



Monitoring IBM DB2 Content Manager V8.3 for z/OS with OMEGAMON XE for DB2 Performance Expert on z/OS

White Paper

OMEGAMON XE for DB2 Performance Expert Development

Michael Reichert

rei@de.ibm.com

IBM Boeblingen Laboratory

Schoenaicherstrasse 220

71032 Boeblingen, Germany

IBM Software Group - Information Management zSeries

Christian Daser

chris.daser@de.ibm.com

IBM Deutschland GmbH

Am Keltenwald 1

71139 Ehningen, Germany





© IBM Corporation 2006

Monitoring Content Manager with OMEGAMON XE for DB2 PE on z/OS



1 About this document

DB2 Universal Database for z/OS is a major component of a DB2 Content Management (CM) environment. Therefore, monitoring performance of DB2 is a key to overall performance of a CM system. Content Manager administrators, however, are often not familiar with DB2 for z/OS tuning measurements and concepts. This is where OMEGAMON XE for DB2 Performance Expert on z/OS helps. It assists DB2 and CM system administrators in monitoring their DB2 subsystem in a CM environment.

From a CM system administrator's point of view, a CM system works correctly if all actions that a Content Manager client executes complete without errors and within a reasonable response time.

From a DB2 administrator's point of view, a database environment works correctly if specific performance metrics, for example, the buffer pool hit ratio, have acceptable values, or if specific events, such as many locks or deadlocks, do not occur.

This document wants to consider both points of view when trying to achieve performance improvements in a CM environment. OMEGAMON XE for DB2 Performance Expert on z/OS provides the CM user with an insight into the overall health of the DB2 subsystems in a CM environment without requiring in-depth knowledge about DB2 on z/OS.

This document describes how you can use OMEGAMON XE for DB2 Performance Expert on z/OS to analyze the performance of a DB2 database in a typical Content Manager environment. Content Manager includes many components, for example, CM Clients, Resource Manager, and Library Server that you can analyze by using the monitoring functions of OMEGAMON XE for DB2 Performance Expert on z/OS. This document also gives an overview of these functions and shows how they help you get the most from your Content Manager installation.

Document structure:

Chapter 4 briefly introduces the components of a Content Management environment. It also gives an overview about the architecture of OMEGAMON XE for DB2 Performance Expert on z/OS.

Chapter 5 provides a DB2 view on a Content Management environment. Because DB2 database systems play an integral part in a Content Management environment, it is also explained how Content Management concepts, such as *item types*, are mapped to DB2 concepts. Moreover, it is also shown how you can customize Content Management environments regarding DB2.

Chapter 6 demonstrates how to monitor a Content Management environment with OMEGAMON XE for DB2 Performance Expert on z/OS by using scenarios.

Terminology:

- The CM environment that is used for the examples and scenarios in this document is referred to as *sample environment*.
- Content Manager Library Server is referred to as Library Server.
- Content Manager Resource Manager is referred to as Resource Manager.

Acknowledgement

Thanks to the following people for their support and contributions to this project:

Ute Baumbach
Siegfried Beuttler
Derek Carter
Uwe Denneler
Norbert Jenninger
Holger König
Mario Lichtsinn
Henry Martens
Larry Schroeder
Jim Tau



© IBM Corporation 2006

Monitoring Content Manager with OMEGAMON XE for DB2 PE on z/OS



2 Table of Contents

1	ABOUT THIS DOCUMENT	3
2	TABLE OF CONTENTS.....	5
3	TABLE OF FIGURES.....	6
4	INTRODUCTION.....	7
4.1	THE CONTENT MANAGER ENVIRONMENT.....	7
4.1.1	CM Library Server.....	8
4.1.2	CM Resource Manager.....	8
4.1.3	CM System Administration Client.....	9
4.1.4	CM Client for Windows.....	9
4.1.5	CM Toolkit for application development.....	9
4.2	OMEGAMON XE FOR DB2 PERFORMANCE EXPERT ON Z/OS.....	9
4.2.1	OMEGAMON Server.....	10
4.2.2	Connect Monitoring Agent.....	10
4.2.3	Performance Expert Client.....	10
4.2.4	Performance Expert ISPF interface.....	10
4.2.5	OMEGAMON Classic.....	11
4.2.6	Performance Expert Batch Reporting.....	11
4.2.7	Tivoli Enterprise Portal.....	11
5	A DB2 VIEW ON YOUR CM ENVIRONMENT	12
5.1	INSTALLATION PARAMETERS	12
5.2	MODIFYING CONTENT MANAGER DDL.....	12
5.3	OBJECT ACCESS METHOD (OAM).....	13
5.4	EXPLORING RELATED DB2 TABLES.....	14
5.5	EXPLORING RELATED TABLES IN OAM.....	16
6	MONITORING CONTENT MANAGER.....	18
6.1	SYSTEM OVERVIEW	18
6.1.1	System Health.....	18
6.1.2	Exception Processing.....	20
6.2	ANALYZING DB2 PERFORMANCE.....	21
6.2.1	Buffer pool.....	21
6.2.2	Access Paths.....	22
6.2.3	Tracing Content Manager queries.....	24
6.2.4	Dynamic SQL.....	29
6.2.5	Object Analysis and I/O.....	33
6.3	MONITORING DB2 CONNECT.....	37
6.3.1	Connection Pooling and Connection Concentrator.....	37
6.3.2	TCP/IP & networking.....	38
6.4	PERFORMANCE WAREHOUSE	39
6.4.1	Determine peak workload times in your CM system.....	41
6.4.2	Analysis of thread resource consumption.....	43
7	APPENDIX A: INSTALLATION TOKENS.....	47
8	APPENDIX B: SPACE REQUIREMENTS	48
8.1.1	Library Server.....	48
8.1.2	Resource Manager.....	50
9	APPENDIX C: REFERENCES.....	51



3 Table of figures

FIGURE 1: CONTENT MANAGER AND OMEGAMON XE FOR DB2 PERFORMANCE EXPERT ON z/OS SETUP.....	7
FIGURE 2: SAMPLE CM INSTALLATION TOKENS.....	12
FIGURE 3: ITEM TYPE PROPERTIES OF CSLDMAILARCHIV2	15
FIGURE 4: ITEM TYPE AND MVS COLLECTION NAMES	16
FIGURE 5: DISPLAY OAM DATABASE IN DB2.....	16
FIGURE 6: DB2 PERFORMANCE EXPERT CLIENT AND SYSTEM OVERVIEW	19
FIGURE 7: PERFORMANCE EXPERT CLIENT - SYSTEM HEALTH GRAPHS.....	20
FIGURE 8: CM INSTALLATION TOKENS	21
FIGURE 9: PERFORMANCE EXPERT CLIENT - BUFFER POOL HIT RATIO	22
FIGURE 10: SQL STATEMENT IN PERFORMANCE EXPERT CLIENT THREAD DETAILS.....	23
FIGURE 11: CONTENT MANAGER QUERY ACCESS PATH IN VISUAL EXPLAIN	24
FIGURE 12: CM CLIENT FOR WINDOWS - SEARCH.....	25
FIGURE 13: PERFORMANCE EXPERT CLIENT - THREAD SUMMARY	25
FIGURE 14: SQL ACTIVITY REPORT - GENERATION	26
FIGURE 15: SQL ACTIVITY REPORT - PROGRESS.....	26
FIGURE 16: CM CLIENT FOR WINDOWS - SEARCH COMPLETE	27
FIGURE 17: SQL ACTIVITY REPORT OUTPUT	28
FIGURE 18: PERFORMANCE EXPERT CLIENT - EFFECTIVENESS OF DYNAMIC STATEMENT CACHE.....	29
FIGURE 19: PERFORMANCE EXPERT CLIENT - SQL STATEMENTS IN DYNAMIC STATEMENT CACHE.....	30
FIGURE 20: PERFORMANCE EXPERT CLIENT - DETAILS ABOUT A STATEMENT IN DYNAMIC STATEMENT CACHE..	31
FIGURE 21: VISUAL EXPLAIN - ACCESS PATH.....	32
FIGURE 22: VTAM - OBJECT ACTIVITY.....	33
FIGURE 23: VTAM - DATABASE ACTIVITY	34
FIGURE 24: VTAM - SPACENAME ALLOCATIONS.....	35
FIGURE 25: VTAM - OBJECT ALLOCATION SUMMARY.....	36
FIGURE 26: VTAM - VOLUME ACTIVITY	36
FIGURE 27: TEP - DB2 CONNECT MONITORING	38
FIGURE 28: TEP - DB2 CONNECT MONITORING, SQL STATEMENT TIME.....	39
FIGURE 29: PERFORMANCE WAREHOUSE - REPORT STEP PROPERTIES	41
FIGURE 30: PERFORMANCE WAREHOUSE GUI - REPORT STEP PROPERTIES	44

4 Introduction

OMEGAMON XE for DB2 Performance Expert on z/OS is a performance monitoring and analysis tool that simplifies DB2 performance tuning.

This chapter describes the environment in which OMEGAMON XE for DB2 Performance Expert on z/OS monitors Content Manager workload. It shows how the two products fit together to analyze the CM activities on DB2 for z/OS. Figure 1 shows the scenario and test installation setup that is based on a real-life production system. It describes the different CM components and gives you an impression how OMEGAMON XE for DB2 Performance Expert on z/OS can be used as a monitoring and analyzing tool for CM purposes.

