

IBM iDoctor for IBM i Job Watcher

IBM iDoctor for IBM i Development Team

10 May 2022

Licensed Materials - Property of IBM

© Copyright International Business Machines Corporation 2022. All rights reserved.

Abstract

Provides in-depth coverage of all major GUI functions for all components at 7.2 and higher. This document covers the Job Watcher component.

Changes

10 May 2022 – Updated for client 1540 – 7.5 GA

20 Apr 2022 – Updated for client 1536

24 Feb 2022 – added SQL compression fields back in to Interval Summary – SQL statistics

14 Feb 2022 – updated for 2022

Table of Contents

1	Introduction.....	12
2	Starting Job Watcher.....	14
3	Job Watcher Component View	15
3.1	Root Folder Menu Options	15
4	Definitions	18
4.1	Add Job Watcher Definition Wizard	18
4.1.1	Welcome.....	18
4.1.2	Basic Options	19
4.1.3	Data Collection Options.....	21
4.1.4	Advanced Options	30
4.1.5	Job Options	30
4.1.6	Job/task selection	32
4.1.7	Finish	36
4.2	Properties	37
5	Start Job Watcher Collection Wizard	38
5.1	Welcome	38
5.2	Basic Options	38
5.3	Scheduling Options	39
5.4	Termination	40
5.5	Finish.....	41
6	Libraries Folder.....	43
6.1	Menu Options	43
7	Monitors	44
8	SQL Tables.....	45
9	Collections	46
9.1	Menu Options	47
9.2	Run ALL Default Analyses	49
9.3	Graph Jobs.....	50
9.4	Search.....	51
9.5	Generate Reports.....	54
9.6	Split	55
9.7	Stop.....	55
9.8	Properties.....	55
9.8.1	General.....	56

9.8.2	Creation Settings	56
9.8.3	Definition.....	57
9.8.4	Wait Buckets.....	58
9.8.5	Situations.....	58
9.8.6	System.....	59
9.8.7	LPAR CPU.....	60
9.8.8	Collection File Locks	60
10	Analyses	61
10.1	Analyze Collection Window	61
10.1.1	Job Watcher Situations Window.....	62
10.1.2	Job Watcher Situations Editor	63
10.2	Call Stack Summary.....	64
10.2.1	SQL Tables.....	65
10.3	Change sensitive user data	66
10.3.1	SQL Tables.....	67
10.4	Collection Summary	67
10.4.1	Additional graphs.....	67
10.4.2	Graphing multiple collections	69
10.4.3	Intra-component drill downs	69
10.4.4	SQL Tables.....	70
10.5	Collection Summary by TDE type.....	71
10.5.1	SQL Tables.....	71
10.6	Destroy all host variable data in QAPYJWSQLH.....	71
10.7	Job Summary	72
10.7.1	Additional graphs.....	73
10.7.2	Selected job(s) over time graphs.....	74
10.7.3	Rankings filtered by <selected job(s)>	75
10.7.4	Job Summary rankings filtered by <X>	75
10.7.5	SQL Tables.....	77
10.8	Lock Trace.....	78
10.8.1	SQL Tables.....	78
10.9	Long Transactions.....	78
10.9.1	SQL Tables.....	78
10.10	Situational Analysis	79
10.10.1	SQL Tables.....	79
10.11	Modules Waiting.....	79
10.11.1	SQL Tables.....	79
11	Holder menu	81
11.1	Holder chase	81

12	Graph notes	82
12.1	CPU metrics	82
12.2	“Interesting” wait buckets.....	83
12.3	Tips / FLAGS.....	84
13	Overview graphs	86
13.1	Drilling down to Rankings.....	86
13.2	Drilling down to Detail reports.....	87
13.3	Split Collection option.....	88
14	Ranking graphs	89
14.1	Drilling down to Selection over time	90
15	Selection over time graphs.....	91
15.1	Drilling up into Rankings.....	91
15.2	Display call stack.....	92
16	Favorites	93
17	Waits	94
17.1	Collection overview time signature	94
17.2	Collection overview time signature with workload capping	96
17.3	Collection overview time signature with dispatched CPU breakdown	96
17.4	Virtual CPU delays	97
17.5	Clients + Workers wait time	97
17.6	Collection overview time signature with max waits in-progress	98
17.7	Current wait duration time signature with max waits in-progress.....	99
17.8	Disk time signature with max disk waits in-progress	99
17.9	Current wait duration time signature.....	100
17.10	Seizes and locks time signature	100
17.11	Disk time signature	100
17.12	Journaling time signature	101
17.13	Communications time signature	101
17.14	Time waiting on objects	102
17.15	CPU queueing breakdown.....	103
17.16	Workload capping delays	104
17.17	Wait counts.....	104
17.18	Average wait times	105
17.19	Average CPU times	105
17.20	Average disk wait times.....	105

17.21	Dispatched CPU rankings	106
17.21.1	Dispatched CPU rankings by thread.....	106
17.21.2	Dispatched CPU rankings by job	107
17.21.3	Dispatched CPU rankings by job user	107
17.21.4	Dispatched CPU rankings by generic job	108
17.21.5	Dispatched CPU rankings by current user	109
17.21.6	Dispatched CPU rankings by pool	109
17.21.7	Dispatched CPU rankings by priority	110
17.21.8	Dispatched CPU rankings by subsystem	110
17.21.9	Dispatched CPU rankings by job type	111
17.21.10	Dispatched CPU rankings by job function	112
17.21.11	Dispatched CPU rankings by generic job current user	112
17.21.12	Dispatched CPU rankings by thread current user	113
17.21.13	Dispatched CPU rankings by QRO hash.....	114
17.22	Clients + Workers wait time rankings	114
17.23	Clients only wait time rankings	115
17.24	CPU queueing breakdown rankings	116
17.25	Workload capping delay rankings.....	117
17.26	Time waiting on objects rankings	117
17.27	Wait counts rankings	118
17.28	Average wait times rankings.....	119
17.29	Average CPU times rankings	119
17.30	Average disk time rankings	120
17.31	Collection totals	120
17.31.1	Collection overview time signature	121
17.31.2	Seizes and locks time signature	122
17.31.3	Disk time signature	122
17.31.4	Communications time signature	122
17.32	Objects waited on	123
17.32.1	Objects waited on for all waits of interest	124
17.32.2	Objects waited on due to page faulting	124
17.32.3	Objects waited on for all waits of interest by job/thread	125
17.32.4	Objects waited on due to page faulting by job/thread	125
17.32.5	Objects waited on for all waits of interest by job/thread, SQL statement	125
17.32.6	Objects waited on due to page faulting by job/thread, SQL statement	126
18	CPU	127
18.1	CPU consumption	127
18.2	CPU utilization.....	127

18.3	Dispatched CPU breakdown and CPUQ	128
18.4	Dispatched CPU/CPUq usage by high/low priority	128
18.5	Dispatched CPU/CPUq usage by high/low priority with CPU utilization	129
18.6	Active processors	129
18.7	Dispatched CPU rankings	130
18.8	CPU utilization rankings	130
18.9	CPU consumption rankings	131
19	Job counts	133
19.1	Job counts	133
19.2	Jobs submitted	134
19.3	Job counts rankings	134
19.3.1	Job counts by job	134
19.3.2	Job counts by job user	135
19.3.3	Job counts by generic job	135
19.3.4	Job counts by current user	136
19.3.5	Job counts by pool	136
19.3.6	Job counts by priority	137
19.3.7	Job counts by subsystem	137
19.3.8	Job counts by job type	138
19.3.9	Job counts by job function	138
19.3.10	Job counts by generic job current user	139
19.3.11	Job counts by job current user	139
19.3.12	Job counts by QRO hash	140
20	Temporary storage	141
20.1	Job temporary storage allocations	141
20.2	Running total new active job current temp storage	142
20.3	Net active job current temp storage	142
20.4	Net active job current temp storage for QZD* jobs	143
20.5	Net active job current temp storage for QZD* jobs TOP 10 current user breakdown	143
20.6	Net active job current temp storage for QZD* jobs current user breakdown	144
20.7	Job temporary storage rankings	144
20.8	Average/peak/max temporary storage rankings	145
20.9	Net active job current temp storage rankings	146
21	Page allocations	148
21.1	Memory page demand	148
21.2	Memory page space allocations	149

21.3	Net page frames requested	149
21.4	Temporary pages allocated/deallocated.....	150
21.5	Pages allocated/deallocated.....	151
21.6	Net pages allocated.....	151
21.7	Net perm/temp pages allocated.....	151
21.8	Net temporary storage pages allocated.....	152
21.9	Pages marked easy to steal	152
21.10	Memory page demand rankings	153
21.11	Memory page space allocations rankings	154
21.12	Net page frames requested rankings	155
21.13	Temporary pages allocated/deallocated rankings	156
21.14	Pages allocated/deallocated rankings	156
21.15	Net pages allocated rankings	157
21.16	Net perm/temp pages allocated rankings	157
21.17	Pages marked easy to steal rankings.....	158
22	I/O	159
22.1	Read and writes totals	159
22.2	Read and writes rates	160
22.3	Physical I/O activity totals.....	160
22.4	Physical I/O activity rates	161
22.5	Physical I/O activity totals with synchronous percentage	161
22.6	Physical I/O activity rates with synchronous percentage	162
22.7	Page fault totals.....	163
22.8	Page fault rates	163
22.9	Synchronous reads and writes	164
22.10	Synchronous reads and writes with avg/max/in-progress response times	164
22.11	Average synchronous read response	165
22.12	Average synchronous write response	165
22.13	Maximum synchronous read response.....	166
22.14	Maximum synchronous write response	166
22.15	Reads and writes totals rankings.....	167
22.16	Reads and writes rates rankings	168
22.17	Physical I/O activity totals rankings	168
22.18	Physical I/O activity rates rankings.....	169
22.19	Page fault totals rankings	169
22.20	Page fault rates rankings.....	170

22.21	Synchronous reads and writes rankings.....	170
22.22	Average synchronous read response rankings	171
22.23	Average synchronous write response rankings	171
22.24	Maximum synchronous read response rankings	172
22.25	Maximum synchronous write response rankings.....	173
22.26	Collection totals	173
22.26.1	Pages allocated/deallocated.....	174
22.26.2	Reads and writes rates	174
22.26.3	Physical I/O activity rates.....	175
23	Logical I/O.....	176
23.1	Logical database I/O totals.....	176
23.2	Logical database I/O rates.....	177
23.3	Advanced logical database I/O totals	177
23.4	Advanced logical database I/O rates.....	178
23.5	SQL logical database I/O totals	178
23.6	SQL logical database I/O rates.....	178
23.7	Non-SQL logical database I/O totals	179
23.8	Non-SQL logical database I/O rates.....	179
23.9	All logical database I/O totals	179
23.10	All logical database I/O rates.....	180
23.11	Logical database FEODs, commit and rollbacks	180
23.12	Logical database index rebuilds and sorts.....	180
23.13	Logical database I/O totals rankings	181
23.14	Logical database I/O rates rankings	182
23.15	Advanced logical database I/O totals rankings.....	182
23.16	Advanced logical database I/O rates rankings	182
23.17	SQL logical database I/O totals rankings.....	183
23.18	SQL logical database I/O rates rankings	183
23.19	Non-SQL logical database I/O totals rankings.....	183
23.20	Non-SQL logical database I/O rates rankings	184
23.21	All logical database I/O totals rankings.....	184
23.22	All logical database I/O rates rankings	184
23.23	Logical database FEODs, commits and rollbacks rankings	184
23.24	Logical database index rebuilds and sorts rankings	185
24	IFS	186
24.1	IFS lookup cache totals	186

24.2	IFS lookup cache rates	187
24.3	IFS opens totals	187
24.4	IFS opens rates	187
24.5	IFS reads totals	187
24.6	IFS reads rates	188
24.7	IFS creates/deletes totals	188
24.8	IFS create/deletes rates	188
24.9	IFS rankings	188
25	JVM	190
25.1	JVM collection wait buckets	191
25.2	JVM heap sizes	193
25.3	JVM collection wait buckets by Thread	193
25.4	JVM heap sizes by job	194
26	Top consumers	195
27	Long Transactions	198
27.1	Long transactions for DB server jobs	198
27.2	Long transactions for all jobs	199
28	Call stack summary	200
29	Opens	202
29.1	Total full opens	202
29.2	Full opens summary	202
29.3	Programs causing full opens	203
29.4	Programs/procedures causing full opens	203
29.5	Programs/procedures/jobs causing full opens	204
29.6	Jobs causing full opens	205
29.7	Jobs causing full opens with 14 levels of program names	206
29.8	Total closes	207
30	SQL	208
30.1	SQL statements executed	208
30.2	SQL statements captured	209
30.3	Full opens rates	210
30.4	Full opens totals	210
30.5	Pseudo opens rates	211
30.6	Pseudo opens totals	211
30.7	QZDA* connections	211

30.8	SQL statements executed rankings.....	212
30.9	SQL statements captured rankings	213
30.10	Full opens rates rankings	214
30.11	Full opens totals rankings.....	214
30.12	Psuedo opens rates rankings	214
30.13	Psuedo opens totals rankings	215
30.14	QZDA* connections by job	215
30.15	Detail reports	216
31	Communications	218
32	Other metrics.....	219
32.1	5250 transaction totals	219
32.2	5250 transaction response times.....	219
32.3	Spool files created	220
32.4	5250 transaction totals rankings.....	221
32.5	5250 transaction response times rankings	222
32.6	Spool files created rankings	222
33	Interval Summary Interface	223
33.1	Quick View	223
33.2	General section	223
33.3	Waits	224
33.4	Wait bucket totals	226
33.5	Objects waited on.....	226
33.6	Holders.....	228
33.7	SQL statistics	229
33.8	Bad Current Waits	229
33.9	Situations.....	230
33.10	Physical disk I/Os	231
33.11	Logical DB	232
33.12	IFS.....	233
33.13	Other statistics.....	234
34	Interval Details Interface.....	236
34.1	General Section.....	236
34.2	Call Stack	237
34.2.1	Display Full Procedure Name.....	239
34.2.2	Call Stack reports	240
34.3	Waits	243

34.4	Object Waited on	244
34.5	Physical disk I/Os	244
34.6	Logical DB	245
34.7	IFS.....	246
34.8	J9 JVM	247
34.9	SQL / Client.....	248
34.10	Other statistics.....	249
35	Data repository	251
36	JVM analysis.....	252
36.1	Analyze JVM	252

1 Introduction

This document provides an overview of the interfaces as well as the green screen commands and database files for the Job Watcher component of iDoctor.

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 interface provided by IBM). These are two different products and purchased and supported separately within IBM. Buying one does not provide a license to the other.

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 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
- J9 JVM statistics

Job Watcher is available for trial evaluation or purchase via this website and is sold via Lab Services.

A license for Job Watcher includes:

- Job Watcher software (licensed by system serial number via an access code)
- Collection Services Investigator
- Disk Watcher
- Plan Cache Analyzer
- 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).

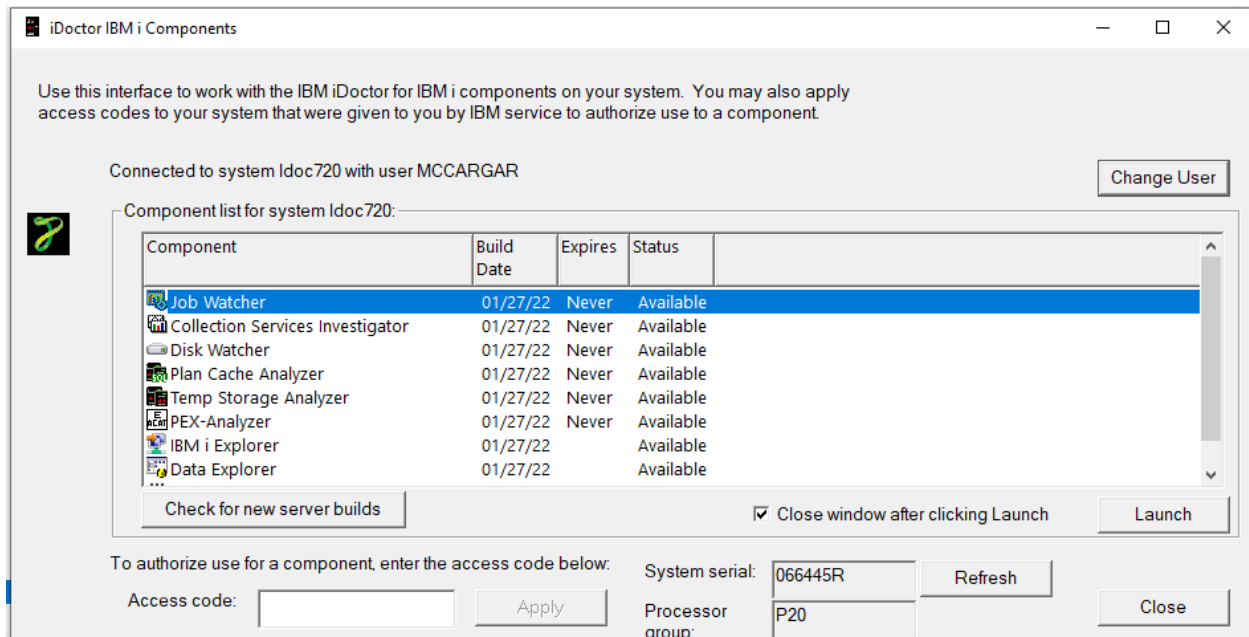
Data is collected in Job Watcher using commands that are included with IBM i which are:

- ADDJWDFN – Adds a Job Watcher definition to the system
- STRJW – Starts a Job Watcher collection
- ENDJW – Ends an active Job Watcher collection

2 Starting Job Watcher

Job Watcher is a component of the iDoctor suite of tools. After launching iDoctor, the Job Watcher component is started from the IBM i Connections List View by double-clicking on the desired system.

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 to continue



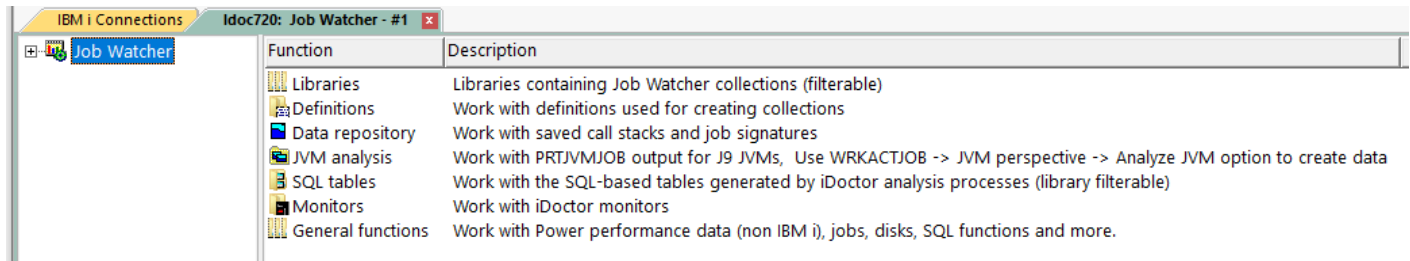
iDoctor IBM i Components Window

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.

3 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 to show Job Watcher collections under a monitor.

This also provides options for working with the Job Watcher [Definitions](#) that exist on the system. These are used for defining the aspects for what data is collected. Several IBM-supplied definitions exist, or the user can make their own.



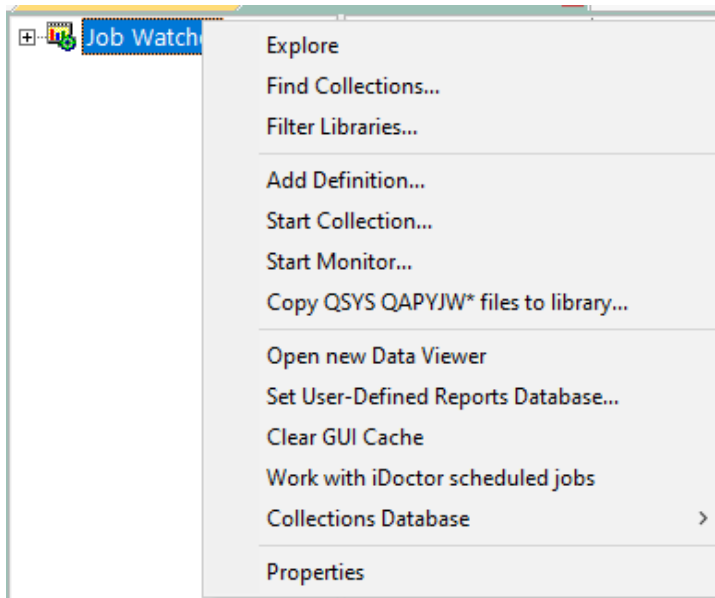
Job Watcher Component View

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) 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.

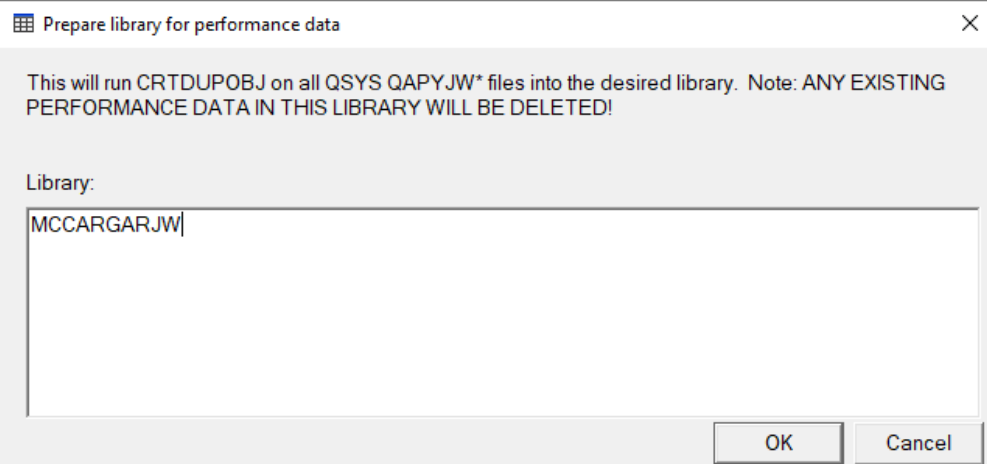
These folders are covered in more detail in the next sections.

3.1 Root Folder Menu Options

The following Job Watcher specific menu options are available by right clicking on the 'Job Watcher' icon in the component view:



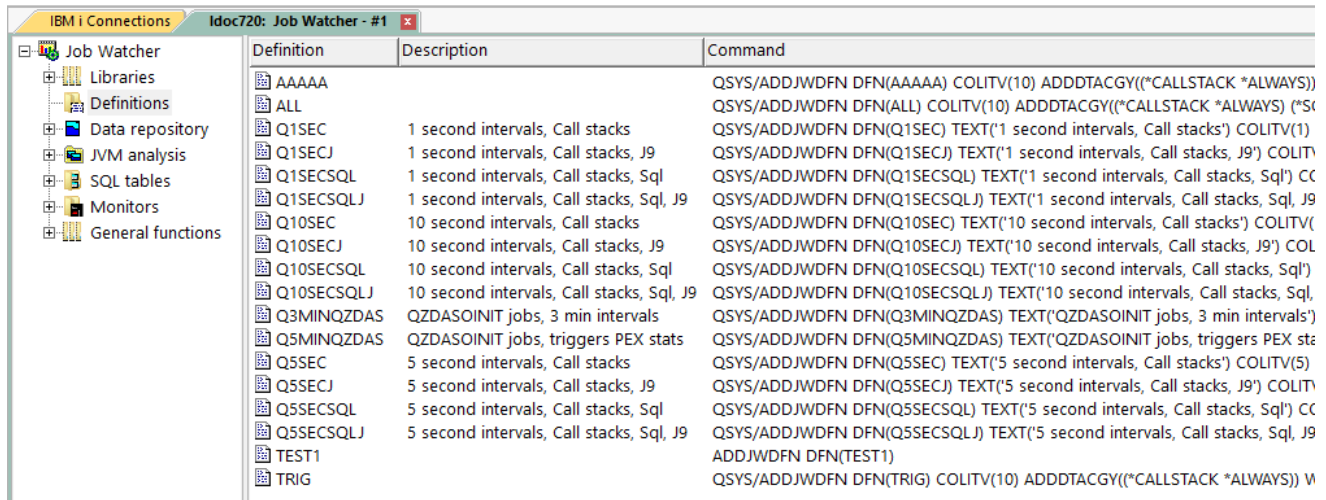
Job Watcher Pop-up-Menu

Menu Item	Description
Find Collections...	This option displays the Find Collections interface which provides the ability to look for collections matching user-defined characteristics. Example SQL statements are provided. The results of these queries are available under the General Functions -> Find collections results folder.
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. Note: A definition is required when starting a collection.
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.
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.
Copy QSYS QAPYJW* files to library...	This option can be used to copy all QAPYJW* files from QSYS to the desired library. This is an optional step that prepares a library for collecting performance data. It also allows the library to appear under the Libraries folder in Job Watcher.  Prepare library for performance data window Note: The reason this exists, is in some situations where previous release DB files are used to collect performance data on the current release, then various problems will occur.
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 GUI cache	This option clears everything loaded in the GUI's cache (like menus, graph definitions, query definitions, stored procedure versions installed, etc)
Work with iDoctor	This option is a shortcut to the General functions -> Work management ->

scheduled jobs	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. It also is used to improve performance when browsing collections on the system.
Properties	Use this menu to display version information for the current component. The build level of the GUI is also displayed here.

4 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:



Definition	Description	Command
AAAAA		QSYS/ADDJWDFN DFN(AAAAA) COLITV(10) ADDDTACGY((*CALLSTACK *ALWAYS))
ALL		QSYS/ADDJWDFN DFN(ALL) COLITV(10) ADDDTACGY((*CALLSTACK *ALWAYS)) (*S
Q1SEC	1 second intervals, Call stacks	QSYS/ADDJWDFN DFN(Q1SEC) TEXT('1 second intervals, Call stacks') COLITV(1)
Q1SECJ	1 second intervals, Call stacks, J9	QSYS/ADDJWDFN DFN(Q1SECJ) TEXT('1 second intervals, Call stacks, J9') COLITV
Q1SECSQL	1 second intervals, Call stacks, Sql	QSYS/ADDJWDFN DFN(Q1SECSQL) TEXT('1 second intervals, Call stacks, Sql') CC
Q1SECSQLJ	1 second intervals, Call stacks, Sql, J9	QSYS/ADDJWDFN DFN(Q1SECSQLJ) TEXT('1 second intervals, Call stacks, Sql, J9
Q10SEC	10 second intervals, Call stacks	QSYS/ADDJWDFN DFN(Q10SEC) TEXT('10 second intervals, Call stacks') COLITV(
Q10SECJ	10 second intervals, Call stacks, J9	QSYS/ADDJWDFN DFN(Q10SECJ) TEXT('10 second intervals, Call stacks, J9') COL
Q10SECSQL	10 second intervals, Call stacks, Sql	QSYS/ADDJWDFN DFN(Q10SECSQL) TEXT('10 second intervals, Call stacks, Sql') CC
Q10SECSQLJ	10 second intervals, Call stacks, Sql, J9	QSYS/ADDJWDFN DFN(Q10SECSQLJ) TEXT('10 second intervals, Call stacks, Sql,
Q3MINQZDAS	QZDASOINIT jobs, 3 min intervals	QSYS/ADDJWDFN DFN(Q3MINQZDAS) TEXT('QZDASOINIT jobs, 3 min intervals')
Q5MINQZDAS	QZDASOINIT jobs, triggers PEX stats	QSYS/ADDJWDFN DFN(Q5MINQZDAS) TEXT('QZDASOINIT jobs, triggers PEX sta
Q5SEC	5 second intervals, Call stacks	QSYS/ADDJWDFN DFN(Q5SEC) TEXT('5 second intervals, Call stacks') COLITV(5)
Q5SECJ	5 second intervals, Call stacks, J9	QSYS/ADDJWDFN DFN(Q5SECJ) TEXT('5 second intervals, Call stacks, J9') COLITV
Q5SECSQL	5 second intervals, Call stacks, Sql	QSYS/ADDJWDFN DFN(Q5SECSQL) TEXT('5 second intervals, Call stacks, Sql') CC
Q5SECSQLJ	5 second intervals, Call stacks, Sql, J9	QSYS/ADDJWDFN DFN(Q5SECSQLJ) TEXT('5 second intervals, Call stacks, Sql, J9
TEST1		ADDJWDFN DFN(TEST1)
TRIG		QSYS/ADDJWDFN DFN(TRIG) COLITV(10) ADDDTACGY((*CALLSTACK *ALWAYS)) V

Job Watcher Definitions Folder

Note: Right-clicking the Definitions folder provides an option to rebuild the IBM-supplied definitions which is sometimes needed if none of the Q* definition names appear.

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. Note: This is only applicable to definitions that aren't named Q*.
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 New 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.

4.1 Add Job Watcher Definition Wizard

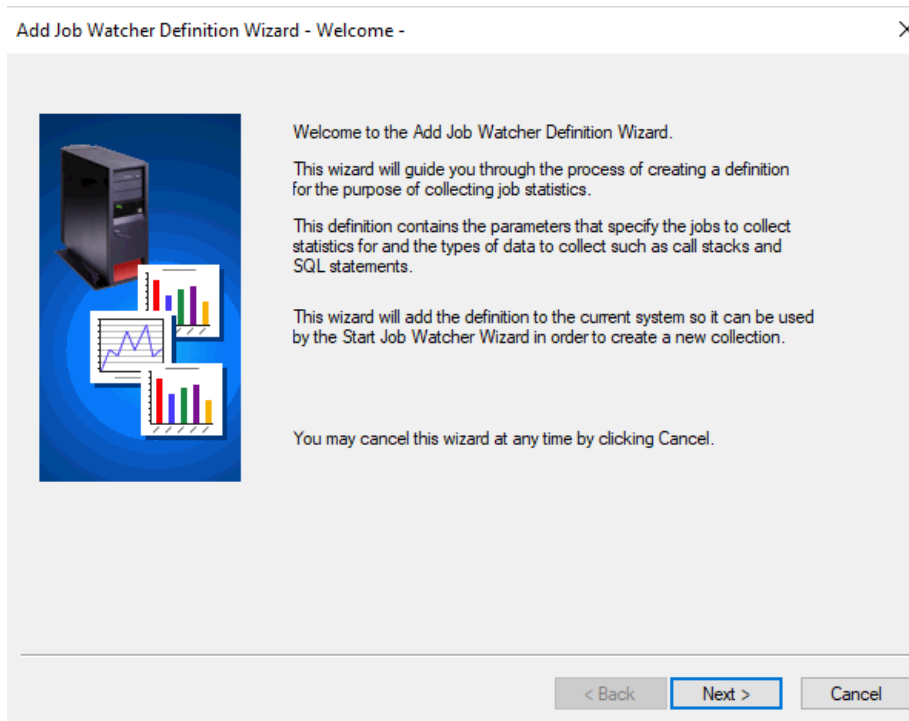
Use this interface to create or change a user-defined Job Watcher definition. This option is accessible by right clicking the Job Watcher root folder and using the Add Definition... menu.

It can also be accessed using the Add Definition menu for a definition.

Note: This is an interface over the IBM i command ADDJWDFN.

4.1.1 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

4.1.2 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 -

Specify the definition name and other optional parameters below.

Definition name:

Description:

Interval duration: 0.1 - 3,600.0 seconds

☐ Collect as fast as possible

Data collection options:

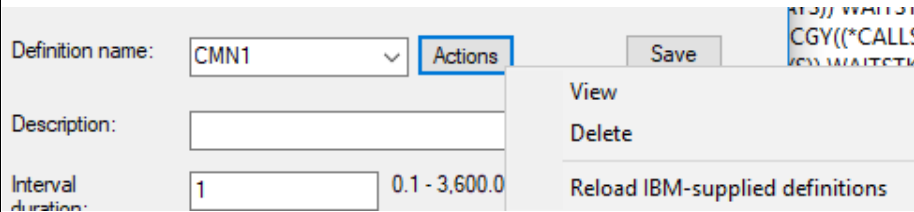
Call stacks: Every interval
Collect for jobs in conflict for at least 1 microseconds
Collect for jobs in bad waits for at least 1 microseconds

☐ Show advanced options

< Back **Next >** Cancel

Add Job Watcher Definition Wizard – Basic Options

The following table describes some of the less obvious options on this screen:

Option	Description
Definition name	The definition name will be added as a new member name in file QAPYJWDFN in QUSRSYS. This cannot start with Q when using this interface.
Actions button	Click this button to display a menu of options relating to Job Watcher definitions:  <i>Actions button menu options</i> 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.
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!

4.1.3 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.

4.1.3.1 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 Collection Options



Activation Groups	Sockets/Communications	Condition Control
Call Stack	SQL	IBM Technology for Java

Indicate below how call stacks should be collected. The first option will only collect jobs that used some CPU within each interval.

Call stack collection frequency (for jobs/tasks using CPU):

☒ Every interval
☐ Never
☐ Only every Nth interval

The second option is used to collect call stacks for jobs in a bad wait. This option only collects call stacks for jobs that have previously used CPU during the course of the job watch unless the "collect idle threads on 1st interval" option is used.

Call stack collection for jobs/tasks in "bad wait" scenarios

☒ Collect the call stack only for jobs in conflict with other jobs for at least N microseconds
 N: 1 - 2147483647

☒ Collect the call stack only for jobs in a "bad wait" (where no conflict with another job exists) for at least N microseconds
 N: 1 - 2147483647

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.

4.1.3.2 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 Collection Options

Activation Groups Sockets/Communications Condition Control

Call Stack SQL IBM Technology for Java

Indicate below the type of SQL information that should be collected and how often it should be collected.

SQL data to include:

- ☒ None
- ☐ Active SQL statements and host variables
- ☐ Active or last executed SQL statements and host variables
- ☐ Active or last executed SQL statements, host variables, prepared statement arrays and open cursors

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.

4.1.3.3 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 Collection Options

Activation Groups Sockets/Communications Condition Control

Call Stack SQL IBM Technology for Java

Indicate below if any Java specific information for jobs running the new 32-bit Java Virtual Machine (JVM) in IBM i5/OS should be collected.

IBM Technology for Java Virtual Machine data to include:

☐ None

☒ JVM statistics and thread information

☐ JVM statistics, thread information and call stacks

Collection frequency:

☒ Every interval

☐ Only every Nth interval

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.

4.1.3.4 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.

4.1.3.5 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 Collection Options

Call Stack SQL IBM Technology for Java

Activation Groups Sockets/Communications Condition Control

Indicate below whether or not socket communications data should be collected:

Sockets data collection frequency:

☐ Never

☒ Every interval

☐ Only every Nth interval

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

4.1.3.6 Condition Control

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 Collection Options

Call Stack	SQL	IBM Technology for Java
Activation Groups	Sockets/Communications	Condition Control

These options are used to collect data, call a program, or even wait to collect any data until the conditions are met. A conditional control definition contains the list of conditions to check for.

☒ Use a conditional control definition

Conditions file (QAIDRJWRD):

Library: Member:

Description:

Loaded conditions SAMPLETRIG successfully.

File contents:

PERCENT(QTIME05).GT.20

Options:

Mode: ☐ Per interval ☒ Trigger ☐ Collect until met

Trigger timeout:

Consecutive intervals needed to cause condition:

Trigger history number of intervals

☐ Call exit program

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. <u>Per interval</u> 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. <u>Trigger</u> 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 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. <u>Collect until met</u> 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.

4.1.4 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.

4.1.5 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 - Idoc720

Indicate below whether to collect all jobs and/or tasks or to select specific jobs and tasks.

Job/task selection:

- ☒ Include all jobs and tasks
- ☐ Include all jobs
- ☐ Include all tasks
- ☐ Select specific jobs and tasks

☐ Collect idle jobs/tasks on 1st interval

Allows job/task names that are idle throughout the entire collection to be visible in the graphs.

< Back Next > Cancel

Add Job Watcher Definition Wizard – Job Options

The following table describes the parameters available on this page of the Wizard.

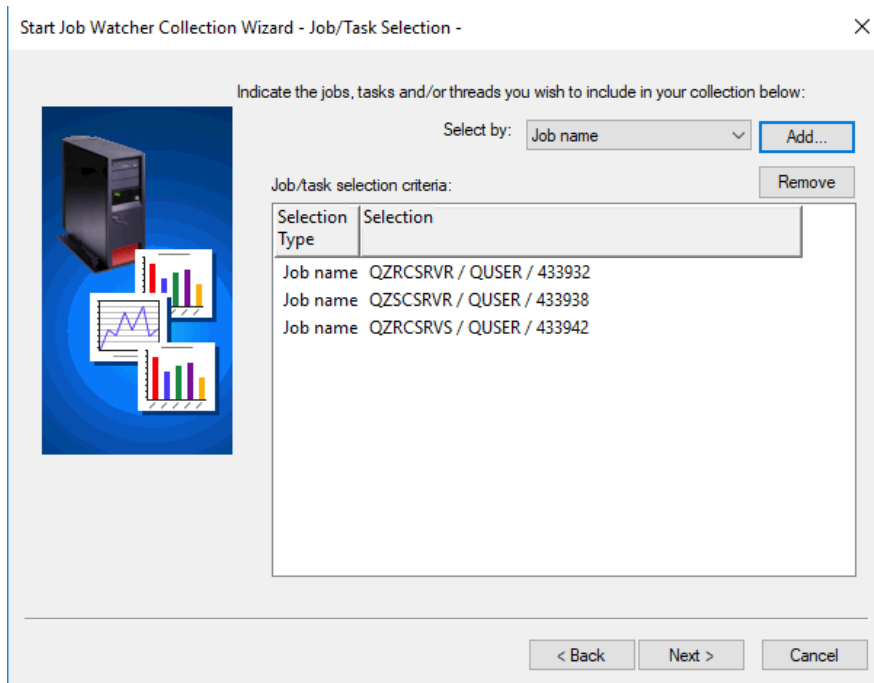
Note: Active jobs/tasks are defined as those jobs or tasks that used the CPU for each interval collected.

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.

4.1.6 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

4.1.6.1 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 Jobs

Indicate the jobs to include in your collection below:

Job Filter Information:

Name: Number: Status:

User: Current user:

Jobs matching the job filter information:

Subsystem	Job Name	User	Number	CPU utilization (%)	CPU time total (ms)	Status	Function	Current User	Enter
	QUSRWRK QZRCRSVS	QUSER	497509	.6	34	RUN		MCCARGAR	201
	QSERVER QZDAINIT	QUSER	433924	0	30	PSRW		QUSER	201
	QCMN QZRCRSVR	QUSER	433932	0	18	PSRW		QUSER	201
	QCMN QZSCSRVR	QUSER	433938	0	14	PSRW		QUSER	201
	QUSRWRK QZRCRSVS	QUSER	433942	0	13	TIMW		QSECOFR	201
	QUSRWRK QZHQSRSV	QUSER	433944	0	20	TIMW		QSECOFR	201
	QUSRWRK QZDASSINIT	QUSER	433948	0	34	PSRW		QUSER	201
	QSERVER QZLSFILE	QUSER	433950	0	29	PSRW		QUSER	201
	QSYSWRK QZBSEVTM	QUSER	433949	0	6	EVTW	PGM-QZBSEVTM	QUSER	201
	QUSRWRK QZRCRSVS	QUSER	434088	0	6	TIMW		QSECOFR	201
	QSYSWRK QZSCSRVSD	QUSER	434124	0	148	SELW		QUSER	201
	QSYSWRK QZHQSRSV	QUSER	434125	0	146	SELW		QUSER	201
	QSYSWRK QZRCRSVSD	QUSER	434127	0	712	SELW		QUSER	201

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.

4.1.6.2 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.

4.1.6.3 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.

4.1.6.4 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.

4.1.6.5 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

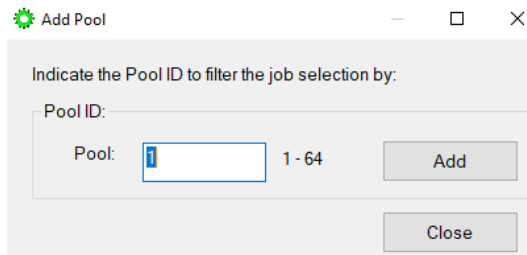
This window displays a list of subsystems that are running on the system to select from. By selecting a subsystem this indicates that all jobs that running in that subsystem will be included in the collection (if not filtered out by other parameters which may also be used).

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.

4.1.6.6 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.

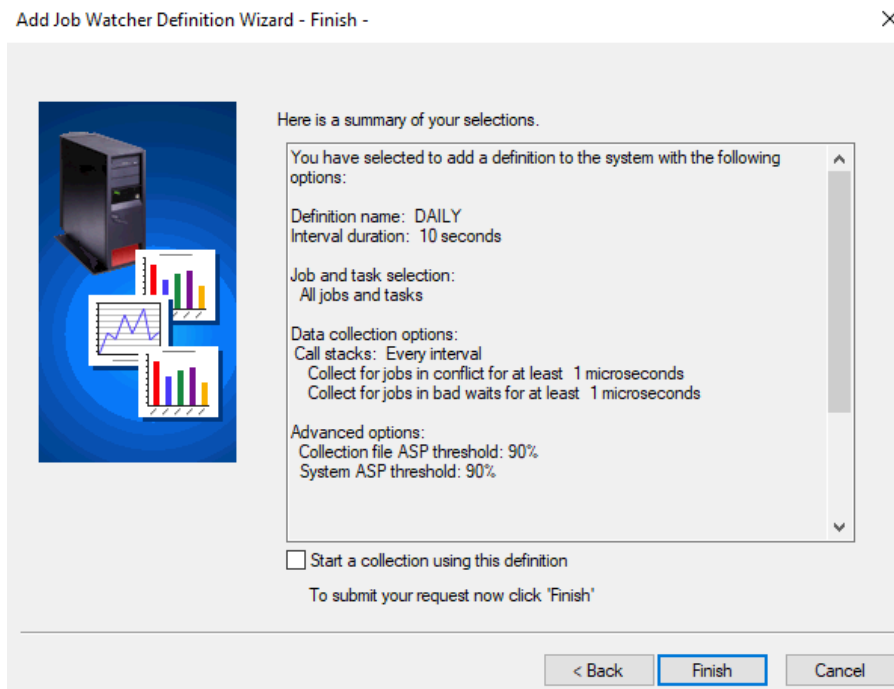
The following table describes the fields on this window:

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.

4.1.7 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:

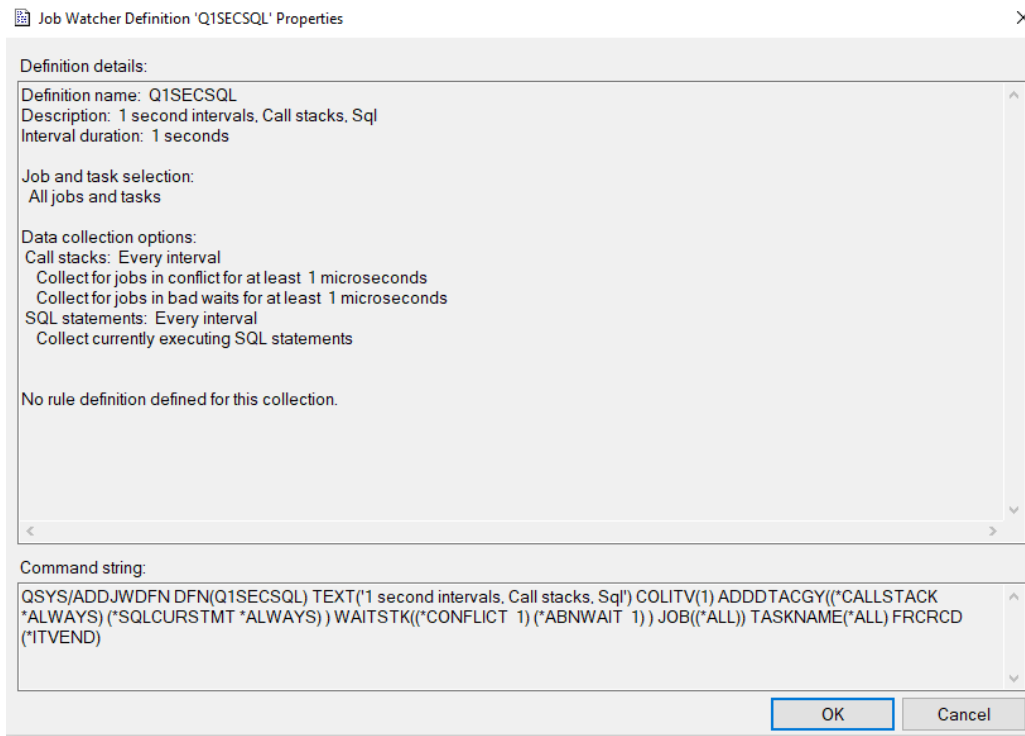


Add Job Watcher Definition Wizard – Finish

Note: Click the “Start a collection using this definition” option to launch the [Start Job Watcher Collection Wizard](#) right after the definition is added to the system.

4.2 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.

5 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 detail 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.

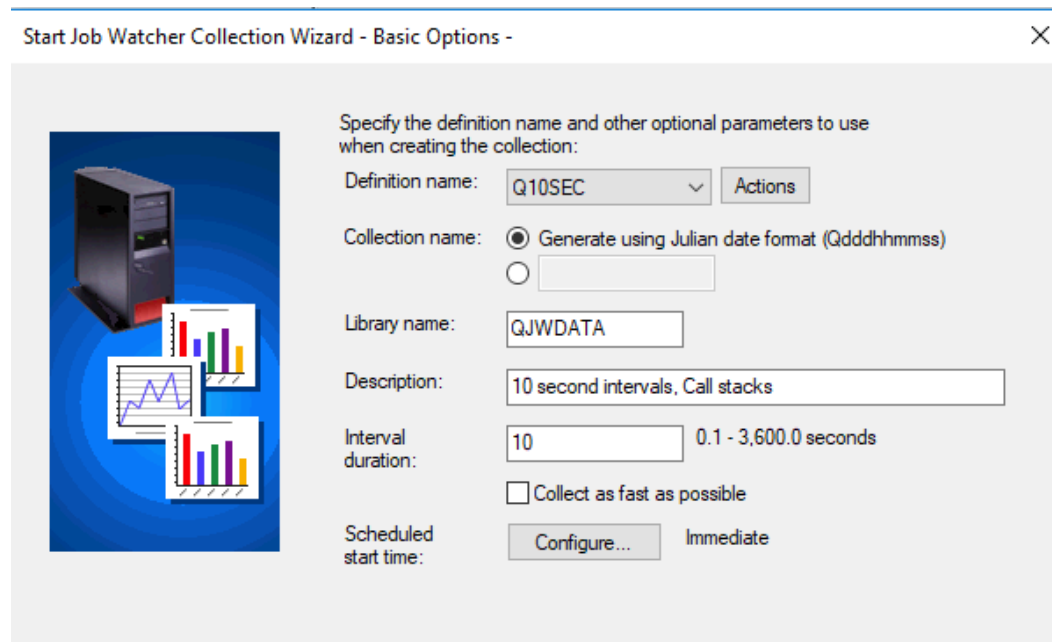
5.1 Welcome

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.

5.2 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:

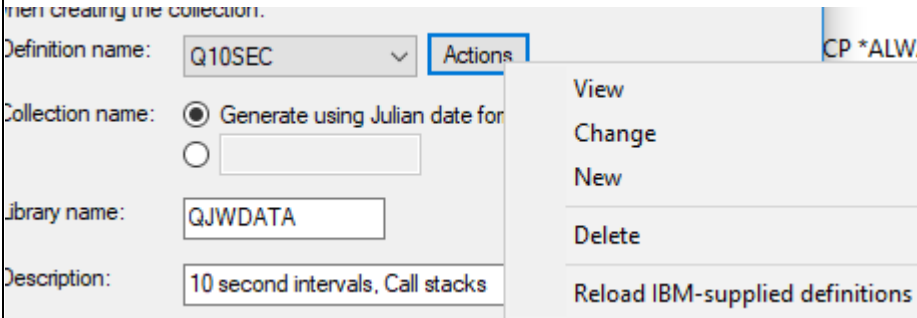


The screenshot shows the 'Start Job Watcher Collection Wizard - Basic Options' dialog box. On the left is a graphic of a server tower and charts. The main area contains the following fields and options:

- Definition name:** A dropdown menu showing 'Q10SEC' and an 'Actions' button.
- Collection name:** Two radio buttons. The first is selected and labeled 'Generate using Julian date format (Qdddhmmss)'. The second is an empty text box.
- Library name:** A text box containing 'QJWDATA'.
- Description:** A text box containing '10 second intervals, Call stacks'.
- Interval duration:** A text box containing '10' and a label '0.1 - 3,600.0 seconds'.
- Scheduling:** A checkbox labeled 'Collect as fast as possible' which is currently unchecked.
- Scheduled start time:** A 'Configure...' button and the text 'Immediate'.

Start Job Watcher Collection Wizard – Basic Options

Some of the less obvious options are described below:

Field	Description
Actions button	Click this button to display a menu of options relating to Job Watcher definitions:  <i>Actions button menu options</i> 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.
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!

5.3 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:

 Schedule collection start time ✕

Use this interface to schedule an action for a later time.

☒ Schedule the collection start time

Note: Date and time values are based on the server's clock, not your PC's clock.

Frequency: Once ▼

Scheduled date:

◀ December 2018 ▶

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

 Today: 12/3/2018

Current (system) time: 2:24:13 PM Idoc720

Scheduled collection start time: 2:15:06 PM

OK
Cancel

Start Job Watcher Collection Wizard – Schedule Collection Start Time

5.4 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 ✕



Indicate below how the collection should end. At least one and up to all three of these options may be specified. The collection will end when one of the specified criteria has been met.

Options (select one or more)

☒ Maximum disk space to consume 1 GB ▼

☐ Maximum intervals to collect 100

☒ Maximum time to collect 1 hours ▼

Start Job Watcher Collection Wizard – Termination

5.5 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 of this page and can be copied to a green screen session and modified if necessary.

After the collection is started it will take several seconds before anything appears in the GUI while the collection is being initialized. Use F5 to refresh the list of collection in the collection library to work with the new collection.

Start Job Watcher Collection Wizard - Finish - ✕



Here is a summary of your selections.

[Submit job options](#)

You have selected to start a collection with the following options:

Definition name: Q1SEC
Collection name: Automatically generate
Library name: QJWDATA
Description: 1 second intervals, Call stacks
Interval duration: 1 seconds

Termination options:
Maximum disk space: 1 gigabytes
Maximum time: 1 hours

Remote Command String:

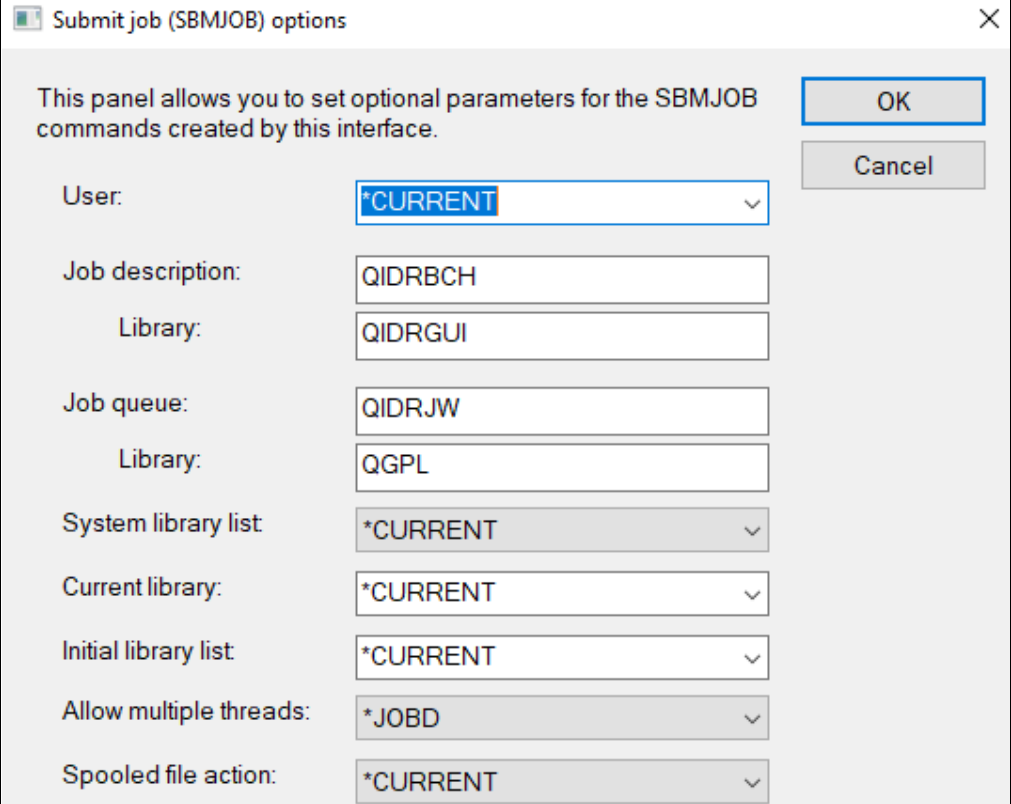
```
QSYS/SBMJOB CMD(QSYS/STRJW DFN(Q1SEC) COL("GEN) LIB
(QJWDATA) TEXT("1 second intervals, Call stacks") RPLDTA("YES)
COLITV(1) ENDCOL(("DASDMB 1024) ("NBRSEC 3600) ) ) JOB
(QPYJWCOL) JOBD(QIDRGUI/QIDRBCH) JOBQ(QGPL/QIDRJW)
RTGDTA("JOBD) CNTRYID(US) CCSID(37) USER("CURRENT)
SYSLIBL("CURRENT) CURLIB("CURRENT) INLLIBL("CURRENT)
ALWMLTTHD("JOBD) SPLFACN("CURRENT)
```

To submit your request now click 'Finish'

< Back
Finish
Cancel

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 SBJOB command. For more information see the help text for the SBJOB command on the IBM i. 

6 Libraries Folder

This folder contains the libraries on the system that contain Job Watcher data. Specifically, these are the libraries containing file QAPYJWRUN1. The list displays each library's name and description. By expanding a library in the tree, you will see the collections that exist within it.

IBM i Connections		Idoc720: Job Watcher - #1		
Job Watcher Libraries Definitions Data repository JVM analysis SQL tables Monitors General functions	Library Name	Description	ASP	Owner
	Bsmenges		0	BSMENGES
	Bsmenges2		0	MCCARGAR
	Dfljwc		0	MCCARGAR
	Dfljw0		0	DFL
	Dfljw0b		0	DFL
	Dfljw1		0	MCCARGAR
	Dfljw2		0	DFL
	Dfljw3		0	DFL
	Ibmjw		0	ADAMB
	Ibmpex2		0	MCCARGAR
	Jwdfn		0	MCCARGAR

Libraries in the Job Watcher Component View

For more information on this, visit the [Main Window PDF](#) documentation on the Libraries Folder and Library Folders.

6.1 Menu Options

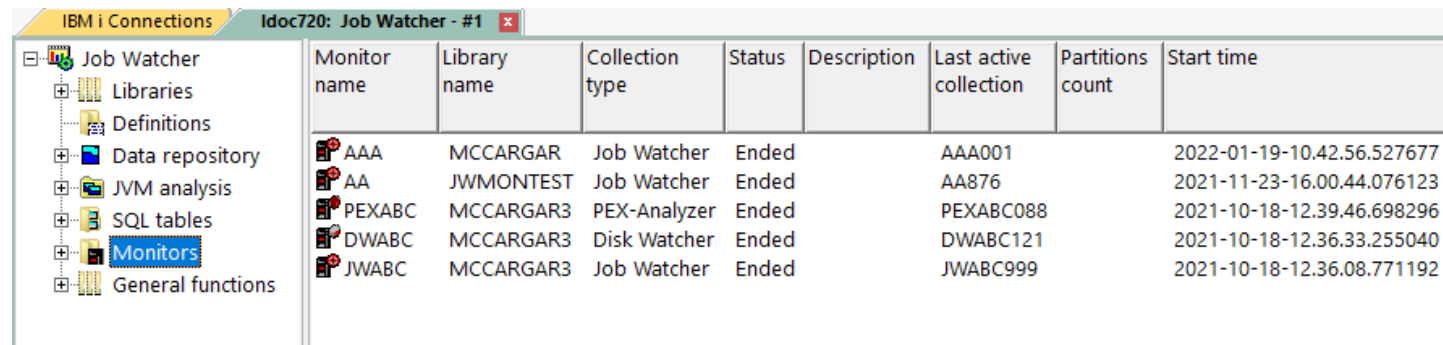
The following menu options are available by right clicking on a library in the component view.

Menu Item	Description
Start Collection...	This menu will open the Start Job Watcher Wizard where the user can define and run a collection.

Additional menu options that are common to all library folders in iDoctor are discussed [Main Window PDF](#) documentation.

7 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. 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.



The screenshot shows the 'Idoc720: Job Watcher - #1' window. On the left is a tree view with folders: Job Watcher, Libraries, Definitions, Data repository, JVM analysis, SQL tables, Monitors (highlighted), and General functions. The main area is a table with the following data:

Monitor name	Library name	Collection type	Status	Description	Last active collection	Partitions count	Start time
AAA	MCCARGAR	Job Watcher	Ended		AAA001		2022-01-19-10.42.56.527677
AA	JWMONTEST	Job Watcher	Ended		AA876		2021-11-23-16.00.44.076123
PEXABC	MCCARGAR3	PEX-Analyzer	Ended		PEXABC088		2021-10-18-12.39.46.698296
DWABC	MCCARGAR3	Disk Watcher	Ended		DWABC121		2021-10-18-12.36.33.255040
JWABC	MCCARGAR3	Job Watcher	Ended		JWABC999		2021-10-18-12.36.08.771192

Monitors Folder

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 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.

A Monitors folder is provided in Job 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 the [Main Window PDF](#) documentation.

8 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 the [Main Window PDF](#) documentation.

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.

IBM i Connections		Idoc720: Job Watcher - #1				
Job Watcher		Collection	Using Collection Summary	Status	Description	Ending reason
Libraries		SQL tables				
Bsmenges		Job Summary				
Bsmenges2		ALL	Yes	Ready		Time limit
Dfljwc		Q342130838	Yes	Ready - Missing: SQL, AIGP, IJVM	10 second intervals, Call stacks	Ended by user
Dfljw0		SP4	Yes	Ready - Missing: SQL, AIGP, IJVM		Ended by user
Dfljw0b		SP5	No	Ready - Missing: SQL, AIGP, IJVM		Ended by user
Dfljw1		Q216073945	Yes	Ready - Missing: SQL, AIGP, IJVM	1 second intervals, Call stacks abc	Ended by user
Dfljw2		Q210062149	No	Ready - Missing: SQL, AIGP, IJVM	1 second intervals, Call stacks	Interval limit
Dfljw3		Q314121655	No	Ready - Missing: SQL, AIGP, IJVM	Q314	Time limit
Ibmjw						
Ibmjw2						
Jwdfn						
Jwmonest						
Locktracej						
Mccargar						

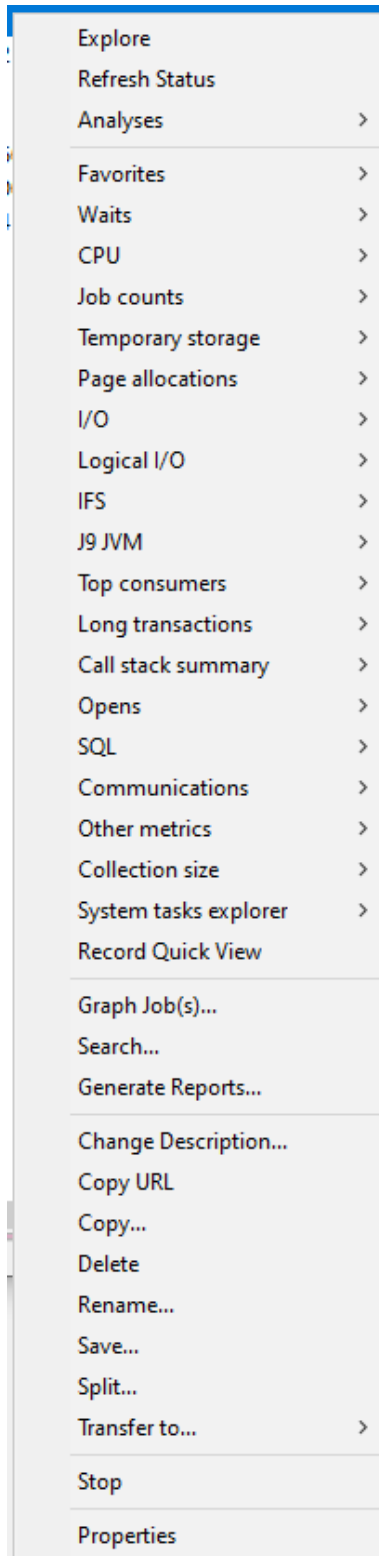
Job Watcher Collections in a Library

Some of the less obvious columns shown in a list of collections are described below:

Column	Description
Using Collection Summary	This column indicates if the Collection Summary analysis has been ran. This is required to produce many of the graphs in Job Watcher and is highly recommended to be used for best results.
Status	This indicates what files are missing in most components. It is normal for some files to be missing in all components. Only if the status indicates: "ERROR – CRITICAL FILES MISSING", then will the collection be unusable. Place your mouse pointer over this column to get more information about the missing files and which reports they apply to. Note: Use the Refresh Status menu option on the collection to update the status if it is incorrect.
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.

9.1 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.



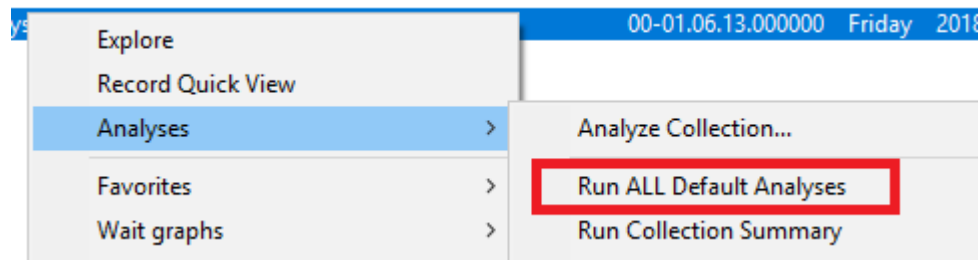
Collection popup-menu

Menu Item	Description
Explore	Show the contents of the collection.
Refresh Status	In some situations, the Status column may indicate files are missing incorrectly. This option is used to refresh the collections cache for the selected collection(s) to be sure that the files are truly missing. This also can be used in cases where report folders are missing.
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 used and are great starting points.
Waits	Contains overview and rankings wait bucket graphs. Tip: If unsure of where to investigate first, the Collection overview time signature is the best place to start.
CPU	Contains CPU, CPU utilization and CPU queueing related metrics.
Job counts	These graphs contain counts for the number of jobs that exist on the system (if Collection Summary has been ran) as well as submitted jobs.
Temporary storage	These graphs provide metrics related to job temporary storage allocations. Note: This folder only appears at 7.2+ and after Collection Summary analysis has been ran.
Page allocations	These graphs include metrics related to page allocations, page frames and pages marked easy to steal.
I/O	These graphs cover disk I/O metrics, page faults, synchronous response and more.
Logical I/O	These graphs show logical I/O metrics of various types.
IFS	These graphs include all metrics available relating to the IFS.
J9 JVM	These graphs summarize the JVM statistics for all J9 JVMs. Note: These graphs only appear if the J9 JVM data was optionally collected.
Top consumers	These graphs show the current users and generic jobs that used the most CPU or spent the most time in any of the “interesting” wait buckets.
Long transactions	This contains reports that shows periods of time where bursts of activity occurred. Note: This only appears after running the Long Transactions analysis.
Call stack summary	This folder contains reports that summarize the call stacks found in the collection. Note: This only appears after running the Call Stack Summary analysis.
Opens	These reports can help give an idea of which programs are causing opens.
SQL	These graphs show metrics related to SQL. This folder only appears at 7.2+.
Communications	These graphs show metrics related to TCP and socket activity. Note: This only appears if socket information has been optionally collected.
Other metrics	This folder contains miscellaneous graphs not covered elsewhere. It includes things like 5250 transactions and spool files created.
Collection size	These reports are used to display detailed information about the size of the collection.
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.

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 2 nd 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.
Change Description...	This option is used to modify the description shown in the list for a single collection.
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 collection into multiple pieces based on an interval range or a time range. Tip: This can be used to improve performance of graphs if the collection is very large.
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.
Stop	Ends an active collection by issuing the ENDJW command.
Properties	Displays the property pages for the collection.

9.2 Run ALL Default Analyses

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.

This interface allows you to select which analysis functions should be performed for the selected collection(s). Additional reports will be provided after performing this option.

Analyses available:

Description	Used by	Program	Run All Default
☒ Situational Analysis	Favorites, Waits, Job counts, Physical Disk I/Os	QIDRJWA3	1
☒ Collection Summary	Favorites, Waits, CPU, Job counts, I/O, IFS and other graphs	QIDRJWSUM1	1
☐ Collection Summary - Clients and workers	Waits -> Clients + Workers Overview, Waits -> Clients + Workers rankings	QIDRJWCLT	0
☐ Call Stack Summary	Call stack summary	QIDRJWSTKA	
☐ Change sensitive user data		QIDRJWXR1	
☐ Collection Summary by TDE type (must run Collection Summary fi...	SQL tables -> Collection Summary by TDE type	QIDRJWTSUM	
☐ Lock Trace	SQL tables -> Lock trace	QIDRJWLCK1	
☐ Destroy all host variable data in QAPYJWSQLH		QIDRJWHSTD	
☐ Job Summary	SQL tables -> Thread/Job totals	QIDRJWCJS	
☐ Long Transactions	Long transactions	QIDRJWS4	
☐ Modules Waiting	SQL tables -> Modules waiting	QIDRJWMOD1	

☐ Submit this request to a batch job instead of using a QZDASOINIT job.

☐ Always run analyses in a batch job

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

Tip: On the green screen these default analyses are ran when using the QIDRGUI/STRIDRSUM and QIDRGUI/RSTIDRDTA SUM(*YES) commands.

9.3 Graph Jobs

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. This allows a user to graph and compare 1 job with another job on the same system or any system and collection they wish.

An example follows:

These options allow you to graph job(s) from the specified collection(s) on any system.

Job/Task/Thread #1

System (IBM i):

Library: Collection:

Job or task contains: Taskcount:

Graph:

☐ **Job/Task/Thread #2**

System (IBM i):

Library: Collection:

Job or task contains: Taskcount:

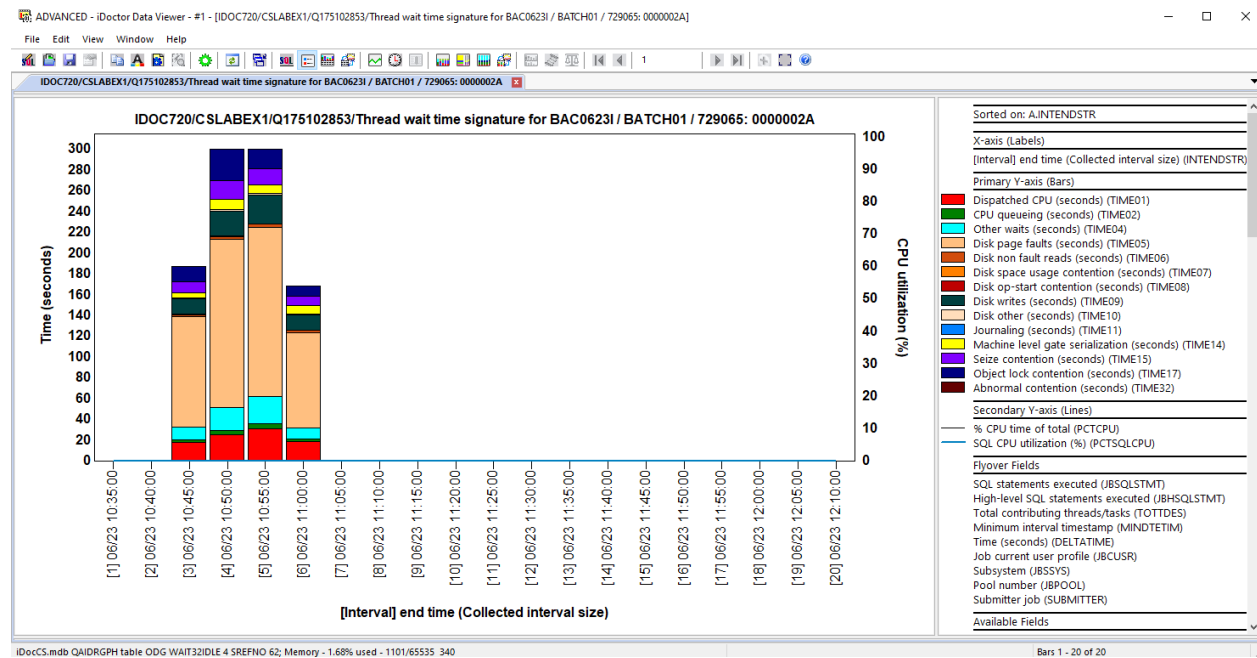
Graph:

☐ Use a case-sensitive search ☐ Keep open

Graph Job(s) Window

Some of the less obvious columns are described below:

Option	Description
Job or task contains	This allows you to enter part of the job name to reduce results when pressing the Browse button which is recommended.
Taskcount	The taskcount is the unique identifier for the job and/or task. It must be provided using the Browse option before using the Open Graph(s) button.
Graph	This is the name of the selection over time graph to open.
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.



Graph Job(s) Example graph

9.4 Search

The Search function in Job Watcher allows the user to look for a known job name, program name, subsystem, user profile, and more, 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 then use the Search menu.

An example of this interface is:

Collection Search ✕

This option allows you to quickly find the data of interest to you based on the search type and criteria specified.

Data to search:

Collection(s):	MCCARGAR/ALL(720)	Total intervals:	31
		Starting interval:	1
Start time:	2022-01-18-06.40.41.834000	Ending interval:	31
End time:	2022-01-18-06.45.42.740000		

Search type:

- ☒ Job or task name
- ☐ Subsystem
- ☐ Pool
- ☐ Current user profile
- ☐ Call stack
- ☐ Taskcount
- ☐ SQL statement
- ☐ Current wait (object)
- ☐ J9 call stack

Search criteria:

Job or task name contains: Browse...

☐ Starts with search

☒ Include system tasks

☒ Include secondary threads

☐ Use a case-sensitive search

Time range (optional):

Start time: ▲▼

End time: ▲▼

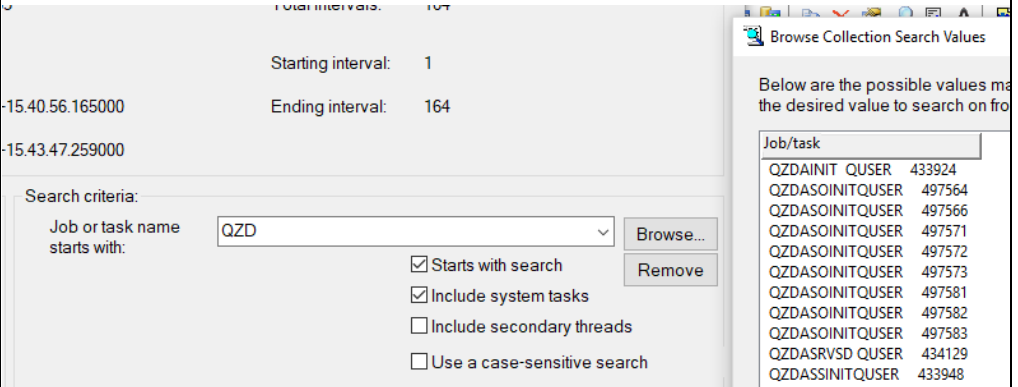
Search destination:

Send search results to: ▼

Search Cancel Help

Collection Search Window

Some of the less obvious fields are described below:

Options	Description
Collection(s)	This is the list of collections to search. It includes library name, collection name and VRM for each.
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:  <i>Job starts with QZD Browse function example</i>
Starts with search	This checkbox is used to indicate if the search should be on the beginning of the data being searched or just contained anywhere in the field.
Use a case-sensitive search	If you need to search on a mixed case system task name, then check this box.

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.

For example, performing a search using search type “Job or task name starts with” with the value QZDA would give a report like this:

Q210062149/Job or task name contains QZDA Tasks included Secondary threads included - #1												
Job name/user/number: thread ID (OBJNAME)	Generic job name (GENJOBNAME)	Dispatched CPU (seconds) (TIME01)	Interval number (INTERVAL)	Time of day at ending snapshot start (STARTOD)	Reserved (TRESERVE1)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Elapsed interval time in microseconds (TDEUSECS)	Microsecs since IPL at ending snapshot start (STARTUSECS)	Microsecs since IPL at ending snapshot end (ENDUSECS)	Thread ID (THREADID)	Initial thread task count (TASKCOUNT)	Jot (TC)
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0823	1	2021-07-29-06:21.50.818000		6,779,806	1,083,218	8,849,224,000...	8,849,224,000...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0375	65	2021-07-29-06:22.56.646000		6,779,806	1,038,460	8,849,289,829...	8,849,289,829...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0355	42	2021-07-29-06:22.98.700000		6,779,806	1,036,389	8,849,266,169...	8,849,266,170...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0301	61	2021-07-29-06:22.52.538000		6,779,806	1,030,109	8,849,285,720...	8,849,285,720...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0301	14	2021-07-29-06:22.04.151000		6,779,806	1,030,090	8,849,237,333...	8,849,237,333...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0301	126	2021-07-29-06:23.59.451000		6,779,806	1,030,071	8,849,352,633...	8,849,352,634...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0301	234	2021-07-29-06:25.50.529000		6,779,806	1,030,065	8,849,463,711...	8,849,463,711...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0301	257	2021-07-29-06:26.14.180000		6,779,806	1,030,055	8,849,487,362...	8,849,487,362...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0301	210	2021-07-29-06:25.86.1000		6,779,806	1,030,053	8,849,439,044...	8,849,439,044...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	304	2021-07-29-06:27.02.552000		6,779,806	1,030,048	8,849,535,735...	8,849,535,735...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	120	2021-07-29-06:23.53.271000		6,779,806	1,030,048	8,849,346,453...	8,849,346,453...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	49	2021-07-29-06:22.40.185000		6,779,806	1,030,047	8,849,273,367...	8,849,273,367...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	168	2021-07-29-06:24.42.676000		6,779,806	1,030,045	8,849,395,858...	8,849,395,858...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	20	2021-07-29-06:22.10.331000		6,779,806	1,030,043	8,849,243,513...	8,849,243,513...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	333	2021-07-29-06:27.32.376000		6,779,806	1,030,042	8,849,565,559...	8,849,565,559...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	351	2021-07-29-06:27.50.898000		6,779,806	1,030,042	8,849,584,081...	8,849,584,081...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	8	2021-07-29-06:21.58.020000		6,779,806	1,030,041	8,849,231,202...	8,849,231,202...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	345	2021-07-29-06:27.44.736000		6,779,806	1,030,041	8,849,577,918...	8,849,577,918...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	381	2021-07-29-06:28.21.761000		6,779,806	1,030,042	8,849,614,944...	8,849,614,944...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	198	2021-07-29-06:25.13.535000		6,779,806	1,030,040	8,849,426,717...	8,849,426,717...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	156	2021-07-29-06:26.08.015000		6,779,806	1,030,039	8,849,383,498...	8,849,383,498...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	269	2021-07-29-06:26.56.400000		6,779,806	1,030,039	8,849,499,722...	8,849,499,723...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	251	2021-07-29-06:26.08.014000		6,779,806	1,030,037	8,849,481,196...	8,849,481,196...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	316	2021-07-29-06:27.14.913000		6,779,806	1,030,037	8,849,548,095...	8,849,548,095...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	67	2021-07-29-06:26.58.698000		6,779,806	1,030,036	8,849,291,880...	8,849,291,880...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	287	2021-07-29-06:26.45.067000		6,779,806	1,030,036	8,849,518,249...	8,849,518,249...	0000000000000102	6,779,790	Q2
QZDASOINIT / QUSER / 104232: 00000102	QZDASOI*	1.0300	114	2021-07-29-06:23.47.091000		6,779,806	1,030,034	8,849,340,273...	8,849,340,273...	0000000000000102	6,779,790	Q2

Job or task name contains QZDA 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.

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))	Definition 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)
		Definition

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 the Main Window PDF for additional information on using this interface.

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.

Tip: If your graphs are taking a very long time to appear, this option can be used to speed up the SQL statements by reducing the amount of data being analyzed.

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.

Note #1: When generating multiple collections, the name must be less than 8 characters.

Note #2: All desired analyses must be recreated in the newly created collection(s).

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 to end the collection. This is not instantaneous and could take several seconds to reflect as ended in the GUI.

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.

9.8.1 General

The General property page provides basic information about the collection such as when it was created, the job that created it, its size, and number of intervals.

IBM i Connections Job Watcher - #1 Job Watcher Collection 'Q3...' x

System	LPAR CPU	Collection File Locks
General	Creation settings	Definition
	Wait Buckets	Situations

Collection: Q342130838
 Description: 10 second intervals, Call stacks
 Library: Mccargar
 Status: Ready - Missing: SQL, AIGP, IJVM
 Job running collection: QPYJWCOL / MCCARGAR / 178407

Summary:

Total time:	00-00.35.46.379000	Refresh
Total initialization time	00-00.00.10.081000	
Start time:	2021-12-08-13.08.48.648000	
End time:	2021-12-08-13.44.24.946000	
Collection size:	58.672 megabytes	
Starting interval:	1	
Ending interval:	214	
Total intervals:	214	
Analysis flags:	111100000111110000010000010000000001110	

Copy Copy URL OK Cancel

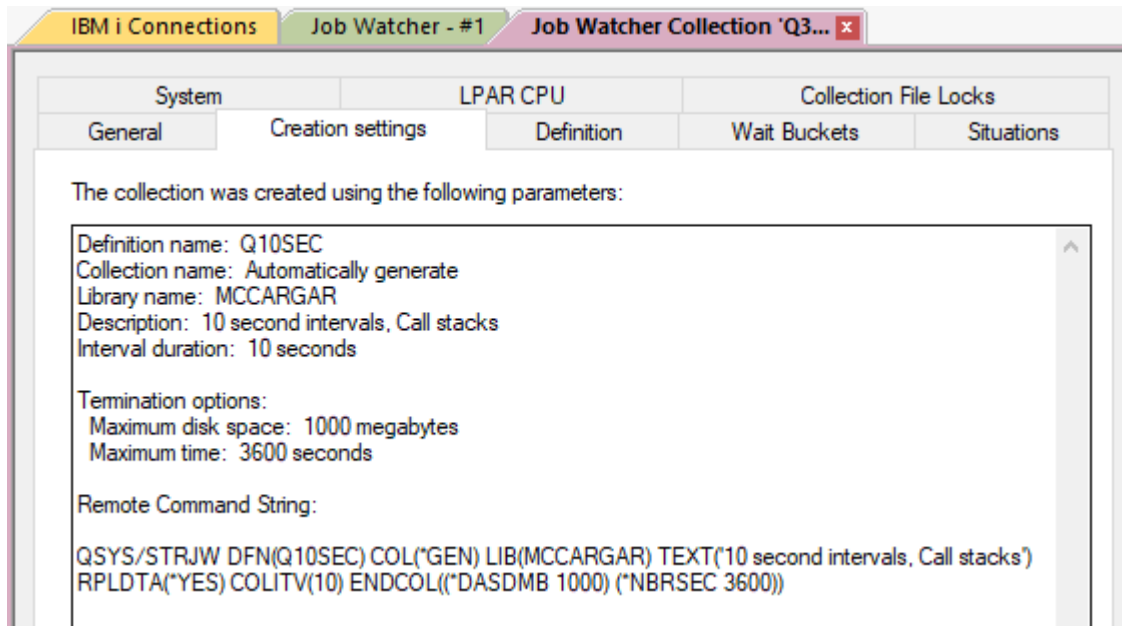
Collection Properties - General

Some of the less obvious information on this screen is described below:

Option	Field Description
Status	This indicates if the collection is usable or not and which types of files or data is missing. Note: It is normal for some files to be missing and not cause for concern.
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 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
Collection size	The total size of the collection. This number does NOT include any SQL tables generated.
Analysis flags	These indicate which tables exist in the collection and is for debug/support purposes only.

9.8.2 Creation Settings

The Creation settings property page provides details about the parameters that were used when creating the collection.

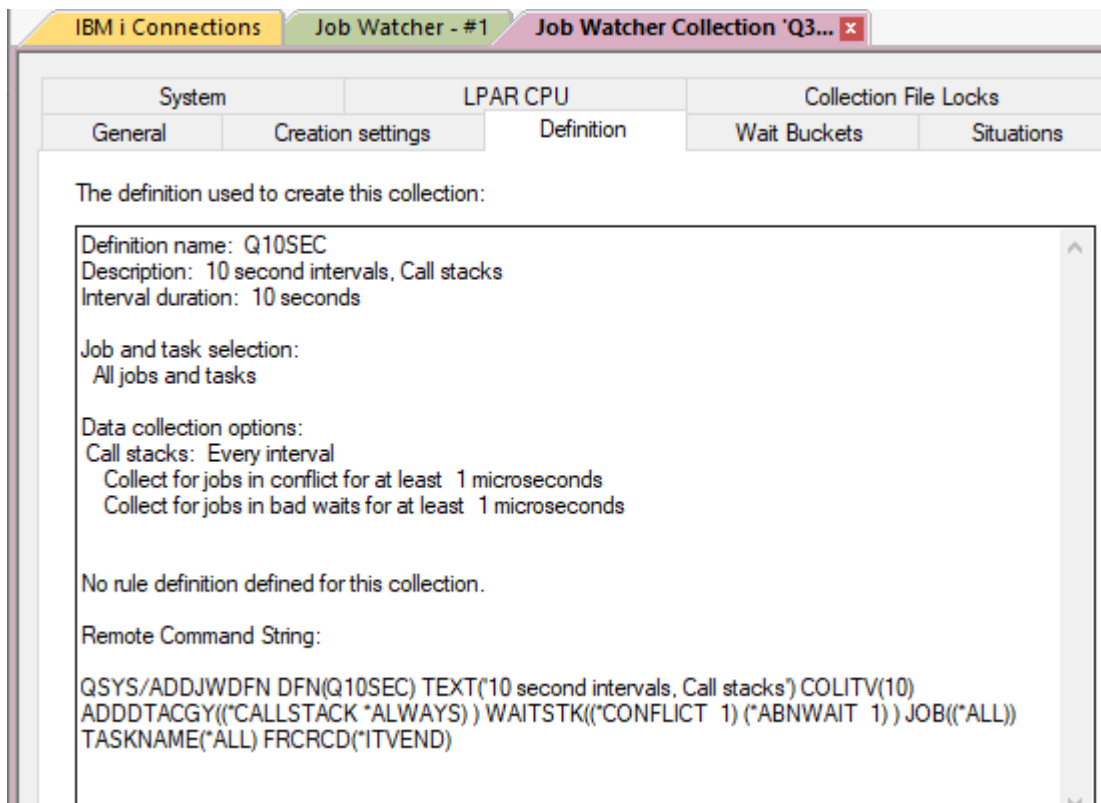


Collection Properties – Creation Settings

The information shown on this window matches the Finish page of the Start Job Watcher Wizard when the collection was created.

9.8.3 Definition

The definition page displays the parameters that were defined in the definition used to create the collection.



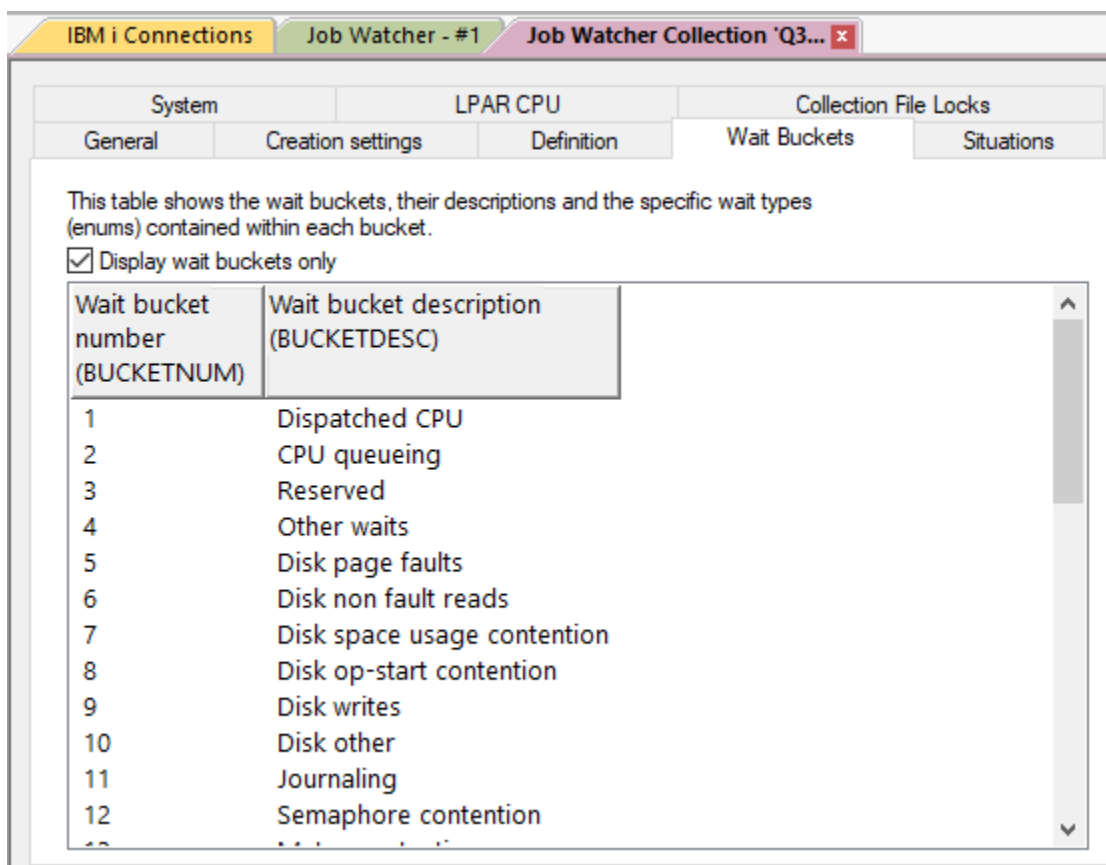
Collection Properties – Definition

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. Some graphs also exist in PEX taskswitch and utilize the same buckets.



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.

For more information on wait buckets, see the [Job Waits White Paper](#).

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.

IBM i Connections				
Job Watcher - #1		Job Watcher Collection 'Q3...		
System		LPAR CPU		Collection File Locks
General	Creation settings	Definition	Wait Buckets	Situations
This table shows the situations defined by the current level of Job Watcher. Also includes a count of the total situations that occurred in this collection for each type.				
ID	Total	Name	Problem Description	
0	0	Multiple situations occurred		
1	0	Seize/lock table large	Job(s) are waiting for fault	
2	0	Starting/ending commitment control	Job(s) appear to be const	
3	0	Poorly written/performing SQL	Job(s) may be executing p	
4	0	Missed jobs	A high percentage of jobs	
5	0	Seize contention due to data forced to disk	Job(s) are using force-end	
6	0	Fixed length of varchar or blob too small	Fixed allocated length sett	
7	0	High number of opens/closes	Contention on 'DB in use' t	
8	0	Contention on user profile	High number of creates ar	
9	0	High synchronous write response time		
10	0	Concurrent write support not enabled	Concurrent write support r	
11	0	Journal cache could help performance	Journal cache may not be	
12	0	Jobs ineligible to run	Jobs are ineligible to run,	

Collection Properties – Situations

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.

IBM i Connections				
Job Watcher - #1		Job Watcher Collection 'Q3...		
System		LPAR CPU		Collection File Locks
General	Creation settings	Definition	Wait Buckets	Situations
System information at the time of collection:				
Description	Value			
SYSTNAME	IDOC720			
OSVRM	V7R2M0			
SYSTSERIAL	066445R			
TYPEMODEL	8231-E2B			
POWERTYPE	P7			
NUMPROC	8			

Collection Properties – System

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.

Description	Average	Maximum	Minimum
Interval delta time (seconds)	10.0298	10.0815	10.0010
Interval CPU time (seconds)	.1350	4.1450	.0090
System % CPU utilization	.16%	5.16%	.01%
Uncapped % CPU utilization	.16%	5.16%	.01%
Current processor capacity	8	8	8

Collection Properties – LPAR CPU

9.8.8 Collection File Locks

Use this interface to see which jobs on the system have locks on any of the Job Watcher collection file/members. **Note:** This is for informational purposes only. No actions can be taken from here.

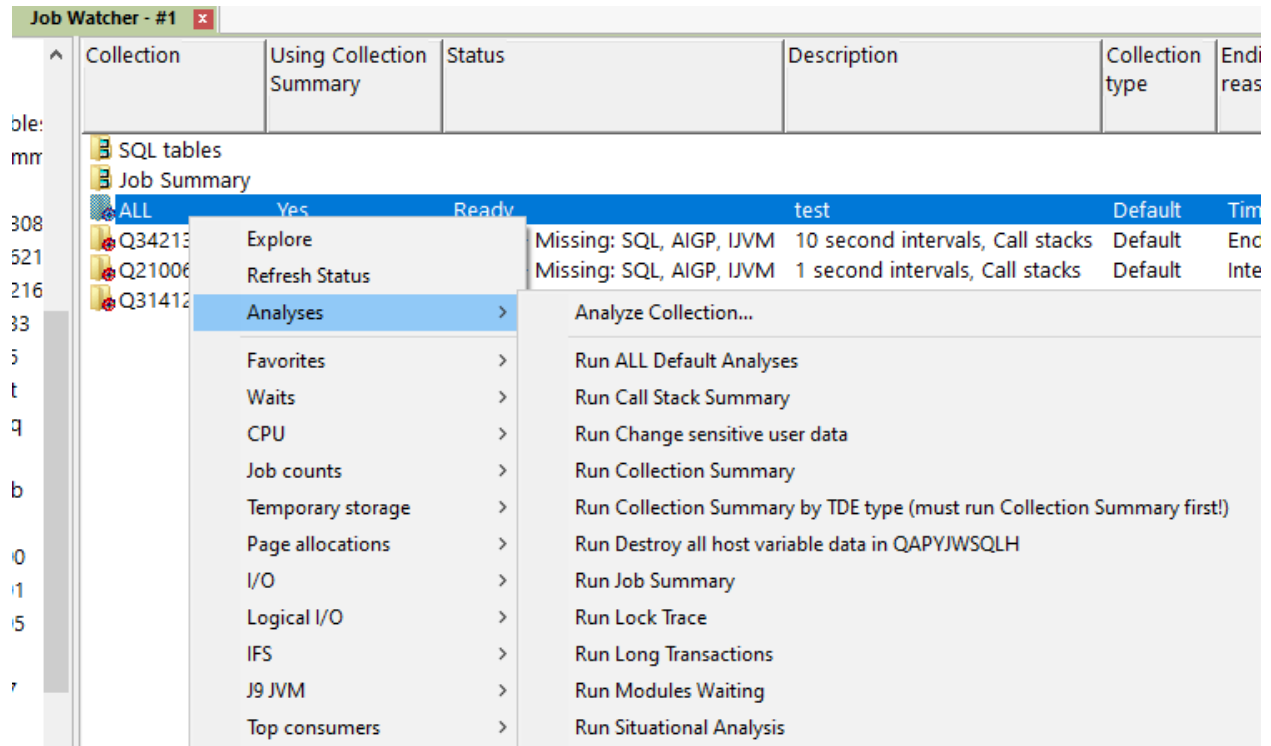
Job name (JOBNAME)	Job user name (JOBUSER)	Job number (JOBNBR)	Thread ID (THREAD_ID)	Object name (OBJECT_NAME)	Member name (SYSTEM_TABLE_MEMBER)	Object type (OBJECT_TYPE)	SQL object type (SQL_OBJECT_TYPE)	Member lock type (MEMBER_LOCK_Ty)
QZDASOINIT	QUSER	185953	0	QAPYJWINTI	ALL	*FILE		MEMBER
QZDASOINIT	QUSER	185953	0	QAPYJWINTI	ALL	*FILE		DATA
QZDASOINIT	QUSER	185953	0	QAPYJWINTI		*FILE		
QZDASOINIT	QUSER	185953	0	QAPYJWRUNI	ALL	*FILE		MEMBER
QZDASOINIT	QUSER	185953	0	QAPYJWRUNI	ALL	*FILE		DATA
QZDASOINIT	QUSER	185953	0	QAPYJWRUNI	Q342130838	*FILE		MEMBER
QZDASOINIT	QUSER	185953	0	QAPYJWRUNI	Q342130838	*FILE		DATA
QZDASOINIT	QUSER	185954	0	QAPYJWRUNI	Q342130838	*FILE		MEMBER
QZDASOINIT	QUSER	185954	0	QAPYJWRUNI	Q342130838	*FILE		DATA
QZDASOINIT	QUSER	185953	0	QAPYJWRUNI		*FILE		
QZDASOINIT	QUSER	185954	0	QAPYJWRUNI		*FILE		

Collection Properties – Collection File Locks

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.



Analyses popup-menu for a collection

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.

This interface allows you to select which analysis functions should be performed for the selected collection(s). Additional reports will be provided after performing this option.

Analyses available: Situations... Clear Toggle Selected

Description	Used by	Program	Run All Default
☐ Call Stack Summary	Call stack summary	QIDRJWSTKA	
☐ Change sensitive user data		QIDRJWXR1	
☐ Collection Summary	Favorites, Waits, CPU, Job counts, I/O, IFS and other graphs	QIDRJWSUM1	1
☐ Collection Summary - Clients and workers	Waits -> Clients + Workers Overview, Waits -> Clients + Workers rankings	QIDRJWCLT	0
☐ Collection Summary by TDE type (must run Collection Summary fi...	SQL tables -> Collection Summary by TDE type	QIDRJWTSUM	
☐ Destroy all host variable data in QAPYJWSQLH		QIDRJWHSTD	
☐ Job Summary	SQL tables -> Thread/Job totals	QIDRJWCJS	
☐ Lock Trace	SQL tables -> Lock trace	QIDRJWLCK1	
☐ Long Transactions	Long transactions	QIDRJWS4	
☐ Modules Waiting	SQL tables -> Modules waiting	QIDRJWMOD1	
☐ Situational Analysis	Favorites, Waits, Job counts, Physical Disk I/Os	QIDRJWA3	1

☐ Submit this request to a batch job instead of using a QZDASOINIT job.

☐ Always run analyses in a batch job

OK Cancel

Job Watcher Analyze Collection(s) Window

Each available analysis is presented to the user on this screen. **Only the checked analyses will be executed.**

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 unchecks all analyses.
Toggle selected button	This button changes the checked state of all analyses in the list.
Analyses available list	This is the list of the analyses available. 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 in a new submitted job. NOTE: This requires FTP access to the IBM i or this option will fail.
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 .

10.1.1 Job Watcher Situations Window

This window is accessed by clicking the Situations... button on the Analyze Collection(s) window. It allows you to turn on or off any situations you want to disable or enable or change their settings.

Note: When changes are made to iDoctor-defined situations, those are indicated with the Changed column showing 'Yes'.

Job Watcher Situations

Use these options to indicate which situations should be ran and the filters to be applied (where applicable). By changing the filter values you can increase or decrease the likelihood of a situation occurring.

Selected Situation Quick Edit Options:

Situation: IBM-defined

Minimum asynchronous reads rate per second - Default = 100 Color:

Situations Available:

Show	Changed	ID	Situation	Filter	Filter description
☒		1	Seize/lock table large		
☒		2	Starting/ending commitment control		
☒		3	Poorly written/performing SQL	100	Minimum asynchronous reads rate per second - Default = 100
☒		4	Missed jobs	.05	Minimum percentage of missed jobs/tasks - Default = 5%
☒		5	Seize contention due to data forced to disk		
☒		6	Fixed length of varchar or blob too small		
☒		7	High number of opens/closes		
☒		8	Contention on user profile		
☒		9	High synchronous write response time	3	Minimum synchronous writes response time - Default = 3 ms
☒		10	Concurrent write support not enabled		
☒		11	Journal cache could help performance		
☒		12	Jobs ineligible to run		
☒		13	Holder job delaying other work	3	Minimum number of threads held up - Default = 3
☒		14	CPU queueing may be less than what is reported by JW		
☒		16	Deadlock due to DB record locks		
☒		17	SQL field procedure called		

Job Watcher Situations

Some of the less obvious options are described below:

Option	Description
Situation (name)	This field displays or allows you to modify the current situation name.
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.
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 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!

