); /* SQL statement text */

QSQSRVR (entry/exit) /* New for V5R3 */

WSQ_CLI_PEX_ID CHAR(14), /* eye catcher */

WSQ_CLI_PEX_JN CHAR(10), /* job name of the calling job */

WSQ_CLI_PEX_USER CHAR(10), /* user of the calling job */

WSQ_CLI_PEX_J# CHAR(8), /* job number of the calling job */
WSQ_CLI_PEX_CUSER CHAR(10); /* profile used on the connect */

function subtype eyecatcher user code

SQLConnect 5 SVC 3

SQLDisconnect 5 SVD 4

QSQSRVR job 5 QSQSRVR ENTRY 5 /* V5R3 */
QSQSRVR job 5 QSQSRVR EXIT 6 /* V5R3 */

user code (entry,exit)

SQLPrepare 11 CLI 1,2

SQLExecute 11 CLI 3,4

SQLExecDirect 11 CLI 5,6

SQLFetch 11 CLI 7,8

SQLFetchScroll 11 CLI 9,10
SQLExtFetch 11 CLI 11,12

12.7.4.40.1.7 OSEVT(*USRTNS)

For AddPexDfn OSEVT(*USRTNS): the layout of the data in QMUDTA is:

Start offset	End offset	Data type	Description
001	004	Char(4)	"API " eye catcher
005	024	Char(20)	Application identifier
025	025	Char(1)	Type of data: '0' - Generic trace point '1' - Start of transaction '2' - End of transaction '3' - Log transaction
026	035	Char(10)	Transaction identifier
036	036	Char(1)	Filler for alignment

			Start of peformance counters
037	044	Unsigned binary(8)	Number of synchronous database reads
045	052	Unsigned binary(8)	Number of synchronous non-database reads
053	060	Unsigned binary(8)	Number of asynchronous database reads
061	068	Unsigned binary(8)	Number of asynchronous non-database reads
069	076	Unsigned binary(8)	Number of synchronous database writes
077	084	Unsigned binary(8)	Number of synchronous non-database writes
085	092	Unsigned binary(8)	Number of asynchronous database writes
093	100	Unsigned binary(8)	Number of asynchronous non-database writes
101	108	Unsigned binary(8)	Number of waits for asynchronous I/O
109	116	Unsigned binary(8)	Number of I/O pending faults
117	124	Unsigned binary(8)	CPU time (milliseconds)
125	132	Unsigned binary(8)	Number of database locks
133	140	Unsigned binary(8)	Number of non-database locks
141	148	Unsigned binary(8)	Number of seizes
149	156	Unsigned binary(8)	Database lock time (milliseconds)
157	164	Unsigned binary(8)	Non-database lock time (milliseconds)
165	172	Unsigned binary(8)	Seize time (milliseconds)
			End of peformance counters
173	176	Unsigned binary(4)	Length of user data
177	*	Char(*)	User data

12.7.4.41 QAYPEPASE

This file includes PASE event data. It is captured by specifying a value other than *NONE on the ADDPEXDFN command's PASEEVT parameter.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QPAPID	Process ID	H (8)	9	8
QPATID	PASE thread ID	H (8)	17	8
QPAPGN	PASE program path name	C (32)	25	32
QPASCV	***SIGPRCSND, *SIGTHDSND** Signal number ***SIGRCV**	H (8)	57	8

	Byte 1: Signal action TiaSignalAuditAction Bytes 2-4: Reserved Byte 5-8: Signal number ***SYSCALLSTR** System call vector number			
QPAS1P	***FORKEND** Parent process ID ***SIGPRCSND, *SIGTHDSND** Target process ID ***SYSCALLSTR** Parameter 1	H (8)	65	8
QPAS2T	***THDINITSTR** New thread ID ***SIGTHDSND** Target thread ID ***SYSCALLSTR** Parameter 2	H (8)	73	8
QPAS3T	***SYSCALLSTR** Parameter 3 ***TRCHOOK** Trace channel	H (8)	81	8
QPAS4H	***SYSCALLSTR** Parameter 4 ***TRCHOOK** Hook ID	H (8)	89	8
QPAS5W	***SYSCALLSTR** Parameter 5 ***TRCHOOK** Data from hook word	H (8)	97	8
QPAS6W	***SYSCALLSTR** Parameter 6 ***TRCHOOK** Data Word 1	H (8)	105	8
QPASC7	***SYSCALLSTR** Parameter 7	H (8)	113	8
QPASC8	***SYSCALLSTR** Parameter 8	H (8)	121	8

QPAERN	***SYSCALLEND** Error number	H (8)	129	8
QPASCR	***SYSCALLEND** Return code ***EXIT** Exit return code/status	H (8)	137	8
QPACMN	***SYSCALLSTR/END, *LOADSTR** System call name	C (32)	145	32
QPATDT	***TRCHOOK** Remaining data up to max of 1024	C (1026)	177	1026

12.7.4.42 QAYPEPERD

This file contains PEX periodic mode event data. It is suspected that this file is no longer generated/used by PEX at current releases.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECNB	Record number (UNIQUE)	B (9,0)	1	4
QPRTY	Periodic mode type	P (3,0)	5	2
QPRSTY	Periodic mode subtype	P (3,0)	7	2
QPRITV	Periodic mode interval	P (9,0)	9	5
QPRVRS	Periodic mode version	P (3,0)	14	2
QPRDTA	Periodic mode data	C (4034)	16	4034

12.7.4.43 QAYPEPGFLT

This file includes page fault event data. It is captured by specifying a value other than *NONE on the ADDPEXDFN command's FAULTEVT parameter.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QPGNIA	Next instruction address	H (8)	9	8
QPGNIK	Key for next instruction address	H (8)	17	8
QPGVAD	Faulting virtual address	H (8)	25	8
QPGTYP	Page fault type	B (4,0)	33	2
QPGEXC	Exception id	B (4,0)	35	2

QPGMSP	MI suspend point address	H (8)	37	8
QPGSPK	Key for MI suspend point	H (8)	45	8
QPGFTC	Processing task count	H (8)	53	8
QPGRET	Related event	B (18,0)	61	8
QOBJKEY	Object key	H (8)	69	8
QPGPGSZ	Page size	B (4,0)	77	2
QPGRSVD01	Reserved	B (4,0)	79	2
QPGRSVD02	Reserved	B (9,0)	81	4

12.7.4.44 QAYPEPPANE

This file provide PEX profile pane data and is only included when running a PEX Profile type collection.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QPNPID	Partition id	B (4,0)	1	2
QPNWID	Window id	B (9,0)	3	4
QPNID	Pane id	B (9,0)	7	4
QPNTBT	Procedure traceback table address	H (8)	11	8
QPNSAD	First instruction in the pane	H (8)	19	8
QPNCNT	Number of sample hits in the pane	B (18,0)	27	8
QPNSTM	MI statement number of instruction	B (9,0)	35	4
QPNSTF	Mapping state for a pane	B (9,0)	39	4