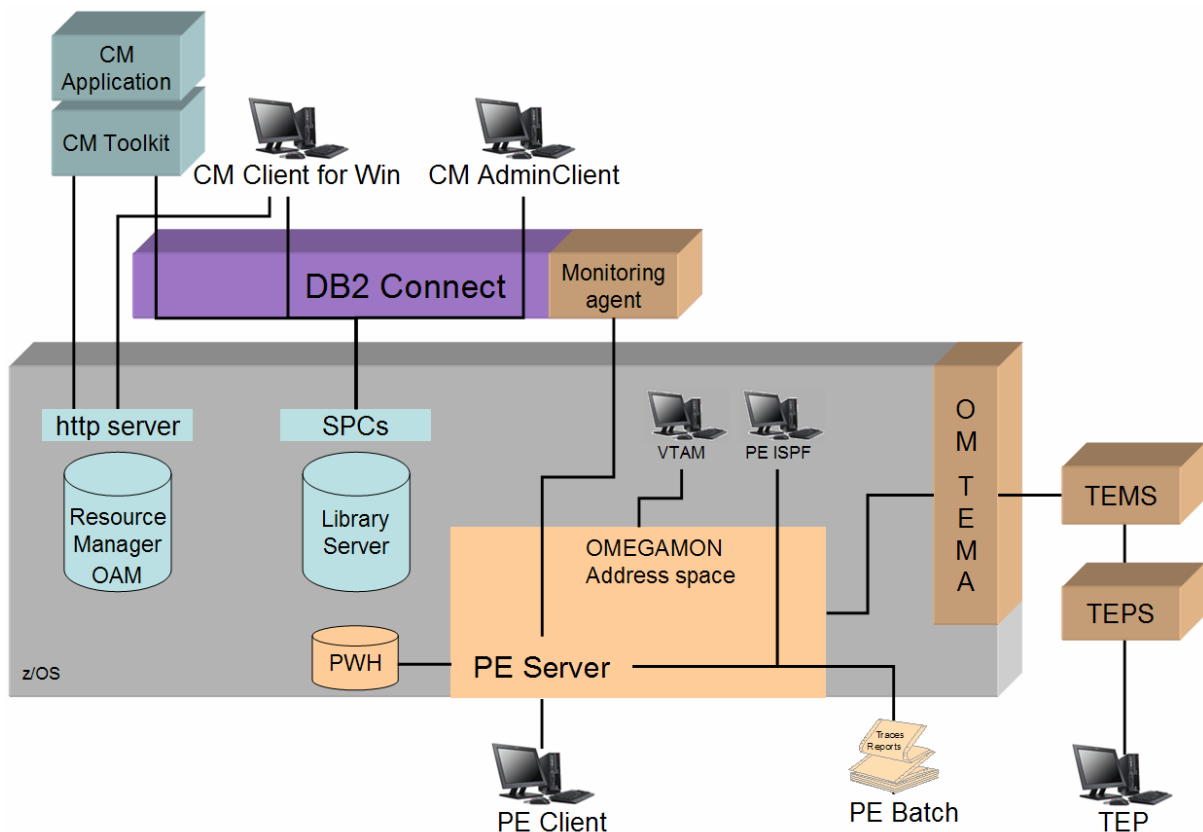


Figure 1: Content Manager and OMEGAMON XE for DB2 Performance Expert on z/OS setup

4.1 The Content Manager Environment

A Content Manager environment on z/OS includes several components that can interact in different ways.

The basic components are:

- CM Library Server (Library Server)
- CM Resource Manager (Resource Manager)
- IBM HTTP server
- Object Access Method (OAM)
- Tivoli Storage Manager (TSM)



4.1.1 CM Library Server

The Library Server in a Content Manager system is the most important component besides the Resource Manager. You can compare the Library Server tasks with a real-life catalog file in a public library. The Library Server manages indexing information, access rights, the relationship with other objects, and the status of documents that are physically stored in the Resource Manager. If a client searches for a document to retrieve, it connects to the Library Server on which the metadata about the document is stored. This metadata is represented as searchable attributes of a specific type of document. These attributes are stored in DB2 tables of the Library Server database.

It is important to manage objects in the Library Server efficiently. Missing indices and badly created referential integrity can affect the performance of the whole system. Because there is only one Library Server in a Content Manager system, it often turns out as a bottleneck.

OMEGAMON XE for DB2 Performance Expert on z/OS can be used to analyze the DB2 database that contains the Library Server tables to increase the throughput and improve performance. This topic is explained in Chapter 6.1.

4.1.2 CM Resource Manager

The Resource Manager physically stores content that is logically managed by using attribute data in the Library Server. When a user retrieves an object from CM, the application first connects to the Library Server to find the correct document according to its attributes. The Library Server returns the actual location of the document. This directs the Content Management application to the Resource Manager that is responsible for storing the document.

The Content Management application then retrieves the object from the Resource Manager through an HTTP request. On z/OS, this HTTP request is handled by the HTTP Server.

The Resource Manager uses these products on z/OS:

IBM DB2 Universal Database

The Resource Manager requires a DB2 Universal Database and creates its database during installation. This database does not contain the archived documents themselves but it is required for internal Resource Manager processing and administration. The documents are stored in OAM or TSM.

You can have the Library Server and the Resource Manager share the same DB2 subsystem if this is applicable for your environment.

Note: Performance analysis of Resource Manager databases is not described in this document. You can, however, also apply the mentioned techniques to Resource Manager tuning.

Object Access Method (OAM)

The Resource Manager can use OAM to store and retrieve documents of a Content Manager system. OAM is a collection of one or more storage groups, where each storage group is a separate DB2 database.

OAM divides an object into segments and stores them as rows in a DB2 table. In the sample environment, all documents that are smaller than 4 KB are stored completely in one table. Bigger documents are divided into 32 KB blocks and are stored row by row in another table.

OAM also provides the option to set up DFSMS storage classes and use the data management features of Data Facility Storage Management Subsystem (DFSMS).

IBM Tivoli Storage Manager for z/OS

With Content Manager Version 8.3, you can also use IBM Tivoli Storage Manager for z/OS instead of OAM. Tivoli Storage Manager provides long-term data storage. TSM is not covered in this document.

**IBM HTTP Server for z/OS**

A Content Manager environment on z/OS requires HTTP Server for z/OS to run the Resource Manager CGI program. While retrieving an object, the Library Server returns a URL to the client that links to the HTTP server. When accessing this URL, it leads to the actual document.

4.1.3 CM System Administration Client

The System Administration Client administrates the Content Management environment. The System Administration Client can run on AIX, Linux, Solaris, and Windows operating systems. The System Administration Client is used by Content Manager administrators and not by users who work with real data and documents that should be stored in the CM environment.

After logging on to the client, the following tasks can be performed:

- Configuration of Library Server parameters (Logon, Tracing, User settings)
- Configuration of Resource Managers (z/OS: MVS Collection properties)
- Defining user and group accounts
- Data modeling (define item types and their properties, attributes)
- Document routing

Note that the CM System Administration Client and the CM Client for Windows deploy the CM Toolkit to access the Library Server and any Resource Manager.

4.1.4 CM Client for Windows

The Content Manager Client for Windows is an application that enables users to work with their Content Management system. The Client for Windows offers storing, searching, retrieving, scanning, viewing, and similar tasks that are ready for you to use out-of-the-box. Typically, you use the Client to handle scanned documents.

You can work with different item types, create folders to structure the electronic content, and analyze your documents in the CM system.

If accessing CM documents is part of a business process, you should consider developing your own applications by using the Content Manager Toolkit.

The Client for Windows application is installed on a Windows system and accesses the CM Library Server through DB2 Connect. Tasks that are performed through the Client for Windows can be monitored and analyzed by using OMEGAMON XE for DB2 Performance Expert on z/OS.

4.1.5 CM Toolkit for application development

You can develop customized DB2 Content Manager applications by using client APIs that are provided through the DB2 Content Manager Toolkit. The DB2 Content Manager Toolkit provides connectivity to a CM system. Thus, it is an alternative to the Windows Client and eClient to search, archive, and store contents. When using the Toolkit, you must also use the Toolkit API to develop a CM application.

User-written CM software often causes performance problems because application developers do not consider database specifics.

OMEGAMON XE for DB2 Performance Expert on z/OS helps to identify programming practices that result in bad performance.

4.2 OMEGAMON XE for DB2 Performance Expert on z/OS

It is important to understand the different components of OMEGAMON XE for DB2 Performance Expert on z/OS and their responsibilities within a specific environment. People might be familiar with some modules of this product but not with all of them. This section describes the components and explains how they help to monitor a Content Manager environment.



4.2.1 OMEGAMON Server

The OMEGAMON Server is an address space that is responsible for collecting DB2 Universal Database on z/OS performance data. You need one address space per LPAR to monitor DB2 performance.

In other words, this address space is responsible for collecting performance data of DB2 databases that are used by the Library Server and Resource Manager on z/OS. The client applications that visualize the performance data that is provided by the address space are described in the following sections.

4.2.2 Connect Monitoring Agent

OMEGAMON XE for DB2 Performance Expert on z/OS monitors not only the DB2 for z/OS database server instances but also DB2 Connect instances running on distributed platforms. Because most CM clients connect from non-z/OS application servers, this type of distributed application monitoring is also explained in this document.