10.1.2 Job Watcher Situations Editor

This screen is shown when creating a new situation or editing an existing one. When making changes to the SQL statement be sure to use the **Test SQL** button before hitting Accept to be sure it will work and provide the desired situations (hits) within the current collection.

Note: The SQL statement examples are only enabled when creating a new situation.

Job Watcher Situations Editor

Situation ID: (50 - 99 allowed for user-defined) Color:

Description:

Filter value: Filter description:

SQL Statement Examples:

SQL Statement:
 -- This situation checks for a 'high' rate of asyc reads per second with page faulting while running SQL statements.
 -- This could mean a poorly implemented SQL statement is running.
 SELECT 3 AS ID, INTERVAL, TASKCOUNT, 1 AS TOTAL
 FROM <<LIBNAME>>/QAPYJWTDE
 WHERE LICWO IN('SFP')
 AND SQLINTHRD = 1 |
 AND (ASYDBRD / (TDEUSECS * .000001)) >= <<FILTER>>

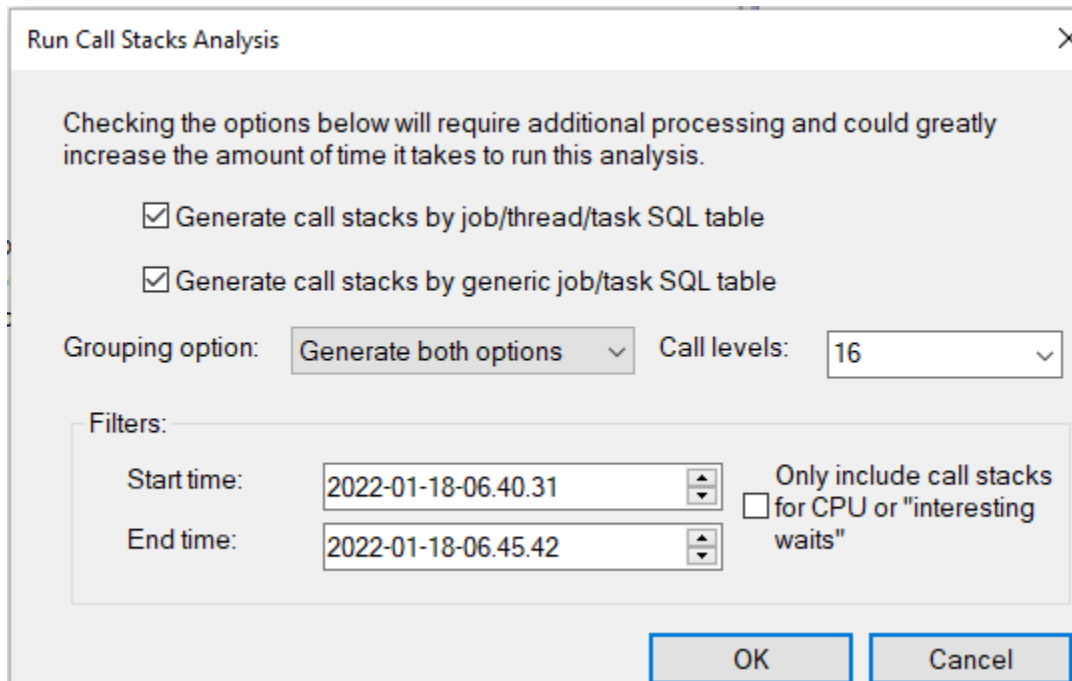
SQL Statement results: ☐ Include job name in results if applicable (for test only)

No rows found.

10.2 Call Stack Summary

This analysis is used to analyze the call stack data 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:



Run Call Stacks Analysis

Checking the options below will require additional processing and could greatly increase the amount of time it takes to run this analysis.

☒ Generate call stacks by job/thread/task SQL table

☒ Generate call stacks by generic job/task SQL table

Grouping option: Generate both options Call levels: 16

Filters:

Start time: 2022-01-18-06.40.31

End time: 2022-01-18-06.45.42

☐ Only include call stacks for CPU or "interesting waits"

OK Cancel

Run Call Stacks Analysis Window

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 create an additional report 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 . Note: If offset and statement number must exist in the reports, then the instruction address grouping option is required.
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 be grouped together. Any value can be entered from 16 to 50. Note: The larger this value is, the slower the analysis will be.
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 **Call stack summary** folder will appear under the collection.

10.2.1 SQL Tables

The list of SQL tables generated by the analysis are shown below:

SQL table	Description
Qaidrjwstkgenjob0_ <MBR>	Generic job call stack summary file grouped by procedure
Qaidrjwstkgenjob1_ <MBR>	Generic job call stack summary file grouped by instruction
Qaidrjwstkgenjobkeys0_ <MBR>	Generic job call stack keys grouped by procedure
Qaidrjwstkgenjobkeys1_ <MBR>	Generic job call stack keys grouped by instruction
Qaidrjwstkgenjobstats0_ <MBR>	Generic job call stack stats grouped by procedure
Qaidrjwstkgenjobstats1_ <MBR>	Generic job call stack stats grouped by instruction
Qaidrjwstkjob0_ <MBR>	Job/thread call stack summary file grouped by procedure
Qaidrjwstkjob1_ <MBR>	Job/thread call stack summary file grouped by instruction
Qaidrjwstkjobkeys0_ <MBR>	Job/thread call stack keys grouped by procedure
Qaidrjwstkjobkeys1_ <MBR>	Job/thread call stack keys grouped by instruction
Qaidrjwstkjobstats0_ <MBR>	Job/thread call stack stats grouped by procedure
Qaidrjwstkjobstats1_ <MBR>	Job/thread call stack stats grouped by instruction

Qaidrjwstkjavaproc0_ <MBR>	J9 JVM call stack summary
Qaidrjwstkjavaproc1_ <MBR>	J9 JVM call stack summary
Qaidrjwstkjavastack0_ <MBR>	J9 JVM call stack keys
Qaidrjwstkjavastack1_ <MBR>	J9 JVM call stack keys
Qaidrjwstksum0_ <MBR>	Collection call stack summary file grouped by procedure
Qaidrjwstksum1_ <MBR>	Collection call stack summary file grouped by instruction
Qaidrjwstksumkeys0_ <MBR>	Collection call stack keys grouped by procedure
Qaidrjwstksumkeys1_ <MBR>	Collection call stack keys grouped by instruction
Qaidrjwstksumstats0_ <MBR>	Collection call stack stats grouped by procedure
Qaidrjwstksumstats1_ <MBR>	Collection call stack stats grouped by instruction
Qaidrjwstksumtmp0_ <MBR>	IBM internal use
Qaidrjwstksumtmp1_ <MBR>	IBM internal use

10.3 Change sensitive user data

These options are used to hide or replace potentially sensitive data in Job Watcher.

Note: This is a one-way process with NO RESTORE option. Be sure to make a backup of the collection before using. You will need to rerun the analyses on this collection as well after using.

It will update the following things in the Job Watcher files:

- 1) User programs names and library names
- 2) User module names
- 3) User procedure names
- 4) System name and system serial number
- 5) User job names, current user profiles, wait object names and holders.
- 6) J9 JVM thread names and procedure names
- 7) SQL package information
- 8) Subsystem names
- 9) Remote DBS names

Other things that could contain sensitive data that are NOT updated include:

- a) Activation group names and program activation program names
- b) SQL statements and host variable data
- c) Socket data (IP address, user profiles, job information)

10.3.1 SQL Tables

The list of SQL tables generated by the analysis is shown below and is accessible under the **SQL tables** -> **Change sensitive user data** folder. These files are mostly mappings of old names to new names.

SQL table	Description
QAIDRJWX <MBR>	List of original job names, user, number
QAIDRJWXUSER_ <MBR>	Mapping of job user names
QAIDRJWXSRM_ <MBR>	Mapping of SQL package source member names
QAIDRJWXSRF_ <MBR>	Mapping of SQL package source file names
QAIDRJWXP_ <MBR>	Mapping of procedure names (in QAPYJWPROC)
QAIDRJWXPKG_ <MBR>	Mapping of SQL package names
QAIDRJWXPGM_ <MBR>	Mapping of program names (in QAPYJWPROC)
QAIDRJWXMOD_ <MBR>	Mapping of module names (in QAPYJWPROC)
QAIDRJWXLIB_ <MBR>	Mapping of library names
QAIDRJWXJTT_ <MBR>	Mapping of J9 JVM thread names
QAIDRJWXJSJ_ <MBR>	Mapping of J9 JVM procedure names
QAIDRJWXJOB_ <MBR>	Mapping of job names
QAIDRJWXJOBSBS_ <MBR>	Mapping of subsystem names
QAIDRJWXDBS_ <MBR>	Mapping of remote DBS names

10.4 Collection Summary

The Collection Summary analysis summarizes the job and wait bucket data 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.

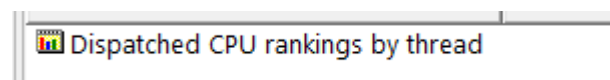
This analysis will automatically create indexes to improve performance of the graphs and reports. **Note:** Due to design limitations no QAPYJW* file/member are indexes are built unless working with the *FIRST member in the library.

After running the analysis, many new features become available in Job Watcher and are described in the next sections.

10.4.1 Additional graphs

After the collection summary analysis is ran, many additional graphs become available.

For example, this is the contents of the Waits -> Dispatched CPU rankings folder before and after the analysis is ran:



Before Collection Summary

-
-  Dispatched CPU rankings by thread
 -  Dispatched CPU rankings by job
 -  Dispatched CPU rankings by job user
 -  Dispatched CPU rankings by generic job
 -  Dispatched CPU rankings by current user
 -  Dispatched CPU rankings by pool
 -  Dispatched CPU rankings by priority
 -  Dispatched CPU rankings by subsystem
 -  Dispatched CPU rankings by job type
 -  Dispatched CPU rankings by job function
 -  Dispatched CPU rankings by generic job | current user
 -  Dispatched CPU rankings by thread | current user
 -  Dispatched CPU rankings by qro hash
 -  Dispatched CPU rankings by sql statement

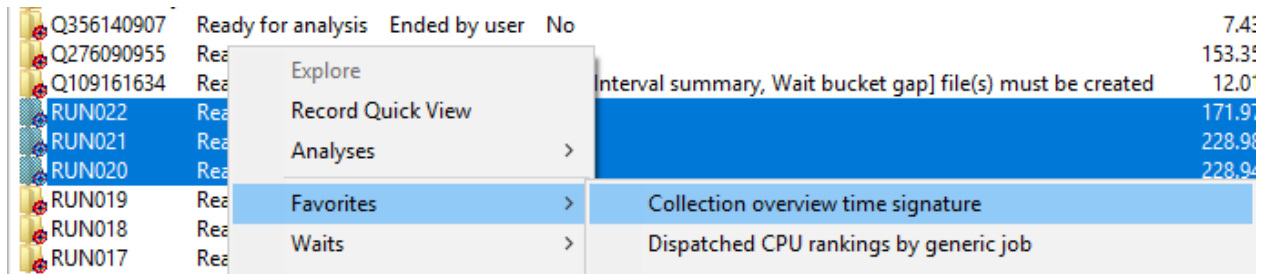
After Collection Summary

Additional graphs available after running the Collection Summary includes:

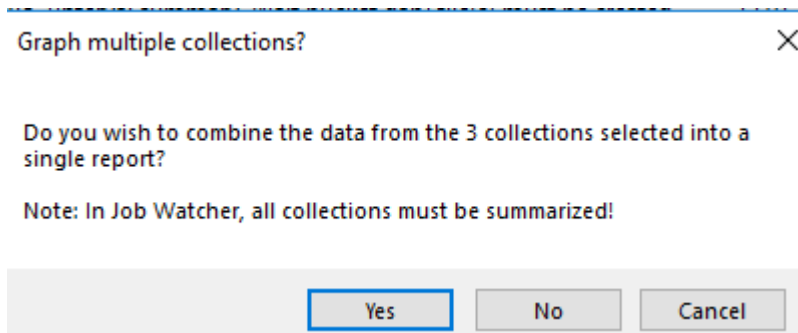
- 6 additional favorites graphs
- Many additional job ranking options
- Waits -> Collection overview time signature with workload capping (if data exists)
- Waits -> Clients + Workers wait time
- Waits -> Collection overview time signature with max waits in-progress
- Waits -> Current wait duration time signature with max waits in-progress
- Waits -> Disk time signature with max disk waits in-progress
- Waits -> Current wait duration time signature
- Waits -> Time waiting on objects
- Waits -> Wait counts
- Waits -> Average wait times
- Waits -> Average CPU times
- Waits -> Average disk wait times
- Job counts -> Job counts
- Temporary storage graphs
- Page allocations -> Temporary pages allocated/deallocated
- Page allocations -> Net perm/temp pages allocated
- Page allocations -> Net temporary pages allocated
- I/O -> Synchronous reads and writes
- I/O -> Synchronous reads and writes with avg/max/in-progress response times
- I/O -> Average synchronous read response
- I/O -> Average synchronous write response
- I/O -> Maximum synchronous read response
- I/O -> Maximum synchronous write response

10.4.2 Graphing multiple collections

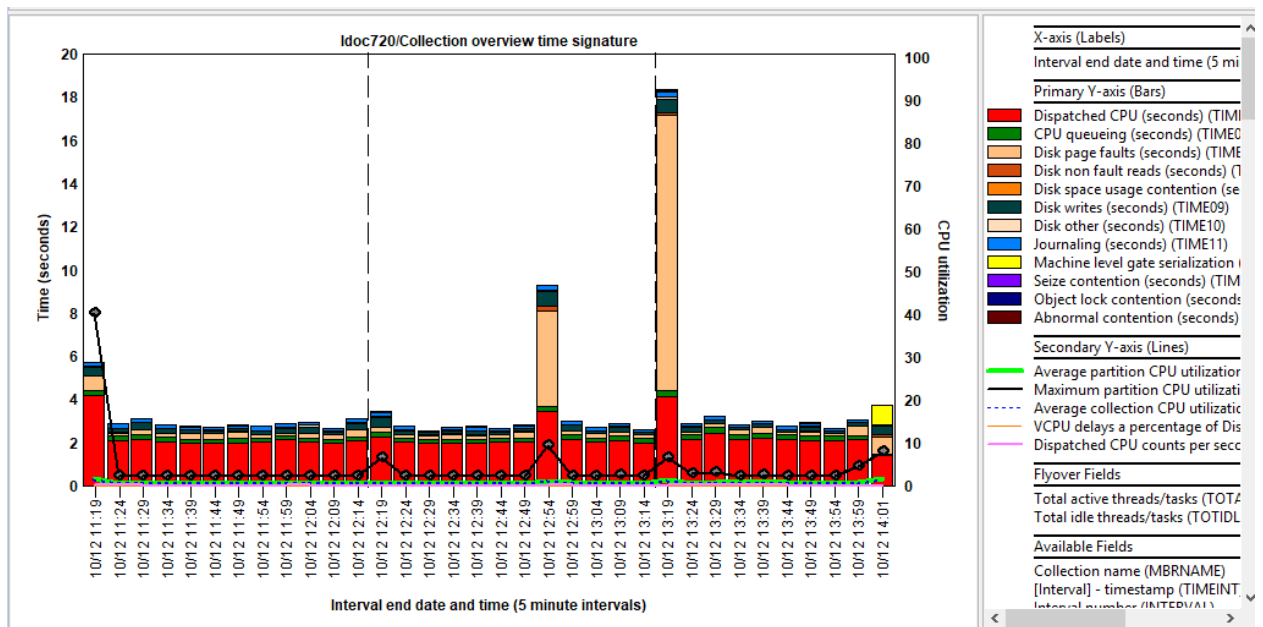
The ability to graph multiple collections within a single chart is available for many of the Job Watcher graphs. To initiate this process select them in the list of collections then right-click and pick the desired graph. **Tip:** Time breaks between collections are indicated by vertical dashed lines.



Executing a graph against 3 collections



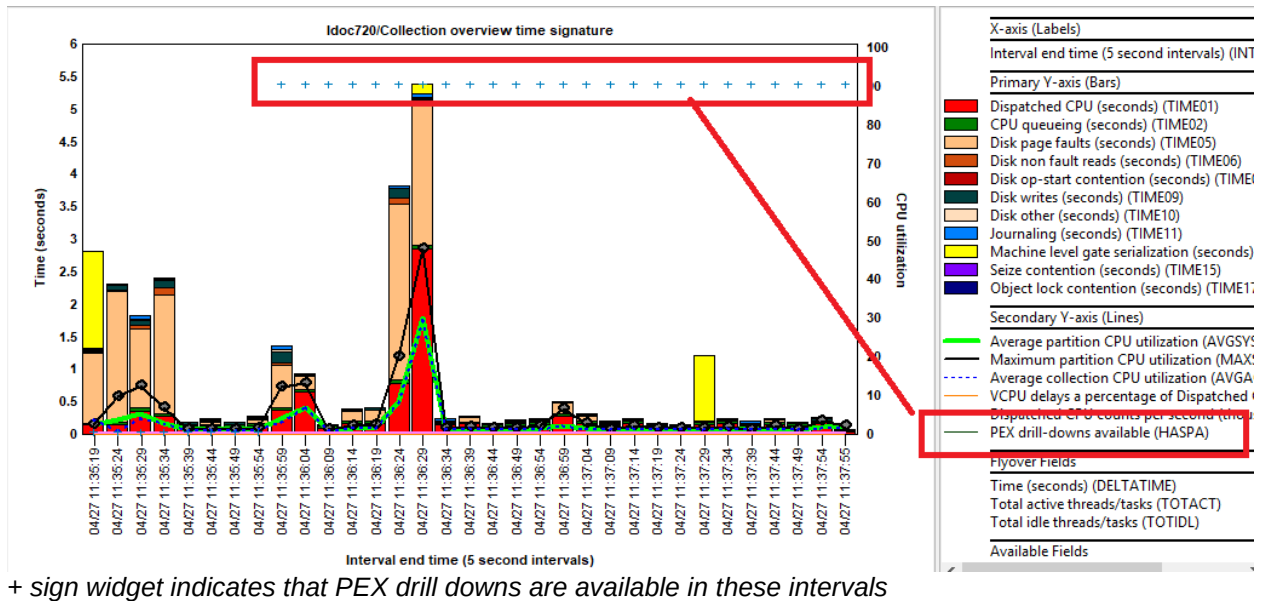
Graph multiple collections prompt



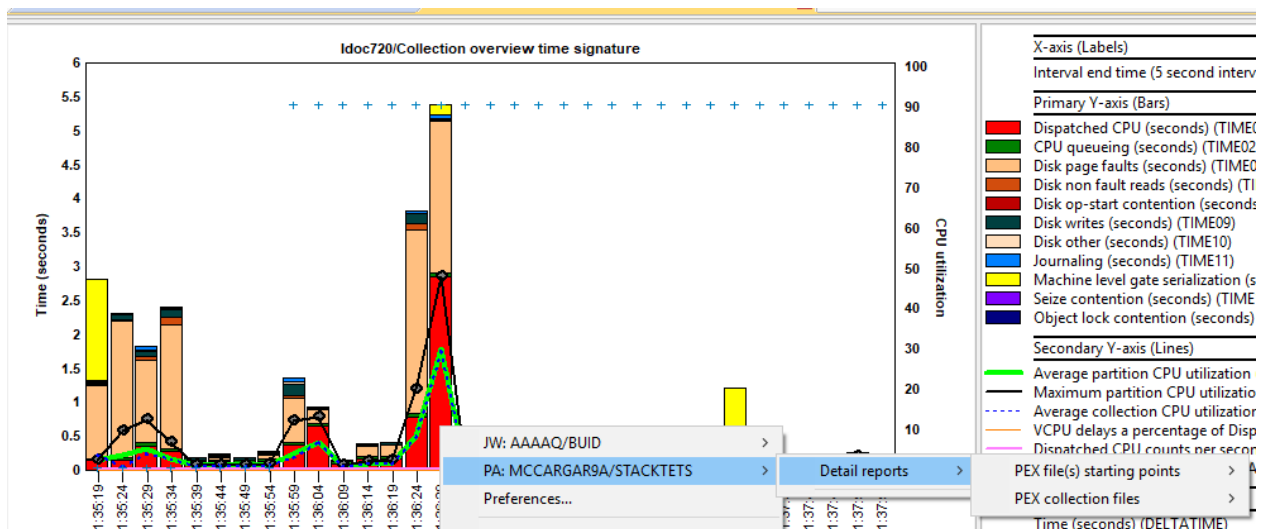
Collection overview time signature

10.4.3 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!



Right-click on data in one of these intervals to have the option to view PEX Data for the selected period.



10.4.4 SQL Tables

The list of SQL tables generated by this analysis are shown below:

SQL table	Description
QAIDRJWSUM_<MBR>	Interval summary file
QAIDRJWGAP_<MBR>>	Active and idle wait bucket times
QAIDRJWTL_<MBR>>	List of identified taskcounts
QAIDRJWCLTSUM_<MBR>	Client and worker interval summary file This refers to QDBSRV* jobs (workers) and the jobs that caused them to work (clients).
QAIDRJWCLT_<MBR>	Client and worker taskcounts
QAIDRJWSTS_<MBR>	This identifies last active intervals over time for every taskcount (TDE) on the system.
QAIDRJWQROHASH_<MBR>	Identifies if QRO data exists in the collection or not.
QAIDRJWWLC_<MBR>	Identifies if workload capping data exists in the collection.

10.5 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 (TOTDES)	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

10.5.1 SQL Tables

The list of SQL tables generated by the analysis are shown below:

SQL table	Description
QAIDRJWTSUM_<MBR>	Interval summary by TDE type

10.6 Destroy all host variable data in QAPYJWSQLH

This option **permanently** removes all host variable data in the 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.

10.7 Job Summary

The Job Summary analysis allows a user to build tables that add up job statistics across 1 or more collections.

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:

The screenshot shows the 'Job Summary' window with the following components:

- Header:** 'Job Summary' title bar with standard window controls.
- Instructions:** 'Use this function to produce summarized totals for all desired collections for each job/thread based on the filters provided.'
- Collections available:**
 - Library:** A dropdown menu showing 'Mccargar'.
 - Collection(s):** A list box containing 'Collection name', 'ALL' (highlighted), 'ALL2', 'Q210062149', 'Q314121655', and 'Q342130838'.
 - Add >>** button to move selected collections to the 'Collections to summarize' list.
- Collections to summarize:**
 - A list box containing 'Collection name' and 'MCCARGAR/ALL(720)'.
 - Remove** and **Remove All** buttons.
- Filters (separate multiple values with commas):**
 - Job (10 max):** A dropdown menu showing 'contains' and a text input field.
 - Match case** checkbox.
 - Current user profile (10 max):** A text input field.
 - Subsystem name (10 max):** A text input field.
 - Start time:** A date/time picker showing '2022-01-18-06.40.31'.
 - End time:** A date/time picker showing '2022-01-18-06.45.42'.
 - Minimum run time (hours):** A text input field showing '0'.
 - Minimum CPU (secs):** A text input field showing '0'.
- Creation options:**
 - Library:** A dropdown menu showing 'Mccargar'.
 - ☒ **Job Totals (all collections)**
 - ☒ **Thread Totals (all collections)**
 - ☐ **Generic Job Totals**
 - Name length:** A dropdown menu showing '7'.
- Comments:** A large text area for user input.
- Buttons:** **Submit** and **Cancel** buttons.

Job Summary Window

The following table describes the less obvious parts of the interface:

Option	Description
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.
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.

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 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 both the **SQL tables** or **Job Summary** folders. 2 subfolders are created for each of the creation options checked.

After running the analysis, the **Job Summary** folder will contain the new SQL tables which can be opened to provide additional analysis options.

	Analysis output	Description	Number of tables
	Job Totals	Job Summary reports grouped by job	1
	Job Totals by Collection	Job Summary reports grouped by job(process) and collection. Does not include System Tasks.	1
	Thread Totals	Job Summary reports grouped by thread	1
	Thread Totals by Collection	Job Summary reports grouped by thread and collection	1

Job Summary folder

Clicking on one of these will show the collection(s) used within the Job Summary as well as any parameter filtering used when creating the analysis.

	Description	Library	Collection(s)	VRM	Comments	Job	Current user
	Thread totals by collection	MCCARGAR	MCCARGAR/ALL(720)	V7R2M0			

Thread Totals by collection folder

10.7.1 Additional graphs

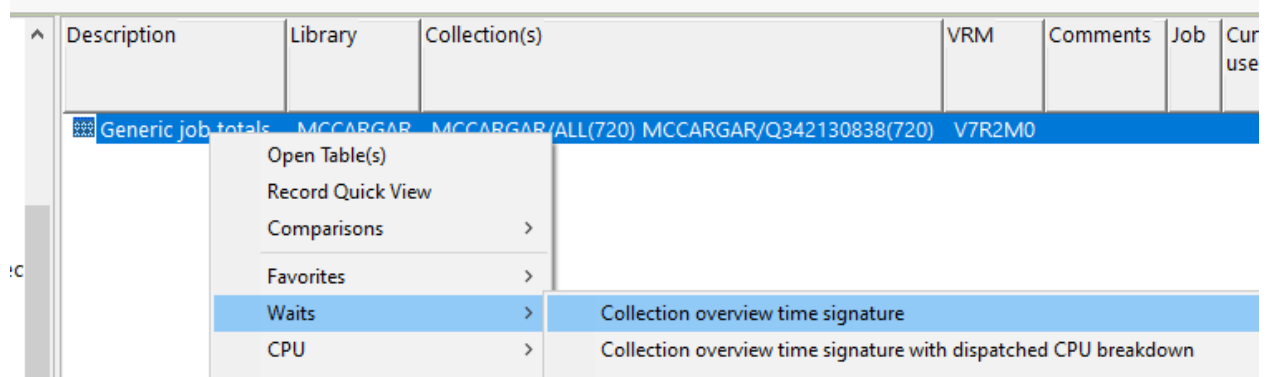
All analysis options are available under the **Job Summary** folder under a library.



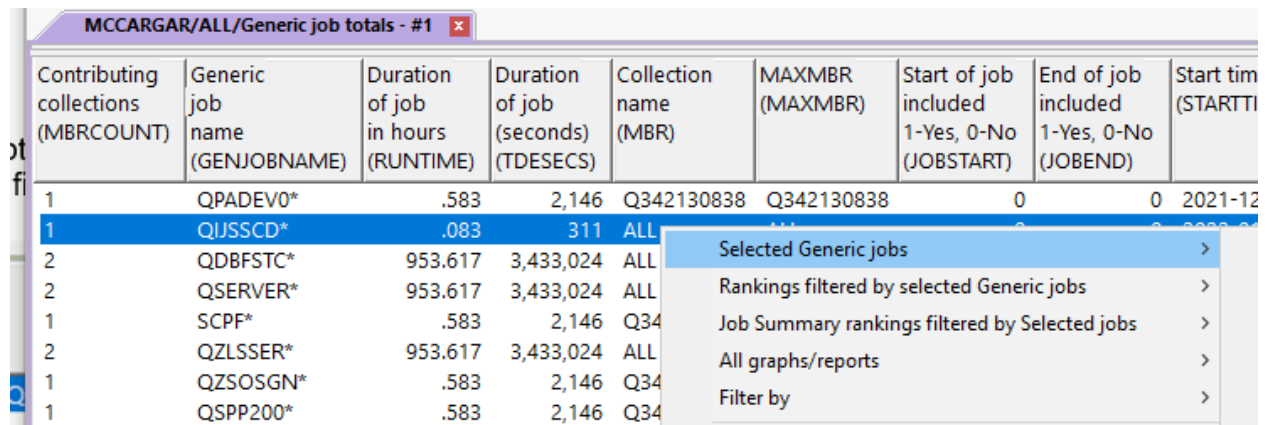
Job Summary folder

The user has 2 main ways to analyze this data.

- 1) Right-click one of the SQL tables and use one of the graphing options. This will open the graph over all collections included in the job summary. **If the data was filtered, then the collection overview only includes metrics for those filtered jobs.**



- 2) Open an SQL table in the Data Viewer, then right-click and use one of the graphing options.



10.7.2 Selected job(s) over time graphs

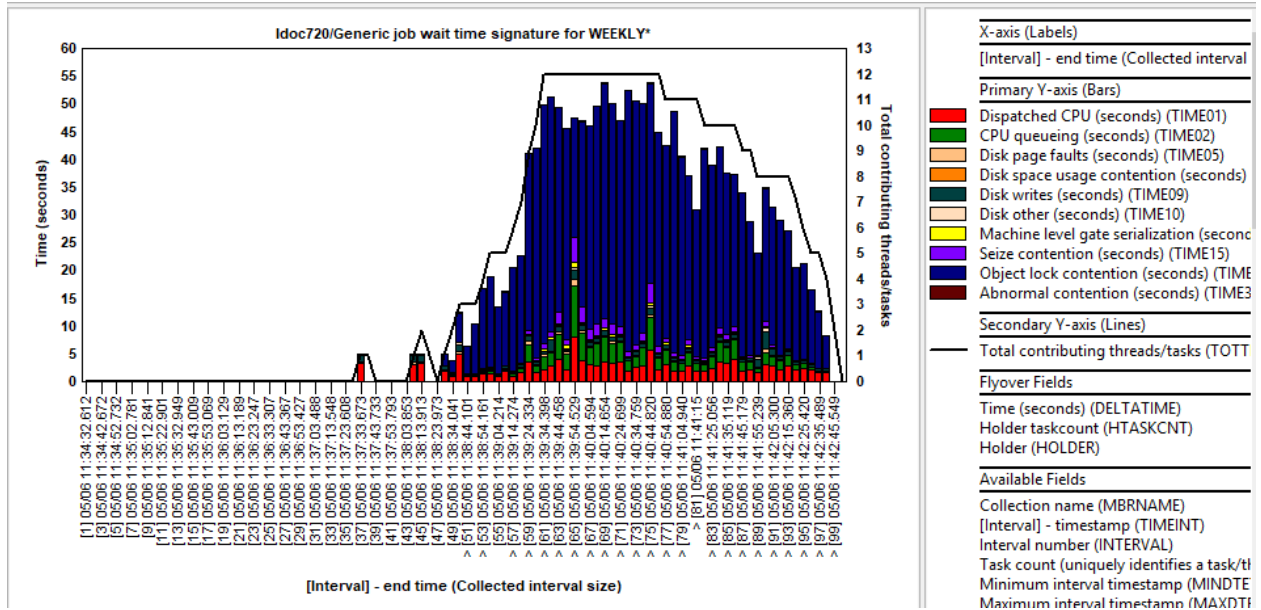
Depending on the SQL table, this option allows the user to graph the selected threads, jobs or generic jobs over time.

Tip: 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.

Note: These graphs will only work if all Job Watcher data in the original collection(s) behind these SQL tables still exists.

/PEXLABEX1/PEXLABEX1/JWMON001/Generic job totals by collection - #1													
Contributing collections (MBRCOUN	Generic job name (GENJOBNA	Collection name (MBR)	Library name (LIBNAME)	Collection description (MBRDESC)	Duration of job in hours (RUNTI	Elapsed seconds (TDESE)	Start of job included 1-Yes, 0-No (JOBSTART)	End of job included 1-Yes, 0-No (JOBEND)	Start timestamp (STARTTIME)	End timestamp (ENDTIME)	CPU time (microseconds) (DELTA CPU)	Job CPU microseconds (DELTA PR CPU)	Sy da re (S)
1	WEEKLY*	JWMON001	PEXLABEX1	5 second intervals, C									
1	WEEKLY*	JWMON002	PEXLABEX1	5 second intervals, C									
1	QZDASO*	JWMON001	PEXLABEX1	5 second intervals, C									
1	QZDASO*	JWMON002	PEXLABEX1	5 second intervals, C									

Selected Generic jobs drill down options



Generic job wait time signature

10.7.3 Rankings filtered by <selected job(s)>

This menu option allows the user to use any of the normal Job Watcher rankings graphs, but filtered down showing only the threads, jobs or generic jobs selected.

Note: These graphs will only work if all Job Watcher data in the original collection(s) behind these SQL tables still exists.

MCCARGAR/ALL/Generic job totals - #1												
Contributing collections (MBRCOUNT)	Generic job name (GENJOBNAME)	Duration of job in hours (RUNTIME)	Duration of job (seconds) (TDESECS)	Collection name (MBR)	MAXMBR (MAXMBR)	Start of job included 1-Yes, 0-No (JOBSTART)	End of job included 1-Yes, 0-No (JOBEND)	Start timestamp (STARTTIME)	End timestamp (ENDTIME)	CPU time (microseconds) (DELTA CPU)	DELTA PR CPU (DELTA PR CPU)	SYNDBR (SYNDBR)
1	QPADEV0*	.583	2,146	Q342130838	Q342130838	0	0	2021-12-08-13.08.38.565839	2021-12-08-13.44.24.943000	4,452	4,456	
1	QJSSCD*	.083	311	ALL	ALL	0	0	2022-01-18-06.40.31.724198	2022-01-18-06.45.42.731000	4,510	4,511	
2	QDBFSTC*	953.617	3,433,024	ALL	Q342130838	0	0	2021-12-08-13.08.38.562352	2022-01-18-06.45.42.731000	351,458	351,465	
2	QSERVER*	953.617	3,433,024	ALL	Q	Holder		8.562672	2022-01-18-06.45.42.731000	38	40	
1	SCPF*	.583	2,146	Q342130838	Q	Selected Generic jobs		8.561974	2021-12-08-13.44.24.943000	9,942	9,949	
2	QZLSER*	953.617	3,433,024	ALL	Q	Rankings filtered by selected Generic jobs						
1	QZSOSGN*	.583	2,146	Q342130838	Q	Job Summary rankings filtered by Selected jobs						
1	QSP200*	.583	2,146	Q342130838	Q	All graphs/reports						

Rankings filtered by selected generic jobs menu

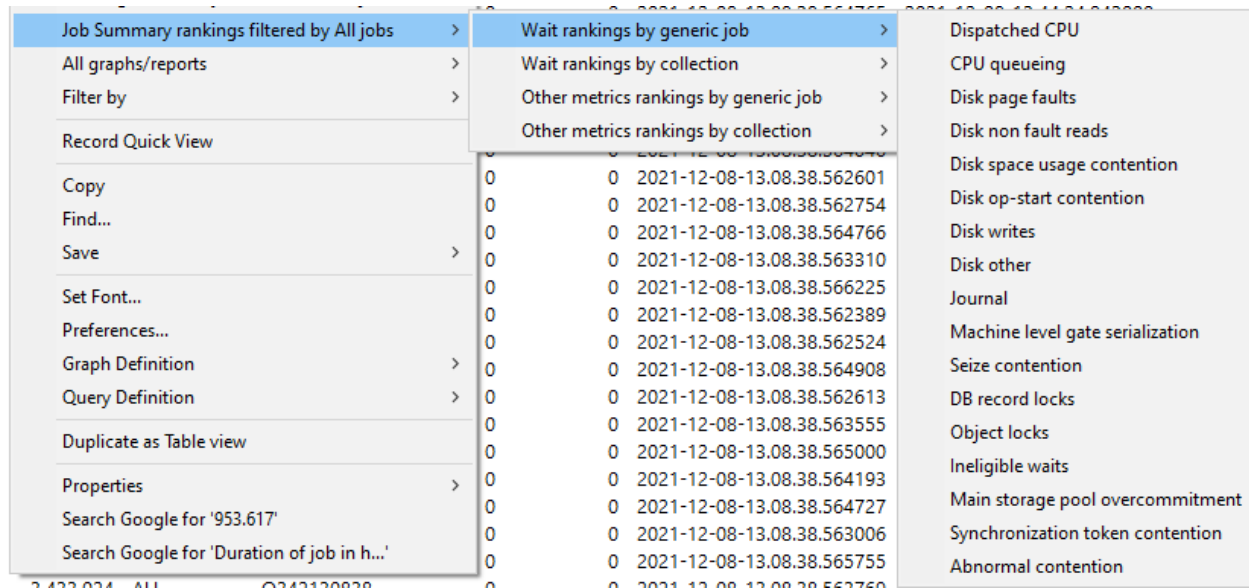
10.7.4 Job Summary rankings filtered by <X>

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:

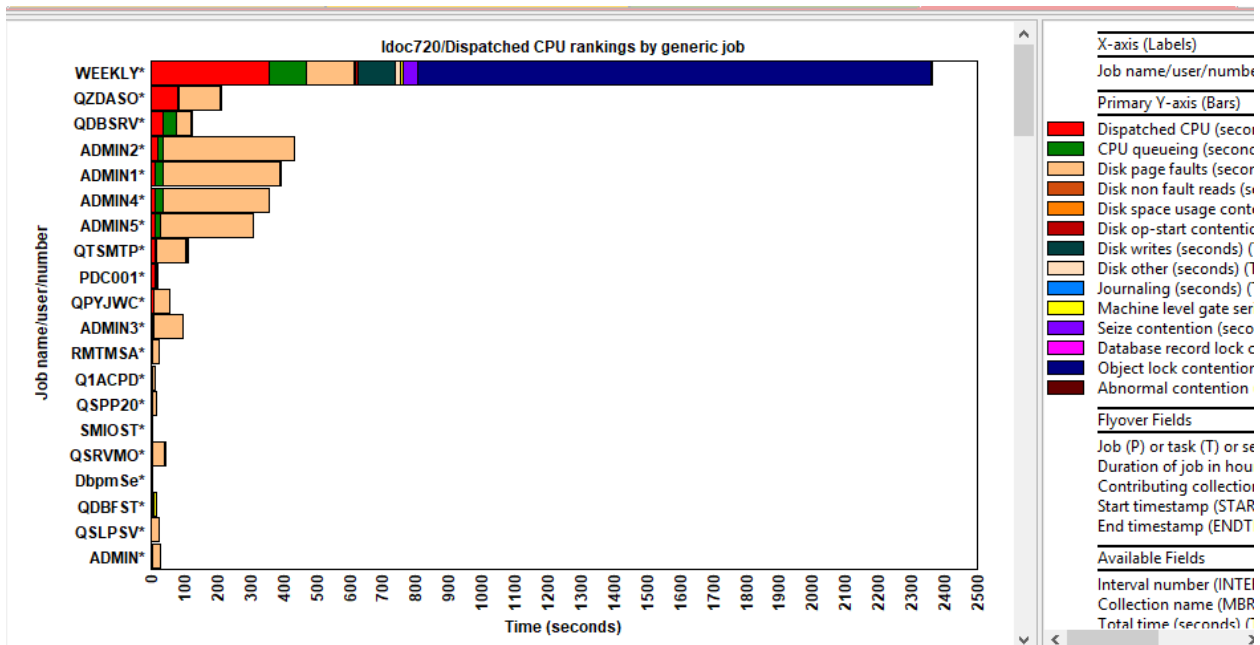
- 1) All jobs
- 2) Selected jobs
- 3) Prompt the user for a generic job name

Note: These graphs use only data found the SQL tables. (i.e. They will still work if the original collections have been deleted.)

The types of rankings wait graphs that can be generated are based on CPU and all the “interesting waits”.



Job Summary rankings filtered by All jobs example

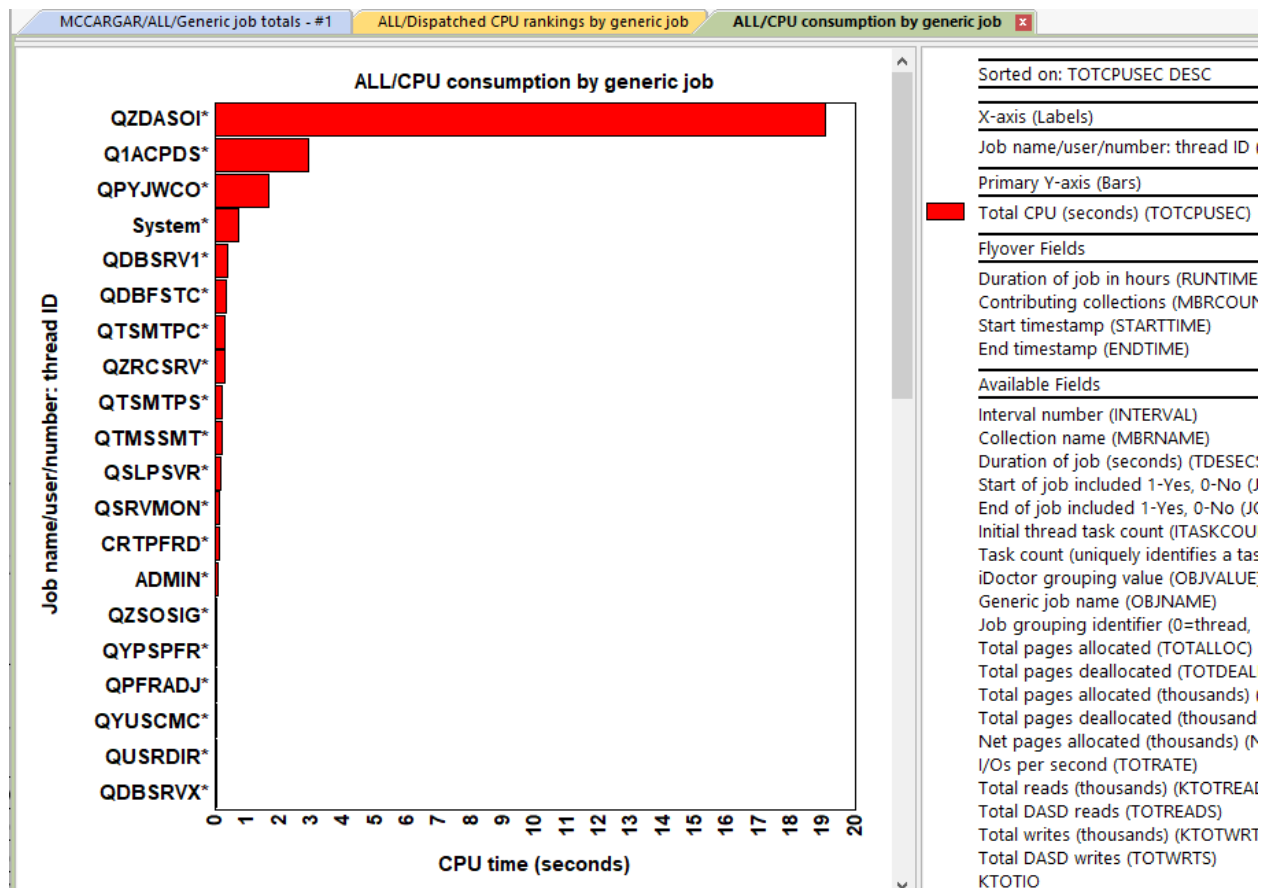


Dispatched CPU rankings by generic job (for all jobs)

The types of graphs available under the Other metrics option are:

Other metrics rankings by generic job	>	CPU consumption
Other metrics rankings by collection	>	Pages allocated/deallocated
2021-12-08-13.08.38.562734	2022-01-18-06.4	Reads and writes totals
2021-12-08-13.08.38.564766	2021-12-08-13.4	Physical I/O activity totals
2021-12-08-13.08.38.563310	2022-01-18-06.4	Logical database I/O totals
2021-12-08-13.08.38.566225	2022-01-18-06.4	Page fault totals
2021-12-08-13.08.38.562389	2022-01-18-06.4	Synchronous reads and writes
2021-12-08-13.08.38.562524	2021-12-08-13.4	Synchronous response
2021-12-08-13.08.38.564908	2022-01-18-06.4	5250 transaction totals
2021-12-08-13.08.38.562613	2022-01-18-06.4	Job run times
2021-12-08-13.08.38.563555	2022-01-18-06.4	
2021-12-08-13.08.38.565000	2022-01-18-06.4	

Job Summary rankings filtered by <X> -> Other metrics rankings by generic job menu



CPU consumption by generic job

10.7.5 SQL Tables

The list of SQL tables generated by the analysis is shown below (where <<X>> is a unique number):

SQL table	Description
QAIDRJW7SUM_ <<X>>	Generic job totals
QAIDRJW6SUM_ <<X>>	Generic job totals by collection
QAIDRJW4SUM_ <<X>>	Job totals
QAIDRJW2SUM_ <<X>>	Job totals by collection
QAIDRJW3SUM_ <<X>>	Thread totals
QAIDRJW1SUM_ <<X>>	Thread totals by collection

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 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 relating to locks.

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, access the data from the **SQL tables -> Lock trace** folder.

10.8.1 SQL Tables

The list of SQL tables generated by the analysis is shown below:

SQL table	Description
QAIDRJWLCKSUM_ <MBR>>	Lock trace job summary
QAIDRJWLCK_ <MBR>	Lock trace hits
QAIDRJWLCKPEX_ <MBR>	Lock trace PEX details Note: This will only be generated if PEX data is found!

10.9 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 tables generated by this analysis a user can view the longest periods of activity in the collection and the jobs that caused them.

After running the analysis, use the **SQL Tables -> Long transactions** folder to view results.

10.9.1 SQL Tables

The list of SQL tables generated by the analysis is shown below:

SQL table	Description
QAIDRJWTXNSUM_ <MBR>	Identifies time periods where no idle waits occurred

10.10 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 overview 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 and Waits folders will show situations.

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 additional features become available:

- Collection properties -> Situation's tab
- Interval Summary -> Situation's tab
- Detail reports -> Situations
- Situation information is displayed in the background when SQL statements are running in the Collection Overview Time Signature graphs.

10.10.1 SQL Tables

The list of SQL tables generated by the analysis is shown below:

SQL table	Description
QAIDRJWANL_DTL_ <MBR>>	Situational Analysis Detail file

10.11 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.

After running the analysis, access the data from the **SQL tables -> Modules Waiting** folder.

10.11.1 SQL Tables

The list of SQL tables generated by the analysis is shown below:

SQL table	Description
QAIDRJWMOD_ <MBR>	Modules waiting

An example follows:

ALL/Module waiting - #1 ✖						
Total (TOTAL)	Total jobs/tasks/threads (JOBCNT)	Total intervals (INTERVALS)	Objtype type (hex) and LIC wait object (LWAITOBJ)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)
523	27	31	0A-QMo			QmRealDequeueMiQueue_FR11QmDeqPrefixPcR5MiPtr13QmDequeueType
472	19	31	00-SLW			sleep_6CfSyncFUIRUIQ2_4Rmpr18InterruptLevelTypeQ2_8TDQSEnum4EnumT4
341	11	31	0A-QMo	QWTMCMNL		
254	11	31	1A-QQu			waitOnSignal_8PxsgAreaFR17PxsgSignalMessageR15PxsgWaitOptionsRUI
254	11	31	1A-QQu	QP0SSRV2	QP0SWAIT	qp0swait_FP13qp0ssigwait_t
238	9	31	00-SLW	QSOSRV1	QSOSYS	poll
230	10	31	00-SLW	QP0LLIB1	QP0LLIB1	select
121	27	31	00-Mcw			waitForObject_13MasoConditionFUIUTt1Rt
93	3	31	1A-U62			nsleep_common_FUIT1i
93	3	31	1A-U62	QP2USER2	QP2API	runpase_common_FiPvT2
73	5	31	00-EMw			emwaitonevent
69	8	31	0A-QMo	QDBSERVE		
65	4	31	0A-QMd			QmRealDequeueMiQueue_FR11QmDeqPrefixPcR5MiPtr13QmDequeueType

Modules waiting SQL table

11 Holder menu

If a holder is listed in any report a Holder menu will appear with options allowing you to drill down into the holder. To perform a drill down on a specific job, simply right-click the row or bar for the waiting job and pick one of the Holder menu options. 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.

The following shows an example of drilling down into a holder from a job within the [Interval Summary -> Objects waited](#) on interface:

The screenshot shows the 'Interval Summary: Library lbmdk2, Collection Split - #1' interface. The 'Objects waited on' tab is selected, displaying a table of threads waiting on objects. A right-click context menu is open over the thread 'JOB243 / USR399 / 675543: 00000A05', showing options like 'Holder', 'Thread wait time signature', 'Selected Threads', 'Rankings filtered by selected Threads', and 'Display call stack'.

Interval Summary – Objects waited on – Holder menu example

The Holder menu should appear in any of these situations:

- 1) The report contains HTASKCNT with a value > 0 or an HTASKNAME value.
- 2) Clicking on the Holder name within the Interval Details interface.
- 3) Any rankings or selection over time graph that contains an “H” within the FLAGS shown.

11.1 Holder chase

In the previous 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.

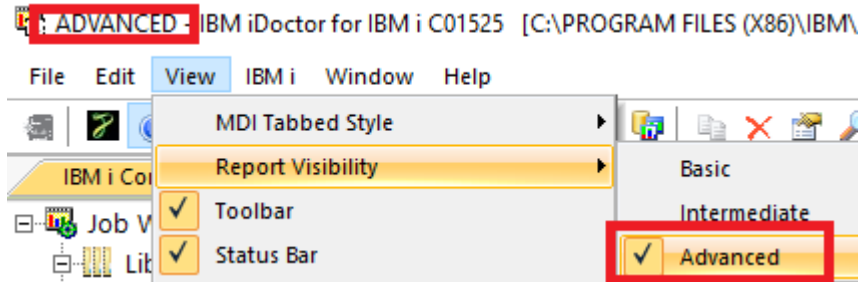
SPLIT/Collection overview time signature			Interval Summary: Library lbmdk2, Collection Split - #1			SPLIT/Holder chase for interval 5 - #1						
Collection name (MBRNAME)	Call level (LEVEL)	Interval number (INTERVAL)	Job name/user/number: thread ID (OBJNAME)	Holder taskcount (HTASKCNT)	Current or last state (CURRSTATE)	Current or last wait bucket (BLOCKBCKT)	Current or last blocking enum (BLOCKENUM)	LIC wait object (LICWO)	Current wait duration (usecs) (CURRWTDR)	Wait object library (WOBJLIB)	Wait object name (WOBJNAM)	Current user protocol (CURPROT)
SPLIT	1	5	JOB243 / USR399 / 675543: 00000A05	22,590,213	WAIT	14	2	QGa	16,158,302	LIB8	PORDER	USF
SPLIT	2	5	JOB963 / QPGMR / 541372: 00000F25	0	WAIT	14	2	QGa	12,032,275	LIB8	QJRDWH	QPI

Holder chase example (The waiter job is at level 1 and the holder job is at level 2 with no other holders)

12 Graph notes

This section provides tips about the graph metrics available in Job Watcher.

NOTE: It is critical that the View menu's Report Visibility level is set to the desired option. This is indicated in the title bar of the Main Window and Data Viewer. Otherwise graphs you might be trying to find will not appear.

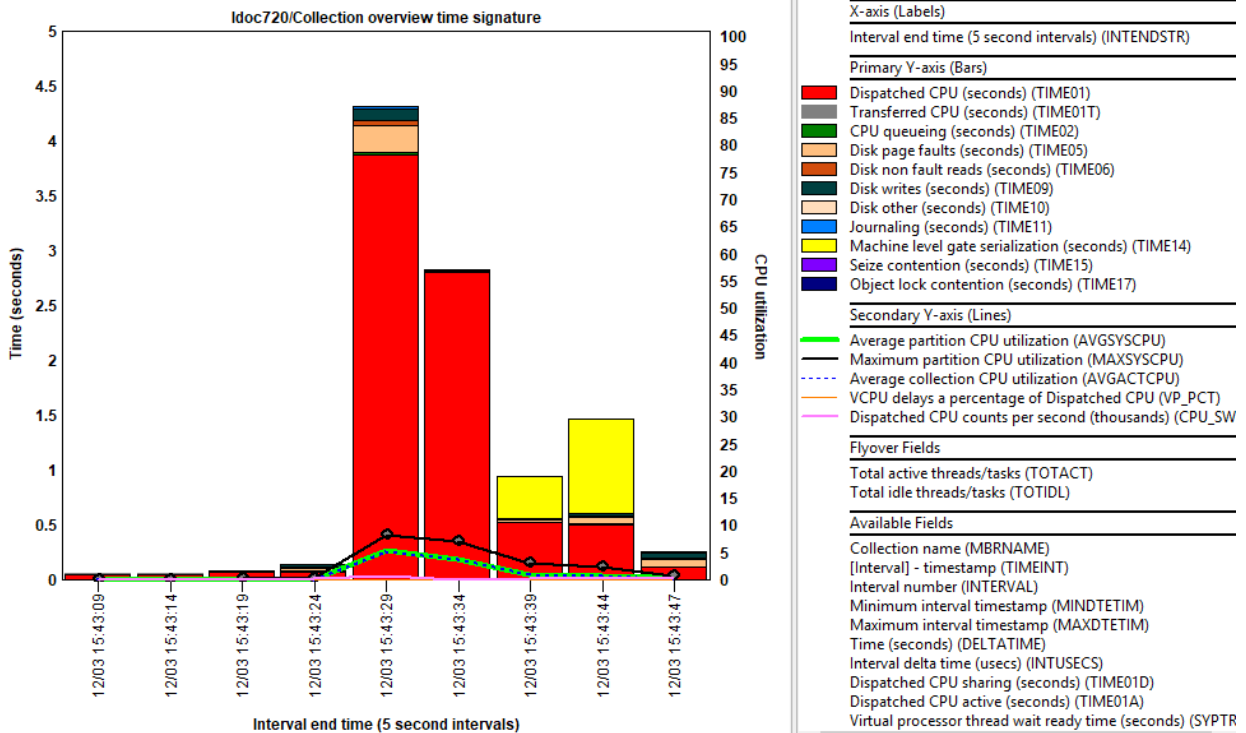


Tip: In addition, if the collection status indicates that certain files are missing, then graphs or reports using those files will NOT appear.

This section also covers briefly a discussion on “interesting” vs “idle” waits and on CPU related fields shown in many Job Watcher graphs.

12.1 CPU metrics

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 QAPYJWTD.

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 $((SYPTREADY + SYPTLATEN) / \text{DOUBLE}(1000)) / \text{TIME01} * 100.00$ 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)

Average CPU rate – 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 around 100%.

12.2 “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

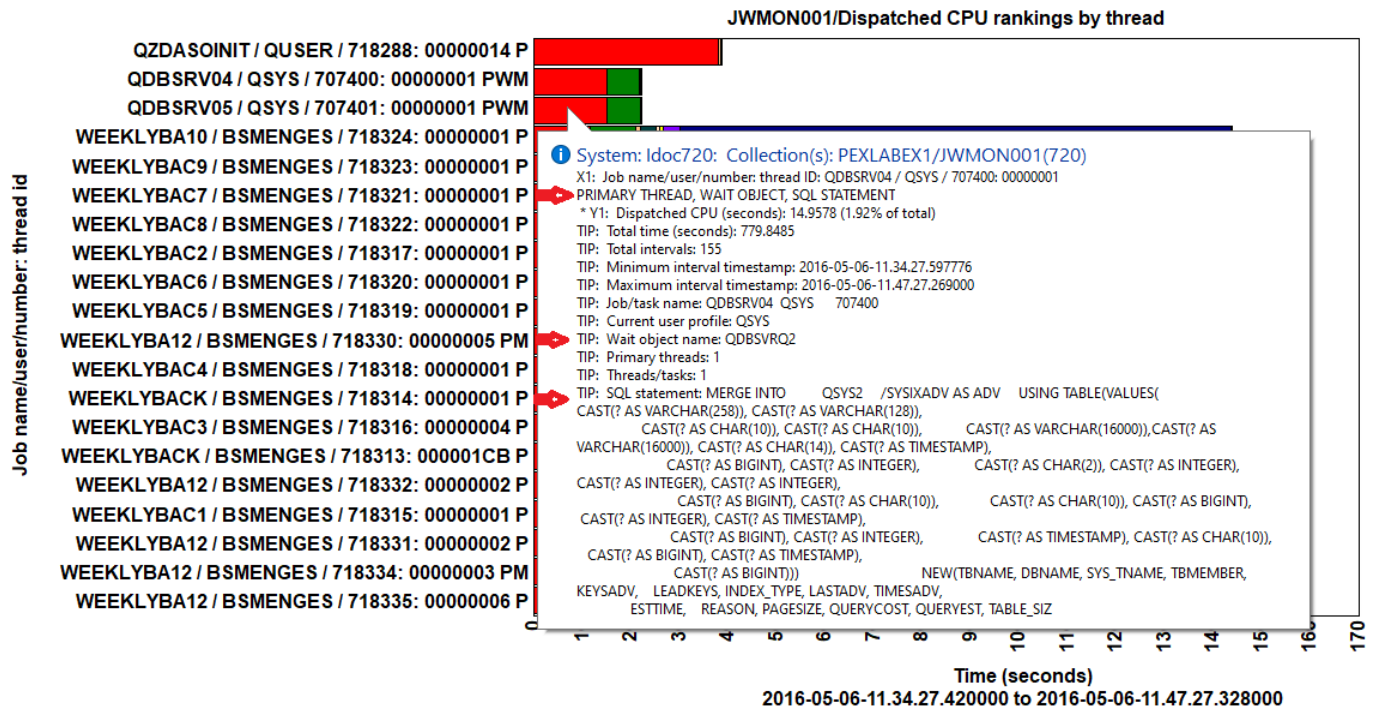
Additional buckets will appear on some graphs, but these are not “real” wait buckets. These are produced through iDoctor calculations, and their usage is only in a few graphs.

Bucket	Description
01T	Transferred CPU This value indicates CPU time spent in system tasks occurred. It is rare to see time spent in this bucket with current IBM i releases.
01A	Dispatched CPU active Time dispatched to the CPU and using it
01D	Dispatched CPU sharing Time dispatched to the CPU and not using it (virtual processors/etc.) Note: Because of processor virtualization, it is expected to see time spent in this bucket. For more info, see the Virtualization best practices section of https://www.ibm.com/downloads/cas/QWXA9XKN
02A	CPU queueing – remainder CPU queueing time NOT because of workload capping delays
02D	CPU queueing – workload capping delay

12.3 Tips / FLAGS

Both Rankings and Selection over time graphs feature an indicator at the end of the label on most graphs that provides more details behind that piece of data in the graph. A letter is used to communicate different pieces of information that is available in the graph flyover. Use the flyover to see those details.

For example, the FLAGS of “PWM” indicates that the bar is a primary thread, with a wait object and SQL statement.



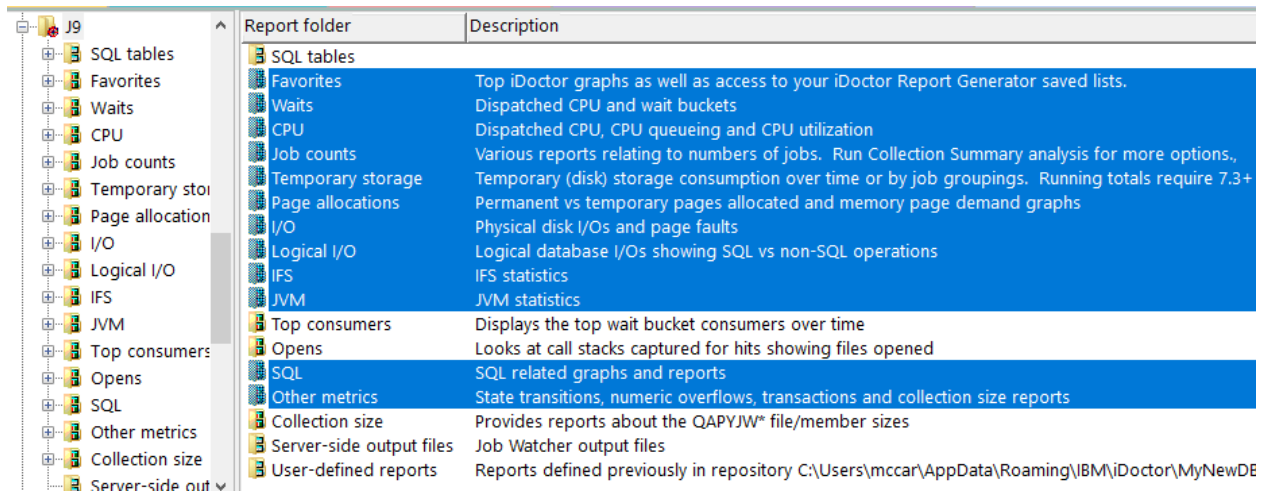
A complete list of identifiers used and what they mean is in the following list:

ID	Description
P	Primary Thread
S	Secondary Thread
T	System Task
W	Wait Object
H	Holder
B	Current wait bucket
M	SQL statement
C	SQL client job
Q	QRO hash

13 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.



Report folder	Description
SQL tables	
Favorites	Top iDoctor graphs as well as access to your iDoctor Report Generator saved lists.
Waits	Dispatched CPU and wait buckets
CPU	Dispatched CPU, CPU queueing and CPU utilization
Job counts	Various reports relating to numbers of jobs. Run Collection Summary analysis for more options.
Temporary storage	Temporary (disk) storage consumption over time or by job groupings. Running totals require 7.3+
Page allocations	Permanent vs temporary pages allocated and memory page demand graphs
I/O	Physical disk I/Os and page faults
Logical I/O	Logical database I/Os showing SQL vs non-SQL operations
IFS	IFS statistics
JVM	JVM statistics
Top consumers	Displays the top wait bucket consumers over time
Opens	Looks at call stacks captured for hits showing files opened
SQL	SQL related graphs and reports
Other metrics	State transitions, numeric overflows, transactions and collection size reports
Collection size	Provides reports about the QAPYJW* file/member sizes
Server-side output files	Job Watcher output files
User-defined reports	Reports defined previously in repository C:\Users\mccar\AppData\Roaming\IBM\iDoctor\MyNewDE

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.

Tip #1: Use the clock icon on the toolbar to change the default time grouping. 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.

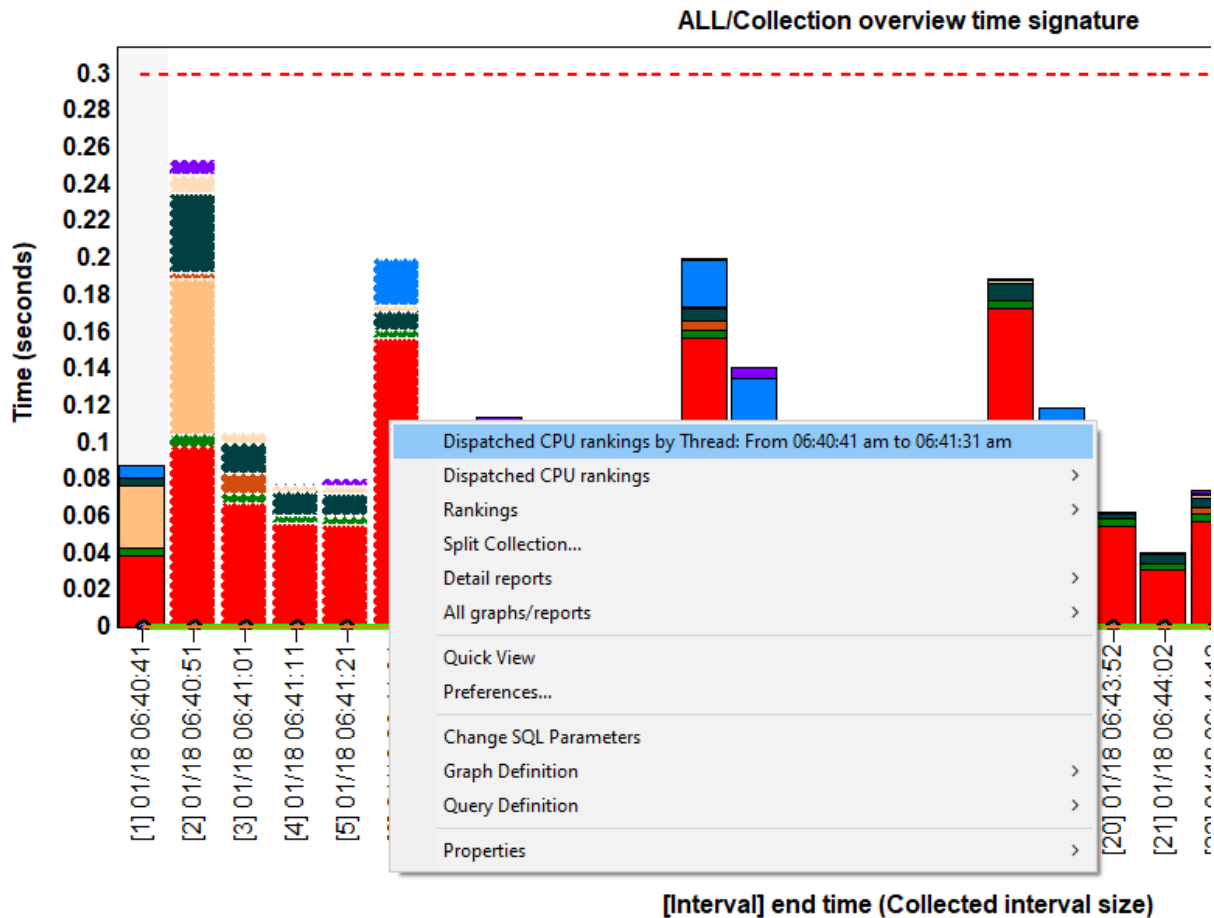
Tip #2: You can make time selections by clicking the 1st bar, hold down the shift key and clicking the last bar of the desired time range and right-click a bar and wait bucket color within that time period in order to drill down into the jobs experiencing the most amount of time in that wait bucket you 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.

13.1 Drilling down to Rankings

When drilling down into ranking graphs (from overview graphs) you can select the desired time of interest by holding down the shift key and clicking the 1st and last bars of the desired time range. Then right-click on one of the bars in the time range and pick the desired drill down graph (typically first in the list).

This action will look something like this:



Drilling down from a Collection Overview Time Signature graph into Dispatched CPU rankings

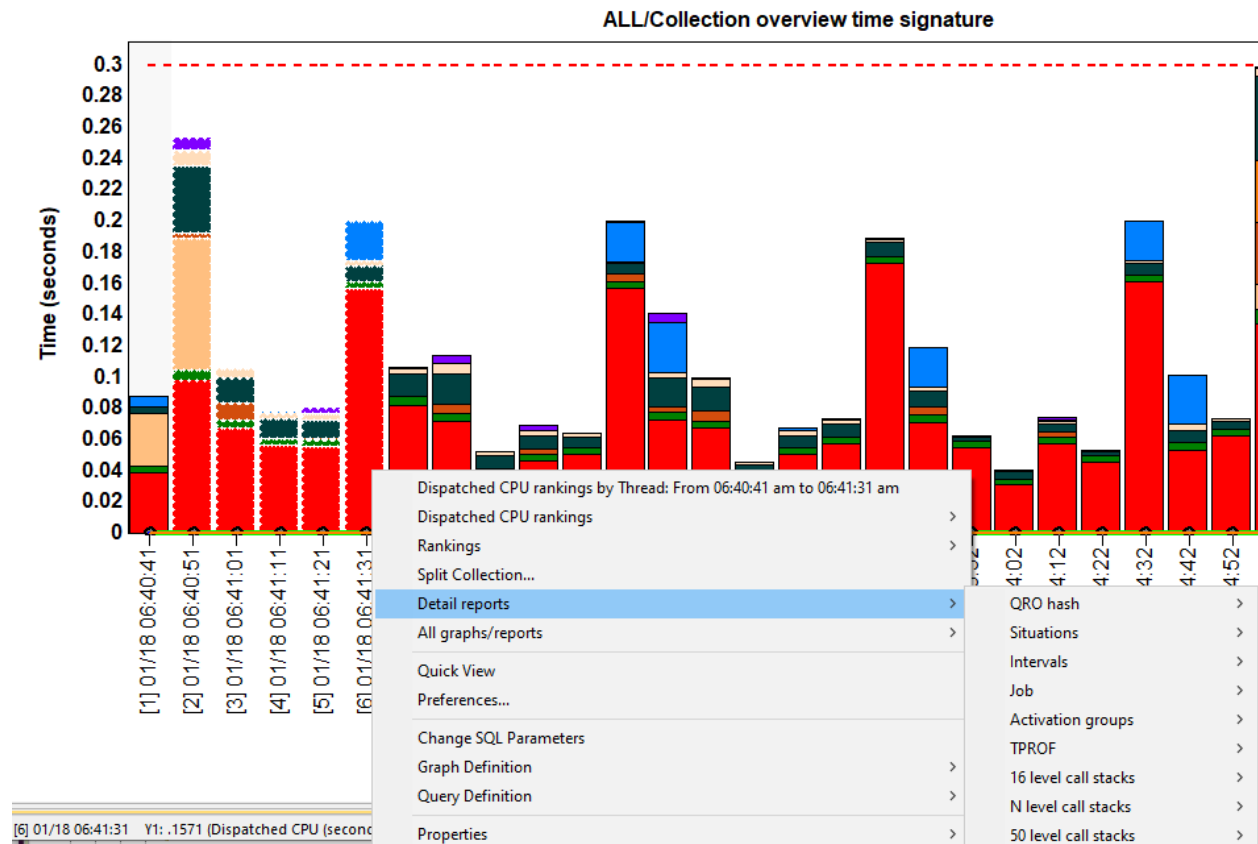
Tip: The first menu is Dispatched CPU rankings by thread. The bucket name shown will vary depending on which bucket in the graph was right-clicked on.

However, if 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 which will be filtered on the desired time range.

13.2 Drilling down to Detail reports

Another drilldown option from the Overview (or Rankings) 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. Some of the reports summarize data and some do not.

An example of this menu and list of report categories it contains is:



Detail reports menu options (truncated)

Note: These reports are based on either the single interval or time range selected. They may also be used from Rankings graphs.

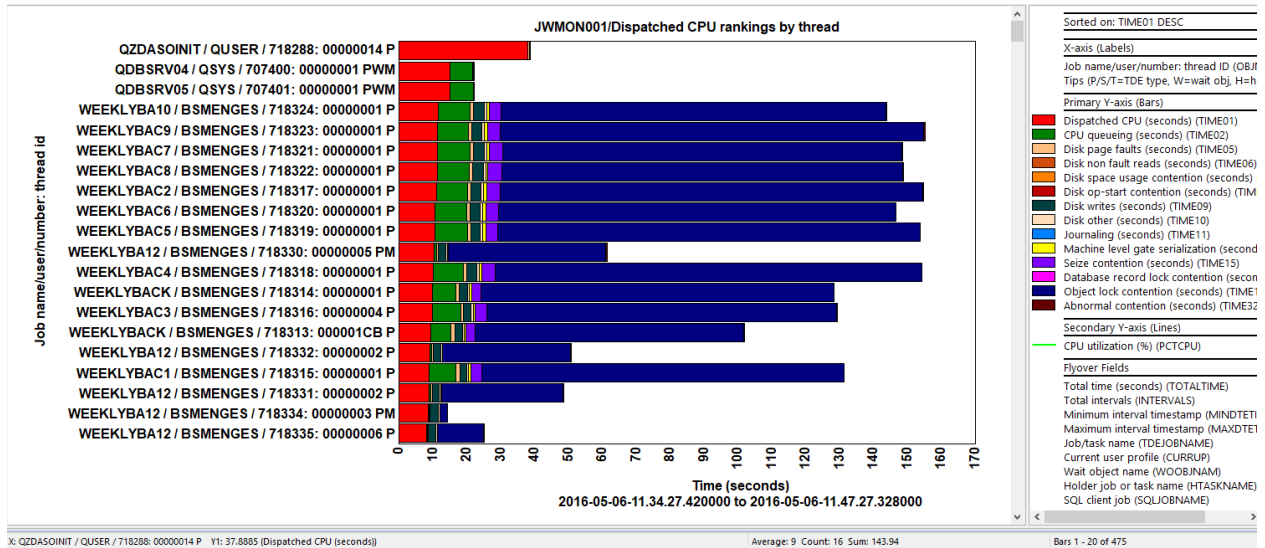
13.3 Split Collection option

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.

14 Ranking graphs

The ranking graphs are ordering jobs by a desired metric using a job grouping such as by thread, by job, by current user, etc. A maximum of 14 job groupings are used. Most rankings folder show all 14, but some may show fewer options depending on the graph.



Dispatched CPU rankings by thread

Note: If the **Collection Summary** analysis has not been ran yet, then the only rankings 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 subsystem
- by job type
- by job function
- by generic job | current user
- by thread | current user
- by qro hash
- by sql statement

Job groupings list for the Waits -> Dispatched CPU rankings folder

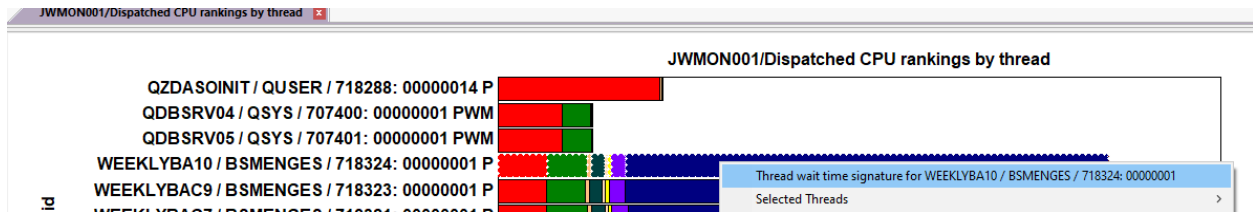
Ranking graphs are accessible from 1 of many possible ways:

- From one of the ranking graphs folders under the collection or popup menu.
- The 1st menu option (**<graph name> rankings by thread**) from an overview graph or selection over time graph.

- 2nd menu option (<graph name> **rankings**) in an overview graph or selection over time graph which shows the same graph but using one of the 14 grouping types.
- The **Rankings** menu from an overview graph
- The **Rankings filtered by <X>** menu from a rankings graph, or a list of jobs found in the **Interval Summary** interface.

14.1 Drilling down to Selection over time

When drilling down from a Rankings graph to a Selection over time graph the option will look something like this:

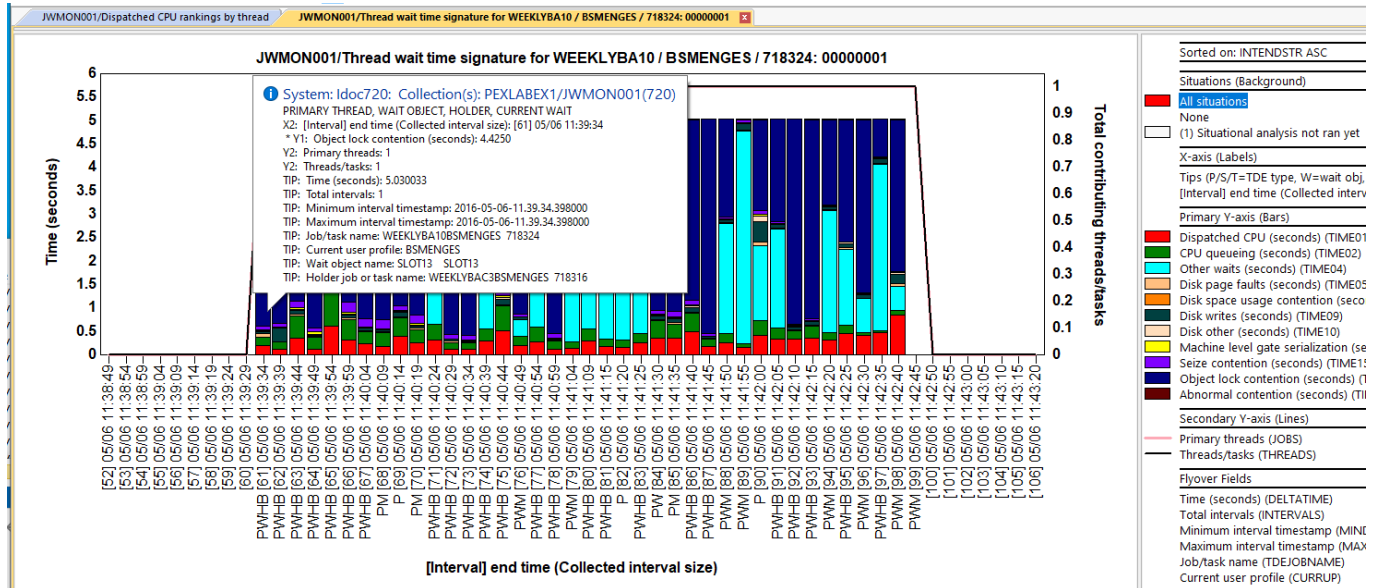


Drilldown from Rankings to Selection over time

15 Selection over time graphs

This graph type shows the currently selected thread (or job, generic job, user, etc) over time. The time groupings are configurable 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 see a big picture view of this job, use the clock icon to effectively “zoom out”.



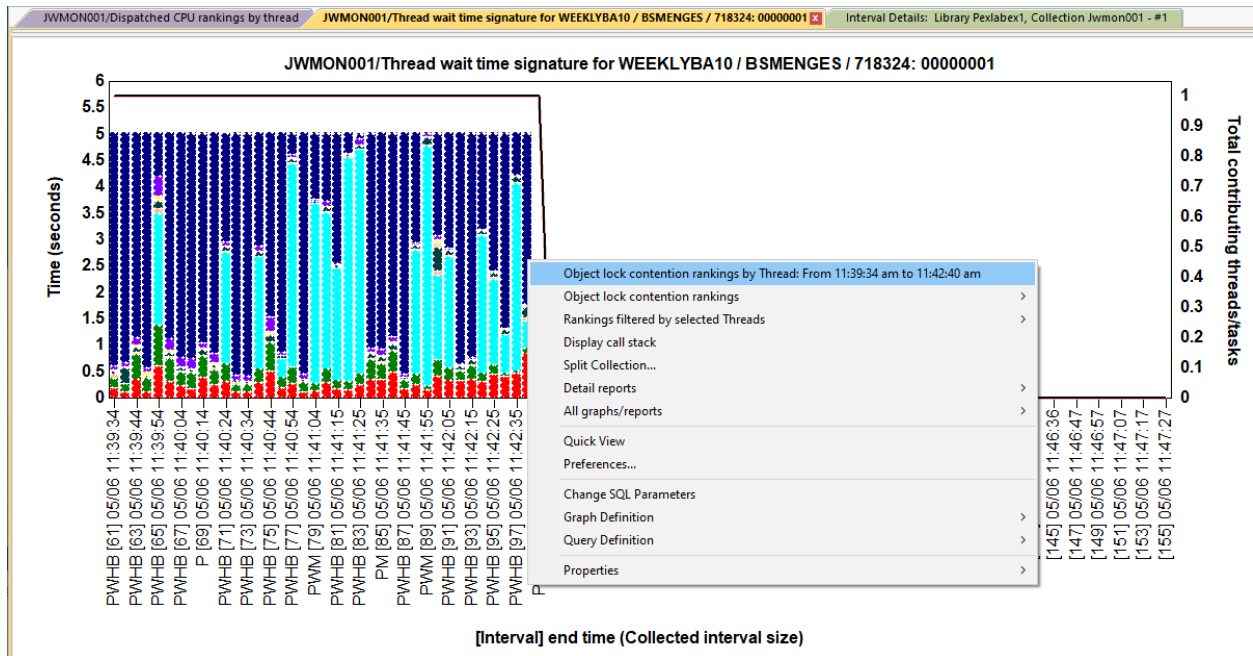
Thread wait time signature graph

Tip: If the flag indicator includes an “H” then a holder is available on that time intervals shown. 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.

15.1 Drilling up into Rankings

From the Selection over time 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.

An example follows:



Drilling up into Object lock contention rankings by thread

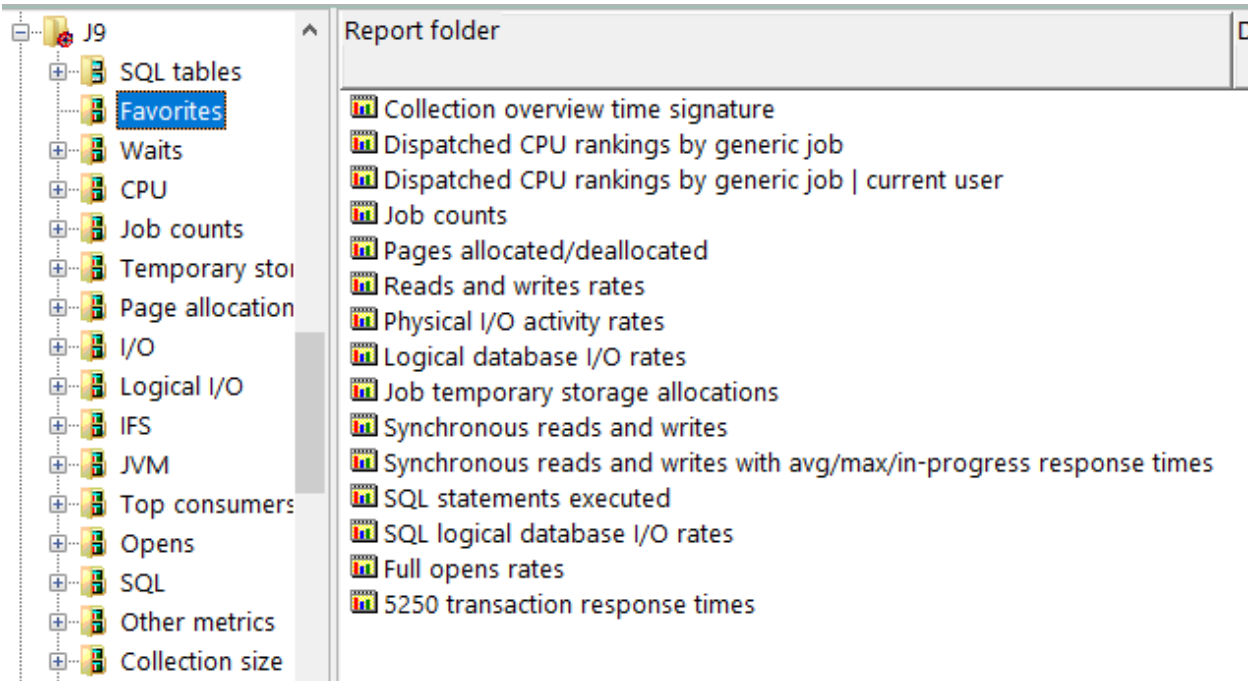
15.2 Display call stack

The Display Call Stack menu from a rankings or selection over time graph allows the user to quickly [go to the call stack](#) for the job and time indicated by the selection.

16 Favorites

This folder contains a list of graphs most 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 IBM i releases or require the Collection Summary analysis to be ran.



Favorites folder

Graph name	Notes
Collection overview time signature	
Collection overview time signature with workload capping	Requires Collection summary and QAPYJWTDE – TRESERVE11 must contain values > 0.
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	Requires Collection summary
Read and writes rates	
Physical I/O activity rates	
Logical database I/O rates	
Job temporary storage allocations	Requires Collection summary
Synchronous reads and writes	Requires Collection summary
Synchronous reads and writes with avg/max/in-progress response times	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	

17 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.

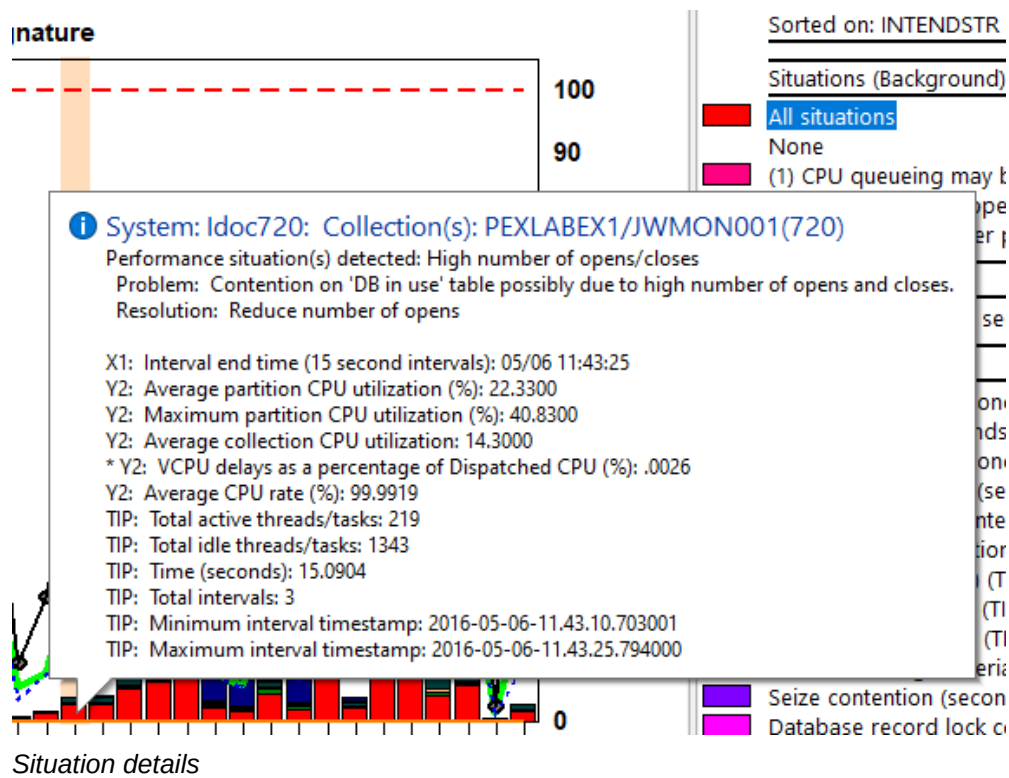
Report folder	Description
IBMJWT13	
SQL tables	
Favorites	
Waits	
CPU	
Job counts	
Temporary :	
Page allocat	
I/O	
Logical I/O	
IFS	
Top consum	
Opens	
SQL	
Other metric	
Collection si	
Server-side	
User-defin	
IBMJWT02	
IBMJWT01	
IBMJWT48	
IBMJWT43	
IBMJWT42	
IBMJWT41	
IBMJWT40	
Ts025307ah	
Ts025307aj	
Ts026565ab	
Ts026565ac	
Ts039464ac	
Ts039464ad	
	Collection overview time signature
	Collection overview time signature with workload capping
	Collection overview time signature with dispatched CPU breakd...
	Virtual CPU delays
	Clients + Workers wait time
	Collection overview time signature with max waits in-progress
	Current wait duration time signature with max waits in-progress
	Disk time signature with max disk waits in-progress
	Current wait duration time signature
	Seizes and locks time signature
	Disk time signature
	Journaling time signature
	Communications time signature
	Time waiting on objects
	CPU queueing breakdown
	Workload capping delay
	Wait counts
	Average wait times
	Average CPU times
	Average disk wait times
	Dispatched CPU rankings
	Clients + Workers wait time rankings
	Clients Only wait time rankings
	CPU queueing breakdown rankings
	Workload capping delay rankings
	Time waiting on objects rankings
	Wait counts rankings
	Average wait times rankings
	Average CPU times rankings
	Average disk wait times rankings
	Collection totals
	Objects waited on
	Ranks jobs by dispatched CPU usage
	Ranks QSQRVR jobs (workers) or their clients jobs by dispatched CPU usage
	Ranks client jobs only by dispatched CPU usage
	Ranks jobs by CPU queueing and breaks includes workload capping delays
	Ranks jobs by workload capping delay
	Shows the jobs experiencing the most time waiting on wait objects
	Ranks jobs by wait bucket counts
	Ranks jobs by average wait bucket time (all interesting waits in seconds)
	Ranks jobs by average CPU/CPU queueing bucket times (in microseconds)
	Ranks jobs by average disk wait bucket time (in milliseconds)
	Pie charts showing wait bucket contributions across the entire collection
	Reports displaying the objects waited on in various ways

Waits folder

17.1 Collection overview time signature

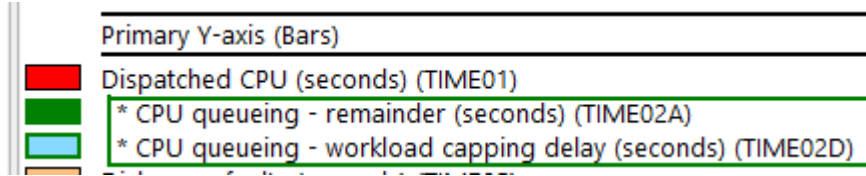
This graph shows CPU time and the ["interesting" wait bucket](#) times added together across all jobs on the system. In the example below, a user could right-click intervals where the dark blue object lock time is showing and drill down to view the jobs which experienced the highest amount of object lock contention time for the selected time period.

This graph is also displaying situations in the background for 1 interval. Placing the mouse over the bar will provide more information. **Note:** Situations will only appear if the Collection Summary analysis and Situational Analysis have been ran first.



17.2 Collection overview time signature with workload capping

This graph is the same as the previous one but divides up the CPU queueing bucket into 2 parts: Workload capping delays and remainder.



Note: This graph only appears if workload capping delay times exist in the collection AND **Collection Summary** analysis has been ran.

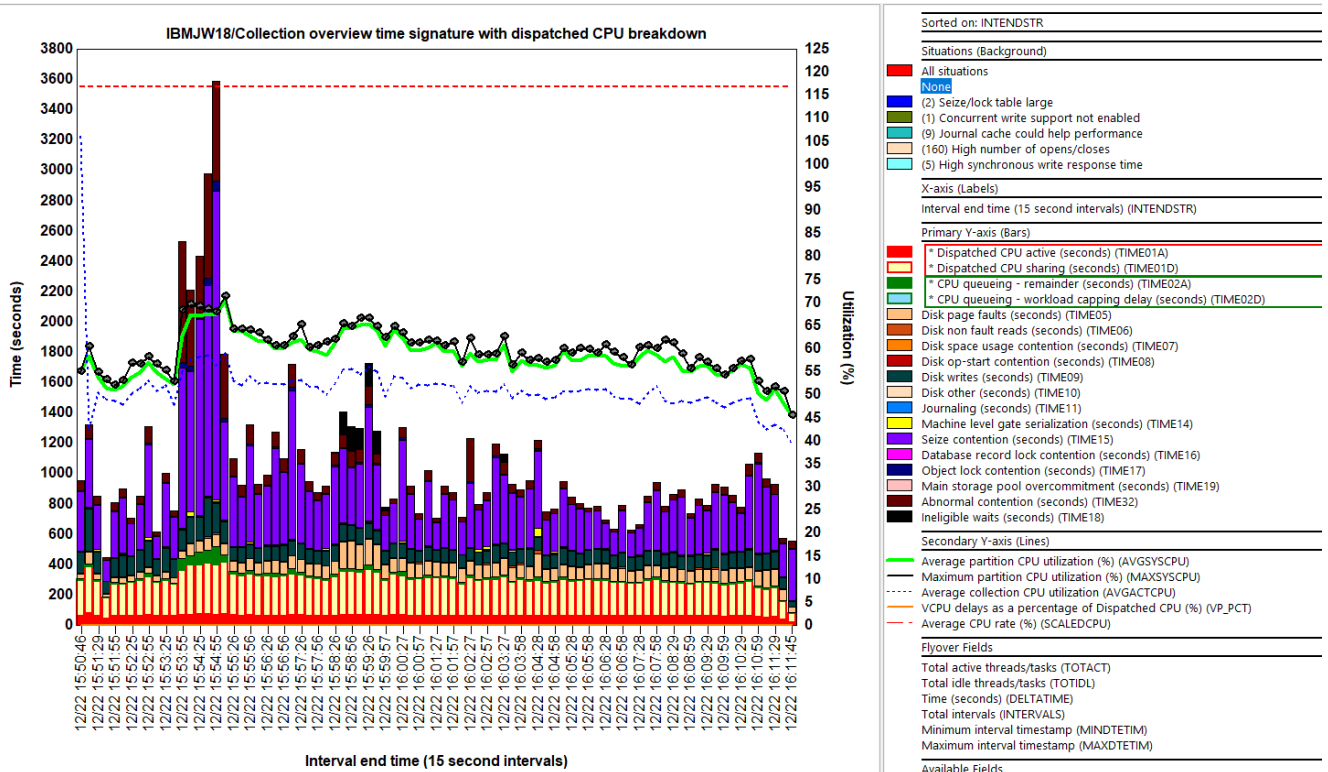
17.3 Collection overview time signature with dispatched CPU breakdown

This graph is identical to the Collection overview time signature graph except the Dispatched CPU time is divided into 2 parts:

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 indicate a performance problem. It is provided here for advanced users.

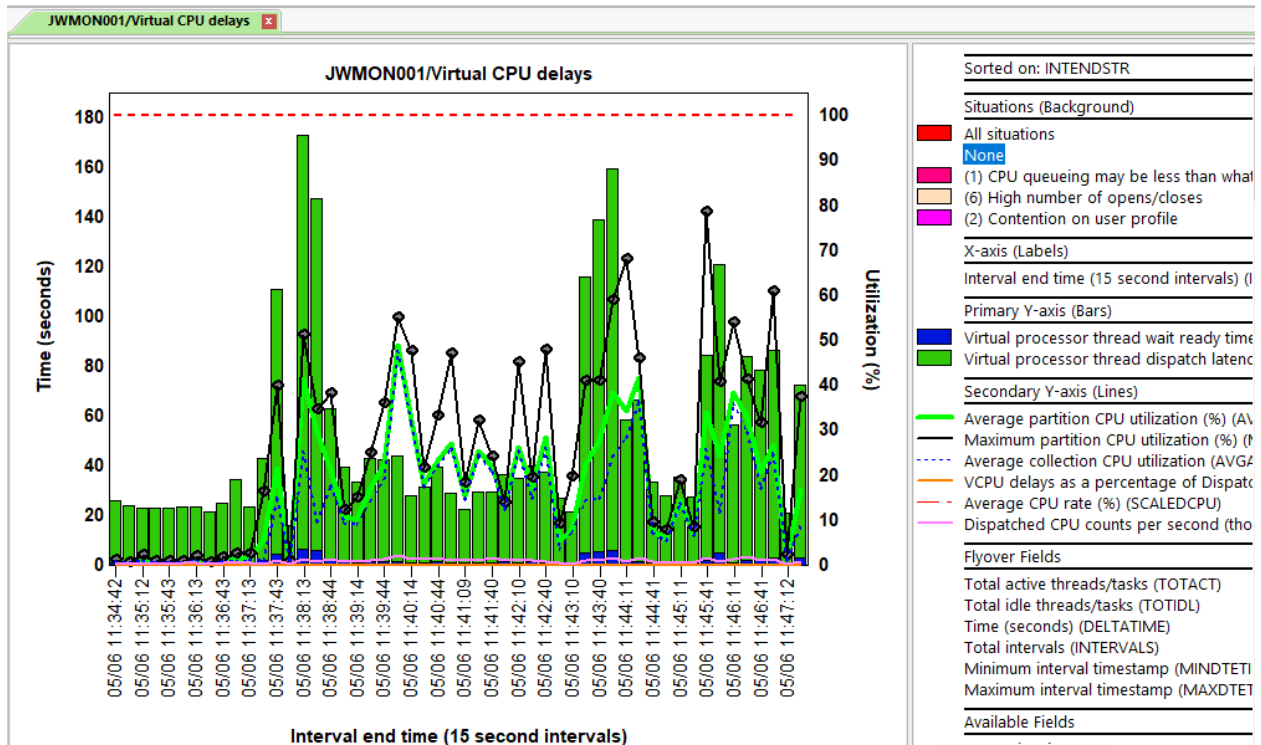
Note: If workload capping delays exist, then the CPU queueing bucket is also divided into 2 pieces.



Collection overview time signature with dispatched CPU breakdown

17.4 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.