12.7.4.45 QAYPEPROCI

This file is used for resolving program/procedure names from procedure trace back table addresses. It is a required file for producing call stacks in PEX from format 2 events. Also see file QAYPEIAD.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QPRKEY	Procedure trace back table	H (8)	1	8
QPRPGN	MI program name	C (30)	9	30
QPRPQL	MI program library name	C (10)	39	10

QPRTY	MI object type	B (4,0)	49	2
QPRSTY	MI object subtype	B (4,0)	51	2
QPRPMD	Model: 0=PASE/LIC 1=ILE 2=srvpgm 3=OPM 4=Java	B (4,0)	53	2
QPRMNM	Module name	C (258)	55	258
QPRMQL	MI module name qualifier (library)	C (10)	313	10
QPRMTM	Module time stamp	C (16)	323	16
QPRRUN	LIC module RU name	C (8)	339	8
QPRMSA	LIC module start address	H (8)	347	8
QPRPNM	Procedure name	C (258)	355	258
QPRSAD	Procedure code start address	H (8)	613	8
QPREAD	Procedure code end address KEY Field	H (8)	621	8
QPRCSZ	Procedure code size in bytes	B (18,0)	629	8
QPRLNG	Procedure language	H (8)	637	8
QPRPSP	PASE load module path	C (258)	645	258
QPRPTI	PASE proc type 0 = ILE 1 = PASE	C (1)	903	1
QPRPFT	PASE fcn type: 1=User 2=Kernel 3=shared object	C (1)	904	1
QPRPAT	PASE address type: 1 = 32 bit 2 = 64 bit	C (1)	905	1
QPRPTC	PASE procedure initial thread task count	H (8)	906	8
QPRHDR	Header	C (2)	914	2

12.7.4.46 QAYPEREF

This file is no longer used by PEX.

12.7.4.47 QAYPERINF

This file provides miscellaneous resolution information.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRKEY	Key to resolution data	H (16)	1	16
QRTYP	Type: 1 = IP addr	B (4,0)	17	2
QRDTA	Type 1 resolution data	C (64)	19	64

12.7.4.48 QAYPERLS

This file provides information about the version of PEX database files.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRLVRM	Database VRM	C (6)	1	6
QRLLVL	PEX level indicator	P (5,0)	7	3

12.7.4.49 QAYPERMPM

This file includes resource management process management event data. It is captured by specifying a value other than *NONE on the ADDPEXDFN command's JOBEVT parameter.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QRPMSP	Main storage pool id	B (4,0)	9	2
QRPSP	Storage pool id	B (4,0)	11	2
QRPCCI	Cost curve index	B (9,0)	13	4
QRPLWR	***LWSTR** Long wait reason	B (9,0)	17	4
QRPITY	***INTERRUPT** Interrupt type	B (4,0)	21	2
QRPOPL	***MPLPOOLCHG, *TOBCHMPLPOOL** Old main storage pool id	B (4,0)	23	2
QRPCPU	***TSLEND** CPU time used	B (18,0)	25	8
QRPTIO	***TSLEND** Total I/O count	B (18,0)	33	8
QRPMTS	***TSLEND** MI time slice expired	C (1)	41	1

12.7.4.50 QAYPERMSL

This file provides resource management seize lock event data. The file is only captured when the command ADDPEXDFN includes values other than *NONE for the LCKEVT parameter.

Note: TCS refers to Transaction Control Structure.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QSLETM	Time elapsed in microseconds	B (18,0)	9	8
QSLSEG	Last object address segment	H (8)	17	8
QSLOFF	Last object address offset	B (9,0)	25	4
QSLNSZ	Number of seizes	B (18,0)	29	8
QSLRET	Number of retries	B (18,0)	37	8
QSLHLD	Integer hold type	B (4,0)	45	2
QSLCFL	Last conflicting hold type	B (4,0)	47	2
QSLLCKSTS	Lock status xferReq = 1 lockCheck = 2 lockSat = 3 unLockSat = 4 xferSat = 5 scopeChgSat = 6 lockConflict = 7 unlockFailed = 8 xferFailed = 9 lockTimeOut = 10 scopeChgFailed = 11 xferConflict = 12	B (4,0)	49	2
QSLHLDT	Hold type DBWK - Lock Record Weak DBRD - Lock Record Read DBUP - Lock Record Update LSRD - Lock shared read LSRO - Lock read only LSUP - Lock allow update LEAR - Lock Exclusive Allow Read LENR - Lock Exclusive NVAL -	C (4)	51	4
QSLHLDST	Hold subtype (I=implicit E=Explicit)	C (1)	55	1
QSLSCOPE	Scope(0 = Job, 1 = thread, 2 = TCS)	B (4,0)	56	2
QSLRRN	Record number	B (18,0)	58	8
QSLTSCOPE	Transfer scope	B (4,0)	66	2

QSLHLDTDE	Holder TDE number	H (8)	68	8
QSLCONTDE	Conflictor TDE number	H (8)	76	8
QSLTFRTDE	Transferee TDE number	H (8)	84	8
QSLTRQDE	Requestor TDE number	H (8)	92	8
QSLHLDTCS	Holder TCS Address of TCS which holds the lock	H (8)	100	8
QSLCONTCS	Conflictor TCS Address of the TCS which is conflicting with the Requesting job/thread/TCS	H (8)	108	8
QSLTFRTCS	Transferee TCS Address of the TCS to which the lock is to be transferred	H (8)	116	8
QSLRQTCS	Requestor TCS Address of the TCS which is requesting the lock operation in question	H (8)	124	8
QSLTCS	Associated thread TCS Address of the Transaction Control Structure associated with the thread.	H (8)	132	8
QSLOBJ	Object address	H (8)	140	8
QSLOBJKEY	Object key	H (8)	148	8
QSLEVTSIG	Event signaled flag	B (4,0)	156	2
QSLTIME	Conflict time (in nanoseconds) Time base delta from when lock was requested and when lock was granted	B (18,0)	158	8
QSLMILNK	MI link register	H (8)	166	8
QSLLICLNK	LIC link register	H (8)	174	8

12.7.4.51 QAYPERUNI

This file provides 1 record about the general collection start and end times, system information and the definition used.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRNID	Collection Id	B (9,0)	1	4
QRNNM	Collection name	C (10)	5	10
QRNDSC	Collection description	C (72)	15	72