Connect Monitoring Agents must be installed on each DB2 Connect instance. Performance data of these agents is stored in a DB2 database on the host. The data can be analyzed from Performance Expert Client, the OMEGAMON Classic interface, or the Tivoli Enterprise Portal.

Monitoring DB2 Connect instances can help to identify performance bottlenecks, such as network problems or connection issues.

How to monitor DB2 Connect by using the Tivoli Enterprise Portal is described in Chapter 6.3.

4.2.3 Performance Expert Client

Performance Expert Client is an interface to monitor DB2 performance from the workstation. Performance Expert Client is most suitable for online monitoring and reporting, Performance Warehouse management, and analyzing buffer pools. It provides an in-depth look into DB2 performance metrics. You can analyze the following activities:

- Online monitoring of open DB2 Connections (Threads)
- SQL activity tracing of DB2 Threads
- Monitoring of database statistics and system parameters
- Analyzing DB2 Connect gateways and their connection to DB2
- Generating Accounting and Statistics reports by using Performance Warehouse
- On-demand collection of DB2 performance data for report generation
- Loading of Accounting and Statistics data into the Performance Warehouse database and analysis of Performance Warehouse data by using SQL queries and rules-of-thumb
- Analyzing buffer pools and their hit ratio
- Defining exception notifications

Performance Expert Client enables CM administrators to perform in-depth analysis of DB2-related CM performance problems. Especially users who do not work in a mainframe environment prefer Performance Expert Client to ISPF or VTAM interfaces because of its graphical user interface. It contains graphs, charts, and clearly laid out tables, which you can handle with the mouse. Because CM administrators usually do not work in a mainframe environment, they like to work with Performance Expert Client.

4.2.4 Performance Expert ISPF interface

The Interactive System Productivity Facility (ISPF) Monitoring Dialogs provide access to the main functions or components of OMEGAMON XE for DB2 Performance Expert on z/OS from TSO/ISPF. You can use this monitoring interface without installing a client on a separate machine. You can access the ISPF interface directly through Time Sharing Option (TSO).

Using the ISPF Online Monitor allows you to run batch reports, display DB2 thread activity with detailed performance data, and view DB2 system parameters. The ISPF interface also provides exception profiling that is based on threshold data sets. These data sets are analyzed periodically. The ISPF interface is not explained further in this document.



4.2.5 OMEGAMON Classic

The OMEGAMON Classic interface is a Virtual Telecommunications Access Method (VTAM) application that provides real-time analysis of a DB2 subsystem. In this document, the VTAM interface is used for monitoring DB2 objects. It provides information about table space usage through the option Object Analysis.

You can use the OMEGAMON Classic interface to:

- Monitor DB2 statistics and threads
- Perform DB2 Connect Monitoring
- Observe counter thresholds
- Collect and show near-term history data
- Trace the application or DB2 plan activity
- Analyze objects and drill down into object allocation
- Analyze the usage of thread activities
- Analyze volume allocation and activities
- Receive detailed runtime performance information on thread, unit of work, program, and SQL statement level by using ATF
- View enclave information from Workload Manager (WLM)

This list also shows that the OMEGAMON Classic interface is the 3270 counterpart to Performance Expert Client on the workstation. Therefore, CM administrators who work with a 3270 interface might prefer the OMEGAMON Classic interface.

4.2.6 Performance Expert Batch Reporting

The interfaces described so far provide DB2 online monitoring facilities. In contrast, the Batch Reporting function of OMEGAMON XE for DB2 Performance Expert on z/OS processes DB2 performance trace data, for example, SMF data. The analysis of this offline (historical) data provides the most granular level of insight into DB2 performance. Using batch reports allows you to determine what happened in your DB2 system.

This document shows how to use Batch Reporting to generate SQL activity traces. It also shows how detailed this kind of information is (refer to Section 6.2.3).

For more information about Batch Reporting, refer to the OMEGAMON XE for DB2 Performance Expert on z/OS Redbook.

4.2.7 Tivoli Enterprise Portal

The Tivoli Enterprise Portal is a browser-based or Java-based interface that offers a single point of view for online monitoring and analysis. This monitoring overview contains not only DB2 but also other subsystems of z/OS. By using TEP as monitoring client, you get an overall view of DB2 system health end-to-end with other products of your environment.

The strength of TEP is landscape monitoring. For in-depth DB2 analysis, such as thread monitoring or SQL activity tracing, Performance Expert Client or the VTAM interface are more appropriate.

The Tivoli Enterprise Monitoring Agent (TEMA) collects performance data from managed systems and communicates with a Tivoli Enterprise Monitoring Server (TEMS). There are several agents for different subsystems on the host.

TEMS acts as the central collector for OMEGAMON XE for DB2 Performance Expert on z/OS data. This data is distributed to the Tivoli Enterprise Portal Server (TEPS). TEMS can also evaluate situations and send alerts to TEPS if specified conditions are met.

The Tivoli Enterprise Portal Server offers the collected data from Tivoli Enterprise Monitoring Servers to clients. The Tivoli Enterprise Portal Server holds the information that is required to build workspaces in the Tivoli Enterprise Portal client. You can structure and customize workspaces in the client by using graphs, tables, or charts.

Chapter 6.3 of this document describes how to use the TEP framework to analyze DB2 Connect performance data and connection issues.

5 A DB2 view on your CM environment

A Content Management environment is a complex system handling objects, referential integrity between objects, user accounts, and access control lists. The following sections describe how important CM concepts, such as *item types* map to DB2 concepts. This information is essential to guide your CM problem analysis to the appropriate DB2 areas.

Moreover, you get general recommendations that help you achieve good performance without monitoring the CM-related database in detail.

5.1 Installation parameters

Each JCL and data input member in the Library Server and Resource Manager libraries have tokens. Tokens are marked by a question mark (?) at the beginning and end of the variable field. The Library Server provides an ISPF edit macro, *ICMCLIST*, in the *SICMINS1* data set for the Library Server and *ICMMRMCL* in the *SICMINS2* data set for the Resource Manager.

The macro contains the tokens that are used in a basic Library Server configuration. If your naming conventions are consistent throughout your environment, you can customize this edit macro for your environment.

Figure 2 shows an extract of available tokens. To get a full list of the parameters, refer to Appendix A.

Token name	Description	Example
?ACLPLAN?	DB2 plan for ICMPCACL.	ICMAACLPL
?BP32?	Buffer pool for table space ICMLFQ32 and ICMLNF32.	BP32K
?BPS4?	Buffer pool for library server tables.	BP4
?BPV4?	Buffer pool for library server indexes.	BP4
?CREATOR?	ID of the creator defining the DB2 objects.	ICMCRTOR
?DB2CONNECTID?	User ID of the DB2 connect ID. It must be a TSO user ID with DB2 user privileges. You can use this user ID as the DB2 connect ID for DB2 Content Manager Version 8 users who do not have a TSO user ID on z/OS.	ICMCONT
?DB2PKGCOLLID?	Collection name of the DB2 package	ICMMLS
?ICM?	High-level qualifier of the DB2 Content Manager Version 8 library.	ICMV830

Figure 2: Sample CM installation tokens

OMEGAMON XE for DB2 Performance Expert on z/OS can monitor storage constraints. This means that you can monitor buffer pool usage and buffer pool hit ratio to determine if your buffer pool settings are good or if you must modify your settings. Buffer Pool Analyzer is explained in Section 6.2.1.

5.2 Modifying Content Manager DDL

In a Content Manager environment, DB2 objects are created dynamically if a new type of document, also called *item type*, is added to the CM environment. If a CM administrator creates a new *item type*, CM builds a Data Definition Language (DDL) to add new DB2 objects to the Library Server database. How *item types* relate to DB2 tables is explained in Section 5.4.



By default, the tables that are created for each *item type* are created in the same table space. The number of tables that DB2 table spaces can contain is limited. If you plan to use a large number of documents for an item type, you should create the tables for each *item type* in their own table space. This approach can help to manage performance, capacity, backup, and recovery of your *item types*. The same applies for creating indexes on these tables.

The *ICMST390Control* table specifies the optional DB2 clause or clauses to append to dynamically generated CREATE TABLE and CREATE INDEX statements when creating user tables and indexes for user tables.

Some examples for using the *ICMST390Control* table include:

- Specifying a single table space name where all tables are to be placed
- Specifying that each new item type is to be placed in its own table space
- Indicating the "Percent Free Space" value when creating indexes on item type tables

The layout of the *ICMST390Control* table is:

```
OPTIONKEY CHAR(5)
OPTIONTEXT VARCHAR(512)
```

OptionKey is the key for the table. It can have one of these values: TABLE, INDEX, or RDRIN. Depending on the value of OptionKey, OptionText has different values.

If OptionKey is set to TABLE, OptionText has a string that indicates which z/OS table space should contain the item type table. For example, OptionText might be set to *IN ICMNLSDB.TBLSPC1*.

If OptionKey is set to INDEX, OptionText has a string that contains one or more options for the CREATE INDEX SQL statement. For example, OptionText might contain *PCTFREE 35*.

Section 6.2.5 describes how you can detect table spaces with high load, for example, with high I/O ratios. This helps to adjust the assignment of tables to table spaces, and to find I/O statistics about different volumes in your installation.

Thus, you can decide how to specify entries in the *ICMST390Control* table.

For more information, refer to the *DB2 Content Management Version 8.3 Information Center*.