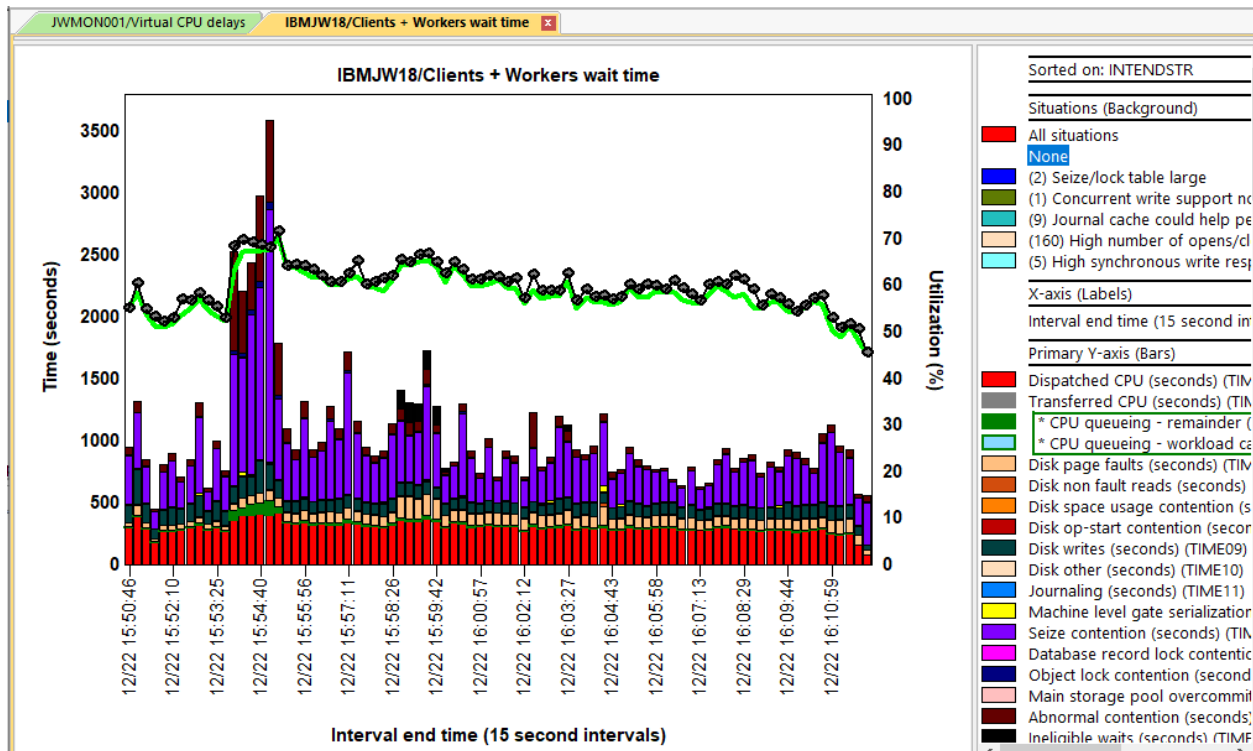
Virtual CPU delays

17.5 Clients + Workers wait time

This graph shows ONLY the contributions of QSQRVR jobs and the client jobs that initiated them. This graph will only appear if the Collection Summary analysis has been executed.

Tip: This can be used to compare with the Collection overview time signature to visually see how much of the overall time is related to this type of work.

Tip #2: To drill-down and see only these same contributions use the Rankings -> Waits -> Clients + Workers rankings or Rankings -> Waits -> Clients Only Rankings menu options.

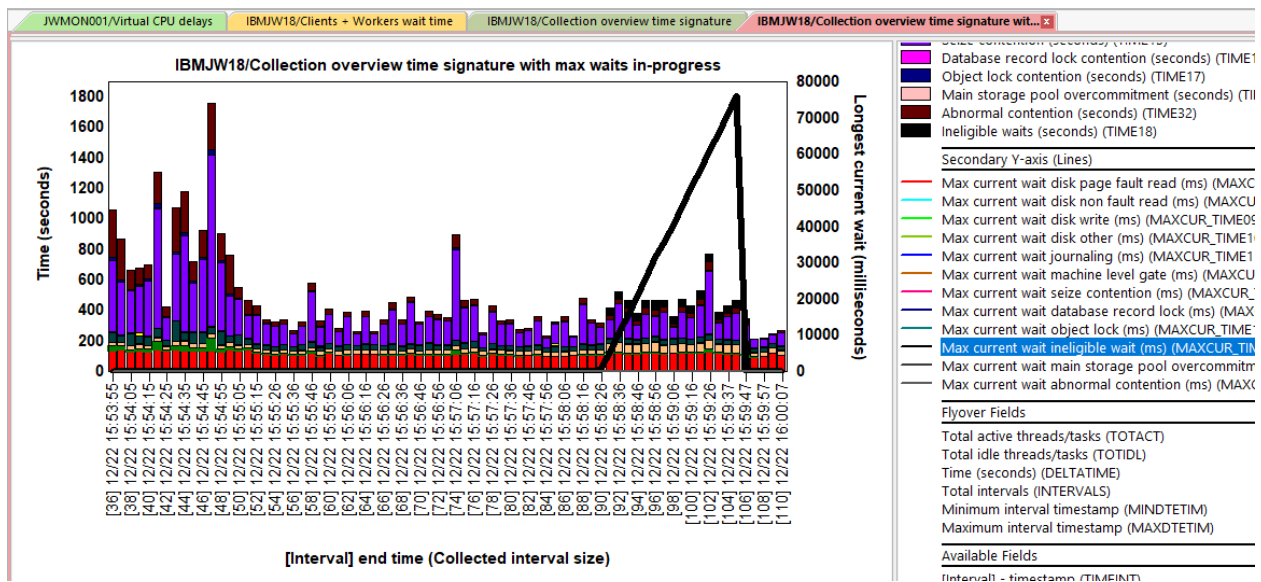


Clients + Workers wait time

17.6 Collection overview time signature with max waits in-progress

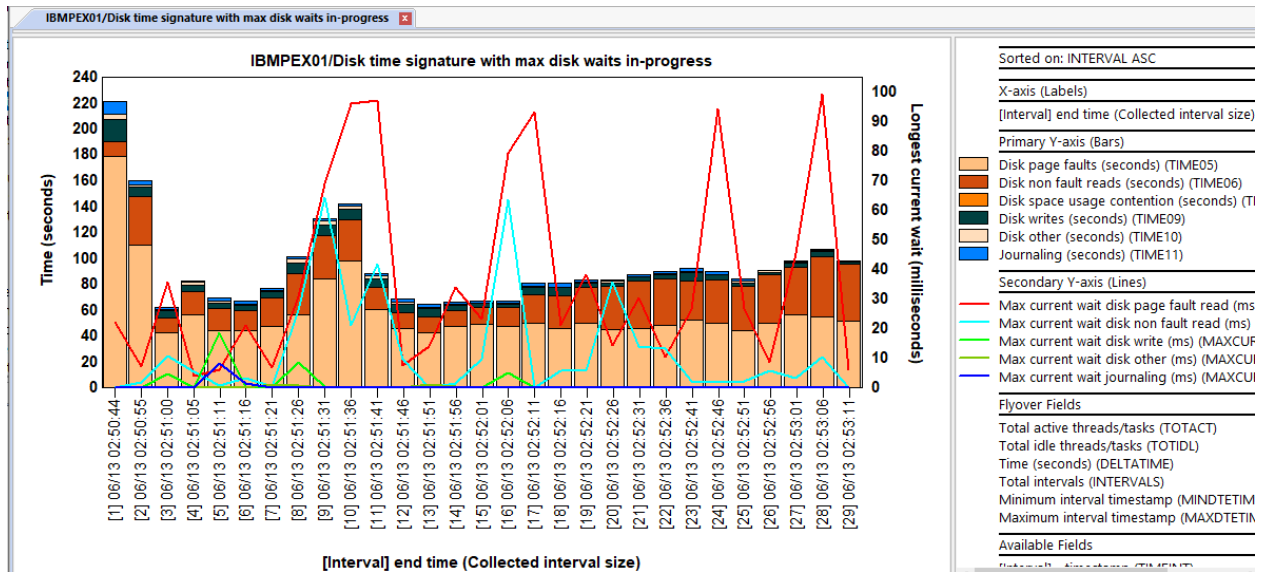
This graph is the same as the Collection Overview Time Signature except the longest waits that occurred in any job are shown on the 2nd Y-axis for any of the "interesting" types of waits. These longest waits are captured from the current wait duration field (CURRWTDUR) in the QAPYJWTD file.

If the values are increasing over multiple intervals this can indicate a performance problem. In the below example, the increasing Y2 line shows job(s) that were likely ineligible to run for about a minute.



Collection overview time signature with max waits in-progress

17.7 Current wait duration time signature with max waits in-progress



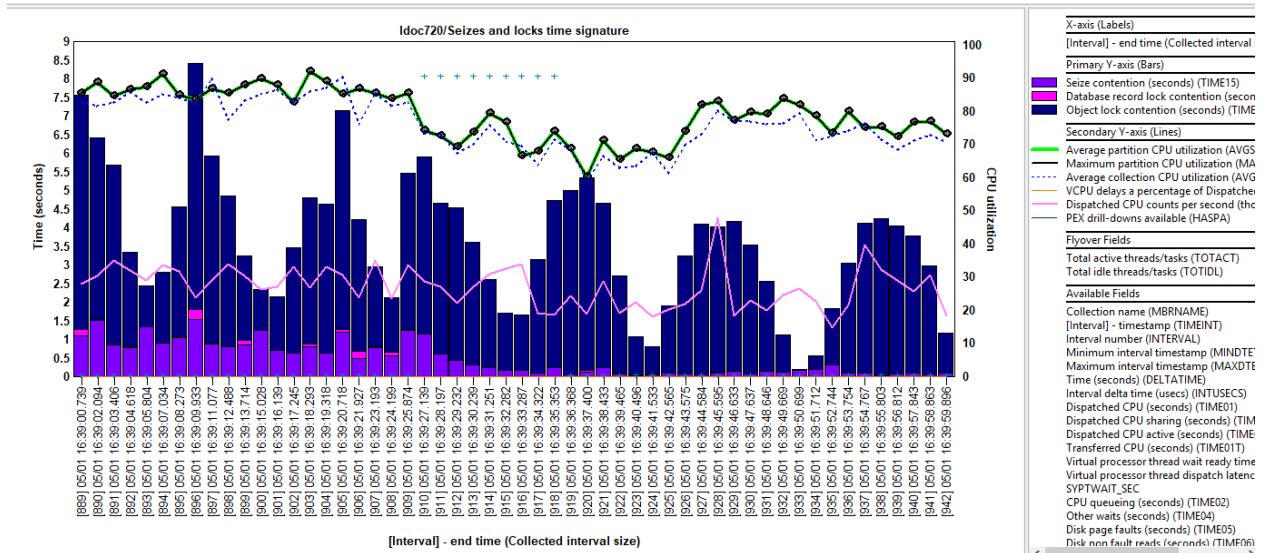
Disk time signature with max disk waits in-progress

17.9 Current wait duration time signature

This graph is identical to the other “current wait duration” graph discussed previously but shows CPU utilization on the 2nd Y-axis.

17.10 Seizes and locks time signature

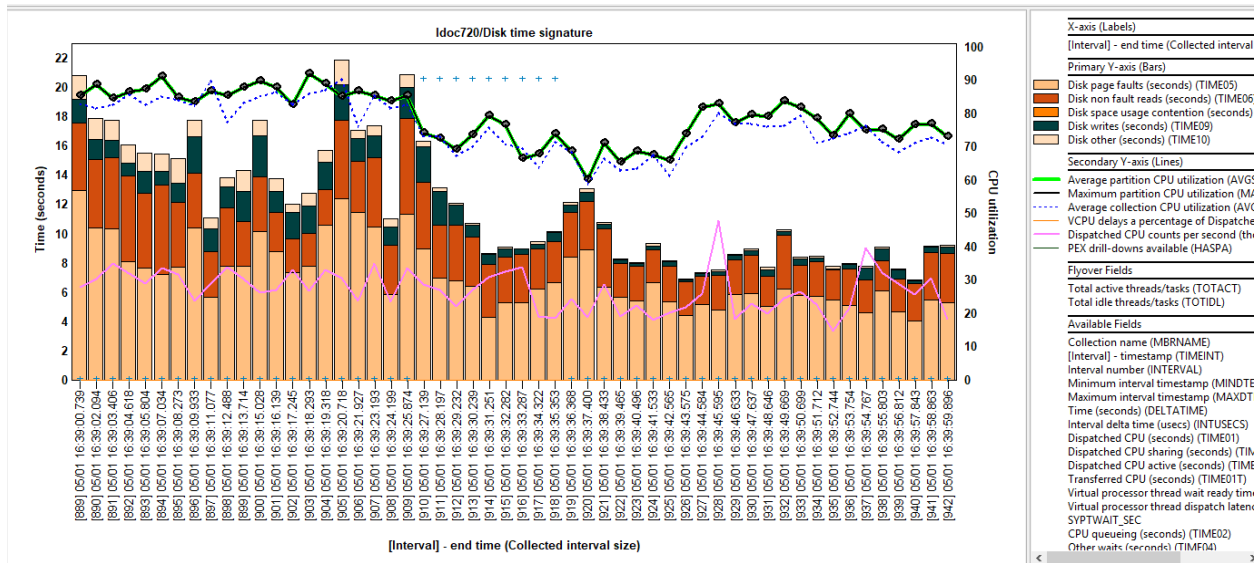
This graph shows only the seizes, record locks and object lock times as well as CPU utilization on the Y2-axis.



Seizes and locks time signature

17.11 Disk time signature

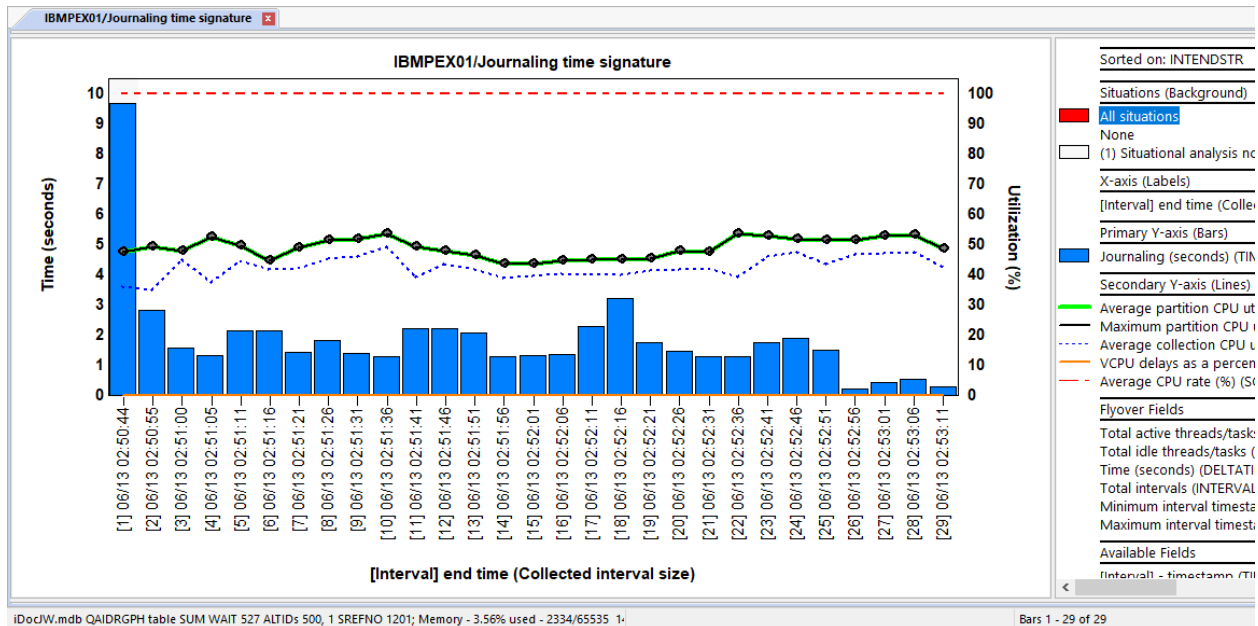
This graph shows only wait buckets associated disk related times. (buckets 05-10)



Disk time signature

17.12 Journaling time signature

This graph shows only wait buckets associated with journaling time. (buckets 11, 20)

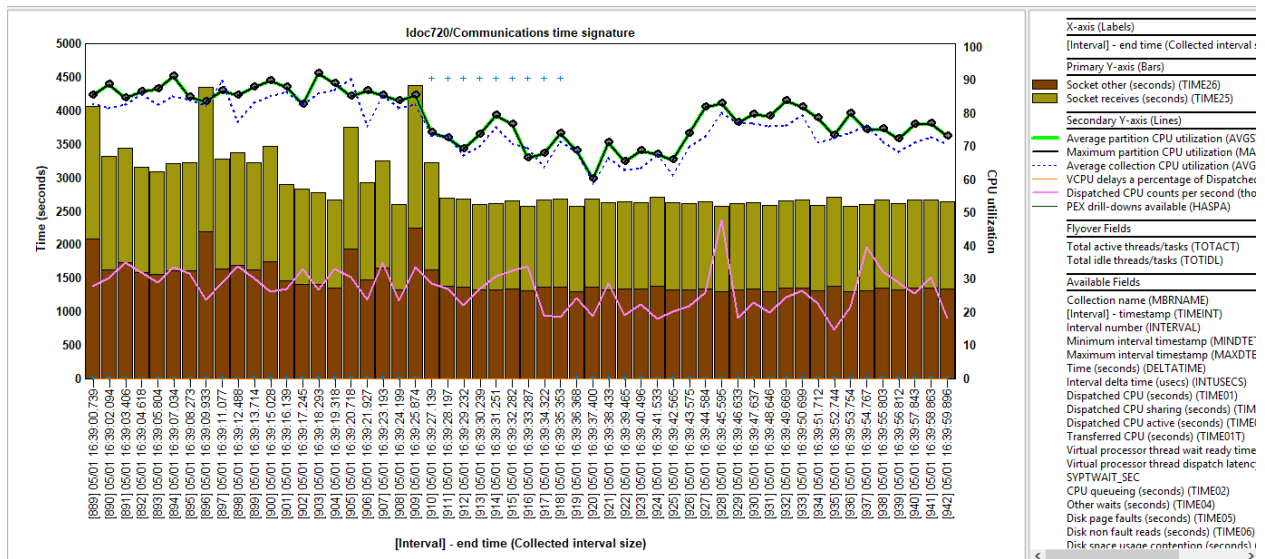


Journaling time signature

17.13 Communications time signature

These waits indicate time waiting to receive or send data or other types of socket waits. (Buckets 24-26)

An example of a socket receive is what a QZRCRSVS 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.

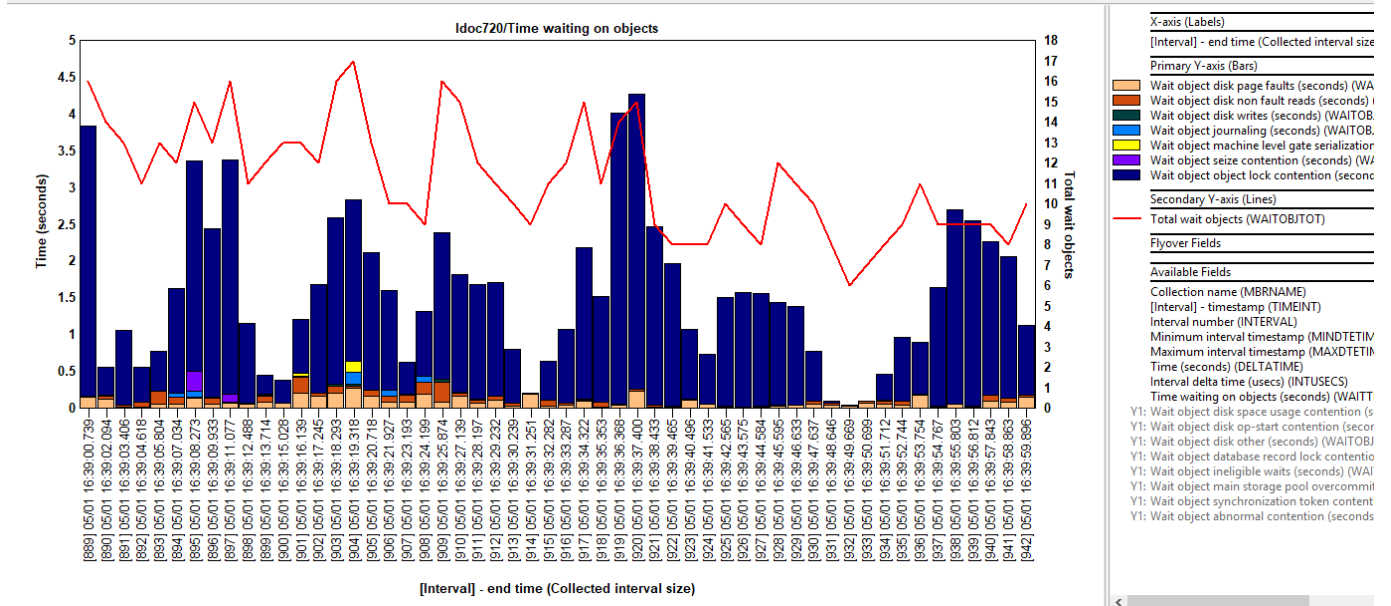


Communications time signature

17.14 Time waiting on objects

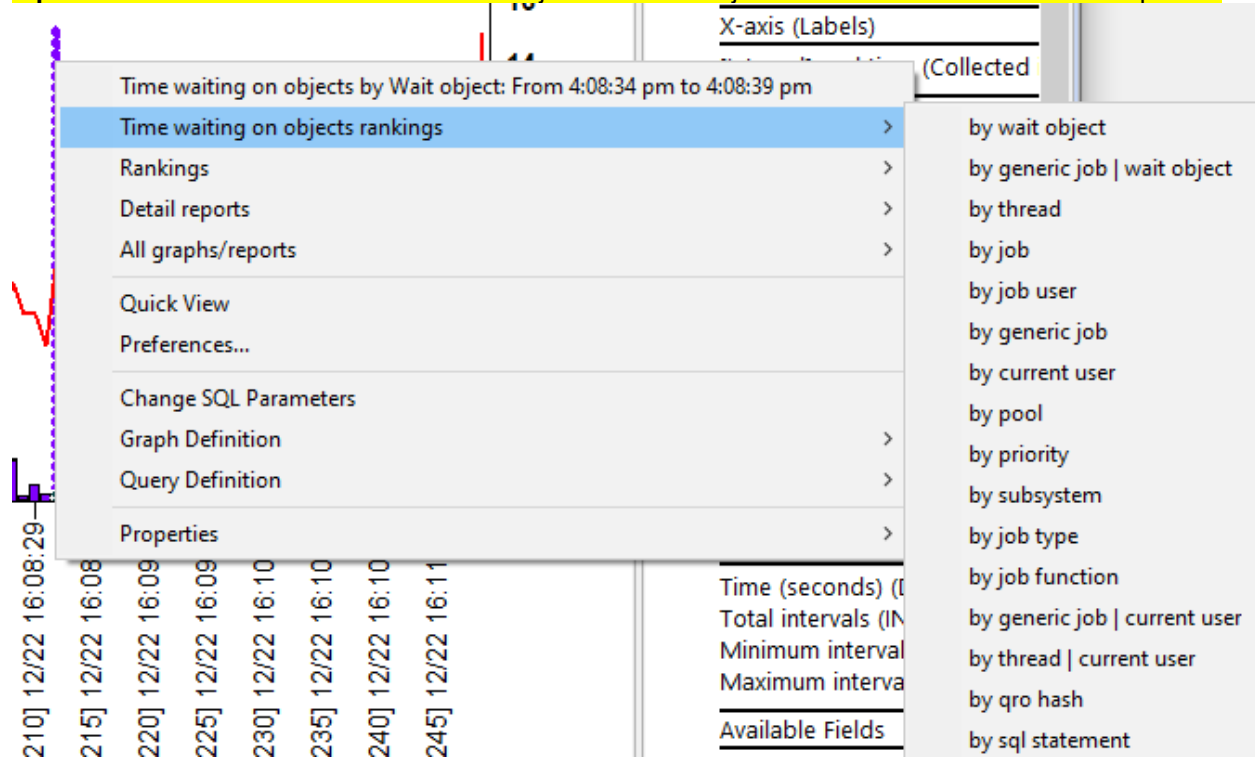
This graph adds up the current wait duration times only where a wait object was associated with each “interesting” wait bucket times and graphs the results. The y2-axis displays the total wait objects found per interval.

Note: This graph requires the Collection summary analysis to be ran to appear!



Time waiting on objects

Tip: You can drill down from here to see the jobs and wait objects detected for the desired time period.

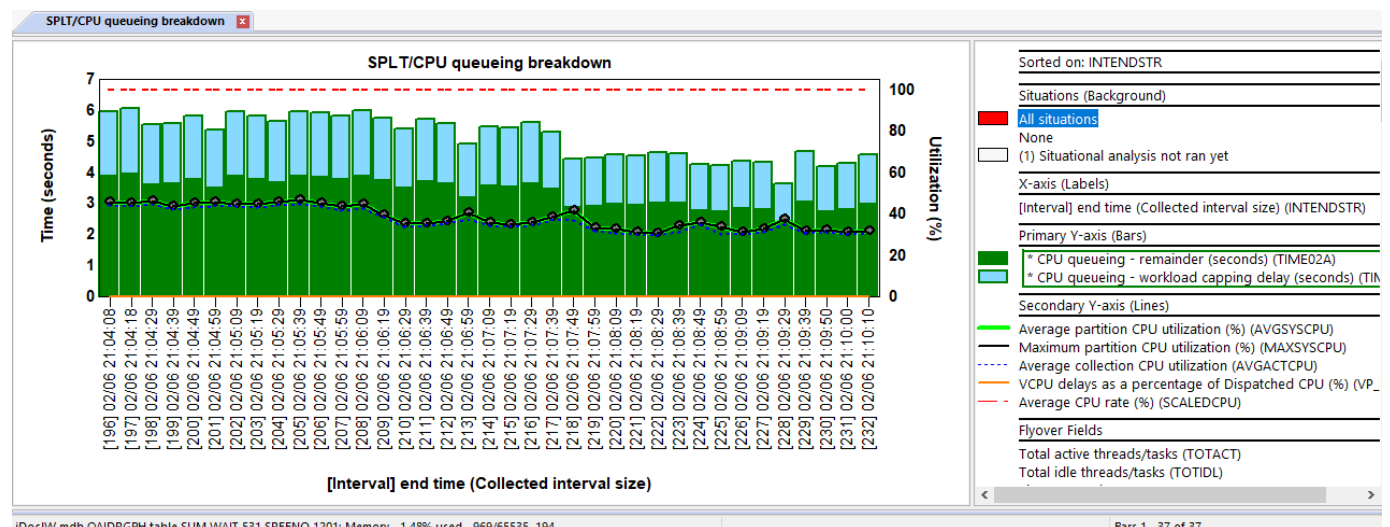


Time waiting on objects graph menu options

17.15 CPU queueing breakdown

This graph will focus just on the CPU queueing time and shows the breakdown between workload capping delays and the rest of the CPU queueing time.

Note: This graph requires that workload capping delays exist, and the **Collection summary** analysis is ran to appear!

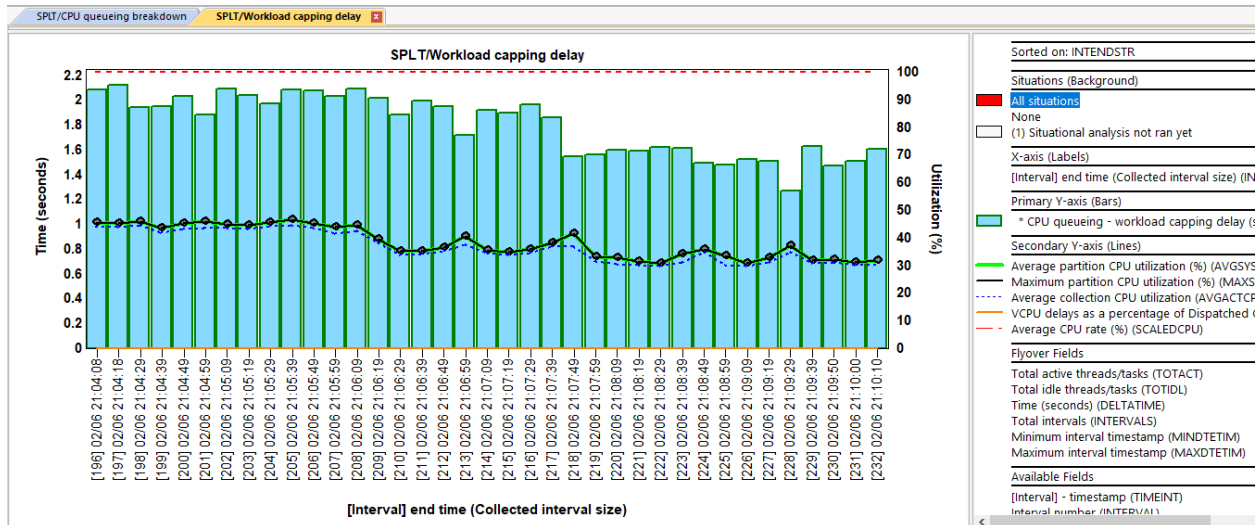


CPU queueing breakdown

17.16 Workload capping delays

This graph only shows workload capping delay times.

Note: This graph requires that workload capping delays exist, and the **Collection summary** analysis is ran to appear!

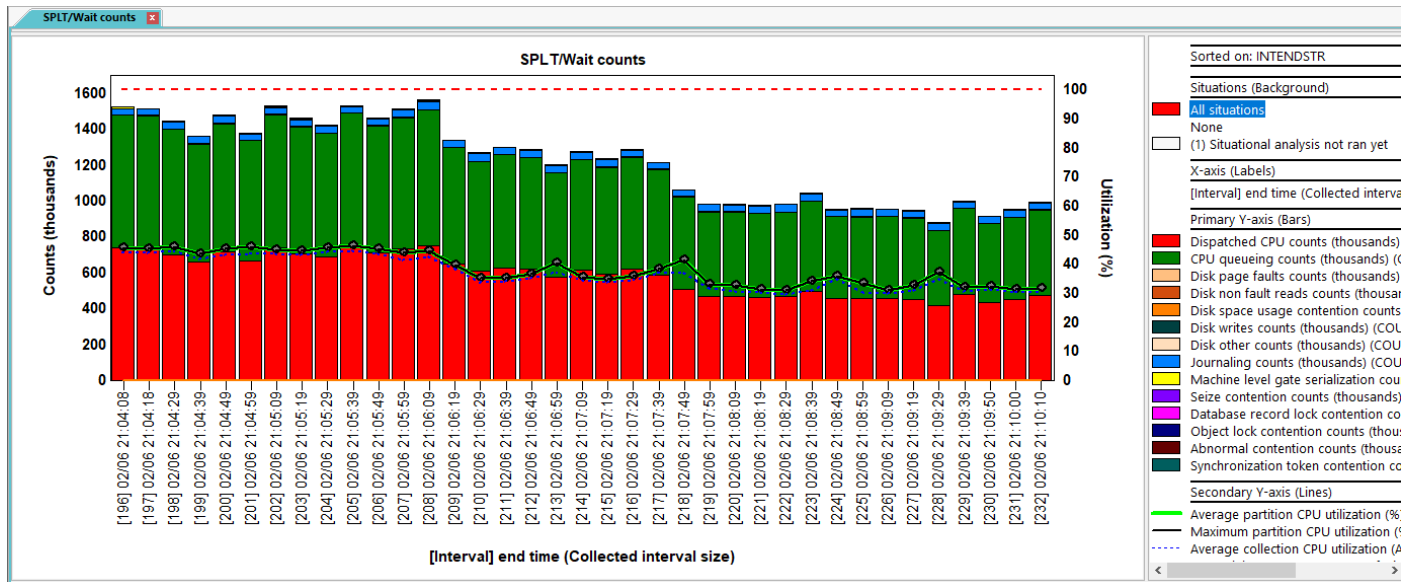


Workload capping delay

17.17 Wait counts

This graph displays the “interesting” wait buckets’ number of distinct waits that occurred using the metric counts per second (In thousands) for each.

Note: This graph requires the **Collection summary** analysis is ran!

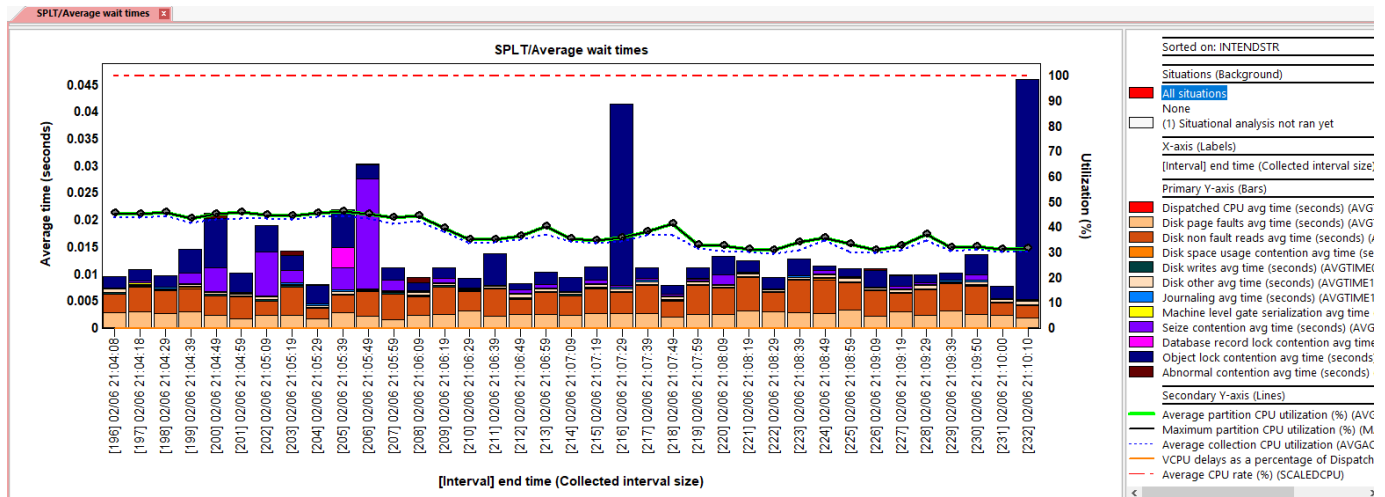


Wait counts

17.18 Average wait times

This graph displays the average wait time for each of the “interesting” wait buckets. This is the average duration of each wait in seconds.

Note: This graph requires the **Collection summary analysis** is ran!

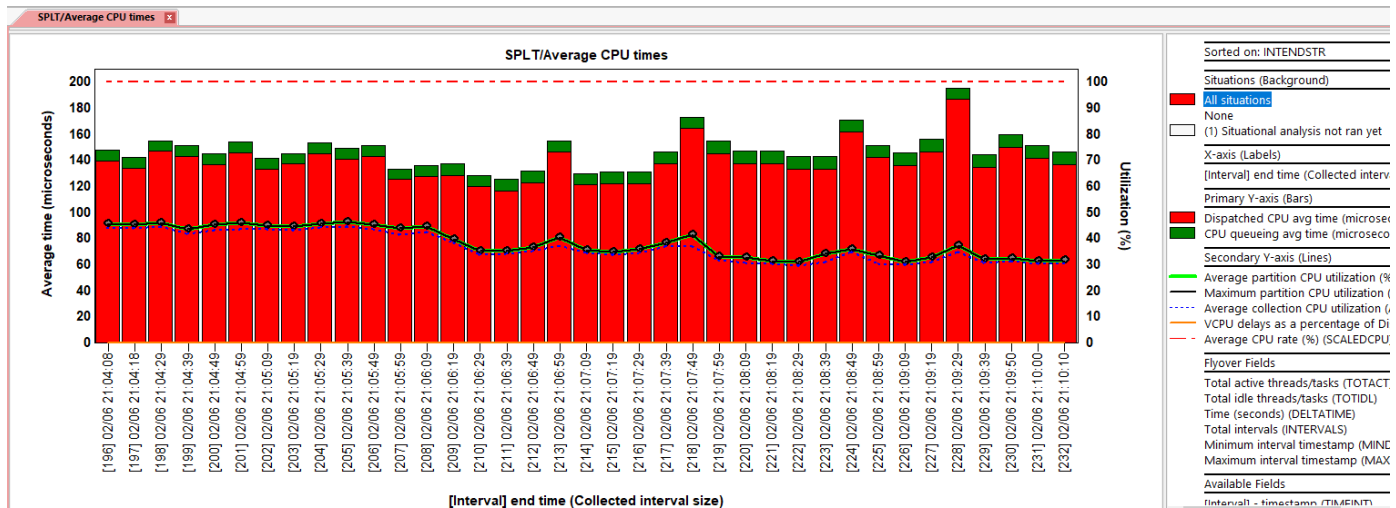


Average wait times

17.19 Average CPU times

This graph displays the average dispatched CPU and CPU queueing bucket times in microseconds.

Note: This graph requires the **Collection summary analysis** is ran!

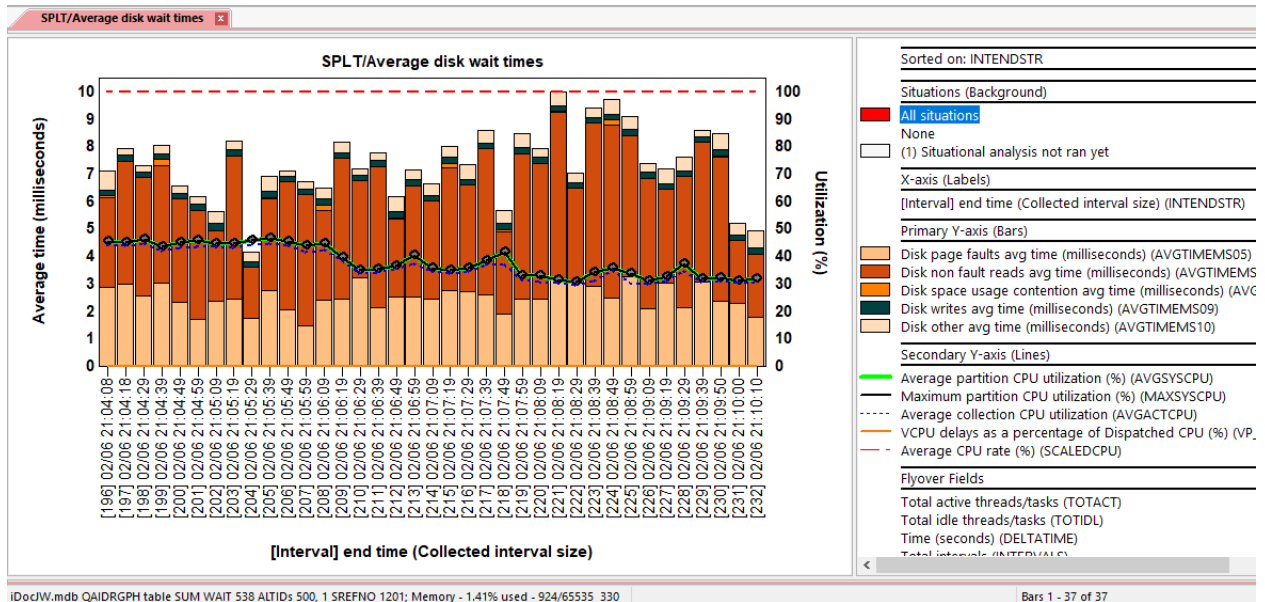


Average CPU times

17.20 Average disk wait times

This graph displays the average wait times for each disk related wait bucket in milliseconds (buckets 5-10.)

Note: This graph requires the **Collection summary analysis** is ran!

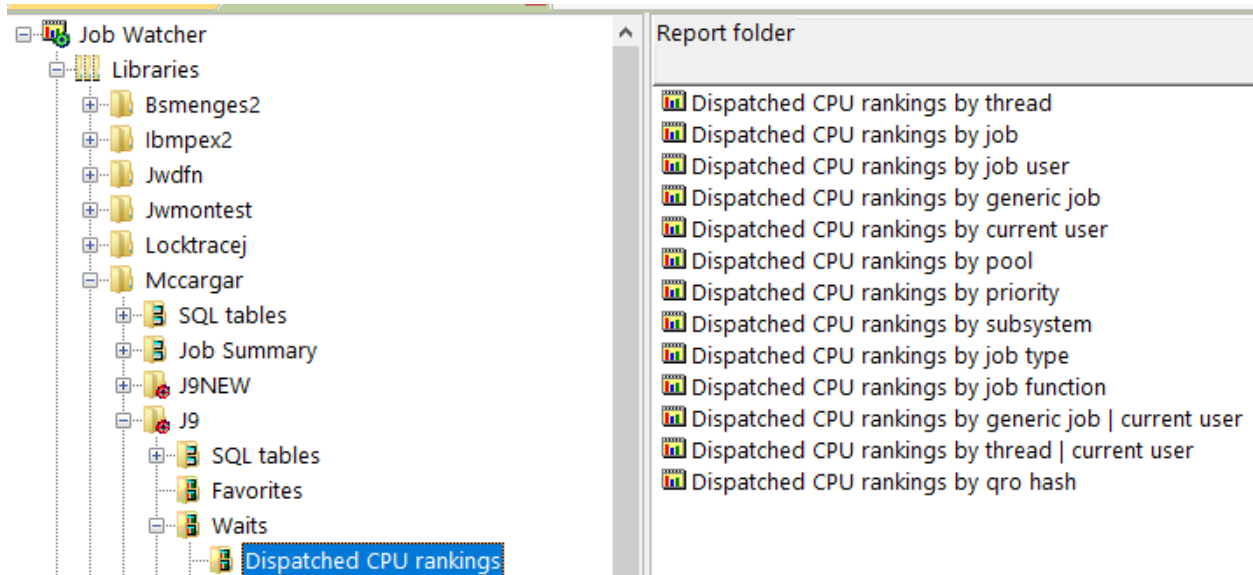


Average disk wait times

17.21 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.

Tip: These graphs only show jobs that used CPU during the collection.

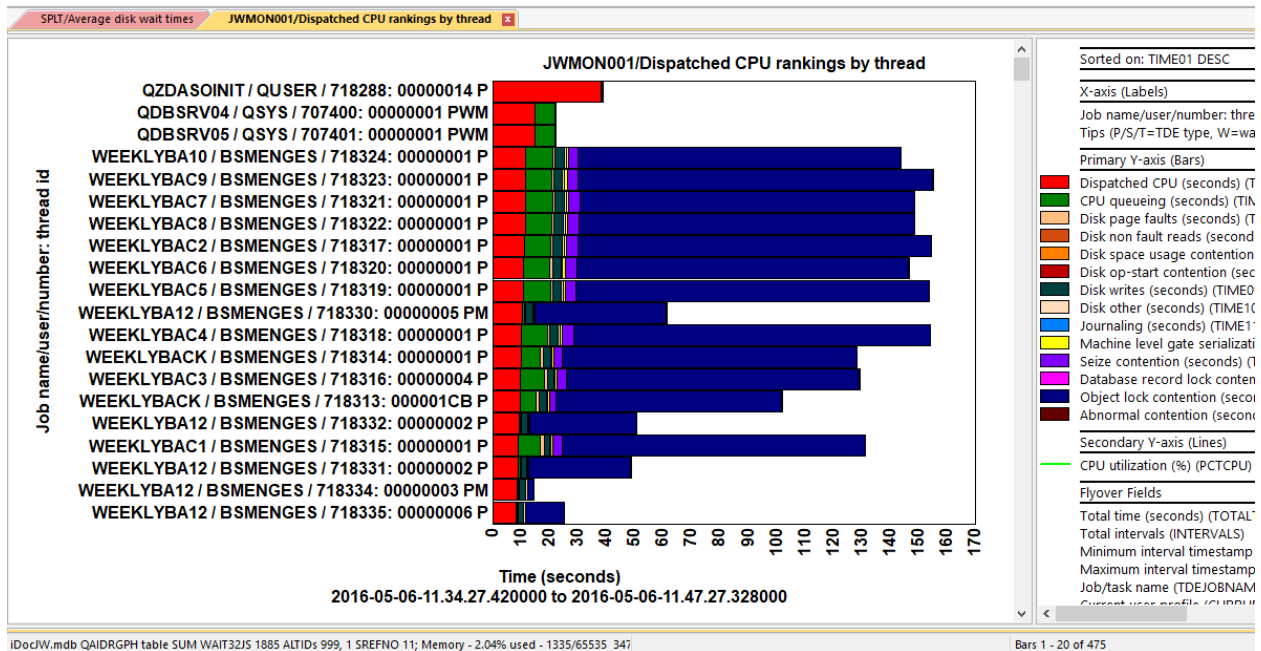


Waits -> Dispatched CPU Rankings

Note: Only the Dispatched CPU rankings by thread graph will exist in this folder unless the **Collection summary** analysis has been ran.

17.21.1 Dispatched CPU rankings by thread

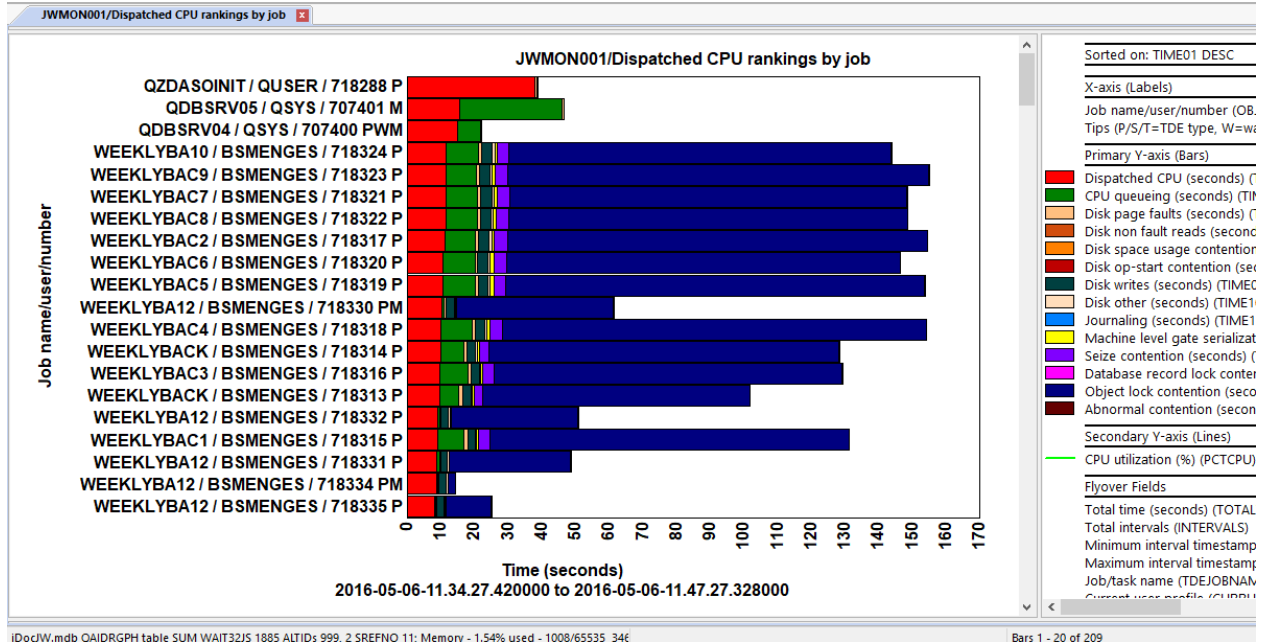
This graph groups the selected time period's wait bucket data by thread or task and sorts by Dispatched CPU.



Dispatched CPU Rankings by thread

17.21.2 Dispatched CPU rankings by job

This graph groups the selected time period's wait bucket data by primary thread or task and sorts by Dispatched CPU.

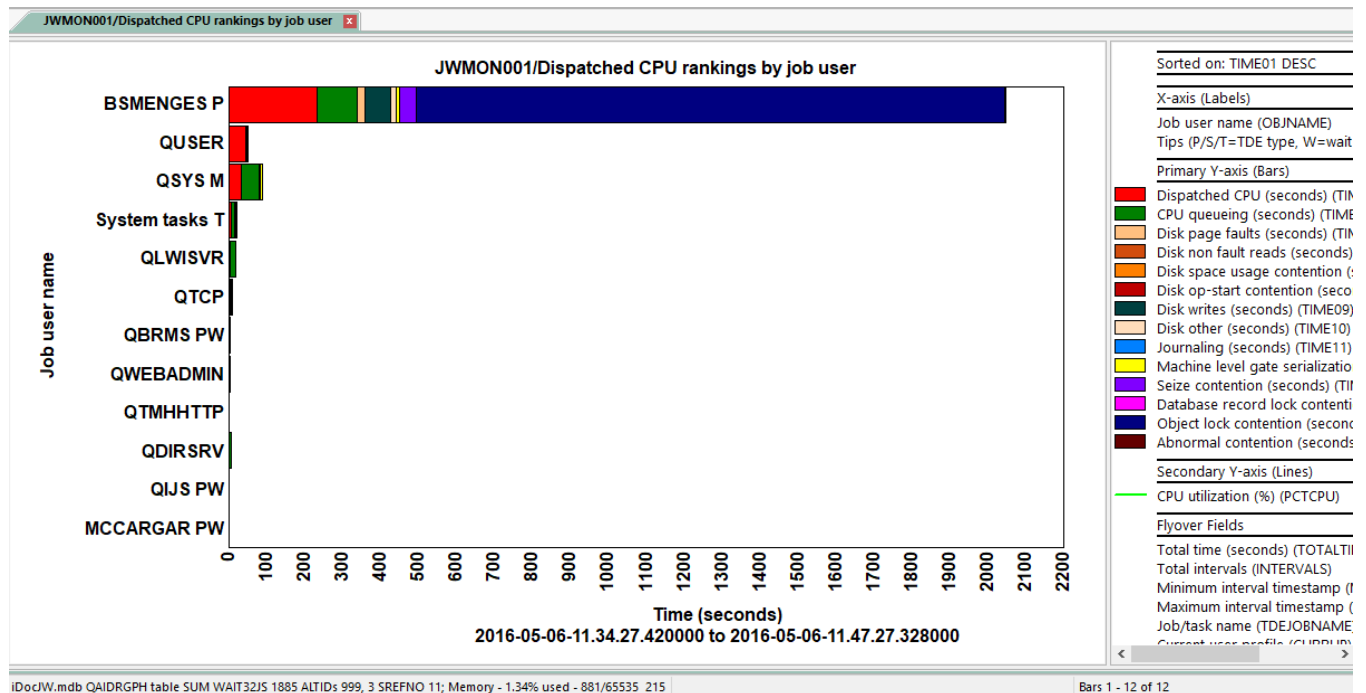


Dispatched CPU Rankings by job

17.21.3 Dispatched CPU rankings by job user

This graph groups the selected time period's wait bucket data by job user (not current user profile) and sorts by Dispatched CPU.

Note: All system tasks are grouped together into one “System tasks” record within this report.

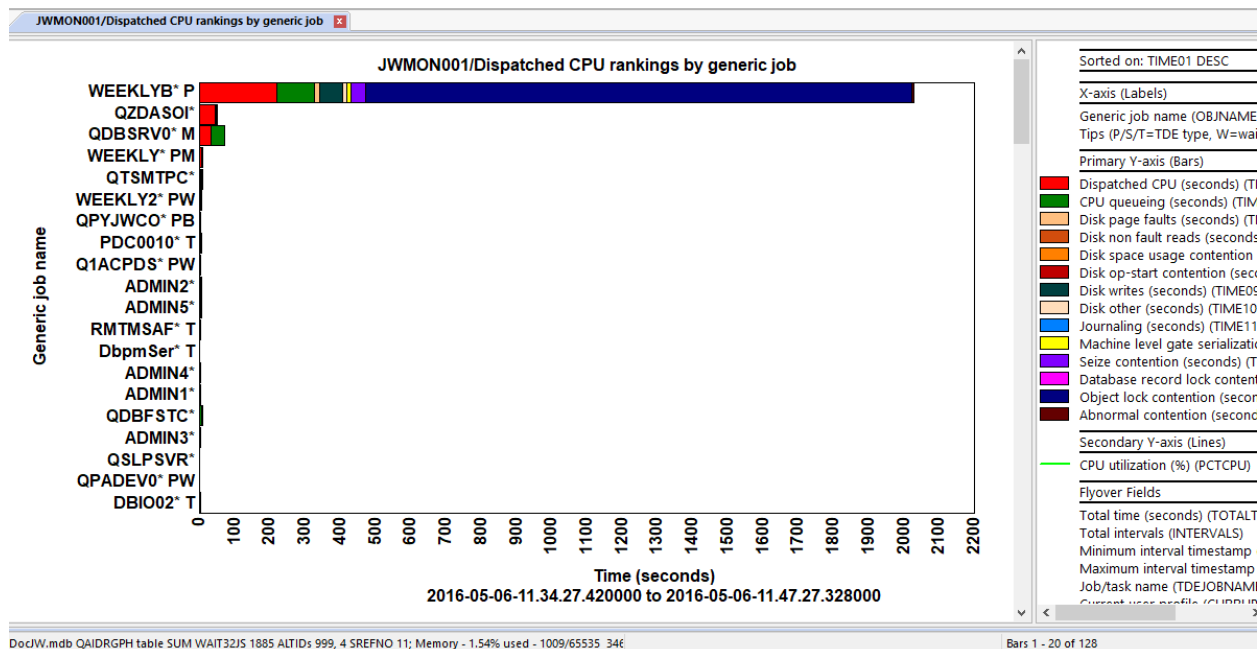


Dispatched CPU rankings by job user

17.21.4 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 name is controlled via the **Preferences - > Data Viewer** options shown below:

Name length for generic name grouping graphs: Start position:

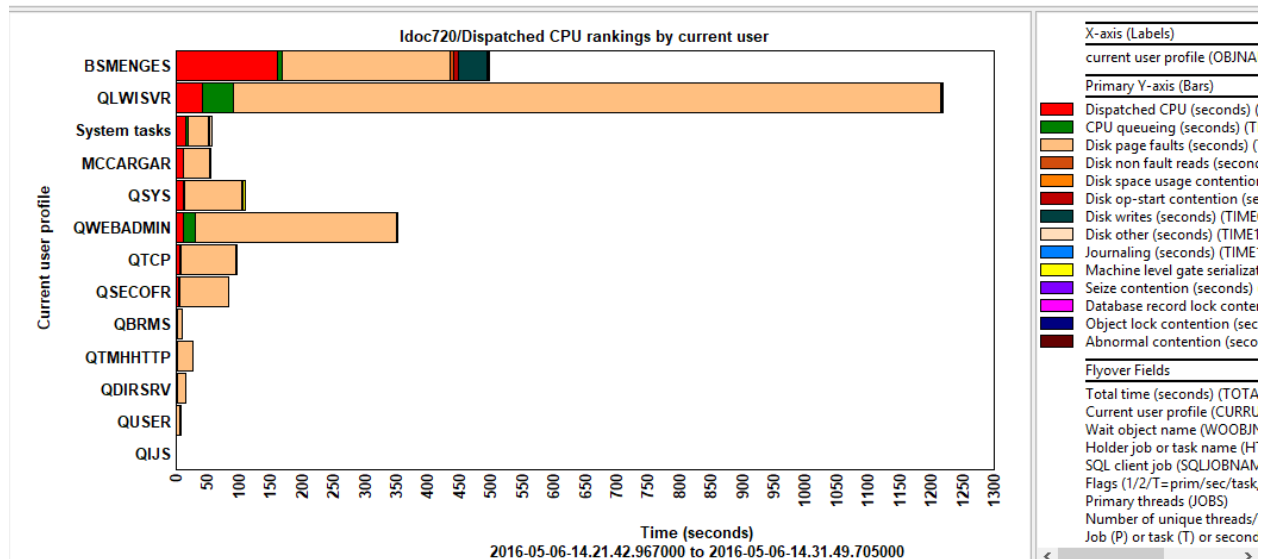


Dispatched CPU rankings by generic job

17.21.5 Dispatched CPU rankings by current user

This graph groups the selected time period's wait bucket data by current user profile and sorts by Dispatched CPU.

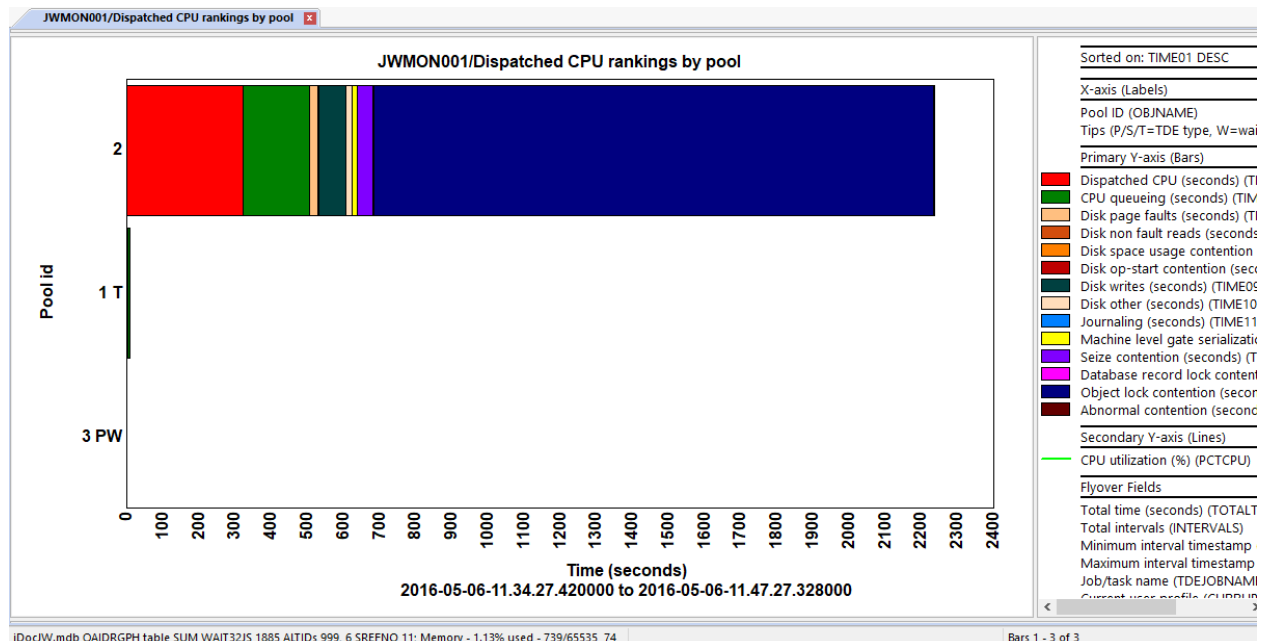
Note: All system tasks are grouped together into one “System tasks” record within this report.



Dispatched CPU rankings by current user

17.21.6 Dispatched CPU rankings by pool

This graph groups the selected time period's wait bucket data by memory pool and sorts by Dispatched CPU.

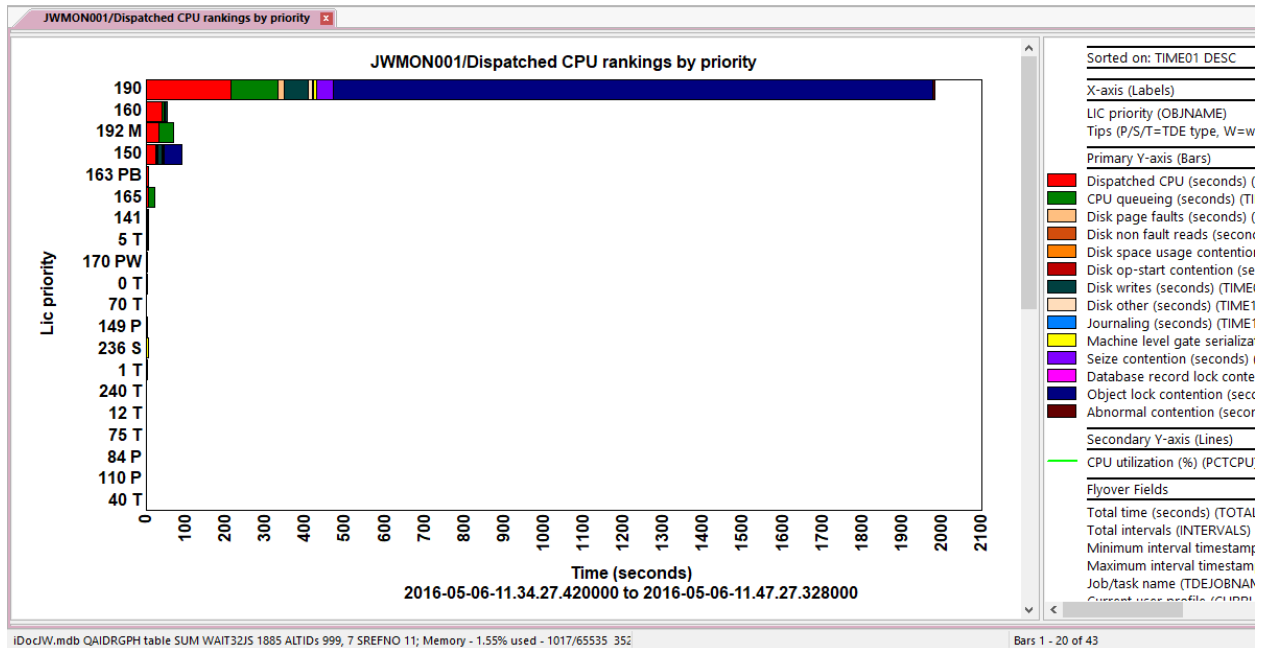


Dispatched CPU rankings by pool

17.21.7 Dispatched CPU rankings by priority

This graph groups the selected time period's wait bucket data by LIC priority and sorts by Dispatched CPU.

Tip: XPF priority can be calculated by subtracting 140 for those values shown exceeding 140.

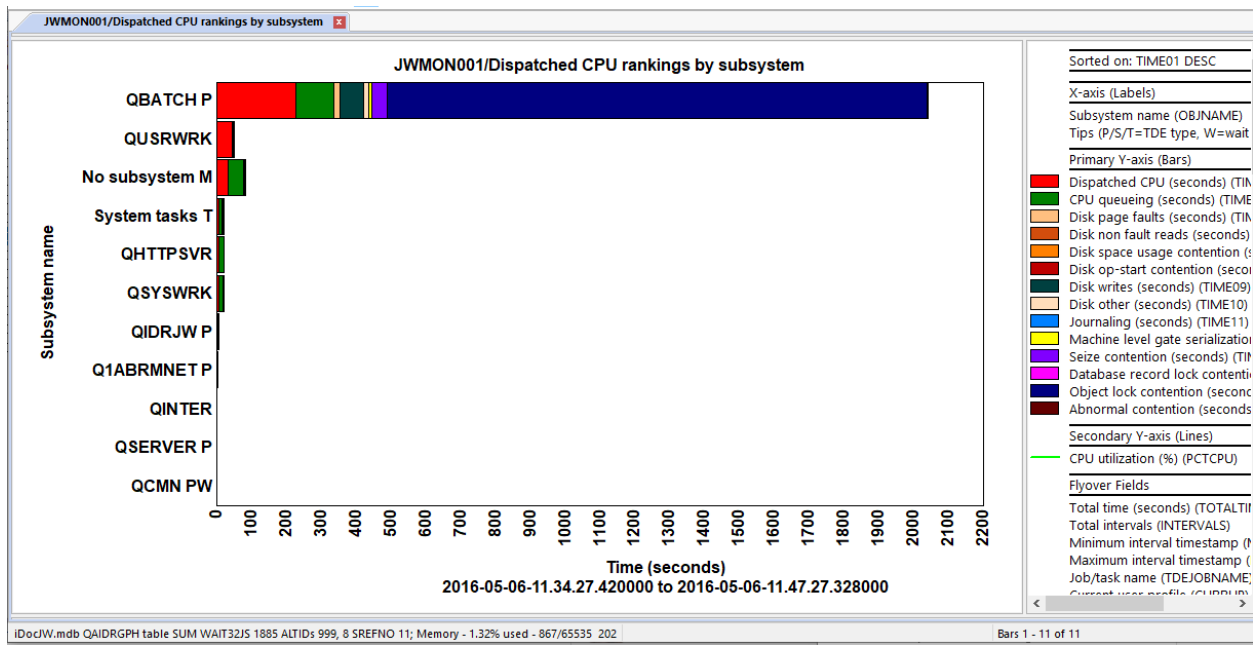


Dispatched CPU rankings by LIC priority

17.21.8 Dispatched CPU rankings by subsystem

This graph groups the selected time period's wait bucket data by subsystem and sorts by Dispatched CPU.

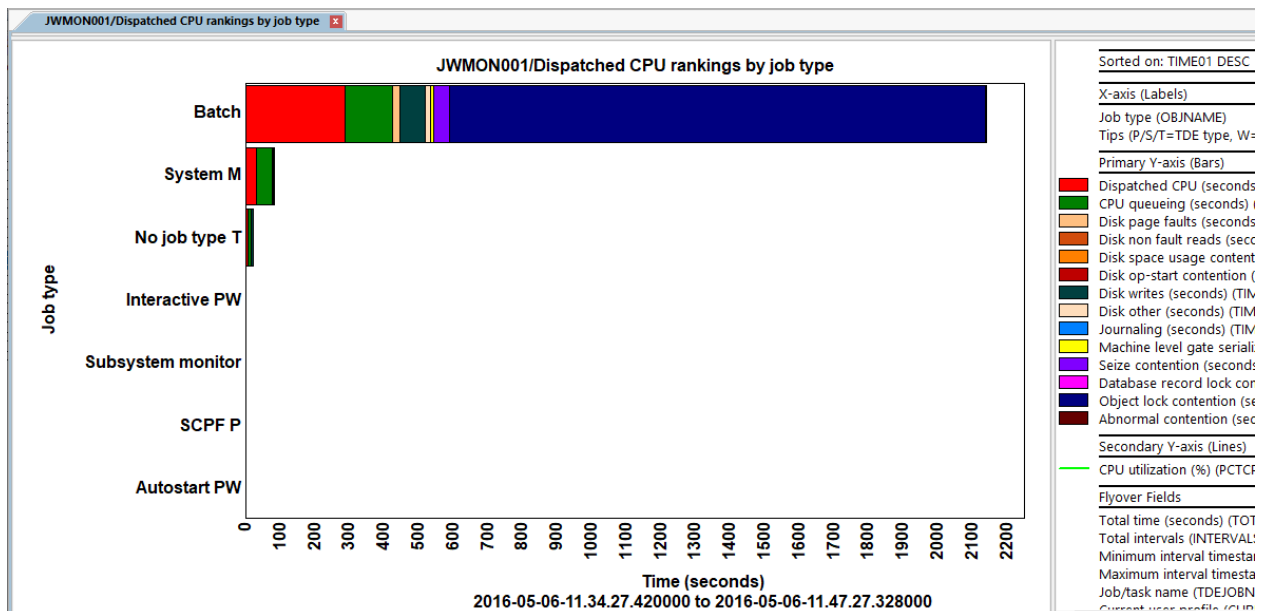
Note: Job times that had no subsystem listed are grouped into 1 bar called **"No subsystem"**. All system tasks are grouped together into one **"System tasks"** bar within this report.



Dispatched CPU rankings by subsystem

17.21.9 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".

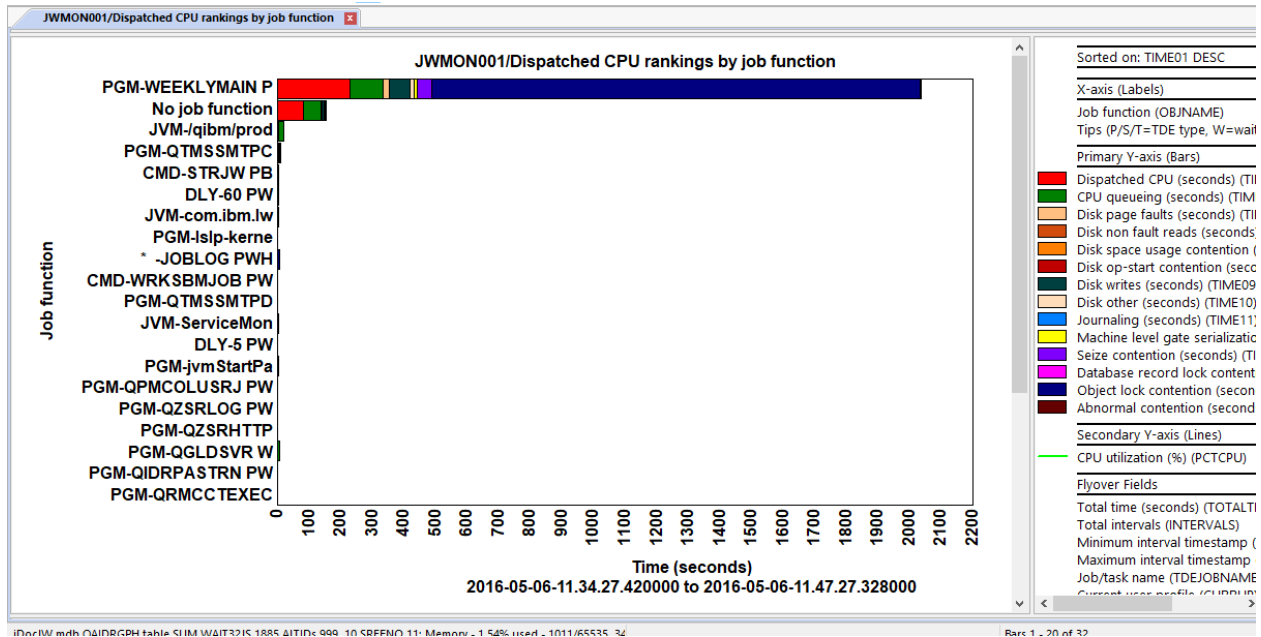


Dispatched CPU rankings by job type

17.21.10 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.



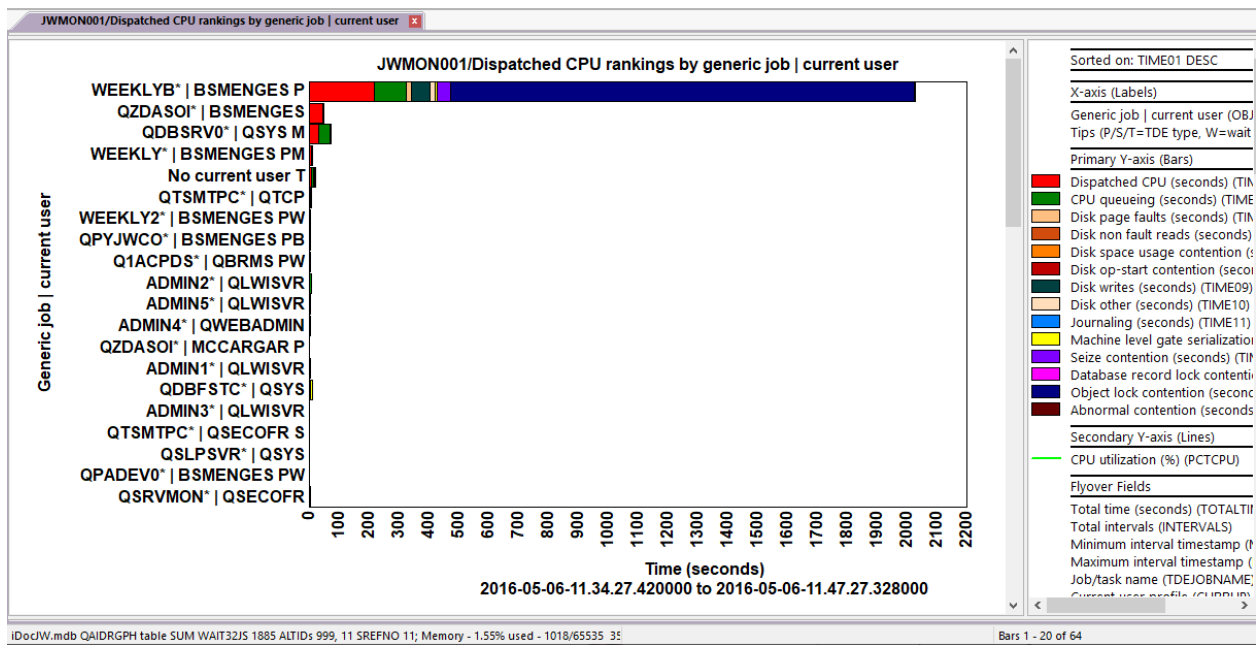
Dispatched CPU rankings by job function

17.21.11 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:

Name length for generic name grouping graphs: Start position:

Note: Job data without a current user profile are shown under the **"No current user"** value in the graph.

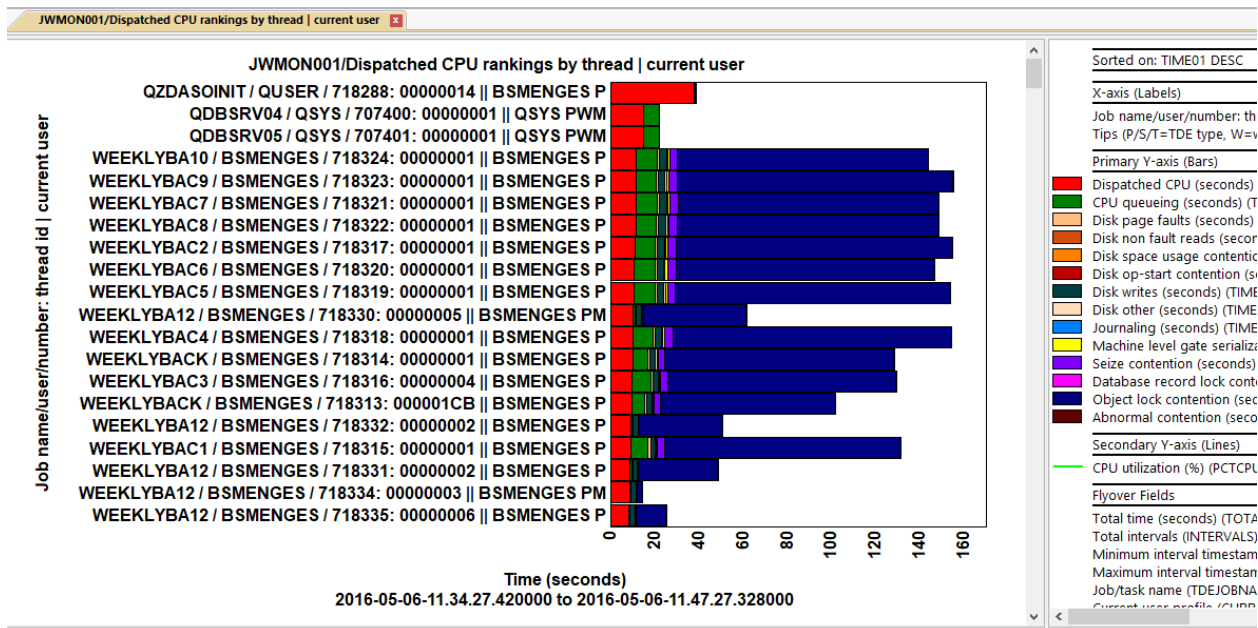


Dispatched CPU rankings by generic job | current user

17.21.12 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.

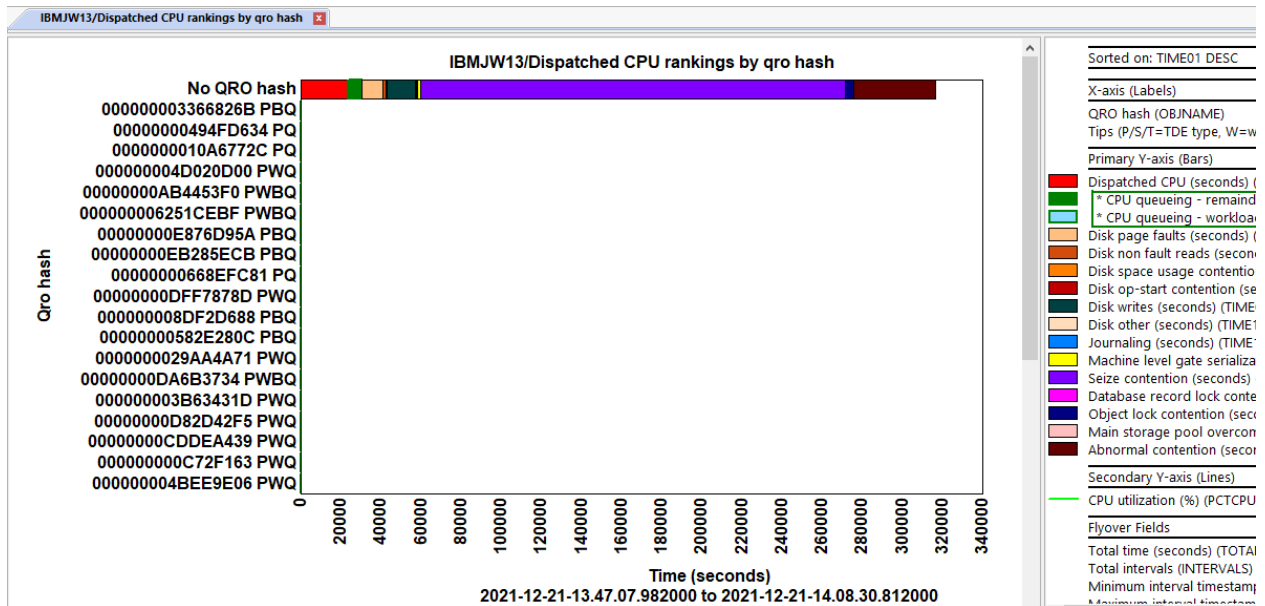
Note: Job data without a current user profile are shown under the "No current user" value in the graph.



Dispatched CPU rankings by thread | current user

17.21.13 Dispatched CPU rankings by QRO hash

This graph groups the selected time period's wait bucket data by QRO hash and sorts by Dispatched CPU. **Note:** The latest Job Watcher PTFs need to be loaded onto your system to produce data.



Dispatched CPU rankings by QRO hash

17.22 Clients + Workers wait time rankings

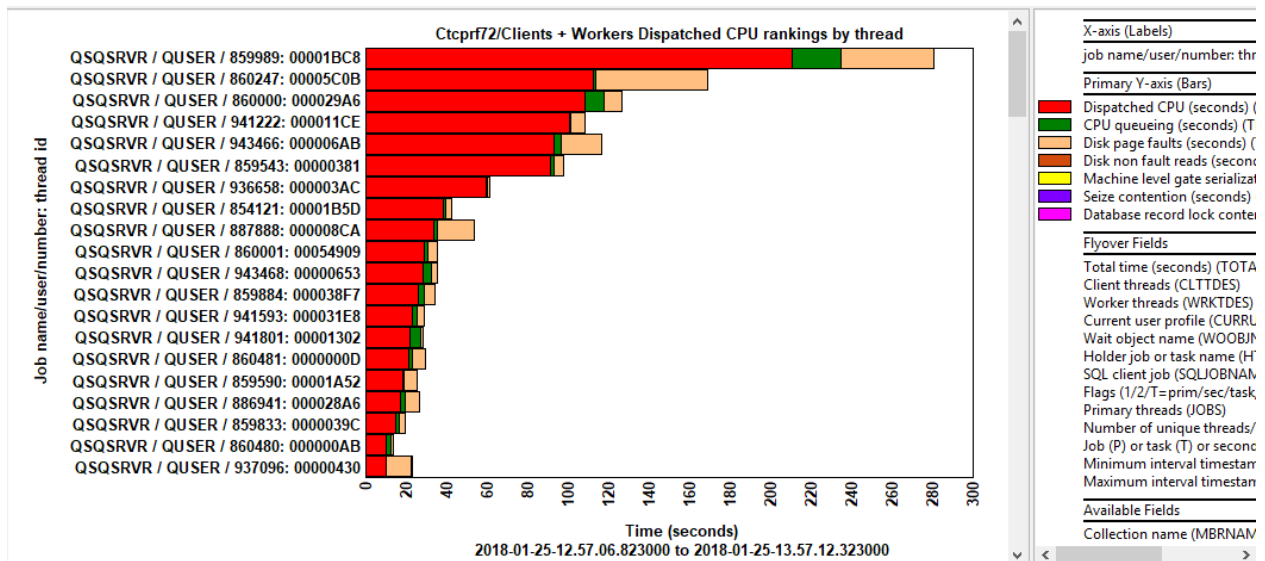
This folder provides the ranking graphs for QSQRVR jobs and their client jobs that initiated them.

Note: These graphs require the **Collection summary** analysis to be ran to appear!

- Clients + Workers wait time rankings by thread
- Clients + Workers wait time rankings by job
- Clients + Workers wait time rankings by job user
- Clients + Workers wait time rankings by generic job
- Clients + Workers wait time rankings by current user
- Clients + Workers wait time rankings by pool
- Clients + Workers wait time rankings by priority
- Clients + Workers wait time rankings by subsystem
- Clients + Workers wait time rankings by job type
- Clients + Workers wait time rankings by job function
- Clients + Workers wait time rankings by generic job | current user
- Clients + Workers wait time rankings by thread | current user
- Clients + Workers wait time rankings by qro hash

Clients + Workers wait time rankings

An example is provided below:



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](#).

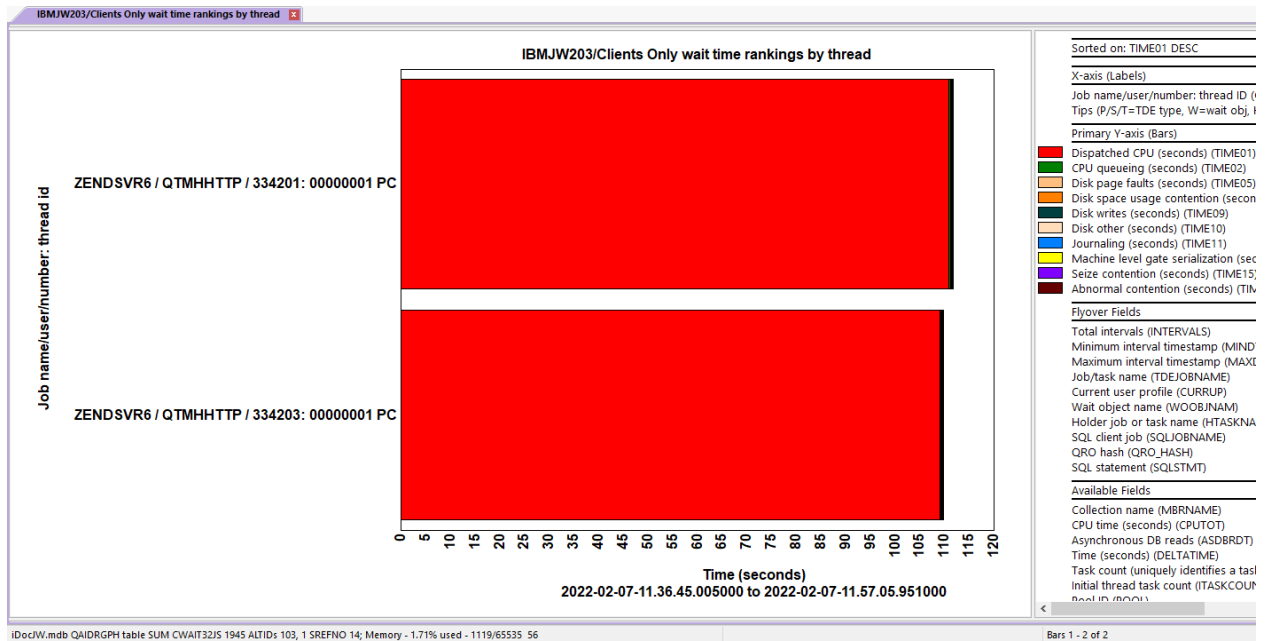
17.23 Clients only wait time rankings

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 to appear!

- Clients Only wait time rankings by thread
- Clients Only wait time rankings by job
- Clients Only wait time rankings by job user
- Clients Only wait time rankings by generic job
- Clients Only wait time rankings by current user
- Clients Only wait time rankings by pool
- Clients Only wait time rankings by priority
- Clients Only wait time rankings by subsystem
- Clients Only wait time rankings by job type
- Clients Only wait time rankings by job function
- Clients Only wait time rankings by generic job | current user
- Clients Only wait time rankings by thread | current user
- Clients Only wait time rankings by qro hash

Clients only wait time rankings



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](#).

17.24 CPU queueing breakdown rankings

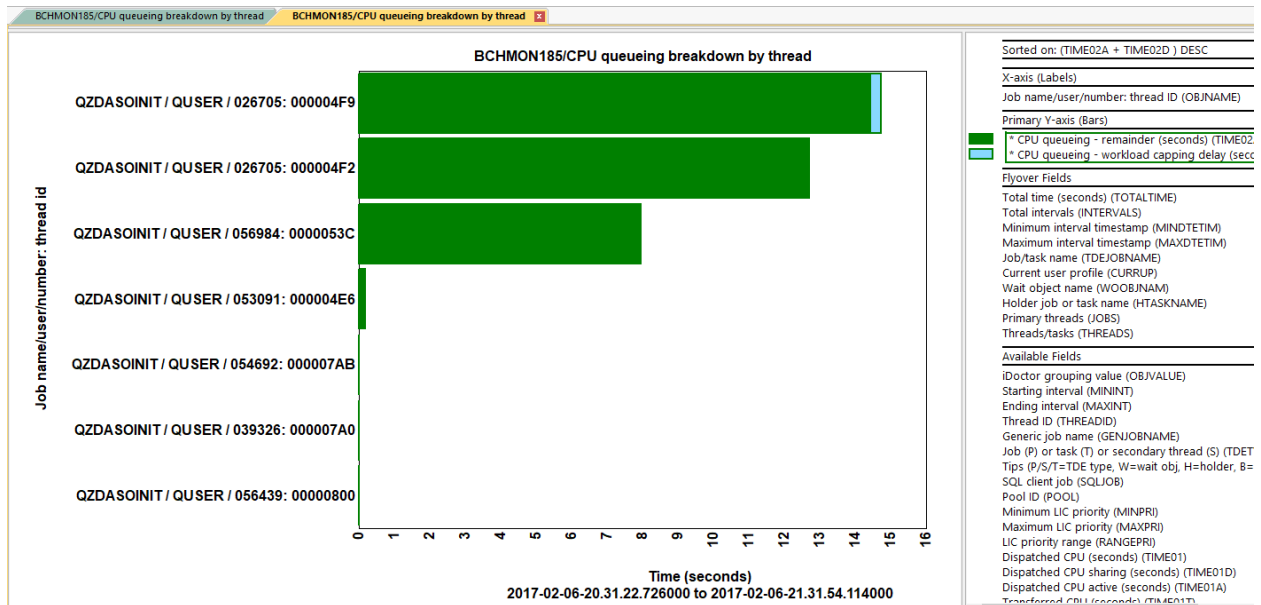
These graphs show the effects of workload capping delays on CPU queueing. The CPU queueing bucket is divided into 2 parts in this chart.

Note: This graph only appears if workload capping delays exist in the collection.

- CPU queueing breakdown by thread
- CPU queueing breakdown by job
- CPU queueing breakdown by job user
- CPU queueing breakdown by generic job
- CPU queueing breakdown by current user
- CPU queueing breakdown by pool
- CPU queueing breakdown by priority
- CPU queueing breakdown by subsystem
- CPU queueing breakdown by job type
- CPU queueing breakdown by job function
- CPU queueing breakdown by generic job | current user
- CPU queueing breakdown by thread | current user

CPU queueing breakdown rankings

An example follows:



CPU queueing breakdown by thread

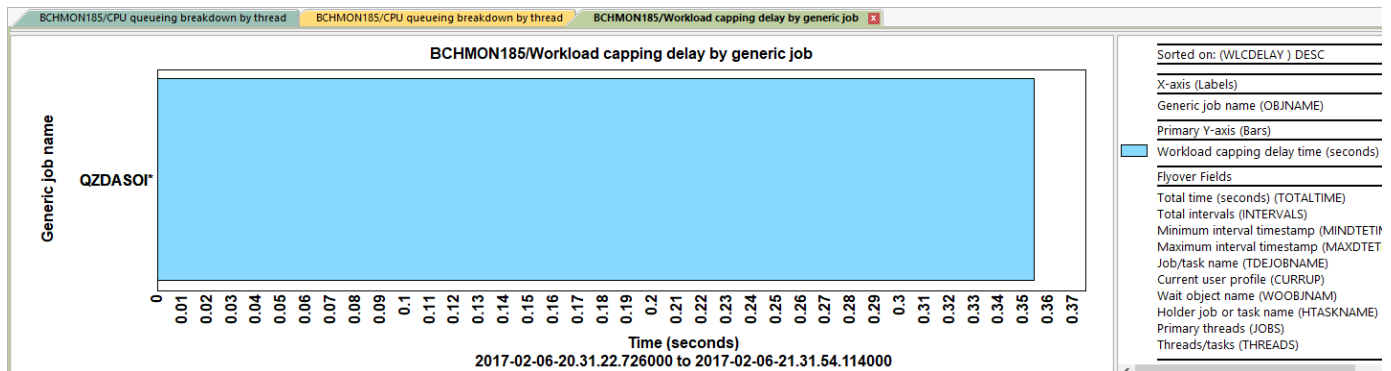
Note: The rest of the ranking graphs of this type will follow the same format as those described previously in the [Waits -> Dispatched CPU rankings folder](#).

17.25 Workload capping delay rankings

This graph groups the workload capping delay times in one of 12 ways shown in the previous section.

Note: This graph only appears if workload capping delays exist in the collection.

An example follows:

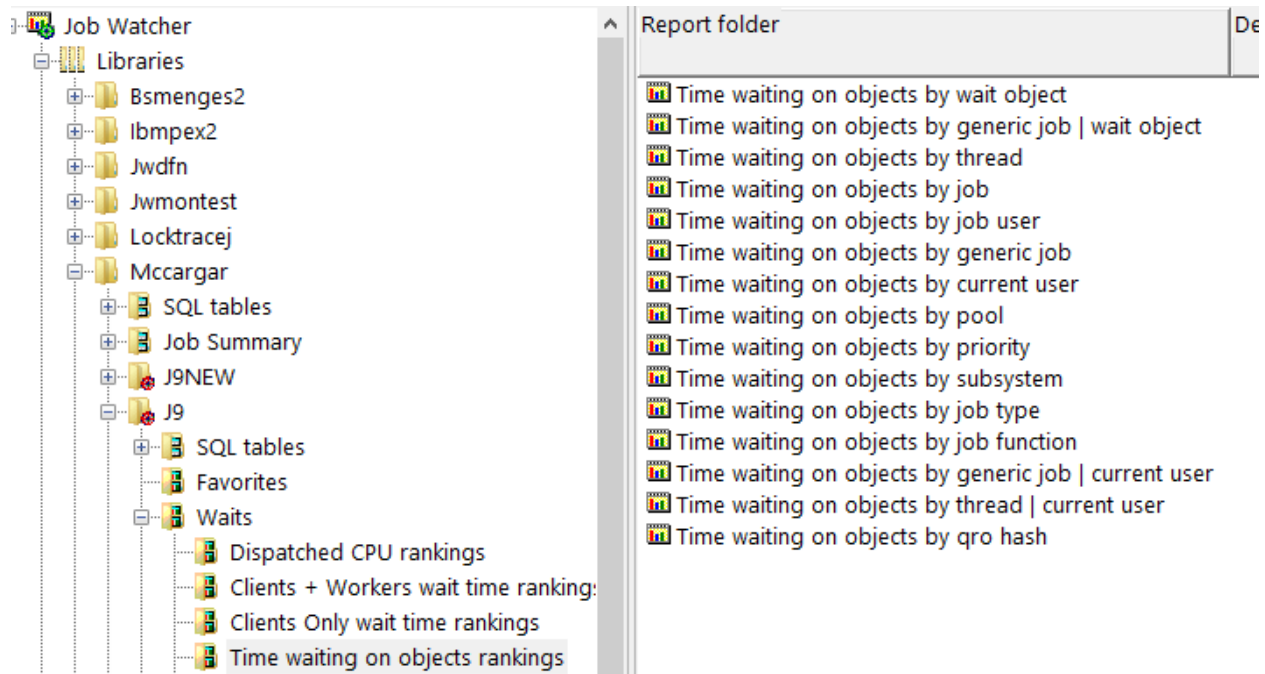


Workload capping delay by generic job

17.26 Time waiting on objects rankings

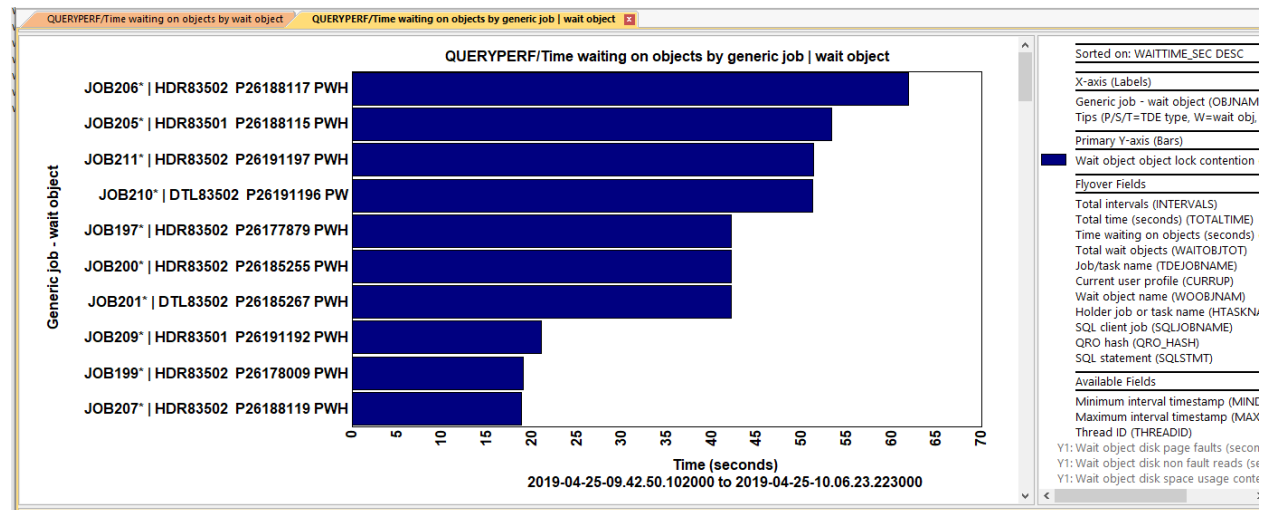
This folder contains the set of ranking graphs that rank current wait times for jobs experiencing interesting waits AND having a wait object associated with it. The jobs are ranked by total time in the interesting waits captured.

Note: These graphs require the **Collection summary analysis is ran!**



Waits -> Time waiting on objects rankings

An example follows:



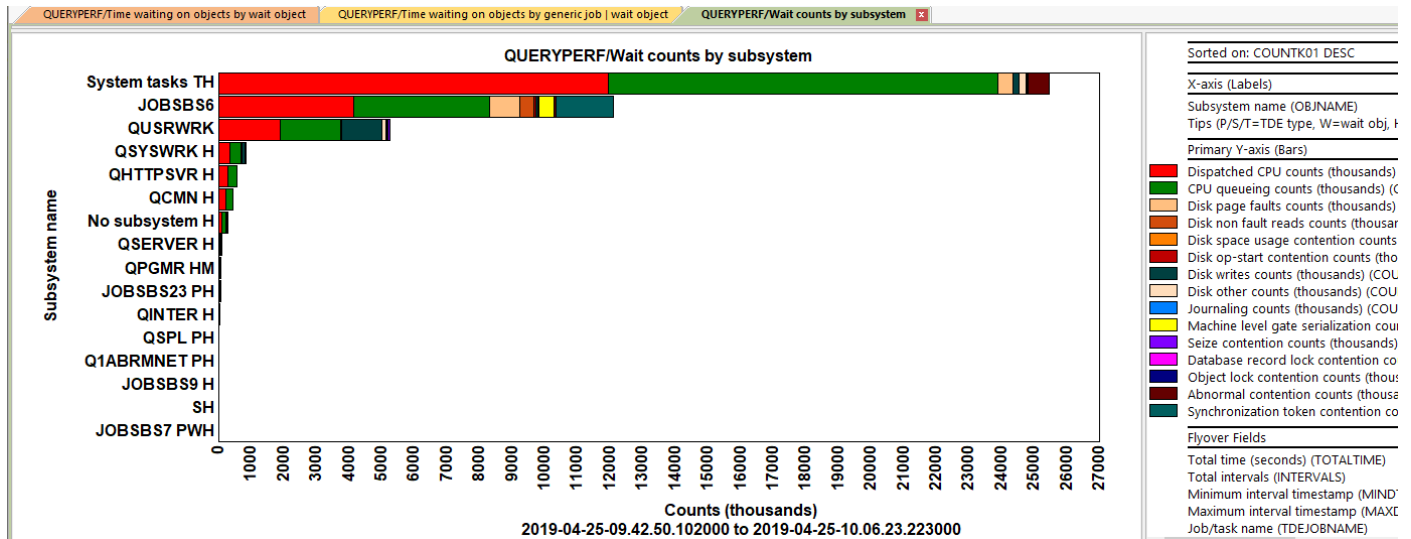
Time waiting on objects by generic job | wait object

17.27 Wait counts rankings

This graph rankings the “interesting” wait buckets’ number of distinct waits that occurred using the metric counts per second (In thousands) for each. The standard 14 job groupings are provided.

Note: This graph requires the **Collection summary** analysis is ran!