QRNVER	Collector version	B (4,0)	87	2
QRNMOD	Collection mode from DST	B (4,0)	89	2
QRNTCR	Collection created time	Z	91	26
QRNTSR	Collection start time	Z	117	26
QRNSTT	Collection start time in time base format	B (18,0)	143	8
QRNTSD	Collection start complete	Z	151	26
QRNTEN	Collection stop time	Z	177	26
QRNTED	Collection stop complete	Z	203	26
QRNTSS	Time suspended	Z	229	26
QRNTRZ	Time resumed	Z	255	26
QRNTTS	Total suspended time in microseconds	B (18,0)	281	8
QRNTRS	Reset time	Z	289	26
QRNEVT	Total events	B (9,0)	315	4
QRNWRP	Records overwritten due to wrap	B (9,0)	319	4
QRNDSZ	Total data size	B (18,0)	323	8
QRNPCR	Process creating collection	C (30)	331	30
QRNHTC	Task count creating collection	H (8)	361	8
QRNTNM	Task name creating collection	C (16)	369	16
QRNUSR	User who started collection	C (8)	385	8
QRNTSY	Target system name	C (8)	393	8
QRNSER	System serial #	C (8)	401	8
QRNTYP	System type	C (4)	409	4
QRNMDL	System model	C (4)	413	4
QRNSSY	System started from: remote or local	C (8)	417	8
QRNCN2	Conversion factor 2	P (10,6)	425	6
QRNTTC	Conversion factor for time base to TOD	B (18,0)	431	8
QRNDTC	Delta from time of day to wall clock time	B (18,0)	439	8
QRNPGS	Total pages of memory	B (18,0)	447	8
QRNLIC	LIC code level	C (3)	455	3
QRNXPF	XPF code level	C (3)	458	3

QRNSVR	System version/release/modification level	C (6)	461	6
QRNDIP	IPLs to DST	B (9,0)	467	4
QRNXIP	IPLs to XPF	B (9,0)	471	4
QRNASP	Configured ASPs	B (9,0)	475	4
QRNLDS	Configured logical DASD	B (9,0)	479	4
QRNDAR	Number of data areas	B (9,0)	483	4
QRNTEX	Number of tasks/processes examined by collector	B (9,0)	487	4
QRNTNO	# tasks/processes not added because already active	B (9,0)	491	4
QRNNOI	Collection id of task/processes not added	B (9,0)	495	4
QRNTAD	# of tasks/processes added at collection start	B (9,0)	499	4
QRNPAD	Number of processes added at collection start	B (9,0)	503	4
QRNTCT	Number of tasks/processes in the collection	B (9,0)	507	4
QRNPCT	Number of processes in the collection	B (9,0)	511	4
QRNTHC	Number of threads in collection	B (9,0)	515	4
QRNLIT	Number of LIC tasks in the collection	B (9,0)	519	4
QRNPFC	Processor feature code	B (9,0)	523	4
QRNMEC	Missed events	B (18,0)	527	8
QRNRES	Reserved	B (18,0)	535	8
QRNTCS	Total collector size	B (18,0)	543	8
QRNNOP	Maximum number of processors	B (4,0)	551	2
QRNAPS	Active processors at collection start	B (4,0)	553	2
QRNAPE	Active processors at collection end	B (4,0)	555	2
QRNLPI	Partition ID (deprecated)	C (2)	557	2
QRNPOF	Processor order feature	B (9,0)	559	4
QRNIOF	Interactive order feature	B (9,0)	563	4
QRNDBC	Total system database CPU microseconds	B (18,0)	567	8
QRNCPS	Total CPU microseconds for secondary workloads	B (18,0)	575	8
QRNFTR	Number of filtered events	B (9,0)	583	4
QRNOVF	Overflow counter	B (2,0)	587	2
QRNCR0	MMCR0 Register	B (18,0)	589	8

QRNCR1	MMCR1 Register	B (18,0)	597	8
QRNCRA	MMCRA Register	B (18,0)	605	8
QRNIMR	IMR or IMC register	B (18,0)	613	8
QRNTE	Table Entry Number	B (9,0)	621	4
QRNPRC	Processor mode	B (4,0)	625	2
QRNTHD	Processor multithreading	B (4,0)	627	2
QRNCAL	Calibration state	B (4,0)	629	2
QRNAFNSTT	Affinity balancer state 1=CHK 2=VFY 3=ANZ 4=CHGTSK	B (4,0)	631	2
QRNPMULVL	Performance Monitor Unit Level	B (4,0)	633	2
QRNEVT2	Event count 2	B (18,0)	635	8
QRNFTR2	Number of filtered events 2	B (18,0)	643	8
QRNDBSIZE	Database file member size	B (18,0)	651	8
QRNVRM	Database VRM	C (6)	659	6
QRNLVLF	File level	B (4,0)	665	2
QRNLVLD	Data level	B (4,0)	667	2
QRNFLAGS1	Flags 1	B (9,0)	669	4
QRNFLAGS2	Flags 2	B (9,0)	673	4
QRNRSVD01	Reserved	B (18,0)	677	8
QRNRSVD02	Reserved	B (18,0)	685	8
QRNRSVD03	Reserved	B (9,0)	693	4
QRNRSVD04	Reserved	B (9,0)	697	4
QRNRSVD05	Reserved	B (9,0)	701	4
QRNRSVD06	Reserved	B (9,0)	705	4
QRNRSVD07	Reserved	C (10)	709	10
QRNRSVD08	Reserved	C (10)	719	10
QRNRSVD09	Reserved	C (10)	729	10
QRNRSVD10	Reserved	C (10)	739	10
QRNEPV	Effective processor version	B (9,0)	749	4
QRNEPC	Effective processor compatibility	B (4,0)	753	2
QRNCURPAR	Current partition index	B (4,0)	755	2

QRNACTPAR	Partitions active	B (4,0)	757	2
QRNMOB	Mobility action	B (4,0)	759	2
QRNFLD	Processor folding action	B (18,0)	761	8
QRNPSTR	Processors at start	B (4,0)	769	2
QRNPEND	Processors at end	B (4,0)	771	2
QRNPFSTR	Proc fold status at start	B (4,0)	773	2
QRNPFEND	Proc fold status at end	B (4,0)	775	2
QRNPFSSTR	Proc fold switch at start	B (4,0)	777	2
QRNPFSEND	Proc fold switch at end	B (4,0)	779	2
QRNPSVSTR	Power save mode at start	C (4)	781	4
QRNPSVEND	Power save mode at end	C (4)	785	4
QRNPDLSTR	Power draw limit at start	C (8)	789	8
QRNPDLEND	Power draw limit at end	C (8)	797	8
>>> QRNCR2	MMCR2 register	B (18,0)	805	8 <<<

12.7.4.52 QAYPESAR

This file provides segment address register event details and is only included when the ADDPEXDFN command includes values on the SAREVT parameter.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QSRSPT	MI suspend point address	H (8)	9	8
QSRSPK	Key for MI suspend point	H (8)	17	8
QSRNIA	Next instruction address	H (8)	25	8
QSRNIK	Key for next instruction address	H (8)	33	8
QSRSAD	Range starting address	H (8)	41	8
QSRSOF	Range starting addr offset	B (9,0)	49	4
QSRPGS	Pages in the range	B (18,0)	53	8
QSRBYT	Bytes in the range	B (18,0)	61	8
QSRMPL	Main storage pool id	B (4,0)	69	2
QSRERS	Exchange range start address	H (8)	71	8