5.3 Object Access Method (OAM)

This document is not intended to provide a tuning reference for the OAM installation in your environment. If your Content Manager installation uses OAM for storing Resource Manager objects, you should invest in careful planning, initial setup, and maintenance.

Some considerations for implementations with OAM include these questions:

- What is the size of my objects?
- How many object storage groups are required?
- Which size should these object storage groups have?
- Which SMS management and storage class criteria are applicable for these groups?
- Is tape or optical used for storing the objects?

OAM is a collection of one or more storage groups, where each storage group is a separate DB2 database. The number of storage groups that a particular implementation requires can vary significantly.

For detailed information, refer to the OAM planning and administration publications.

For additional information about OAM, refer to the IBM z/OS: DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support, SC35-0426.



OMEGAMON XE for DB2 Performance Expert on z/OS can monitor the OAM databases like all other databases. You can analyze buffer pool effectiveness, DB2 objects and their I/O rates, and storing and retrieving data into OAM. It is important to understand how Content Manager items that are logically cataloged in the Library Manager map physically to the OAM databases and tables. How to find items in OAM by using information in the Library Server with DB2 DML, is explained in section 5.5.

5.4 Exploring related DB2 tables

A Content Manager environment contains many different tables. This includes system tables and user-created tables. If a new *item type* is created, DDL statements are generated and processed, which results in new tables in DB2. Use the following statements to determine which tables belong to your *item types*.

1. Issue the following statement to determine the *itemtypeid* of a given item type:

```
SELECT keywordcode FROM icmv830.icmstnlkeywords WHERE keywordclass = 2 AND  
languagecode = <languagecode> AND keywordname = <itemtype description>
```

The statement returns the *keywordcode* that is automatically assigned to the keyword by Content Manager. When a new keyword string is created, Content Manager assigns a unique code to it. Column *keywordclass* specifies the type of keyword. *Keywordclass = 2* determines an item type. If you search for an attribute, specify *keywordclass = 1*.

You can have different national languages in your CM installation. For each language in CM, the keywords are duplicated. The 3-character *languagecode* determines the language that you use for the *keywordname*. For more details, refer to the description of ICMSTNLSLANGUAGES in the Content Manager Information Center.

Column *keywordname* specifies the name of the keyword in the national language. This represents the item type name.

2. Issue the following statement to determine the *componenttypeid*

```
SELECT componenttypeid FROM icmv830.icmstcompdefs WHERE parentcomptypeid = 0 AND  
itemtypeid = <itemtypeid>
```

The statement returns the *componenttypeid* that is a unique identifier for the component type. When a new component table is defined, CM generates this unique value. Use this ID to identify the tables that belong to your item type.

The *parentcomptypeid* is used to build a type hierarchy among component types. If this value is set to 0, it points to the root table of the hierarchy.

The *itemtypeid* is the code generated by CM for each item type. You determine this ID with the SELECT statement described in 1.

Each item type has a corresponding table in DB2 where all item attributes and meta data are stored. If the *componenttypeid* of an item type is XXXX, the table name is:

```
ICMV830.ICMUT0XXXX001
```

Example for the sample environment:

To find the *keywordcode* of our item type *CSLDMailArchiv2*, the following statements are issued:

```
SELECT keywordcode FROM icmv830.icmstnlkeywords WHERE keywordclass = 2 AND  
languagecode = 'ENU ' AND keywordname = 'CSLDMailArchiv2'
```

This statement returns 1029 as *keywordcode* (*itemtypeid*).

SELECT componenttypeid FROM icmv830.icmstcompdefs WHERE parentcomptypeid = 0 AND itemtypeid = 1029

This statement returns 1047 as componenttypeid.

The DB2 table that stores meta data of item type *CSLDMailArchiv2* is *ICMV830.ICMUT01047001*

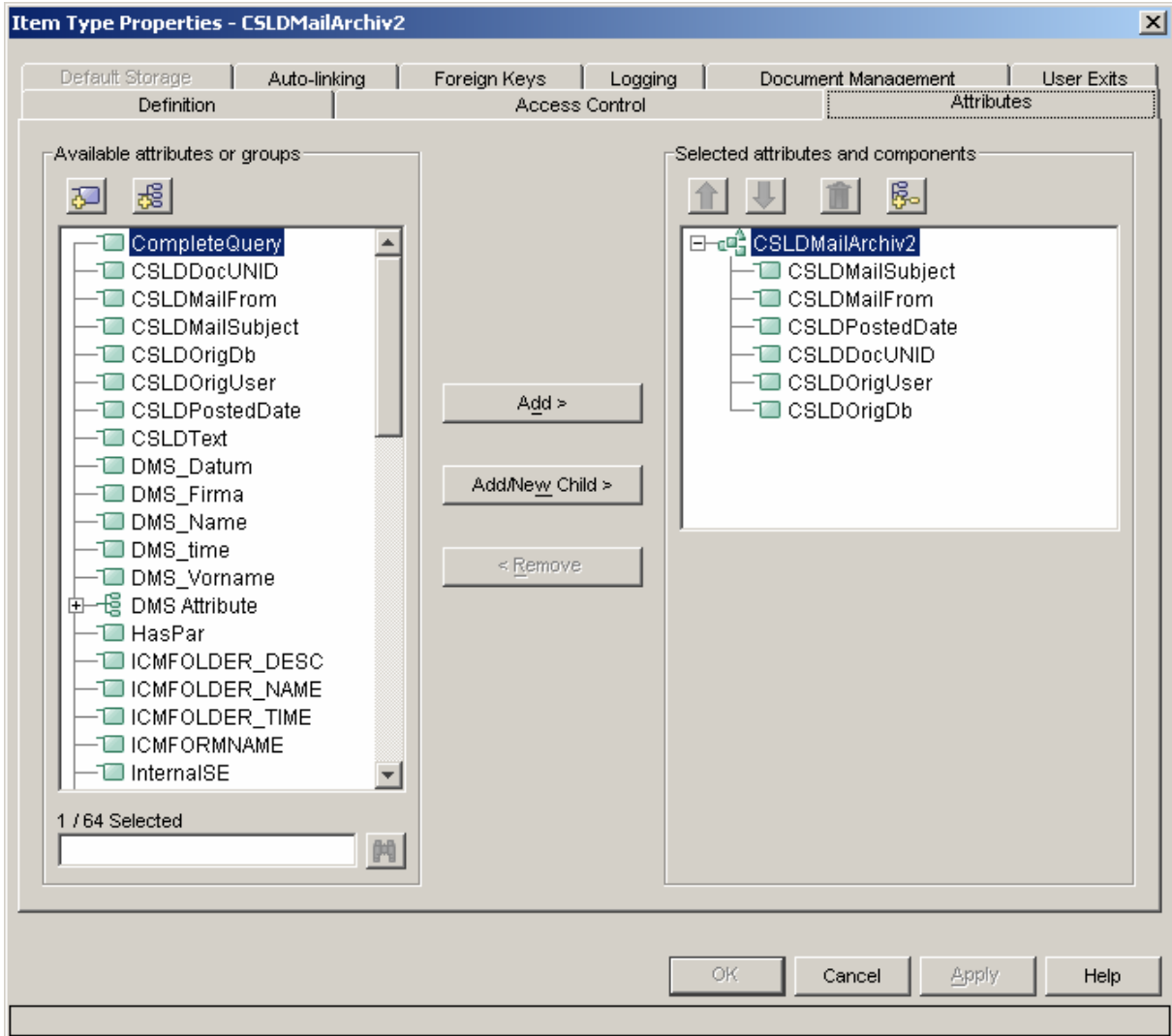


Figure 3: Item Type Properties of CSLDMailArchiv2

Item Type *CSLDMailArchiv2* in the sample environment is defined with attributes as displayed in Figure 3 . The item type is represented regarding DB2 objects as follows:

<i>Content Manager Object</i>	<i>Represented in DB2 as</i>
Item type CSLDMailArchiv2	Table ICMUT01047001
Attribute CSLDMailSubject	Column ATTR0000001045
Attribute CSLDMailFrom	Column ATTR0000001046
Attribute CSLDPostedDate	Column ATTR0000001047
Attribute CSLDOrigUser	Column ATTR0000001048
Attribute CSLDOrigDb	Column ATTR0000001049
Attribute CSLDDocUNID	Column ATTR0000001050

5.5 Exploring related tables in OAM

This section explains how to find CM item types in DB2 tables of OAM. It is assumed that you use OAM to store documents. For the sample environment, the necessary steps are:

1. Explore the MVS collection that you use for the item type. This is configured at item type level. You can browse the CM System Administration Client for these properties:

Select your *DB2 instance* → *Data Modeling* → *Item Types* → *CSLDMailArchiv2*:

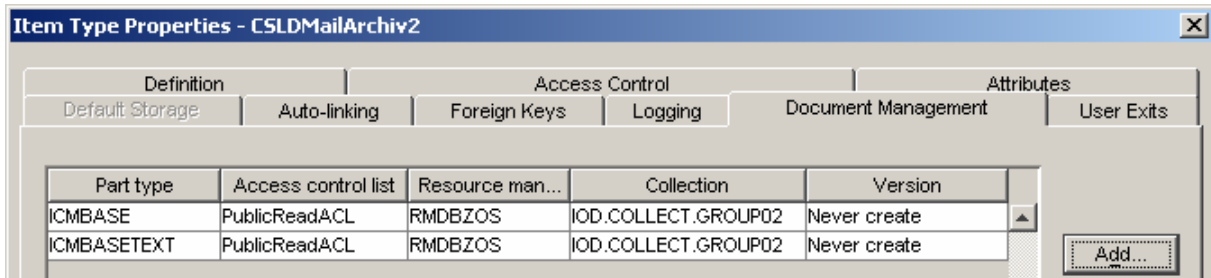


Figure 4: Item Type and MVS Collection names

Our MVS Collection is *IOD.COLLECT.GROUP02*, which maps to the DB2 database *GROUP02*. For more information about OAM administration, refer to the *DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426

For the sample environment, the database *GROUP02* looks as follows:

```

-DIS DB(GROUP02) SPACE(*) LIMIT(*)

***** Top of Data *****
DSNT360I -DDVG *****
DSNT361I -DDVG * DISPLAY DATABASE SUMMARY
          * GLOBAL
DSNT360I -DDVG *****
DSNT362I -DDVG DATABASE = GROUP02 STATUS = RW
          DBD LENGTH = 4028
DSNT397I -DDVG
NAME      TYPE PART STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
OSMDTS   TS          RW
OSMOTS04 TS          RW
OSMOTS32 TS          RW
OBJDIRX1 IX          RW
OBJDIRX2 IX          RW
OBJDIRX3 IX          RW
OBJT04X1 IX          RW
OBJT32X1 IX          RW
***** DISPLAY OF DATABASE GROUP02 ENDED *****

```

Figure 5: Display OAM database in DB2

All documents with a size greater than 4KB are stored in table space *OSMOTS32* in table *OSM_4K_OBJ_TBL*. Documents with a size greater than 32KB are divided into 32KB blocks and stored row by row in table *OSM_32K_OBJ_TBL*.

2. Identify the itemid of an item in the Library Server

For the sample environment, the item type table in the Library Server looks as follows (for details about how to find item type tables in Library Server, refer to 5.4).



Table *ICMUT01047001* contains the following columns:

<i>Name</i>	<i>Data Type</i>	<i>Length</i>	<i>CM Attribute</i>
COMPCLUSTERID	INTEGER	4	
COMPONENTID	CHAR	18	
ITEMID	CHAR	26	
VERSIONID	SMALLINT	2	
ACLCODE	INTEGER	4	
SEMANTICTYPE	INTEGER	4	
EXPIRATIONDATE	DATE	4	
COMPKEY	VARCHAR	23	
CREATETS	TIMESTAMP	10	
CREATEUSERID	CHAR	32	
LASTCHANGEDTS	TIMESTAMP	10	
LASTCHANGEDUSERID	CHAR	32	
ATTR0000001045	VARCHAR	254	CSLDMailSubject
ATTR0000001046	VARCHAR	100	CSLDMailFrom
ATTR0000001047	TIMESTAMP	10	CSLDPoastedDate
ATTR0000001050	CHAR	32	CSLDDocUNID
ATTR0000001048	VARCHAR	254	CSLDOrigUser
ATTR0000001049	CHAR	17	CSLDOrigDb

3. Select the itemid of a certain item in the Library Server

```
SELECT itemid FROM icmv830.icmut01047001 WHERE attr0000001045 = 'Hello Pete' AND  
attr0000001046 = 'Mister Q' AND attr0000001048 = 'CMPE'
```

This returns the itemid = A1001001A06D05B00543B06701

This itemid is also referenced in the OAM table. In the example, it is contained in table *GROUP02.OSM_32K_OBJ_TBL* in column *OTNAME*. The next statement receives the 32k blocks of data for version 1 of the specified document.

```
SELECT oobj FROM group02.osm_32k_obj_tbl WHERE otname =  
'A1001001.A06D05.B00543.B06701.V001'
```

6 Monitoring Content Manager

To meet the performance objectives of the Content Manager administrator, and to ensure that the users of Content Manager experience good response times without errors, it is important that your Library Server and Resource Managers work correctly.

This section covers the following CM tuning scenarios:

- Monitoring system health of Library Server and Resource Manager databases
- Improving the I/O performance of your CM environment by using Buffer Pool Analyzer and Object Analysis
- CM workload performance analysis by using SQL activity and Visual Explain
- DB2 Connect and networking (TCP/IP) monitoring by using TEP
- Using Performance Warehouse to perform capacity planning and performance trend analysis

6.1 System Overview

OMEGAMON XE for DB2 Performance Expert on z/OS allows monitoring the overall health of a DB2 subsystem by using different options. This document covers the options *System Health* and *Exception Processing* of Performance Expert Client.

6.1.1 System Health

Monitoring a DB2 subsystem by using *System Health* gives an overall impression of the DB2 part of your Content Manager environment. *System Health* gives administrators a feeling about the overall status of DB2. Monitoring by using *System Health*, lets them perform trend analysis, and detect potential performance issues. An example for trend analysis is the *Dynamic Statement Cache* effectiveness graph of *System Health*. If you detect the trend that the cache hit ratio of your system decreases over time, you should have a closer look at your dynamic statement cache settings by using the in-depth performance monitoring features of OMEGAMON XE for DB2 Performance Expert on z/OS. For more details about dynamic statement cache and dynamic SQL, refer to Section 6.2.4.

System Health graphs are graphical data views that show how critical performance counters change over time. You can use this information to verify if your Library Server or Resource Manager works correctly, or to detect performance problems immediately. The data views are defined in the *System Health* component of OMEGAMON XE for DB2 Performance Expert on z/OS. You can use predefined data views or define your own data view. More critical data views can be directly stored in the *System Overview* window by defining them in the predefined System Overview folder. This allows you to see immediately how your system works after you log on.

Figure 6 shows sample System Health graphs that should be monitored in a Content Manager environment. The figure shows the graphs in the *System Overview* section of the DB2 subsystem that runs the Library Server and Resource Manager.

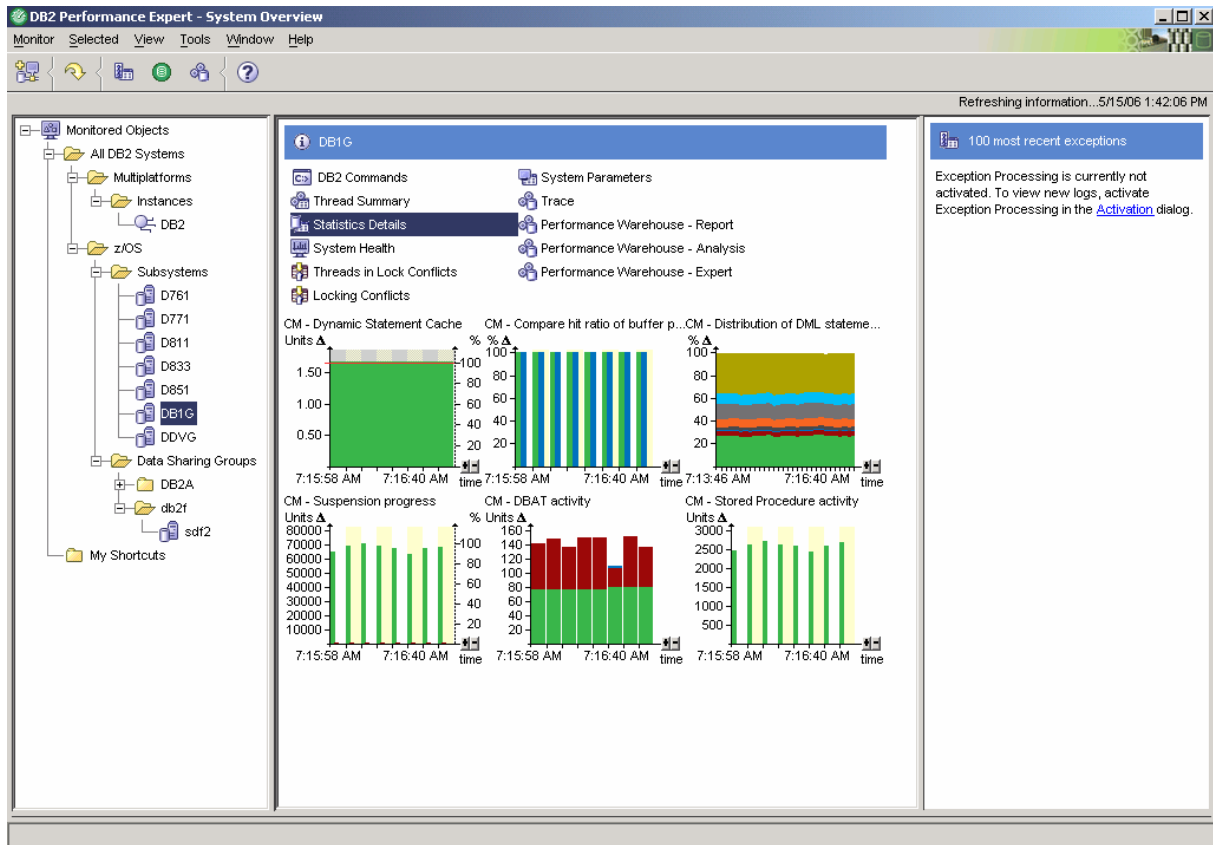


Figure 6: DB2 Performance Expert Client and System Overview

Dynamic Statement Cache

This data view displays the effectiveness of your dynamic statement cache over time by using the counter *Global Cache Hit Ratio (%)*. See also Section 6.2.4 for more information on this topic.