An example follows:



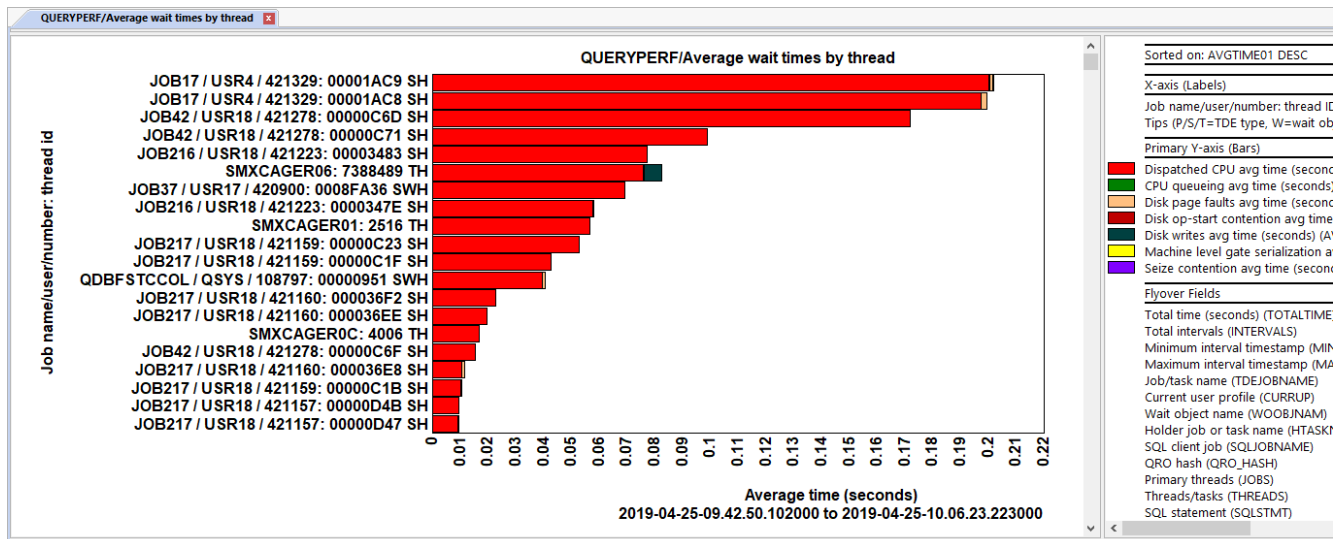
Wait counts by subsystem

17.28 Average wait times rankings

This graph displays the average wait time for each of the “interesting” wait buckets. This is the average duration of each wait in seconds.

Note: This graph requires the **Collection summary analysis is ran!**

An example follows:



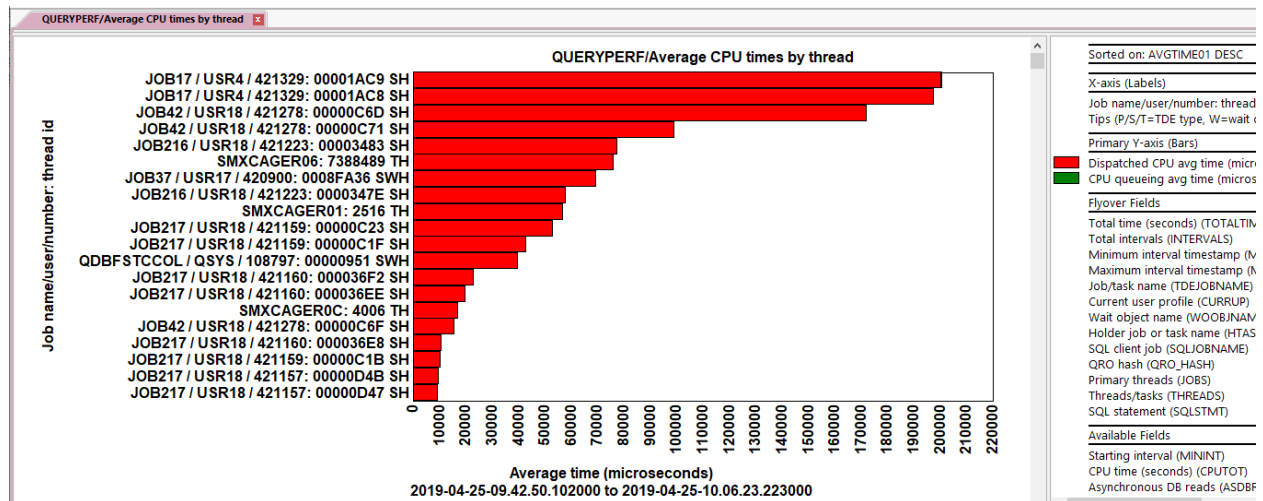
Average wait times by thread

17.29 Average CPU times rankings

This graph displays the average dispatched CPU and CPU queueing bucket times in microseconds.

Note: This graph requires the **Collection summary analysis is ran!**

An example follows:



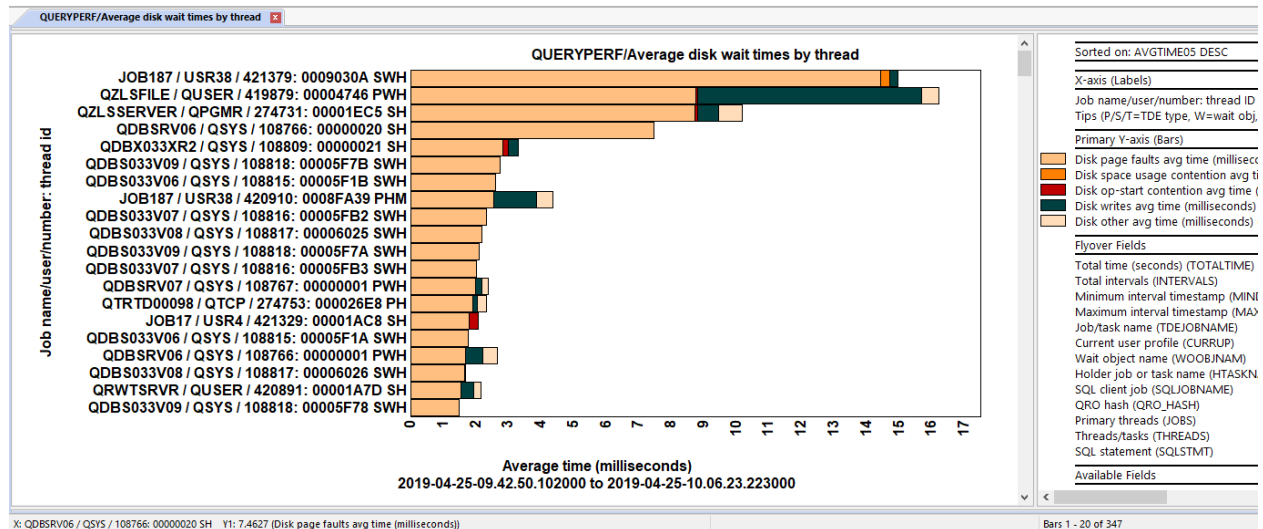
Average CPU times by thread

17.30 Average disk time rankings

This graph displays the average wait times for each disk related wait bucket in milliseconds (buckets 5-10.)

Note: This graph requires the **Collection summary analysis is ran!**

An example follows:

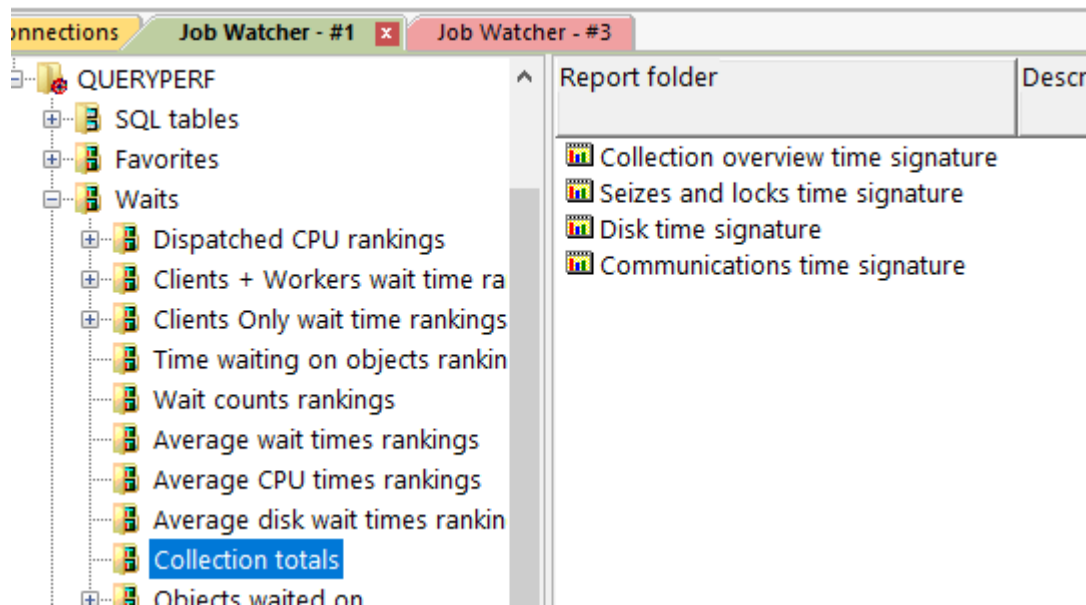


Average disk times by thread

17.31 Collection totals

Under the Collection totals subfolder are a set of pie charts showing some of the same graphs from the Waits folder except the times are added together across the entire collection instead of on a per interval basis.

Note: This folder requires the **Collection summary analysis is ran!**



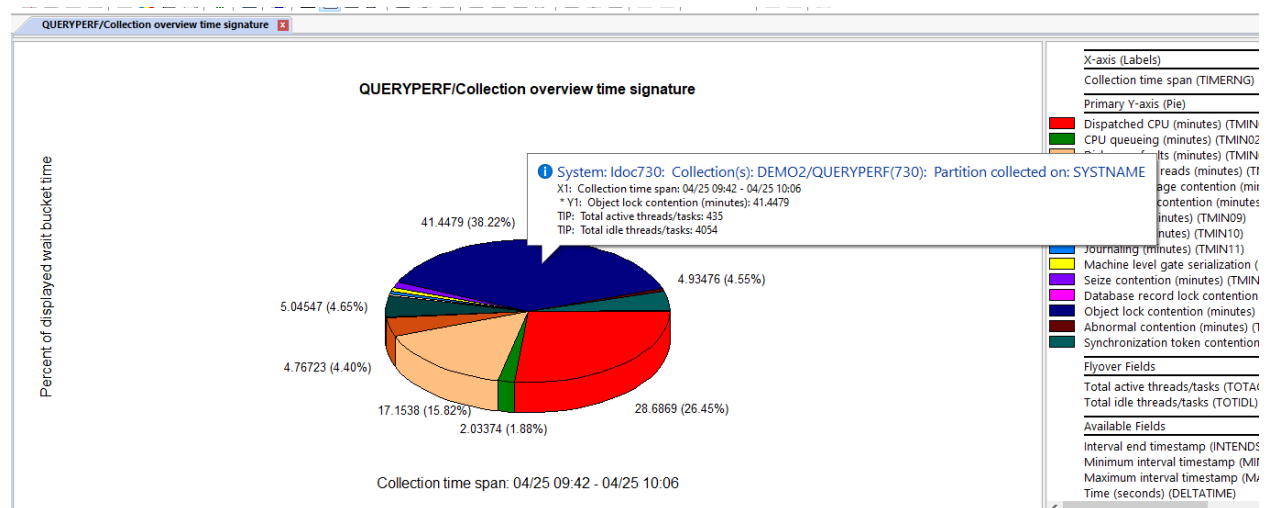
Waiters -> Collection totals folder

17.31.1 Collection overview time signature

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.

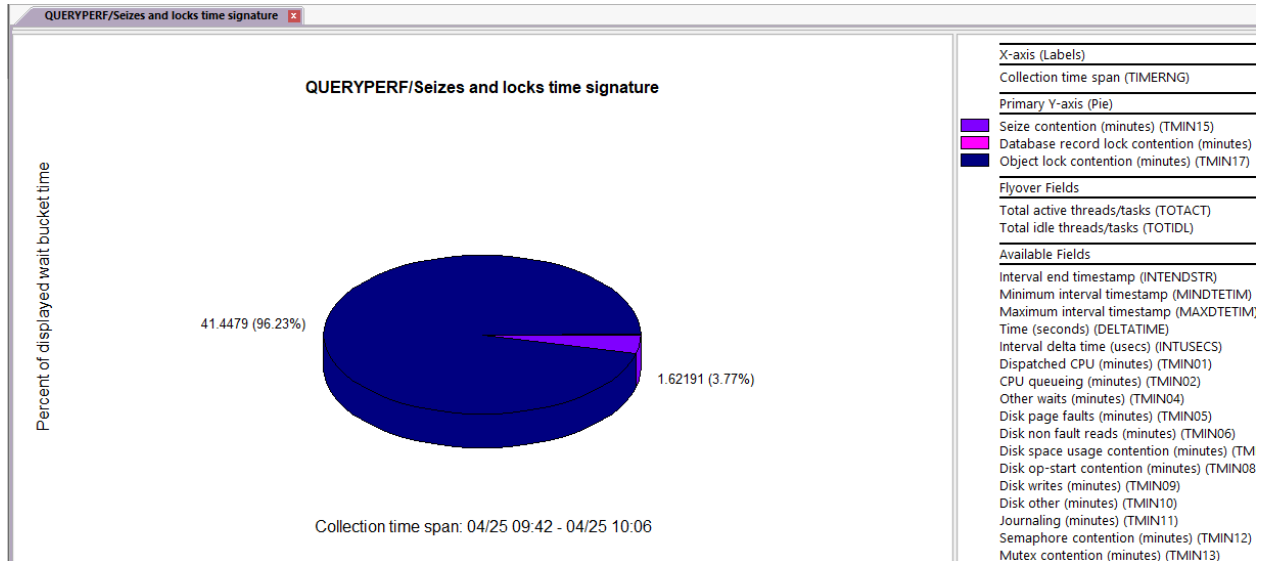
Tip: You can drill down from these graphs in the same way that you drill down from the Collection overview time signature graph, but keep in mind that all drill downs will be against the entire collection.



Collection overview time signature

17.31.2 Seizes and locks time signature

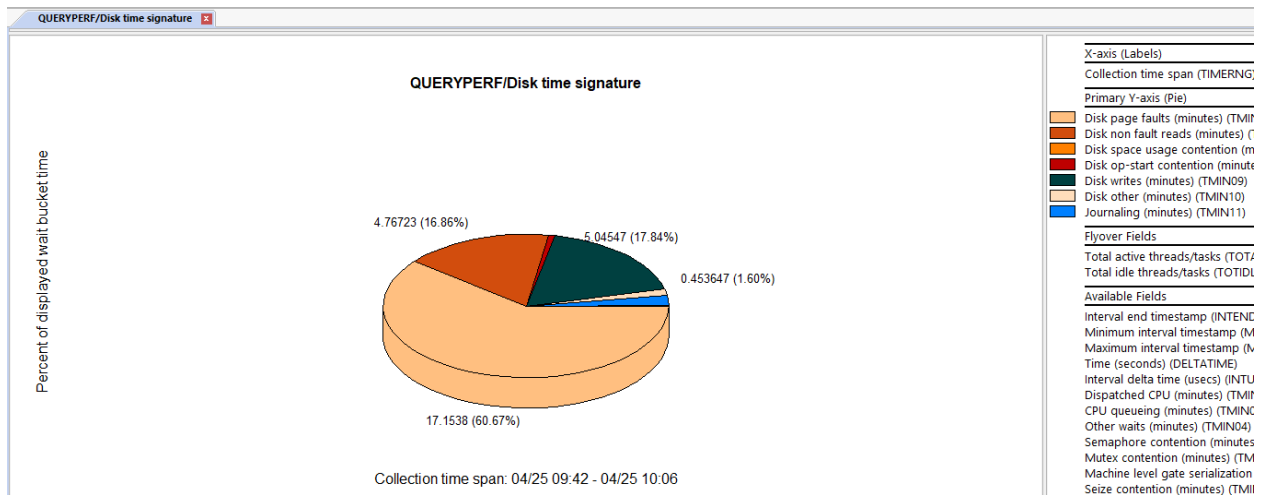
This graph displays only the wait bucket times related to seizures, DB record locks and object lock contention for the entire collection.



Seizes and locks time signature

17.31.3 Disk time signature

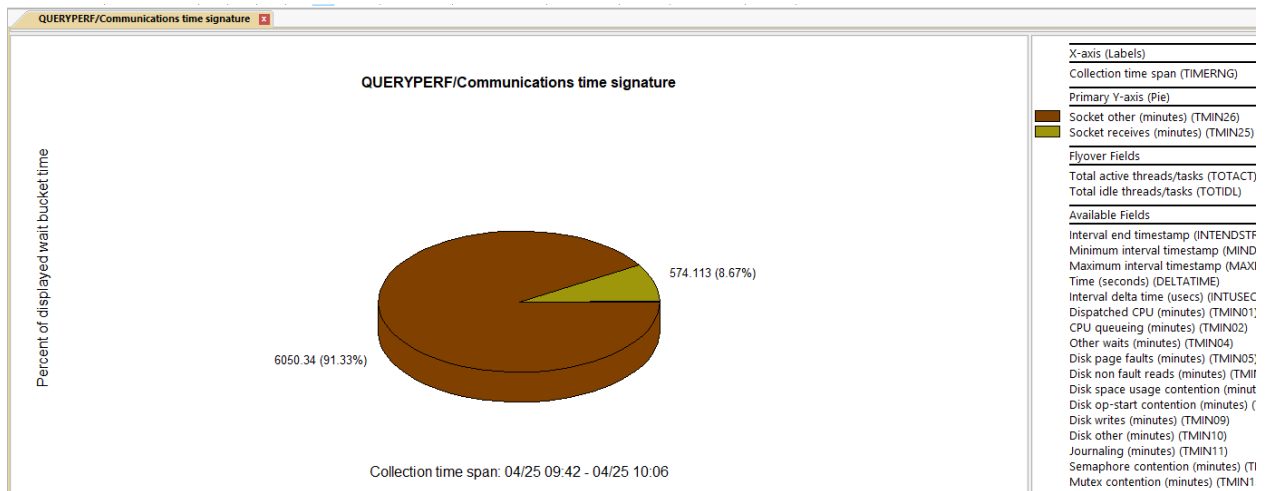
This graph shows only wait buckets associated disk related times.



Disk time signature

17.31.4 Communications time signature

This graph displays wait bucket times related to socket communications only.

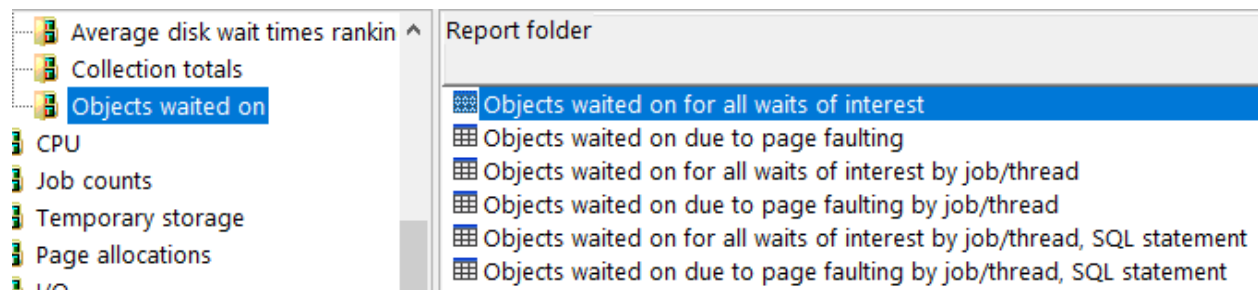


Communications time signature

17.32 Objects waited on

This folder contains a set of reports that summarize information about the wait objects found in the data.

Note: This folder requires the **Collection summary** analysis is ran!



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.

17.32.1 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.

QUERYPERF/Objects waited on for all waits of interest - #1										
Total intervals (INTERVALS)	Total wait time (ms) (WAITTIME/MS)	Total waits at snapshot (TOTWAITS_SNAP)	Total wait counts (could be for other wait objects) (TOTWAITCOUNTS)	Maximum wait time (ms) (MAXWAITMS)	Average wait time (ms) (AVGWAITMS)	Wait object library (WOBJLIB)	Wait object name (WOBJNAM)	Wait object type description (WOBJTYPD)	Wait bucket number (BUCKETNUM)	Wait bucket description (BUCKETDESC)
50	110,3710	51	100,886	44,9460	2,1641		ARPOH ARPOH	PHYSICAL FILE MBR - DATA PART	6	Disk non-fault reads
31	24,8910	31	57,953	11,6830	.8029		CUPOHSVEIDCUPOHSVEID	DB2 ACCESS PATH	5	Disk page faults
29	50,1950	29	62,696	23,3370	1,7309		CUIOHSVID5CUIOHSVID5	DB2 ACCESS PATH	5	Disk page faults
23	21,7160	23	48,965	3,8250	.9442		CUIOHSVID4CUIOHSVID4	DB2 ACCESS PATH	5	Disk page faults
21	5,9440	21	69,186	1,1470	.2830		CULIF6 CULIF6	DB2 ACCESS PATH	5	Disk page faults
16	7,3330	16	50,195	3,5000	.4583		CULOH1 CULOH1	DB2 ACCESS PATH	5	Disk page faults
13	61,805,7320	1	13	61,805,7370	31,634,6112		HDR83502 P26188117	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
11	53,290,3680	1	11	53,290,3730	28,145,3394		HDR83501 P26188115	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
11	51,268,8980	1	11	51,268,9030	26,085,9197		HDR83502 P26191197	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
11	51,169,1250	2	11	29,991,3490	14,539,2766		DTL83502 P26191196	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
9	42,128,0030	1	9	42,128,0060	22,009,8387		HDR83502 P26177879	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
9	42,124,2750	1	18	42,124,2790	22,006,1544		HDR83502 P26185255	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
9	42,116,7540	1	18	42,116,7590	21,998,6646		DTL83502 P26185267	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
7	3,7040	7	12,361	2,3020	.5291		CUPOHT CUPOHT	PHYSICAL FILE MBR - DATA PART	6	Disk non-fault reads
7	2,8700	7	21,779	1,8990	.4100		CULO3 CULO3	DB2 ACCESS PATH	5	Disk page faults
6	35,1810	13	3,961	6,0340	2,7062		N000000345HSFEPRSF	PHYSICAL FILE MBR - DATA PART	15	Seize contention
5	21,047,7110	1	5	21,047,7130	10,992,4344		HDR83501 P26191192	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
5	8,5660	5	23,029	6,8370	1,7132		CULOHT10 CULOHT10	DB2 ACCESS PATH	5	Disk page faults
5	1,0280	5	16,141	.3520	.2056		CULIF9 CULIF9	DB2 ACCESS PATH	5	Disk page faults
4	18,971,0560	1	4	18,971,0580	11,398,5465		HDR83502 P26178009	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
4	18,849,9490	1	8	18,849,9510	11,278,4353		HDR83502 P26188119	PHYSICAL FILE MBR - DATA PART	17	Object lock contention
4	9,4650	4	6,961	8,7420	2,3663		ARLOH2 ARLOH2	DB2 ACCESS PATH	5	Disk page faults
4	2,5610	4	5,413	1,1550	.6402		N000000345HSFEPRSF	DB2 ACCESS PATH	9	Disk writes
4	1,2910	4	18,019	.4900	.3227		CULOHT11 CULOHT11	DB2 ACCESS PATH	5	Disk page faults

Objects waited on for all waits of interest

17.32.2 Objects waited on due to page faulting

This report displays the time spent waiting on page faulting with a wait object present in the data.

ADVANCED - iDoctor Data Viewer - #1 - [QUERYPERF/Objects waited on due to page faulting - #1]										
File Edit View Window Help										
										
QUERYPERF/Objects waited on due to page faulting - #1										
Total intervals (INTERVALS)	Total wait time (ms) (WAITTIME/MS)	Total page faults at snapshot (TOTFLTS_SNAP)	Total page faults (could be for other wait objects) (TOTFLTS_OTHERS)	Maximum wait time (ms) (MAXWAITMS)	Average wait time (ms) (AVGWAITMS)	Wait object library (WOBJLIB)	Wait object name (WOBJNAM)	Wait object type description (WOBJTYPD)	Total jobs/tasks/threads (JOBCNT)	
31	24.8910	31	57,953	11.6830	.8029		CUPOHSVEIDCUPOHSVEID	DB2 ACCESS PATH	8	
29	50.1950	29	62,696	23.3370	1.7309		CUIOHSVID5CUIOHSVID5	DB2 ACCESS PATH	8	
23	21.7160	23	48,965	3.8250	.9442		CUIOHSVID4CUIOHSVID4	DB2 ACCESS PATH	8	
21	5.9440	21	69,186	1.1470	.2830		CULIF6 CULIF6	DB2 ACCESS PATH	1	
16	7.3330	16	50,195	3.5000	.4583		CULOH1 CULOH1	DB2 ACCESS PATH	3	
7	2.8700	7	21,779	1.8990	.4100		CULO3 CULO3	DB2 ACCESS PATH	1	
5	8.5660	5	23,029	6.8370	1.7132		CULOHT10 CULOHT10	DB2 ACCESS PATH	2	
5	1.0280	5	16,141	.3520	.2056		CULIF9 CULIF9	DB2 ACCESS PATH	1	
4	9.4650	4	6,961	8.7420	2.3663		ARLOH2 ARLOH2	DB2 ACCESS PATH	2	
4	1.2910	4	18,019	.4900	.3227		CULOHT11 CULOHT11	DB2 ACCESS PATH	1	
3	1.9930	3	13,756	1.3300	.6643		CULOHT9 CULOHT9	DB2 ACCESS PATH	1	
3	1.9140	3	7,316	1.3200	.6380		ARLOH3 ARLOH3	DB2 ACCESS PATH	2	
3	.8150	3	12,187	.6310	.2717		CULIF4 CULIF4	DB2 ACCESS PATH	1	
2	52.0090	2	1,698	51.2970	26.0045		CULOH17 CULOH17	DB2 ACCESS PATH	1	
2	51.5770	2	8,843	50.8080	25.7885		CULOH31 CULOH31	DB2 ACCESS PATH	2	
2	8.9600	2	7,264	8.3580	4.4800		CULOH64 CULOH64	DB2 ACCESS PATH	2	
2	4.3390	2	9,243	4.1610	2.1695		CULOHT16 CULOHT16	DB2 ACCESS PATH	1	
2	1.5320	2	10,727	1.0830	.7660		CULOHT4 CULOHT4	DB2 ACCESS PATH	1	
2	1.2980	2	8,683	.8960	.6490		CULOHT11B CULOHT11B	DB2 ACCESS PATH	1	
2	1.1510	2	7,722	.9100	.5755		CULOHT15 CULOHT15	DB2 ACCESS PATH	1	
2	1.1490	2	4,282	.6810	.5745		CUPPVSVEIDCUPPVSVEID	DB2 ACCESS PATH	2	
2	1.0290	2	8,099	.5870	.5145		CULOHT8 CULOHT8	DB2 ACCESS PATH	1	
2	.7600	2	5,258	.4230	.3800		CUIPYSVID4CUIPYSVID4	DB2 ACCESS PATH	2	
2	.7090	2	4,322	.3820	.3545		CUIF CUIF	PHYSICAL FILE MBR - DATA PART	1	
2	.5090	2	4,125	.3400	.2545		CULPY14 CULPY14	DB2 ACCESS PATH	1	

Objects waited on due to page faulting

17.32.3 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											
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 (WOBJNAM)	Wait object type description (WOBJTYPD)	Bucket number (BUCKETNU)	Description (BUCKETDESC)	Job name/user/number: thread ID (JTTHREAD)	Task count identifies a task/thread (TASKCOU)
42	9,595.8780	41	156	1,127.4 seconds	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	
22	4,153.3600	21	154	1,171.4070	186.1003	BSMENGES	USER PROFILE	15	Seize contention	DRIVER3 / BSMENGES / 011788: 00000003	

Objects waited on for all waits of interest by job/thread

17.32.4 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 (TOTFLT_SNAP)	Total page faults (could be for other wait objects) (TOTFLT_OTHERS)	Maximum wait time (ms) (MAXWAIT)	Average wait time (ms) (AVGWAITM)	Wait object name (WOBJNAM)	Wait object type description (WOBJTYPD)	Job name/user/number: thread ID (JTTHREAD)	Task count identifies a task/thread (TASKCOUNT)	
9	856.2910	9	1,737	722.8320	95.1434	QAPYJWTE RUN1	PHYSICAL FILE MBR - DATA PART	QZDASOINIT / QUSER / 011234: 00000025		
5	1,288.3560	5	192	911.1820	257.6712	QAPYJWSTK RUN1	PHYSICAL FILE MBR - DATA PART	QDBFSTCCOL / QSYS / 009542: 00000005		
3	307.2180	3	36	157.7510	102.4060	QXQTEMP1	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

17.32.5 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.

Note: This report will only appear if QAPYJWSQL contains data.

QUERYPERF/Objects waited on for all waits of interest by job/thread, SQL statement - #1											
Total intervals (INTERVALS)	Total wait time (ms) (WAITTIMEMS)	Total waits at snapshot (TOTWAITS_SNAP)	Total wait counts (could be for other wait objects) (TOTWAITCOUNTS)	Maximum wait time (ms) (MAXWAITMS)	Average wait time (ms) (AVGWAITMS)	Wait object library (WOBJLIB)	Wait object name (WOBJNAM)	Wait object type description (WOBJTYPD)	Wait bucket number (BUCKETNUM)	Wait bucket description (BUCKETDESC)	Job name/user/number: thread ID (JTTHREAD)
1	1,211.9010	1	1	1,211.9010	1,211.9010	ARPOH	ARPOH	PHYSICAL FILE MBR - DATA PART	15	Seize contention	JOB47 / USR10 / 4;
1	1,132.0870	1	1	1,132.0870	1,132.0870	ARPOH	ARPOH	PHYSICAL FILE MBR - DATA PART	15	Seize contention	JOB47 / USR10 / 4;
1	1,070.6670	1	1	1,070.6670	1,070.6670	ARPOH	ARPOH	PHYSICAL FILE MBR - DATA PART	15	Seize contention	JOB47 / USR10 / 4;
1	33.5570	1	31	33.5570	33.5570	REMIT00001	REMIT00001	PHYSICAL FILE MBR - DATA PART	9	Disk writes	QRWTSRVR / QUSI
1	33.5570	1	31	33.5570	33.5570	REMIT00001	REMIT00001	PHYSICAL FILE MBR - DATA PART	9	Disk writes	QRWTSRVR / QUSI
1	1.7090	1	16	1.7090	1.7090	REMITINFO	REMITINFO	PHYSICAL FILE MBR - DATA PART	9	Disk writes	QRWTSRVR / QUSI
1	1.1550	1	400	1.1550	1.1550	N00000034	HSFEPFRSF	DB2 ACCESS PATH	9	Disk writes	QZDASOINIT / QUSI
1	.7550	1	10	.7550	.7550	REMIT00001	REMIT00001	PHYSICAL FILE MBR - DATA PART	9	Disk writes	QRWTSRVR / QUSI
1	.1260	1	2,324	.1260	.1260	ARPOH	ARPOH	PHYSICAL FILE MBR - DATA PART	6	Disk non-fault reads	JOB37 / USR17 / 4;
1	.1260	1	2,324	.1260	.1260	ARPOH	ARPOH	PHYSICAL FILE MBR - DATA PART	6	Disk non-fault reads	JOB37 / USR17 / 4;

Objects waited on for all waits of interest by job/thread, SQL statement

17.32.6 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.

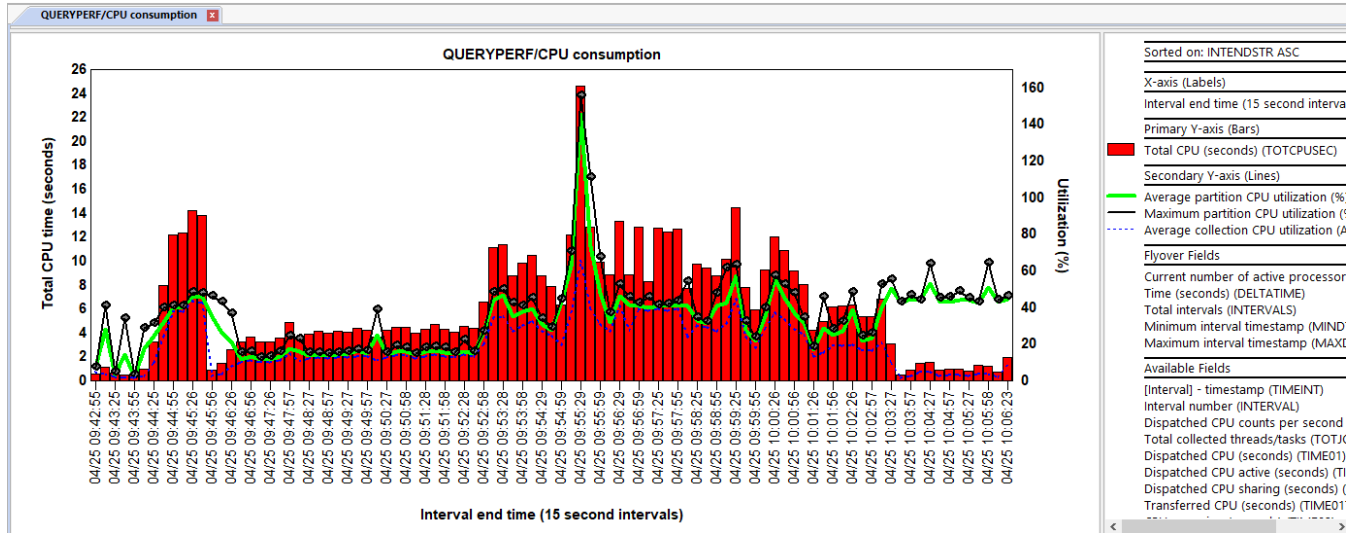
Note: This report will only appear if QAPYJWSQL contains data.

18 CPU

These graphs show CPU utilization and CPU times for the collection in various ways.

18.1 CPU consumption

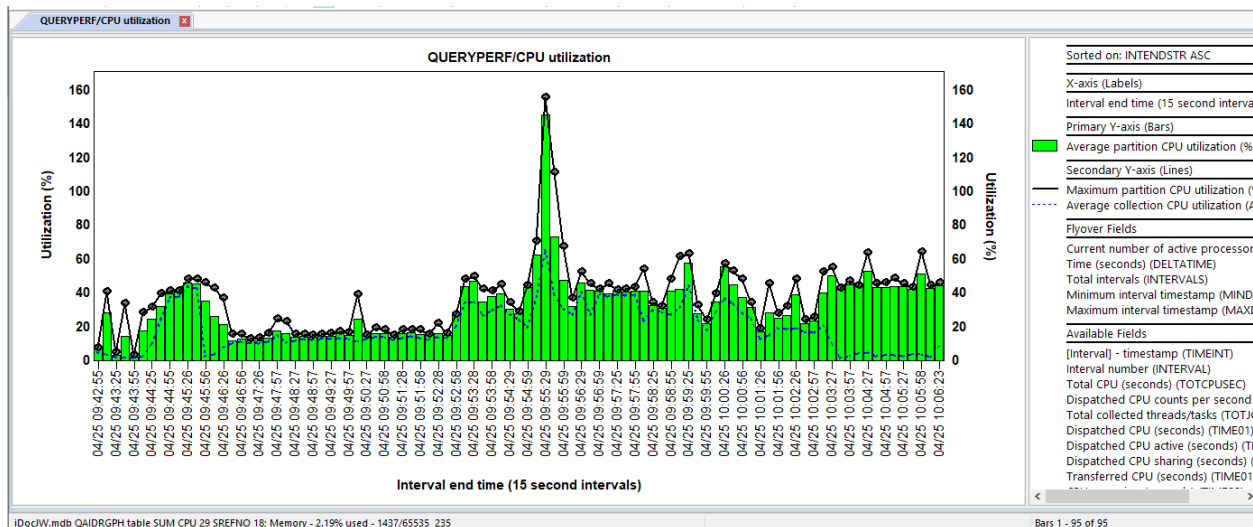
This graph displays the total CPU time consumed over time with the CPU utilization percentages of the Y2-axis.



CPU consumption

18.2 CPU utilization

This graph displays the CPU utilization average on the Y1-axis and the Y2-axis displays the maximum and minimum utilizations. **Note:** The later values are the minimum and maximum averages when graphing multiple intervals per bar.



CPU utilization

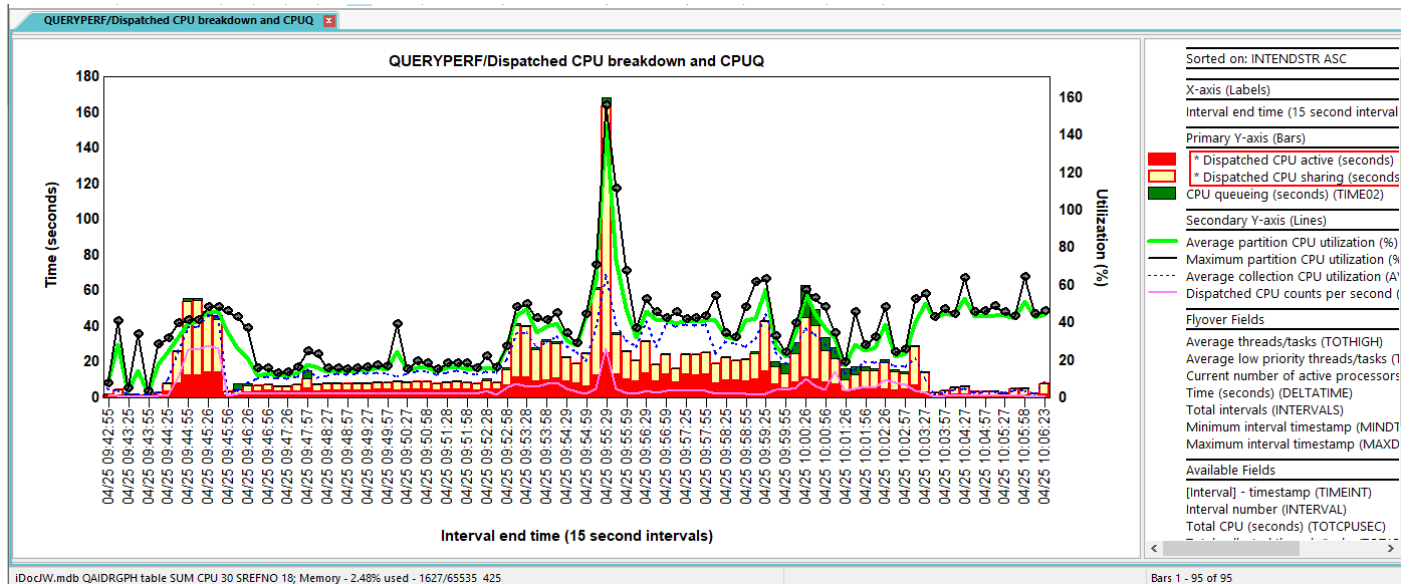
18.3 Dispatched CPU breakdown and CPUQ

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 other 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.

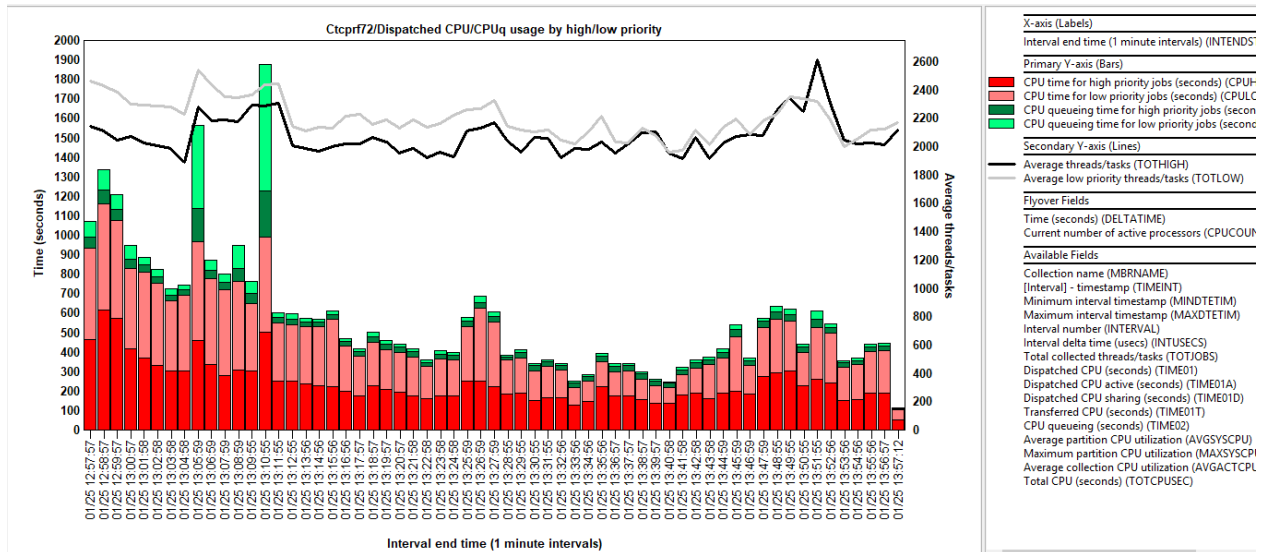


Dispatched CPU breakdown and CPUQ

18.4 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.

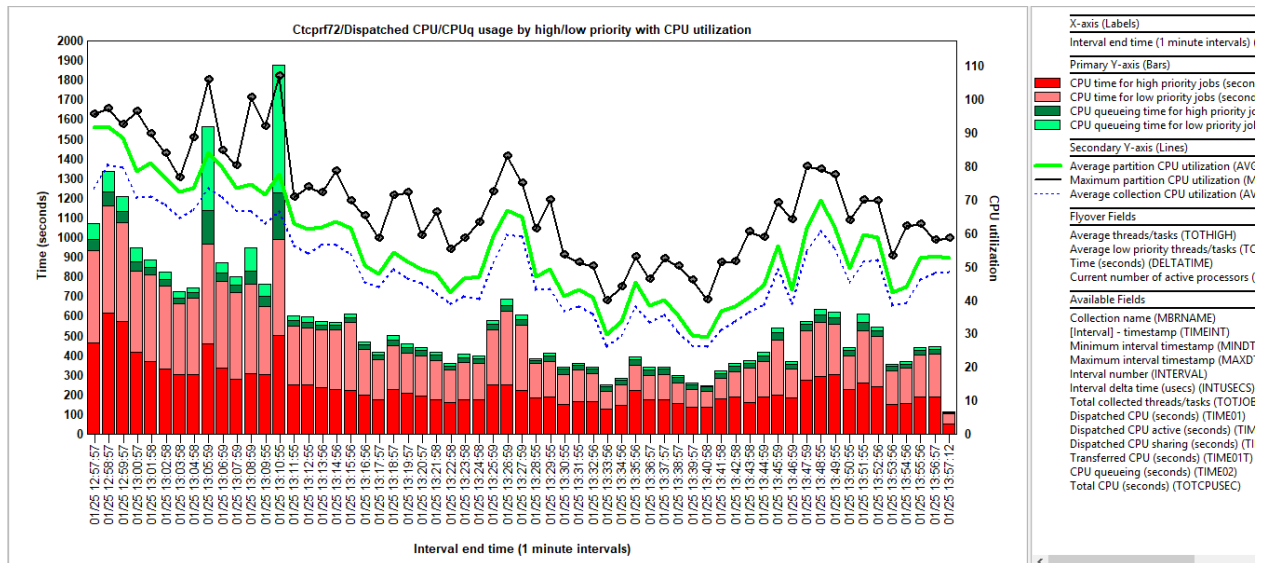
The graph also shows the average number of threads/tasks and the average number of low priority threads/tasks each interval.



Dispatched CPU/CPUq usage by high/low priority

18.5 Dispatched CPU/CPUq usage by high/low priority with CPU utilization

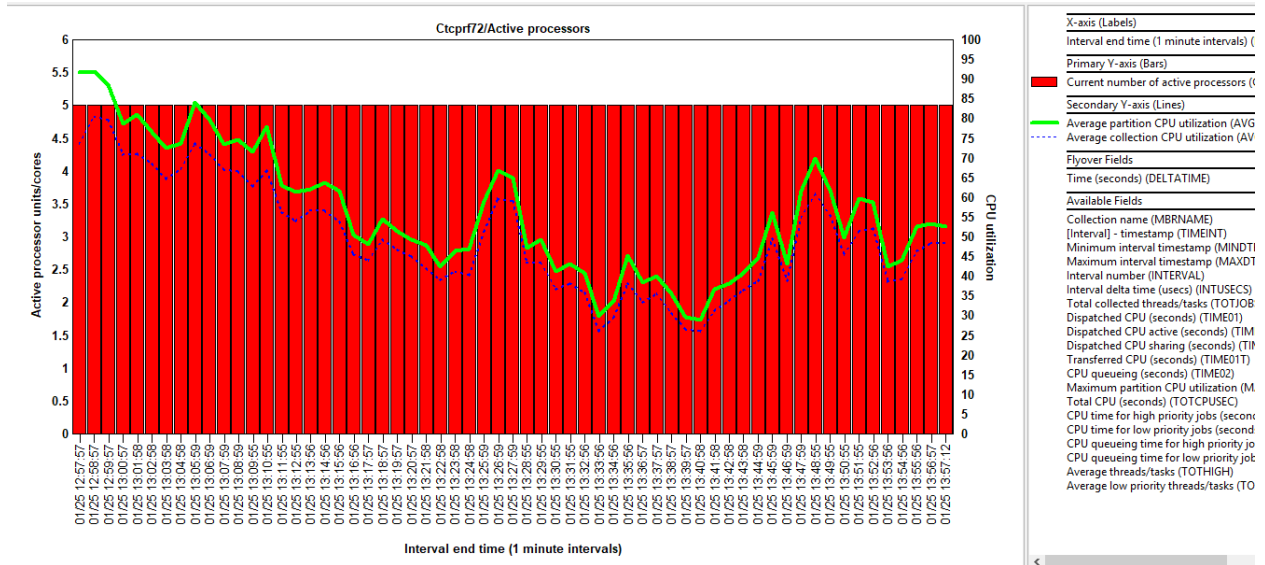
This graph shows the same graph as the previous one but with CPU utilization on the secondary Y-axis.



Dispatched CPU/CPUq usage by high/low priority with CPU utilization

18.6 Active processors

This graph displays the active virtual processors on the system over time along with CPU utilization on the Y2-axis.



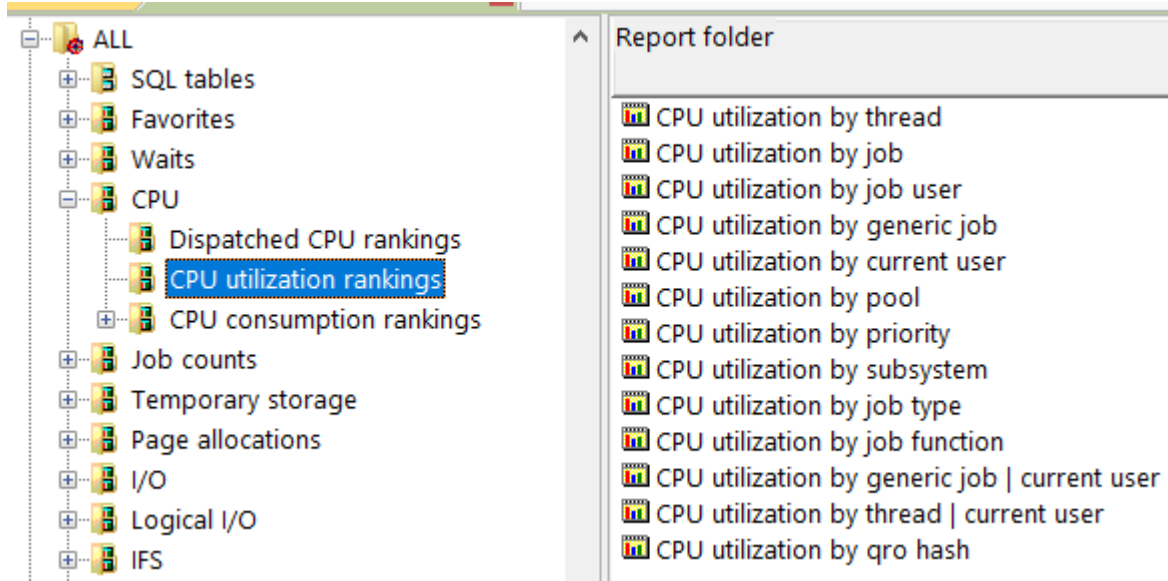
Active processors

18.7 Dispatched CPU rankings

These graphs are identical to the graphs in the [Waits -> Dispatched CPU rankings](#) folder.

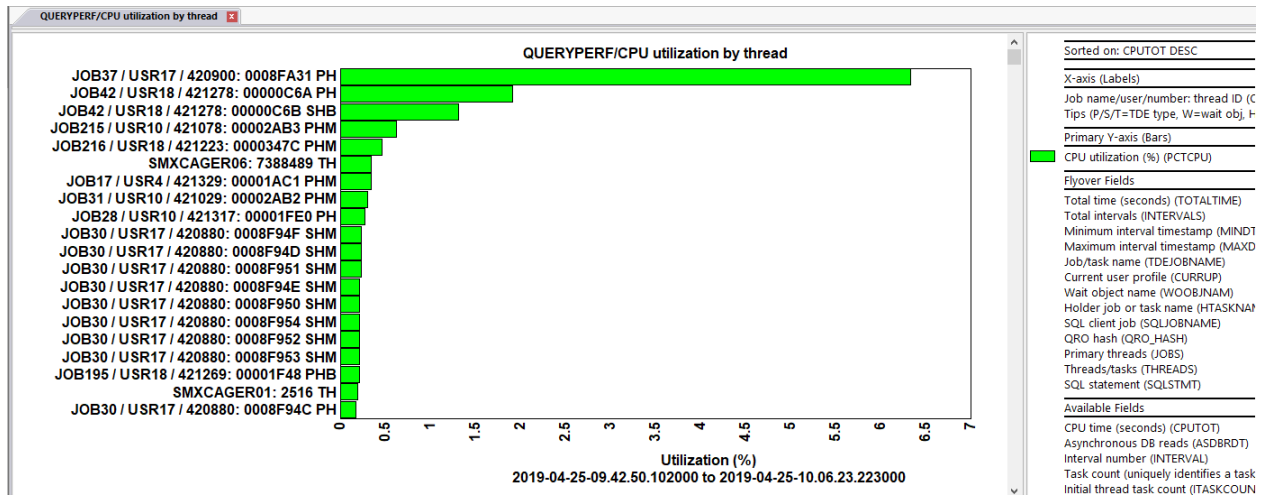
18.8 CPU utilization rankings

These graphs display CPU utilization ranked by one of many different types of job groupings.



CPU utilization rankings

An example follows:

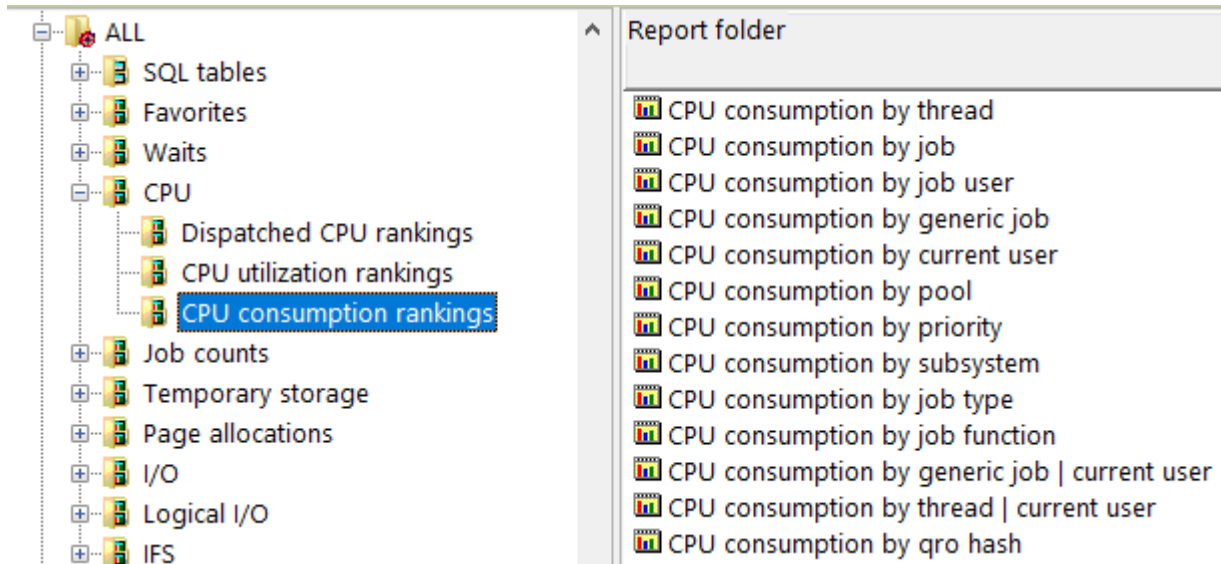


CPU utilization by thread

18.9 CPU consumption rankings

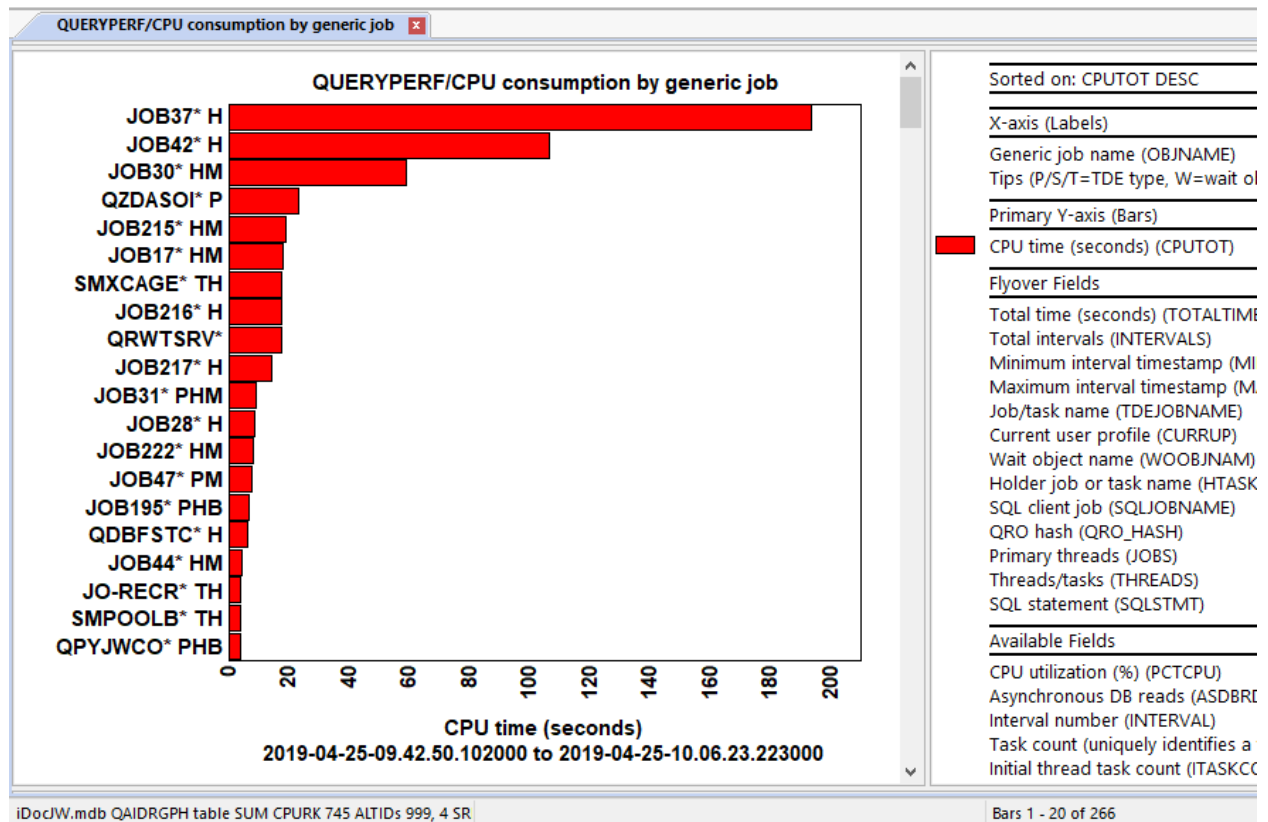
These graphs display CPU time consumed ranked by one of many different types of job groupings.

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 potentially sharing it with other threads.



CPU consumption rankings

An example follows:



CPU consumption by generic job

19 Job counts

The graphs in this folder show job counts and/or jobs submitted. **Note:** The Job counts graph requires the Collection Summary analysis is ran.

connections Job Watcher - #1 Job Watcher - #3

Job counts

Temporary storage

Page allocations

I/O

Logical I/O

IFS

Report folder	Description
Job counts	
Jobs submitted	
Job counts rankings	Ranks job counts by job grouping
Jobs submitted rankings	Jobs submitted rankings by job grouping

Job counts folder

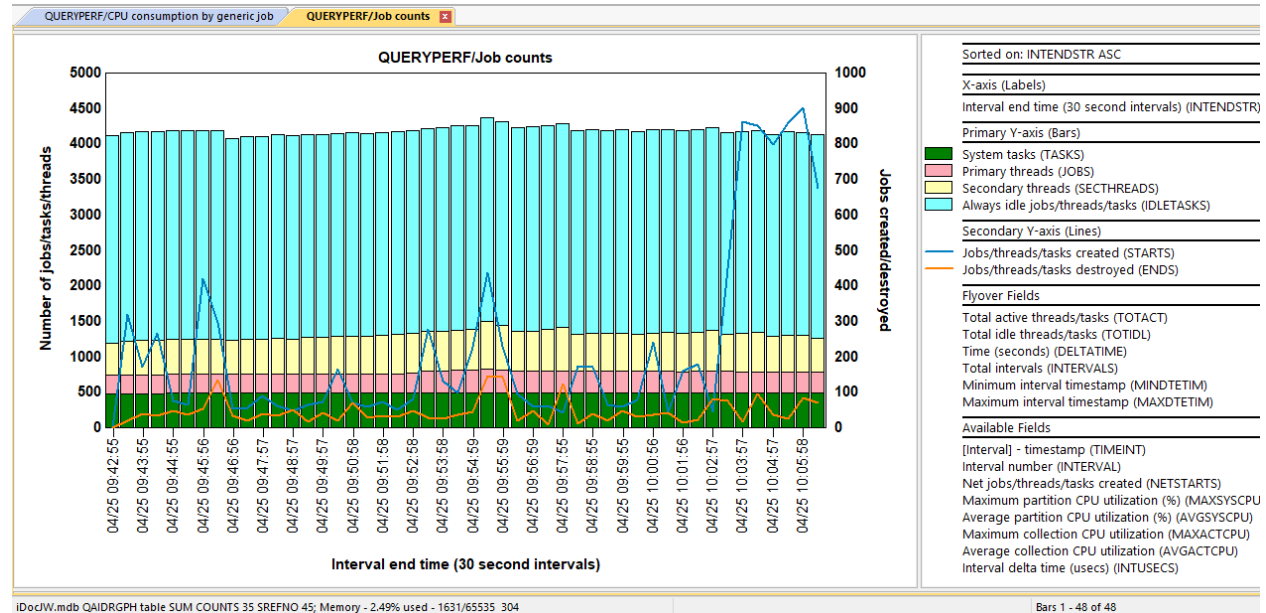
The job counts graph shows the total jobs (primary threads), system tasks and threads (secondary threads) that exist on the system.

Tip: 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 guaranteed.

19.1 Job counts

This graph shows the total number of active tasks, processes (primary threads) and secondary threads over time. The number of created and destroyed taskcounts 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.

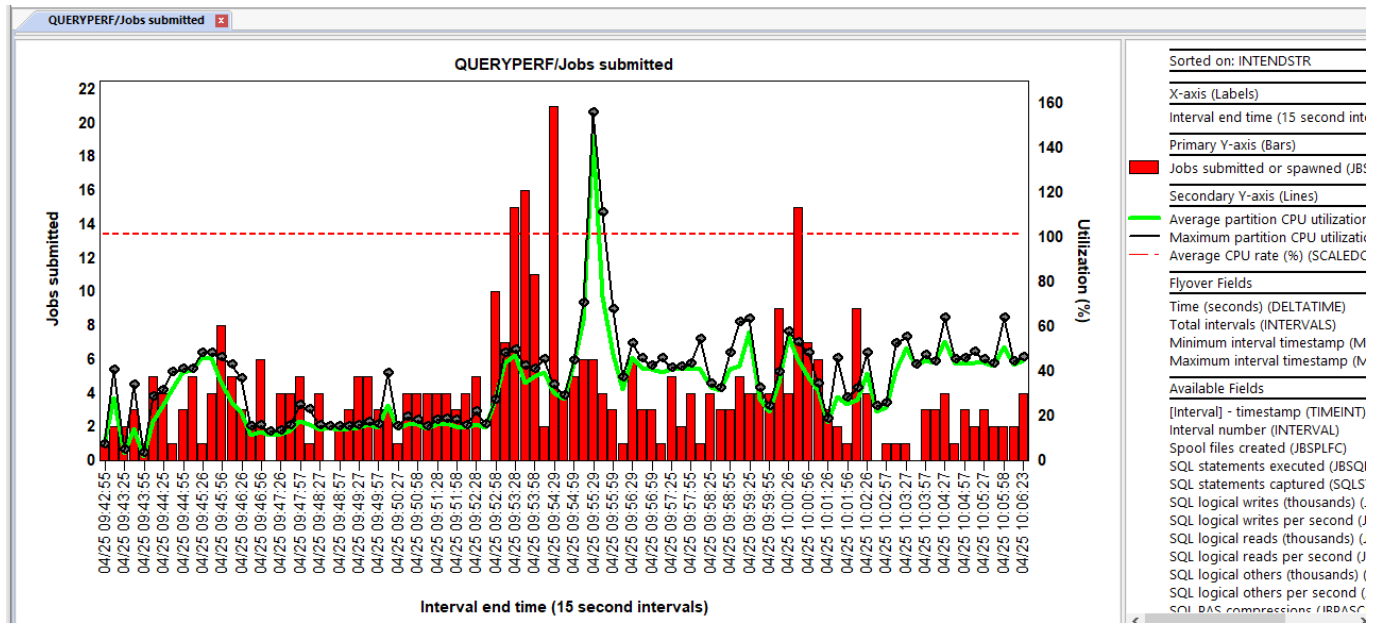


Job counts

Note: The only way to avoid seeing the "Always idle jobs/threads/tasks" is to either use Collection Services Investigator to get more accurate counts (recommended), or 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 and could be resource intensive.

19.2 Jobs submitted

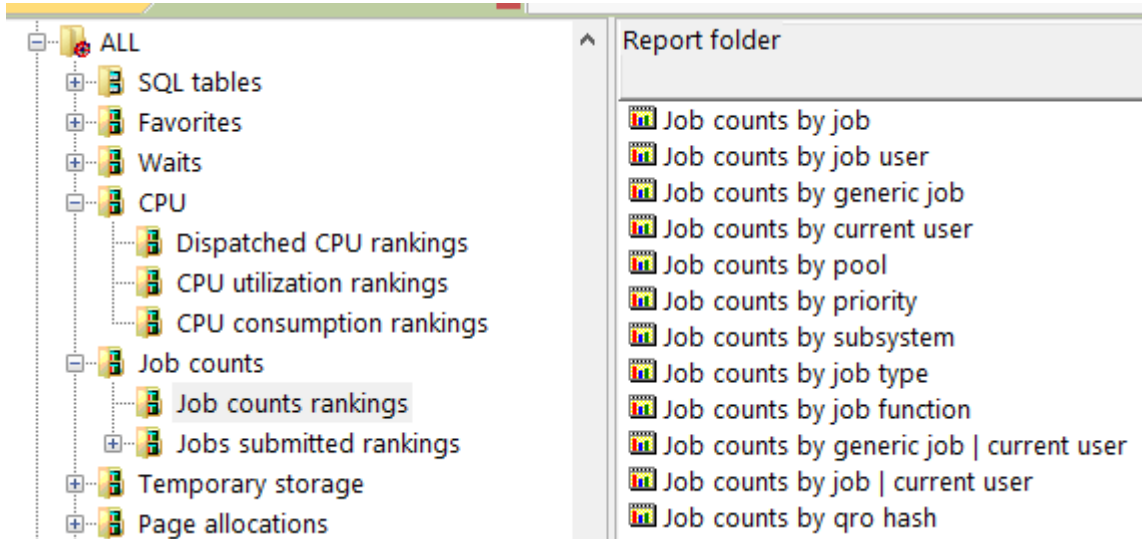
This graph displays the number of submitted jobs over time during the collection. The Y2-axis contains CPU utilization metrics.



Jobs submitted

19.3 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.

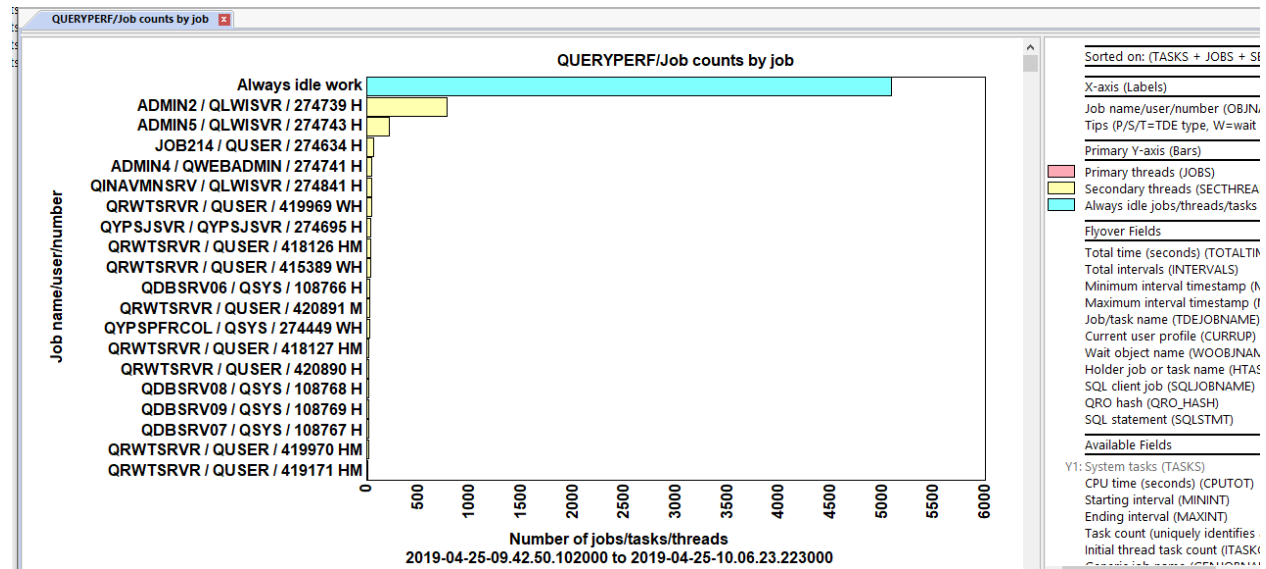


Job counts rankings

19.3.1 Job counts by job

This graph ranks the total number of job/tasks/thread by job. This will show you jobs with the highest number of secondary threads.

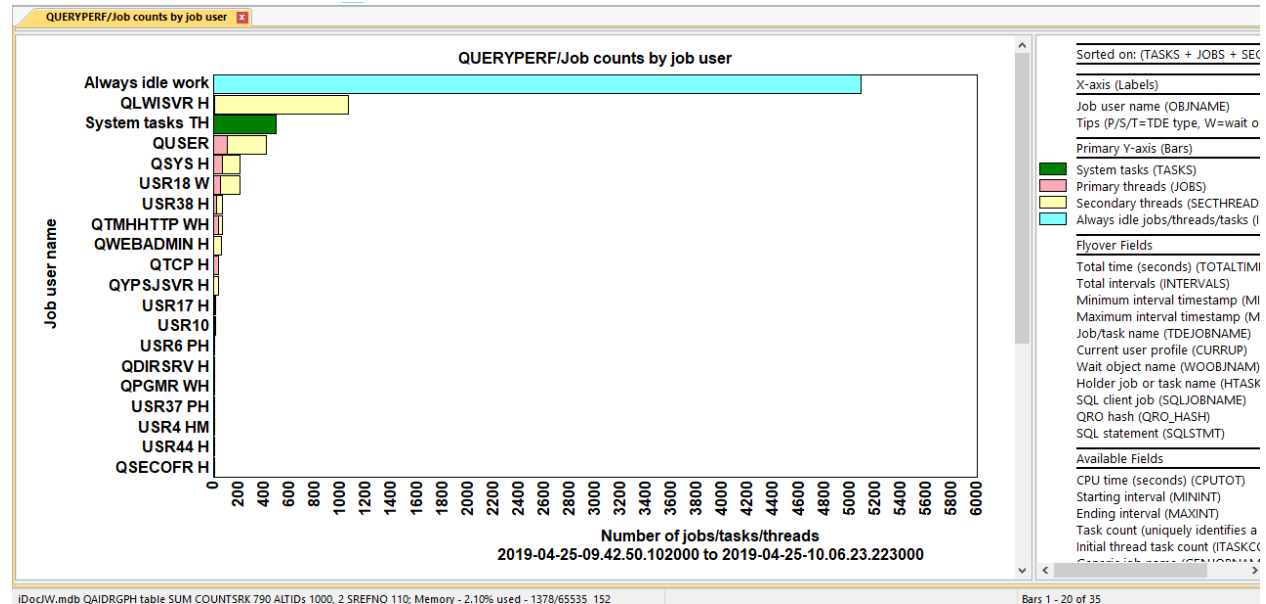
Note: This will typically include the always idle bucket discussed in the previous section.



Job counts by job

19.3.2 Job counts by job user

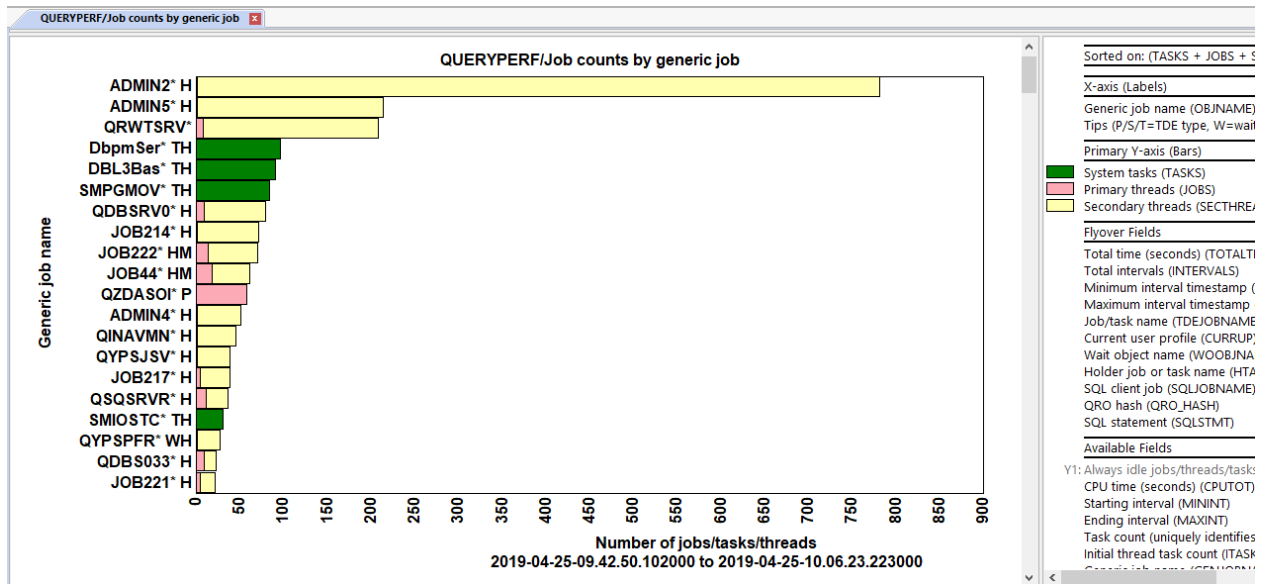
This graph shows the total job counts ranked by job user name. All system tasks are grouped together in this graph.



Job counts by job user

19.3.3 Job counts by generic job

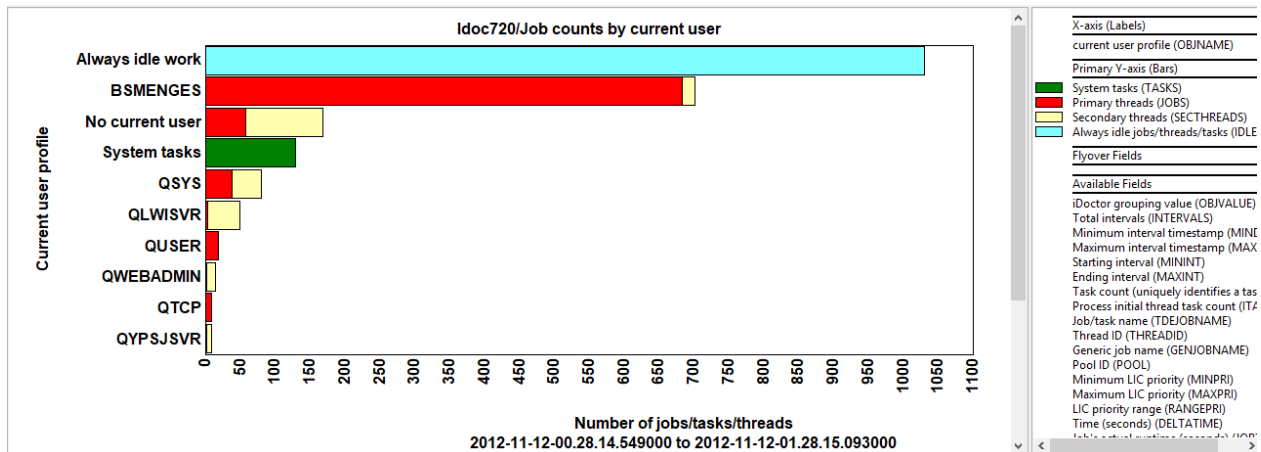
This graph shows the total job counts ranked by generic job name or generic task name.



Job counts by generic job

19.3.4 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."

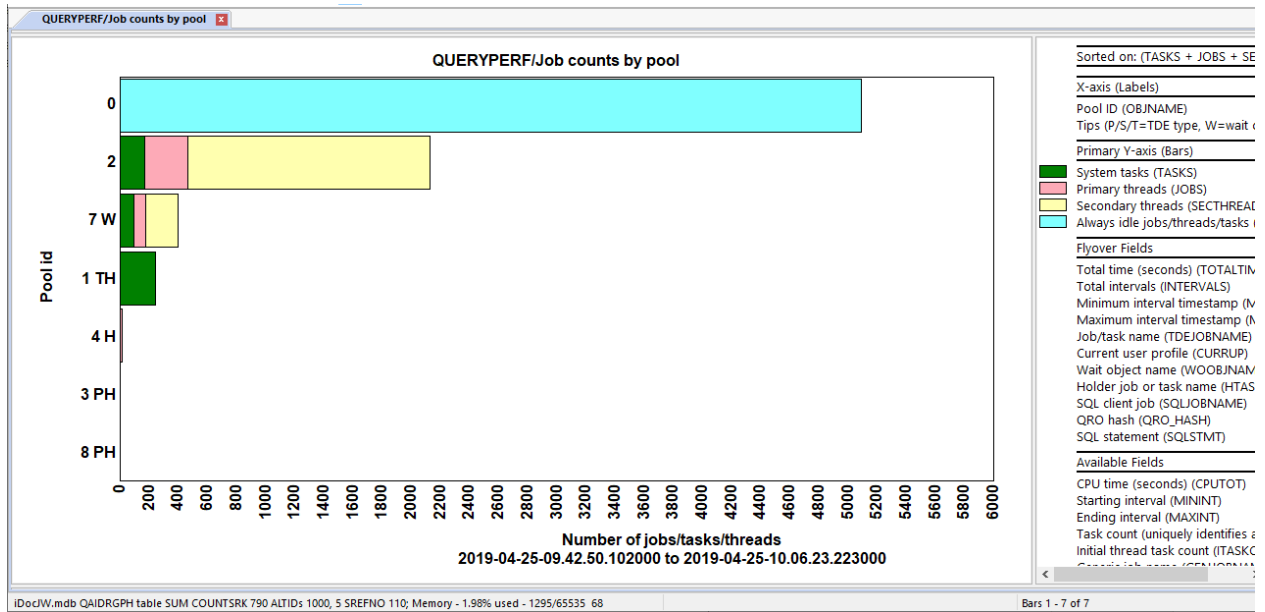


Job counts by current user

19.3.5 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 where the pool is unknown.

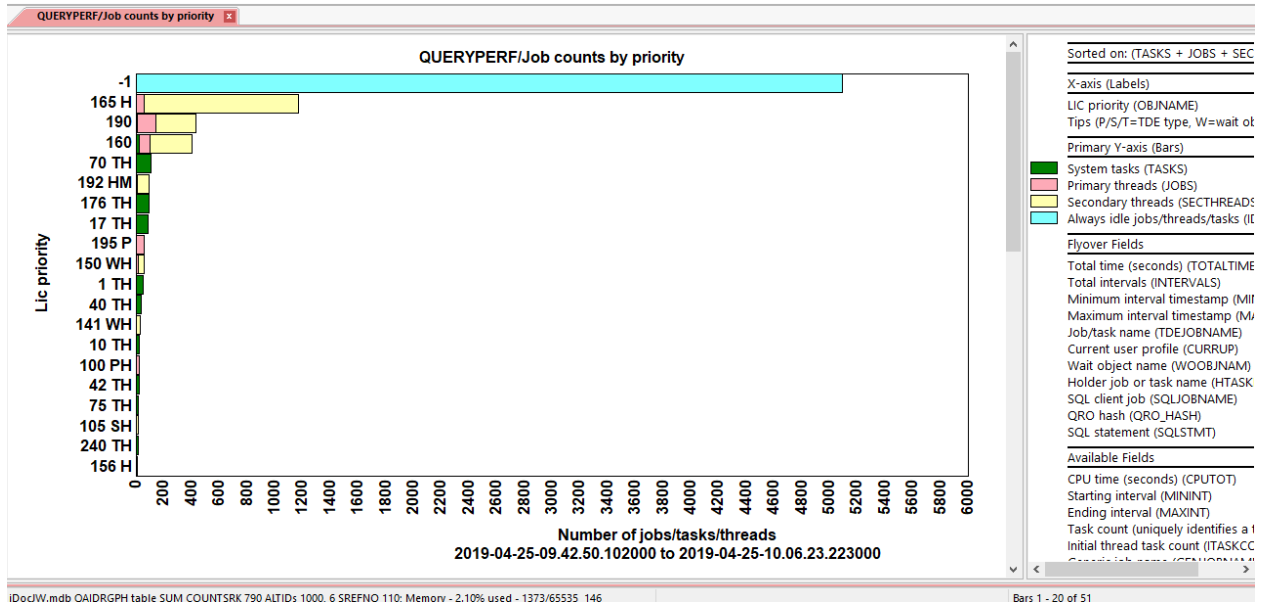


Job counts by pool

19.3.6 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.

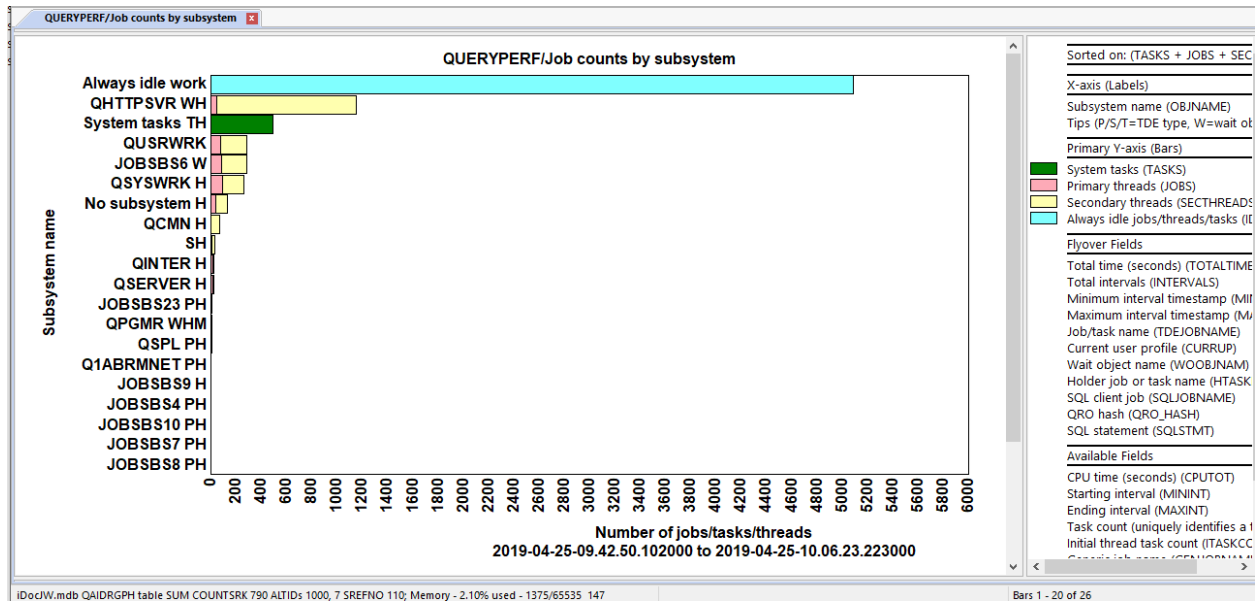
Note: -1 is not a valid LIC priority, but these are from the always idle bucket and the actual priority value is unknown!



Job counts by priority

19.3.7 Job counts by subsystem

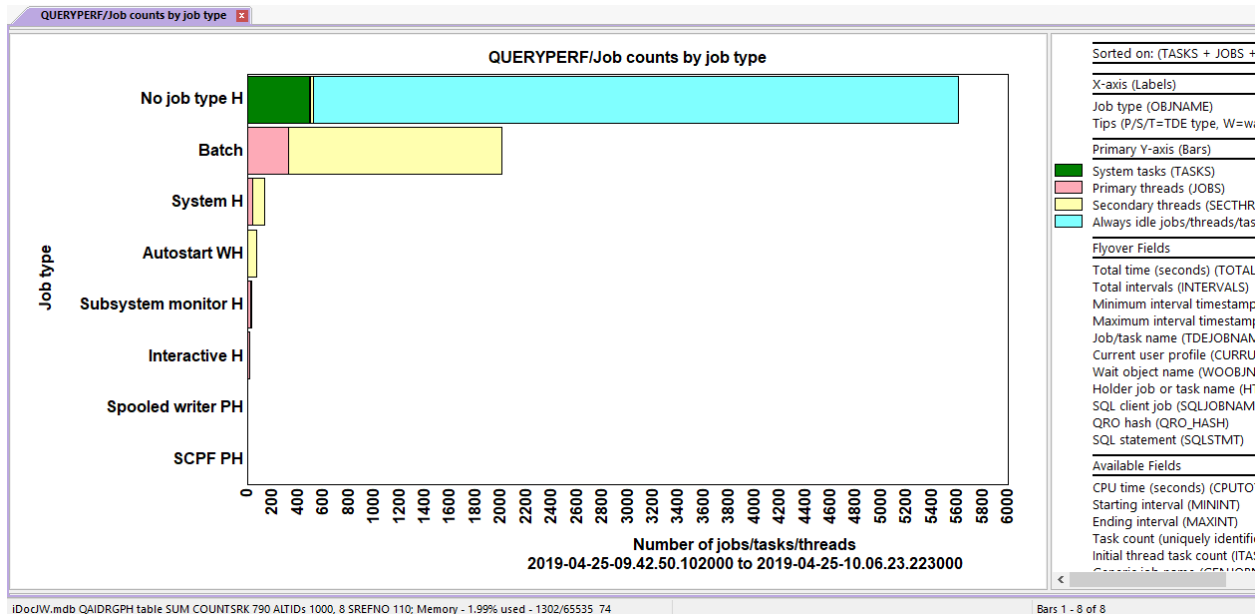
This graph shows the total job counts ranked by subsystem. This graph will include a System tasks bar that groups all the system task counts together. A "No subsystem" entry is also included in this graph.



Job counts by subsystem

19.3.8 Job counts by job type

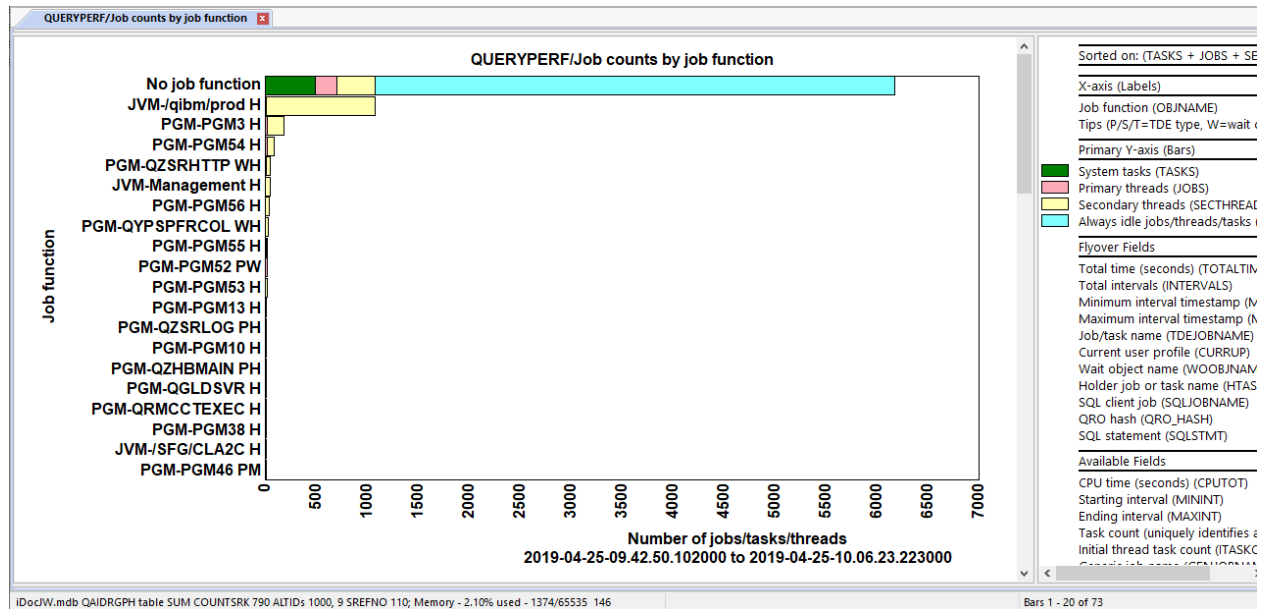
This graph shows the total job counts ranked by job type (i.e. Batch, Interactive, etc). A "No job type" entry is shown on this graph.



Job counts by job type

19.3.9 Job counts by job function

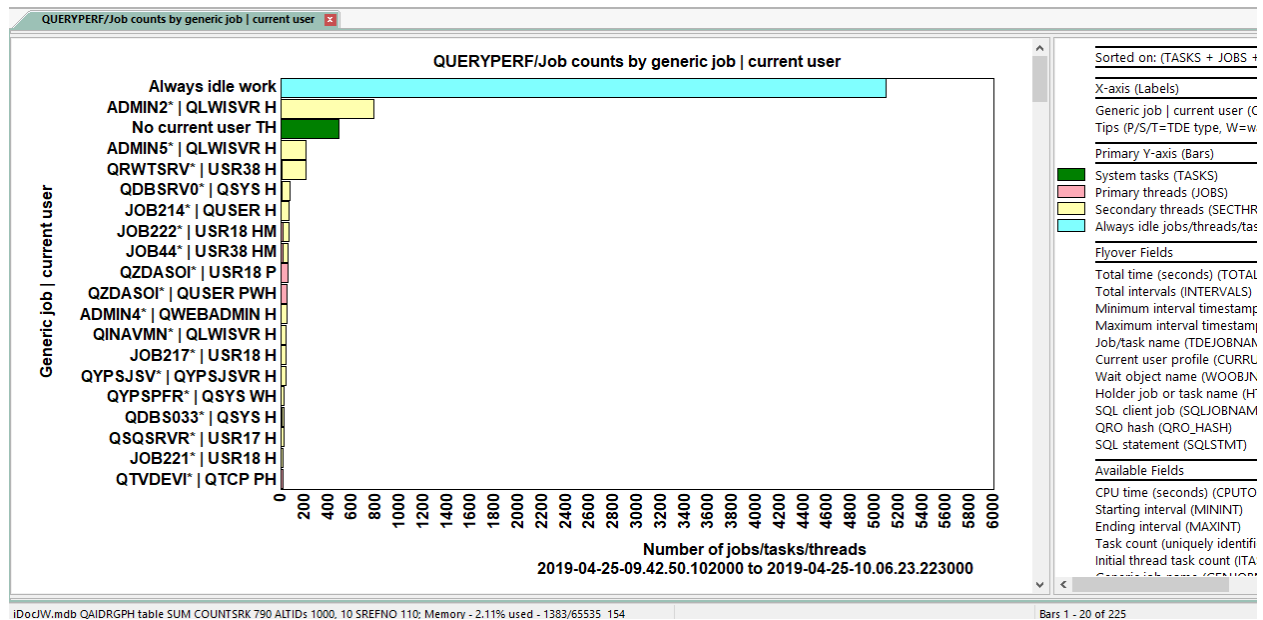
This graph shows the total job counts ranked by job function (i.e. as shown in WRKACTJOB.) A "No job function" entry is shown on this graph.



Job counts by job function

19.3.10 Job counts by generic job | current user

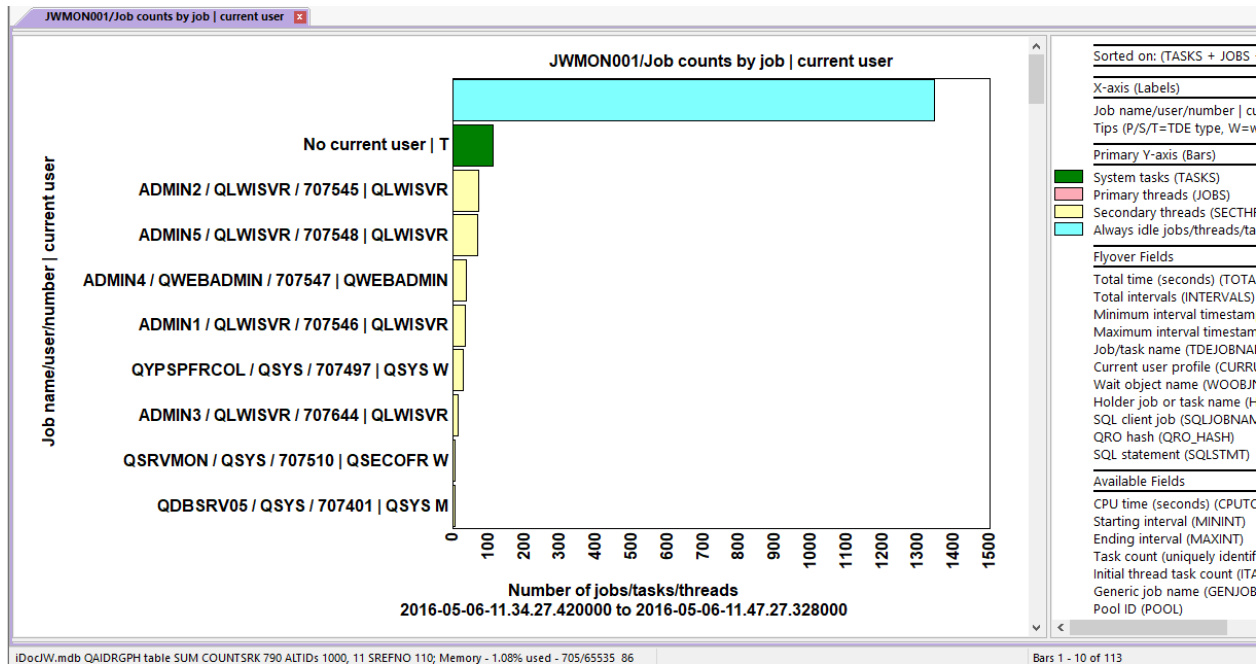
This graph shows the total job counts ranked by generic job and current user combination. A “No current user” entry is included in this graph.



Job counts by generic job | current user

19.3.11 Job counts by job | current user

This graph shows the total job counts ranked by job and current user combination. A “No current user” entry is included in this graph.



Job counts by job | current user

19.3.12 Job counts by QRO hash

This graph shows the total job counts grouped by QRO hash. A “No QRO hash” entry is included in this graph.

20 Temporary storage

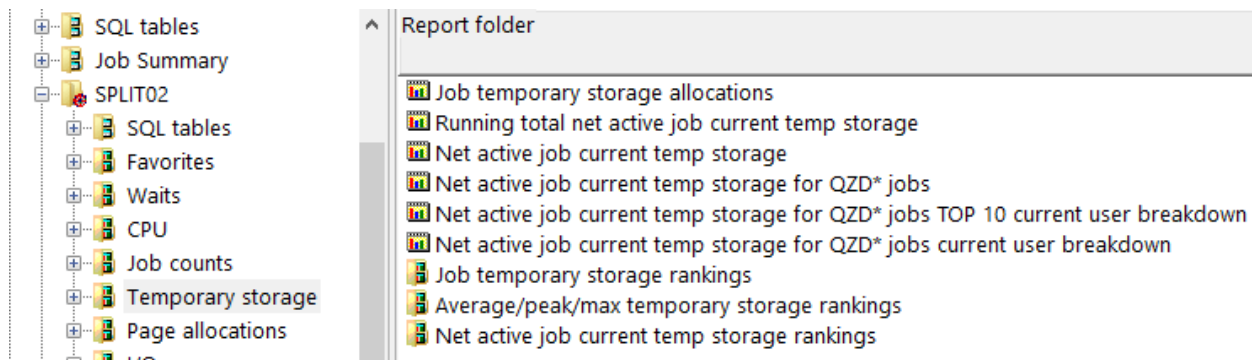
These graphs show the estimated consumption of temporary storage used by jobs on the system.

Note: This folder requires the **Collection summary** analysis to be ran. Some graphs such as the “running totals” graphs only appear at 7.3+.

Keep in mind that Job Watcher may miss and not capture very short-lived jobs which may result in the temporary storage values shown being less than reality. **Jobs that lived and died within a single 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.

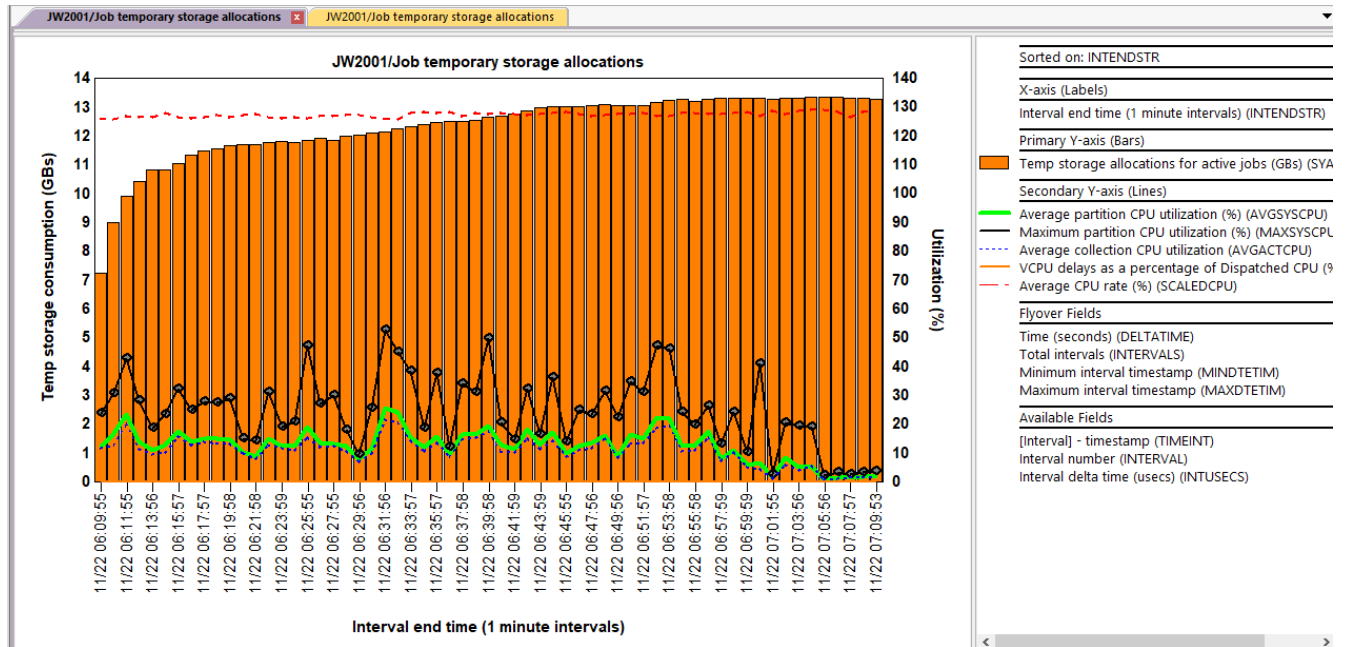
Note: Because the total disk space of the system ASP within the Job Watcher metrics is not known it is not possible to give temporary storage as a percentage of the system ASP.