QSRERO	Exchange range start address offset	B (9,0)	79	4
QSREBY	Bytes in exchange	B (18,0)	83	8
QSRBSZ	Block size in bytes	B (9,0)	91	4
QSRPLR	Main storage buffer page list requested - Y or N	C (1)	95	1
QSRPNA	Pin action	B (9,0)	96	4
QSRSTL	Steal status	B (9,0)	100	4
QSRDEN	Changed data density	B (9,0)	104	4
QSRVAD	Virtual addr of page in which error was detected	H (8)	108	8
QSRIOC	I/O count for operation	B (18,0)	116	8
QSRREQ	Request type	B (4,0)	124	2
QSRERR	Error action type	B (4,0)	126	2
QSREXI	Detected exception id	B (4,0)	128	2
QSROPT	Operation type	B (4,0)	130	2
QSRRET	Related event entry count	B (9,0)	132	4
QSRFTC	Async I/O task count	H (8)	136	8
QOBJKEY	Object key	H (8)	144	8

12.7.4.53 QAYPESEGI

This file provide additional details about the objects and segments that were detected by various types of PEX trace events.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QSGSAD	Segment start address	H (8)	1	8
QSGEAD	Segment end address	H (8)	9	8
QSGTYP	Segment type	B (9,0)	17	4
QSGSZ	Seg size in 512 (64k if qsgbl = U or V) byte units	B (9,0)	21	4
QSGNFL	Segment new flags	H (1)	25	1
QSGFLG	Segment flag	H (1)	26	1
QSGASP	Segment ASP number	B (4,0)	27	2
QSGXSZ	Segment block transfer size	B (4,0)	29	2
QSGBL	Size attr: B=16MB L=64KB X=Teraspace U=256MB V=4GB	C (1)	31	1

QSGPT	Permanent or temporary: P or T	C (1)	32	1
QSGHLY	Holey segment: Y or N	C (1)	33	1
QSGRES	Resident segment: Y or N	C (1)	34	1
QSGIRG	In resident address range: Y or N	C (1)	35	1
QSGPRE	Preassigned segment: Y or N	C (1)	36	1
QSGAG	Access group segment: Y or N	C (1)	37	1
QSGMSD	Main store dump seg: Y or N	C (1)	38	1
QSGDIR	Directory segment: Y or N	C (1)	39	1
QSGCRI	Critical segment: Y or N	C (1)	40	1
QSGMAP	SID map segment: Y or N	C (1)	41	1
QSGIPL	Created in this IPL: Y or N	C (1)	42	1
QSGOVR	Segment overflowed: Y or N	C (1)	43	1
QSGREL	Segment is real: Y or N	C (1)	44	1
QSGDES	Segment destroyed: Y or N	C (1)	45	1
QSGDB	Database segment: Y or N	C (1)	46	1
QSGBSA	Segment base start address	H (8)	47	8
QSGBSZ	Base segment size (pages)	B (9,0)	55	4
QSGTY	Segment object type	B (9,0)	59	4
QSGSTY	Segment object subtype	B (9,0)	63	4
QSGONM	Object name	C (30)	67	30
QSGOCX	Object context (location)	C (30)	97	30
QSGOSZ	Obj size in 512 (64k if qsgbl = U or V) byte units	B (18,0)	127	8
QSGIFS	IFS data: Y or N	C (1)	135	1
QSPANM	IFS path name	G (514)	136	514
QOBJKEY	Object key	H (8)	650	8

12.7.4.54 QAYPESTATS

This file will only be included if the definition specified to collect PEX Stats data.

Field Name	Description	Attribute	Buffer Position	Buffer Length

QSTTCT	Task count (UNIQUE)	H (8)	1	8
QSTTBT	Traceback table address for the procedure	H (8)	9	8
QSTCMI	Complex MI index	B (4,0)	17	2
QSTNDE	Parent or child entry id STATS Flat Merge No - order of first call to procedure within task STATS Hier = defines call tree order in task	B (9,0)	19	4
QSTPAR	Parents id number (hierarchical mode)	B (9,0)	23	4
QSTCLV	Call level (hierarchical mode)	B (9,0)	27	4
QSTPTY	Procedure type: MI or LIC	C (1)	31	1
QSTINV	Procedure invocation count Number of times procedure called	B (18,0)	32	8
QSTCCT	Number of procedures called Number of procedure calls made by this procedure	B (18,0)	40	8
QSTXCT	MI complex instruction count Number of MI complex instruction calls made within this procedure	B (18,0)	48	8
QSTSTY	Event id: 2=MI proc, 6=MI, 10=java, 14=ntv method)	B (4,0)	56	2
QSTICPN	Inline CPU time in nanoseconds Inline procedure execution time (procedure only Pdc over head removed) (timebase format)	B (18,0)	58	8
QSTIETN	Inline elapsed time in nanoseconds	B (18,0)	66	8
QSWI01	Inline software counter #1	B (18,0)	74	8
QSWI02	Inline software counter #2	B (18,0)	82	8
QSWI03	Inline software counter #3	B (18,0)	90	8
QSWI04	Inline software counter #4	B (18,0)	98	8
QIISDR	Inline synchronous DB reads	B (9,0)	106	4
QIISNR	Inline synchronous non-DB reads	B (9,0)	110	4
QIISDW	Inline synch DB writes	B (9,0)	114	4
QIISNW	Inline synchronous non-DB writes	B (9,0)	118	4
QIIADR	Inline asynchronous DB reads	B (9,0)	122	4
QIIANR	Inline asynchronous non-DB reads	B (9,0)	126	4
QIIADW	Inline asynchronous DB writes	B (9,0)	130	4