Compare the hit ratio of buffer pools

This predefined data view shows the hit ratio percentage for each buffer pool. Buffer pool hit ratios should be between 85% and 100%. If the hit ratio is considerably lower when you perform the usual workload, you might consider increasing the buffer pool sizes or do some additional buffer pool tuning. See also Section 6.2.1 for more information about buffer pools.

Distribution of DML statements

This predefined data view shows the distribution of different Data Manipulation Language (DML) statements, for example, SELECT, INSERT, UPDATE, or DELETE. You can use it to investigate changes in workload characteristics over time. See also Section 6.4.1 for more information on this topic.

Suspension progress

This predefined data view shows the total number of lock requests and how often a lock request causes a suspension. It also shows how many lock requests result in a timeout or a deadlock. Locking conflicts, deadlocks, and timeouts are critical to the overall performance of Library Server and Resource Manager databases.

DBAT activity

A Data Base Access Thread (DBAT) is a DB2 thread that processes DB2 workload on the host from a remote location. This user-defined data view shows the evolution of the following DBAT- related DB2 performance counters: *Active DBATs – current*, *DBATs not in use – current*, *DBATs created*.

Because most of the requests to the Library Server or Resource Manager databases are issued from remote workstations, monitoring DBAT activity can help you decide if your DB2 connection concentrator and connection pooling settings are optimal. See also Section 6.3 for more information on this topic.

Stored procedure activity

This user-defined data view shows the evolution of the following stored-procedure-related DB2 performance counters: *SP-CALL statements executed*, *SP procedure abends*, *SP-CALL statement timeouts*, *SP-CALL statements rejected*. Monitoring stored procedure performance is essential in a CM environment because the Library Server uses stored procedures to process any user request to the Library Server database.

Figure 7 shows the System Health for a Content Manager environment. The data views in this group can be used to fine-tune some database system parameters of your Library Server and Resource Managers.

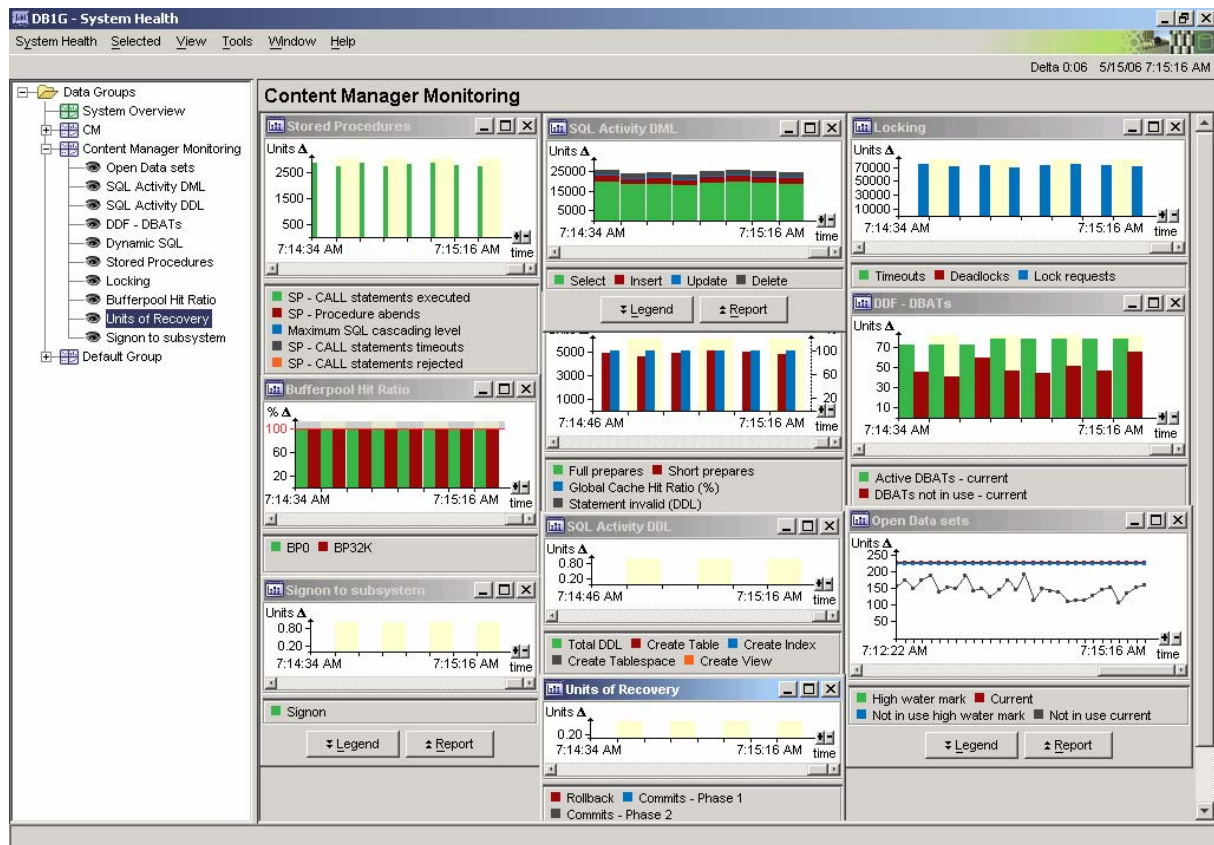


Figure 7: Performance Expert Client - System Health graphs

6.1.2 Exception Processing

OMEGAMON XE for DB2 Performance Expert on z/OS lets you perform monitoring by exception. OMEGAMON Server periodically compares the warning and problem thresholds that are set on DB2 snapshot values or derived snapshot values against actual snapshot values. Exceptions are raised on



Performance Expert Client if a threshold is exceeded. Thresholds on different snapshot values can be combined in threshold sets. OMEGAMON XE for DB2 Performance Expert on z/OS offers predefined threshold sets that can be applied as is or adapted to your own needs.

6.2 Analyzing DB2 performance

6.2.1 Buffer pool

In case of bad response times for queries against the Content Manager, check the quality of your buffer pool configuration and the buffer pool hit ratio. A bad hit ratio means that a lot of I/O is performed to return data to applications. Reading data from data sets is much slower than reading directly from the buffer pool.

The buffer pool settings of your Content Manager System are configured during the CM installation process through replacement tokens. Refer to Section 5.1 for further details.

Token name	Description	Example
?ACLPLAN?	DB2 plan for ICMPCACL.	ICMACLPL
?BP32?	Buffer pool for table space ICMLFQ32 and ICMLNF32.	BP32K
?BPS4?	Buffer pool for library server tables.	BP4
?BPV4?	Buffer pool for library server indexes.	BP4
?CREATOR?	ID of the creator defining the DB2 objects.	ICMCRTOR
?DB2CONNECTID?	User ID of the DB2 connect ID. It must be a TSO user ID with DB2 user privileges. You can use this user ID as the DB2 connect ID for DB2 Content Manager Version 8 users who do not have a TSO user ID on z/OS.	ICMCONT
?DB2PKGCOLLID?	Collection name of the DB2 package	ICMMLS
?ICM?	High-level qualifier of the DB2 Content Manager Version 8 library.	ICMV830

Figure 8: CM installation tokens

To determine active buffer pools in your system, check your installation jobs or check them online in DB2. For the sample environment, the following buffer pools are used:

Buffer pool	Table space	Description
BP32K	ICMLFQ3	Holds the large, persistent, "frequently used" tables – most notably, all item type tables. When a new item type is defined, the system administration client defines the new tables (ICMUT0XXX001) in this table space.
	ICMLNF32	Holds the large but less frequently used tables, such as event logs, versions, and replicas. If you do not use any of these features, this table space does not require much space.
BP0	ICMVFQ04	Holds the "volatile" tables, the size of which is usually small but can considerably vary, and for which both updates and deletions are very common. For example, the tables associated with document routing "in progress" items, checked out items, and so on.
	ICMSFQ04	Holds the small "frequently used" but seldom updated tables, such as system information, user definitions, ACLs, and so on, that do not use much space but are always required in memory for good performance.

A good hit ratio of BP32K is a key for good performance of queries against the Library Server. After adding new item types, you should monitor the hit ratio to determine if the buffer pool can still handle the additional new amount of data. If you consider increasing the buffer pools, this is the buffer pool that could exploit the extra space best.

BP0 holds tables that often vary in size. Receiving a good hit ratio for this buffer pool is difficult because the content is volatile.

Figure 9 shows the buffer pool hit ratio window of Performance Expert Client in the sample environment.