Temporary storage

20.1 Job temporary storage allocations

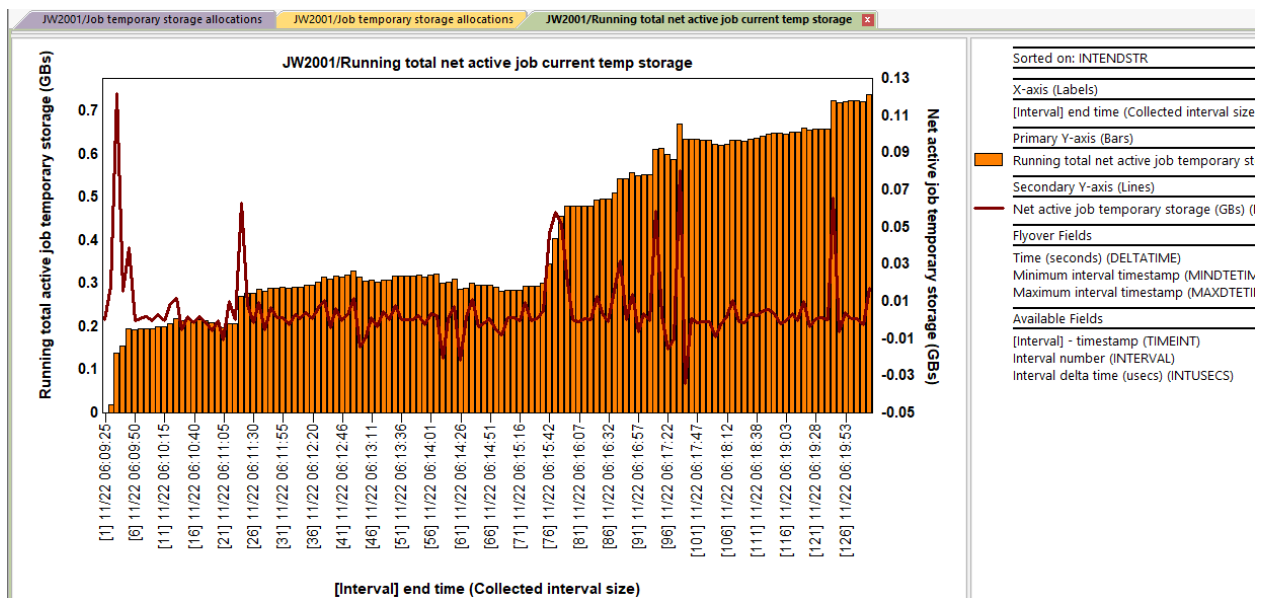
This graph adds up the current temporary storage allocations across all jobs on the system and displays it over time. Please note that this total may likely not include 100% percent of all jobs because jobs that never used CPU are not included typically in Job Watcher (unless the force 1st interval option is used on the JW definition) and jobs that are short-lived are not included either.



Job temporary storage allocations

20.2 Running total new active job current temp storage

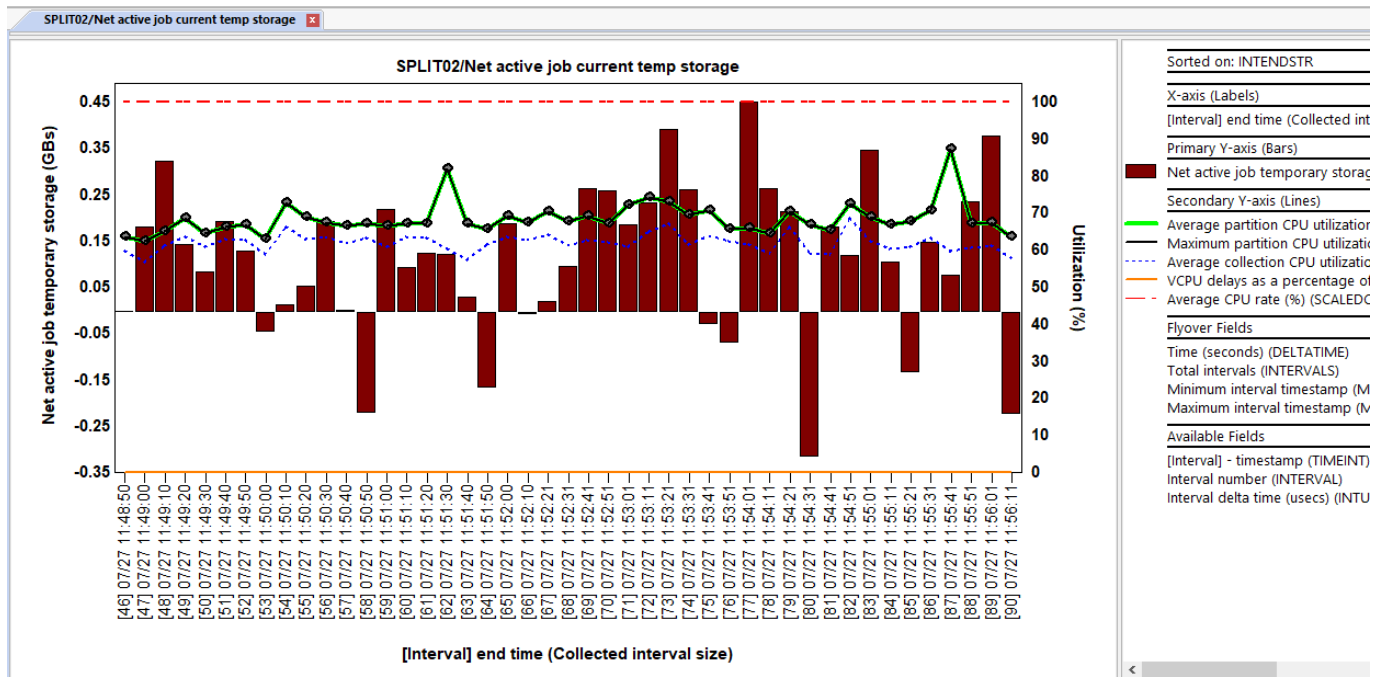
This graph displays the cumulative increases or decreases to active job temporary storage over time. The 2nd Y-axis displays the net (increases – decreases) temporary storage. **This graph requires 7.3+.**



Running total net active job current temp storage

20.3 Net active job current temp storage

This graph displays the net increases or decreases in temp storage over time.



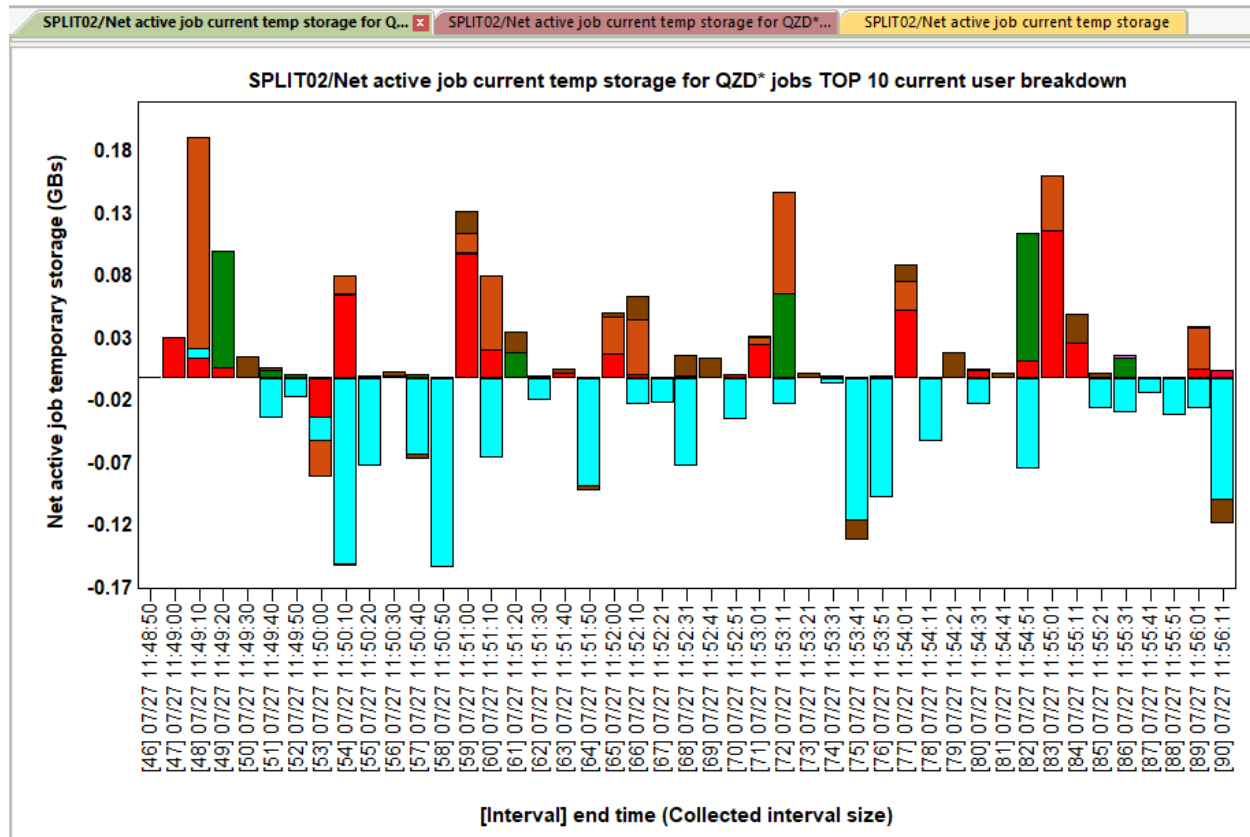
Net active job current temp storage

20.4 Net active job current temp storage for QZD* jobs

This graph is the same as the previous one, except only shows contributions from jobs named QZD*.

20.5 Net active job current temp storage for QZD* jobs TOP 10 current user breakdown

This graph is the same as the previous one except uses a different color for each of the top current users contributing to temp storage growth.



Net active job current temp storage for QZD* jobs TOP 10 current user breakdown

20.6 Net active job current temp storage for QZD* jobs current user breakdown

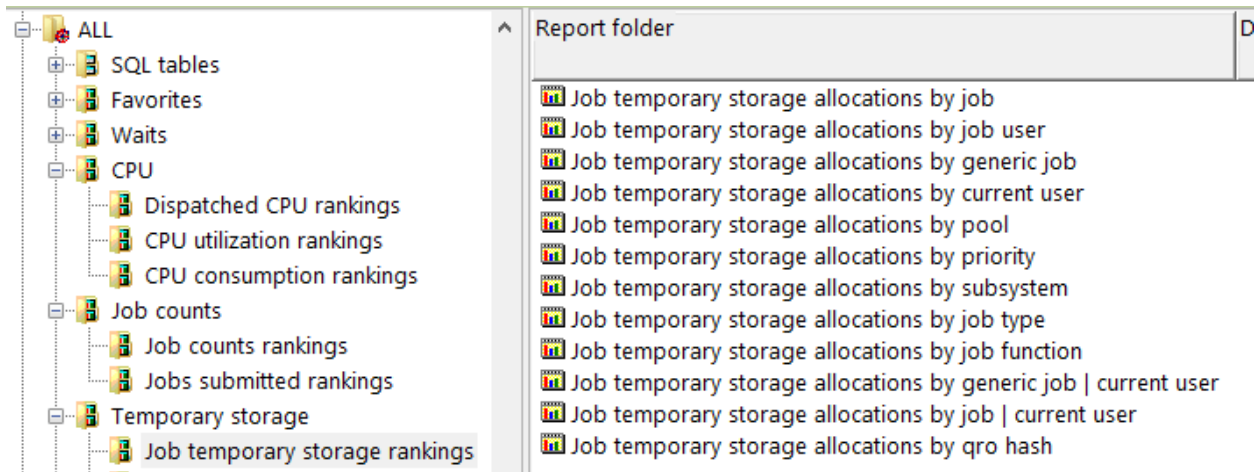
This graph is identical to the previous graph except does not limit the number of colors to the top 10 current user profiles. **Note:** On very large systems, this may not work well or at all if the number of users is too high.

20.7 Job temporary storage rankings

These graphs rank the temporary storage consumption for jobs found in the collection. These statistics are based on job-based metrics found in QAPYJWPRC and do not include any system tasks.

The metrics in this graph include:

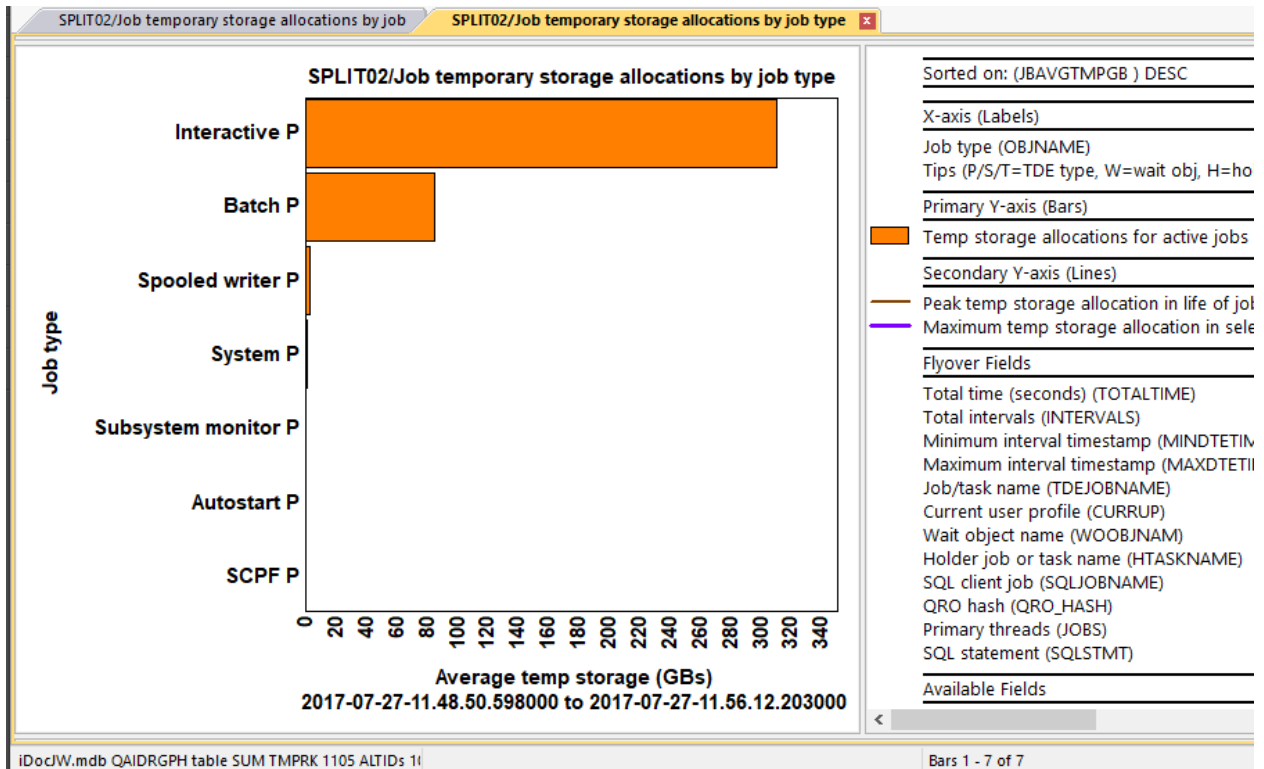
- 1) Peak temp storage allocations for life of the job
- 2) Maximum temp storage allocations in the current time range
- 3) Average temp storage allocations



Temporary storage -> Job temporary storage rankings

Note: These graphs offer an optional Y2-axis showing the peak and maximum temp storage used for each job grouping. Use toggle graph format toolbar button to view.

An example is provided below:



Job temporary storage allocations by job type

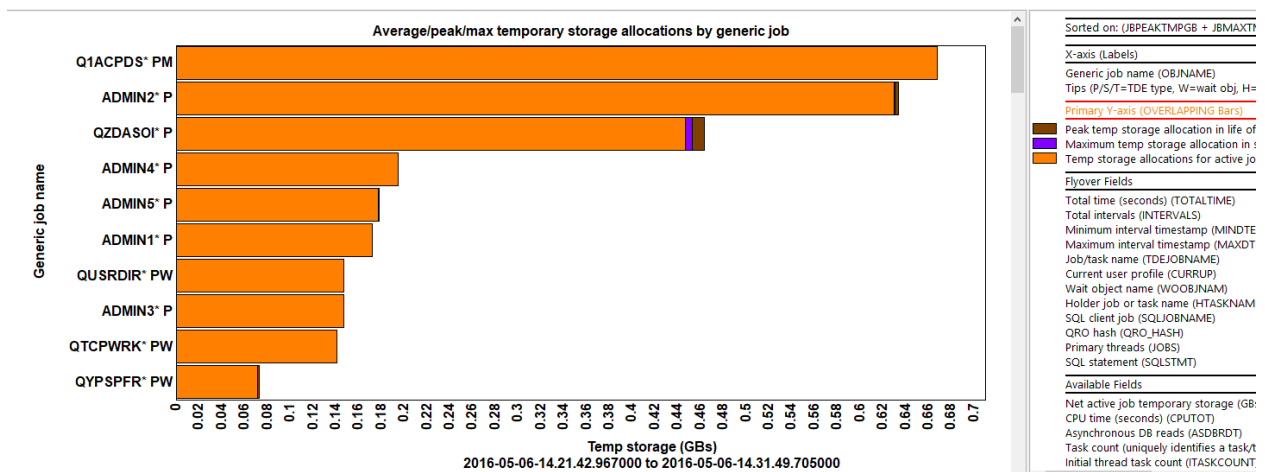
20.8 Average/peak/max temporary storage rankings

These graphs show the same data as in the previous chart but uses an overlapping bar graph instead and do not provide a 2nd Y-axis.

The screenshot shows the IBM iDoctor interface. On the left, a tree view under 'ALL' includes categories like SQL tables, Favorites, Waits, CPU, Job counts, and Temporary storage. The 'Temporary storage' category is expanded, showing 'Job temporary storage rankings' and 'Average/peak/max temporary storage allocations by job', which is highlighted. On the right, the 'Report folder' pane lists various reports, with 'Average/peak/max temporary storage allocations by job' selected at the top.

Temporary storage -> Average/peak/max temporary storage rankings

An example is provided below:



Average/peak/max temporary storage allocations by generic job

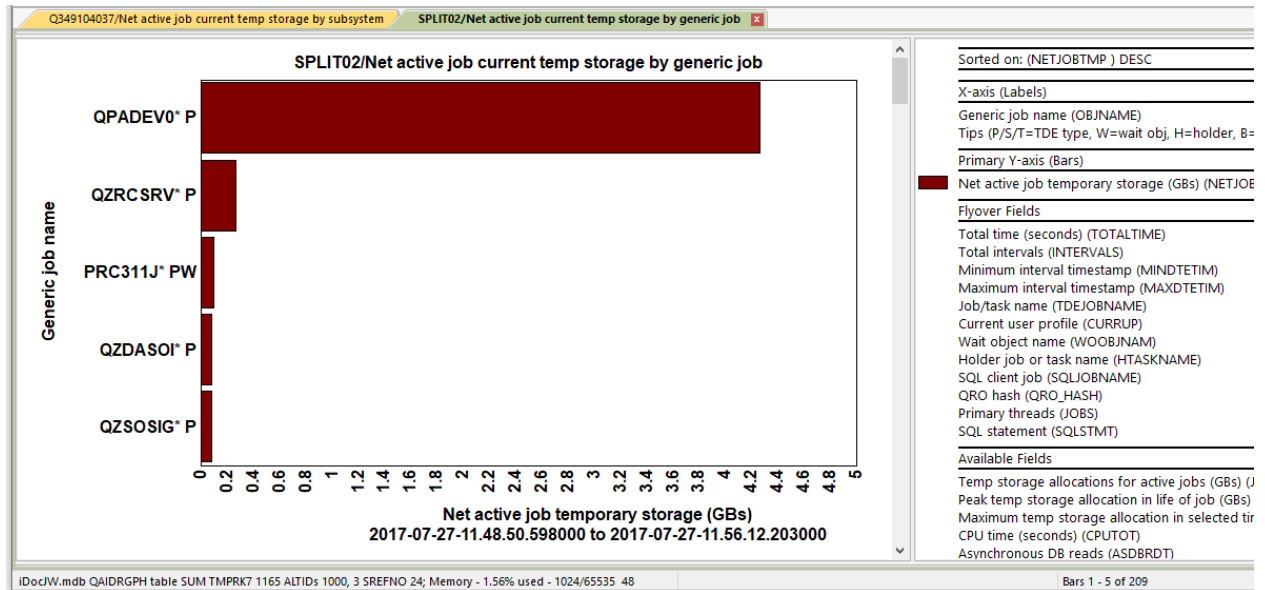
20.9 Net active job current temp storage rankings

This graph displays the net increases or decreases in temp storage grouped in one of several ways.

The screenshot shows the IBM iDoctor interface. On the left, the 'Temporary storage' category is expanded, and 'Net active job current temp storage by job' is highlighted. On the right, the 'Report folder' pane lists various reports, with 'Net active job current temp storage by job' selected at the top.

Temporary storage -> Net active job current temp storage rankings

An example is provided below:



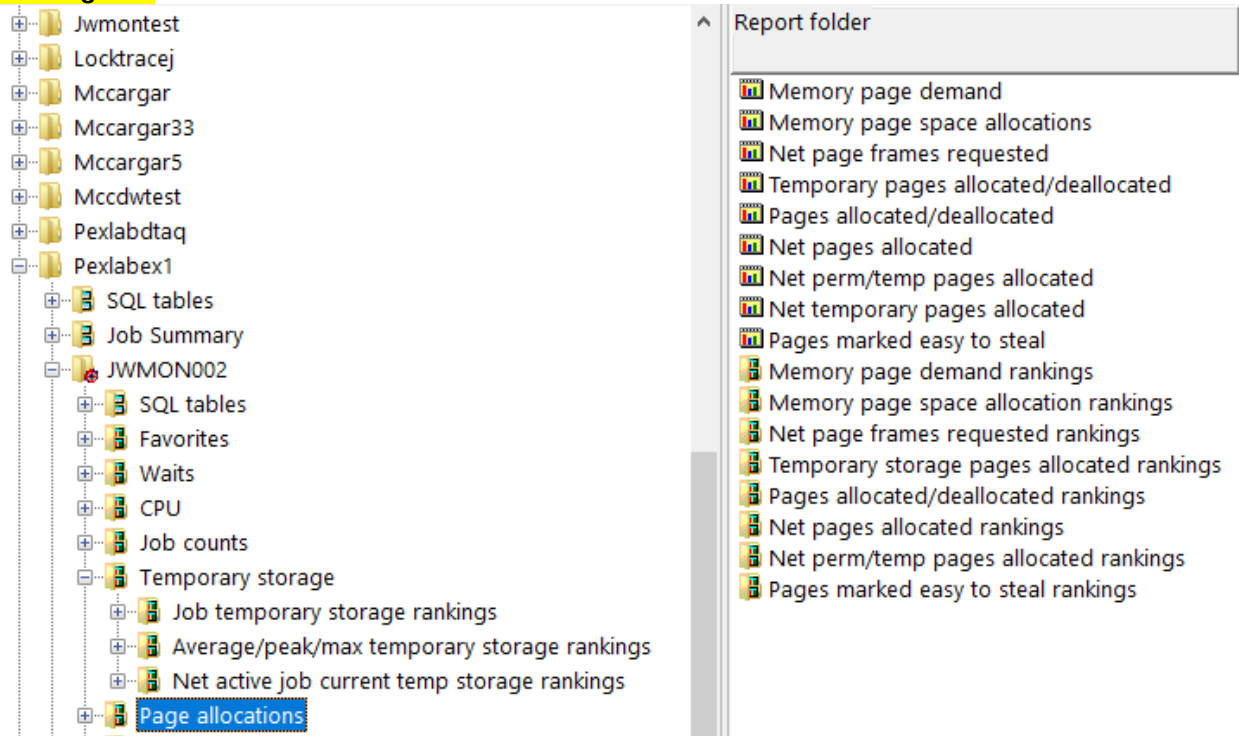
Net active job current temp storage by generic job

21 Page allocations

This folder contains graphs related to memory page usage in Job Watcher.

Note: Some graphs will only appear if the **Collection Summary** analysis has been ran.

Tip: For graphs related to memory pool sizes and consumption you will need to use **Collection Services Investigator**.

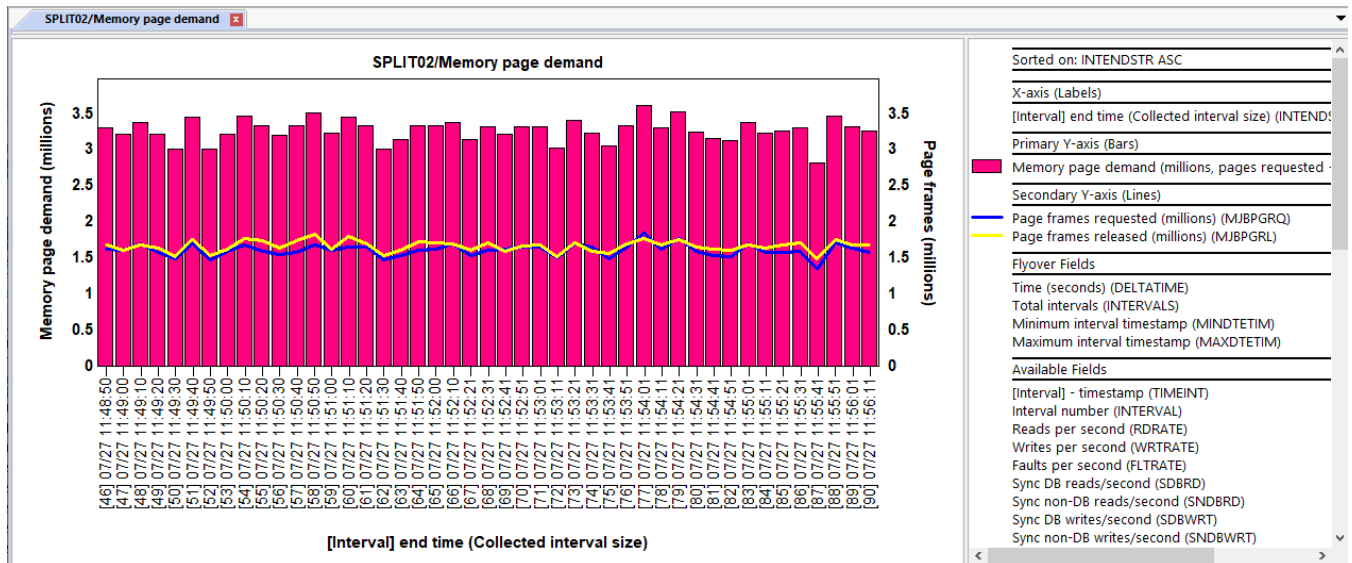


Page allocations folder

21.1 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.

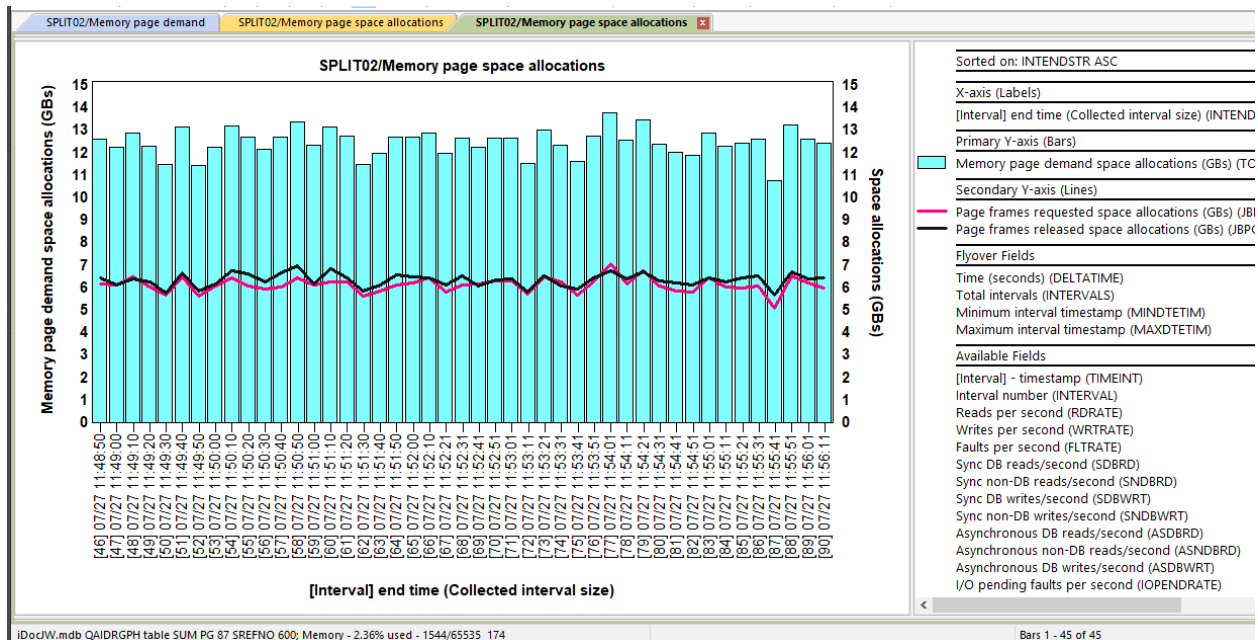
The bars display the total of the page frames requested and released. The 2nd Y-axis displays the 2 values as separate lines instead.



Memory page demand

21.2 Memory page space allocations

This graph is the same as the previous graph but rather than showing 4K page counts it shows the values in Gigabytes.

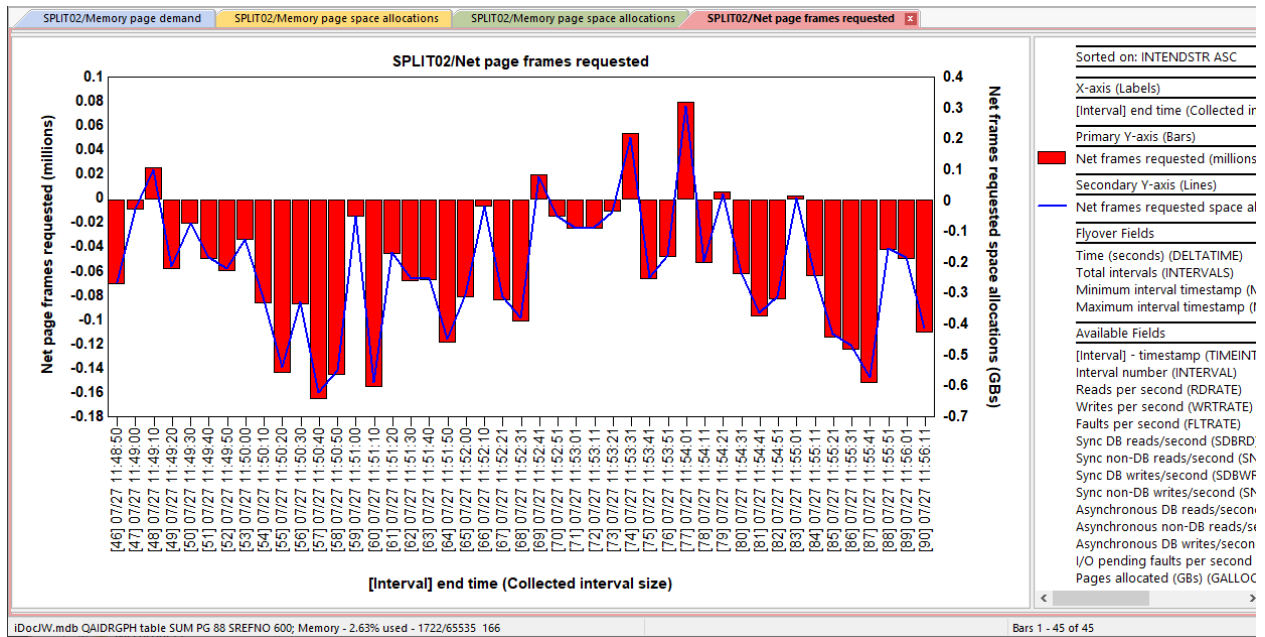


Memory page space allocations

21.3 Net page frames requested

This graph shows the net 4K page frames requests (FRMESTOL – SREMOVE in file QAPYJWTDE.)

The net space allocations are provided on the 2nd Y-axis on this graph.



Net page frames requested

21.4 Temporary pages allocated/deallocated

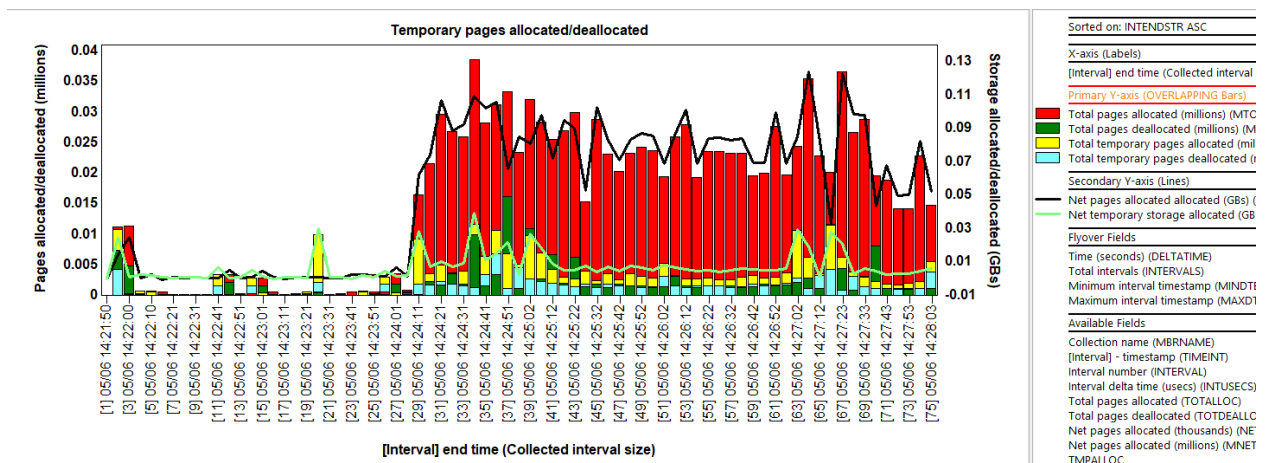
This graph displays 4 metrics, but they are overlapping with the highest values shown at the top of each bar. The smaller values will be displayed at the bottom. The metrics are:

- 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 2nd Y-axis displays the net storage allocated for all pages or just the temp storage pages.

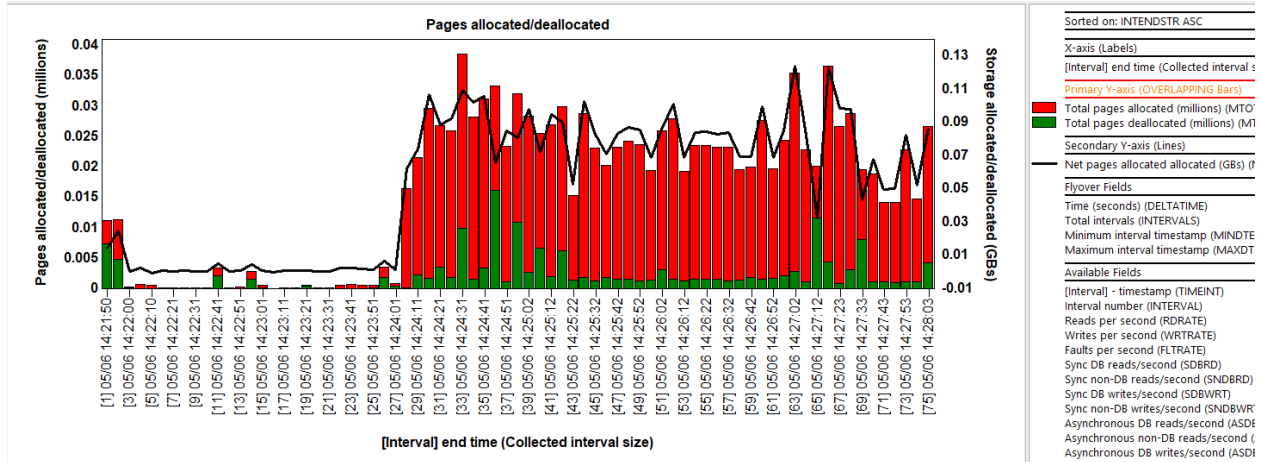
Note: This graph requires the Collection Summary analysis is ran.



Temporary pages allocated/deallocated

21.5 Pages allocated/deallocated

This is a simplified version of the previous graph but removes the temporary storage metrics.

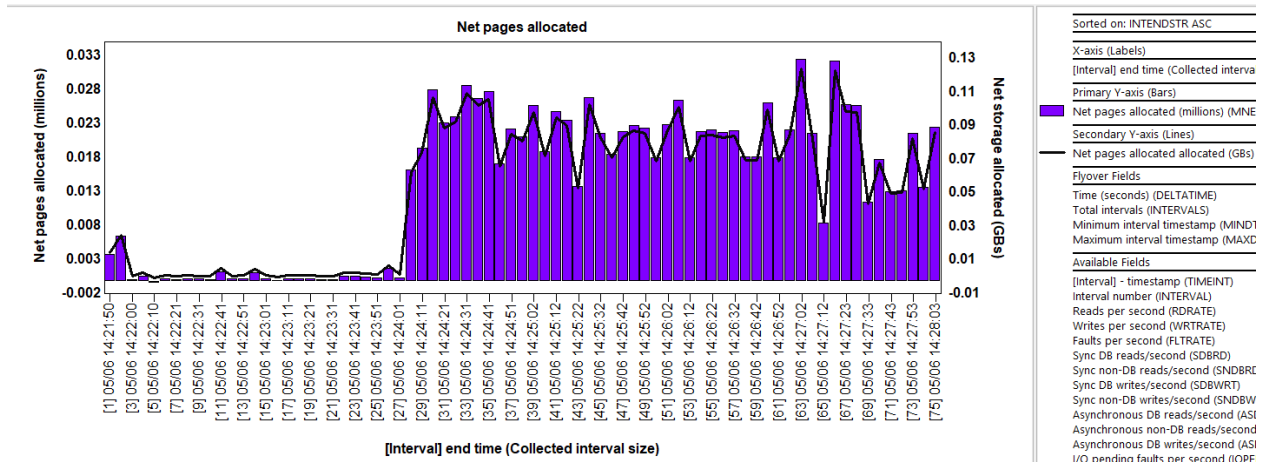


Pages allocated/deallocated

21.6 Net pages allocated

This graph displays the net pages allocated (allocations – deallocations) as well as the net storage allocations in GBs on the 2nd Y-axis.

Note: This graph requires the Collection Summary analysis is ran.

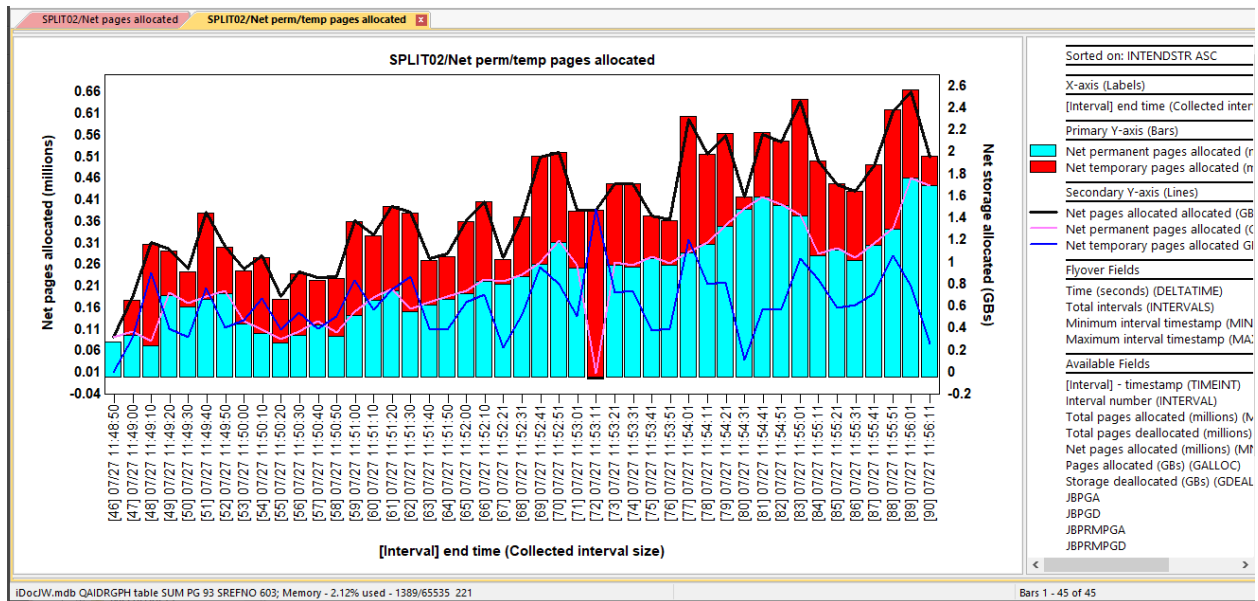


Net pages allocated

21.7 Net perm/temp pages allocated

This graph is the same as the previous graph but divides up permanent vs temporary net page allocations.

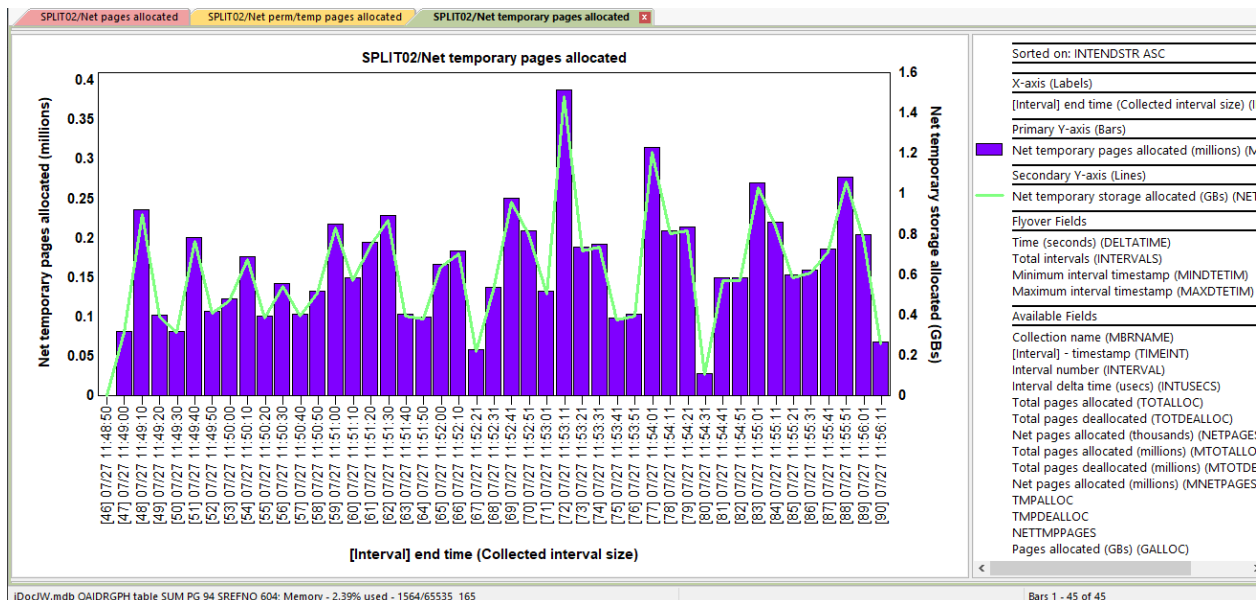
Note: This graph requires the Collection Summary analysis is ran.



21.8 Net temporary storage pages allocated

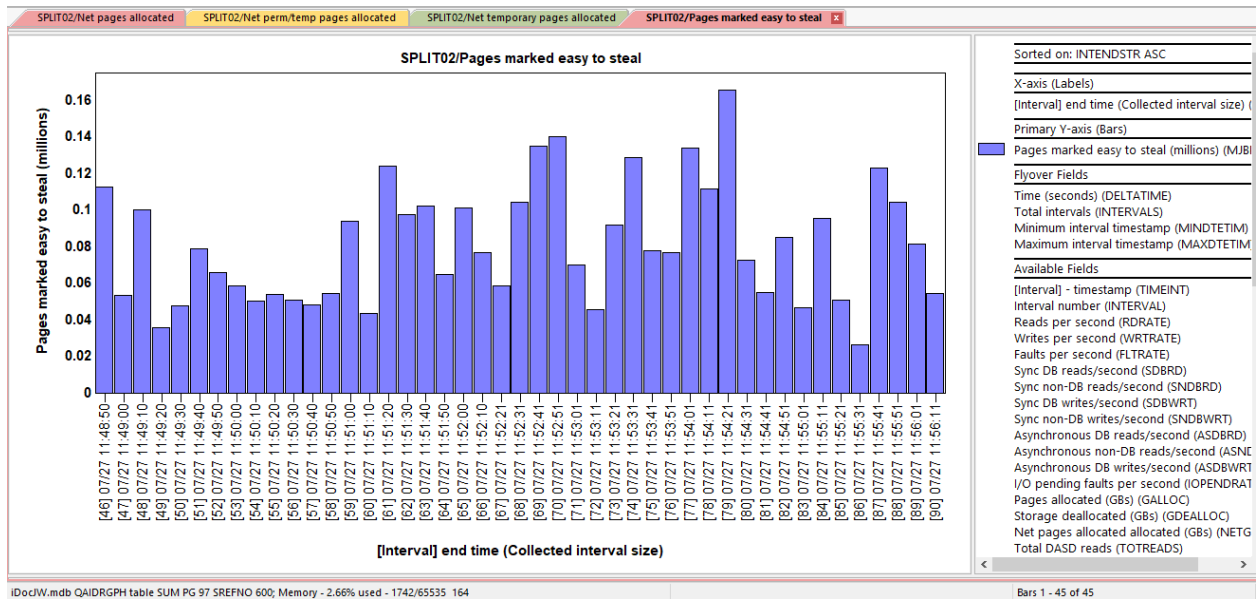
This graph displays the net temporary storage pages allocated over time. It includes the net storage size (in gigabytes) as well on the Y2-axis.

Note: This graph requires the Collection Summary analysis is ran.



21.9 Pages marked easy to steal

This graph displays the total pages marked easy to steal (in millions) for all jobs in the collection.

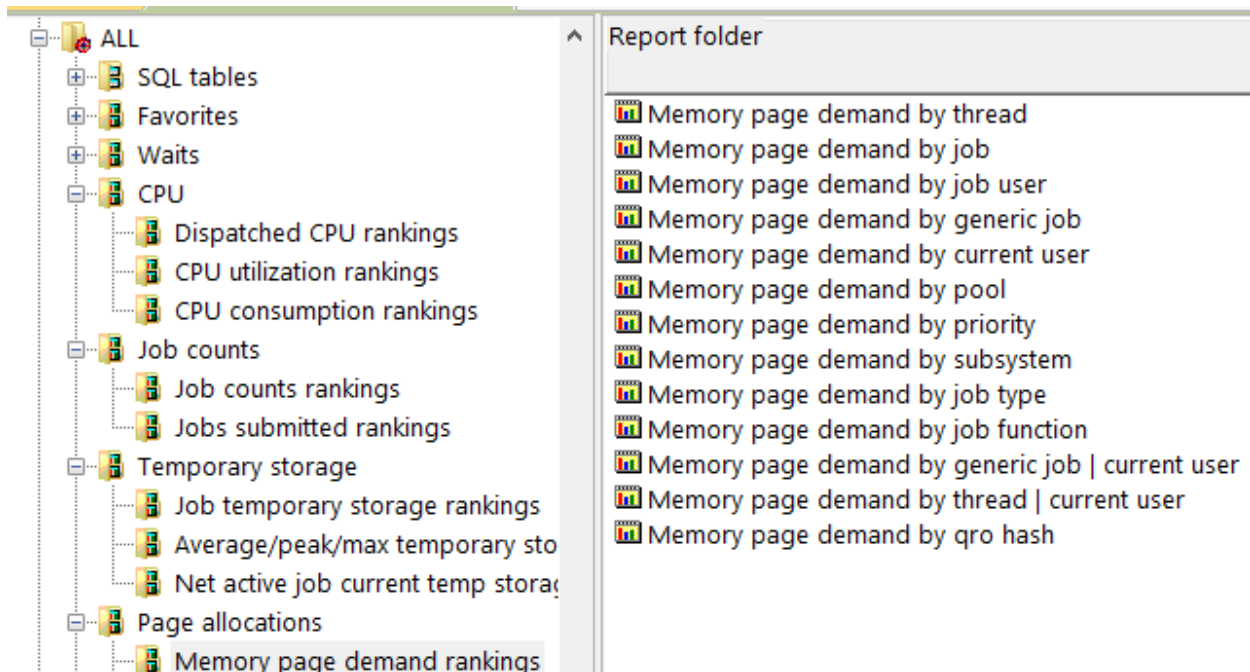


Pages marked easy to steal

21.10 Memory page demand rankings

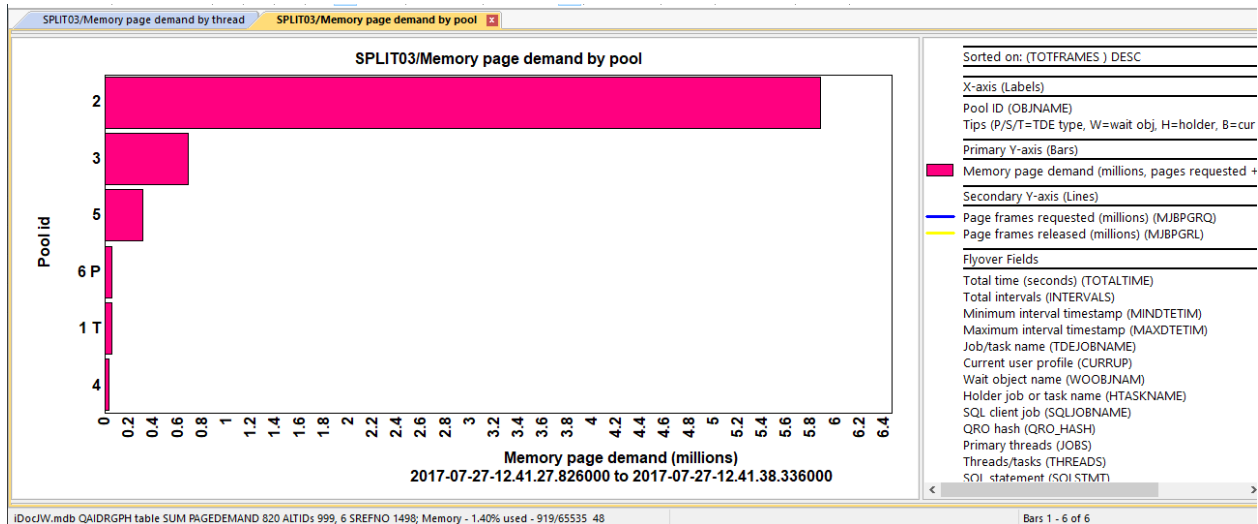
This graph displays the memory pages requested and released which together gives a sense of how much memory is being utilized by the jobs captured by Job Watcher. The data is ranked in several possible ways.

The bars display the total of the page frames requested and released. An optional 2nd Y-axis displays the 2 values as separate lines instead.



Page allocations -> Memory page demand rankings

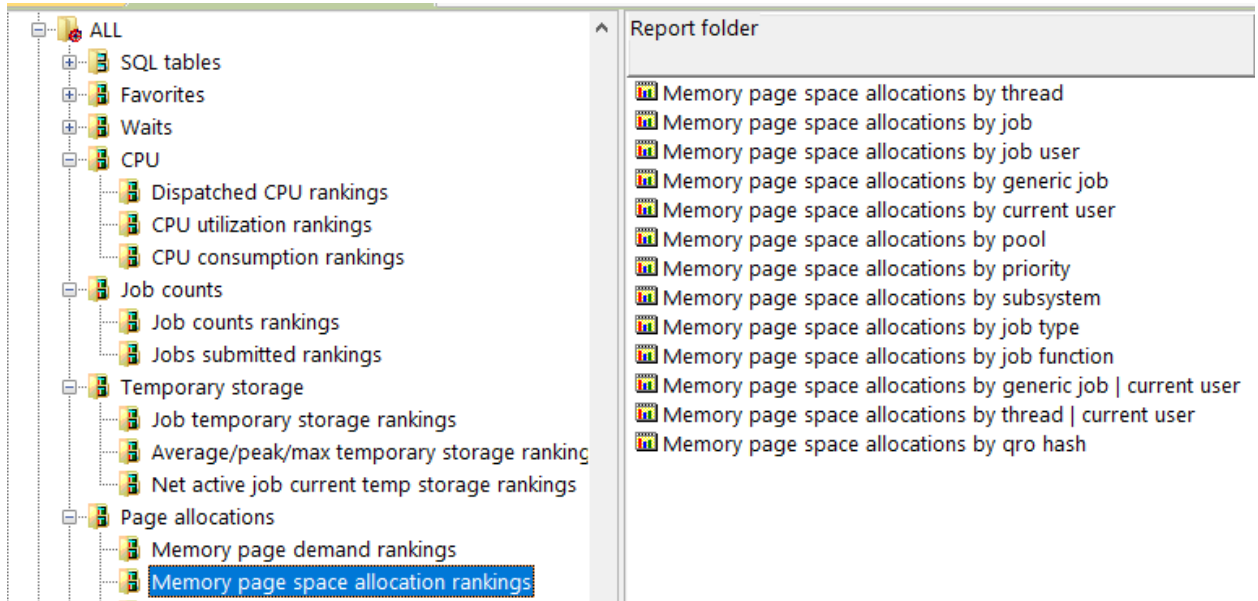
An example follows:



Memory page demand by pool

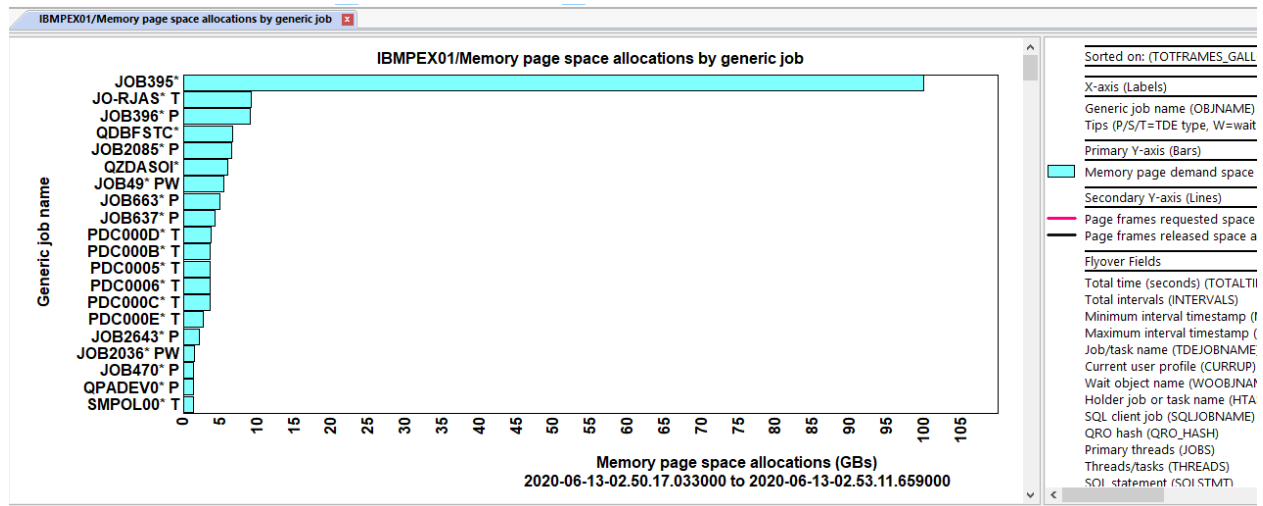
21.11 Memory page space allocations rankings

This graph is the same as the previous graph but rather than showing 4K page counts it shows the values in Gigabytes.



Page allocations -> Memory page space allocation rankings

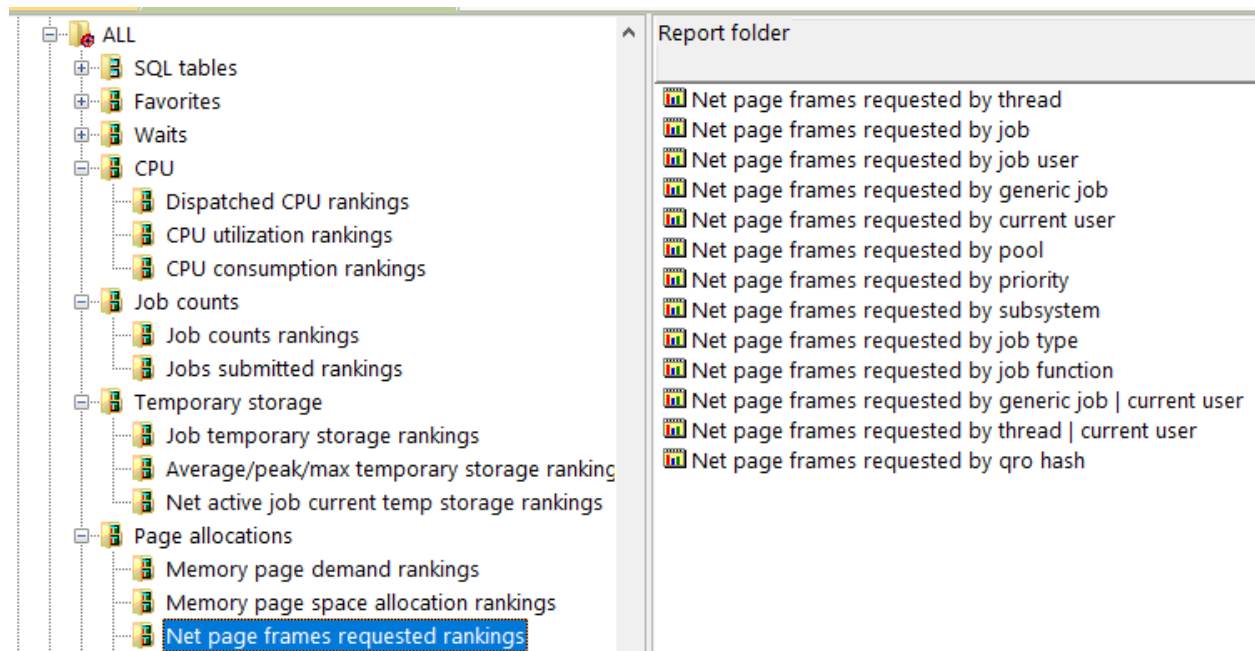
An example follows:



Memory page space allocations by generic job

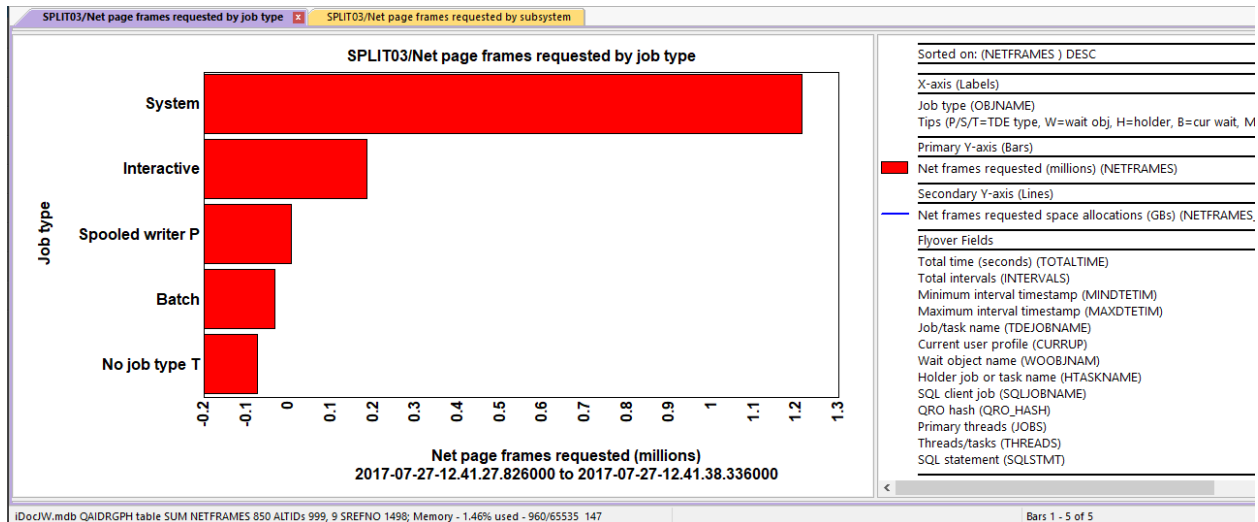
21.12 Net page frames requested rankings

This graph shows the net 4K page frames requests (FRMESTOL – SREMOVE in file QAPYJWTDE.) The net space allocations are provided on the optional 2nd Y-axis on this graph.



Page allocations -> Net page frames requested rankings

An example follows:



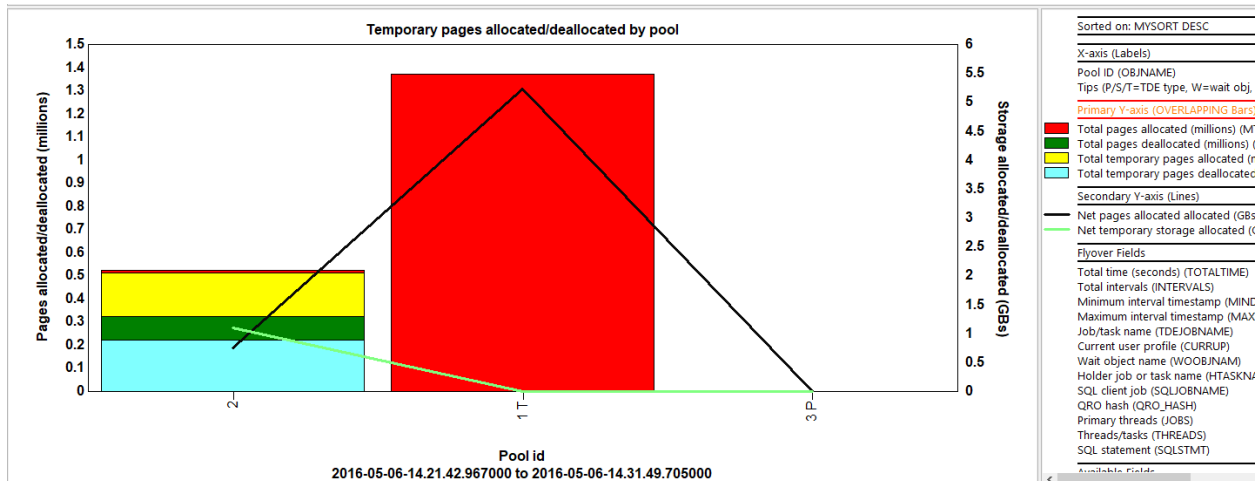
Net page frames requested by job type

21.13 Temporary 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)

An example follows:

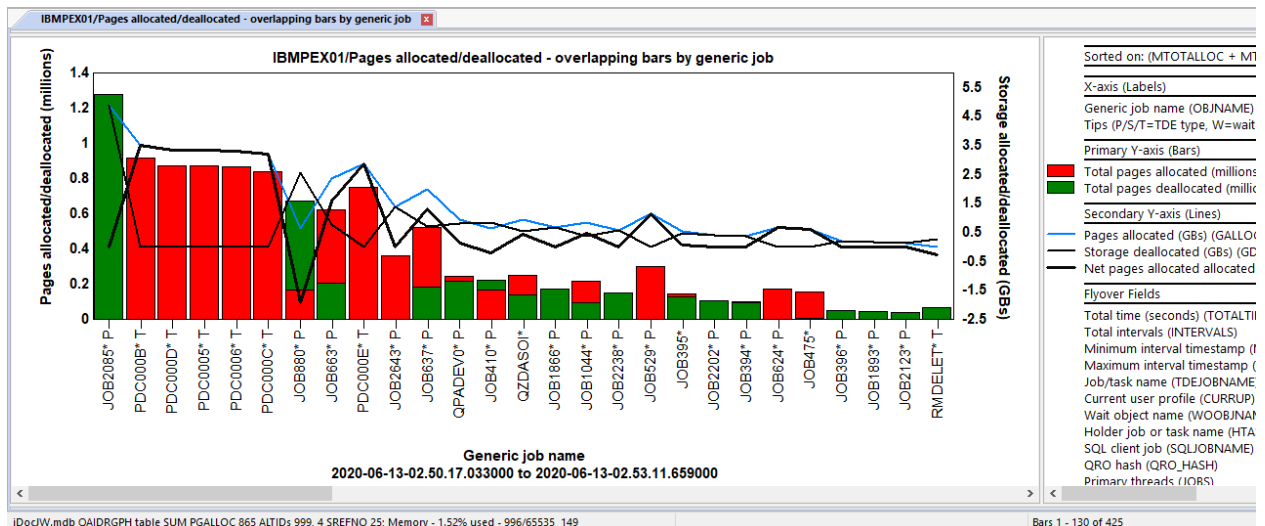


Temporary pages allocated/deallocated by pool

21.14 Pages allocated/deallocated rankings

This is a simplified version of the previous graph but removes the temporary storage metrics.

An example follows:



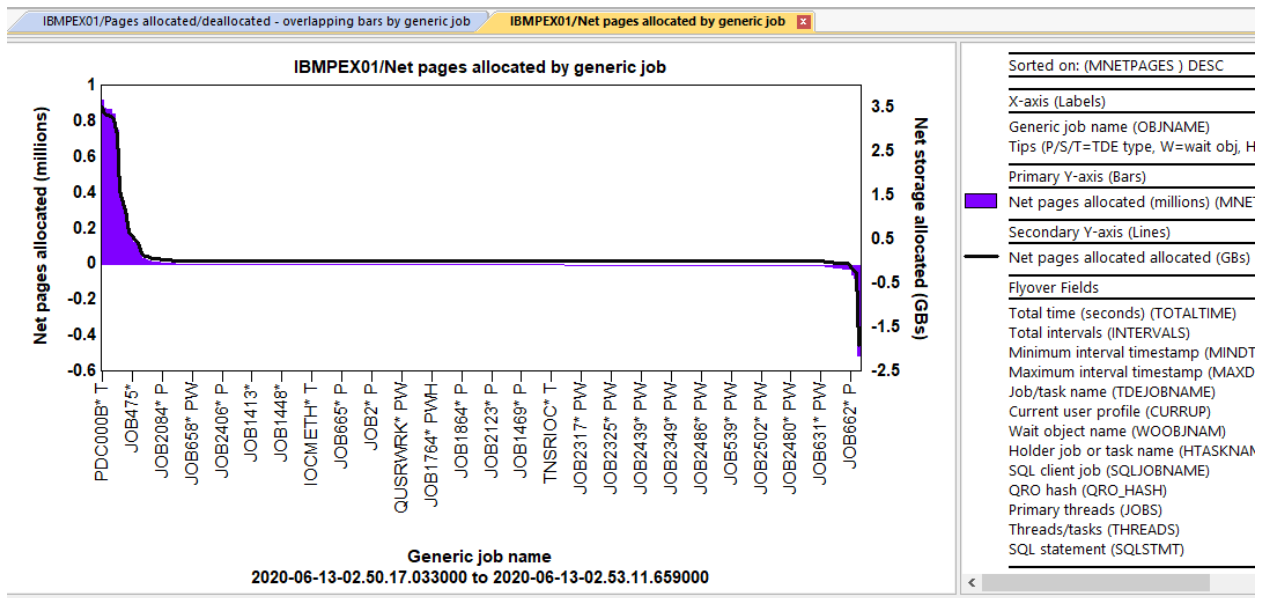
Pages allocated/deallocated by generic job

21.15 Net pages allocated rankings

This graph displays the net pages allocated (allocations – deallocations) as well as the net storage allocations in GBs on the 2nd Y-axis.

Note: This graph requires the Collection Summary analysis is ran.

An example follows:



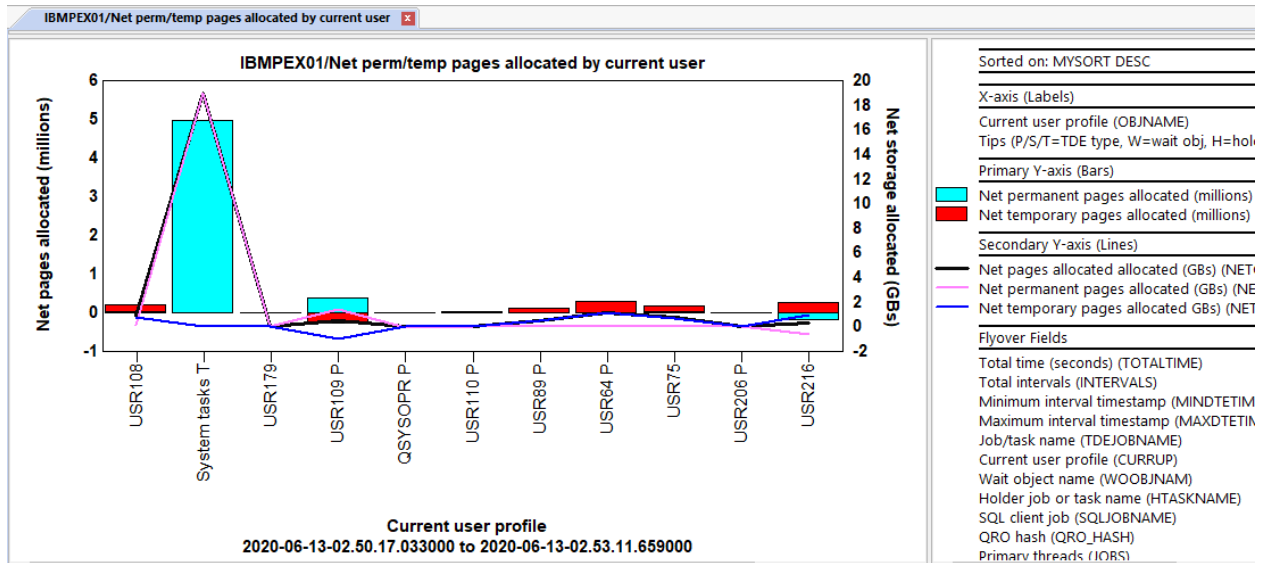
Net pages allocated by generic job

21.16 Net perm/temp pages allocated rankings

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.

Note: This graph requires the Collection Summary analysis is ran.

An example follows:

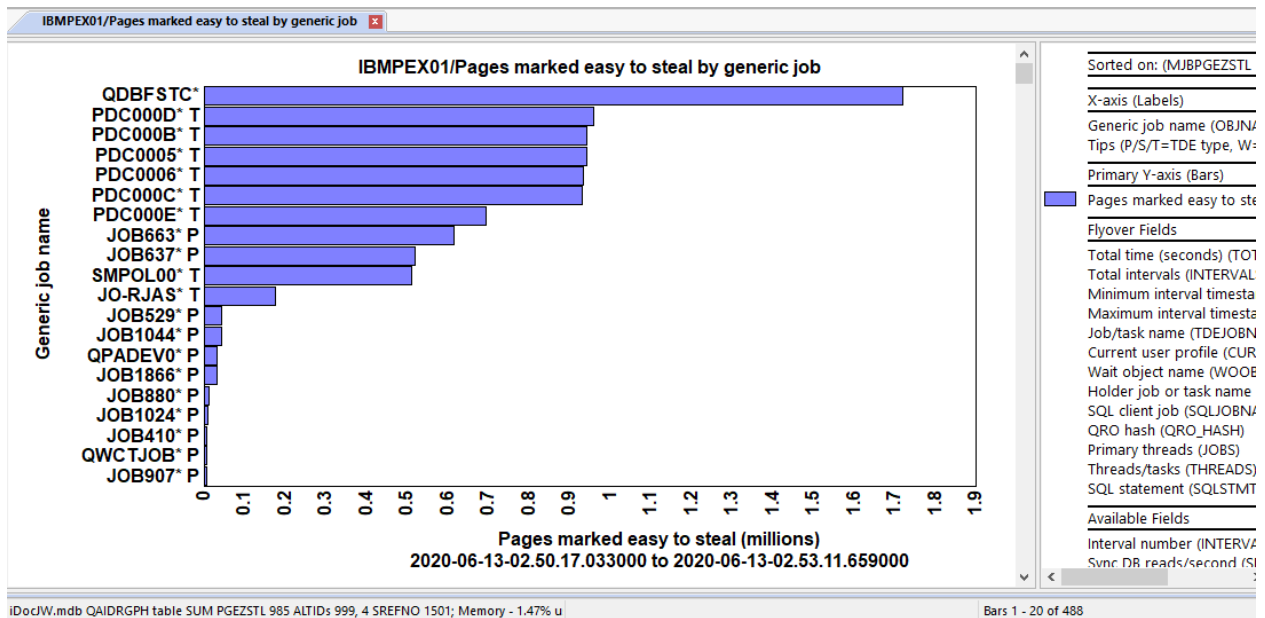


Net perm/temp pages allocated by current user

21.17 Pages marked easy to steal rankings

This graph displays the total pages marked easy to steal (in millions) grouped in various possible ways.

An example follows:

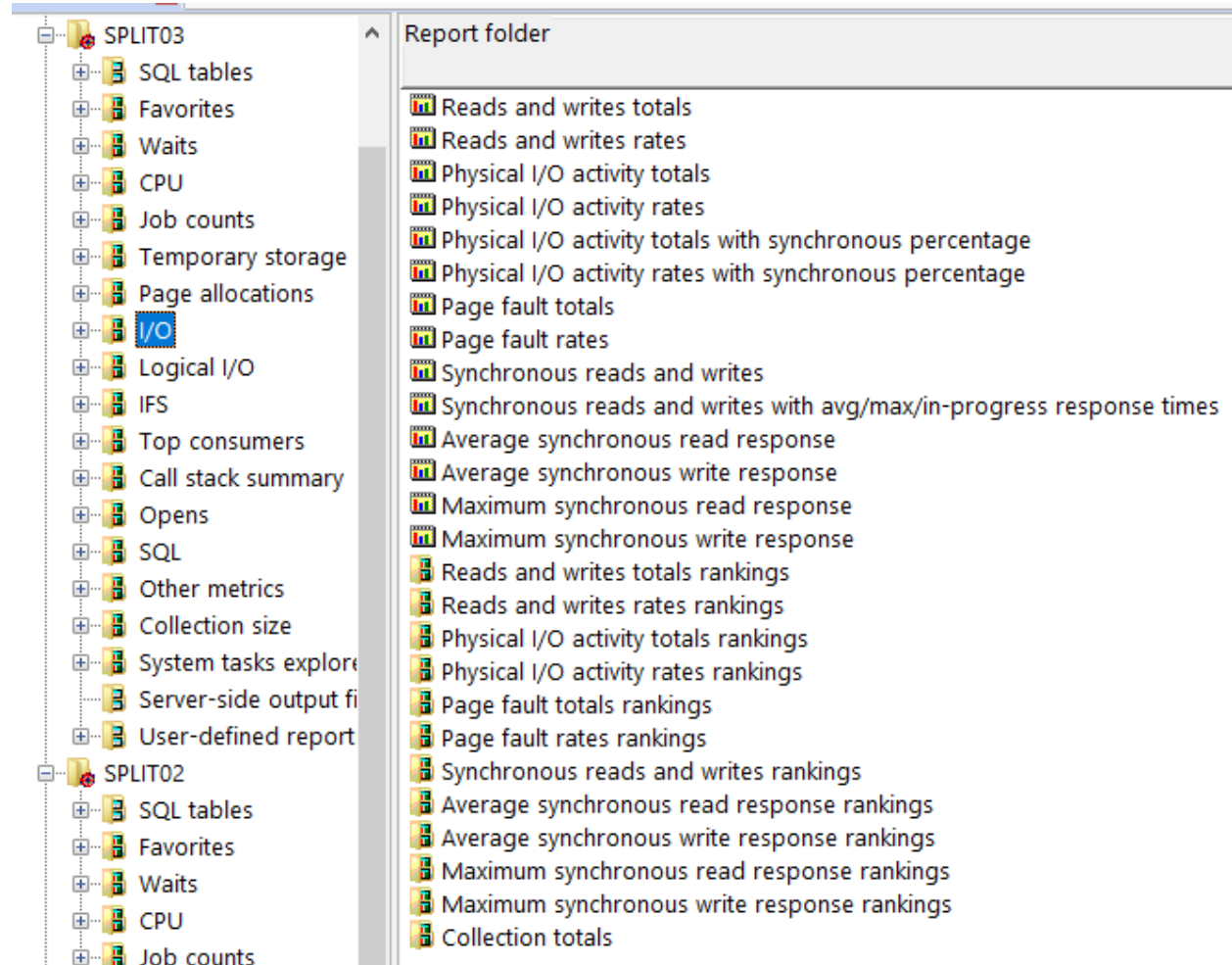


Pages marked easy to steal by generic job

22 I/O

This folder contains graphs related to disk reads/writes, page faults and synchronous or asynchronous physical disk I/Os. Both overview graphs and ranking graphs (by job) are provided.

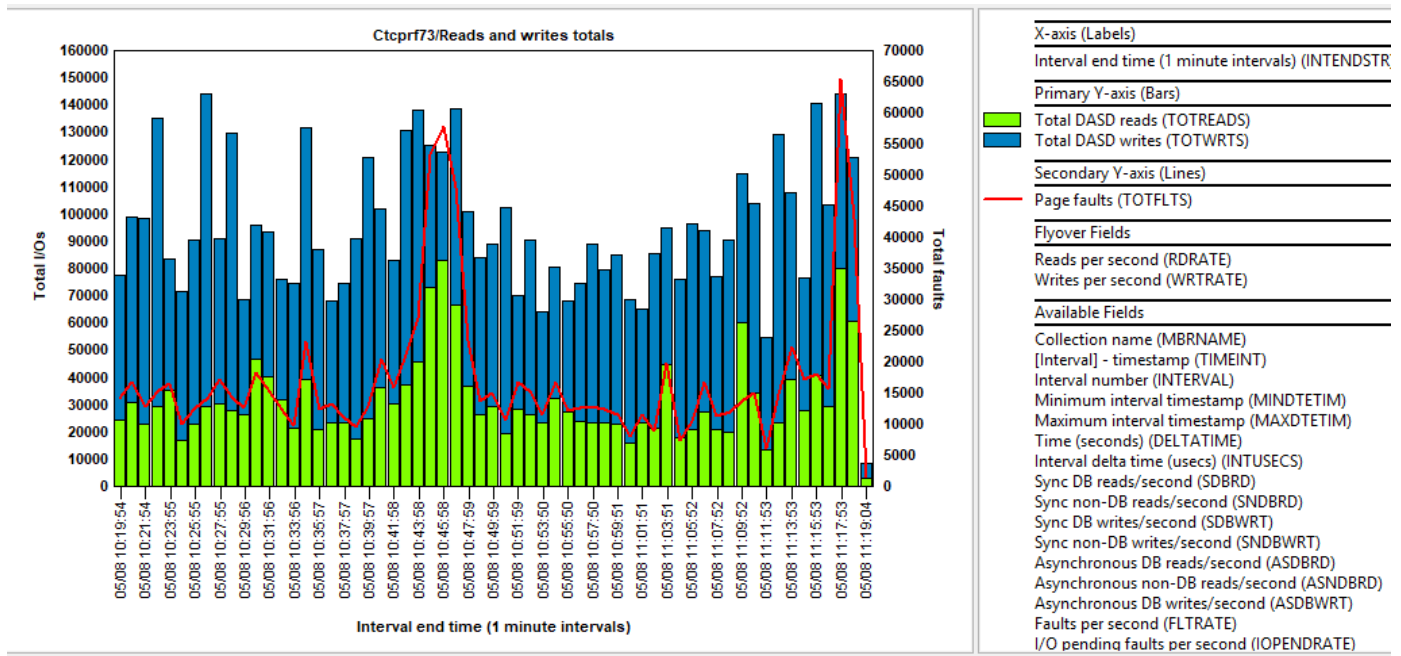
Note: Some graphs require the **Collection Summary** analysis is ran.



I/O folder

22.1 Read and writes totals

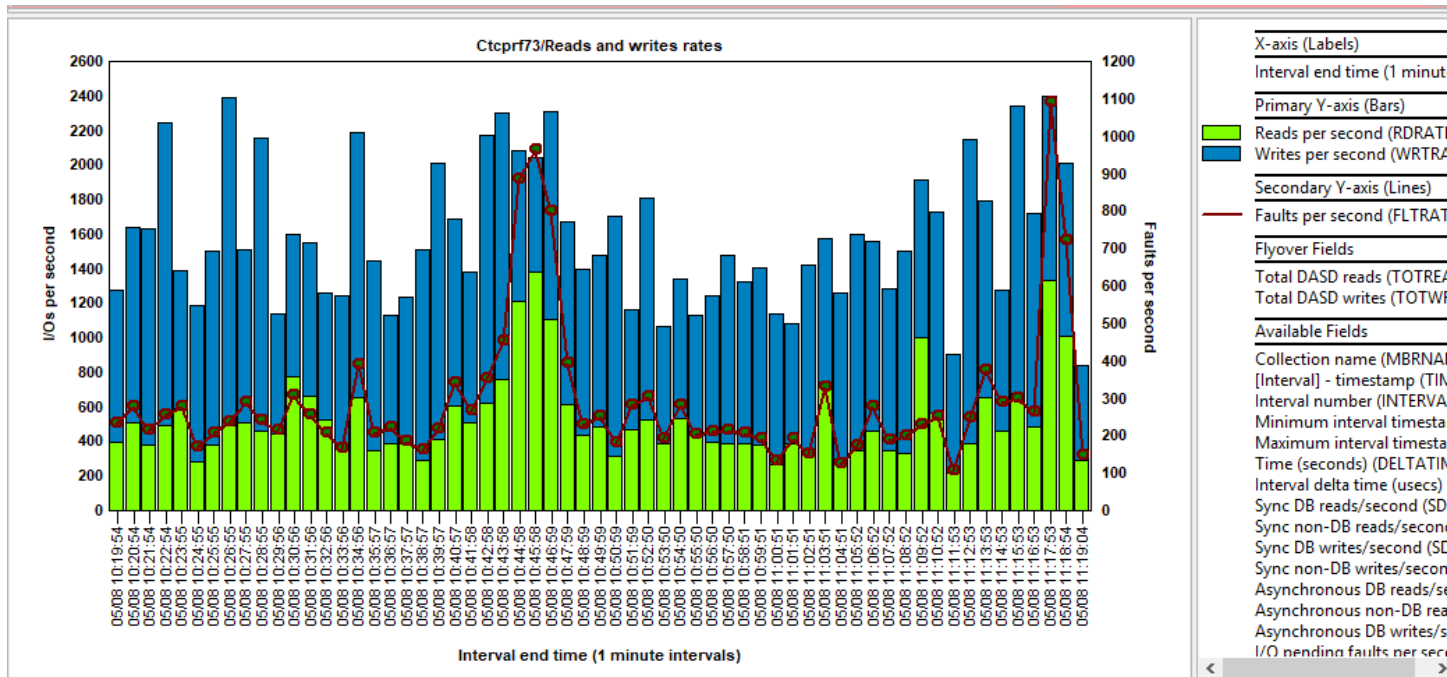
This graph displays the total disk reads and writes for all jobs captured in the collection. The Y2-axis displays the total page faults.



Reads and writes totals

22.2 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.

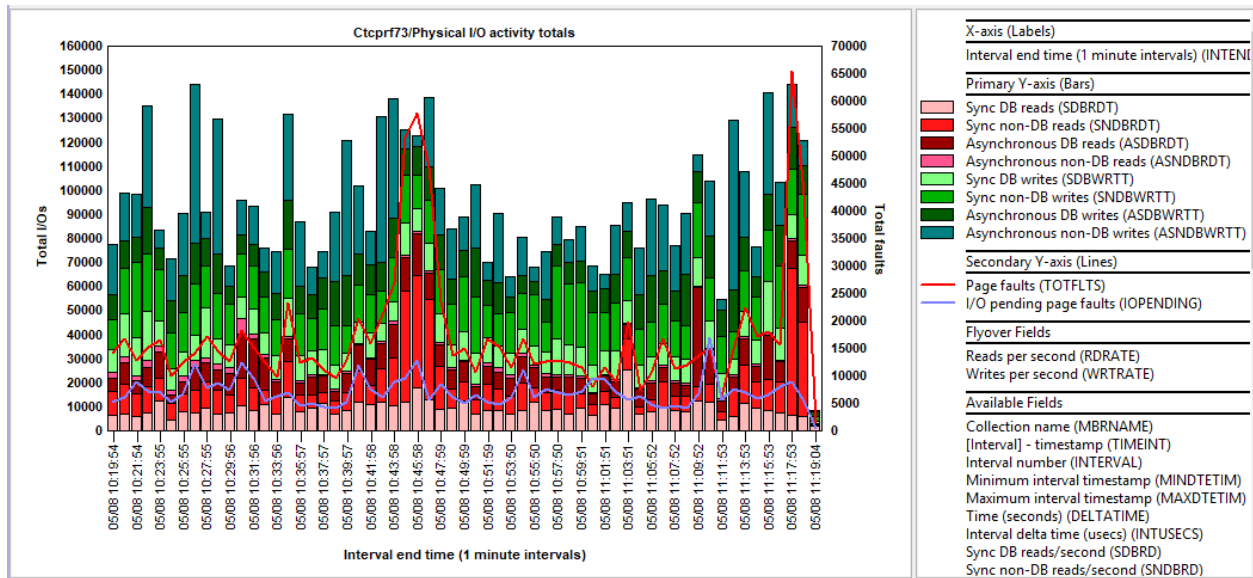


Reads and writes rates

22.3 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.

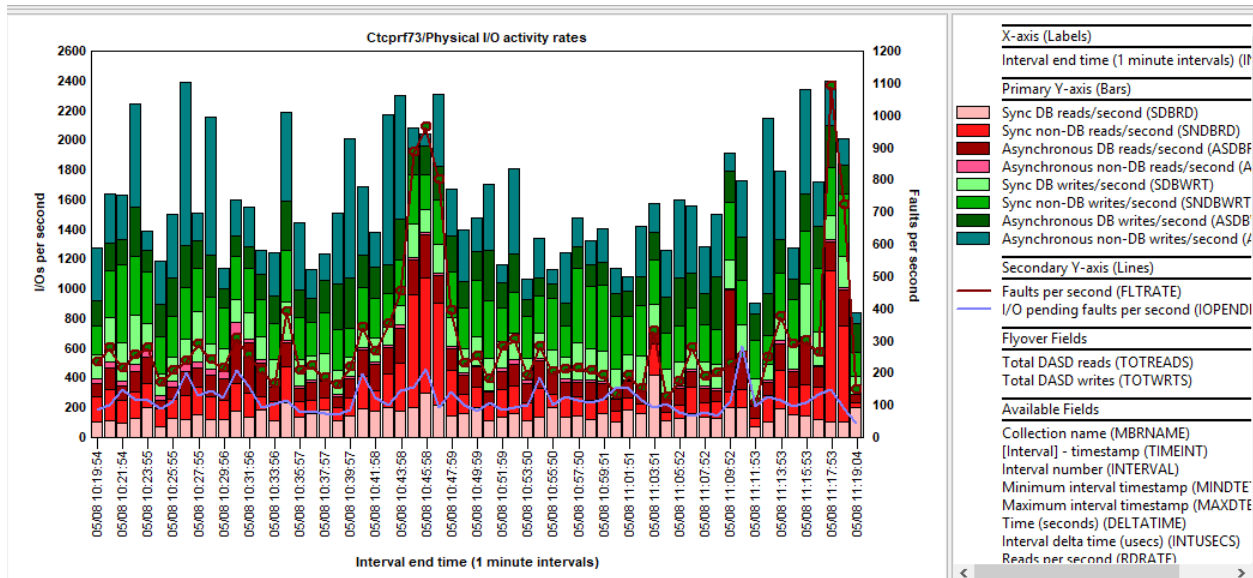


Physical I/O activity totals

22.4 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.

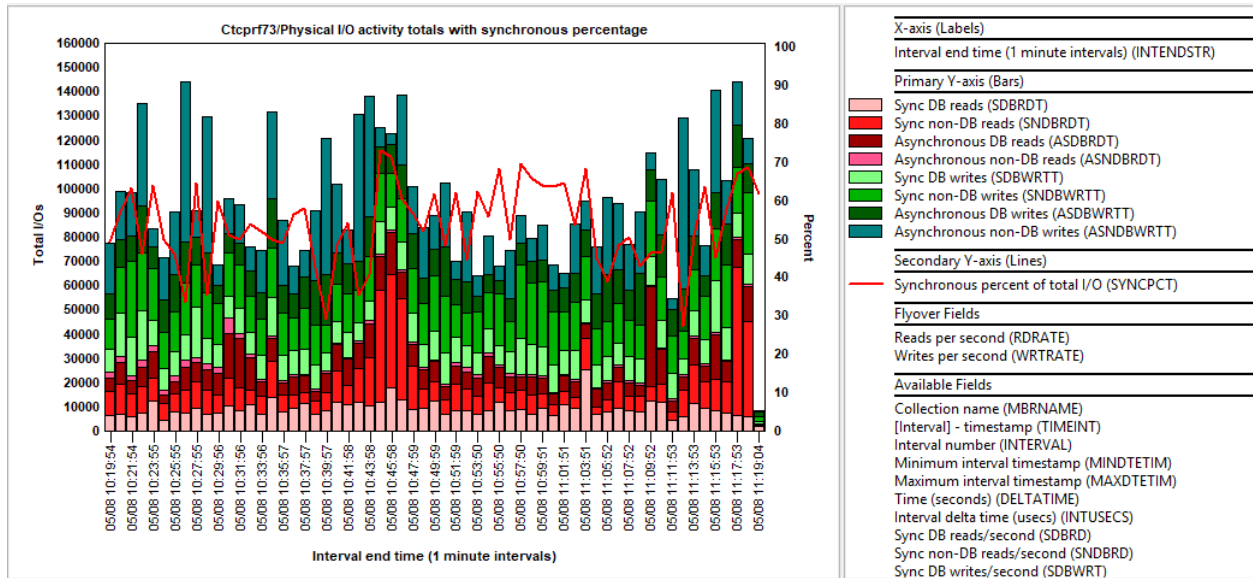


Physical I/O activity rates

22.5 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.

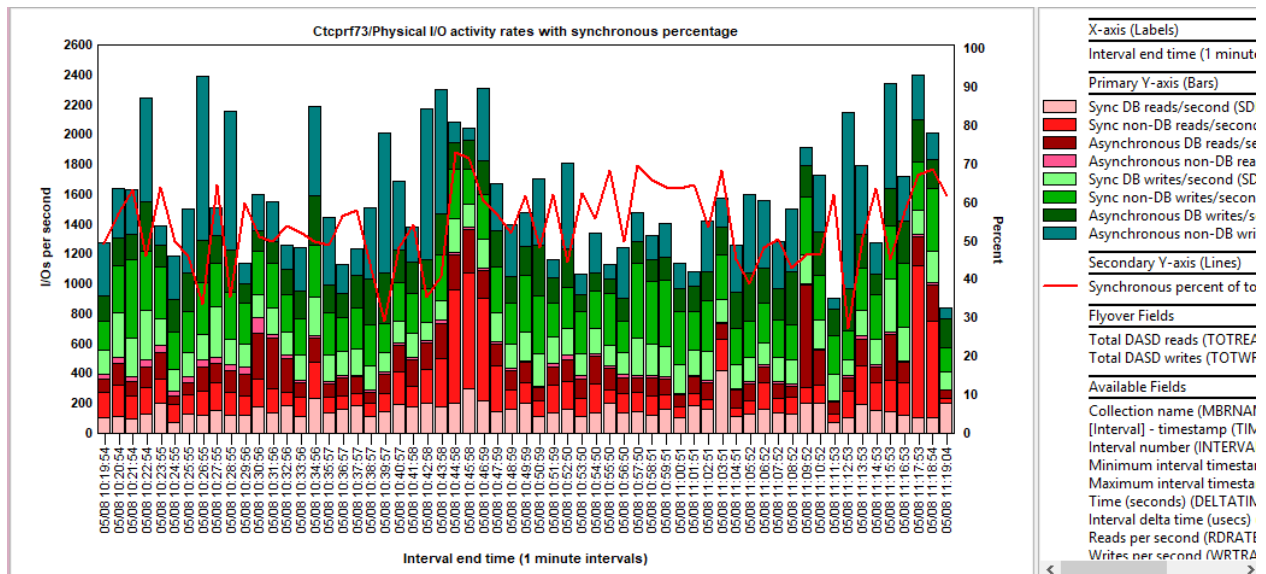


Physical I/O activity totals with synchronous percentage

22.6 Physical I/O activity rates with synchronous percentage

This graph shows physical I/O rates per second for jobs added together per time interval.

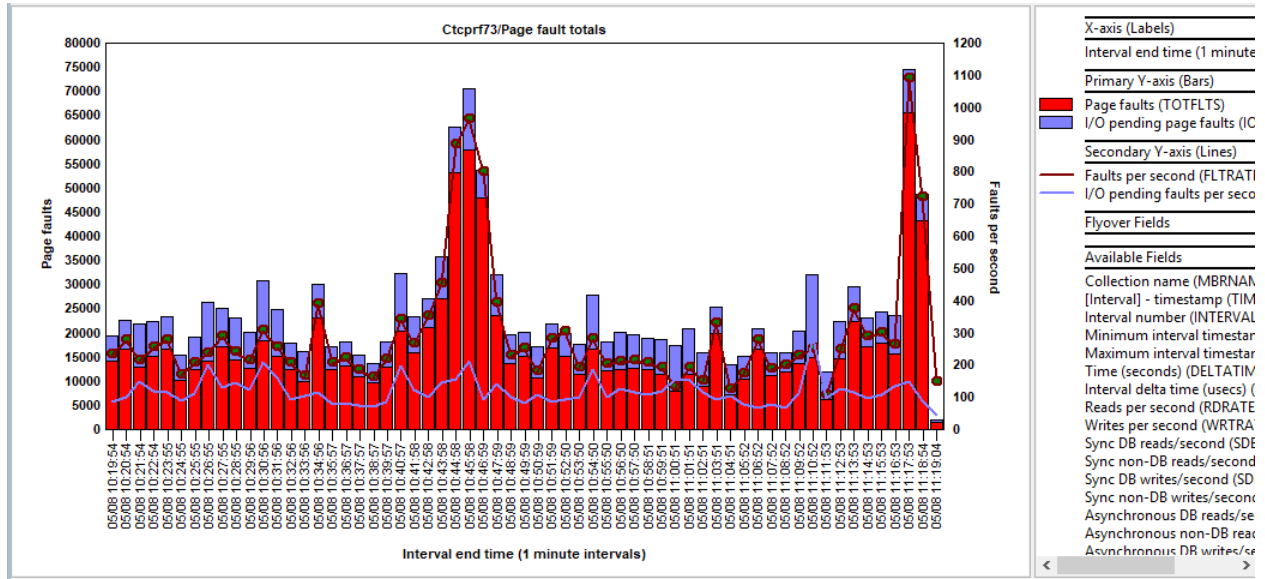
These counters include synchronous or asynchronous, database or non-database reads and writes. The Y2-axis displays the percentage of synchronous I/Os of the total physical disk I/Os.