QIIANW	Inline asynchronous non-DB writes	B (9,0)	134	4
QIIPWA	Inline I/O pending waits Inline total waits due to I/O operations	B (9,0)	138	4
QIISWA	Inline synchronous I/O waits Inline count of waits due to synchronous I/O operations	B (9,0)	142	4
QSTCCPN	Cumulative CPU time in nanoseconds Cumulative procedure execution time (procedure plus everything called by this procedure Pdc overhead removed) (timebase format)	B (18,0)	146	8
QSTCETN	Cumulative elapsed time in nanoseconds	B (18,0)	154	8
QSWC01	Cumulative software counter #1	B (18,0)	162	8
QSWC02	Cumulative software counter #2	B (18,0)	170	8
QSWC03	Cumulative software counter #3	B (18,0)	178	8
QSWC04	Cumulative software counter #4	B (18,0)	186	8
QCISDR	Cumulative synchronous DB reads	B (9,0)	194	4
QCISNR	Cumulative synchronous non-DB reads	B (9,0)	198	4
QCISDW	Cumulative synchronous DB writes	B (9,0)	202	4
QCISNW	Cumulative synchronous non-DB writes	B (9,0)	206	4
QCIADR	Cumulative asynchronous DB reads	B (9,0)	210	4
QCIANR	Cumulative asynchronous non-DB reads	B (9,0)	214	4
QCIADW	Cumulative asynchronous DB writes	B (9,0)	218	4
QCIANW	Cumulative asynchronous non-DB writes	B (9,0)	222	4
QCIPWA	Cumulative I/O pending waits Cumulative total waits due to I/O operations	B (9,0)	226	4
QCISWA	Cumulative synchronous I/O waits	B (9,0)	230	4
QSTSTS	Partial Count Status bit 0 - merge no indicator (used by PDC) bits 1 to 14 - reserved bit 15 - Partial count indicator: 0 - complete: procedure entered stack after collection started and left stack before collection ended 1 - partial: procedure entered stack before collection started or left stack after collected ended	C (1)	234	1
QSWI05	Inline software counter #5	B (18,0)	235	8
QSWI06	Inline software counter #6	B (18,0)	243	8

QSWI07	Inline software counter #7	B (18,0)	251	8
QSWI08	Inline software counter #8	B (18,0)	259	8
QSWC05	Cumulative software counter #5	B (18,0)	267	8
QSWC06	Cumulative software counter #6	B (18,0)	275	8
QSWC07	Cumulative software counter #7	B (18,0)	283	8
QSWC08	Cumulative software counter #8	B (18,0)	291	8
>>> QSIIC	Inline procedure hardware instructions	B (18,0)	299	8 <<<
>>> QSCIC	Cumulative procedure hardware instructions	B (18,0)	307	8 <<<
>>> QSIVTBC	Inline procedure execution cycles	B (18,0)	315	8 <<<
>>> QSCVTBC	Cumulative procedure execution cycles	B (18,0)	323	8 <<<

12.7.4.55 QAYPESYNC

This file is produced if the PEX trace collection includes synchronization event data which can be captured using the ADDPEXDFN command SYNCEVT parameter.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QSYETM	Wait time (in microseconds) TOD format: Bit 48 = 8 microsecond resolution	B (18,0)	9	8
QSYRTC	Return code	B (9,0)	17	4
QSYMI1	MI suspend point code address 1	H (8)	21	8
QSYMK1	Key for MI suspend point address 1	H (8)	29	8
QSYMI2	MI suspend point code address 2	H (8)	37	8
QSYMK2	Key for MI suspend point address 3	H (8)	45	8
QSYMI3	MI suspend point code address 3	H (8)	53	8
QSYMK3	Key for MI suspend point address 3	H (8)	61	8
QSYMI4	MI suspend point code address 4	H (8)	69	8
QSYMK4	Key for MI suspend point address 4	H (8)	77	8
QSYLSO	LIC mutex, condition, semaphore or token ***PTRMTXLOCK, *PTRMTXUNLOCK, *HDLMTXWAIT, *HDLMTXRLS, *MTXCLEANUP** SLIC mutex address	H (8)	85	8

	***CONDWAIT, *CONDSET** SLIC condition address ***PTRSEMWAIT, *PTRSEMPOST, *NAMSEMWAIT, *NAMSEMPOST** SLIC semaphore address ***TXNLOCK, *TCNUNLOCK** SLIC token address			
QSYUSO	User mutex, semaphore or token address ***PTRMTXLOCK, *PTRMTXUNLOCK, *MTXCLEANUP** User mutex address ***PTRSEMWAIT, *PTRSEMPOST, *NAMSEMWAIT, *NAMSEMPOST** User semaphore address ***TXNLOCK, *TCNUNLOCK** User token address	H (8)	93	8
QSYMXT	***PTRMTXLOCK, *HDLMTXWAIT** TDE address of mutex holder ***CONDWAIT** TDE address of last condition setter ***PTRSEMWAIT, *NAMSEMWAIT** TDE address of last semaphore poster ***TXNLOCK** TDE address of token holder ***MTXCLEANUP** TDE address of mutex creator	H (8)	101	8
QSYSPI	Indicator: 0 = set 1 = pulse	C (1)	109	1
QSYMTXNM	Mutex name	C (16)	110	16

12.7.4.56 QAYPETASKI

This file provides details about the jobs/tasks/threads collected by PEX. Depending on the value provided on the ADDPEXDFN command LSTALLJOB parameter the data will either include only the jobs/tasks specified on the JOB or TASK parameter or will include all jobs/tasks on the system during the collection regardless on if they were captured elsewhere in the collection or not.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QTSTCT	Task count (UNIQUE)	H (8)	1	8
QTSADR	Task address	H (8)	9	8
QTSNM	Task name	C (16)	17	16

QTSFTC	Initial thread task count	H (8)	33	8
QTSID	Task id in hex	H (4)	41	4
QTSPL	Task pool id	B (4,0)	45	2
QTSRES	Task resident flag: Y or N	C (1)	47	1
QTSMI	MI task flag: Y or N	C (1)	48	1
QTSPRI	Initial task priority	B (4,0)	49	2
QTSTSL	Task time slice	B (18,0)	51	8
QTSAST	Task active (clock) microseconds at last start	B (18,0)	59	8
QTSASP	Task active (clock) microseconds at last stop	B (18,0)	67	8
QTSRST	Task running (CPU) microseconds at last start	B (18,0)	75	8
QTSRSP	Task running (CPU) microseconds at last stop	B (18,0)	83	8
QTSXST	Task existed at start: Y or N	C (1)	91	1
QTSXSP	Task existed at stop: Y or N	C (1)	92	1
QTSJNM	Process job name	C (10)	93	10
QTSJUS	Job user	C (10)	103	10
QTSJNB	Job number	C (6)	113	6
QTSPID	Partition id	B (4,0)	119	2
QTSITF	Initial thread flag Y/N	C (1)	121	1
QTSTHI	Thread identifier	H (8)	122	8
QTSAP1	Accumulated Run Cycles	B (18,0)	130	8
QTSAP3	Accumulated Run Instructions	B (18,0)	138	8
QTSWRT	Task writes	B (18,0)	146	8
QTSPWR	Task permanent writes	B (18,0)	154	8
QTSSDR	Task synchronous DB reads	B (18,0)	162	8
QTSSNR	Task synchronous non-DB reads	B (18,0)	170	8
QTSSDW	Task synchronous DB writes	B (18,0)	178	8
QTSSNW	Task synchronous non-DB writes	B (18,0)	186	8
QTSADB	Task asynchronous DB reads	B (18,0)	194	8
QTSANR	Task asynchronous non-DB reads	B (18,0)	202	8
QTSADW	Task asynchronous DB writes	B (18,0)	210	8