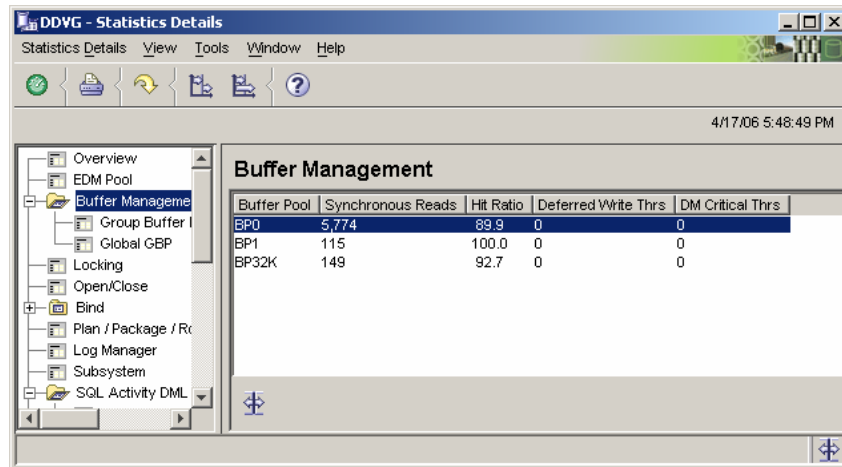


Figure 9: Performance Expert Client - buffer pool hit ratio

If query response times are not sufficient although query access paths are tuned, the buffer pool might be too small. OMEGAMON XE for DB2 Performance Expert on z/OS offers various options for monitoring buffer pools. For more information about access paths, refer to Section 6.2.2. For more information about buffer pools, refer to the Redbook *IBM DB2 Performance Expert for z/OS Version 2*.

You can use Buffer Pool Analyzer that helps to manage buffer pools more efficiently. Buffer Pool Analyzer is integrated in OMEGAMON XE for DB2 Performance Expert on z/OS and supports object placement and simulation.

Buffer Pool Analyzer shows you which tables reside in which table space and which table space uses which buffer pool. The object placement component of Buffer Pool Analyzer uses specific DB2 for z/OS functions (ALTER BUFFERPOOL) to improve the performance. The basic idea is that you can assign objects that were accessed in a similar way to the same buffer pool and tune this buffer pool for this kind of access, for example, predominantly sequential access versus predominantly random access.

Object placement assigns objects to buffer pools. The assignment process uses performance metric patterns that are compared to previously collected performance data.

The simulation component of Buffer Pool Analyzer lets you simulate different object-to-buffer-pool assignments without changing the original system. You can also simulate resized buffer pools and use Buffer Pool Analyzer as a sizing tool to find the best performance.

6.2.2 Access Paths

DB2 optimizer needs database statistics to calculate good access paths for queries against the Library Server database. The Content Manager Tuning Guide recommends to collect statistics by using RUNSTATS and keep the access paths updated by rebinding plans and packages. In a CM environment, outdated statistics are indicated if you experience high sorting activity together with sort overflows, or if you discover a decrease in stored procedure throughput. Monitoring stored procedure throughput with the Performance Warehouse component is explained in Section 6.4.

The Library Server and Resource Manager databases are installed with appropriate indexes on the tables to optimize execution of the standard “CRUD” (Create, Retrieve, Update, and Delete) operations. When you define a new item type, the appropriate indexes on the new tables are also created automatically. If you run specific searches by using user-defined attributes for your item types, you should consider creating additional indexes that are based on these attributes to improve response times and reduce Library Server resource usage for queries. You should, however, also consider that an index can affect response times for create, update, and delete operations for this item type, as well as increase database size and buffer pool usage. Therefore, you should monitor overall system performance and response times before and after adding an index to evaluate its overall value.

OMEGAMON XE for DB2 Performance Expert on z/OS provides several functions to display the access paths of a query. You can for example display the access path of statements in the dynamic statement cache (see Section 6.2.4) or display the access paths of current SQL statements of a DB2 Thread. Visual Explain, which can be integrated into the product, lets you display a graphical view of an access path.

Figure 10 and Figure 11 show how Visual Explain is integrated in Performance Expert Client to draw graphical access paths of Content Manager queries.