Physical I/O activity rates with synchronous percentage

22.7 Page fault totals

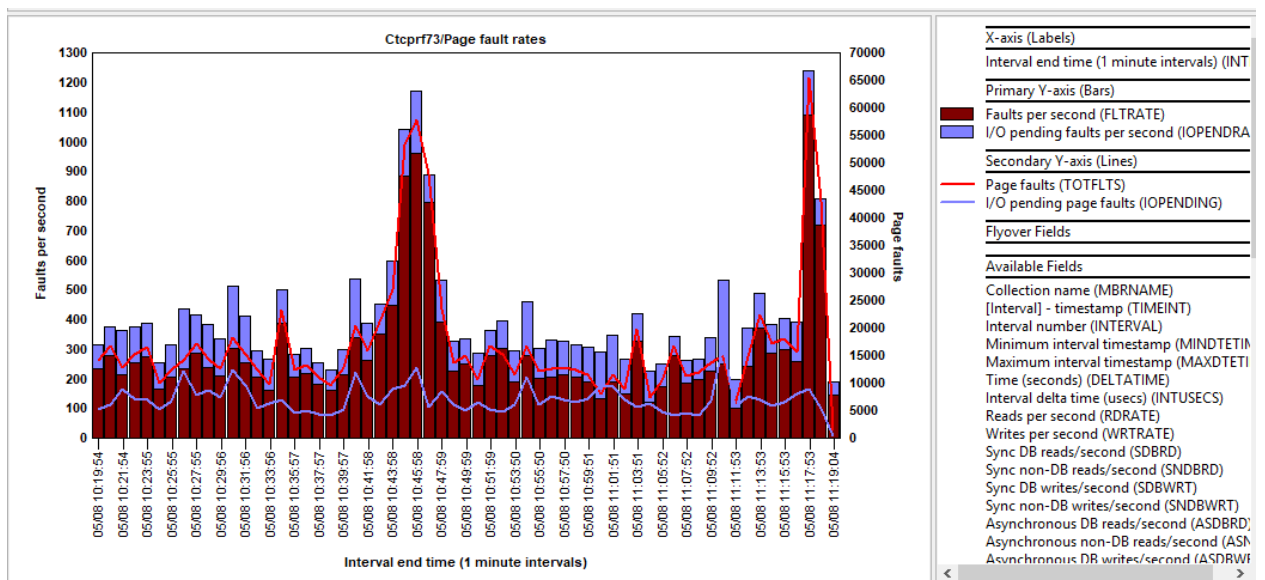
This graph shows total page faults, total I/O pending page faults as well as the rates for each on the secondary Y-axis.



Page fault totals

22.8 Page fault rates

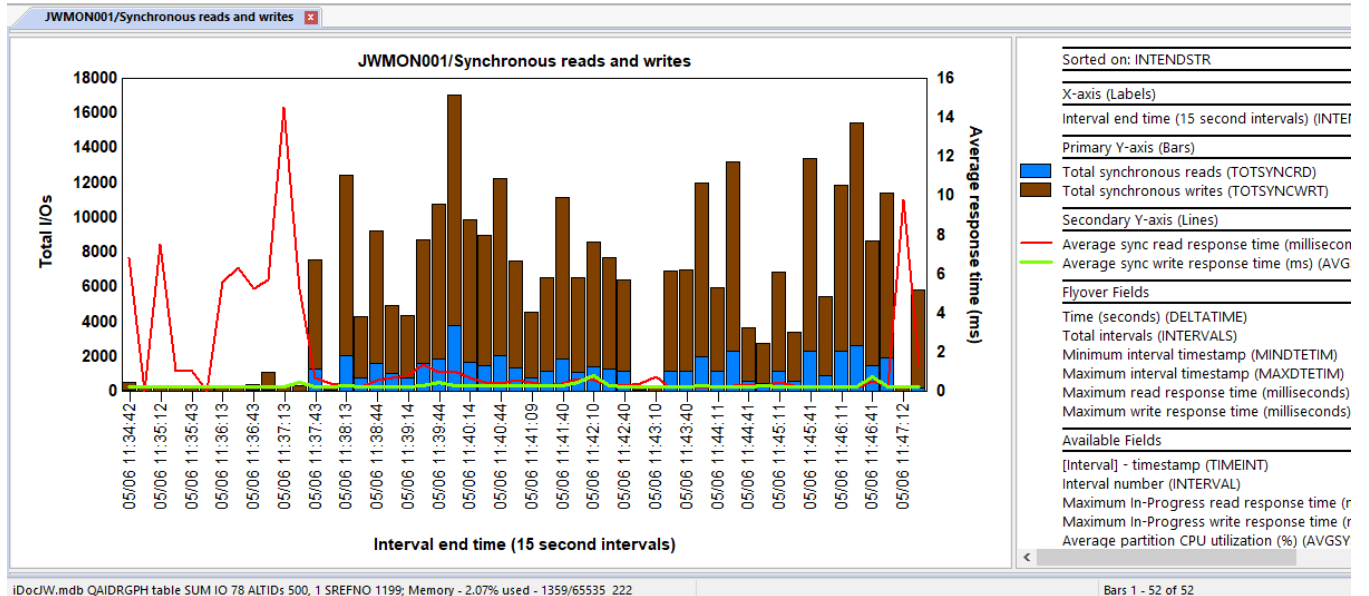
This graph shows page faults per second, I/O pending page faults per second on the primary Y-axis with the totals for each on the secondary Y-axis.



Page fault rates

22.9 Synchronous reads and writes

This graph shows the total number of synchronous reads and writes as well as the average response times.

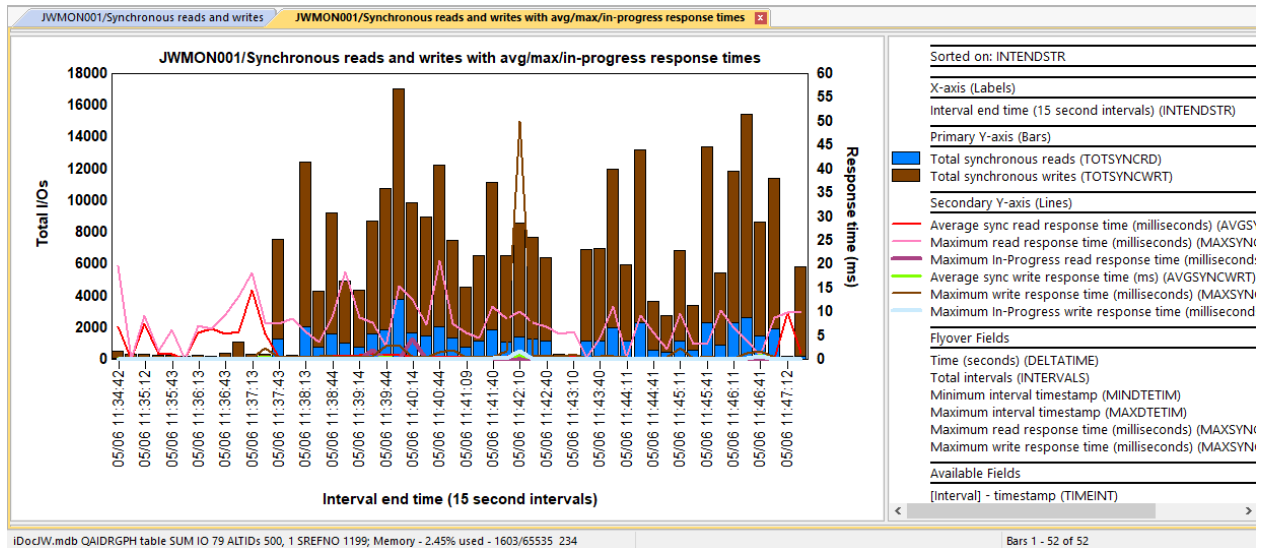


Synchronous reads and writes

22.10 Synchronous reads and writes with avg/max/in-progress response times

This graph is the same as the previous one but adds 4 additional Y2-axis lines intended for advanced users.

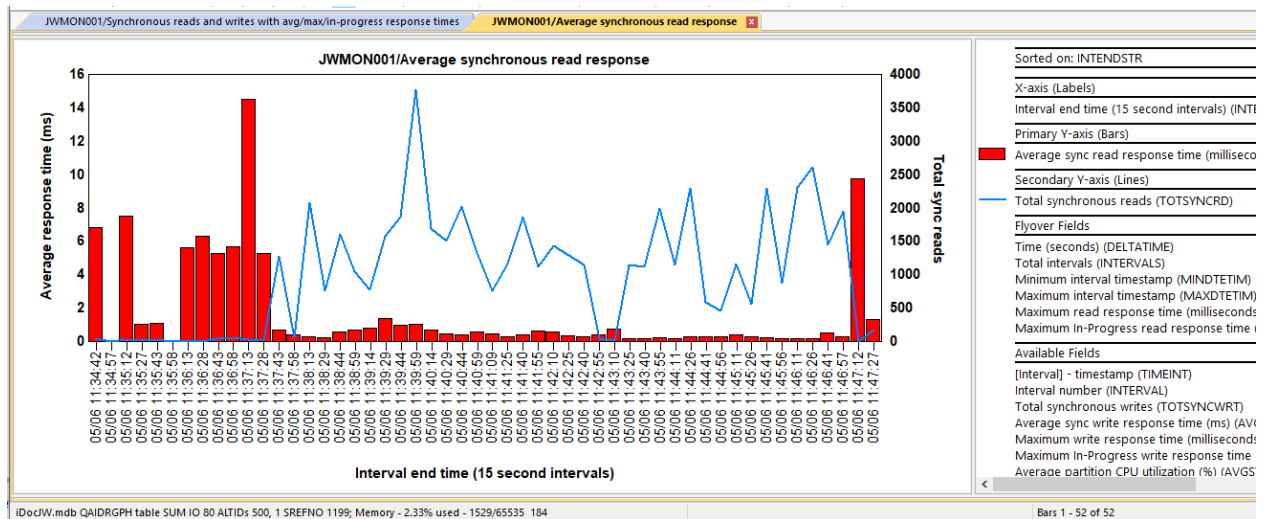
- 1) Maximum read response time
- 2) Maximum in-progress read response time
- 3) Maximum write response time
- 4) Maximum in-progress write response time



Synchronous reads and writes with avg/max/in-progress response times

22.11 Average synchronous read response

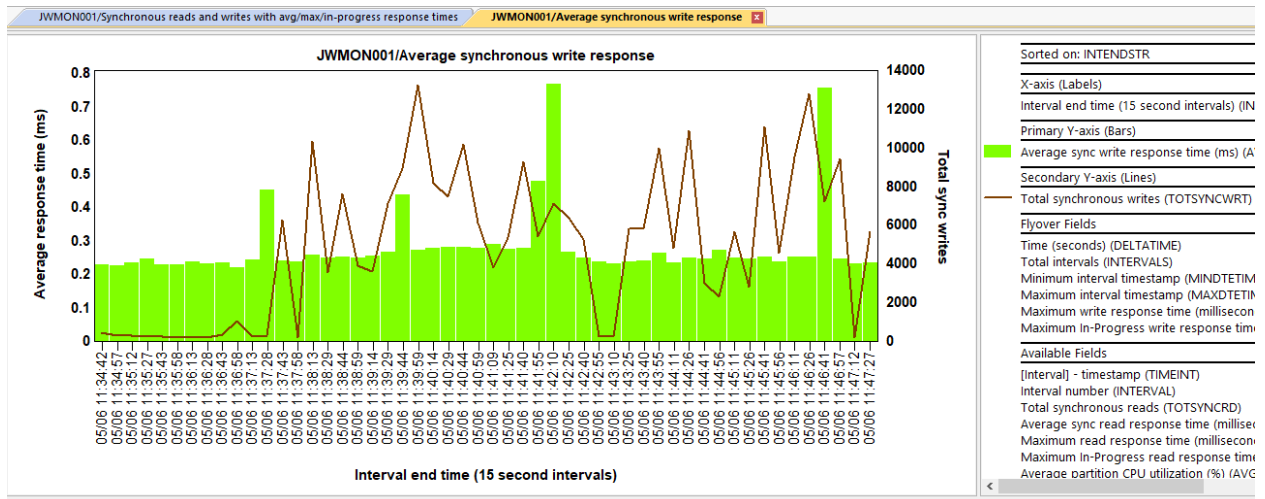
This graph shows the average synchronous read response times along with the total occurrences on the 2nd Y-axis.



Average synchronous read response

22.12 Average synchronous write response

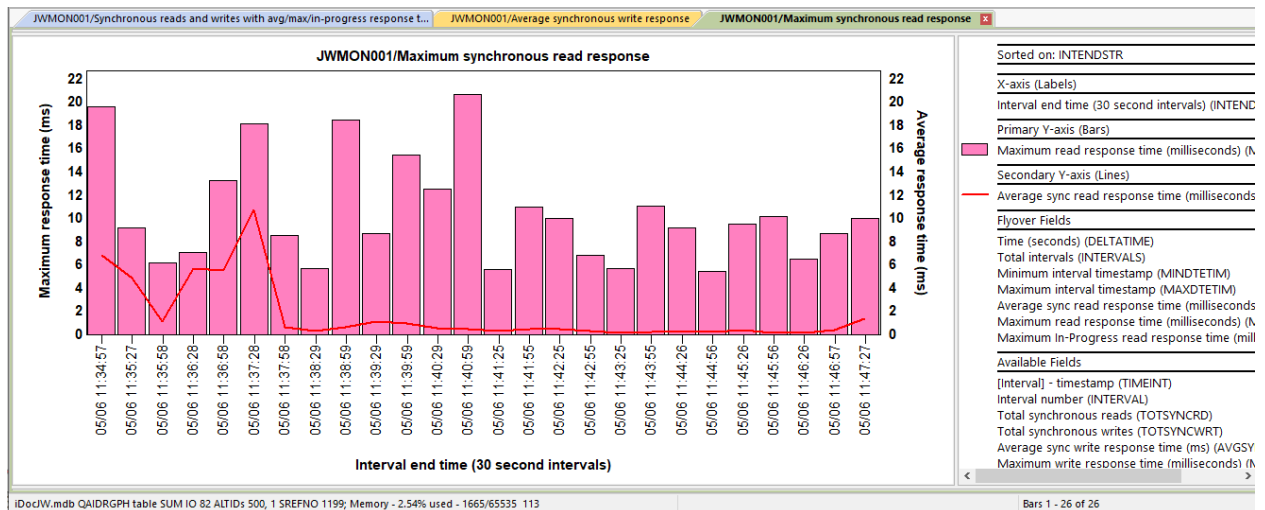
This graph shows the average synchronous write response times along with the total occurrences on the 2nd Y-axis.



Average synchronous write response

22.13 Maximum synchronous read response

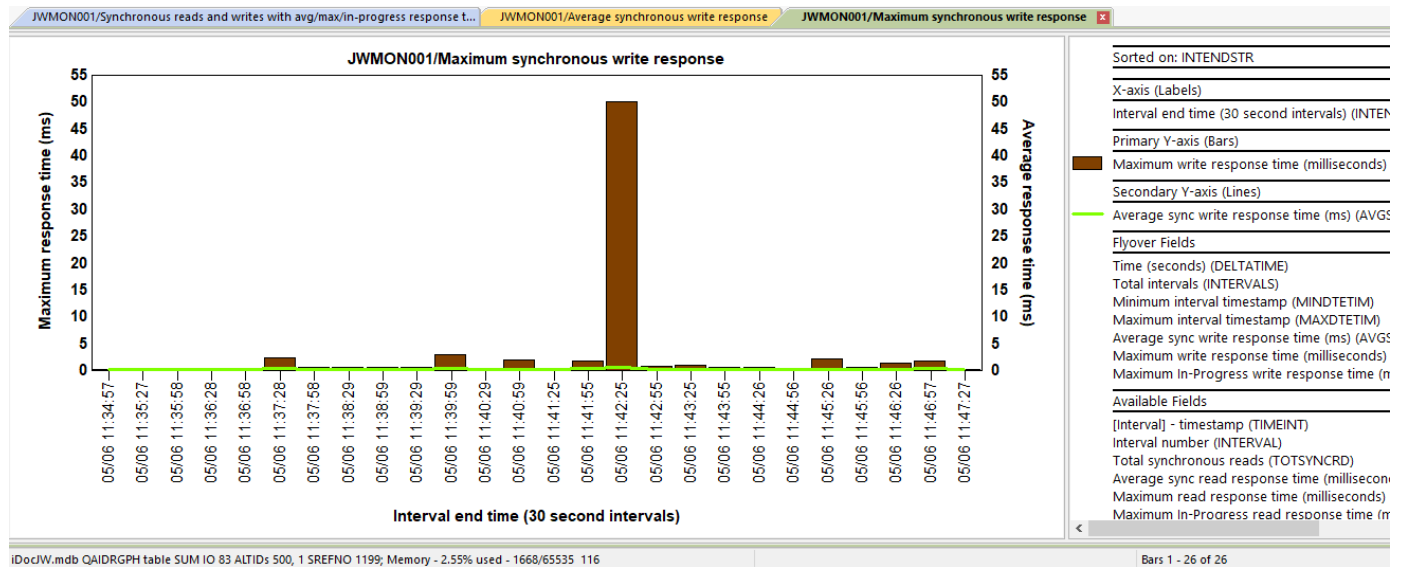
This graph shows the maximum synchronous read response times along with the total occurrences on the 2nd Y-axis.



Maximum synchronous read response

22.14 Maximum synchronous write response

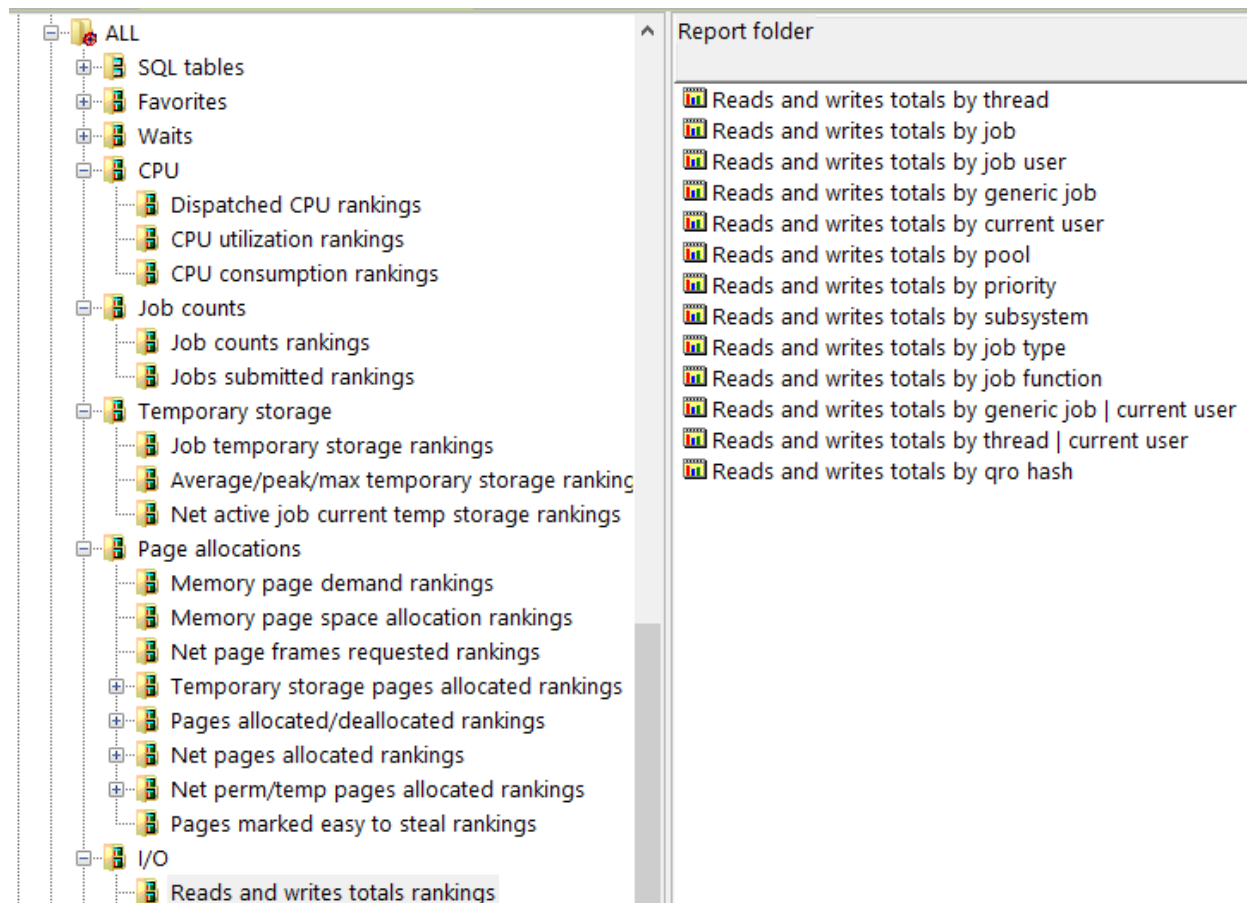
This graph shows the maximum synchronous write response times along with the total occurrences on the 2nd Y-axis.



Maximum synchronous write response

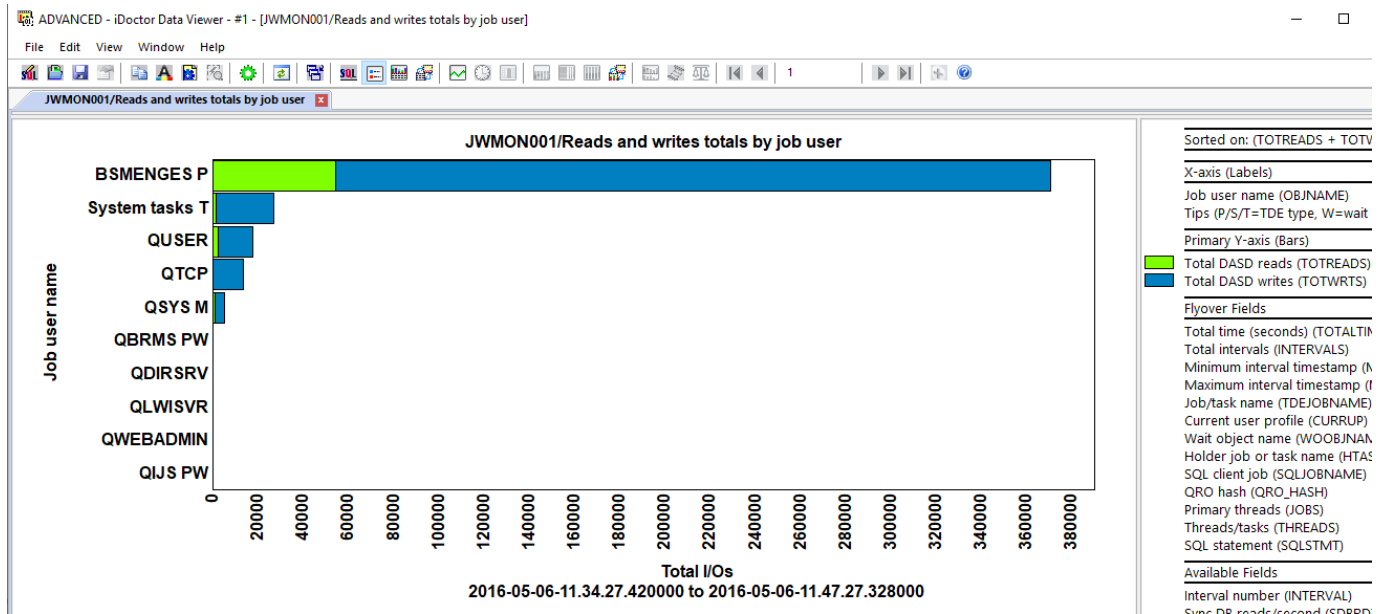
22.15 Reads and writes totals rankings

This folder contains the set of job rankings graphs for the **Reads and writes totals** graph.



I/O -> Reads and writes totals rankings

An example follows:



Reads and writes totals by job user

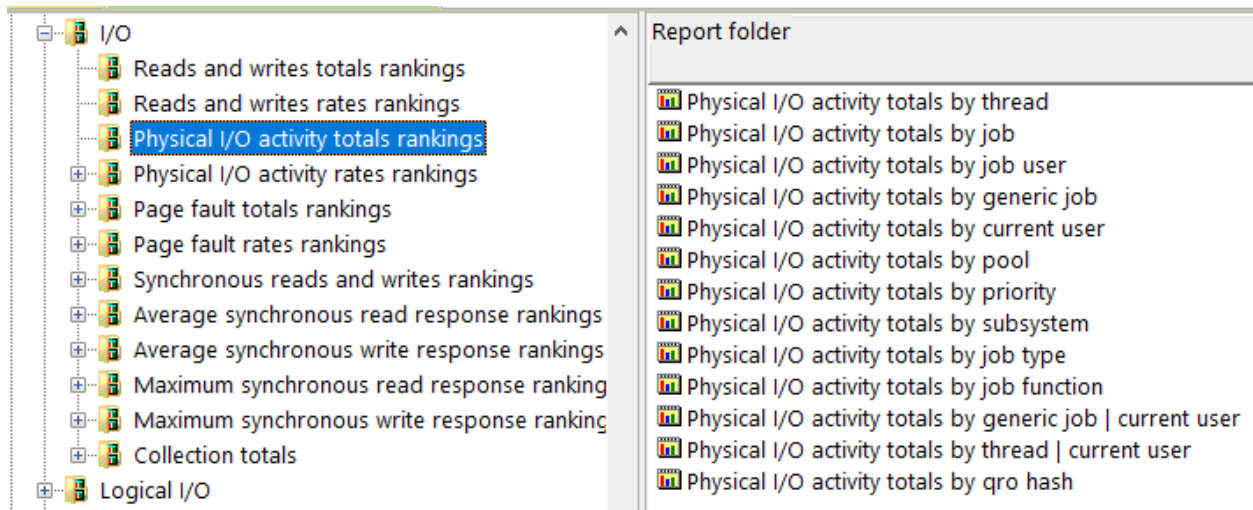
22.16 Reads and writes rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

22.17 Physical I/O activity totals rankings

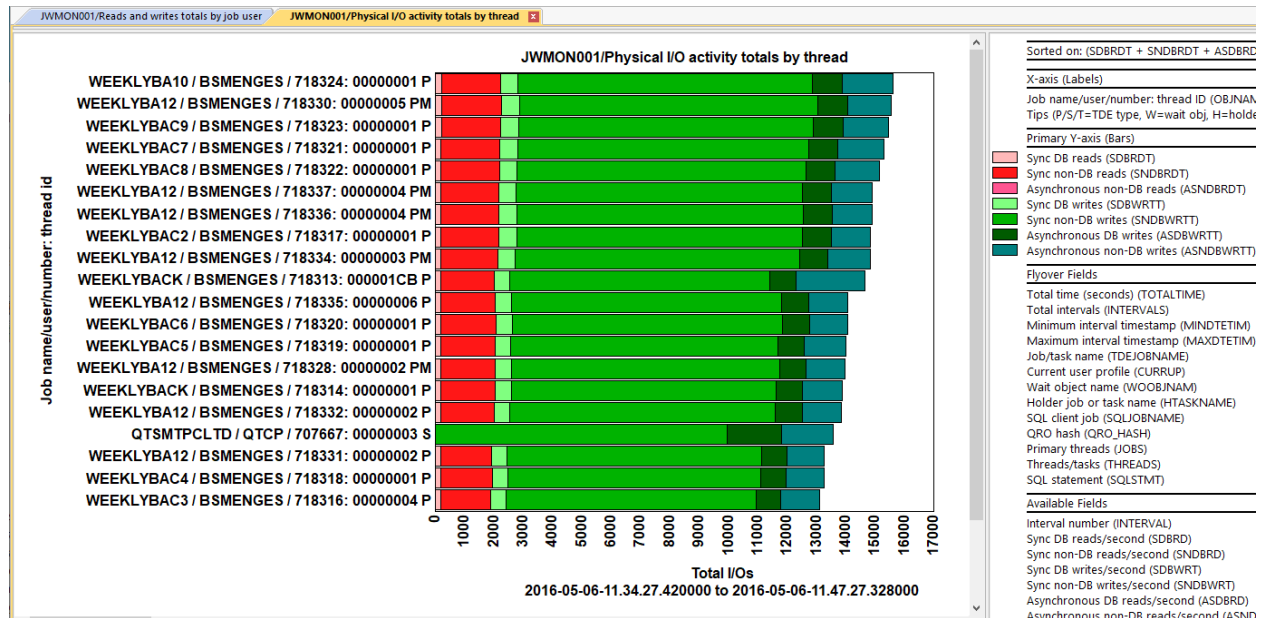
This folder contains the set of job rankings graphs for the **Physical I/O activity totals** graph.

This graph contains synchronous or asynchronous, database or non-database reads and writes.



I/O -> Physical I/O activity totals rankings

An example follows:



Physical I/O activity totals by thread

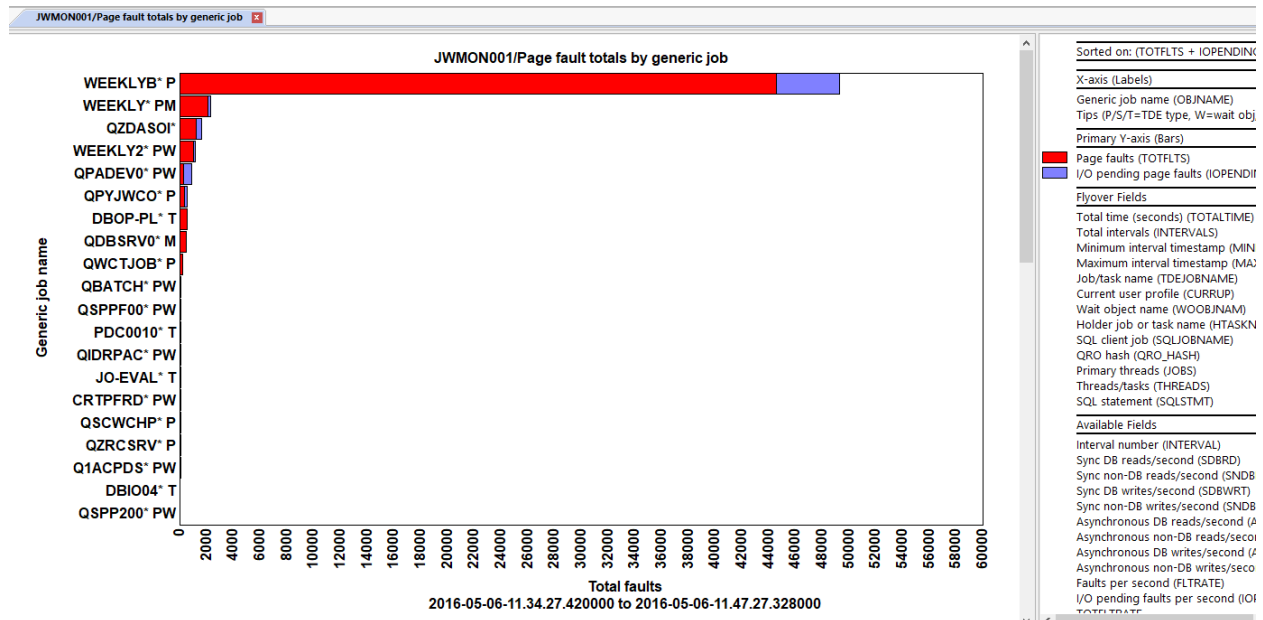
22.18 Physical I/O activity rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

22.19 Page fault totals rankings

This folder contains the set of job rankings graphs for the **Page fault totals** graph. Unlike the overview graph, these rankings graphs do not provide a Y2-axis.

An example follows:



Page fault totals by generic job

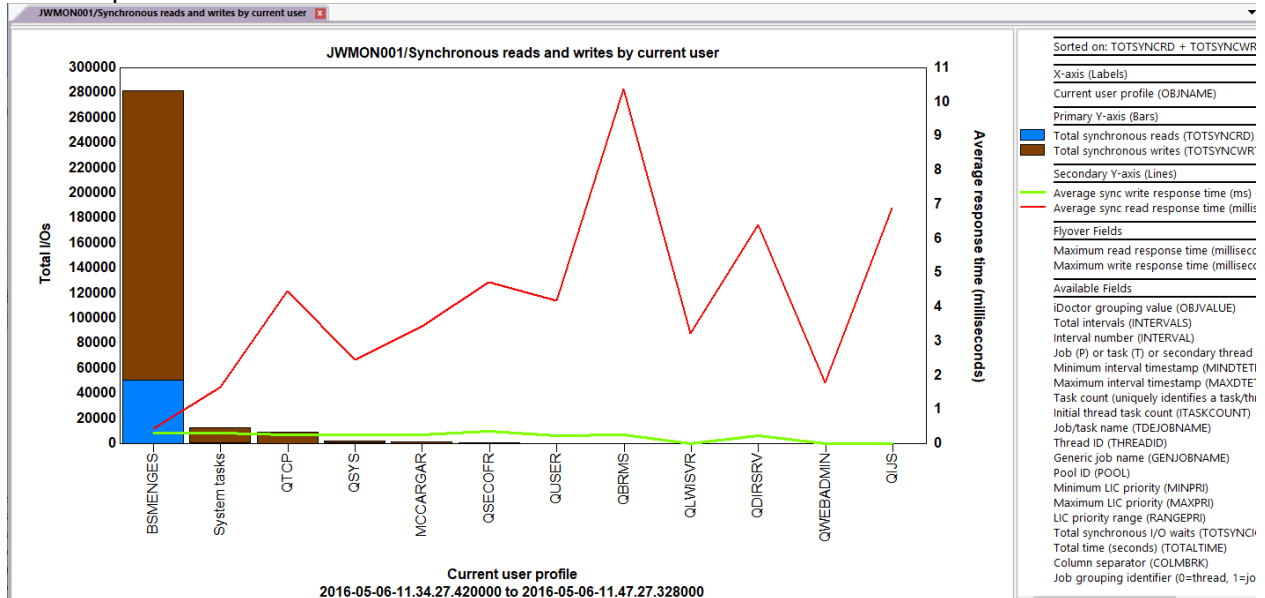
22.20 Page fault rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

22.21 Synchronous reads and writes rankings

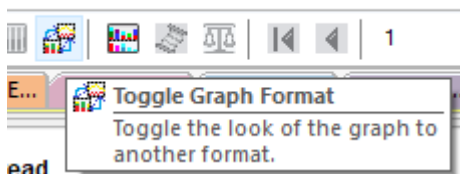
This folder contains the set of job rankings graphs for the **Synchronous reads and writes** graph.

An example follows:

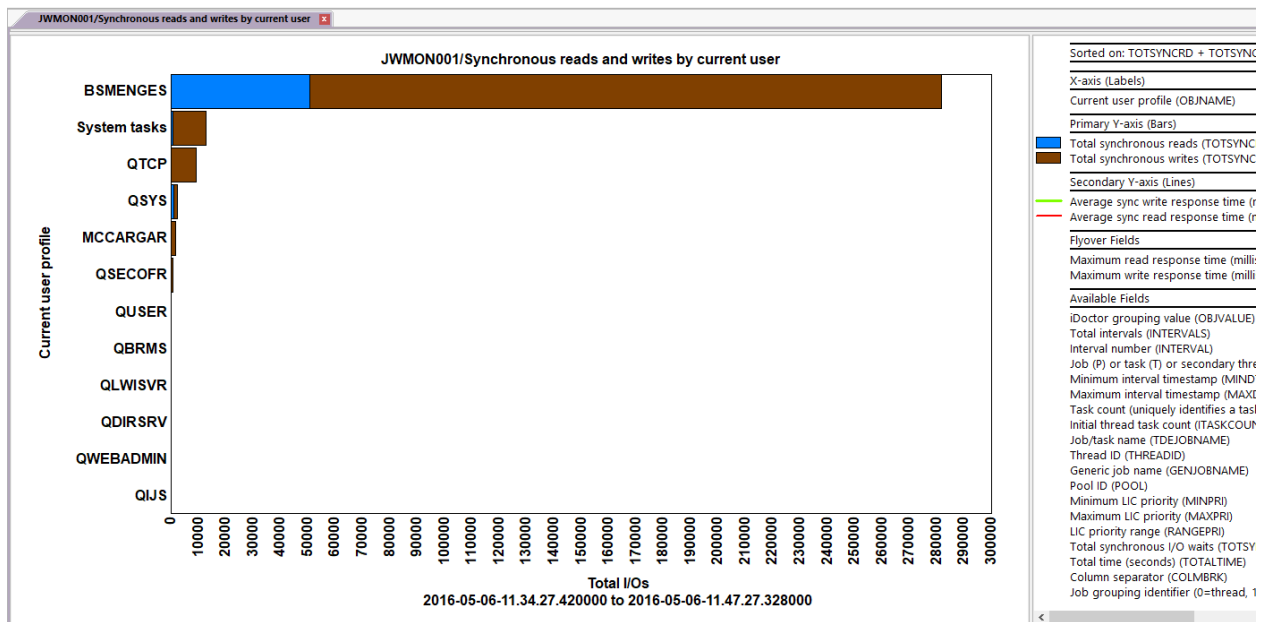


Synchronous reads and writes by current user (vertical bars)

Tip: Press the Toggle Graph Format toolbar button to switch this graph to a horizontal bar graph if desired.



Toggle graph format button

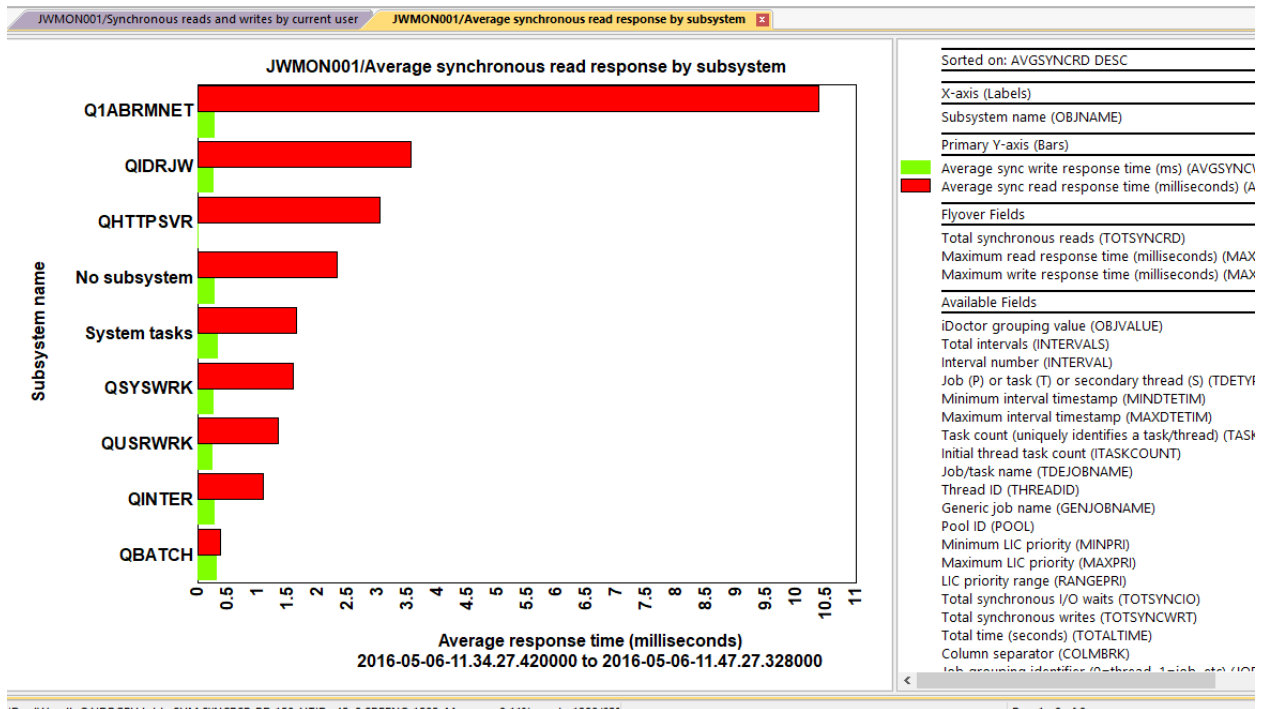


Synchronous reads and writes by current user (horizontal bars)

22.22 Average synchronous read response rankings

This folder contains job groupings ranked by average sync read response times.

An example follows:

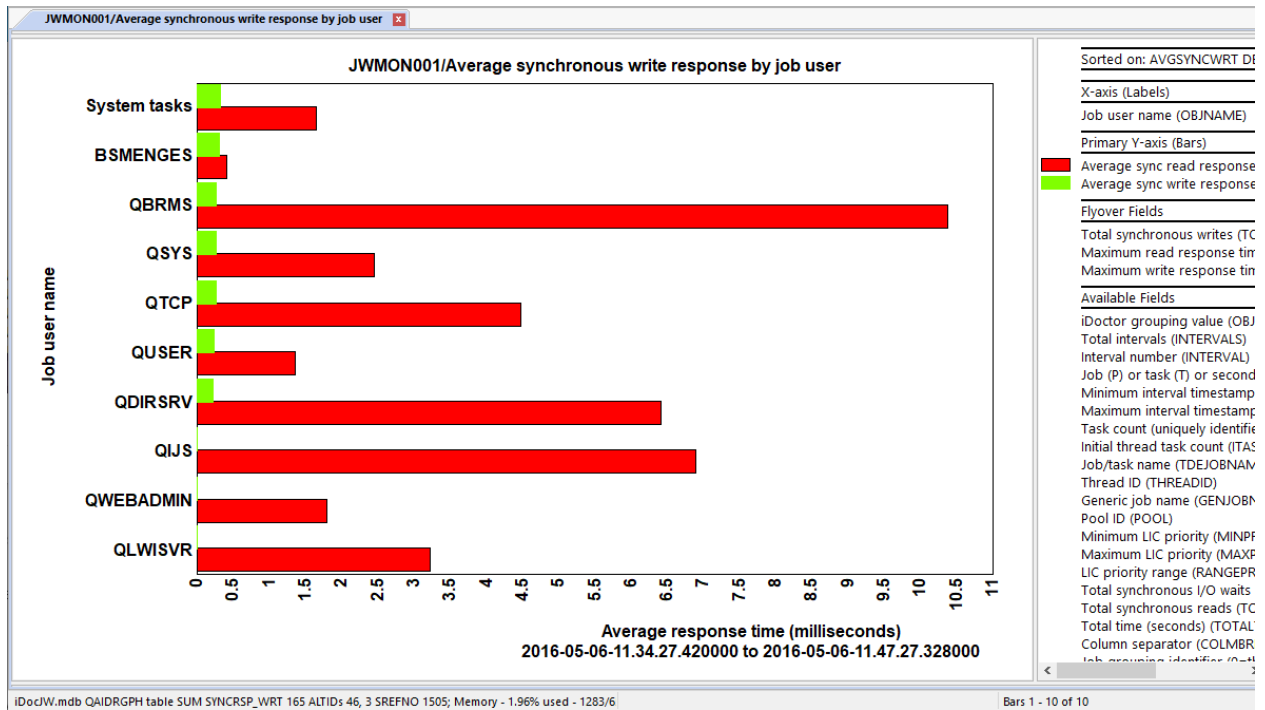


Average synchronous read response by subsystem

22.23 Average synchronous write response rankings

This folder contains job groupings ranked by average sync write response times.

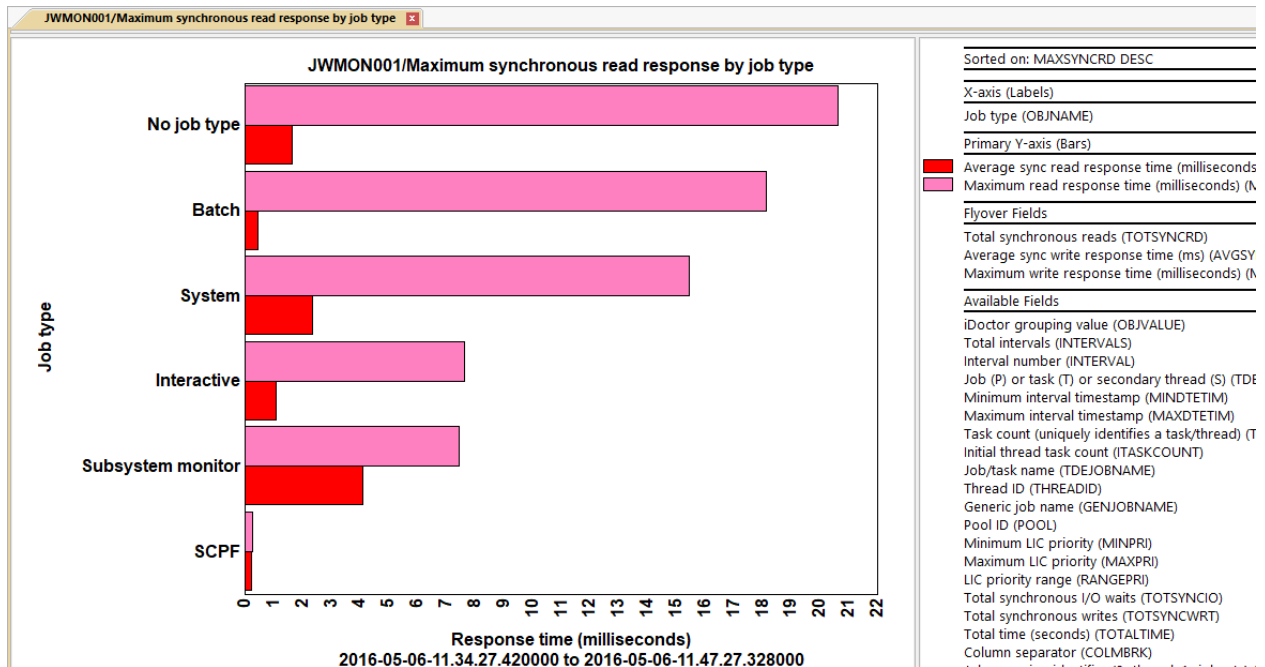
An example follows:



22.24 Maximum synchronous read response rankings

This folder contains job groupings ranked by the maximum sync read response times. The average is also displayed for comparison purposes.

An example follows:

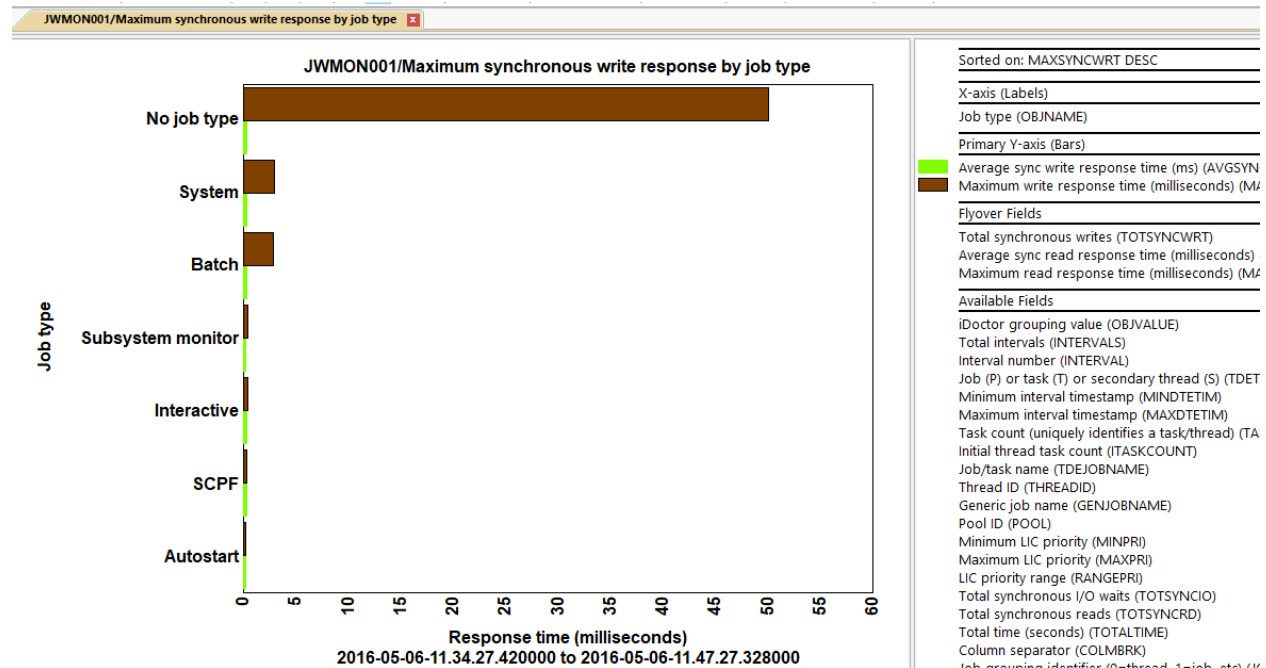


Maximum synchronous read response by job type

22.25 Maximum synchronous write response rankings

This folder contains job groupings ranked by the maximum sync write response times. The average is also displayed for comparison purposes.

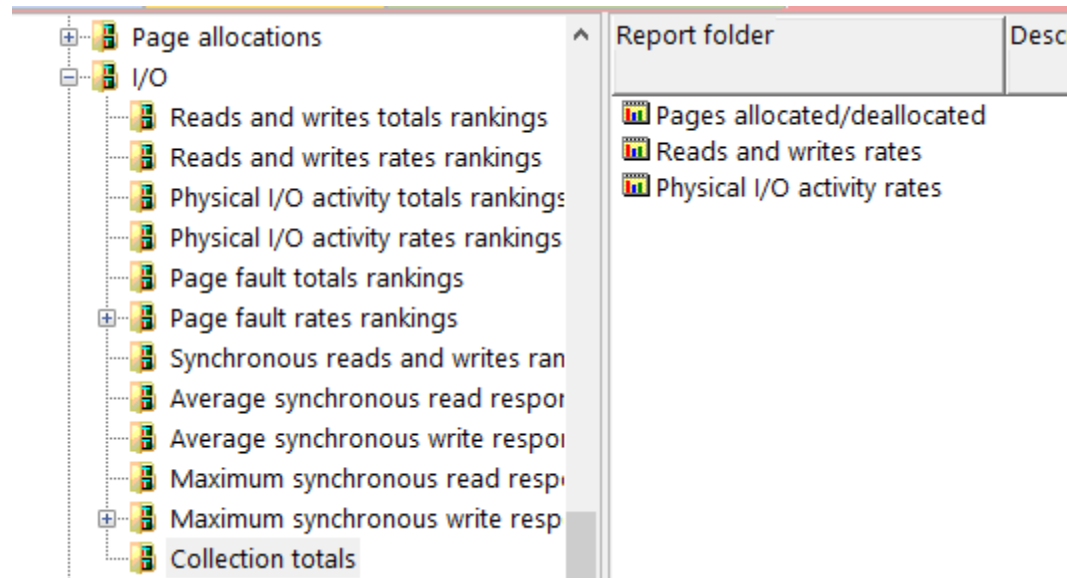
An example follows:



Maximum synchronous write response by job type

22.26 Collection totals

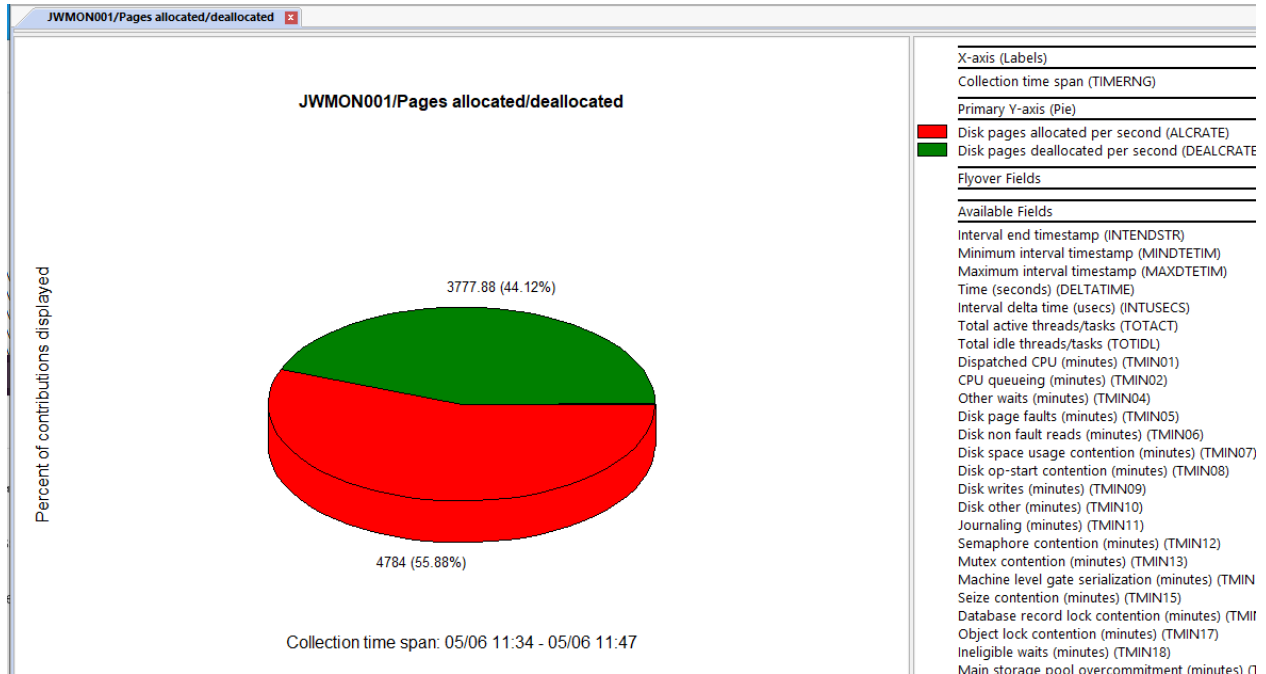
This folder contains a set of pie charts that show high-level statistics about the entire collection related to I/Os and page allocations.



I/O -> Collection totals

22.26.1 Pages allocated/deallocated

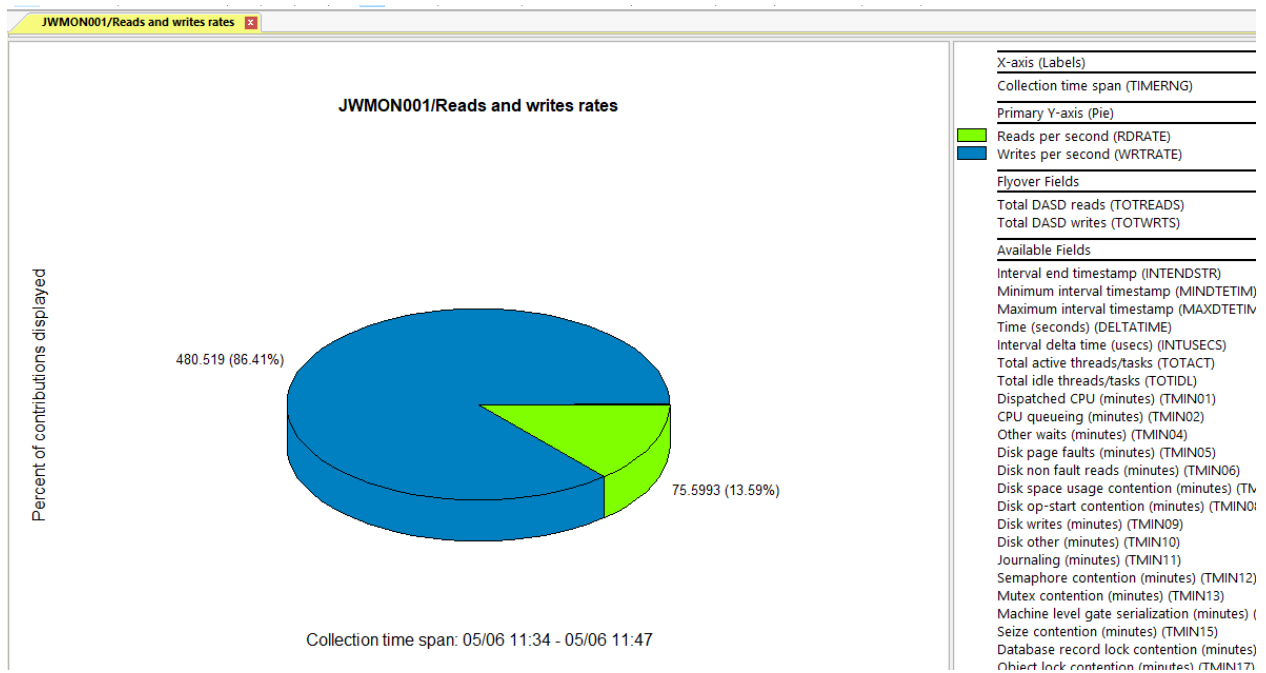
This pie chart simply compares pages allocated vs deallocated in the entire collection.



Pages allocated/deallocated

22.26.2 Reads and writes rates

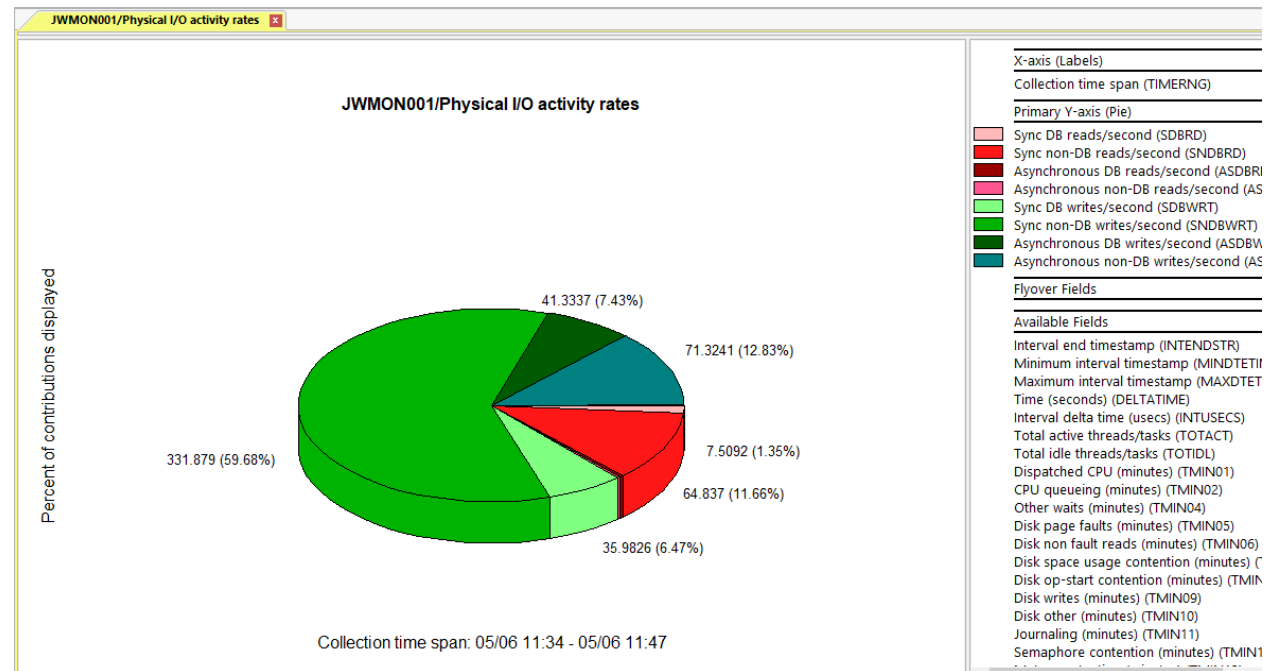
This pie chart compares disk read rates vs write rates for the entire collection.



Reads and writes rates

22.26.3 Physical I/O activity rates

This pie chart compares physical disk I/O metric rates for the entire collection.

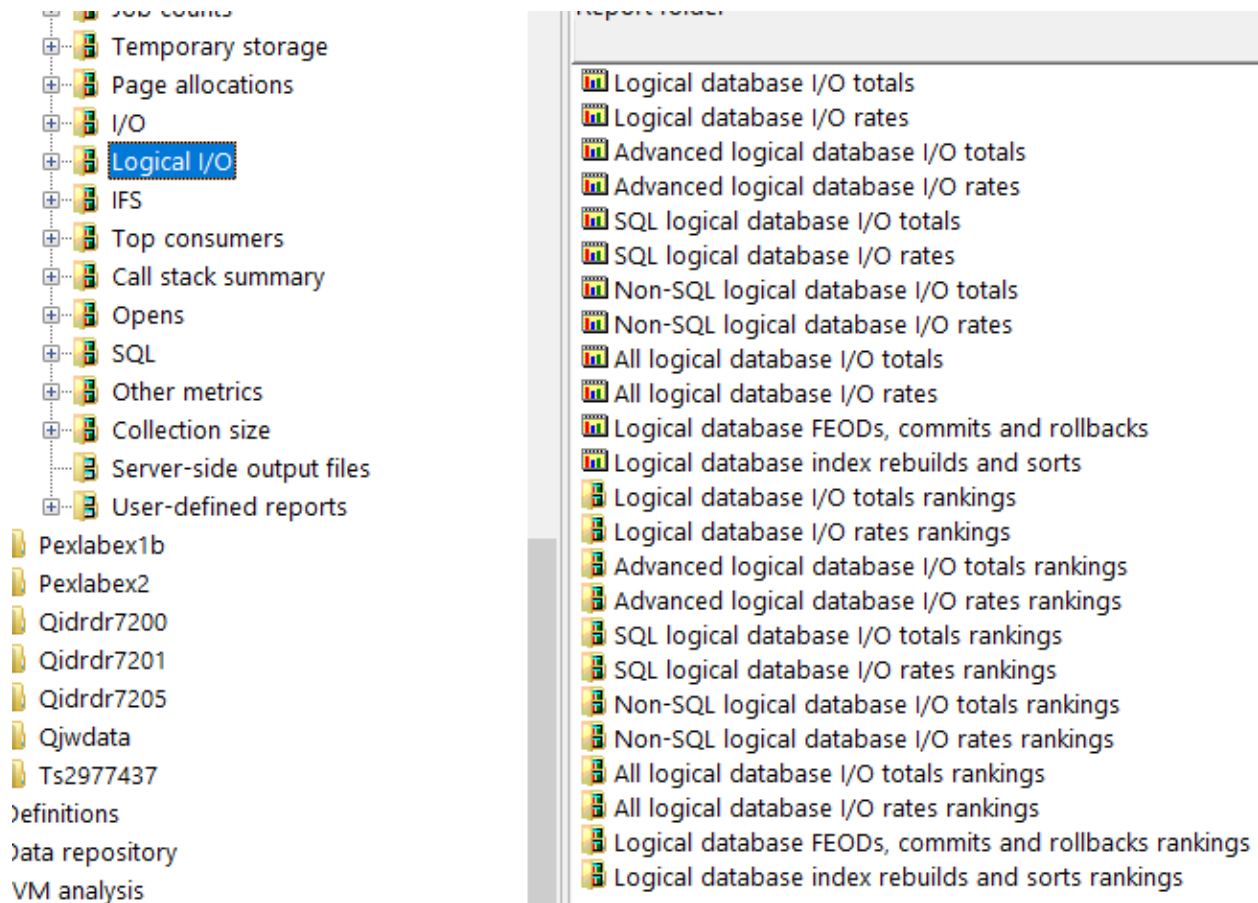


Physical I/O activity rates

23 Logical I/O

This folder contains graphs related to logical database statistics. Both overview graphs and ranking graphs (by job) are provided.

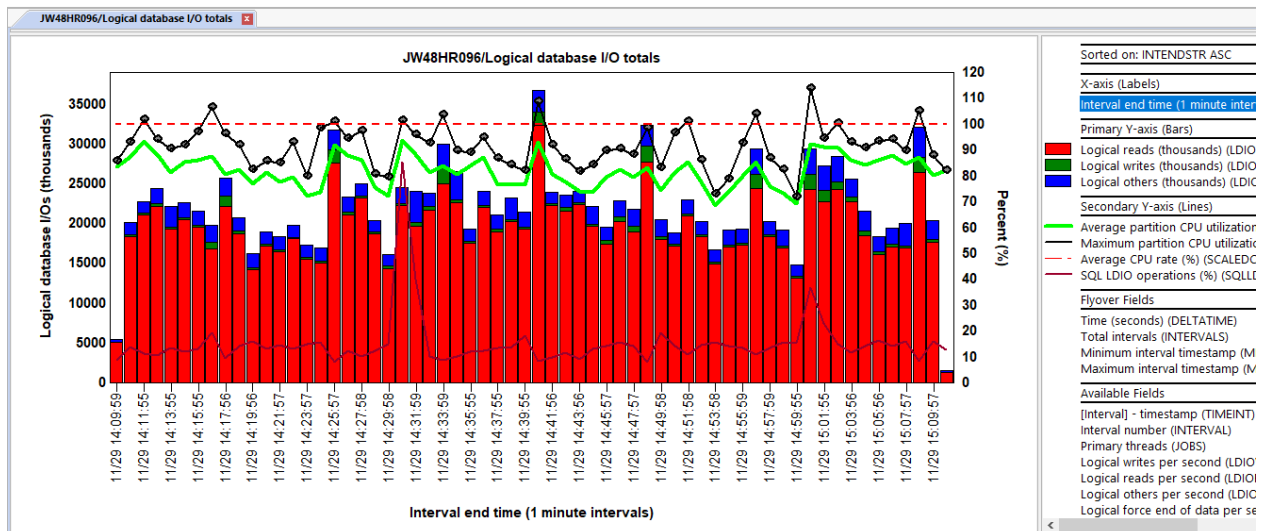
Note: These graphs include logical SQL contributions which were added to the OS at 7.2. As of February 2022, there appears to be accuracy problems with some of these metrics that do not make sense (SQL logical counters can sometimes exceed the logical I/O counters as shown in the below examples).



Logical I/O Folder

23.1 Logical database I/O totals

This graph provides metrics for logical database reads, writes and other (which is updates and deletes combined.)



Logical database I/O totals

23.2 Logical database I/O rates

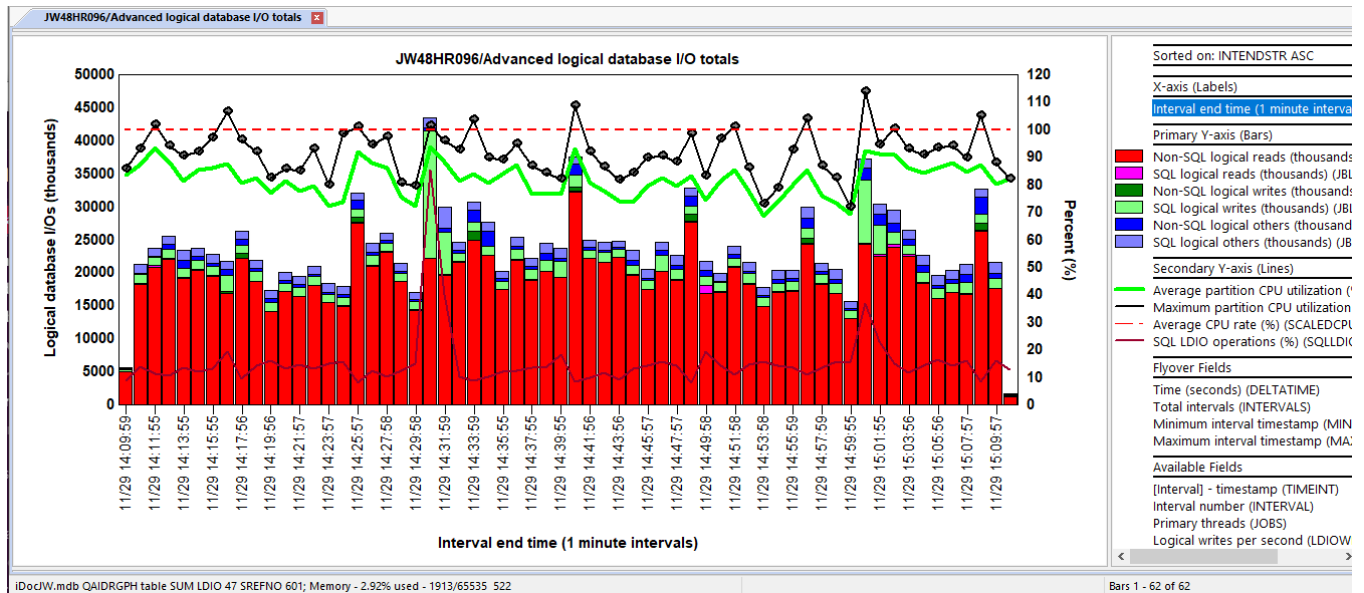
This graph is identical to the previous one except shows the metrics as a rate per second.

23.3 Advanced logical database I/O totals

This graph provides 6 metrics which attempts to show both SQL and non-SQL contributions to logical database I/O operations.

The metrics shown on this graph are:

- 1) Non-SQL logical reads (thousands)
- 2) SQL logical reads (thousands)
- 3) Non-SQL logical writes (thousands)
- 4) SQL logical writes (thousands)
- 5) Non-SQL logical others (thousands)
- 6) SQL logical others (thousands)



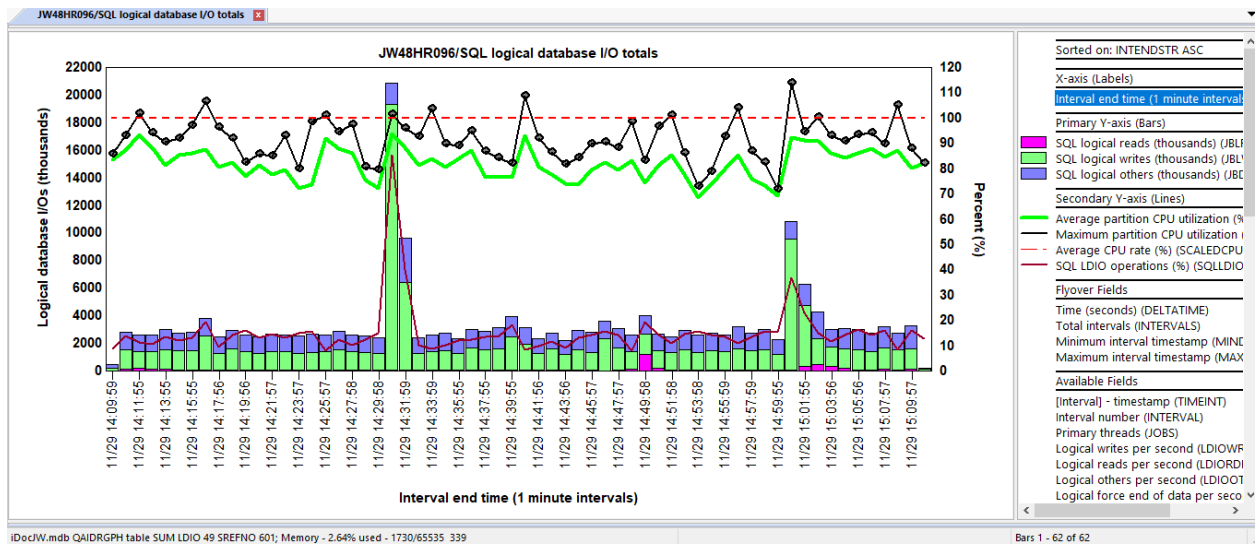
Advanced logical database I/O totals

23.4 Advanced logical database I/O rates

This graph is identical to the previous one except shows the metrics as a rate per second.

23.5 SQL logical database I/O totals

This graph shows the SQL logical reads, writes and others (in thousands.)



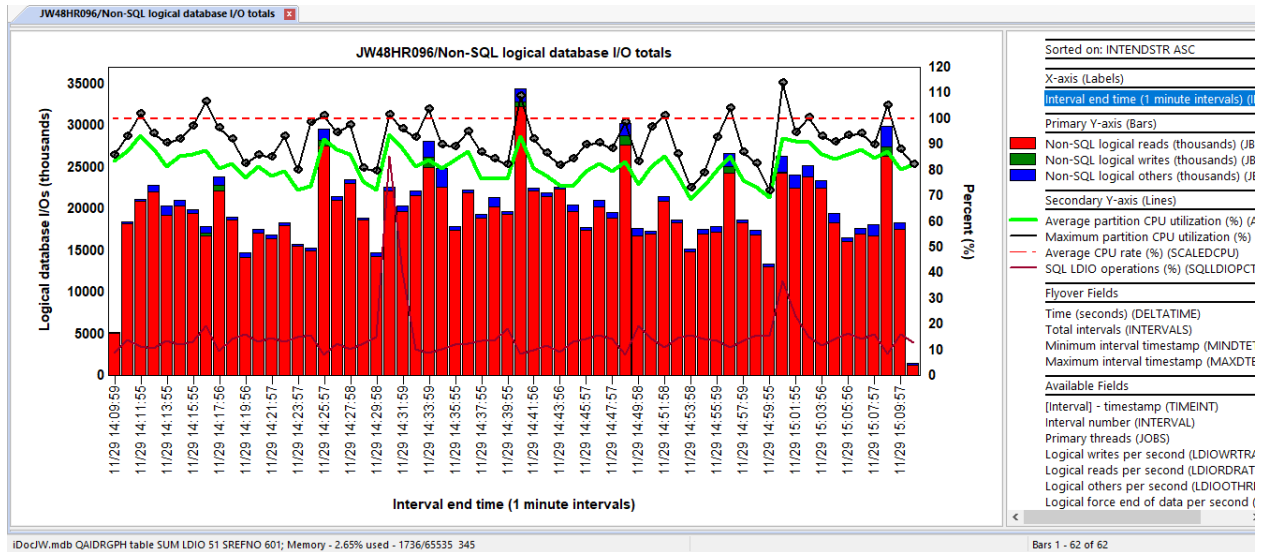
SQL logical database I/O totals

23.6 SQL logical database I/O rates

This graph is identical to the previous one except shows the metrics as a rate per second.

23.7 Non-SQL logical database I/O totals

This graph subtracts the SQL logical counts from the logical counts to provide the non-SQL logical I/O metrics.



Non-SQL logical database I/O totals

23.8 Non-SQL logical database I/O rates

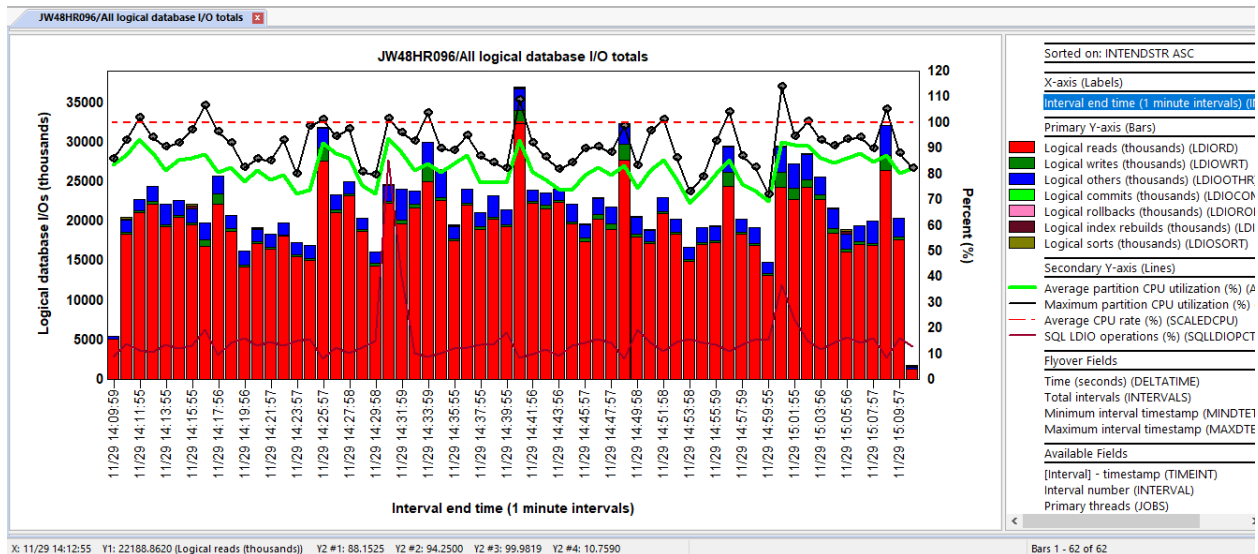
This graph is identical to the previous one except shows the metrics as a rate per second.

23.9 All logical database I/O totals

This graph shows all logical I/O metrics available in Job Watcher.

The metrics shown in this graph include the following:

- 1) Reads
- 2) Writes
- 3) Others (Updates and deletes combined)
- 4) Commits
- 5) Rollbacks
- 6) Index rebuilds
- 7) Sorts



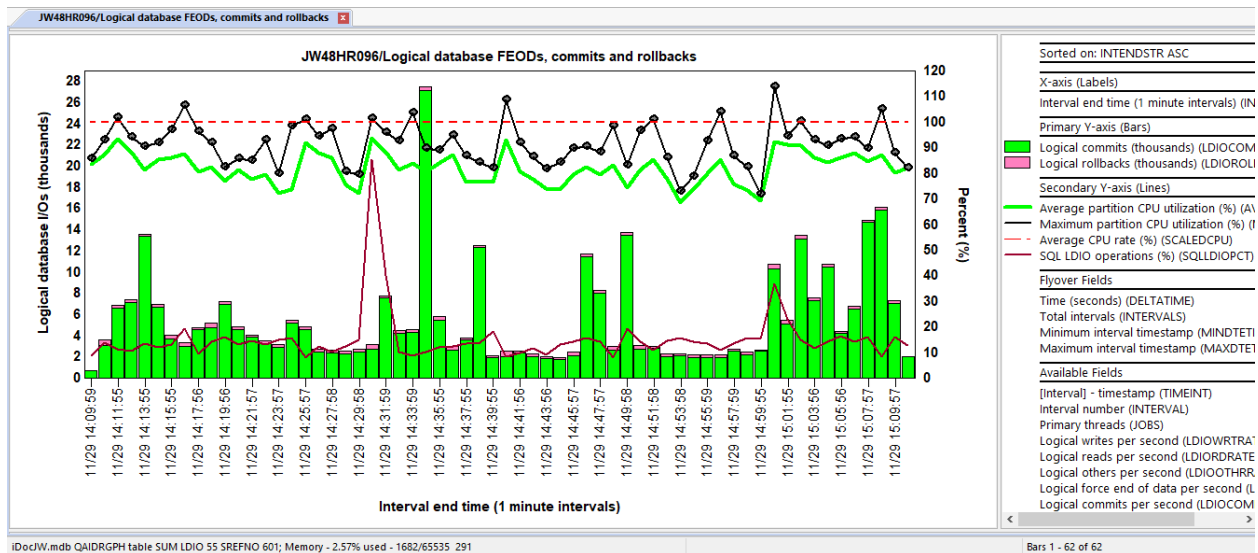
All logical database I/O totals

23.10 All logical database I/O rates

This graph is identical to the previous one except shows the metrics as a rate per second.

23.11 Logical database FEODs, commit and rollbacks

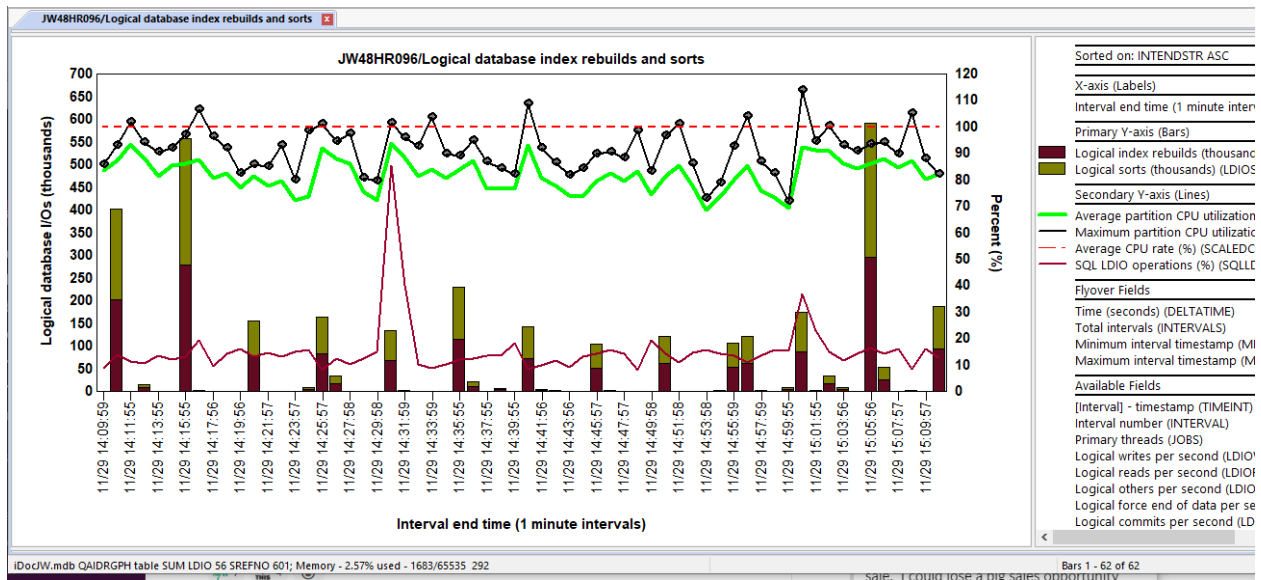
This graph contains totals for just logical DB force end of data, commits and rollbacks over time.



Logical DB FEODs, commit and rollbacks

23.12 Logical database index rebuilds and sorts

This graph contains totals for just logical DB index rebuilds and sorts over time.

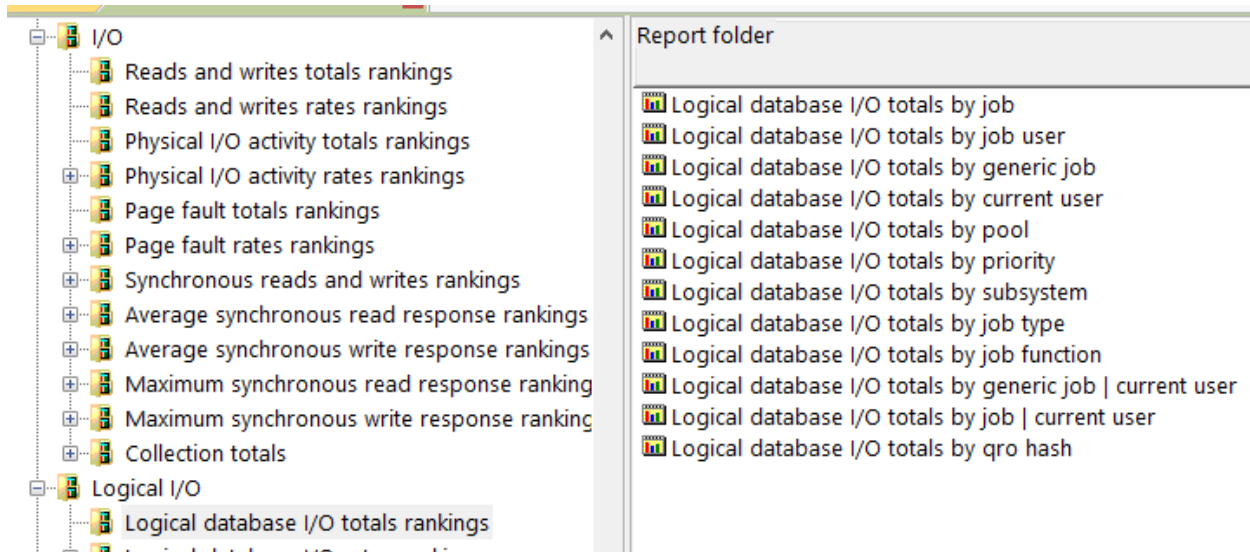


Logical DB index rebuilds and sorts

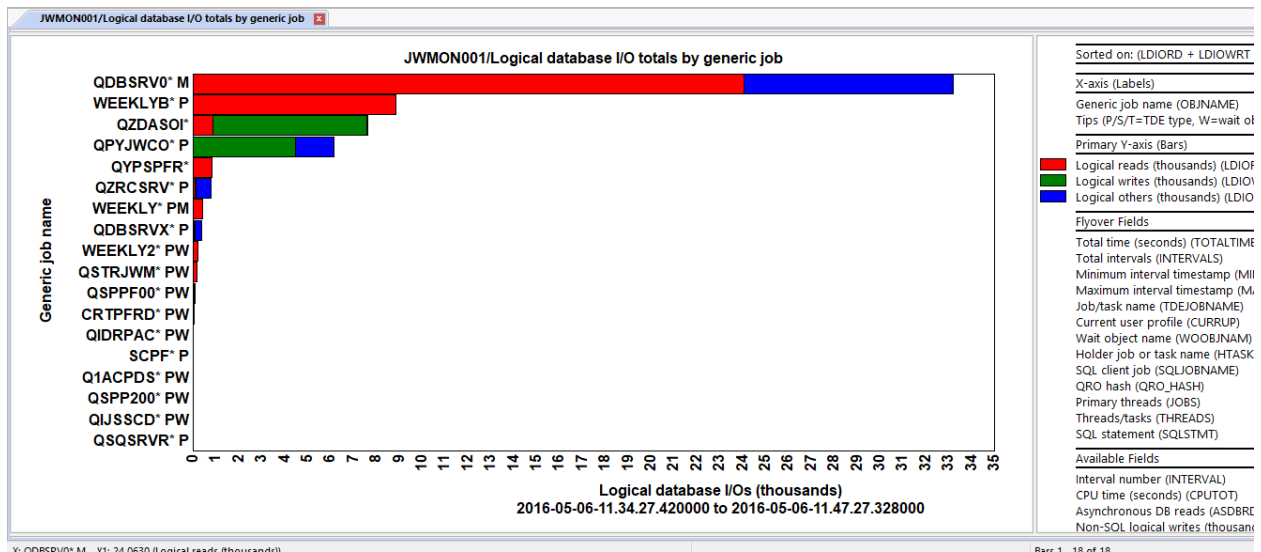
23.13 Logical database I/O totals rankings

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 I/O -> Logical database I/O totals rankings



Logical database I/O totals by generic job

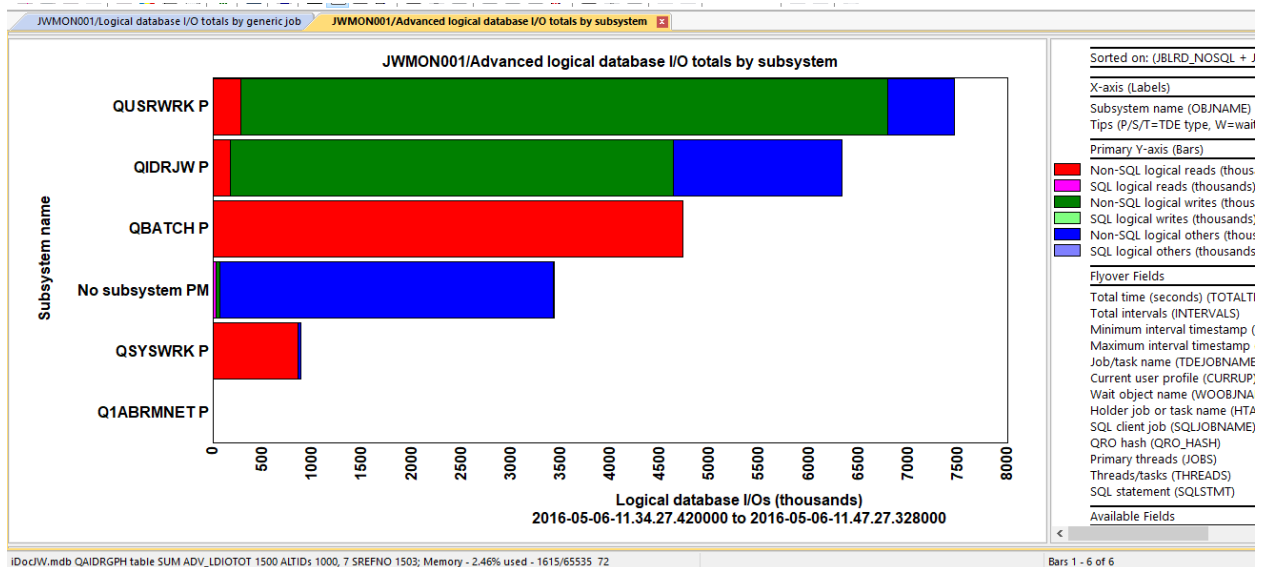
23.14 Logical database I/O rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

23.15 Advanced logical database I/O totals rankings

This folder contains a set of job ranking graphs based on logical DB totals for all jobs in the collection. These graphs show both SQL and non-SQL contributions to logical database I/O operations.

An example follows:



Advanced logical database I/O totals by subsystem

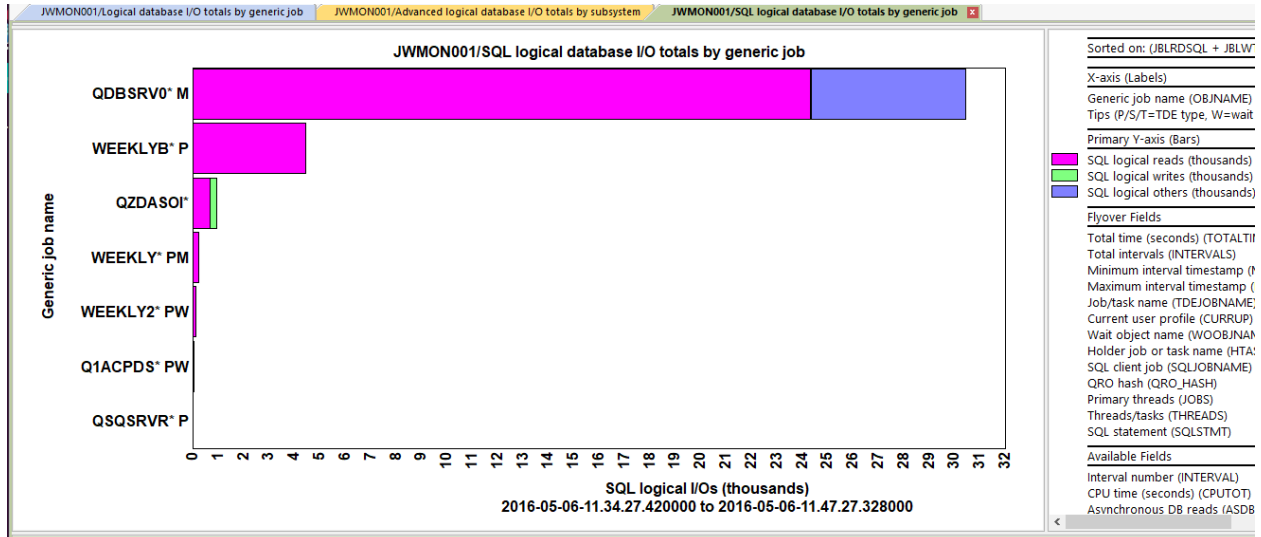
23.16 Advanced logical database I/O rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

23.17 SQL logical database I/O totals rankings

This folder contains a set of job ranking graphs based on logical DB totals for all jobs in the collection. These graphs show both SQL and non-SQL contributions to logical database I/O operations.

An example follows:



SQL logical database I/O totals by generic job

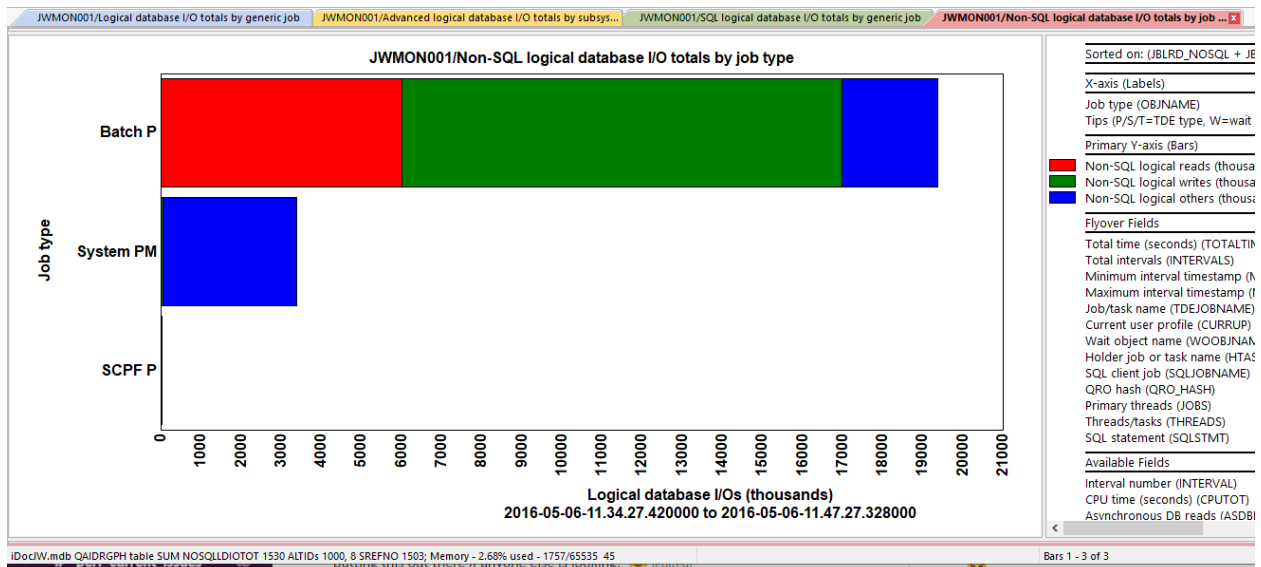
23.18 SQL logical database I/O rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

23.19 Non-SQL logical database I/O totals rankings

This folder contains a set of job ranking graphs based on non-SQL logical DB totals for all jobs in the collection.

An example follows:



Non-SQL logical database I/O totals by job type

23.20 Non-SQL logical database I/O rates rankings

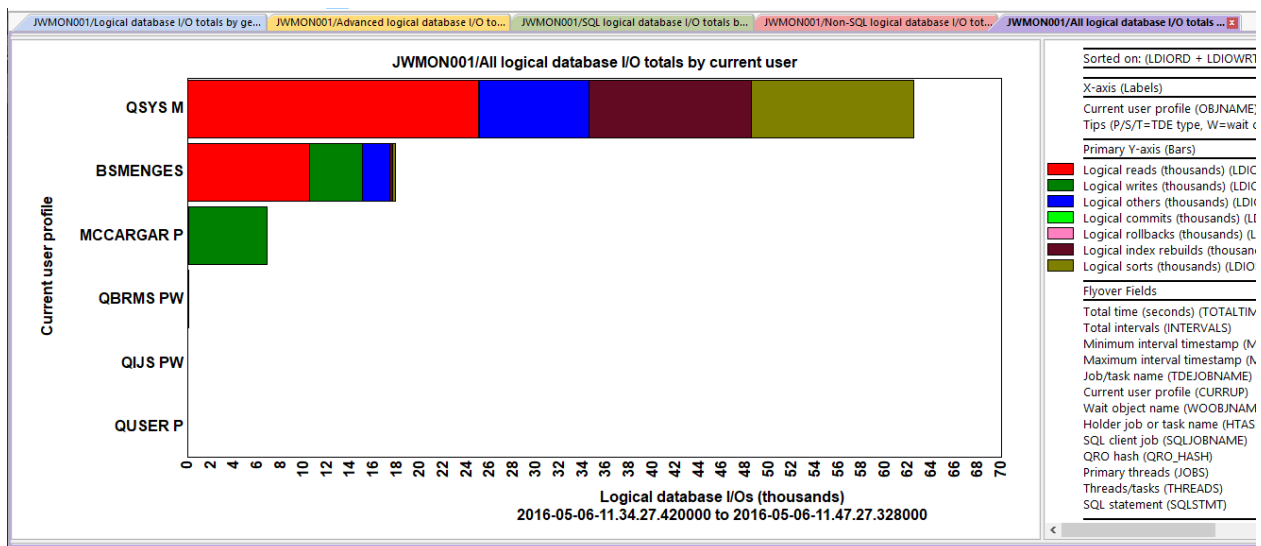
This graph is identical to the previous one except shows the metrics as a rate per second.