QTSANW	Task asynchronous non-DB writes	B (18,0)	218	8
QTSPWA	Task I/O pending waits	B (18,0)	226	8
QTSSWA	Task synchronous I/O waits	B (18,0)	234	8
QTSDBC	Reserved	B (18,0)	242	8
QTSSTA	Storage pages allocated	B (18,0)	250	8
QTSSTD	Storage pages deallocated	B (18,0)	258	8
QTSPGF	Database and non-db page faults	B (18,0)	266	8
QTSOHD	Collection overhead in CPU microseconds	B (18,0)	274	8
QTSACT	Task active (clock) time	B (18,0)	282	8
QTSRUN	Task running (CPU) time	B (18,0)	290	8
QTSADDS	Times added to collection	B (9,0)	298	4
QTSJVTHD	Java Thread Name	G (514)	302	514
QTSINCOL	Task in collection: Y or N	C (1)	816	1
QTSHNODE	Home node	B (4,0)	817	2
QTSAFLVL	Affinity level	B (4,0)	819	2
QTSAFID	Affinity identifier	B (9,0)	821	4
QTSPRAFID	Preferred affinity resource identifier	B (4,0)	825	2
QTSSRST	Scaled running (CPU) microseconds at last start	B (18,0)	827	8
QTSSRSP	Scaled running (CPU) microseconds at last stop	B (18,0)	835	8
QTSSRUN	Scaled running (CPU) time	B (18,0)	843	8
QTSSBS	Subsystem name	C (10)	851	10
QTSPMC1A	Accumulated PMC 1	B (18,0)	861	8
QTSPMC2A	Accumulated PMC 2	B (18,0)	869	8
QTSPMC3A	Accumulated PMC 3	B (18,0)	877	8
QTSPMC4A	Accumulated PMC 4	B (18,0)	885	8
QTSPMC5A	Accumulated PMC 5	B (18,0)	893	8
QTSPMC6A	Accumulated PMC 6	B (18,0)	901	8
QTSTHDNM	Thread name	C (16)	909	16
QTSWLCGRPS	Workload group ID at start	B (9,0)	925	4
QTSWLCGRPE	Workload group ID at end	B (9,0)	929	4

>>> QTSPROCRES	Processor resources priority	B (4,0)	933	2 <<<
>>> QTSIC	Instruction count	B (18,0)	935	8 <<<
>>> QTSICOVR	Instruction count overhead	B (18,0)	943	8 <<<
>>> QTSVTBC	Virtual time base cycles	B (18,0)	951	8 <<<
>>> QTSVTBCOVR	Virtual time base cycles overhead	B (18,0)	959	8 <<<
QTSRSVD01	Reserved	B (18,0)	967	8
QTSRSVD02	Reserved	B (18,0)	975	8
QTSRSVD03	Reserved	B (9,0)	983	4
QTSRSVD04	Reserved	B (9,0)	987	4
QTSRSVD05	Reserved	B (9,0)	991	4
QTSRSVD06	Reserved	B (9,0)	995	4
QTSRSVD07	Reserved	C (10)	999	10
QTSRSVD08	Reserved	C (10)	1009	10

12.7.4.57 QAYPETBRKT

This file provides trace job style bracketing event data. It is only included in the collection when the ADDPEXDFN command PGMEVT parameter includes *PRCEXIT or *PRCENTRY. These events provide similar functionality to what is used/provided by the TRCJOB command.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QTBIAD	Instruction address	H (8)	9	8
QTBTBT	Procedure traceback table address	H (8)	17	8
QTBHLL	HLL statement number	B (9,0)	25	4
QTBCIA	Caller instruction address	H (8)	29	8
QTBCTB	Caller traceback table address	H (8)	37	8
QTBCHL	Caller HLL statement	B (9,0)	45	4
QTBFCN	Function ID	B (4,0)	49	2
QTBEVM	Thread event mask 0=events allowed 1=not allowed	C (1)	51	1
QTBCLL	Invocation call level	B (9,0)	52	4
QTBSDR	Synchronous DB reads	B (18,0)	56	8

QTBSNR	Synchronous non-DB reads	B (18,0)	64	8
QTBSDW	Synchronous DB writes	B (18,0)	72	8
QTBSNW	Synchronous non-DB writes	B (18,0)	80	8
QTBADR	Asynchronous DB reads	B (18,0)	88	8
QTBANR	Asynchronous non-DB reads	B (18,0)	96	8
QTBADW	Asynchronous DB writes	B (18,0)	104	8
QTBANW	Asynchronous non-DB writes	B (18,0)	112	8
QTBPWA	I/O pending waits	B (18,0)	120	8
QTBSWA	Synchronous I/O waits	B (18,0)	128	8

12.7.4.58 QAYPETIDX

This file is used to identify the information needed for every type of PEX trace event captured such as the type of event, when it occurred and the job/task/thread responsible for causing it. One record will always be generated for each trace event.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QTITY	Event type Join to QEVTY in QAYPEEVENT for event descriptions.	B (4,0)	9	2
QTISTY	Event subtype Join to QEVSTY in QAYPEEVENT for event descriptions.	B (4,0)	11	2
QTITSP	Time of day timestamp	Z	13	26
QTITIMN	Nanoseconds since the collection started	B (18,0)	39	8
QTIECY	Task execution cycles	B (18,0)	47	8
QTIFTC	Task count	H (8)	55	8
QTIPRN	Processor id of the currently active processor	B (4,0)	63	2
QTIFLAGS	Flags	B (4,0)	65	2
QTITIMB	Hypervisor timebase	B (18,0)	67	8
>>> QMISSEDEVT	Missed event count Events missed prior to this one	B (4,0)	75	2 <<<
>>> QHVLPRCIDX	Hypervisor logical processor index	B (4,0)	77	2 <<<

>>> QINSTCNT	Accumulated instruction count in this task	B (18,0)	79	8 <<<
>>> QVTBC	Virtual timebase cycles Accumulated virtual timebase cycles in this task	B (18,0)	87	8 <<<

12.7.4.59 QAYPETSKSW

This file contains task switch event data. It is collected by using one or more *TASKSWT* or *CPUSWT values on the ADDPEXDFN command BASEVT parameter on the PEX definition.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QTWIAD	Instruction address ***TASKSWTIN** Address of a task ***TASKWTOUT, *TASKSWTOUTINT** Instruction address that caused the switch out ***TASKSWTOUTQ, *TASKAVAIL** Caller address ***CPUSWT** Address corresponding to the yield	H (8)	9	8
QTWIAK	Key for instruction address	H (8)	17	8
QTWWOA	***TASKSWTOUTQ, *TASKAVAIL** Wait object address ***CPUSWT** Faulting phyp AMS address	H (8)	25	8
QTWWDN	Wait object description - numeric portion	B (4,0)	33	2
QTWWDC	Wait object description - character portion	C (3)	35	3
QTWWOR	Wait object reason ***TASKSWTOUTQ, *TASKAVAIL** This is the specific type of wait that occurred (i.e. enum.)	B (9,0)	38	4
QTWPTY	Apparent task priority	B (9,0)	42	4
QTWTAD	***TASKSWTIN, *TASKSWTOUTQ, *TASKSWTOUTINT, *TASKAVAIL** Task address ***CPUSWT** Converted virtual page address of phyp AMS faulting address	H (8)	46	8

QTWTWC	***TASKSWTIN** Task wait cycles ***CPUSWT** Time base cycles between Waiting and enqueued to dispatcher.	B (18,0)	54	8
QTWTSC	***TASKAVAIL** Task sleep cycles ***CPUSWT** Time base cycles between enqueued to dispatcher and executing	B (18,0)	62	8
QTWCTC	***TASKSWTIN (with 7.1 MF50194 or higher)** Task workload capping latency cycles ***TASKAVAIL** Task count causing AFD	H (8)	70	8
QTWHPI	Hypervisor processor ID	B (9,0)	78	4
QTWUNP	Reserved	B (9,0)	82	4
QTWIM	Reserved	B (9,0)	86	4
QTWYT	***CPUSWT** Trace log buffer index	B (4,0)	90	2
QTWSIP	Reserved	B (9,0)	92	4
QTWHRC	***CPUSWT** Hypervisor CPU switch in reason code 0: The virtual processor was dispatched at the external interrupt vector location to handle an IOA interrupt, Virtual interrupt, or interprocessor interrupt. 1: The virtual processor was dispatched to handle firmware internal events. 2: The virtual processor was dispatched at the next sequential instruction due to an H_PROD call by another partition processor. 3: The virtual processor was dispatched at the DECR interrupt vector due to a decrementer interrupt. 4: The processor was dispatched at location specified in load module (boot) or at the system reset interrupt vector. (virtual yellow button). 5: The virtual processor was dispatched to handle firmware internal events 6: The virtual processor was dispatched at the next sequential instruction to use cycles conferred from another partition processor 7: The virtual processor was dispatched at the next sequential instruction for its entitled time slice. 8: The virtual processor was dispatched at the faulting instruction following a virtual partition memory page fault	B (4,0)	96	2

QTWHNODE	Home physical processor node ID	B (4,0)	98	2
QTWMSR	***CPUSWT** Copy of MSR bits 0:32, 37:41, and 48:63 at time of preemption	H (8)	100	8
QTWTRCB	***CPUSWT** Hypervisor preempt reason code 0: Not used (for compatibility with earlier versions of the facility) 1: Firmware internal event 2: Virtual processor called H_CEDE 3: Virtual processor called H_CONFER 4: Virtual processor reached the end of its timeslice (HDEC) 5: Partition migration/hibernation page fault 6: Virtual/real memory page fault	B (9,0)	108	4
QTWBLXCOD	***TASKSWTOUTQ** Task block exit code (added in v5r4m0) 0=TDNoBlockExit 1=TDIpcfBlockExit 2=TDIocmBlockExit 3=TDVioBlockExit ***TASKAVAIL** wait-object reason (QuBlockRC) 0x00 = QuUnblocked 0x01 = QuInterrupt 0x02 = QuMiInterrupt 0x03 = QuPartialMatch or QuRetry 0x04 = QuDestroyed 0x05 = QuIncoherent 0x06 = QuAtomicRetry 0x07 = QuAborted 0xFE = QuNotDispatched 0xFF = QuAvailable	B (4,0)	112	2
QOBJKEY	Object key	H (8)	114	8

Notes on *CPUSWT event:

1. QTWTWC = xDeltaToEnqueued + xDeltaToReadyToRun
2. QTIFTC = the TDE field in this event does NOT correspond to the task that was running with the logical processor was preempted. This is the TDE that happened to be active

when the hypervisor trace buffers were unloaded and the corresponding PDC events were generated.

3. QTITSP = the timestamp on this event IS the time when the preemption/block of the partition logical processor occurred. This is NOT the time that hypervisor trace buffers were unload and the PDC events written.

The CPU Dispatch event records information when the hypervisor goes through the process of removing/redispatching a virtual processor on a physical processor. The CPU Dispatch event is recorded when the processor is dispatched (switched in) to the partition. The reason the processor was preempted (switched out) of the partition is presented in the event as the preempt reason code. The reason the processor was dispatched (switched in) is presented in the event as the dispatch reason code.
Explanation of hypervisor preempt reason codes:
0: Not used
1: Firmware internal event - example would be firmware detected a processor failure.
2: Processor called H_CEDE - normal flow when the partition gives up a processor when nothing to dispatch.
3: Processor called H_CONFER - the partition gives up the processor time for this virtual processor and confers those cycles to another virtual processor in the same partition. An example would be when one VP to wake up or give cycles to another processor.
4: Processor reached the end of its time slice (H_DEC) - end of entitlement.
5: Partition migration/hibernation page fault - page fault during mobility action
6: Advance memory sharing page fault - page fault that occurs on a pure virtual partition that belong to a shared memory pool.

Explanation of hypervisor dispatch reason codes:
0: Processor dispatched to handle a virtual IO or real external interrupt.
1: Processor dispatched to handle partition to partition communication.
2: Processor dispatched to handle H_PROD call by another processor - back end of H_CONFER preempt reason.
3: Processor dispatched due to a decrementer interrupt.
4: Processor dispatched for IPL.
5: Processor dispatched to handle firmware internal events - like machine check.
6: Processor dispatched to use cycles conferred from another processor.
7: Processor dispatched for its entitled time slice - start of entitlement.
8: Processor dispatched when a partition page fault is complete.

Example:

Sequence	VP # xHvProcIndex	Proc Preempt Reason xPreemptReason	Proc Dispatch Reason xDispReason	
	1		2	VP1 dispatched because VP0 did an H_CONFER
	0	3	6	VP0 yielded waiting for lock held by VP1.

	1	4	7	VP1

Definition of dispatch reasons:

- **Preempt** - virtual processor is being dispatched after returning from a preempt condition. It is now allowed to consume more processing units
- **Timeout** - virtual processor is being dispatched to handle the expiration of a timer
- **External interrupt** - virtual processor is being dispatched to handle an external interrupt condition
- **LpProd** - virtual processor is being dispatched due to a request from the logical partition to force it to run
- **LpEvent** - virtual processor is being dispatched due to another Logical partition signaling an event to the virtual processor in this partition
- **Fault** - virtual processor is being dispatched due to the completion of a page fault
- **IPL** - virtual processor is being dispatched to perform an initial program load within the partition
- **Terminate** - virtual processor is being dispatched to handle the termination of the partition
- **Unblocked** - virtual processor is being dispatched due to the releasing of a block condition

12.7.4.60 QAYPEUSRDF

This file is typically used by IBM service only. The *SERVICE events in each category will write data to this file (where implemented.)