23.21 All logical database I/O totals rankings

This folder contains a set of job ranking graphs based on logical DB totals for all jobs in the collection. The metrics shown in this graph include the following:

- 1) Reads
- 2) Writes
- 3) Others (Updates and deletes combined)
- 4) Commits
- 5) Rollbacks
- 6) Index rebuilds
- 7) Sorts

An example follows:



All logical database I/O totals by current user

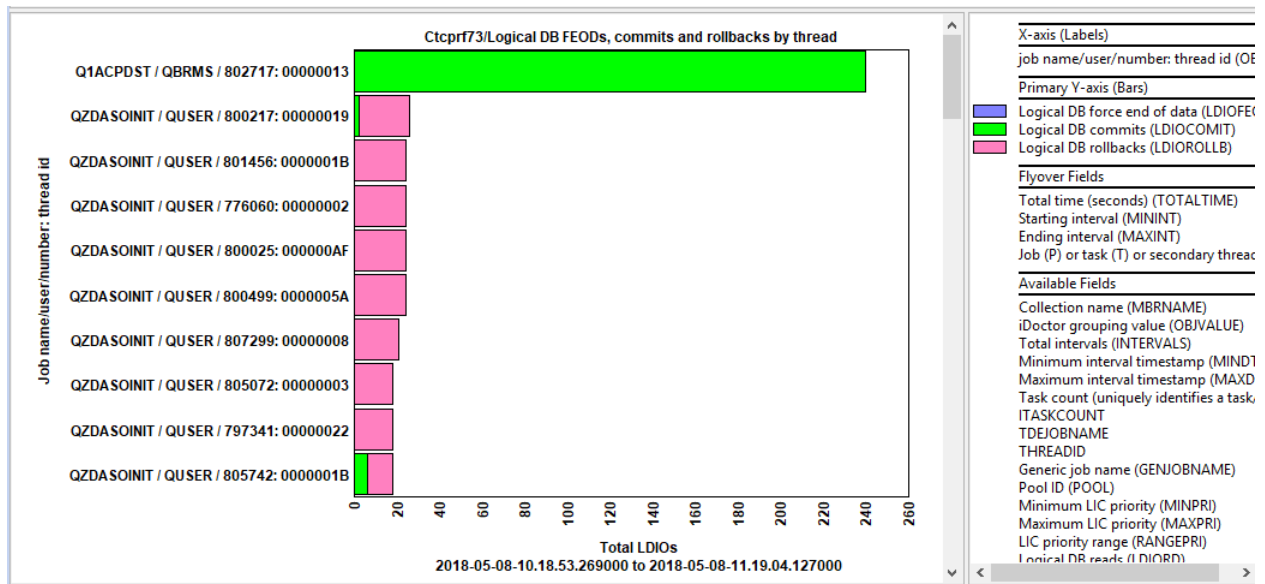
23.22 All logical database I/O rates rankings

This graph is identical to the previous one except shows the metrics as a rate per second.

23.23 Logical database 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.

An example follows:

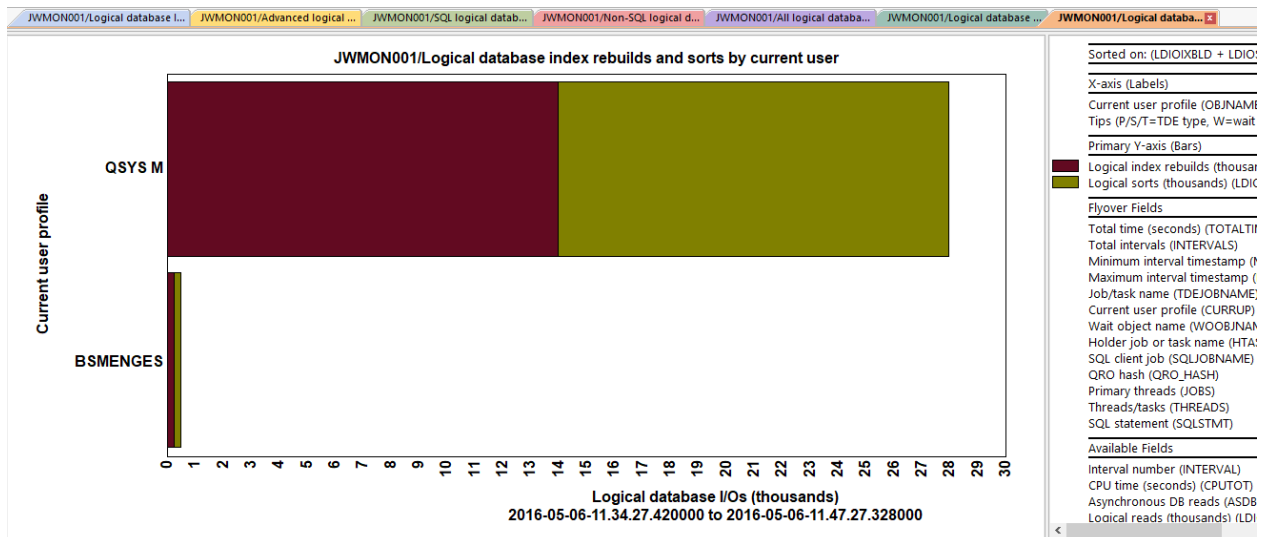


Logical DB FEODs, commit and rollbacks by thread

23.24 Logical database 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.

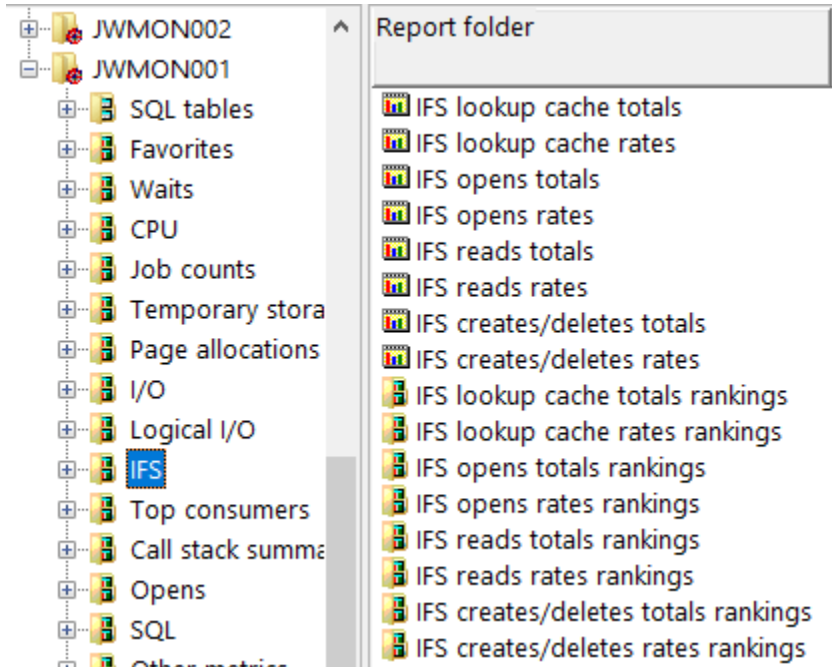
An example follows:



Logical database index rebuilds and sorts by current user

24 IFS

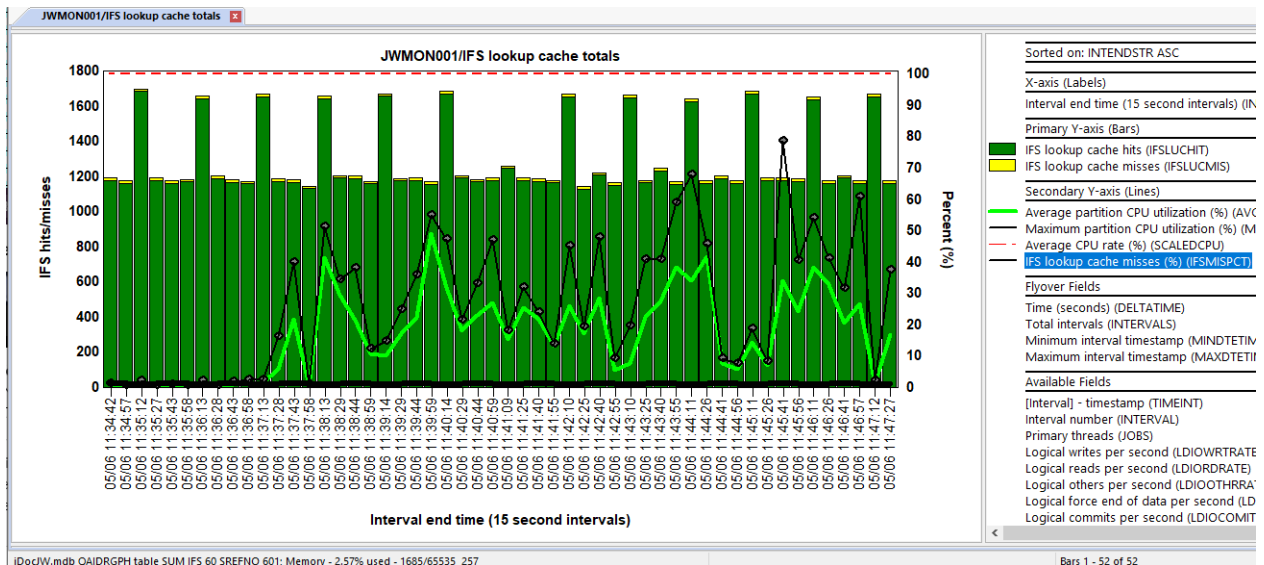
These graphs show IFS statistics for all jobs as either rates or totals over time. These statistics include IFS lookup cache hits/misses, opens, reads (symbolic link reads and directory reads), and creates/deletes.



IFS Folder

24.1 IFS lookup cache totals

This graph shows the IFS lookup cache hits and misses as well as the percentage missed on the Y2-axis along with CPU utilization.



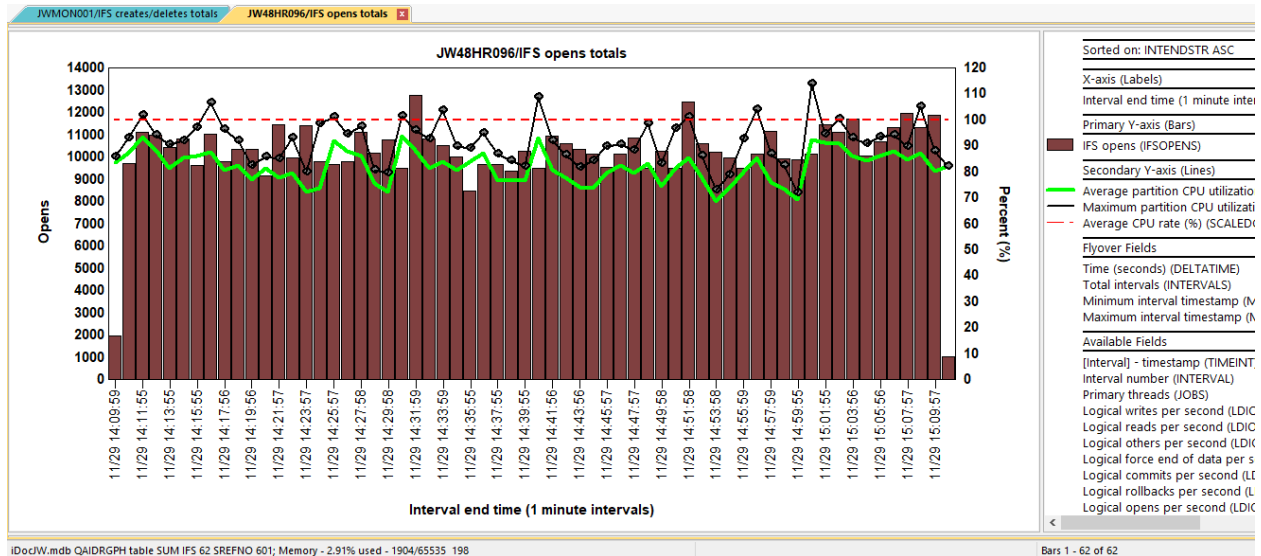
IFS lookup cache totals

24.2 IFS lookup cache rates

This graph is identical to the previous one except shows the metrics as a rate per second.

24.3 IFS opens totals

This graph shows the IFS opens totals over time with CPU utilization on the second Y-axis.



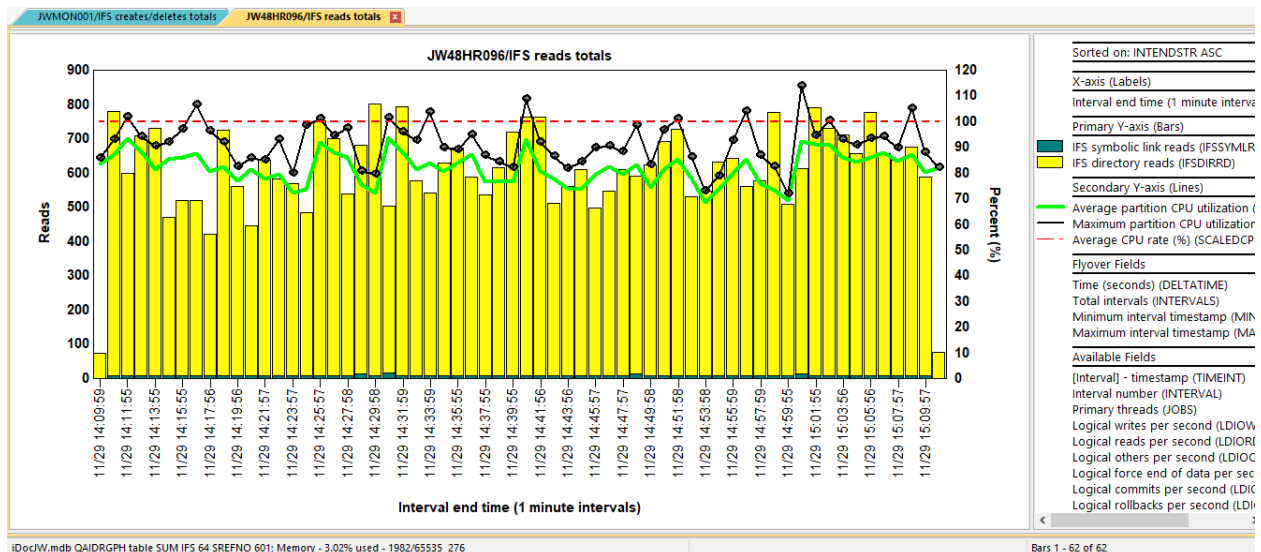
IFS opens totals

24.4 IFS opens rates

This graph is identical to the previous one except shows the metrics as a rate per second.

24.5 IFS reads totals

This graph shows the IFS symbolic link reads and directory reads over time with CPU utilization on the second Y-axis.



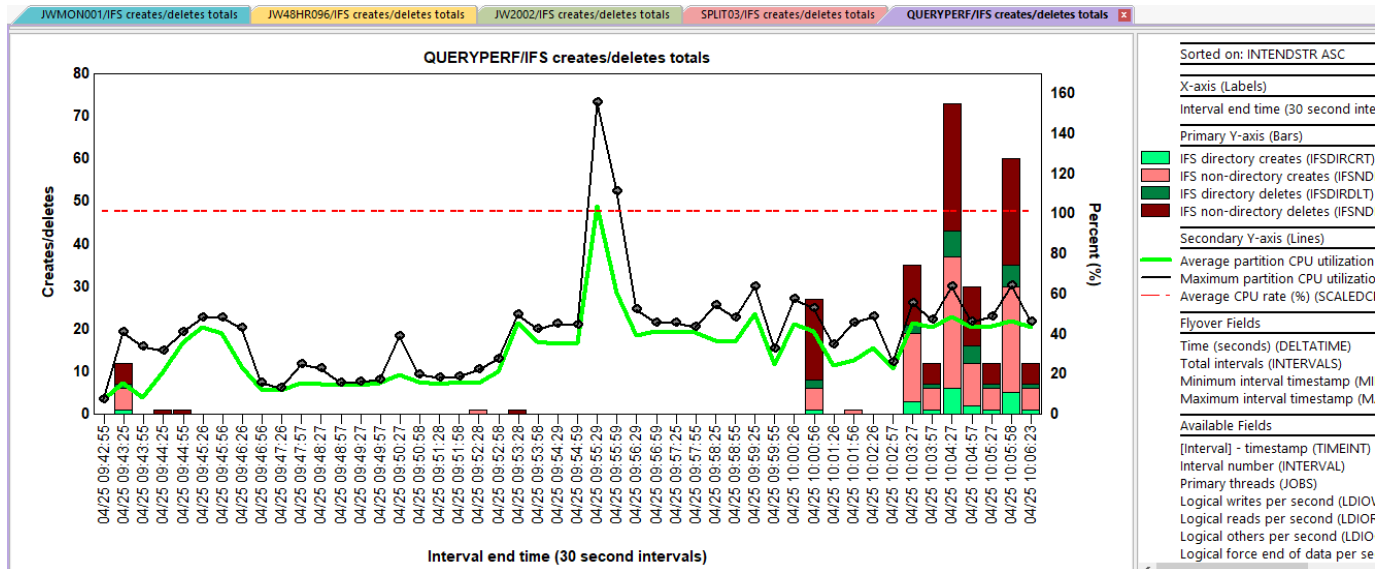
IFS reads totals

24.6 IFS reads rates

This graph is identical to the previous one except shows the metrics as a rate per second.

24.7 IFS creates/deletes totals

This graph shows the IFS creates/deletes totals for both directories and non-directories as well as CPU utilization on the Y2-axis.



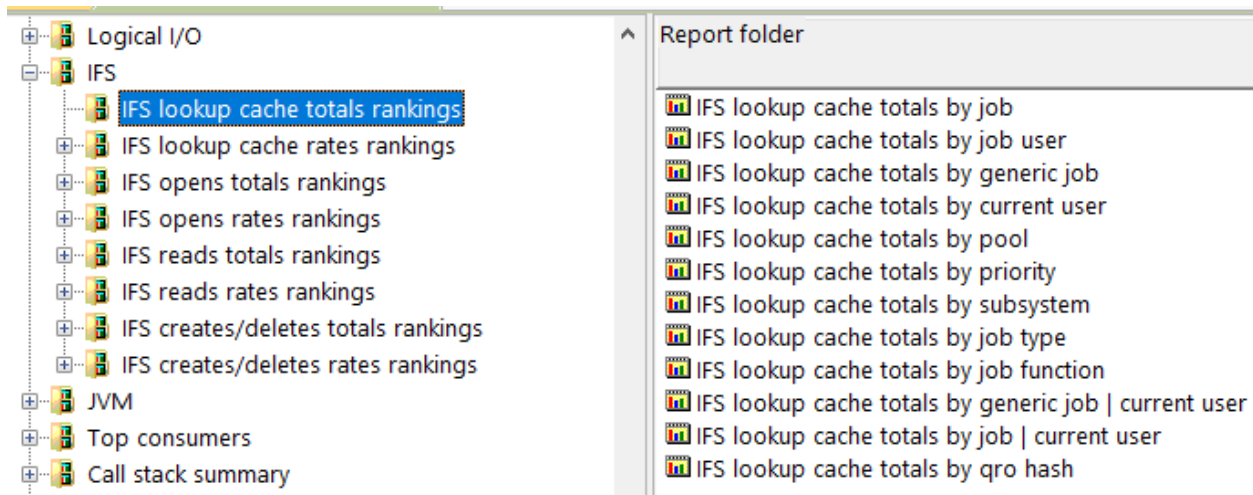
IFS creates/delete totals

24.8 IFS create/deletes rates

This graph is identical to the previous one except shows the metrics as a rate per second.

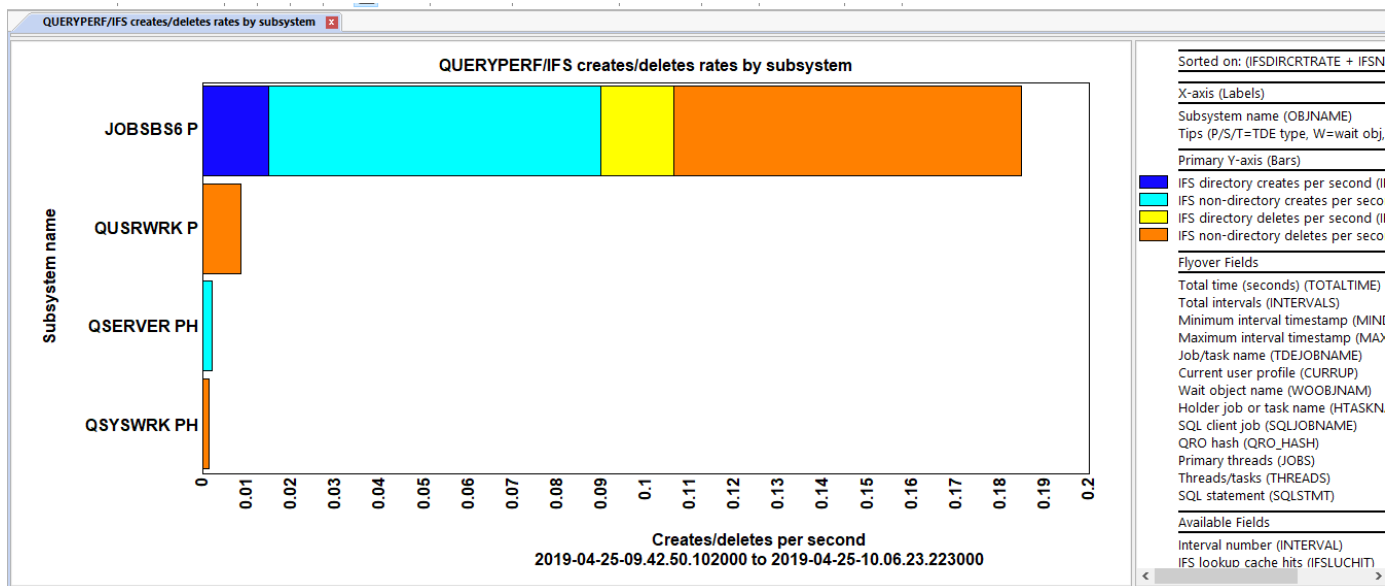
24.9 IFS rankings

Each of the overview graph offers 14 ranking graphs showing the same metrics but grouped in one of several possible ways.



IFS -> IFS lookup cache totals rankings

An example is shown below:

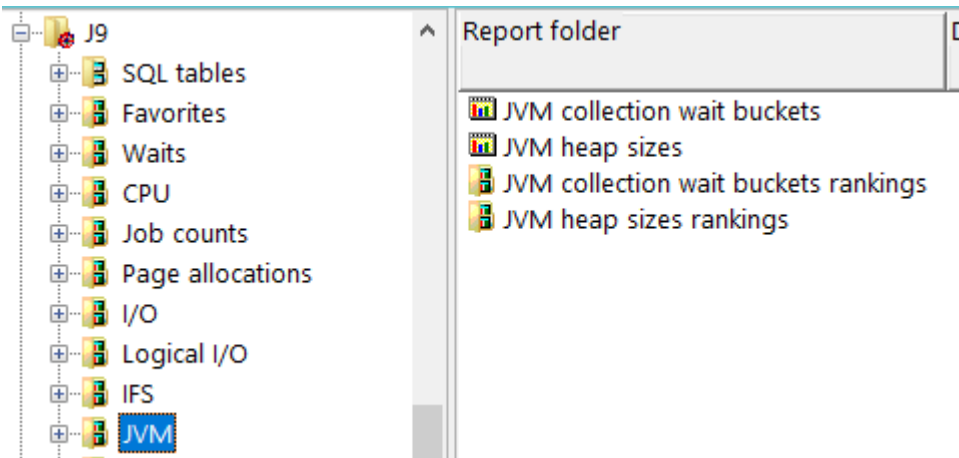


IFS creates/deletes rates by subsystem

25 JVM

The JVM folder shows statistics related to the J9 JVMs (IBM Technology for Java) jobs and the JVMs running within them found in the collection.

Tip: The **JVM collection wait buckets graph** shows the wait buckets for all JVM running jobs added up together.



JVM Folder

Note #1: 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.

Note #2: A flaw in the STRJW engine when collecting these statistics may cause the collection intervals to become longer than they should be. For example, 5 second intervals, may become 30 seconds or a minute of random durations. As of April 2022, this is not yet fixed.

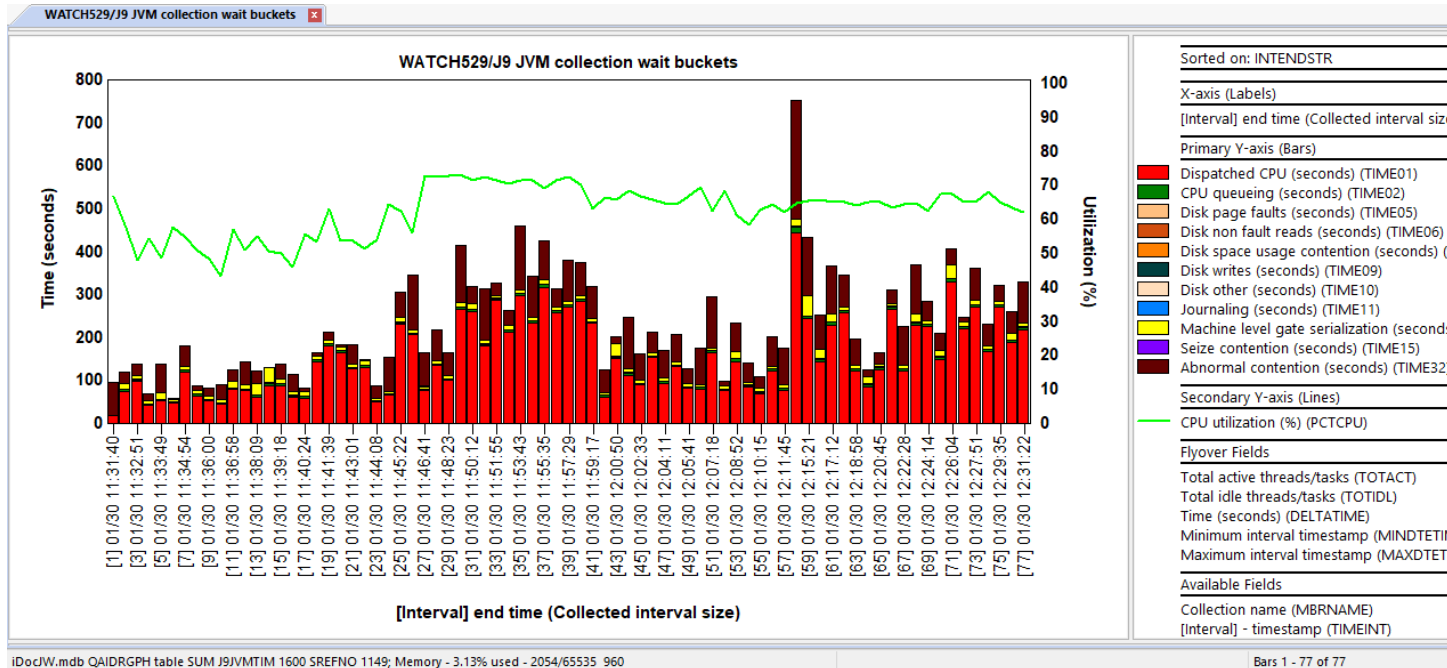
Other flaws in STRJW in table QAPYJWIJVM are:

- 1) The last GC cycle duration is sometimes incorrectly reported in file QAPYJWIJVM. Field JMGCLTME (GC time last cycle) will often exceed JMTGCTTME (Total GC time) which should not happen.
- 2) The mark, sweep and compact durations are frequently way too large and higher than GC duration which is impossible. For that reason, those graphs have been removed.
- 3) The references cleared metrics are often negative and assumed to be trash. For that reason, those graphs have been removed.
- 4) Empty records in the file may exist like this with no taskcount.

Interval number (INTERVAL)	Process identifier (JMPID)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	JVM version (JMVRSN)	JVM type (JMSTYPE)	Garbage collection policy (JMPOLICY)	Reserved (JMRESERVE2)	JVM start time (JMSTRTIM)	Initial heap size (KB) (JMINITSZ)	Max heap size (KB) (JMHEAPSZ)
1	1,633,809	0	0	0			1928-08-23-12.03.06.315000	0	0
1	1,633,892	0	0	0			1928-08-23-12.03.06.315000	0	0
1	1,633,737	0	0	0			1928-08-23-12.03.06.315000	0	0
1	1	0	0	0			1928-08-23-12.03.06.315000	0	0

25.1 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.

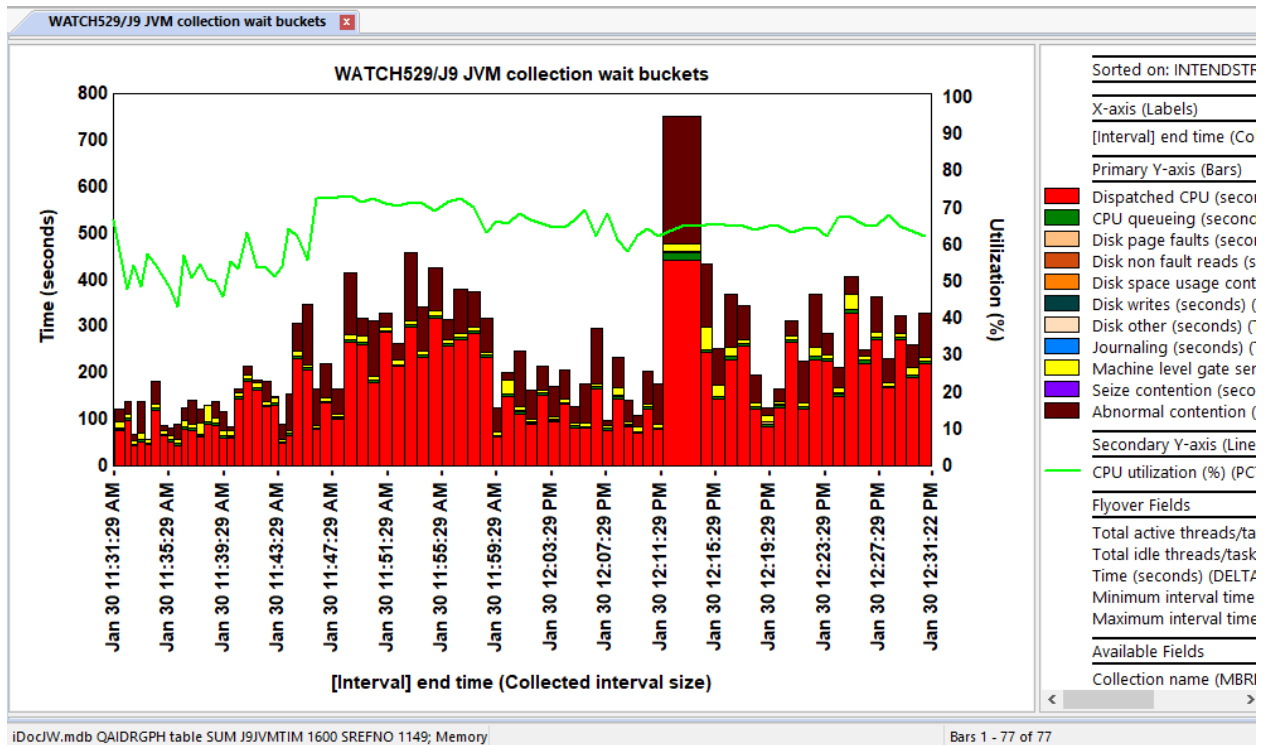


JVM collection wait buckets (10 second intervals)

Regarding the issue with the STRJW command and this graph, this is a good example of the benefit of using the variable-width bar mode toolbar button.



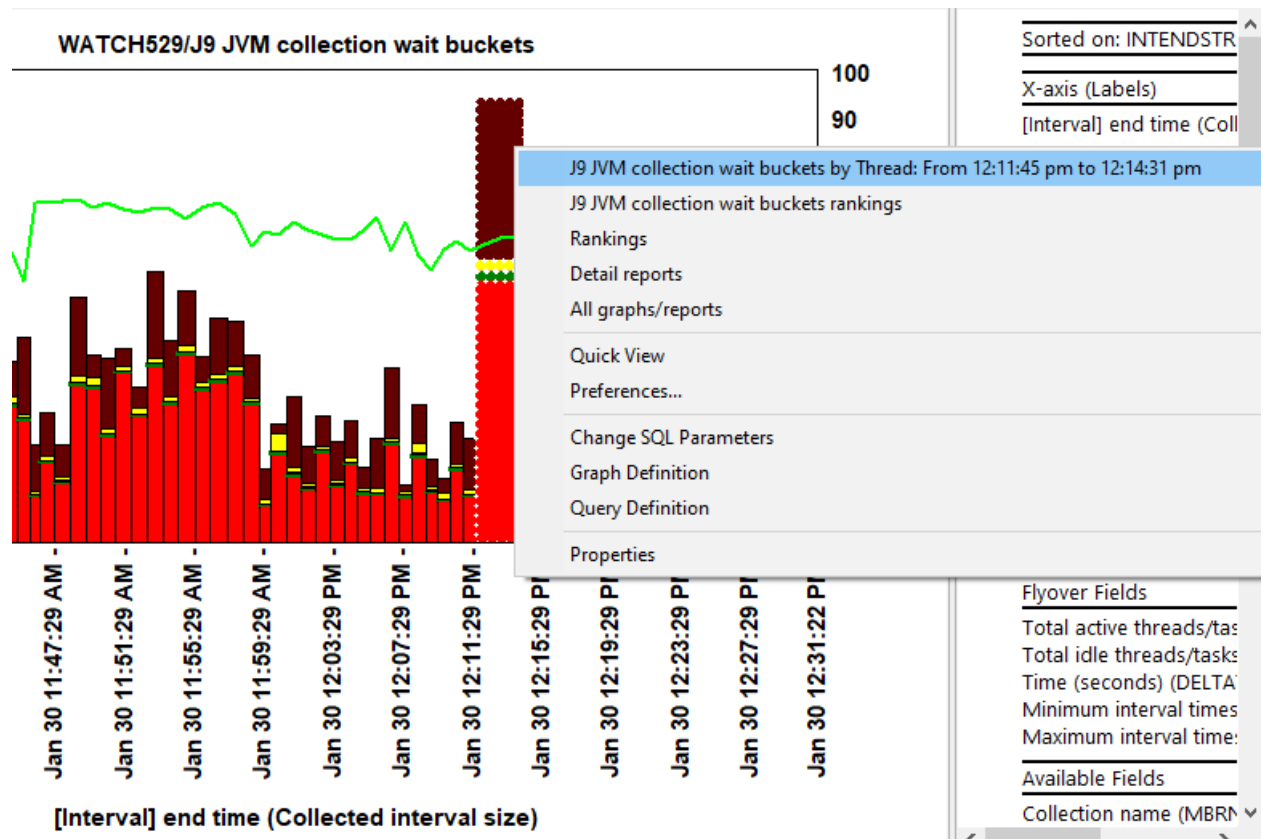
Using that, the graph is redrawn showing wider bars for longer duration intervals. The longest was 165 seconds!



J9 JVM collection wait buckets (variable-width bars)

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.



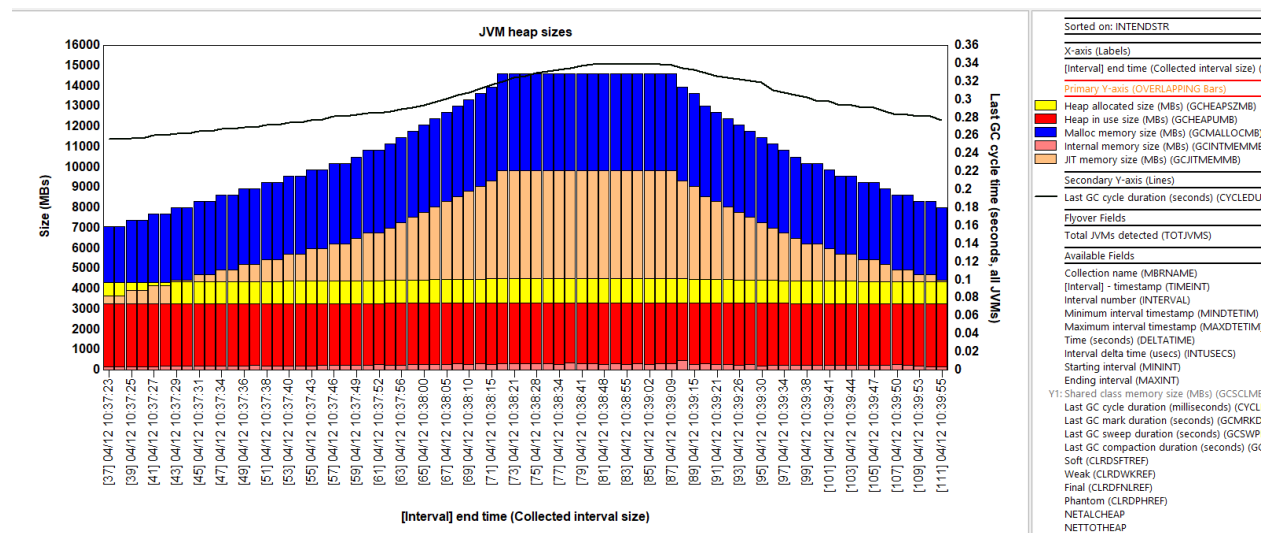
JVM collection wait buckets drill-down menu

25.2 JVM heap sizes

This graph displays the following 6 metrics (all in megabytes) using overlapping bars:

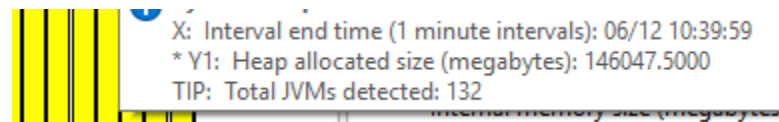
- 1) Heap allocated size
- 2) Heap in use size
- 3) Malloc memory size
- 4) Internal memory size
- 5) JIT memory size
- 6) Shared class memory size

These values are added up across all JVMs found in the collection. This graph also includes the total last GC cycle duration (in milliseconds) for all JVMs added up on the Y2-axis.



JVM heap sizes

Tip: The total JVMs detected can be shown by looking at the graph flyover.



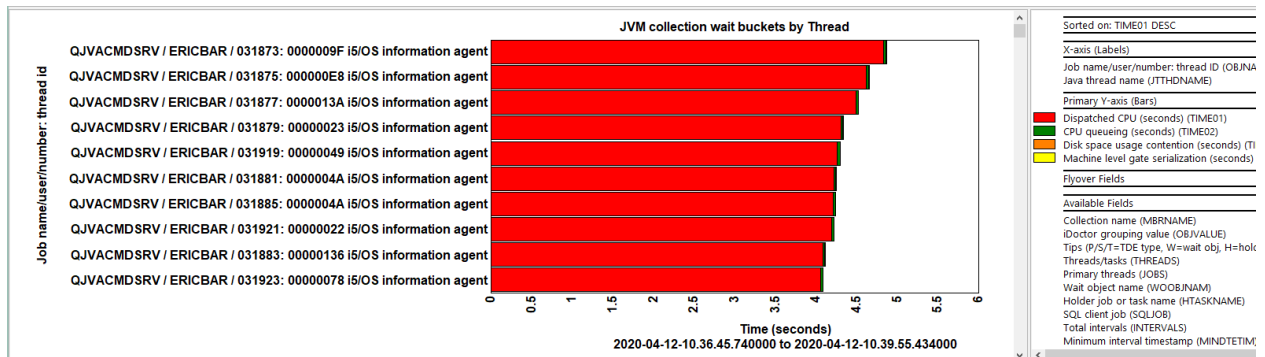
JVM heap sizes graph flyover

25.3 JVM collection wait buckets by Thread

This graph provides a ranking of the J9 JVM job/threads sorted by Dispatched CPU time. 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.

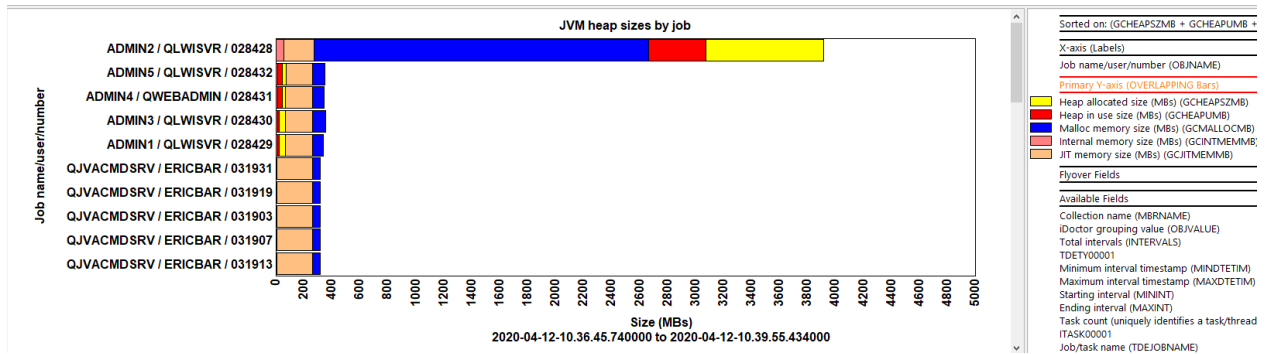
Note: This graph requires the Collection summary analysis to be ran in order to appear!



JVM collection wait buckets by thread

25.4 JVM heap sizes by job

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.

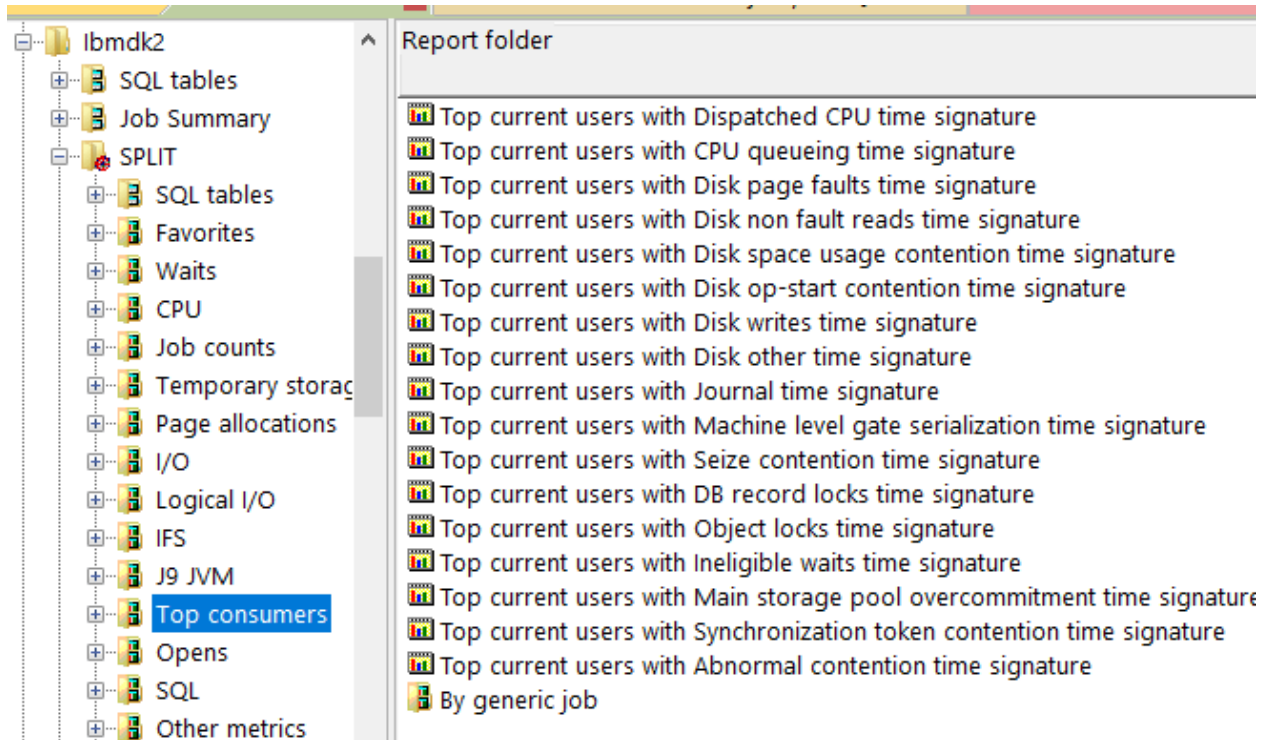


JVM heap sizes by job

26 Top consumers

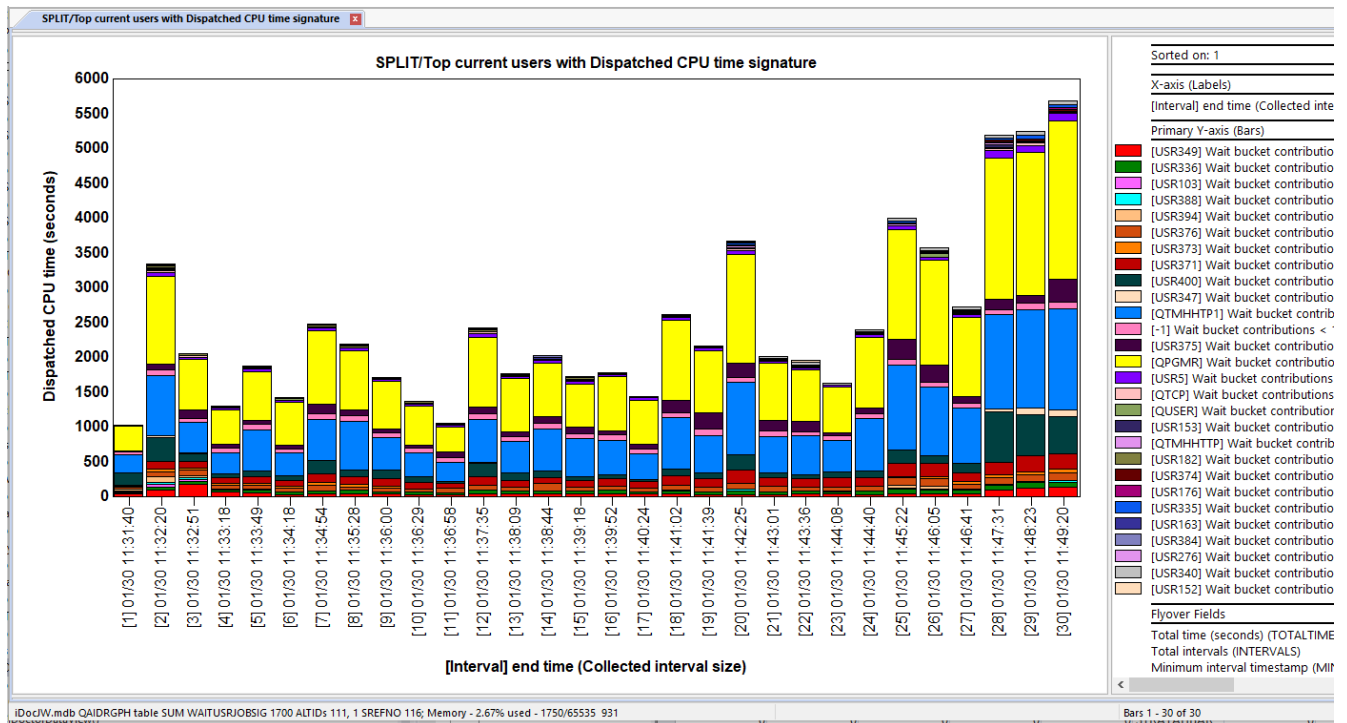
These graphs are a special-type of graph called “flattened graphs” that will show a variable number of top contributors of the desired metric over time. These show the top current users or generic jobs experiencing the most time one of the “interesting” wait buckets.

The graphs work best with smaller number of contributors per bar (< 20 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 10 seconds and will group all values having < 10 seconds of the bucket time together into 1 color.



Top consumers folder

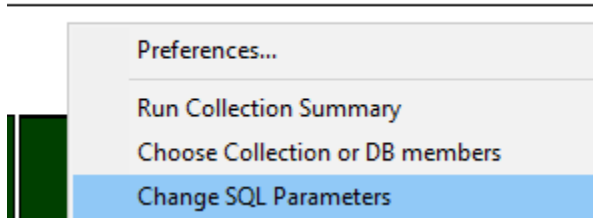
An example of this type of graph is:



Top current users using Dispatched CPU time signature (10 second filter)

To change the filter, right-click the graph and use the **Change SQL Parameters** menu:

patched CPU time signature



Change SQL Parameters menu

Here you can make the “Flattened” graphing filter value larger or smaller as desired.

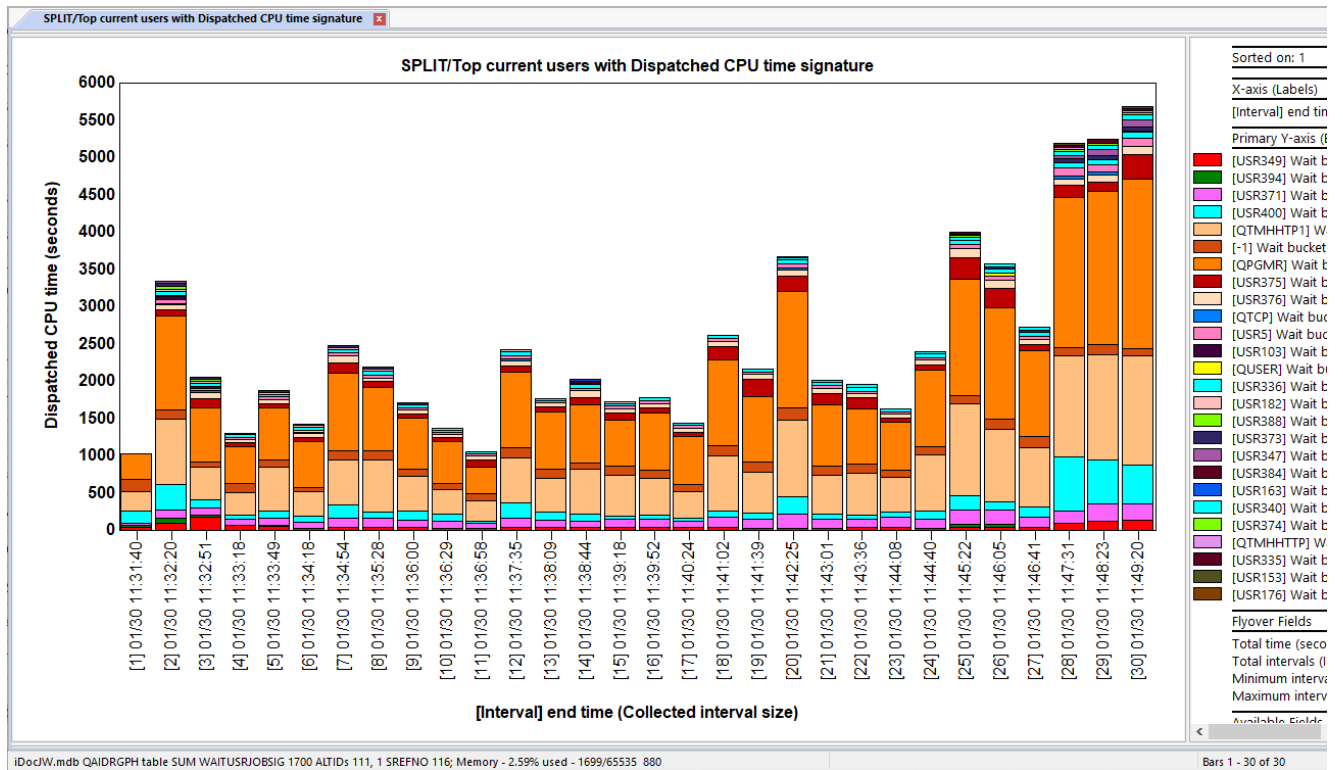
Change SQL Parameters

This interface allows you to modify the current SQL statement by changing the parameters shown.

Minimum value to include (<<STACKEDFILTER>>)

Label contains (filter) (<<STACKEDNAME>>)

Change SQL Parameters window

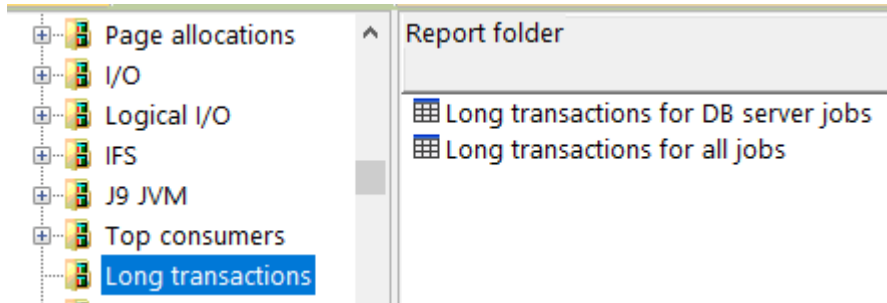


Top current users using Dispatched CPU time signature (20 second filter)

27 Long Transactions

This contains reports that shows periods of time where bursts of activity occurred where zero “idle” wait bucket time exists. These reports look at and apply to only the wait bucket times and do not show anything related to 5250 transactions.

Note: This folder appears after running the **Long Transactions** analysis.



Long transactions folder

27.1 Long transactions for DB server jobs

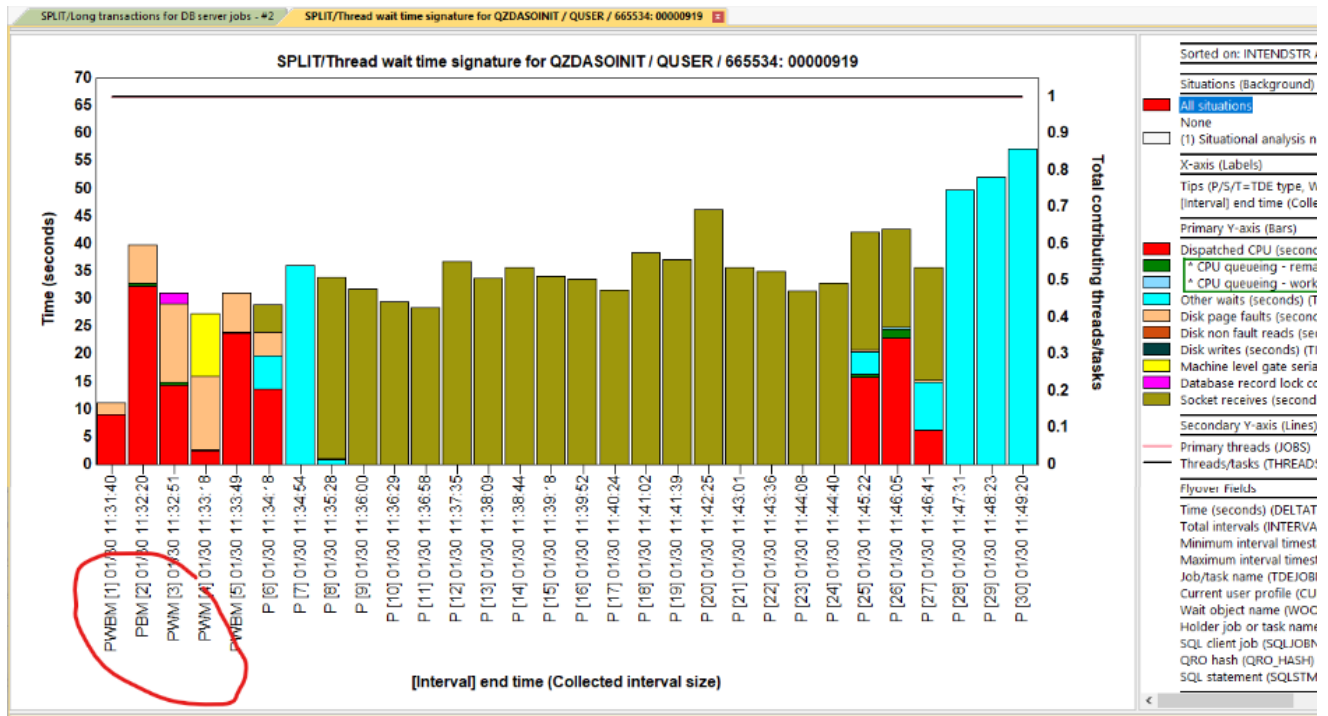
This report contains only jobs and periods of time related to database server jobs.

The job information, number of intervals and start and end intervals where bursts of activity occurred is indicated in the report.

SPLIT/Long transactions for DB server jobs - #2						
Total intervals (INTERVALS)	Job/task name (TDEJOBNAME)	Thread ID (THREADID)	Job user profile (if constant) (JOBCURRUP)	Interval number (INTERVAL)	Ending interval (MAXINT)	Task count (uniquely identifies a task/thread) (TASKCOUNT)
2	QZDASOINITQUSER	665534	0000000000000919	-1	1	3
						22,793,476

Long transactions for DB server jobs

Tip: Right-click the desired job for drill down options such as graph the job over time.



Thread wait time signature for QZDASOINIT

27.2 Long transactions for all jobs

This report contains all jobs where bursts of activity with no idle waits were detected.

The job information, number of intervals and start and end intervals where bursts of activity occurred is indicated in the report.

SPLIT/Long transactions for DB server jobs - #2		SPLIT/Thread wait time signature for QZDASOINIT / QUSER / 665534: 00000919		SPLIT/Long transactions for all jobs - #			
Total intervals (INTERVALS)	Job/task name (TDEJOBNAME)	Thread ID (THREADID)	Job user profile (if constant) (JOBCURRUP)	Interval number (INTERVAL)	Ending interval (MAXINT)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	
8	JOB986 QPGMR	676632	0000000000000091	17	25	22,815,285	
6	JOB986 QPGMR	540452	00000000000000C27	19	25	22,586,427	
6	JOB964 QPGMR	542070	000000000000013D4	19	25	22,594,805	
6	JO-RECRA-U-03-00		00000000000000000	20	26	5,859	
6	JO-RECRA-U-03-01		00000000000000000	20	26	5,860	
6	JO-RECRA-U-05-01		00000000000000000	20	26	5,864	
5	JOB924 QPGMR	550409	000000000000009D8	5	10	22,608,659	
5	JOB63 QPGMR	676596	00000000000000823	13	18	22,815,241	
5	JOB32 USR349	674752	00000000000001030	13	18	22,816,412	
5	JOB964 QPGMR	540409	00000000000000AF4	19	24	22,586,157	
5	JOB1007 QPGMR	541087	000000000000012B6	19	24	22,589,276	
5	JOB63 QPGMR	673999	00000000000000752	19	24	22,810,335	
-	-----	-----	-----	--	--	-----	

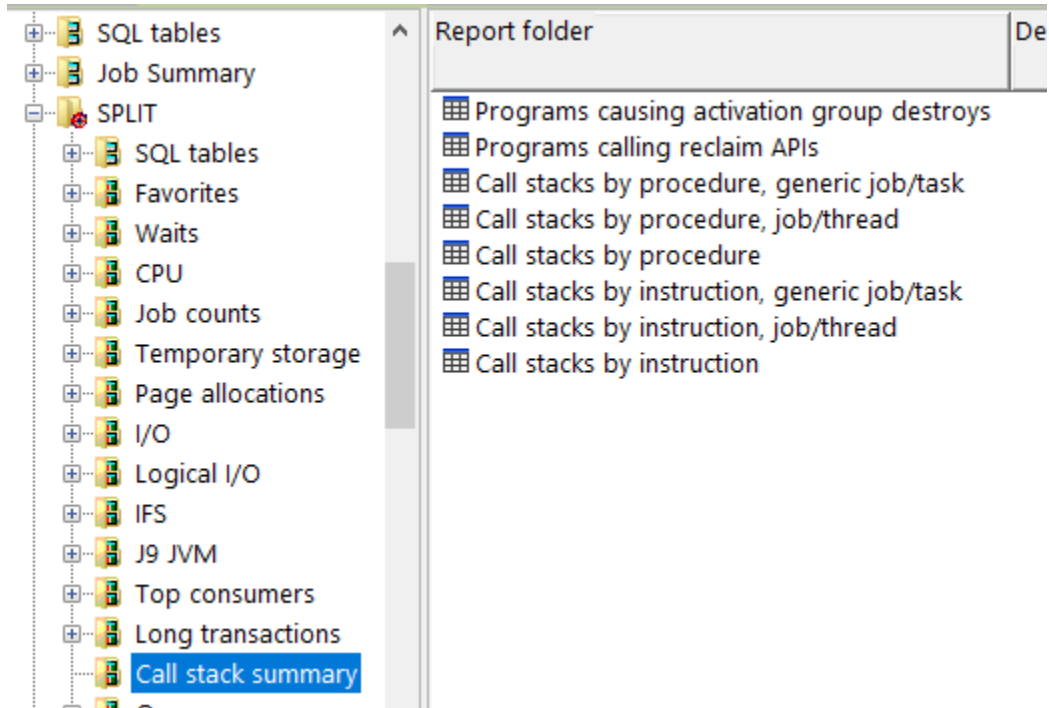
Long transactions for all jobs

Tip: Right-click the desired job for more options.

28 Call stack summary

This folder contains reports that summarize the call stacks found in the collection. The most commonly occurring call stacks are grouped together based on the options taken when the analysis was last executed.

Note: This folder requires that the Call stack summary analysis is ran!



Call stack summary folder

Tip: Only the **by instruction** reports contain the offset and statement number columns.

An example follows:

IBM iDoctor for IBM i

SPLIT/Call stacks by instruction - #1												
Total (TOTAL)	Call level (LEVEL)	Program model (MODEL)	Program name (PGMN...	Module name (MOD...	Procedure (PROCNAME)					Offset (ADDR_OFFSET)	Statement number (STMTNBR)	Ir a (I
146655	1	LIC			qutde_block_trace					000000F8	248	f
146655	2	LIC			longWaitBlock_23QuSingleTaskBlockerCodeFP20QuBaseLongWaitObjectR12RmprReceiverQ2_8TDQSEnum4Enum					000002CC	716	f
146655	3	LIC			do_sleepWait_12PpPaseThreadFQ2_8TDQSEnum4EnumUI					000000F8	248	f
146655	4	LIC			do_tsleep_FUIT1					000000D0	208	f
146655	5	LIC			tia_schandler					0000012C	300	f
146655	6	PASE			P3					00000008	0	(
146655	7	PASE			P98					00000584	0	(
146655	8	PASE			P99					00000350	0	(
146655	9	PASE			P93					000004FC	0	(
146655	10	PASE			P92					000000C8	0	(
146655	11	PASE			P2125					000001A0	0	(
146655	12	PASE			P1936					000008D4	0	(
146655	13	PASE			P1842					00000038	0	(
146655	14	PASE			P2439					00000168	0	(
146655	15	PASE			P1437					000002D0	0	(
146655	16	PASE			P1836					000002EC	0	(
44814	1	LIC			qutde_block_trace					000000F8	248	f
44814	2	LIC			longWaitBlock_23QuSingleTaskBlockerCodeFP20QuBaseLongWaitObjectR12RmprReceiverQ2_8TDQSEnum4Enum					000002CC	716	f
44814	3	LIC			do_sleepWait_12PpPaseThreadFQ2_8TDQSEnum4EnumUI					000000F8	248	f
44814	4	LIC			do_tsleep_FUIT1					000000D0	208	f
44814	5	LIC			tia_schandler					0000012C	300	f
44814	6	PASE			P4					00000008	0	(
44814	7	PASE			P98					00000584	0	(
44814	8	PASE			P99					00000348	0	(
44814	9	PASE			P93					000004E0	0	(
44814	10	PASE			P92					000000CC	0	(
44814	11	PASE			P2125					000001A0	0	(
44814	12	PASE			P1936					00000884	0	(
44814	13	PASE			P1844					00000044	0	(
44814	14	PASE			P1939					00000190	0	(
44814	15	PASE			P3589					00000020	0	(

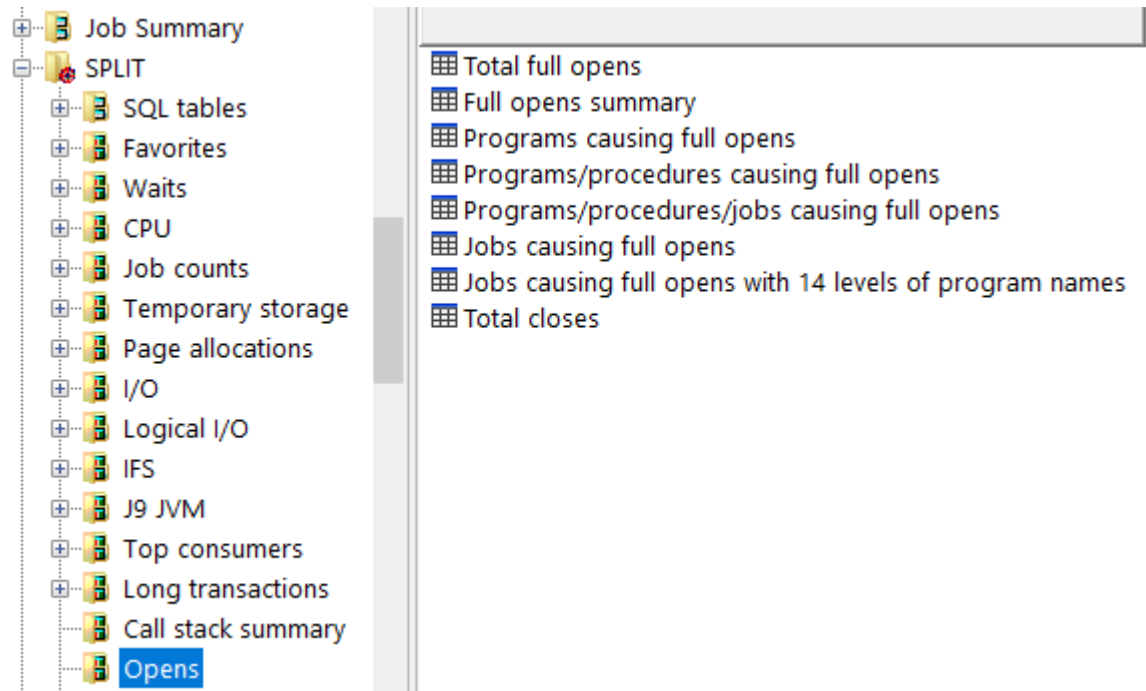
Call stacks by instruction

29 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.

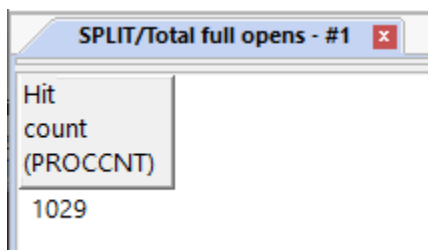
Tip: An easier option is to use PEX Analyzer's **Database opens** analysis instead.



Opens folder

29.1 Total full opens

This report simply counts the total full opens captured in the collection.



Total full opens

29.2 Full opens summary

This report identifies the types of opens found in the collection.

SPLIT/Full opens summary - #1		
Hit count (PROCCNT)	Program open type (OPENTYPE)	Procedure (PROCNAME)
1027	ILE RPG Native	_QRNX_OPEN
3	CQE	OPENUFCB

Full opens summary

29.3 Programs causing full opens

This report shows which programs most commonly caused the full opens found in the Job Watcher call stacks sampled.

SPLIT/Programs causing full opens - #1				
Hit count (PROCCNT)	Program open type (OPENTYPE)	Procedure type (PROCTYPE)	Program library (PGMLIB)	Program name (PGMNAME)
112	ILE RPG Native	ILE	LIB9	PGM46
68	ILE RPG Native	ILE	LIB9	PGM87
64	ILE RPG Native	ILE	LIB9	PGM49
55	ILE RPG Native	ILE	LIB9	PGM43
46	ILE RPG Native	ILE	LIB9	PGM32
43	ILE RPG Native	ILE	LIB9	PGM22
34	ILE RPG Native	ILE	LIB9	PGM67
29	ILE RPG Native	ILE	LIB9	PGM25
25	ILE RPG Native	ILE	LIB9	PGM28
24	ILE RPG Native	ILE	LIB9	PGM995
22	ILE RPG Native	ILE	LIB9	PGM13
22	ILE RPG Native	ILE	LIB9	PGM85
22	ILE RPG Native	ILE	LIB9	PGM62
21	ILE RPG Native	ILE	LIB9	PGM23
19	ILE RPG Native	ILE	LIB70	PGM1567
18	ILE RPG Native	ILE	LIB9	PGM40
18	ILE RPG Native	ILE	LIB9	PGM54
18	ILE RPG Native	ILE	LIB10	PGM325
18	ILE RPG Native	ILE	LIB9	PGM53
16	ILE RPG Native	ILE	LIB9	PGM35

Programs causing full opens

29.4 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.

SPLIT/Programs/procedures causing full opens - #1						
Hit count (PROCCNT)	Program open type (OPENTYPE)	Procedure type (PROCTYPE)	Program library (PGMLIB)	Program name (PGMNAME)	Procedure (PROCNAME)	Open procedure (PROCCALLER)
112	ILE RPG Native	ILE	LIB9	PGM46	P125	_QRNX_OPEN
68	ILE RPG Native	ILE	LIB9	PGM87	P125	_QRNX_OPEN
64	ILE RPG Native	ILE	LIB9	PGM49	P125	_QRNX_OPEN
55	ILE RPG Native	ILE	LIB9	PGM43	P125	_QRNX_OPEN
46	ILE RPG Native	ILE	LIB9	PGM32	P125	_QRNX_OPEN
43	ILE RPG Native	ILE	LIB9	PGM22	P125	_QRNX_OPEN
34	ILE RPG Native	ILE	LIB9	PGM67	P125	_QRNX_OPEN
29	ILE RPG Native	ILE	LIB9	PGM25	P125	_QRNX_OPEN
25	ILE RPG Native	ILE	LIB9	PGM28	P125	_QRNX_OPEN
24	ILE RPG Native	ILE	LIB9	PGM995	P5076	_QRNX_OPEN
22	ILE RPG Native	ILE	LIB9	PGM62	P125	_QRNX_OPEN
22	ILE RPG Native	ILE	LIB9	PGM85	P125	_QRNX_OPEN
22	ILE RPG Native	ILE	LIB9	PGM13	P125	_QRNX_OPEN
21	ILE RPG Native	ILE	LIB9	PGM23	P125	_QRNX_OPEN
19	ILE RPG Native	ILE	LIB70	PGM1567	P4920	_QRNX_OPEN
18	ILE RPG Native	ILE	LIB9	PGM40	P125	_QRNX_OPEN
18	ILE RPG Native	ILE	LIB10	PGM325	P3388	_QRNX_OPEN
18	ILE RPG Native	ILE	LIB9	PGM54	P125	_QRNX_OPEN
18	ILE RPG Native	ILE	LIB9	PGM53	P125	_QRNX_OPEN
16	ILE RPG Native	ILE	LIB9	PGM35	P125	_QRNX_OPEN
14	ILE RPG Native	ILE	LIB10	PGM252	P3092	_QRNX_OPEN
13	ILE RPG Native	ILE	LIB82	PGM1519	P3251	_QRNX_OPEN
13	ILE RPG Native	ILE	LIB9	PGM14	P125	_QRNX_OPEN

Programs/procedures causing full opens

29.5 Programs/procedures/jobs causing full opens

This report is the same as the previous one but also includes the job name and job user.

SPLIT/Programs/procedures causing full opens - #1				SPLIT/Programs/procedures/jobs causing full opens - #1				
Hit count (PROCCNT)	Program open type (OPENTYPE)	JOB (JOB)	USER (USER)	Procedure type (PROCTYPE)	Program library (PGMLIB)	Program name (PGMNAME)	Procedure (PROCNAME)	Open procedure (PROCCALLER)
33	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM49	P125	_QRNX_OPEN
32	ILE RPG Native	JOB77	QPGMR	ILE	LIB9	PGM46	P125	_QRNX_OPEN
21	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM46	P125	_QRNX_OPEN
17	ILE RPG Native	JOB114	QPGMR	ILE	LIB9	PGM46	P125	_QRNX_OPEN
17	ILE RPG Native	JOB63	QPGMR	ILE	LIB9	PGM46	P125	_QRNX_OPEN
16	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM43	P125	_QRNX_OPEN
16	ILE RPG Native	JOB63	QPGMR	ILE	LIB9	PGM43	P125	_QRNX_OPEN
13	ILE RPG Native	JOB205	QPGMR	ILE	LIB70	PGM1567	P4920	_QRNX_OPEN
12	ILE RPG Native	JOB63	QPGMR	ILE	LIB9	PGM32	P125	_QRNX_OPEN
12	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM87	P125	_QRNX_OPEN
12	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM62	P125	_QRNX_OPEN
12	ILE RPG Native	JOB114	QPGMR	ILE	LIB9	PGM22	P125	_QRNX_OPEN
12	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM30	P125	_QRNX_OPEN
11	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM67	P125	_QRNX_OPEN
10	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM32	P125	_QRNX_OPEN
10	ILE RPG Native	JOB63	QPGMR	ILE	LIB9	PGM87	P125	_QRNX_OPEN
10	ILE RPG Native	JOB68	QPGMR	ILE	LIB9	PGM54	P125	_QRNX_OPEN

Programs/procedures/jobs causing full opens

29.6 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.

SPLIT/Programs/procedures causing full opens - #1				SPLIT/Programs/procedures/jobs causing full opens - #1		SPLIT/Jobs causing full opens - #1	
Hit count (PROCCNT)	Job/task name (TDEJOBNAME)			Thread ID (THREADID)	SQL statement (SQLSTMT)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	
11	JOB68	QPGMR	676156	000000000000061C		22,814,105	
9	JOB208	QPGMR	676777	0000000000000E6B		22,815,753	
8	JOB63	QPGMR	676473	0000000000000C70		22,814,778	
8	JOB68	QPGMR	676047	00000000000004FB		22,813,962	
8	JOB63	QPGMR	676664	0000000000000B06		22,815,356	
7	JOB114	QPGMR	677078	00000000000015B8		22,816,142	
7	JOB63	QPGMR	676301	000000000000002D		22,814,436	
7	JOB963	QPGMR	676520	00000000000006D5		22,815,069	
7	JOB68	QPGMR	676568	0000000000000B5F		22,815,133	
7	JOB68	QPGMR	676783	000000000000176F		22,815,761	
7	JOB63	QPGMR	676773	0000000000000FE4		22,815,748	
7	JOB68	QPGMR	676449	000000000000080C		22,814,657	
7	JOB68	QPGMR	676013	00000000000010F2		22,813,910	
7	JOB63	QPGMR	676761	0000000000001538		22,815,730	
7	JOB68	QPGMR	676807	0000000000000B4F		22,815,805	
7	JOB963	QPGMR	676036	00000000000011A6		22,813,944	
7	JOB68	QPGMR	676824	00000000000005D3		22,815,846	

Jobs causing full opens

29.7 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.

SPLIT/Programs/procedures causing full opens - #1			SPLIT/Programs/procedures/jobs causing full opens - #1			SPLIT/Jobs causing full opens - #1			SPLIT/Jobs causing full opens with 14 levels of program na... x				
Total call stacks (STACKCNT)	Job name and user name (JOB_AND_USER)	Generic job name (GENJOBNAME)	Ending interval (MAXINT)	Interval number (INTERVAL)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Program lib 2 (PGMLIB2)	Program name 2 (PGMNAME2)	Procedure name 2 (PROCNAME2)	Program lib 3 (PGMLIB3)	Program name 3 (PGMNAME3)	Procedure name 3 (PROCNAME3)	Program lib 4 (PGMLIB4)	Program name 4 (PGMNAME4)
32	JOB77 QPGMR	JOB77	27	5	22,819,779			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
20	JOB68 QPGMR	JOB68	25	7	22,819,430			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
13	JOB205 QPGMR	JOB205	30	1	22,773,210			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
12	JOB114 QPGMR	JOB114	25	7	22,819,611			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
11	JOB68 QPGMR	JOB68	20	7	22,817,601			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
10	JOB114 QPGMR	JOB114	25	7	22,817,016			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
10	JOB68 QPGMR	JOB68	26	9	22,819,603			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
10	JOB77 QPGMR	JOB77	26	8	22,819,755			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
9	JOB68 QPGMR	JOB68	26	8	22,817,601			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
9	JOB68 QPGMR	JOB68	19	5	22,817,981			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
9	JOB68 QPGMR	JOB68	24	8	22,818,953			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
8	JOB68 QPGMR	JOB68	22	10	22,818,155			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
8	JOB68 QPGMR	JOB68	26	3	22,817,167			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
7	JOB114 QPGMR	JOB114	23	10	22,819,164			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
7	JOB68 QPGMR	JOB68	24	6	22,817,811			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN
6	JOB114 QPGMR	JOB114	21	6	22,816,771			cblabbranch			aiuser_program_call_portal	QSYS	QDMCOPEN

Jobs causing full opens with 14 levels of program names

Tip: Double-click a row to get into the Record Quick View to make the display of the call stack information easier.

Record Quick View		
Call stack Waits Objects waited on Physical disk I/Os Logical DB IFS J9 JVM SQL		
Selected record(s): ☒ Hide all 0 or blank values		
Field	Description	Record 11
STACKCNT	Total call stacks	9
JOB_AND_USER	Job name and user name	JOB68 QPGMR
GENJOBNAME	Generic job name	JOB68
MAXINT	Ending interval	24
INTERVAL	Interval number	8
TASKCOUNT	Task count (uniquely identifies a task/thread)	22818953
PROCNAME2	Procedure name 2	cblabbranch
PROCNAME3	Procedure name 3	aiuser_program_call_portal
PGMLIB4	Program lib 4	QSYS
PGMNAME4	Program name 4	QDMCOPEN
PROCNAME5	Procedure name 5	cblabbranch
PROCNAME6	Procedure name 6	aiuser_program_call_portal
PGMLIB7	Program lib 7	QSYS
PGMNAME7	Program name 7	QRNXIO
PROCNAME7	Procedure name 7	_QRNX_OPEN
PGMLIB8	Program lib 8	LIB9
PGMNAME8	Program name 8	PGM23
PROCNAME8	Procedure name 8	P125
PGMLIB9	Program lib 9	LIB9
PGMNAME9	Program name 9	PGM23
PROCNAME9	Procedure name 9	P3322
PGMLIB10	Program lib 10	LIB9
PGMNAME10	Program name 10	PGM14
PROCNAME10	Procedure name 10	P2976
PGMLIB11	Program lib 11	LIB82
PGMNAME11	Program name 11	PGM1248
PROCNAME11	Procedure name 11	P3673

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.)

29.8 Total closes

This report counts the total closes captured in the collection.

SPLIT/Total closes - #1	
Hit count (PROCCNT)	7

Total closes

30 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+.

Note #2: Additional SQL metrics are found in the Job Watcher data in files QAPYJWTDE and QAPYJWPRC but contain inaccurate metrics (SQL high-level statements, SQL CPU, SQL PDIO and SQL statements in progress) and are therefore not graphed here.

Tip: SQL logical I/Os are found under the Logical I/O folder but also may be inaccurate due to SQL logicals and exceed the logical I/Os.

Report folder	Description
SQL statements executed	
SQL statements captured	
Full opens rates	
Full opens totals	
Pseudo opens rates	
Pseudo opens totals	
QZDA* connections	
SQL statements executed rankings	SQL statements executed rankings by job grouping
SQL statements captured rankings	SQL statements captured by Job Watcher rankings by job grouping
Full opens rates rankings	SQL vs native full opens rates rankings
Full opens totals rankings	SQL vs native full opens rankings
Pseudo opens rates rankings	SQL-based pseudo opens rates rankings by job grouping
Pseudo opens totals rankings	SQL-based pseudo opens rankings
QZDA* connection rankings	QZDA* connection rankings by job grouping
Detail reports	Additional reports in table format

SQL Folder

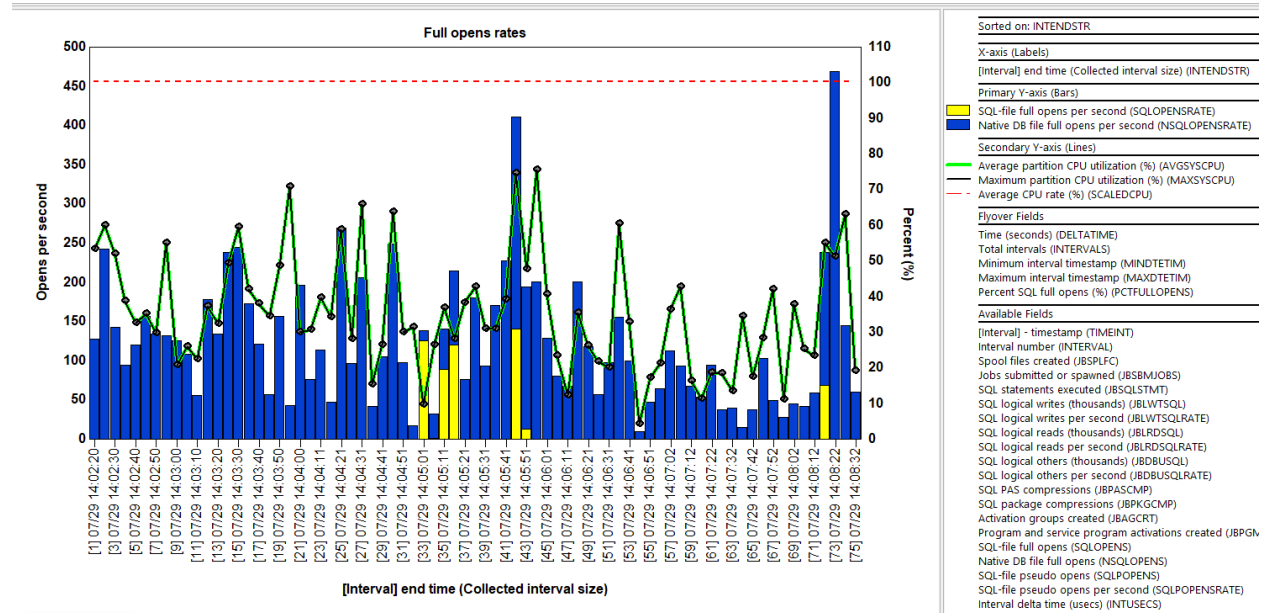
30.1 SQL statements executed

This graph shows a total count of SQL statements executed across all jobs on the system over time.

30.3 Full opens rates

This graph displays the 2 types of full opens available over time as a rate per second:

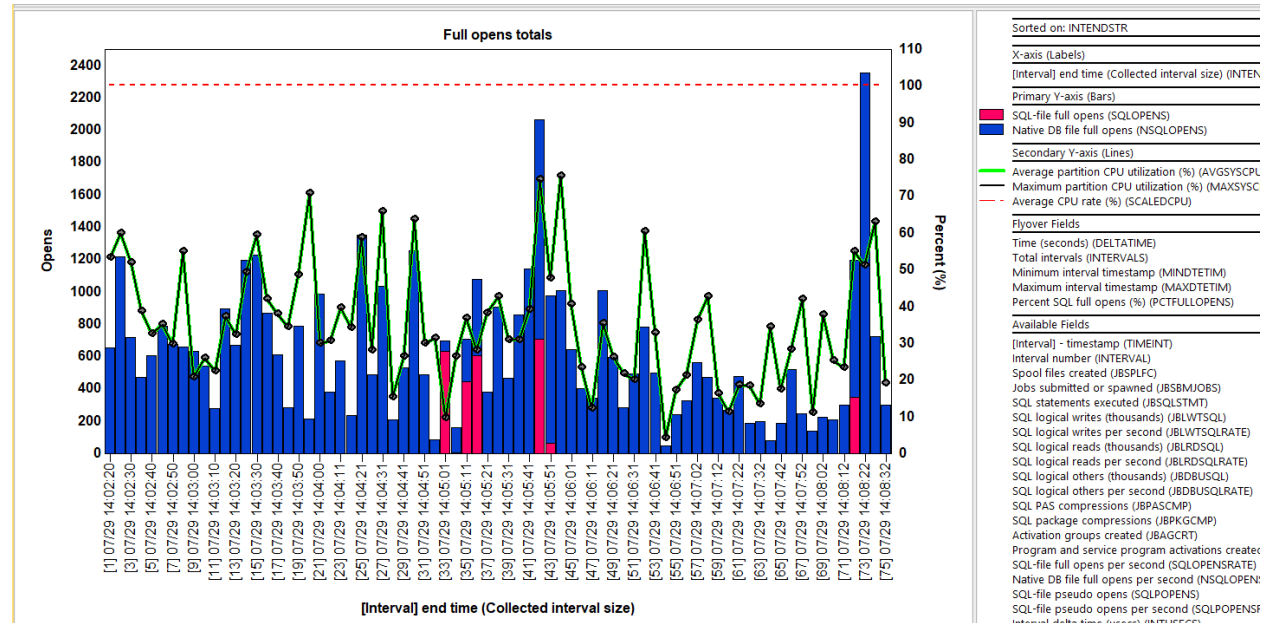
- 1) SQL-file full opens
- 2) Native DB file full opens



Full opens rates

30.4 Full opens totals

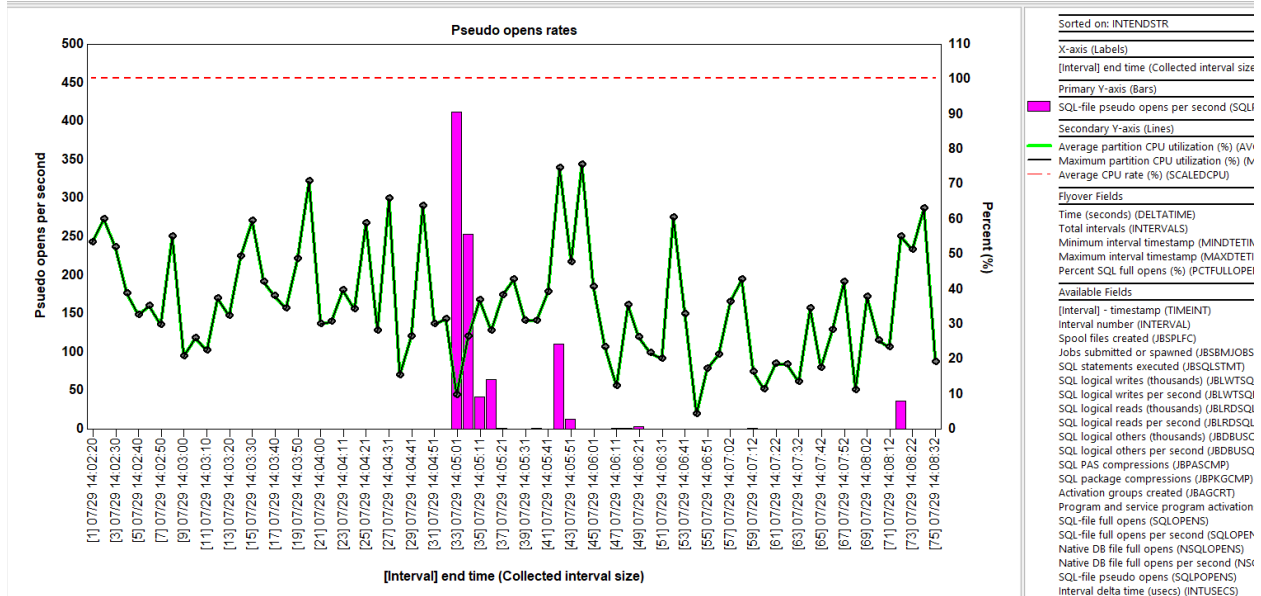
This graph displays the 2 types of full opens available over time.



Full opens totals

30.5 Pseudo opens rates

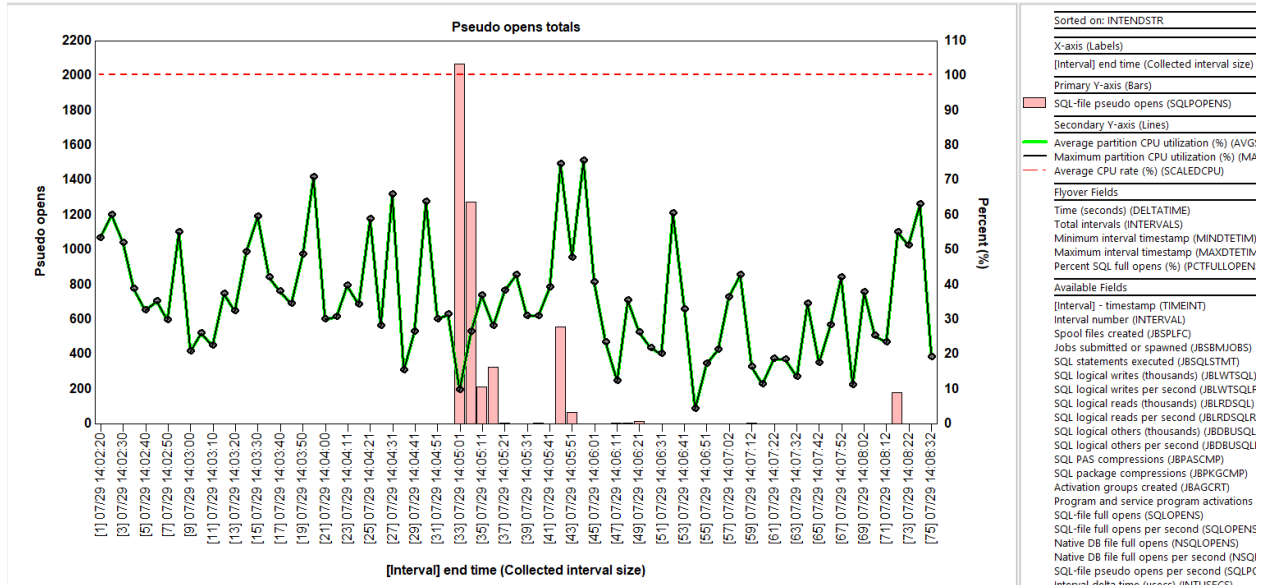
This graph displays SQL pseudo opens as a rate per second.



Pseudo opens rates

30.6 Pseudo opens totals

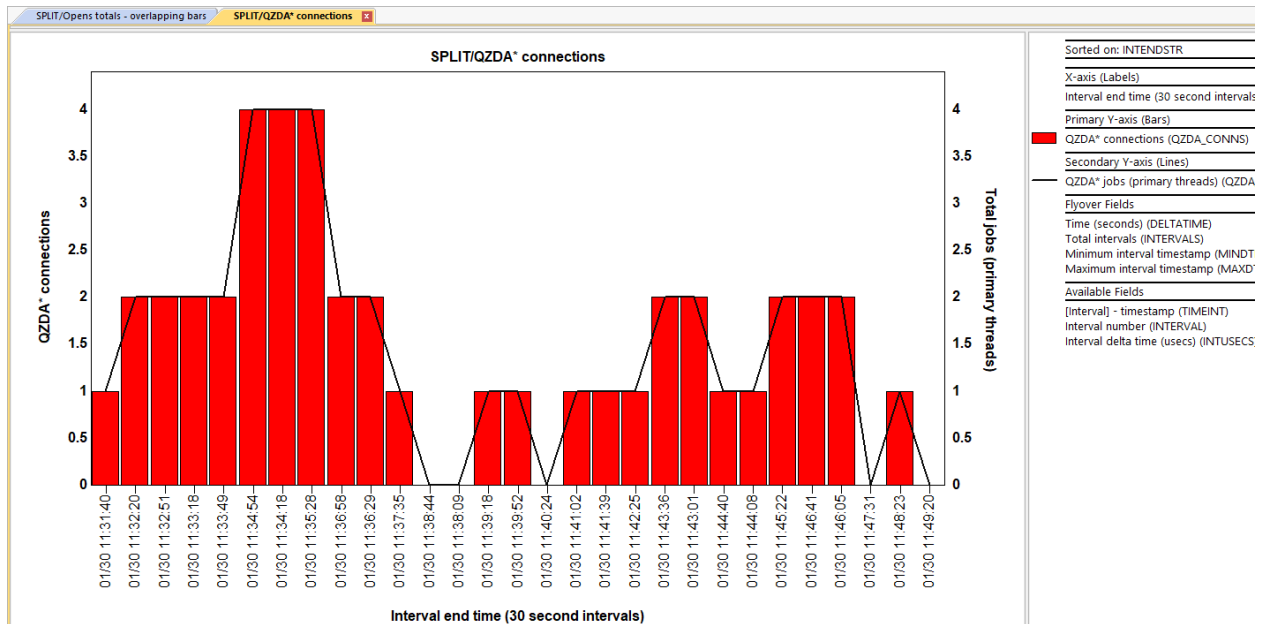
This graph displays SQL pseudo opens over time.



Pseudo opens totals

30.7 QZDA* connections

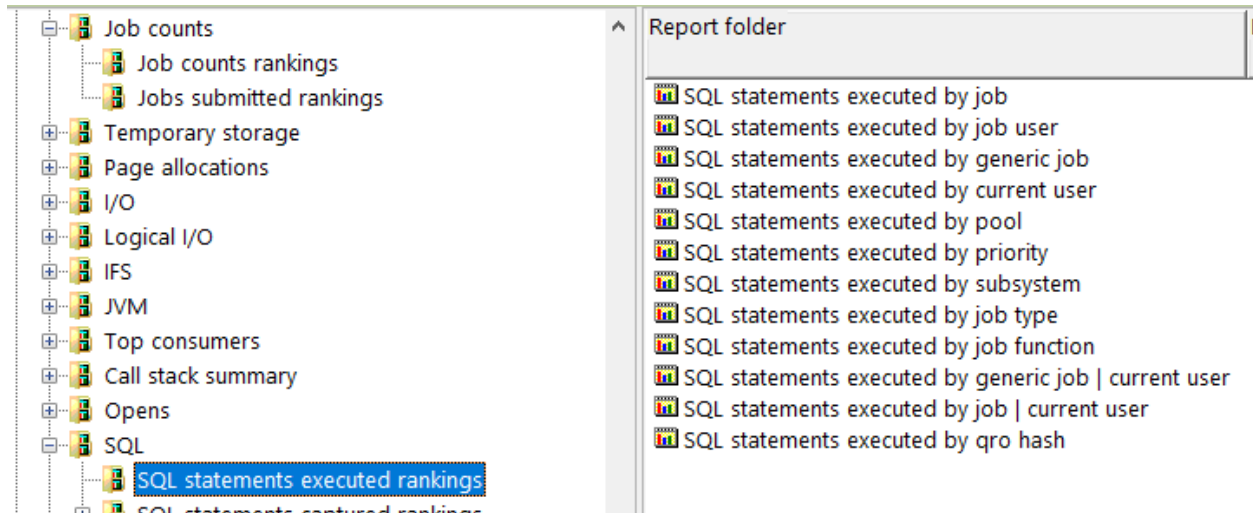
This graph displays the total QZDA* connections detected over time.



QZDA* connections

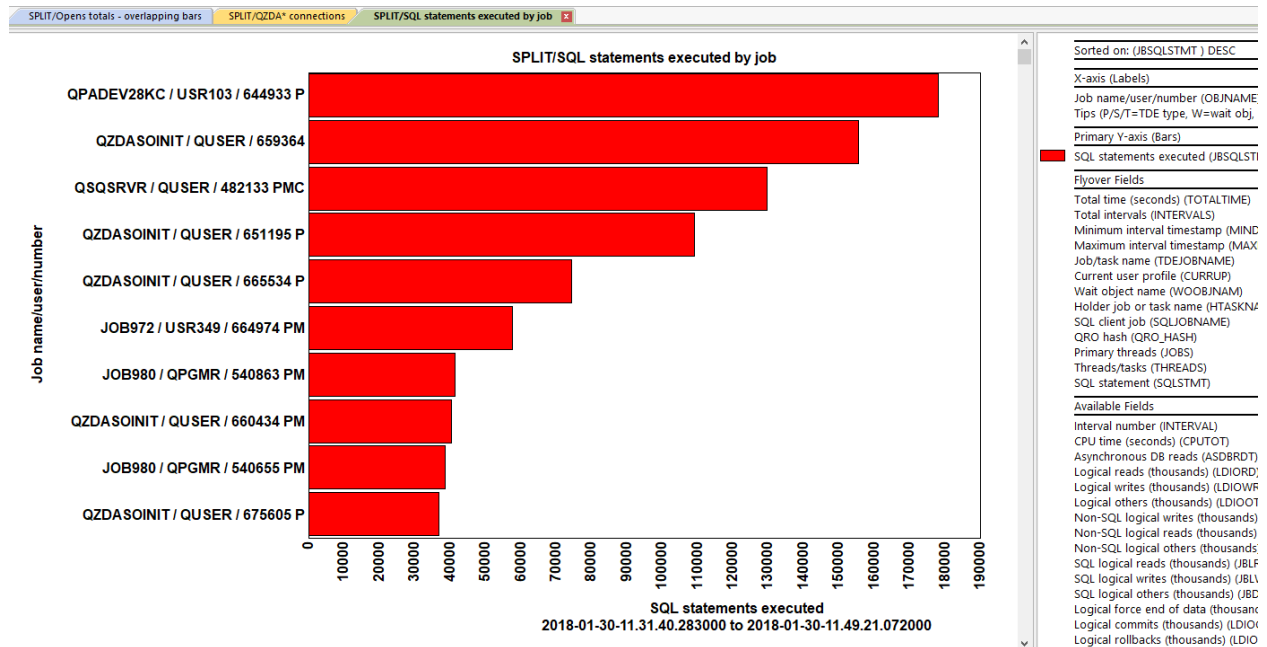
30.8 SQL statements executed rankings

These graphs display the total SQL statements executed ranked by various job groupings.