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QUSDTA	Data field Typically this will contain the following: Bytes 1 - 8: B (18,0): Format ID Bytes 9 – 16: : B (18,0): Resolution info Bytes 17+: C(*) Client data	C (502)	9	502

12.7.4.61 QAYPEVIO

This file is produced when virtual I/O events (ADDPEXDFN command parameter VRTIOEVT) are included in the PEX definition used for the collection.

Note: The contents of some fields will vary depending on the specific event type collected.

Field Name	Description	Attribute	Buffer Position	Buffer Length
QRECN	Record number (UNIQUE)	B (18,0)	1	8
QVILUN	Logical unit number	H (8)	9	8
QVITAG	Command tag	H (8)	17	8
QVIMTAG	Managed tag ***SCSITSKMGMT** Tag of managed command	H (8)	25	8
QVIBUS	Adapter bus number	B (4,0)	33	2
QVIBOARD	Adapter board number	B (4,0)	35	2
QVICARD	Adapter card number	B (4,0)	37	2
QVIUNIT	Adapter unit address	B (9,0)	39	4
QVIEVTQ	Event qualifier ***SCSISERVER** 1 = SRP Request 2 = SRP Response 3 = MAD Request 4 = MAD Response 5 = Initialization or Transport Event ***SCSICLIENT** 1 = Outgoing SRP Request 2 = Incoming SRP Response 3 = SRP Request queued, CRQ full 4 = SRP Request queued, Server busy 5 = SRP Request queued, Previous cmd pending 6 = SRP Request aborted 7 = Incoming event, no IU 8 = Outgoing event, no IU 9 = SRP Request queued, Insufficient resources ***SCSICMD** 1 = CmdRouted 2 = CmdStarted 3 = CmdComplete ***SCSITSKMGMT** 1 = Request Received (iSCSI only) 2 = Cmd Started 3 = Cmd Complete ***ISCSI** 1 = Request Received 2 = Response Sent	B (4,0)	43	2

	***DISKSTR** 1 = Asynchronous Read 2 = Asynchronous Write 3 = Asynchronous Read For Write 4 = Synchronous Read 5 = Synchronous Clear 6 = Synchronous Write ***OPTSTR** 0x0001 = Read 0x0002 = Write 0x0003 = Request Key 0x0004 = Release Key 0x0005 = Import Media 0x0006 = Export Media 0xFExx = SCSI Data Request Op xx 0xFFxx = SCSI Pipe Op xx ***TAPSTR** 0x0001 = Read 0x0002 = Write 0x0003 = Read Block Limits 0x0004 = Rewind 0x0005 = Clear Tape 0x0006 = Space Block Forward 0x0007 = Space Block Backward 0x0008 = Space File Forward 0x0009 = Space File Backward 0x000A = Space End of Data 0x000B = Write Tape Mark 0x000C = Write Buffer 0x000D = Retrieve Header 0x000E = Read Position 0x000F = Set Position 0xFExx = SCSI Pipe Op xx (cached) 0xFFxx = SCSI Pipe Op xx (passthru) ***ETHADPT** 1 = Transmit outbound frame 2 = Receive inbound frame 3 = Transmit to internal switch 4 = Receive from internal switch 5 = Dropped frame 6 = Link status			
QVIFUN	Function ***SCSITSKMGMT** Task management function 0x01 = Abort Task 0x02 = Abort Task Set 0x03 = Clear ACA 0x04 = Clear Task Set 0x05 = Logical Unit Reset 0xFF = Invalid Request	B (4,0)	45	2
QVISTS	Status ***SCSICMD, *ISCSI**	B (4,0)	47	2

	0x00 = Good 0x02 = CheckCondition 0x04 = ConditionGood 0x08 = Busy 0x10 = Intermediate 0x18 = ReservationConflict 0x28 = TaskSetFull 0x30 = AcaActive 0x40 = TaskAborted ***SCSITSKMGMT** 0x00 = Success 0x01 = Failure 0x02 = Not Supported			
QVIATTR	Task attribute ***SCSICMD, *ISCSI** 0x00 = HeadOfQueue 0x01 = Ordered 0x02 = Simple	B (4,0)	49	2
QVISARSTS	SAR status	H (8)	51	8
QVILBA	Logical block address	H (8)	59	8
QVIBLOCKS	Number of blocks	B (18,0)	67	8
QVILENGTH	Length ***TAPSTR** Command-specific transfer length	B (18,0)	75	8
QVISARADR0	SAR 0 address	H (8)	83	8
QVISARADR1	SAR 1 address	H (8)	91	8
QVISARADR2	SAR 2 address	H (8)	99	8
QVISARLEN0	SAR 0 length	H (8)	107	8
QVISARLEN1	SAR 1 length	H (8)	115	8
QVISARLEN2	SAR 2 length	H (8)	123	8
QVIDATA	Miscellaneous data ***SCSISERVER** 256 bytes - SCSI SRP Information Unit (SRP/MAD) -OR- 16 bytes - CRQ Entry (Init/Transport Event) ***SCSICLIENT** 256 bytes - SCSI SRP Information Unit (SRP) -OR- 16 bytes - CRQ Entry (No IU) ***SCSICMD** Command Descriptor Block (CDB) ***ISCSI**	C (258)	131	258

	16 bytes - Command Descriptor Block (CDB) (requests) -OR- 16 bytes - SCSI Sense Data (responses) ***ETHADPT** First 128 bytes of frame			
QVIDTALEN	Data length ***ETHADPT** Total frame length	B (9,0)	389	4
QVIRESCNT	Residual count	B (18,0)	393	8
QVIOPEN	Open indicator ***TAPSTR** 0x00 = Closed 0x01 = Open 0xFF = Unknown	B (4,0)	401	2
QVITOKEN	Media token ***OPTSTR** Optical volume token	H (8)	403	8
QVIKEY	Key ***OPTSTR** Optical volume key	H (8)	411	8
QVIRC	Return code ***ETHADPT** Frame or link status Special values for Dropped frames: 0xC0DE4110xxxxxxxx = Reason code xxxxxxxx 0xBADDEED000000001 = No Client 0xBADDEED000000002 = No Filter 0xBADDEED000000003 = No Transmit Port 0xBADDEED000000004 = Bad Frame Routing 0xBADDEED000000005 = No Connection 0xBADDEED000000006 = Bad MAC Address 0xBADDEED000000007 = Bad Port State 0xBADDEED000000008 = Bad Checksum 0xBADDEED000000009 = Frame Too Big 0xBADDEED000000006 = No Space Available 0xBADDEED000000006 = Could Not Map Frame Data Special values for Link Status: 0xC8C40001 = Link Up 0xC8C40002 = Link Down 0xC8C40006 = Shutdown	B (18,0)	419	8
QVISIOARC	SIOA return code	H (8)	427	8