SQL -> SQL statements executed rankings

An example follows:



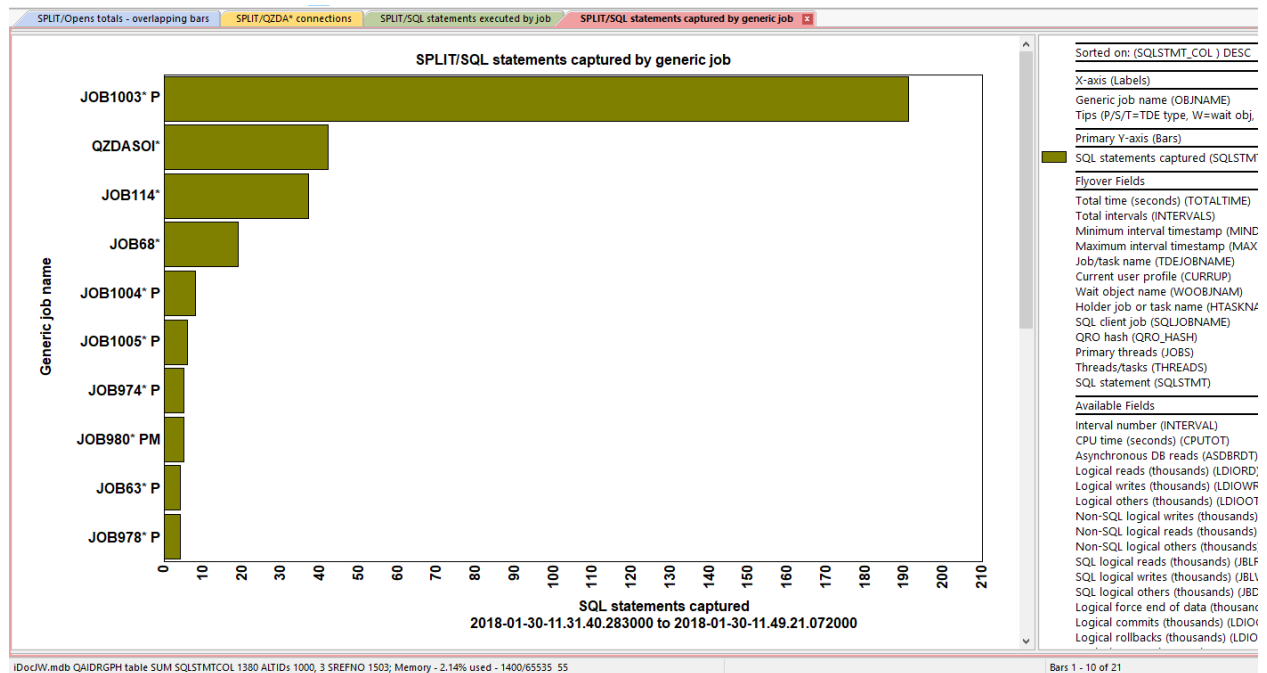
SQL statements executed by job

30.9 SQL statements captured rankings

These graphs display the **unique** SQL statements captured by Job Watcher and ranked by various job groupings.

Note: This graph will only appear if QAPYJWSQL contains data.

An example follows:

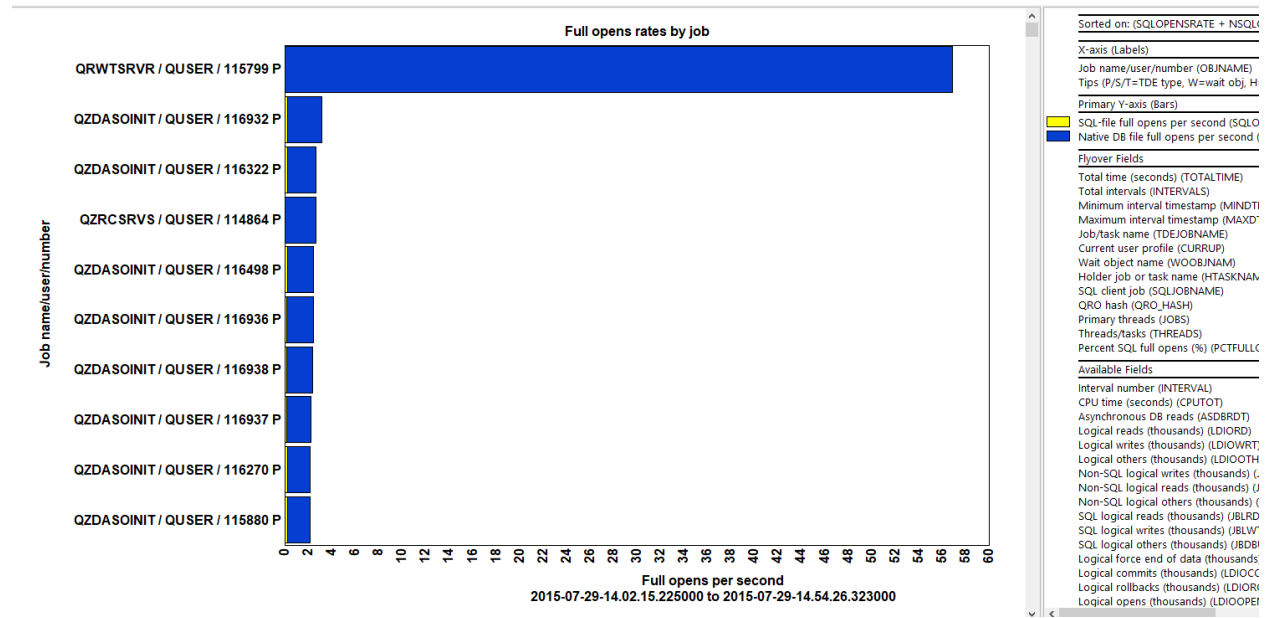


SQL statements captured by generic job

30.10 Full opens rates rankings

These graphs display the full opens (native and SQL) ranked by the one of the job groupings.

An example follows:

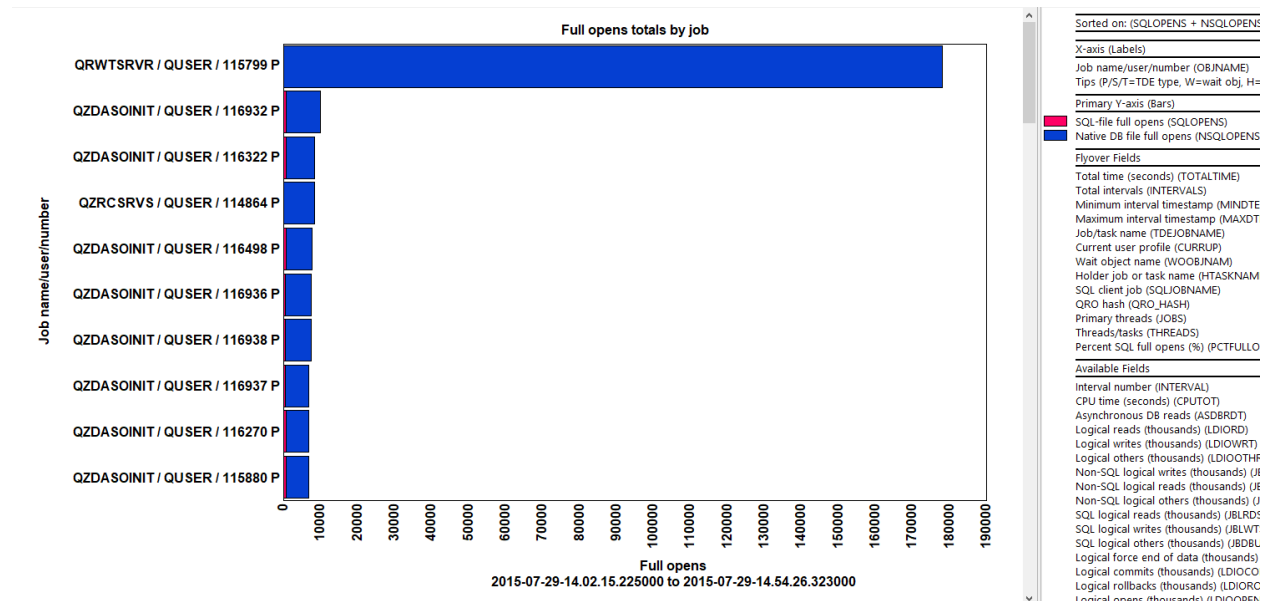


Full opens rates by job

30.11 Full opens totals rankings

These graphs display the full opens (native and SQL) ranked by the one of the job groupings.

An example follows:

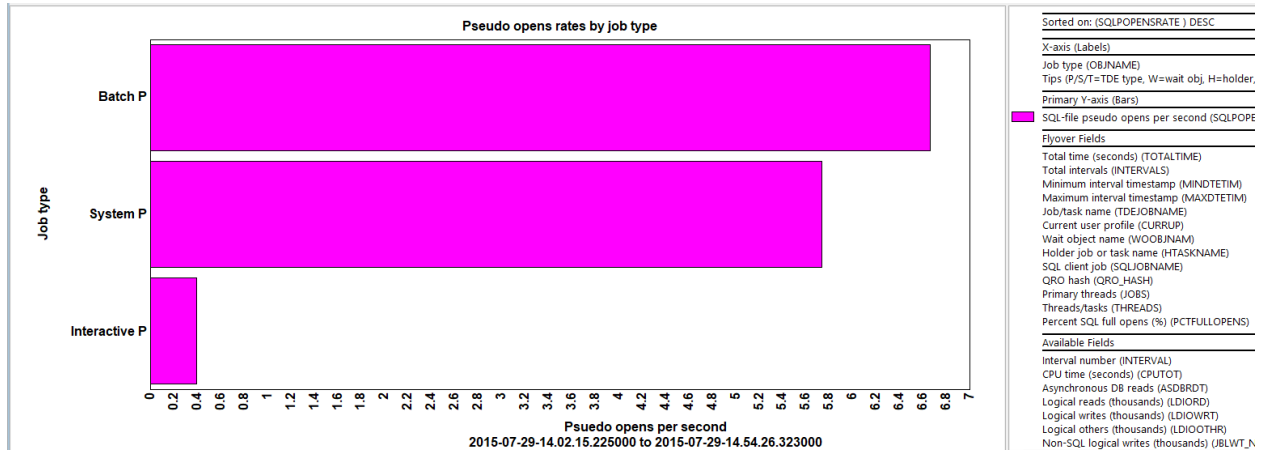


Full opens totals by job

30.12 Psuedo opens rates rankings

This graph shows the SQL pseudo opens as a rate per second and ranked by one of the job groupings.

An example follows:

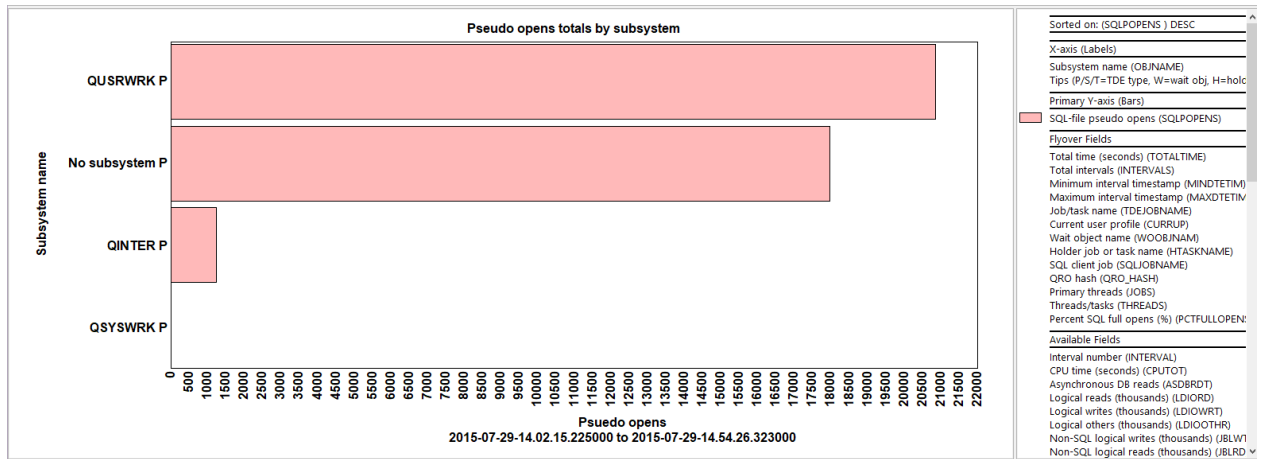


Pseudo opens rates by job type

30.13 Psuedo opens totals rankings

This graph shows the SQL pseudo opens ranked by one of the job groupings.

An example follows:

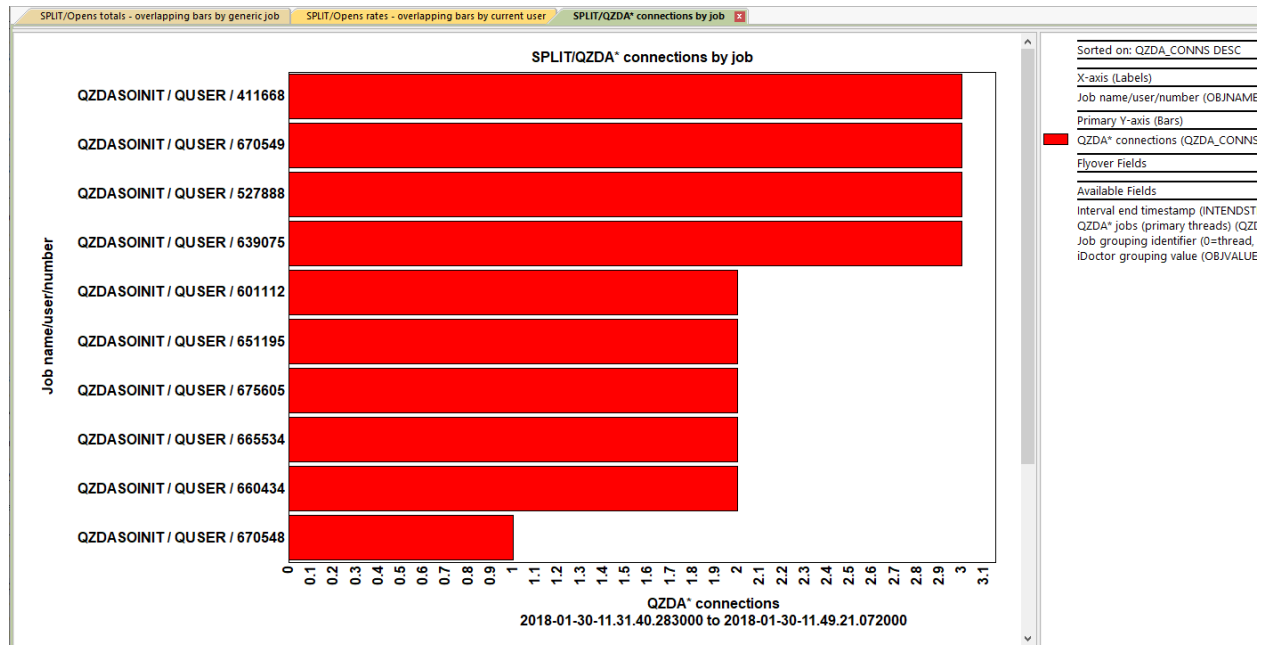


Pseudo opens totals by subsystem

30.14 QZDA* connections by job

The QZDA connections rankings subfolder contains a single graph that displays the total unique QZDA* connections by job (taskcount.)

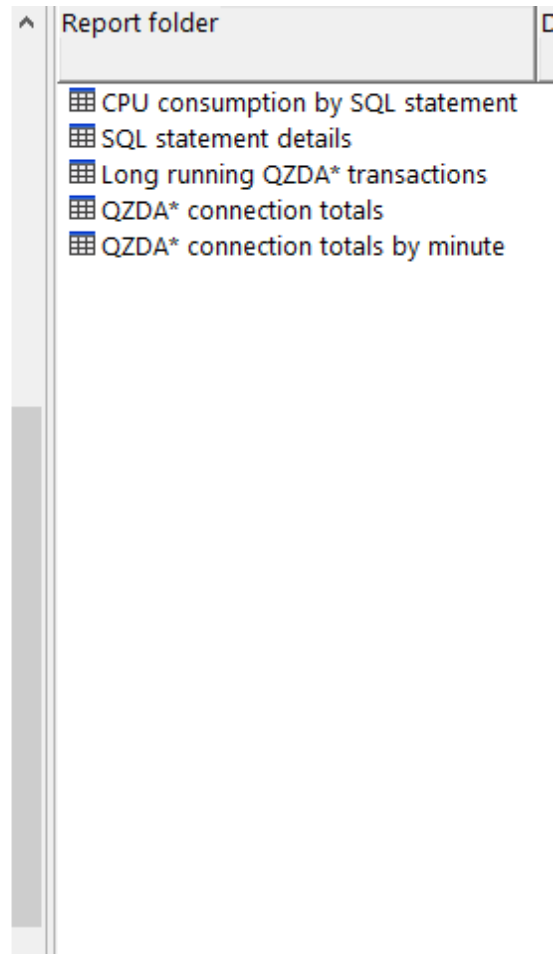
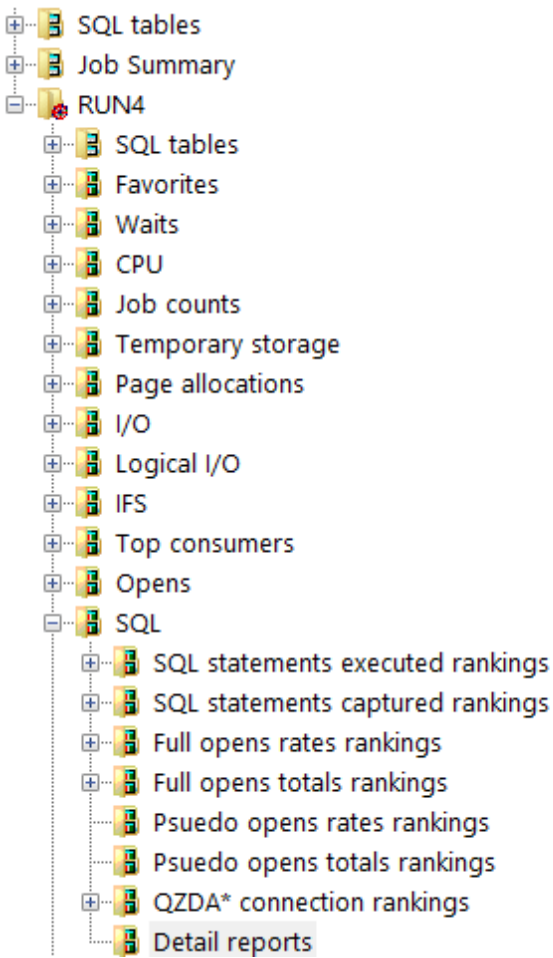
This graph shows the jobs with the highest total unique QZDA connections (servicing potentially different users.)



QZDA connections by job*

30.15 Detail reports

This folder contains additional reports intended for advanced users.



SQL -> Detail reports folder

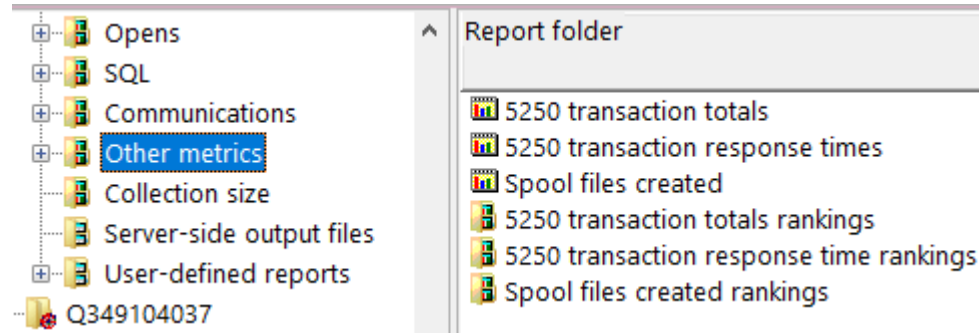
31 Communications

These graphs contain metrics related to the TCP sockets found in the data and are intended for advanced users so are not documented.

Note: This folder only appears if the socket metrics were captured.

32 Other metrics

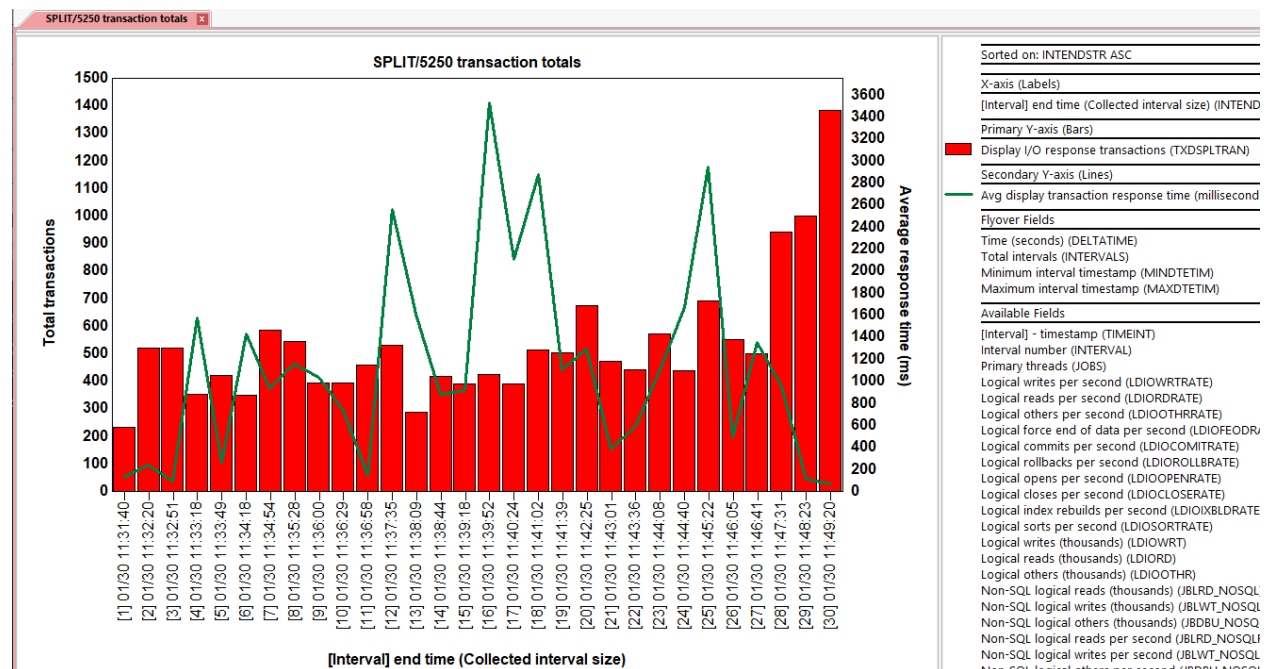
This folder contains some additional graphs covering other statistics not found in the previous graphs. These statistics currently include spool file creations and 5250 transactions.



Other metrics folder

32.1 5250 transaction totals

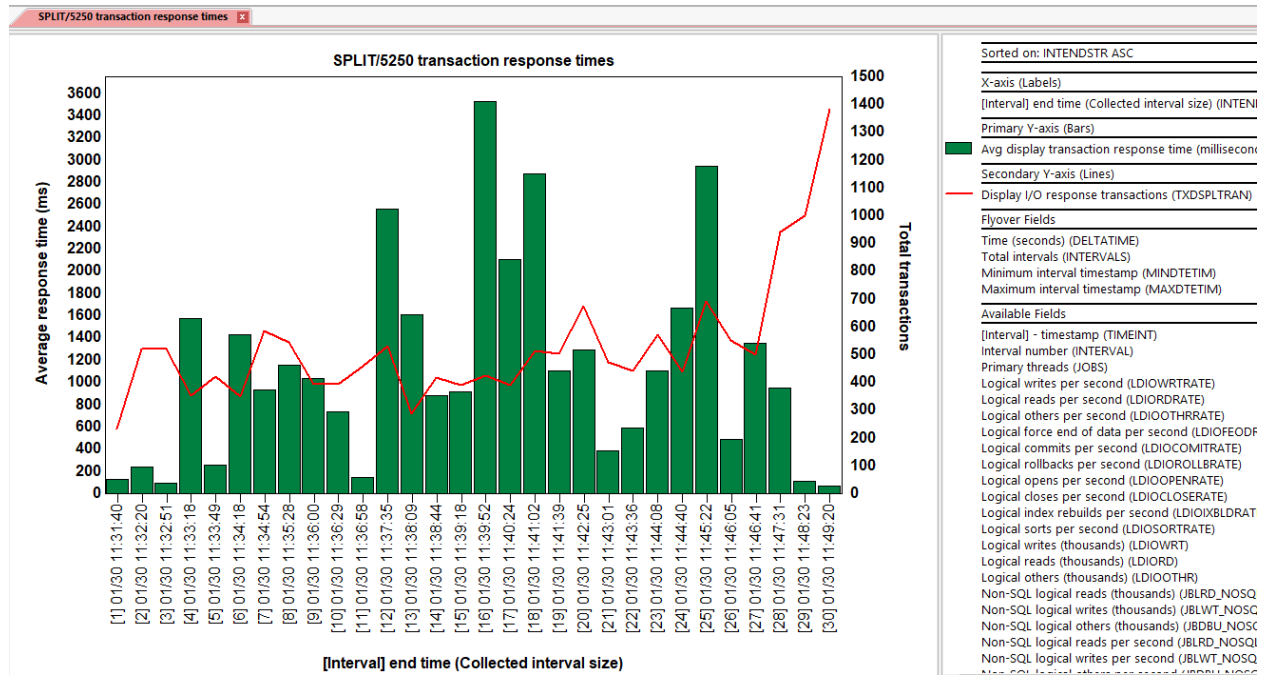
This graph displays the total 5250 display transactions that occurred during the collection along with their average response times (in milliseconds) on the Y2-axis. These are recorded for interactive jobs only. The transaction starts on detection of enter from the workstation; the transaction ends when the keyboard is unlocked.



5250 transaction totals

32.2 5250 transaction response times

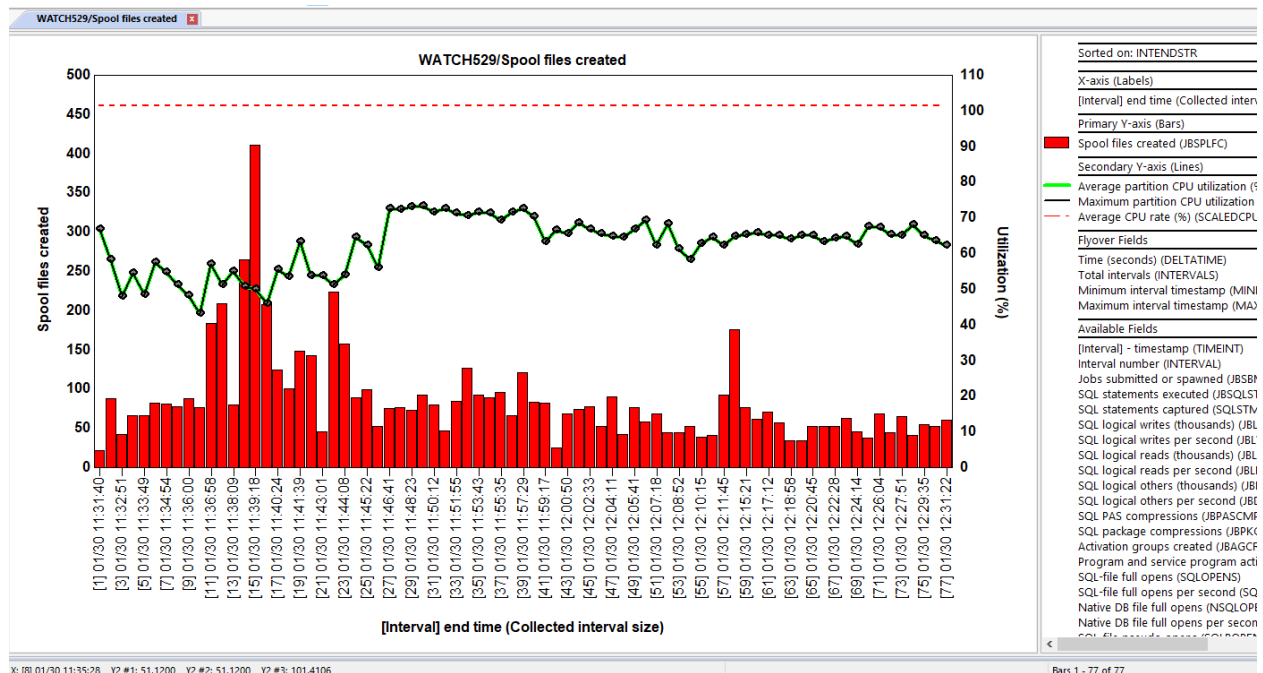
This graph is identical to the previous graph except the Y1 and Y2 axes are flipped around.



5250 transaction response times

32.3 Spool files created

This graph displays the number of spool files created over time during the collection. The Y2-axis contains CPU utilization metrics.



X: 01/30 11:35:28 Y2 #1: 51:1200 Y2 #2: 51:1200 Y2 #3: 101:4106

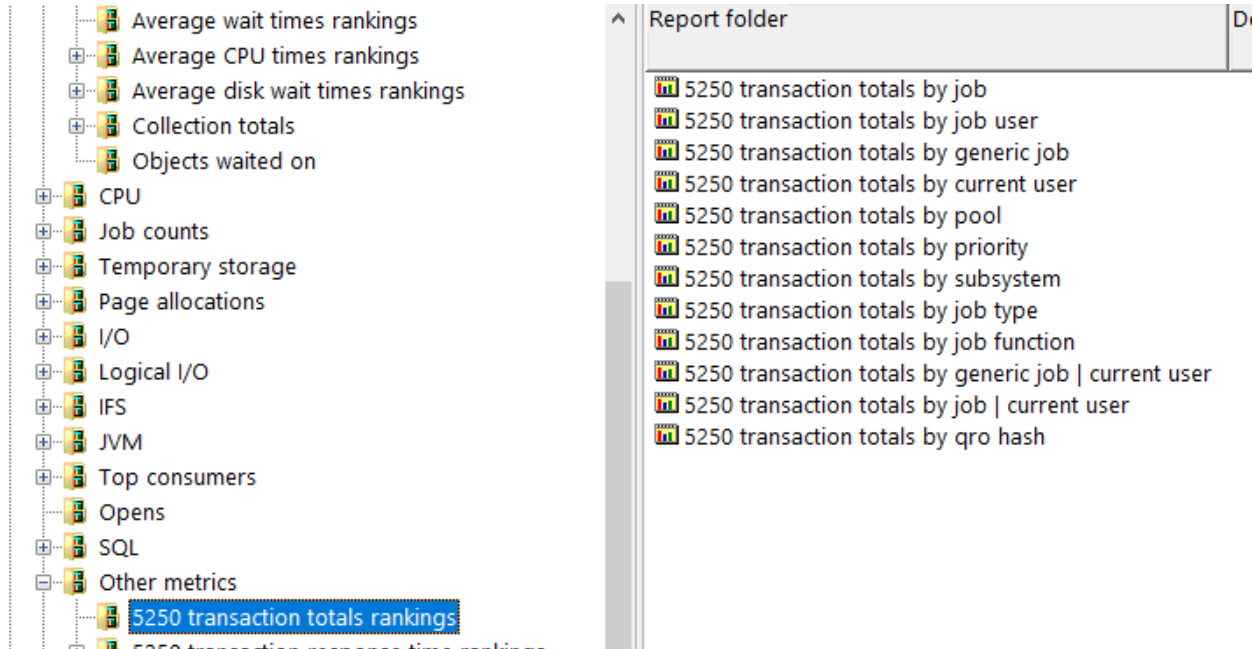
Bars 1 - 77 of 77

Spool files created

32.4 5250 transaction totals rankings

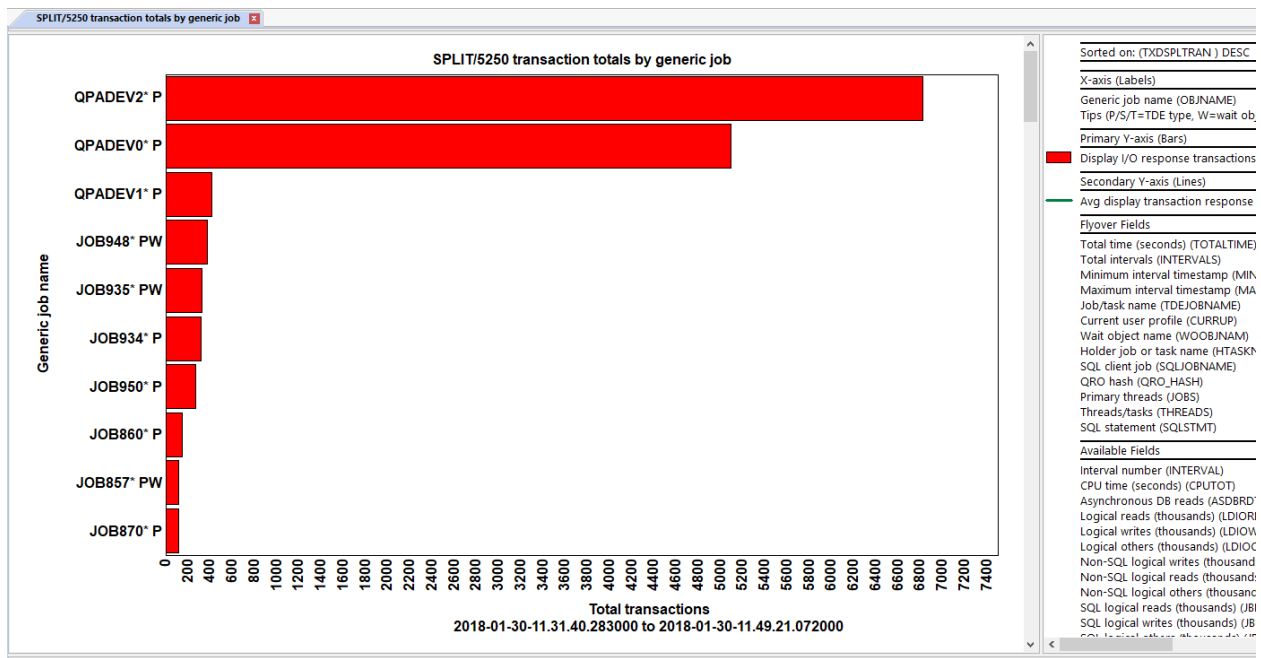
This folder contains a set of ranking graphs for 5250 display transactions totals. These allow you to rank the number of transactions that occurred by several different types of job groupings.

This graph also includes an optional Y2-axis that shows average response times. Use the **toggle graph format** toolbar button to hide or show the Y2-axis.



Other metrics -> 5250 transaction totals rankings

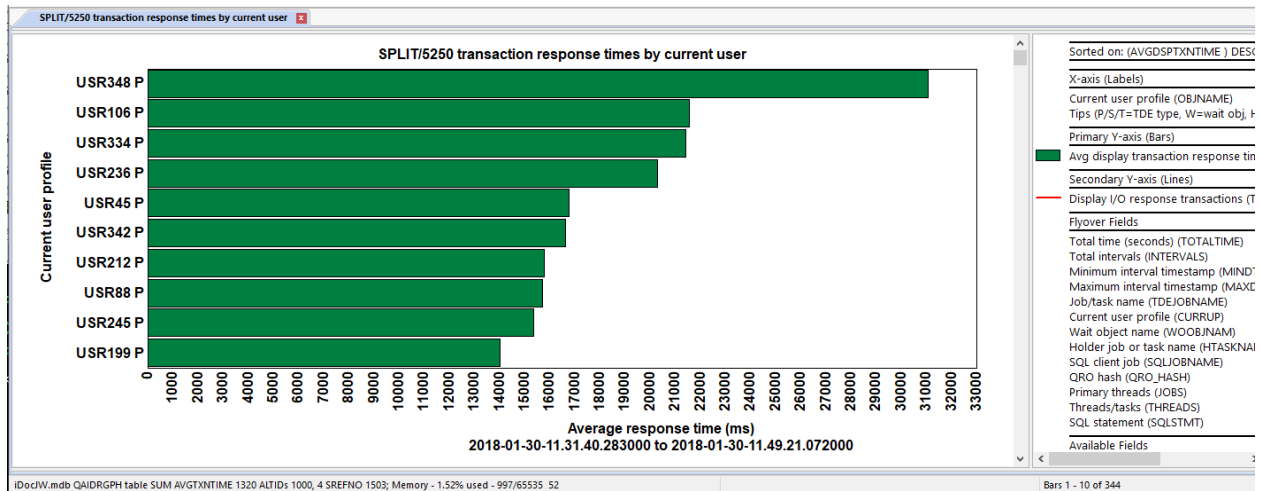
An example follows:



5250 transaction totals by generic job

32.5 5250 transaction response times rankings

This graph is identical to the previous graph except the Y1 and Y2 axes are flipped around.

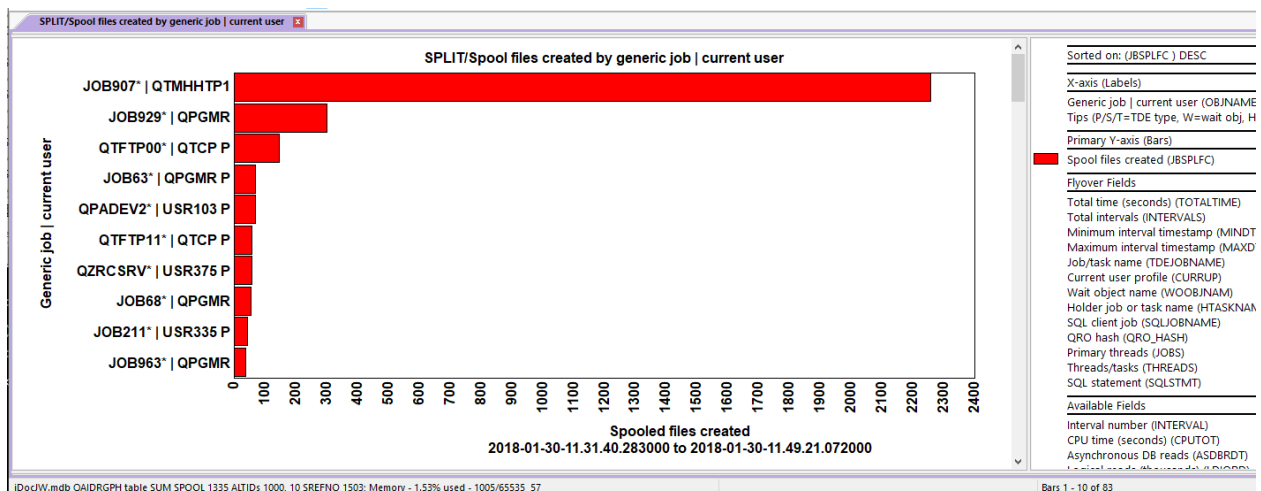


5250 transaction response times by current user

32.6 Spool files created rankings

These graphs rank the number of spool files created using various job groupings.

An example is shown below:



Spool files created by generic job | current user

33 Interval Summary Interface

The interval summary interface is 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.

33.1 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.

Tip: This panel is shown when coming from a graph and offers the options to hide/show the X, Y1, Y2 parts of the graph.

Using the hide all 0 or blank values option is recommended. That is like how the graph legend works as well.

The screenshot shows the 'Interval Summary: Library lbmdk2, Collection Split - #1' window. The 'Quick View' tab is selected. The 'Selected point details' section has checkboxes for 'Hide all 0 or blank values' (checked), 'X' (checked), 'Y1' (checked), 'Y2' (checked), 'Flyovers' (checked), and 'Other' (checked). The main table displays the following data:

Description	Value_4
X-axis:	
[Interval] end time (Collected interval size)	2018-01-30-11.33.49.733000
Primary Y-axis (Y1):	
Clicked Machine level gate serialization (seconds)	5768.4202
Dispatched CPU (seconds)	1918.1810
CPU queueing - remainder (seconds)	36.7291
CPU queueing - workload capping delay (seconds)	.0126
Disk page faults (seconds)	260.7024
Disk non fault reads (seconds)	222.9426
Disk space usage contention (seconds)	.0180
Disk writes (seconds)	281.6200
Disk other (seconds)	601.9860
Journaling (seconds)	818.4982
Seize contention (seconds)	110.4101
Database record lock contention (seconds)	344.0350
Object lock contention (seconds)	107.2275
Abnormal contention (seconds)	86.3893
Synchronization token contention (seconds)	.0882
Secondary Y-axis (Y2):	
Average partition CPU utilization (%)	48.59
Maximum partition CPU utilization (%)	48.59
Average collection CPU utilization	46.74
VCPU delays as a percentage of Dispatched CPU (...)	.0015
Average CPU rate (%)	101.4106
Flyover fields:	
Total active threads/tasks	11889
Total idle threads/tasks	30159

At the bottom, there are buttons for 'Copy', 'Copy URL', 'OK', and 'Cancel'.

Interval Summary – Quick View

33.2 General section

This section describes the common part of the interface found on all tabs except Quick View, SQL and Columns. An example follows:

General:			
Threads/tasks using CPU:	11889	Interval:	5 ◀ ▶
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

General section

Some of the less obvious fields are described below:

Option	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.

33.3 Waits

The Waits tab by default shows the top 35 jobs in the interval that experienced wait time in the wait bucket that was clicked on when this interface was opened. Use the **Max** text box to change this value if desired.

In the example, below only jobs that had some machine level gate time are included. The data is sorted in descending order by the bucket indicated in the filter by drop down list.

Interval Summary: Library Ibmdbk2, Collection Split - #1 x SPLIT/Collection overview time signature

IFS		Other statistics			SQL		Columns		
Quick View	Waits	Wait bucket totals	Objects waited on	Holders	SQL statistics	Bad Current Waits	Situations	Physical disk I/Os	Logical DB

General:

Threads/tasks using CPU: 11889 Interval: 5

Threads/tasks idle: 30159 CPU utilization: 48.59%

Threads/tasks waiting on objects: 3167 CPU time: 11.294 minutes

Threads/tasks with holder identified: 31 Interval duration: 30.921 seconds

Temp storage job allocations (GB): 316.3921 Interval end: 2018-01-30-11.33.49.733000

Threads/tasks: Max ☐ Exclude jobs not in current wait Filter by: 14 - Machine level gate serialization

Job name/user/number: thread ID (OBJNAME)	Current wait duration (usecs) (CURRWTDUR)	Current or last wait bucket (BLOCKBCKT)	Current wait enum and description (WAITINFO)	Dispatched CPU (seconds) (TIME01)	Dispatched CPU counts per second (COUNTR01)	CPU queueing (seconds) (TIME02)
JOB63 / QPGMR / 676029: 00000876	11,882,005	14	(2) Qu gate - high perfor...	.0080	.6890	.000!
JOB983 / USR153 / 675939: 0000085E	12,493,482	14	(2) Qu gate - high perfor...	.0259	.8860	.000!
JOB1004 / QPGMR / 540874: 00000207	12,041,464	14	(2) Qu gate - high perfor...	.0021	.6323	.000!
JOB1005 / QPGMR / 540895: 000008A1	12,044,394	14	(2) Qu gate - high perfor...	.0369	.4997	.000!
JOB963 / QPGMR / 541347: 0000054A	12,034,755	14	(2) Qu gate - high perfor...	.0096	.5846	.000!
JOB964 / QPGMR / 540406: 00005380	12,031,272	14	(2) Qu gate - high perfor...	.0171	.2439	.001!
JOB1007 / QPGMR / 540620: 00000C5E	12,049,513	14	(2) Qu gate - high perfor...	.1122	.2213	.001!
JOB243 / USR399 / 676017: 000009CA	11,880,827	14	(2) Qu gate - high perfor...	.3700	.6890	.000!
JOB963 / QPGMR / 541418: 00001057	12,027,956	14	(2) Qu gate - high perfor...	.0314	.2079	.002!
JOB68 / QPGMR / 676047: 000004FB	11,860,923	14	(2) Qu gate - high perfor...	.4015	.5958	.000!
JOB63 / QPGMR / 675966: 00000C87	11,884,355	14	(2) Qu gate - high perfor...	.1674	.2819	.001!
JOB963 / QPGMR / 676044: 000006B4	11,885,493	14	(2) Qu gate - high perfor...	.3044	.2845	.001!
JOB243 / USR399 / 676016: 0000079F	11,886,350	14	(2) Qu gate - high perfor...	.4116	.5255	.000!
JOB63 / QPGMR / 676014: 0000009E	11,895,585	14	(2) Qu gate - high perfor...	.4566	.6893	.000!
JOB973 / USR349 / 664515: 00004FD1	16,522,051	14	(2) Qu gate - high perfor...	.4789	.7381	.000!

1 - 16 of 35

Interval Summary - Waits

Some of the less obvious fields are described below:

Max	Use this text box to increase/decrease the number of jobs shown in the list. Type a new value and press enter to rerun the query.
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.

Tip: Users can right-click a job in the list for additional options:

Threads/tasks: Max ☐ Exclude jobs not in current wait Filter by: 14 - Machine level gate serialization

Job name/user/number: thread ID (OBJNAME)	Current wait duration (usecs) (CURRWT DUR)	Current or last wait bucket (BLOCKBCKT)	Current wait enum and descri (WAITINFO)	Dispatched CPU (seconds) (TIME01)
JOB63 / QPGMR / 676029: 00000876	11,882,005	14	(2) On gate - high perfor	0080
JOB983 / USR153 / 675939: 0000085E	Thread wait time signature for JOB63 / QPGMR / 676029: 00000876			
JOB1004 / QPGMR / 540874: 00000207	Selected Threads			>
JOB1005 / QPGMR / 540895: 000008A1	Rankings filtered by selected Threads			>
JOB963 / QPGMR / 541347: 0000054A	Display call stack			>
JOB964 / QPGMR / 540406: 00005380	All graphs/reports			>
JOB1007 / QPGMR / 540620: 00000C5E				

Drill-down example from Interval Summary – Waits

33.4 Wait bucket totals

This interface shows all wait buckets (except reserved ones) in the list with several metrics as follows:

Results:

Wait bucket number (BUCKET)	Wait bucket description (BUCKETDESC)	Total bucket time (seconds) (TOTBUCKETTIME)	Total occurrences (BUCKETCNT)	Average wait time (seconds) (AVGWAIT)	Contributing jobs/tasks (BUCKETJOBS)
1	Dispatched CPU	1,918.1810	2,523,995	.000759	11,803
2	CPU queueing	36.7418	2,523,994	.000014	11,799
4	Other waits	294,488.5323	1,016,926	.289586	11,022
5	Disk page faults	260.7024	115,944	.002248	1,990
6	Disk non fault reads	222.9426	79,835	.002792	939
7	Disk space usage contention	.0180	1,955	.000009	690
9	Disk writes	281.6200	255,744	.001101	1,858
10	Disk other	601.9860	73,352	.008206	596
11	Journaling	818.4982	475,740	.001720	627
12	Semaphore contention	10,384.6604	17	610.862377	335
13	Mutex contention	1.3657	172	.007939	132
14	Machine level gate serialization	5,768.4202	115,293	.050032	5,252
15	Seize contention	110.4101	28,638	.003855	1,610
16	Database record lock contention	344.0350	6,883	.049983	554
17	Object lock contention	107.2275	368	.291379	151
25	Socket receives	16,590.1735	3,438	4.825530	846

Interval Summary – Wait bucket totals

33.5 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. Sometimes the current job may be waiting for another job to release its lock on the object.

An example of this interface is:

Quick View Waits Wait bucket totals **Objects waited on** Holders SQL statistics Bad Current Waits Situations Physical disk I/Os Logical DB IFS Other statistics SQL Columns

General:

Threads/tasks using CPU: 11889 Interval: 5
 Threads/tasks idle: 30159 CPU utilization: 48.59%
 Threads/tasks waiting on objects: 3167 CPU time: 11.294 minutes
 Threads/tasks with holder identified: 31 Interval duration: 30.921 seconds
 Temp storage job allocations (GB): 316.3921 Interval end: 2018-01-30-11.33.49.733000

Threads waiting on objects: Max 35 ☐ Include segments Filter by: 14 - Machine level gate serialization

Job name/user/number: thread ID (OBJNAME)	Current wait duration (usecs) (CURRWT DUR)	Current or last wait bucket (BLOCKBCKT)	Current wait enum and descri (WAITINFO)	Wait object library (WOBJLIB)	Wait object name (WOBJNAM)	Object type and description (OBJINFO)	Segment type and descri (SEGINFO)
JOB10 / USR22 / 675240: 00000C79	18,727,327	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB221 / USR5 / 118137: 00000001	16,846,784	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB221 / USR5 / 118106: 00000001	16,840,817	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB221 / USR5 / 118133: 00000001	16,818,154	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB221 / USR5 / 050255: 00000361	16,805,877	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB221 / USR5 / 118094: 00000001	16,801,618	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB922 / QPGMR / 540647: 00005711	16,689,016	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB973 / USR349 / 664515: 00004FD1	16,522,051	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB877 / USR347 / 670538: 000008A0	16,456,691	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
DbpmServer141: 3094	16,161,887	14	(2) Qu gate - high perfor...		PORDER PORDER	0C90-DB2 ACCESS PATH	0001-BASE MI SYSTEM C
JOB243 / USR399 / 675543: 00000A05	16,158,302	14	(2) Qu gate - high perfor...		PORDER PORDER	0C90-DB2 ACCESS PATH	001C-MACHINE INDEX F
JOB221 / USR5 / 118088: 00000001	15,356,341	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB221 / USR5 / 118129: 00000001	15,308,851	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB970 / QPGMR / 540175: 000000F1	12,694,286	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV
JOB70 / QPGMR / 540541: 000012DC	12,608,132	14	(2) Qu gate - high perfor...	LIB8	QJRDWH	0901-JOURNAL	20C5-JOURNAL RESERV

1 - 16 of 35

Interval Summary - Objects waited on

Tip: If you want to include segments waited on in the output, then check the "Include segments" 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 filter by 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
Job name/user/number: Thread ID	This is the complete job name/user/number: thread ID or task name that is waiting on the object.
Current wait duration (usecs)	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 or last wait bucket	Lists the wait bucket
Current wait enum and 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 name	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. Tip: If a Holder is present, a Holder menu will appear on a right-click.

33.6 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.

SPLIT/Collection overview time signature Interval Summary: Library ibmdk2, Collection Split - #1 SPLIT/Holder chase for interval 5 - #1

Quick View Waits Wait bucket totals Objects waited on **Holders** SQL statistics Bad Current Waits Situations Physical disk I/Os Logical DB IFS Other statistics SQL Columns

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

Holders (at end of interval): Max Filter by: 14 - Machine level gate serialization

Job name/user/number: thread ID (OBJNAME)	Holder job or task name (HTASKNAME)	Current wait duration (usecs) (CURRWTDUR)	Current or last wait bucket (BLOCKBCKT)	Current wait enum and description (WAITINFO)	Wait object library (WOBJLIB)	Wait object name (WOBJNAME)
JOB243 / USR399 / 675543: 00000A05	JOB963 / QPGMR / 541372: 00000F25	16,158,302	14	(2) Qu gate - high perfor...		PORDER
JOB1003 / QPGMR / 540220: 0000014C	JOB63 / QPGMR / 649820: 00000E07	12,112,111	14	(2) Qu gate - high perfor...		WOACT6
JOB964 / QPGMR / 542070: 000013D4	JOB63 / QPGMR / 649820: 00000E07	12,022,483	14	(2) Qu gate - high perfor...		WOACT6
QPADEV046J / USR295 / 659999: 0000116F	JOB63 / QPGMR / 649820: 00000E07	11,950,508	14	(2) Qu gate - high perfor...		WOACT6
JOB978 / QPGMR / 540160: 00001378	JOB963 / QPGMR / 541372: 00000F25	7,947,184	14	(2) Qu gate - high perfor...		PORDER
JOB978 / QPGMR / 540169: 000007B0	JOB963 / QPGMR / 541372: 00000F25	7,836,309	14	(2) Qu gate - high perfor...		PORDER
JOB876 / USR336 / 670537: 0000184F	JOB63 / QPGMR / 666733: 00001447	14,240	14	(2) Qu gate - high perfor...		PKWORK

Interval Summary - Holders

33.7 SQL statistics

This interface provides summarized metrics relating to SQL for the entire collection in the specified time interval. **Note:** Additional metrics exist in the Job Watcher data but are not shown here because of problems with the data in the OS.

Quick View
Waits
Wait bucket totals
Objects waited on
Holders
SQL statistics
Bad Current Waits
Situations
Physical disk I/C

General:

Threads/tasks using CPU:
11889

Interval:

Threads/tasks idle:
30159

CPU utilization:
48.59%

Threads/tasks waiting on objects:
3167

CPU time:
11.294 minutes

Threads/tasks with holder identified:
31

Interval duration:
30.921 seconds

Temp storage job allocations (GB):
316.3921

Interval end:
2018-01-30-11.33.49.733000

SQL statistics:

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
SQL logical reads	372,730	12,034....
Native DB file full opens	129,115	4,168.65
SQL statements executed	41,249	1,331.78
Fully opened SQL cursors	2,685	86.68
Pseudo closed SQL cursors	1,900	61.34
SQL logical updates and deletes	147	4.74
SQL logical writes	70	2.26
SQL PAS compressions	0	0
SQL package compressions	0	0
SQL-file full opens	0	0
SQL-file pseudo opens	0	0

Interval Summary – SQL statistics

33.8 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!

For 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 are the same as at 7.1 but also includes #20.

SPLIT/Collection overview time signature Interval Summary: Library Ibmdbk2, Collection Split - #1

Quick View Waits Wait bucket totals Objects waited on Holders SQL statistics Bad Current Waits Situations Physical disk I/Os Logical DB IFS Other statistics SQL Columns

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

Results: Max 35

Job name/user/number: thread ID (OBJNAME)	Current wait time (microseconds) (CURRWTDUR)	Current wait bucket (BLOCKBCKT)	Wait bucket description (BUCKETDESC)	Current wait (enum) descriptive (WAITINFO)	Task count (uniquely identifies a task/thread) (TASKCOUNT)
JOB89 / USR371 / 670690: 0000047E	19	32	Abnormal contention	(40) QuGateB, generic Qu...	22,804,727
LDDPST: 22728613	10,005,620	32	Abnormal contention	(40) QuGateB, generic Qu...	22,728,613
JOB127 / QPGMR / 541150: 00000DAC	336,558	17	Object lock contention	(117) Lock: exclusive no re...	22,589,425
Taskcount 21966962	169,606,403,4...	17	Object lock contention	(117) Lock: exclusive no re...	21,966,962
QDBSRV46 / QSYS / 115135: 00000001	1,943,672	17	Object lock contention	(114) Lock: shared read o...	6,146
JOB1005 / QPGMR / 540890: 00000AA7	4,300,068	16	Database record lock contention	(123) DB record lock: weak	22,587,991
JOB1007 / QPGMR / 541087: 000012B6	19,521,913	16	Database record lock contention	(111) DB record lock: upd...	22,589,276
JOB983 / USR153 / 669527: 00000E37	36,242,230	16	Database record lock contention	(111) DB record lock: upd...	22,802,391
JOB116 / QPGMR / 658236: 0000100F	10,728,074	16	Database record lock contention	(123) DB record lock: weak	22,777,775
JOB116 / QPGMR / 658358: 00000CB1	11,487,127	16	Database record lock contention	(123) DB record lock: weak	22,778,002
JOB116 / QPGMR / 541406: 00000BA4	11,641,348	16	Database record lock contention	(123) DB record lock: weak	22,592,151
JOB116 / QPGMR / 643745: 00000A6C	9,749,152	16	Database record lock contention	(123) DB record lock: weak	22,743,782
JOB116 / QPGMR / 651635: 00000D94	3,833,019	16	Database record lock contention	(123) DB record lock: weak	22,761,211
JOB68 / QPGMR / 652089: 00000E5D	10,793,709	16	Database record lock contention	(123) DB record lock: weak	22,762,799
JOB116 / QPGMR / 667328: 00000A13	8,982,497	16	Database record lock contention	(123) DB record lock: weak	22,797,418

1 - 15 of 35

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 to graph the job’s data over time or view the call stack.

33.9 Situations

The Situations tab displays jobs that were detected by Situational Analysis as experiencing one or more problems.

Note: This tab will only appear if the **Collection summary** analysis has been ran.

SPLIT/Collection overview time signature **Interval Summary: Library Ibmdk2, Collection Split - #1**

Quick View Waits Wait bucket totals Objects waited on Holders SQL statistics Bad Current Waits Situations P

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11:...

Results: Max 35

Performance situation ID (SITID)	Job name/user/number: thread ID (OBJNAME)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Total (TOTAL)	Job group (JOBGRP)
11	JOB172 / USR182 / 675626: 000008BC	22,813,219	1	
11	11 : Journal cache could help performance JOB172 / USR182 / 675626: 000008BC	22,813,230	1	
10	JOB77 / QPGMR / 540512: 00000073	22,586,876	1	
10	JOB77 / QPGMR / 636961: 000010DE	22,730,405	1	
10	JOB77 / QPGMR / 667642: 0000085B	22,798,281	1	
10	JOB77 / QPGMR / 667764: 000001FE	22,798,523	1	
10	JOB77 / QPGMR / 676045: 0000048D	22,813,960	1	
10	JOB77 / QPGMR / 676120: 00000A5F	22,814,078	1	
7	JOB892 / QTMHHTTP / 539288: 00000D2E	22,578,310	1	
10	JOB954 / QPGMR / 661234: 0000061C	22,784,010	1	

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 to graph the job’s data over time or view the call stack.

33.10 Physical disk I/Os

This interface provides summarized metrics relating to page allocations and physical disk I/Os for the entire collection in the specified time interval.

SPLIT/Collection overview time signature **Interval Summary: Library Ibmdk2, Collection Split - #1**

Quick View Waits Wait bucket totals Objects waited on Holders SQL statistics Bad Current Waits Situations **Physical disk I/Os**

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

Results:

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Pages deallocated	6,949,794	224,383.81
Page frames requested	6,825,013	220,355.08
Pages allocated	6,806,660	219,762.53
Page frames released	6,549,917	211,473.22
Asynchronous DB reads	961,956	31,058.09
Asynchronous DB writes	493,305	15,927.04
Synchronous DB writes	147,143	4,750.71
Waits for asynchronous writes	135,446	4,373.06
Synchronous DB reads	125,099	4,038.99
Synchronous Non-DB reads	121,973	3,938.06
Page faults	100,847	3,255.98
IO pending page faults	74,081	2,391.80
Asynchronous Non-DB writes	39,559	1,277.21
Synchronous Non-DB writes	38,808	1,252.97
Asynchronous Non-DB reads	544	17.56

Interval Summary – Physical disk I/Os

33.11 Logical DB

This interface provides summarized metrics relating to Logical I/Os for the entire collection in the specified time interval.

SPLIT/Collection overview time signature Interval Summary: Library lbmdk2, Collection Split - #1

Quick View Waits Wait bucket totals Objects waited on Holders SQL statistics Bad Current Waits Situations Physical disk I/Os Logical DB IFS

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

Results:

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Reads	33,139,3...	1,069,949.32
Non reads/writes (All Others)	1,852,538	59,811.77
Writes	695,134	22,443.37
Rollbacks	241	7.78
Commits	9	.29
Updates	0	0
Deletes	0	0
Forced end of data	0	0
Opens	0	0
Closes	0	0
Index rebuilds	0	0
Sorts	0	0

Interval Summary – Logical I/Os

33.12 IFS

This interface provides summarized metrics relating to the IFS for the entire collection in the specified time interval.

SPLIT/Collection overview time signature **Interval Summary: Library Ibmdk2, Collection Split - #1**

Quick View Waits Wait bucket totals Objects waited on Holders SQL statistics Bad Current Waits Situations Physical disk I/Os Logical DB IFS

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

Results:

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Lookup cache hits	273,297	8,823.77
Opens	12,038	388.66
Symbolic link reads	1,324	42.74
Directory reads	432	13.94
Lookup cache misses	380	12.26
Non-directory creates	79	2.55
Non-directory deletes	37	1.19
Directory creates	16	.51
Directory deletes	0	0

Interval Summary - IFS

33.13 Other statistics

This interface provides summarized metrics relating to the 5250 transactions, stream file I/Os and more for the entire collection in the specified time interval.

SPUT/Collection overview time signature Interval Summary: Library lbmdk2, Collection Split - #1 x

Quick View Waits Wait bucket totals Objects waited on Holders SQL statistics Bad Current Waits Situations Physical disk I/Os Logical DB IFS Other statistics SQL

General:

Threads/tasks using CPU:	11889	Interval:	5
Threads/tasks idle:	30159	CPU utilization:	48.59%
Threads/tasks waiting on objects:	3167	CPU time:	11.294 minutes
Threads/tasks with holder identified:	31	Interval duration:	30.921 seconds
Temp storage job allocations (GB):	316.3921	Interval end:	2018-01-30-11.33.49.733000

Other statistics:

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Display I/O response transactions time (ms)	107,047,...	3,456,188...
Stream file reads	17,029	549.80
Stream file writes	2,937	94.82
Display I/O response transactions	420	13.56
Jobs submitted	114	3.68
Spool files created	66	2.13
Binary overflows	4	.12
Decimal overflows	4	.12
Float overflows	0	0
Wait to ineligible transitions	0	0
Active to ineligible transitions	0	0

Interval Summary – Other statistics

34 Interval Details Interface

This interface provides 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 and more.

Tip: See the previous section for more information on the [Quick View](#) tab.

34.1 General Section

The interval details property pages contain a section at the top that is consistent for all tabs except Quick View, SQL and Columns. 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:

Quick View	Call stack	Waits	Objects waited on	Physical disk I/Os	Logical DB	IFS	J9 JVM	SQL / Client	Other statistics	SQL		
General:												
Primary thread:		JOB63 / QPGMR / 676029: 00000876				Interval:		5				
Job subsystem:		JOB SBS45	Job status:		RUN	Job function:		PGM-PGM1622	Job CPU %:	0	Pool:	6
Current user profile:		QPGMR	Current state:		WAIT	Priority (XPF/LIC):		20/160	Original LIC:			176
Current or last wait:		(2/QGa) Qu gate - high performance, low-overhead ser				Wait duration:		11.882 seconds				
Object waited on:		LIB8/QJRDWH				Interval duration:		31.005 seconds				
Holding job or task:		None detected this interval				Interval end:		2018-01-30-11.33.49.733000				
SQL client job:		None detected this interval				Temp storage (MB):		72.74 / 72.74 (peak)				

Some of the less obvious data is described below:

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 may click this field 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.   These buttons increase or decrease the interval number to the next or prior one.
Job status	The status indicates what the job/thread was doing when the call stack was captured. The possible field values are the same as those for the Status field in WRKACTJOB's help text.
Current state	Indicates the running or waiting state for the thread. The possible values are: RUN, CPUQ or WAIT
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.
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.

34.2 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:

SPLIT/Collection overview time signature SPLIT/Machine level gate serialization rankings by thread: F... Interval Details: Library Ibmdk2, Collection Split - #1

Quick View Call stack Waits Objects waited on Physical disk I/Os Logical DB IFS J9 JVM SQL / Client Other statistics SQL Columns

General:

Primary thread: **JOB63 / QPGMR / 676029: 00000876** Interval: 5 [Refresh] [Previous] [Next] [End]

Job subsystem: JOBSBS45 Job status: RUN Job function: PGM-PGM1622 Job CPU %: 0 Pool: 6

Current user profile: QPGMR Current state: WAIT Priority (XPF/LIC): 20/160 Original LIC: 176

Current or last wait: (2/QGa) Qu gate - high performance, low-overhead ser Wait duration: 11.882 seconds

Object waited on: **LIB8/QJRDWH** Interval duration: 31.005 seconds

Holding job or task: None detected this interval Interval end: 2018-01-30-11.33.49.733000

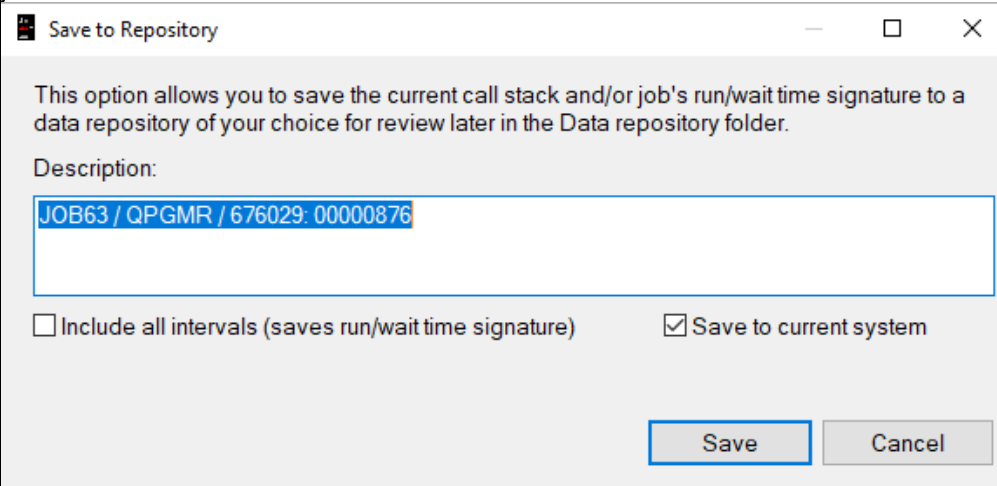
SQL client job: None detected this interval Temp storage (MB): 72.74 / 72.74 (peak)

Call stack contents: Advanced ☒ Include LIC Stack frames: 35 Save

Call level	Program model	Program	Module	Procedure	Offset	LIC statement number
001	LIC			qutde_block_trace	000000F8	248
002	LIC			slowLock_10QuGateCodeFQ2_2Qu8LockModeUIN32Q2_8TDQSEnum4Enum	000004FC	1276
003	LIC			#journal	00004BDC	19420
004	LIC			#dbdelim	00004298	17048
005	LIC			#dbdelen	00001B50	6992
006	LIC			#cfmir	000000E8	232
007	LIC			syscall_A_portal	00000148	328
008	ILE	QDBUDR	QDBUDR	QDBUDR	000083E0	
009	LIC			cblabbranch	000001D0	464
010	LIC			aiuser_program_call_portal	000000C0	192
011	ILE	QRNXIO	QRNXDBIO	_QRNX_DB_DELETE	00000458	
012	ILE	LIB9/PGM57	MOD64	P4450	00000F78	
013	ILE	LIB9/PGM28	MOD26	P3429	00000AC4	
014	ILE	LIB9/PGM83	MOD95	P2838	00001D1C	
015	ILE	LIB9/PGM83	MOD95	P131	00000284	
016	LIC			cblabbranch	000001D0	464

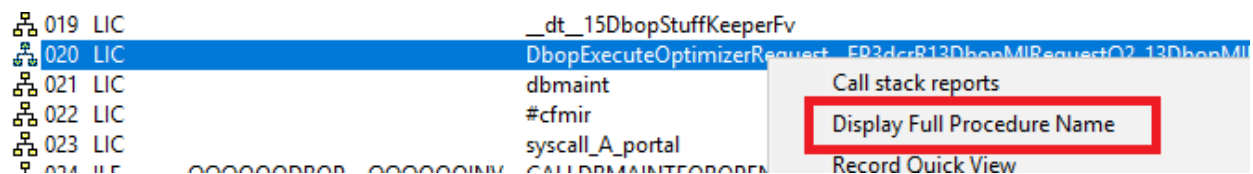
Interval Details – Call Stack

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 contains these choices: - Basic - Shows the call level, model, program, module and procedure - Detailed – Basic + Offset, LIC statement number - Advanced – Detailed + 4 addresses
Include LIC	Uncheck this box to remove LIC call levels from the call stack
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. - Stack frames (number of call levels) - Reason collected: Holder (call stack was collected because it is a holding another job) - Reason collected: Waiter (call stack was collected because it is held by another job) - Reason collected: Bad wait (call stack was collected because the job was stuck in a bad wait)
Save	Click this button to save the call stack to the current systems Job Watcher repository. Tip: These saved call stacks are retrievable later from the Job Watcher -> Data repository folder. If you wish to include all intervals and not just the single call stack for the current job/thread then check the “Include all intervals” checkbox in the window below.  <i>Save to Repository Window</i>

34.2.1 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 name, you can right-click the row where the procedure is found and choose the Display Full Procedure Name menu.



Display Full Procedure Name menu

34.2.2 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 call-level information in other jobs in the collection.

If you select one 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 traceback table address entries for those call levels selected must exist 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.

34.2.2.1 Total occurrences: all intervals

This report looks for the desired call stack entries in the entire collection and counts the total occurrences.

SPLIT/Collection overview time signature								
SPLIT/Machine level gate serialization ranki...								
Interval Details: Library lbmdk2, Collection...								
SPLIT/Total occurrences: all intervals for ...								
Library name (LIBNAME)	Collection name (MBRNAME)	Total (TOTAL)	Program library (PGMLIB)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Procedure type (PROCTYPE)	Traceback table address (TBTADDR)
IBMDK2	SPLIT	29	LIB9	PGM57	MOD64	P4450	1	3895B8D2DD03EB60

Total occurrences: all intervals

34.2.2.2 Total occurrences: this interval

This report is the same as the previous one except it only looks for matches for the current interval.

SPLIT/Collection overview time signature								
SPLIT/Machine level gate serialization ranki...								
Interval Details: Library lbmdk2, Collection...								
SPLIT/Total occurrences: this interval for...								
Library name (LIBNAME)	Collection name (MBRNAME)	Total (TOTAL)	Program library (PGMLIB)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Procedure type (PROCTYPE)	Traceback table address (TBTADDR)
IBMDK2	SPLIT	4	LIB9	PGM57	MOD64	P4450	1	3895B8D2DD03EB60

Total occurrences: this interval

34.2.2.3 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.

SPLIT/Collection overview time signature			SPLIT/Machine level gate serialization ranki...			Interval Details: Library lbmdk2,		
Library name (LIBNAME)	Collection name (MBRNAME)	Total (TOTAL)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Job/task name (TDEJOBNAME)			Thread ID (THREADID)	G
IBMDK2	SPLIT	4	22,584,324	JOB63	QPGMR	540180	00000000000001B2	J
IBMDK2	SPLIT	4	22,814,435	QPADEV2B4FUSR58	676300	00000000000009F8		C
IBMDK2	SPLIT	3	22,814,151	JOB63	QPGMR	676212	0000000000000CF6	J
IBMDK2	SPLIT	1	22,738,891	QPADEV05P8USR66	641265	0000000000000899		C
IBMDK2	SPLIT	1	22,730,210	JOB1004	QPGMR	636924	0000000000000602	J
IBMDK2	SPLIT	1	22,798,371	JOB63	QPGMR	667726	000000000000111A	J
IBMDK2	SPLIT	1	22,813,732	JOB63	QPGMR	675966	0000000000000C87	J
IBMDK2	SPLIT	1	22,583,228	JOB1005	QPGMR	539924	0000000000001F7D	J
IBMDK2	SPLIT	1	22,814,603	JOB68	QPGMR	676412	0000000000000A61	J
IBMDK2	SPLIT	1	22,729,626	JOB1004	QPGMR	636860	0000000000000110	J
IBMDK2	SPLIT	1	22,814,436	JOB63	QPGMR	676301	000000000000002D	J
IBMDK2	SPLIT	1	22,595,231	JOB1007	QPGMR	542106	000000000000011AE	J
IBMDK2	SPLIT	1	22,760,062	JOB63	QPGMR	651350	00000000000001255	J
IBMDK2	SPLIT	1	22,583,591	JOB966	USR153	540033	0000000000000D43	J
IBMDK2	SPLIT	1	22,707,636	JOB211	USR349	621046	0000000000000EC9	J
IBMDK2	SPLIT	1	22,584,425	JOB1003	QPGMR	540220	000000000000014C	J
IBMDK2	SPLIT	1	22,815,371	JOB1007	QPGMR	676668	00000000000006C6	J
IBMDK2	SPLIT	1	22,729,526	JOB1004	QPGMR	636848	0000000000000DF0	J
IBMDK2	SPLIT	1	22,587,079	JOB970	QPGMR	540700	000000000000052A	J
IBMDK2	SPLIT	1	22,813,933	JOB63	QPGMR	676029	0000000000000876	J
IBMDK2	SPLIT	1	22,587,973	JOB1004	QPGMR	540874	0000000000000207	J

Total occurrences: by job

34.2.2.4 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.

SPLIT/Collection overview time signature			SPLIT/Machine level gate serialization ranki...			Interval Details: Library lbmdk2, Collection...		SPLIT/Total occurrences by offset: all int...	
Library name (LIBNAME)	Collection name (MBRNAME)	Program library (PGMLIB)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Offset time (seconds) (OFFSET)	Traceback table address (TBTADDR)	Total (TOTAL)	
IBMDK2	SPLIT	LIB9	PGM57	MOD64	P4450	00000F78	3895B8D2DD03EB60	29	
IBMDK2	SPLIT	LIB9	PGM57	MOD64	P4450	00004088	3895B8D2DD03EB60	15	
IBMDK2	SPLIT	LIB9	PGM57	MOD64	P4450	00000064	3895B8D2DD03EB60	5	
IBMDK2	SPLIT	LIB9	PGM57	MOD64	P4450	0000267C	3895B8D2DD03EB60	4	
IBMDK2	SPLIT	LIB9	PGM57	MOD64	P4450	00000690	3895B8D2DD03EB60	1	

Total occurrences: by offset, all intervals

34.2.2.5 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.

SPLIT/Collection overview time signature			SPLIT/Machine level gate serialization ranki...			Interval Details: Library lbmdk2, Collection...		SPLIT/Total occurrences for this job/task...	
Library name (LIBNAME)	Collection name (MBRNAME)	Total (TOTAL)	Program library (PGMLIB)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Procedure type (PROCTYPE)	Traceback table address (TBTADDR)	
IBMDK2	SPLIT	1	LIB9	PGM57	MOD64	P4450	1	3895B8D2DD03EB60	

Total occurrences for this job/task: all intervals

34.2.2.6 Jobs with this occurrence: this interval

This report displays the list of jobs matching the current selection in the current time interval.

Interval Details: Library lbmdk2, Collection...										
Interval number (INTERVAL)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Job/task name (TDEJOBNAME)		Thread ID (THREADID)		Generic job name (GENJOBNAME)	Thread status (THRDSTATUS)	Current user profile (CURRUP)	Program library (PGMLI)	
5	22,584,324	JOB63	QPGMR	540180	0000000000001B2	JOB63	RUN	QPGMR	LIB9	
5	22,813,732	JOB63	QPGMR	675966	000000000000C87	JOB63	RUN	QPGMR	LIB9	
5	22,587,973	JOB1004	QPGMR	540874	000000000000207	JOB1004	RUN	QPGMR	LIB9	
5	22,813,933	JOB63	QPGMR	676029	000000000000876	JOB63	RUN	QPGMR	LIB9	

Jobs with this occurrence this interval

34.2.2.7 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.

Interval Details: Library lbmdk2, Collection...										
Interval number (INTERVAL)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Job/task name (TDEJOBNAME)		Thread ID (THREADID)		Generic job name (GENJOBNAME)	Thread status (THRDSTATUS)	Current user profile (CURRUP)		
3	22,595,231	JOB1007	QPGMR	542106	0000000000011AE	JOB1007	RUN	QPGMR		
5	22,813,933	JOB63	QPGMR	676029	000000000000876	JOB63	RUN	QPGMR		
5	22,584,324	JOB63	QPGMR	540180	0000000000001B2	JOB63	RUN	QPGMR		
5	22,587,973	JOB1004	QPGMR	540874	000000000000207	JOB1004	RUN	QPGMR		
5	22,813,732	JOB63	QPGMR	675966	000000000000C87	JOB63	RUN	QPGMR		
7	22,584,425	JOB1003	QPGMR	540220	00000000000014C	JOB1003	RUN	QPGMR		
7	22,760,062	JOB63	QPGMR	651350	0000000000001255	JOB63	RUN	QPGMR		
12	22,587,079	JOB970	QPGMR	540700	000000000000052A	JOB970	RUN	QPGMR		
12	22,729,626	JOB1004	QPGMR	636860	0000000000000110	JOB1004	RUN	QPGMR		
12	22,815,371	JOB1007	QPGMR	676668	00000000000006C6	JOB1007	RUN	QPGMR		
14	22,738,891	QPADEV05P8USR66		641265	0000000000000899	QPADEV0*	RUN	USR66		
15	22,814,436	JOB63	QPGMR	676301	000000000000002D	JOB63	RUN	QPGMR		
18	22,707,636	JOB211	USR349	621046	0000000000000EC9	JOB211	RUN	USR349		
19	22,730,210	JOB1004	QPGMR	636924	0000000000000602	JOB1004	RUN	QPGMR		
19	22,814,603	JOB68	QPGMR	676412	0000000000000A61	JOB68	RUN	QPGMR		

Jobs with this occurrence: all intervals

34.2.2.8 Occurrences by interval

This report displays the total hits per interval for the call stack entries selected.

Interval Details: Library lbmdk2, Collection...										
Library name (LIBNAME)	Collection name (MBRNAME)	Interval number (INTERVAL)	Total (TOTAL)	Program library (PGMLIB)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Procedure type (PROCTYPE)	Traceback table address (TBTADDR)	Offset time (seconds) (OFFSET)
IBMDK2	SPLIT	5	4	LIB9	PGM57	MOD64	P4450		1 389588D2DD03EB60	00000F78
IBMDK2	SPLIT	12	3	LIB9	PGM57	MOD64	P4450		1 389588D2DD03EB60	00000F78
IBMDK2	SPLIT	21	3	LIB9	PGM57	MOD64	P4450		1 389588D2DD03EB60	00000F78
IBMDK2	SPLIT	22	3	LIB9	PGM57	MOD64	P4450		1 389588D2DD03EB60	00000F78
IBMDK2	SPLIT	23	3	LIB9	PGM57	MOD64	P4450		1 389588D2DD03EB60	00000F78

Occurrences by interval

34.2.2.9 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.

w time signature		SPLIT/Machine level gate serialization ranki...			Interval Details: Library lbmdk2, Collection...			SPLIT/Occurrences by interval for		
Interval number (INTERVAL)	Total (TOTAL)	Program library (PGMLIB)	Program name (PGMNAME)	Module name (MODNAME)	Procedure (PROCNAME)	Procedure type (PROCTYPE)	Traceback table address (TBTADDR)	Offset time (seconds) (OFFSET)	Task id (TASKID)	Task address (TASKADDR)
5	1	LIB9	PGM57	MOD64	P4450		1 3895B8D2DD03EB60	00000F78		

Occurrences by interval for this job/task

34.2.2.10 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.

SPLIT/Collection overview time signature		SPLIT/Machine level gate serialization rankings by thread: From 1...			Interval Details: Library lbmdk2, Collection Split - #1			SPLIT/Jobs and programs (14 levels) calling the selected pgm/...						
Total call stacks (STACKCNT)	Job name and user name (JOB_AND_USER)	Generic job name (GENJOBNAME)	Ending interval (MAXINT)	Interval number (INTERVAL)	Task count (uniquely identifies a task/thread) (TASKCOUNT)	Program lib 2 (PGMLIB2)	Program name 2 (PGMNAME2)	Procedure name 2 (PROCNAME2)	Program lib 3 (PGMLIB3)	Program name 3 (PGMNAME3)	Procedure name 3 (PROCNAME3)	Program lib 4 (PGMLIB4)	Program name 4 (PGMNAME4)	Procedure name 4 (PROCNAME4)
14	JOB63 QPGMR	JOB63	26	4	22,815,730	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
5	QPADEV281CUSR176	QPADEV2*	17	9	22,745,326	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
4	JOB114 QPGMR	JOB114	19	12	22,814,780	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
4	QPADEV284FUSR58	QPADEV2*	23	20	22,814,435	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
4	JOB68 QPGMR	JOB68	26	16	22,815,805	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
3	JOB1007 QPGMR	JOB1007	19	3	22,817,326	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
2	JOB211 USR349	JOB211	19	18	22,707,636	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
2	JOB1004 QPGMR	JOB1004	19	12	22,730,210	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
2	JOB63 QPGMR	JOB63	5	5	22,813,732	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
2	JOB970 QPGMR	JOB970	13	12	22,587,079	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
2	QPADEV05N3USR78	QPADEV0*	19	18	22,813,927	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
1	JOB966 USR153	JOB966	24	24	22,583,591	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
1	JOB1004 QPGMR	JOB1004	23	23	22,729,526	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131
1	JOB923 QPGMR	JOB923	30	30	22,589,405	LIB9	PGM28	P3429	LIB9	PGM43	P4159	LIB9	PGM46	P3260
1	JOB983 USR153	JOB983	24	24	22,819,364	LIB9	PGM28	P3429	LIB9	PGM83	P2838	LIB9	PGM83	P131

Jobs and programs (14 levels) calling the selected pgm/procedure

34.3 Waits

The Waits 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 provided.

Quick View

Call stack

Waits

Objects waited on

Physical disk I/Os

Logical DB

IFS

J9 JVM

SQL / Client

Other statistics

SQL

Columns

General:

Primary thread:

JOB63 / QPGMR / 676029: 00000876

Interval:

5

Job subsystem:

JOBSBS45

Job status:

RUN

Job function:

PGM-PGM1622

Job CPU %:

0

Pool:

6

Current user profile:

QPGMR

Current state:

WAIT

Priority (XPF/LIC):

20/160

Original LIC:

176

Current or last wait:

(2/QGa) Qu gate - high performance, low-overhead ser

Wait duration:

11.882 seconds

Object waited on:

LIB8/QJRDWH

Interval duration:

31.005 seconds

Holding job or task:

None detected this interval

Interval end:

2018-01-30-11.33.49.733000

SQL client job:

None detected this interval

Temp storage (MB):

72.74 / 72.74 (peak)

Wait bucket number (BUCKET)	Wait bucket description (BUCKETDESC)	Percent of total time (PCTOFTIME)	Time (seconds) (BKTIME)	Total occurrences (BUCKETCNT)	Average wait time (seconds) (AVGWAIT)	Occurrences per second (BKTRATE)	Current wait duration (usecs) (CURRWTDUR)
1	Dispatched CPU	.0258	.007	45	.000177	1.451	0
2	CPU queueing	.0017	0	45	.000011	1.451	0
4	Other waits	.0000	0	1	.000010	.032	0
5	Disk page faults	.0059	.001	3	.000614	.096	0
6	Disk non fault reads	.0042	.001	1	.001313	.032	0
11	Journaling	.0644	.019	6	.003327	.193	0
14	Machine level gate serialization	99.8980	30.973	34	.910989	1.096	11,882,005

Interval Details – Waits

The last column in the example above shows the current wait duration in machine level gate serialization.

34.4 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.

Wait object information:

Description	Value
Wait object name	LIB8/QJRDWH
Wait object type description	JOURNAL
Wait object segment type description	JOURNAL RESERVED
Object type (hex)	0901
Segment type (hex)	20C5
LIC wait object	QGa
Current or last LIC wait object handle	D457A20A51000180
Wait object base segment address	06C09EFFF4000000

Interval Details – Object Waited on

34.5 Physical disk I/Os

This tab contains additional metrics relating to physical disk I/Os, page allocations and page faults.

Interval Details: Library Ibmdbk2, Collection Split - #1 x SPLIT/Collection overview time signature SPLIT/Machine level gate seriali

Quick View Call stack Waits Objects waited on Physical disk I/Os Logical DB IFS J9 JVM SQL / Client Other statistics SQL C

General:

Primary thread: JOB63 / QPGMR / 676029: 00000876 Interval: 5

Job subsystem: JOBSBS45 Job status: RUN Job function: PGM-PGM1622 Job CPU %: 0 Pool: 6

Current user profile: QPGMR Current state: WAIT Priority (XPF/LIC): 20/160 Original LIC: 176

Current or last wait: (Z/QGa) Qu gate - high performance, low-overhead ser Wait duration: 11.882 seconds

Object waited on: LIB8/QJRDWH Interval duration: 31.005 seconds

Holding job or task: None detected this interval Interval end: 2018-01-30-11.33.49.733000

SQL client job: None detected this interval Temp storage (MB): 72.74 / 72.74 (peak)

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Pages allocated since job start	19,206	620.09
Pages allocated	1,233	39.80
Page deallocated since job start	206	6.65
Memory page demand	106	3.42
Page frames requested	106	3.42
Page faults	3	.09
Synchronous DB reads	2	.06
Synchronous non-DB reads	2	.06
Synchronous DB writes	0	0
Synchronous non-DB writes	0	0
Asynchronous DB reads	0	0
Asynchronous non-DB reads	0	0
Asynchronous DB writes	0	0
Asynchronous non-DB writes	0	0
IO pending page faults	0	0
Waits for asynchronous writes	0	0
Page deallocated	0	0
Page frames released	0	0

Interval Details -> Physical I/Os

34.6 Logical DB

This tab contains metrics relating to Logical I/O operations for the current job in the interval.

Quick View Call stack Waits Objects waited on Physical disk I/Os **Logical DB** IFS J9 JVM SQL / Client Other statistics SQL Co

General:

Primary thread: **JOB63 / QPGMR / 676029: 00000876** Interval: **5**

Job subsystem: JOBSBS45 Job status: RUN Job function: PGM-PGM1622 Job CPU %: 0 Pool: 6

Current user profile: QPGMR Current state: WAIT Priority (XPF/LIC): 20/160 Original LIC: 176

Current or last wait: (2/QGa) Qu gate - high performance, low-overhead ser Wait duration: 11.882 seconds

Object waited on: **LIB8/QJRDWH** Interval duration: 31.005 seconds

Holding job or task: None detected this interval Interval end: 2018-01-30-11.33.49.733000

SQL client job: None detected this interval Temp storage (MB): 72.74 / 72.74 (peak)

Note: These numbers reflect the job's logical IOs for this interval for all threads.

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Logical reads	5	.16
Logical writes	0	0
Logical updates and deletes	0	0
Logical force end of data	0	0
Logical commits	0	0
Logical rollbacks	0	0
Logical index rebuilds	0	0
Logical sorts	0	0

Interval Details -> Logical DB

34.7 IFS

This panel shows the IFS statistics for the current job in the current interval.

Quick View Call stack Waits Objects waited on Physical disk I/Os Logical DB **IFS** J9 JVM SQL / Client Other statistics SQL

General:

Primary thread: **JOB63 / QPGMR / 676029: 00000876** Interval:

Job subsystem: JOBSBS45 Job status: RUN Job function: PGM-PGM1622 Job CPU %: 0 Pool: 6

Current user profile: QPGMR Current state: WAIT Priority (XPF/LIC): 20/160 Original LIC: 176

Current or last wait: (2/QGa) Qu gate - high performance, low-overhead ser Wait duration: 11.882 seconds

Object waited on: **LIB8/QJRDWH** Interval duration: 31.005 seconds

Holding job or task: None detected this interval Interval end: 2018-01-30-11.33.49.733000

SQL client job: None detected this interval Temp storage (MB): 72.74 / 72.74 (peak)

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Symbolic link reads	0	0
Directory reads	0	0
Lookup cache hits	0	0
Lookup cache misses	0	0
Opens	0	0
Directory creates	0	0
Non-directory creates	0	0
Directory deletes	0	0
Non-directory deletes	0	0

Interval Details -> IFS

34.8 J9 JVM

This tab contains metrics relating to Java J9 JVMs (if one exists) in the current job in the interval.

Quick View Call stack Waits Objects waited on Physical disk I/Os Logical DB IFS J9 JVM SQL / Client Other statistics SQL Color

General:

Secondary thread: **JOB899 / USR375 / 259533: 00001084** ↑ Interval: 4

Job subsystem: QHTTSPVR Job status: JVAW Job function: JVM-/qibm/prod Job CPU %: .30 Pool: 3

Current user profile: USR375 Current state: WAIT Priority (XPF/LIC): 25/165 Original LIC: 181

Current or last wait: (200/JUW) Java: user wait Wait duration: 1.755 seconds

Object waited on: Segment type LIC HEAP (MWS) AREA DATA Interval duration: 27.355 seconds

Holding job or task: None detected this interval Interval end: 2018-01-30-11.33.18.761000

SQL client job: None detected this interval Temp storage (MB): 3379.44 / 3379.44 (peak)

Description	Value
Thread name	JTT3756
Last object name	
JVM version	1.8.0
JVM type	1
Garbage collection policy	-Xgcpolicy:genc
Garbage collection cycle number	75275
Garbage collection reason	1
Garbage collection area	0
Garbage collection compaction reason	0
Total GC time (ms)	1569228
GC time last cycle (ms)	28
JVM start time	2018-01-28-12.28.42.376000
JVM process ID	1577788
PASE TID (hex)	0000000044C3017
PASE TID	72101911
Heap in use size	1590.1 MB
Maximum heap size	2048 MB
Initial heap size	2048 MB
Current heap allocated size	2048 MB
Memory malloc size	1180.4 MB
Internal memory size	51.6 MB
JIT memory size	270 MB
Shared class size	0

Interval Details -> J9 JVM

34.9 SQL / Client

The SQL tab displays information about any SQL statements that were running in the job. Depending on how the data was collected, these SQL statements were either captured at the end of the interval or were the last executed statement(s).

SPLIT/JOB WATCHER - SQL STATEMENT INFO - #1 Interval Details: Library Ibmdk2, Collection Split - #1

Record Quick View Call stack Waits Objects waited on Physical disk I/Os Logical DB IFS J9 JVM SQL / Client Other statistics SQL Columns

General:

Primary thread: JOB68 / QPGMR / 540342: 00000936 Interval: 1

Job subsystem: JOBSBS41 Job status: RUN Job function: PGM-PGM1622 Job CPU %: .03 Pool: 5

Current user profile: QPGMR Current state: WAIT Priority (XPF/LIC): 20/160 Original LIC: 176

Current or last wait: (167/SWt) Mainstore/logical-dasd-io: dasd write Wait duration: 1.623 milliseconds

Object waited on: CO9QVO88IF/CO9QVO88IF Interval duration: 11.288 seconds

Holding job or task: None detected this interval Interval end: 2018-01-30-11.31.40.283000

SQL client job: None detected this interval Temp storage (MB): 418.59 / 418.99 (peak)

Launch Run SQL Scripts Include host variables ☒ Other information: SQL related metrics + host variables

Description	Value
QRO hash	0000000000000000
SQL statements executed	499
SQL logical reads	102
SQL logical reads per second	9.2518
Native DB file full opens	160
Native DB file full opens per second	14.5126
Pseudo closed SQL cursors	1
Remote DBS name	*LOCAL
Package library	LIB17
Package name	PKG11
Package source library	LIB17
Package source file	QRPGLESRC
Package source member	SRM11
Package source date	2017-03-24-12.56.23.000000
SQL file reserved #2	~
SQL file reserved #1	ZCE{

OPEN FETCH1

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.

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 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 will be required.

34.10 Other statistics

This panel show 5250 display transaction, numeric overflows and other metrics less often used for the current job in the current interval.

SPLIT/JOB WATCHER - SQL STATEMENT INFO - #1

Interval Details: Library Ibmdk2, Collection Split - #1

Record Quick View

Call stack

Waits

Objects waited on

Physical disk I/Os

Logical DB

IFS

J9 JVM

SQL / Client

Other statistics

General:

Primary thread: JOB68 / QPGMR / 540342: 00000936

Interval: 1

Job subsystem: JOBSBS41

Job status: RUN

Job function: PGM-PGM1622

Job CPU %: .03

Pool: 5

Current user profile: QPGMR

Current state: WAIT

Priority (XPF/LIC): 20/160

Original LIC: 176

Current or last wait: (167/SWt) Mainstore/logical-dasd-io: dasd write

Wait duration: 1.623 milliseconds

Object waited on: CO9OVO88IF/CO9OVO88IF

Interval duration: 11.288 seconds

Holding job or task: None detected this interval

Interval end: 2018-01-30-11.31.40.283000

SQL client job: None detected this interval

Temp storage (MB): 418.59 / 418.99 (peak)

Other statistics

Description (DESC)	Total (TOTCNT)	Rate per second (RATED)
Display I/O transactions	0	0
Display I/O transactions time	0	0
Binary overflows	0	0
Decimal overflows	0	0
Float overflows	0	0
Stream file reads	0	0
Stream file writes	0	0
Mutex wait time (us)	0	0
Wait to ineligible transitions	0	0
Active to ineligible transitions	0	0

Interval Details -> Other statistics

35 Data repository

This folder in Job Watcher contains any desired information saved from previously viewed call stacks within the [Interval Details -> Call Stack](#) interface. Click the Save button within that interface to add a new entry to the data repository.

This is typically done if you wish to keep track of certain call stacks and associate these with a job name and/or description.

An example follows:

IBM i Connections Job Watcher - #1 Job Watcher - #2									
Collection	Library	Description	Collection type	DB files VRM	Partition collected on VRM	Partition collected on	Start time		
T212111104	QIDRDR7205	QSPL / QSYS / 180239: 00000005	Call stack	7.2	7.2	IDOC720	2022-01-18-06.40.31.74		
T122418942	QIDRDR7200	QDBFSTCCOL	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T182446254	QIDRDR7200	QZDASOINIT / QUSER / 105283: 000000D3	Job signature	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T518849369	QIDRDR7200	QZDASOINIT / QUSER / 105296: 00000125	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T182017368	QIDRDR7200	QRMCTCASD / QSYS / 100729: 00000002	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T182851554	QIDRDR7200	QUSRDIR / QDIRSRV / 100776: 00000001	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T123517347	QIDRDR7200	QDBFSTCCOL / QSYS / 100601: 00000001	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T121936814	QIDRDR7200	QZDASOINIT RECEIVE	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T183841785	QIDRDR7200	aaaaa	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T183345637	QIDRDR7200	IOCMETHLINE 01: (1932)	Call stack	7.2	7.2	IDOC720	2021-08-04-07.39.45.91		
T521929657	QIDRDR7205	SYSTEMMONITOR: (1007)	Call stack	7.2	7.2	IDOC720	2020-11-09-12.16.55.56		
T152165645	QIDRDR7205	QFILESYS1 / QSYS / 020657: 00000001	Call stack	7.2	7.2	IDOC720	2020-11-09-12.16.55.56		
T122129143	QIDRDR7201	QDBSERVE	Call stack	7.2	7.2	IDOC720	2012-11-12-00.28.14.54		

Data repository folder

The collections within this folder are typically stored in 1 or more QIDRDR* libraries. These are typically split collections that contain only 1 interval and 1 thread of data. If the collection type column in the list indicates **Job signature** then the all intervals for the selected job/thread was saved and can be viewed instead of job the single call stack.

Tip: Right-click one of these collections to view either the call stack or job run wait signature graph.

IBM i Connections Job Watcher - #1 Job Watcher - #2									
Collection	Library	Description	Collection type	DB files VRM	Partition collected on VRM	Partition collected on	Start time		
T212111104	QIDRDR7205	QSPL / QSYS / 180239: 00000005	Call stack	7.2					
T122418942	QIDRDR7200	QDBFSTCCOL	Call stack	7.2					
T182446254	QIDRDR7200	QZDASOINIT / QUSER / 105283: 000000D3	Job signature	7.2					
T518849369	QIDRDR7200	QZDASOINIT / QUSER / 105296: 00000125	Call stack	7.2					
T182017368	QIDRDR7200	QRMCTCASD / QSYS / 100729: 00000002	Call stack	7.2					
T182851554	QIDRDR7200	QUSRDIR / QDIRSRV / 100776: 00000001	Call stack	7.2					

Data repository – Display call stack

36 JVM analysis

This folder in Job Watcher contains any previously created collections of PRTJVMJOB data submitted using iDoctor. Right-click the JVM analysis folder and use the Browse JVMs option to view the JVMs active on the system.

Connections Job Watcher - #1 Job Watcher - #2 IDOC720: WRKACTJOB Results Name: *ALL, Use... x								
WRKACTJOB Results	Job name	Job user	Job number	Date/time JVM started	Accumulated GC time(ms)	Last GC cycle #	GC policy name	Proc ID
	QSRVMON	QSYS	180320	2021-12-21-14.06.16.671000	305	70	optthruput	78

Browse JVMs (via WRKACTJOB Results)

36.1 Analyze JVM

Use this menu option to analyze the JVM using PRTJVMJOB command.

Watcher - #1 Job Watcher - #2 IDOC720: WRKACTJOB Results Name: *ALL, Use... x								
Job name	Job user	Job number	Date/time JVM started	Accumulated GC time(ms)	Last GC cycle #	GC policy name	Proce ID	
	QSRVMON	QSYS	180320	2021-12-21-1	...	...	optthruput	78

Explore
 Record Quick View
 Select fields...
Analyze JVM
 Add Job Watcher Definition...

Analyze JVM menu for a JVM job

Analyze JVM ✕

This runs the PRTJVMJOB command against the desired JVM to create GC table output for analysis. Open Job Watcher -> JVM Analysis folder to view the results. Please provide the desired library to create the data in.

Output library:

MYLIB

OK

Cancel

Analyze JVM Window