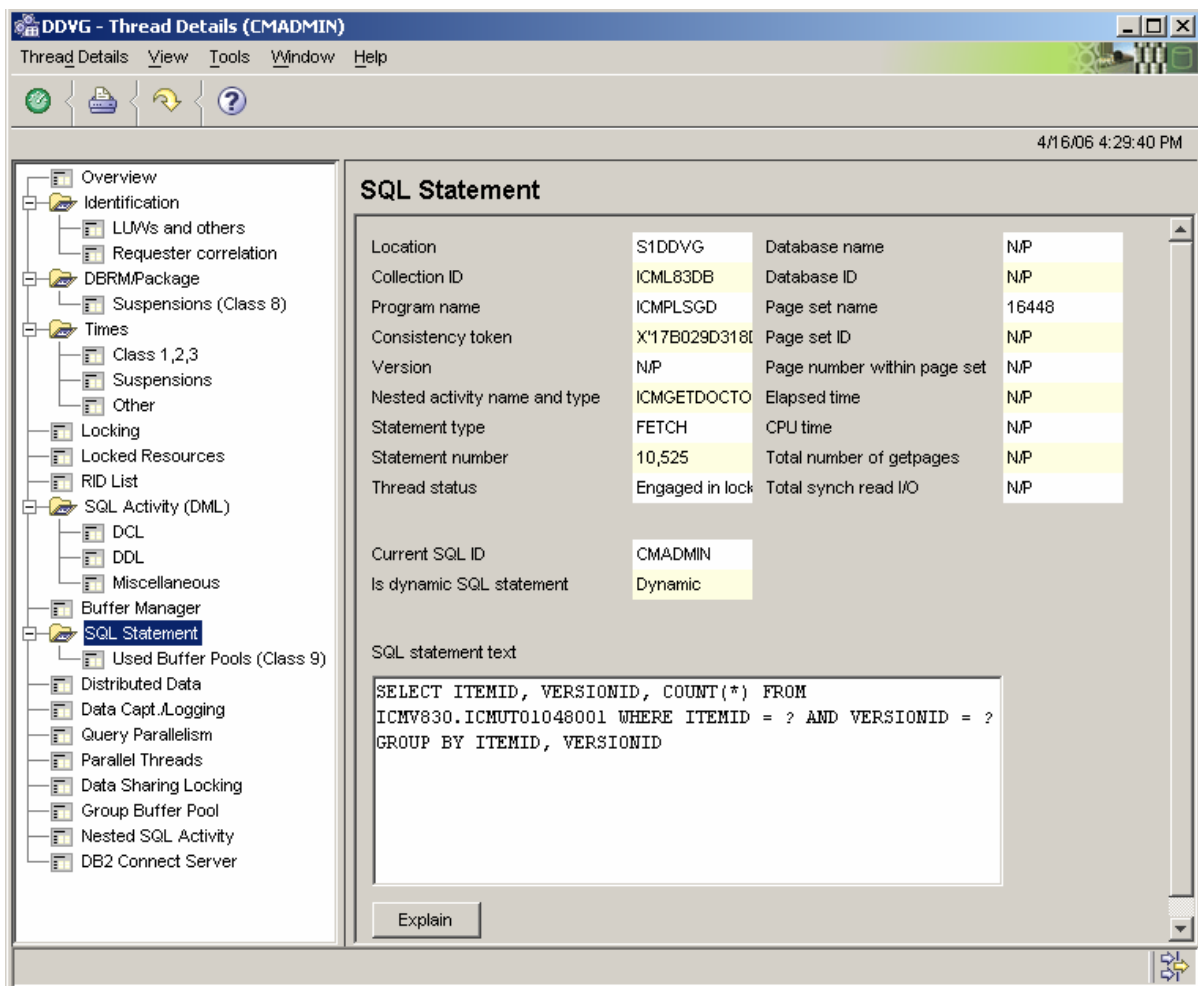


Figure 10: SQL Statement in Performance Expert Client Thread Details

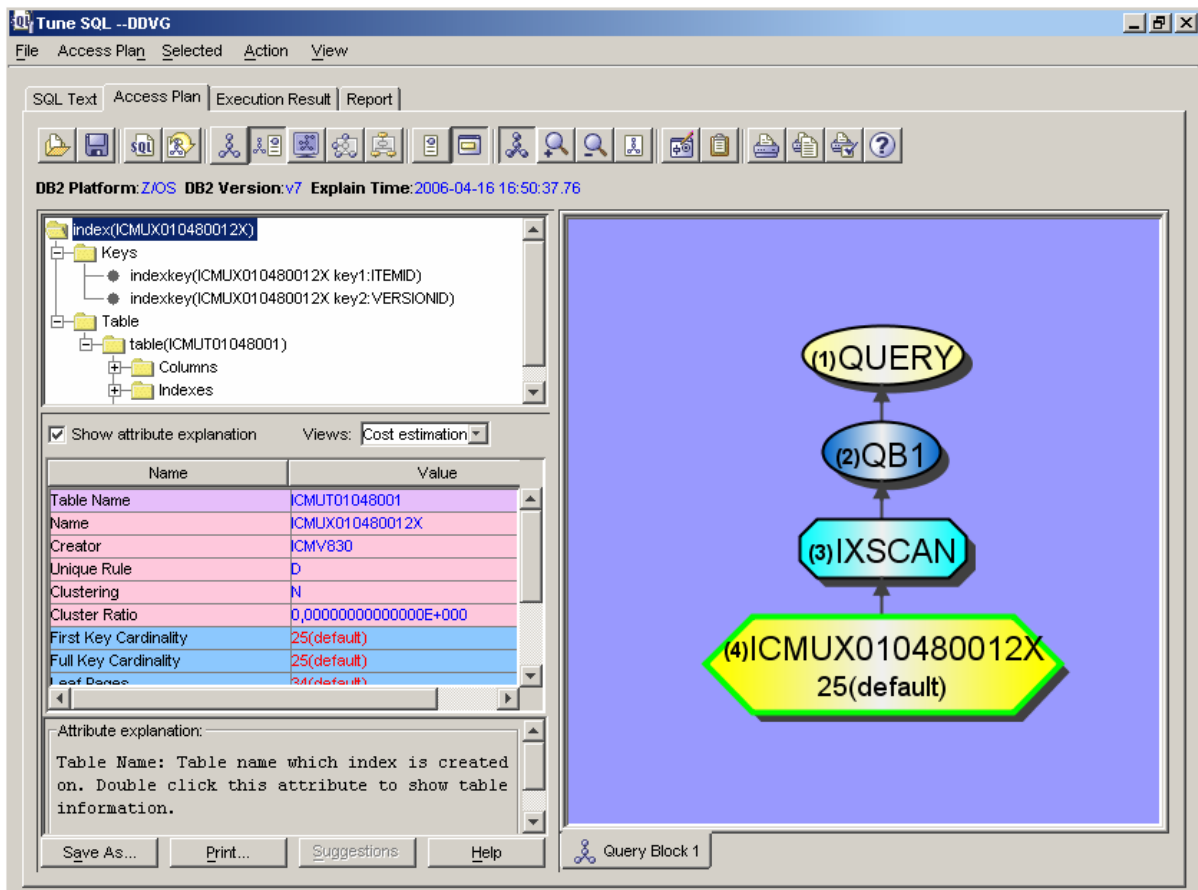


Figure 11: Content Manager query access path in Visual Explain

6.2.3 Tracing Content Manager queries

Content Manager users might complain about bad performance for specific queries, for example, searches, retrievals, storing, or similar CM processes. In this case, it is difficult for a CM administrator to identify the performance bottleneck. If a DB2 issue causes performance problems, an SQL activity trace can help solving it. An SQL activity trace, which is started by Performance Expert Client, tells you exactly which statements have been executed and gives details about the execution of each statement.

The following procedure shows how you can monitor a simple search in the database of the Content Manager Library Server by using OMEGAMON XE for DB2 Performance Expert on z/OS:

The CM Client for Windows can be used to search a document based on attributes of the item type. In the following sample, all documents of type *CSLDMailArchiv2* that have “BK99IKJ234” as mail subject and the sender of which is “Mister Q” are searched (refer to Figure 12).

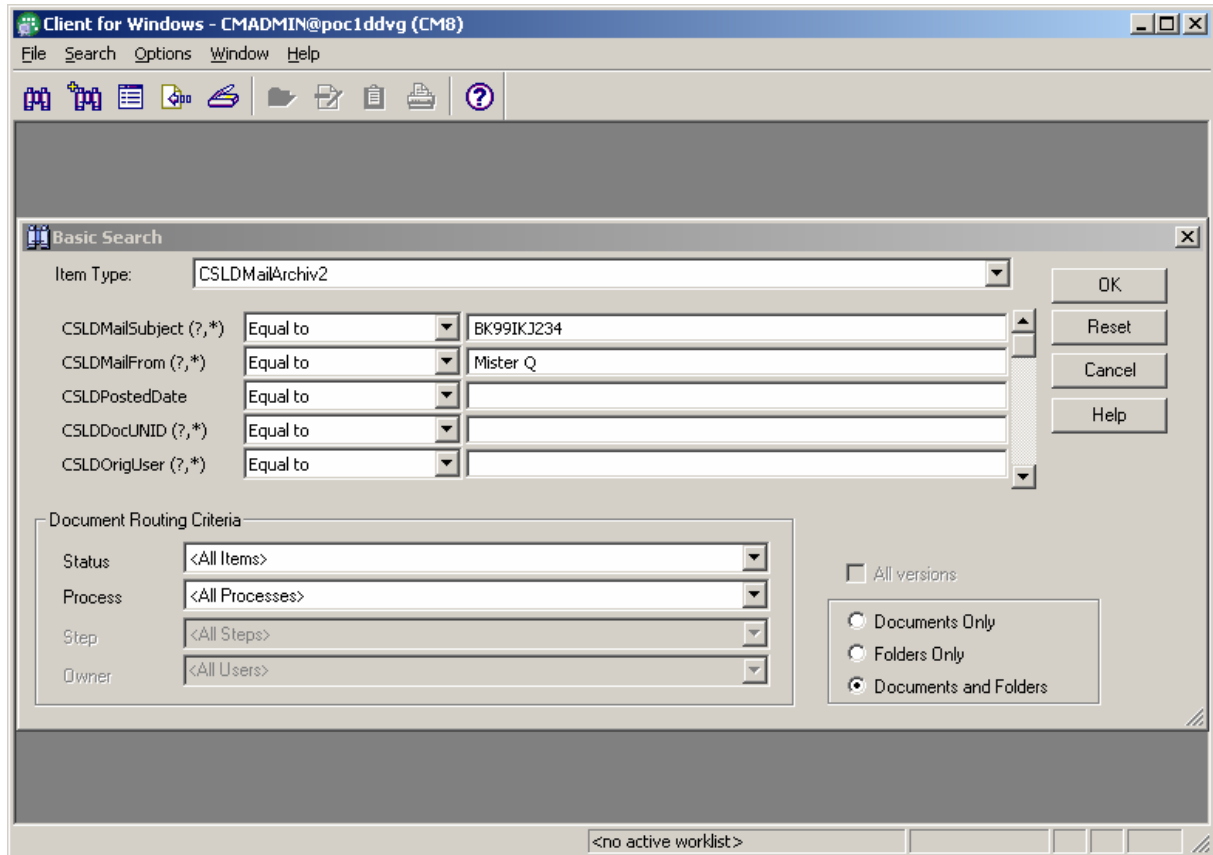


Figure 12: CM Client for Windows - Search

Before starting the search by clicking *OK*, you must identify the thread in Performance Expert Client. To do this, change to the *Performance Expert Client* → *Thread Summary*, select the Client for Windows thread, and click *SQL Activity Tracing*. In Figure 13, the Client for Windows thread is identified by the primary authorization ID *CMADMIN*.

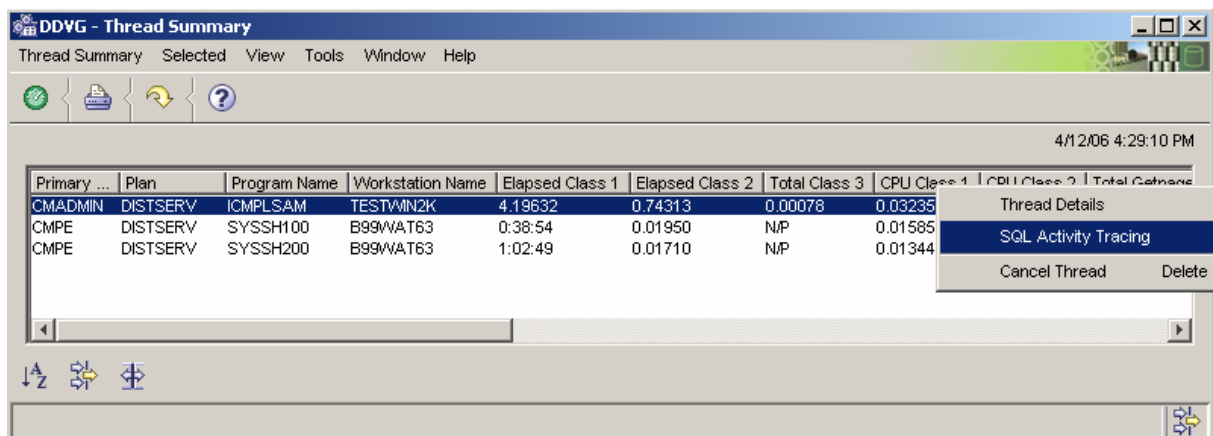


Figure 13: Performance Expert Client - Thread Summary

A new window opens in which you must specify additional settings for the SQL activity trace. Elapsed time defines the maximum time frame to collect SQL activity trace data. You can always stop the trace manually.

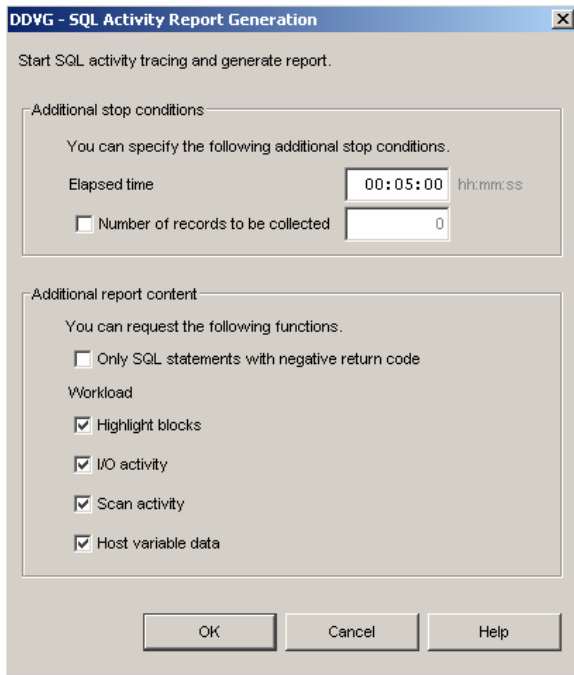


Figure 14: SQL activity Report - Generation

If you click *OK*, the trace starts and collects the SQL activity of the specified thread.

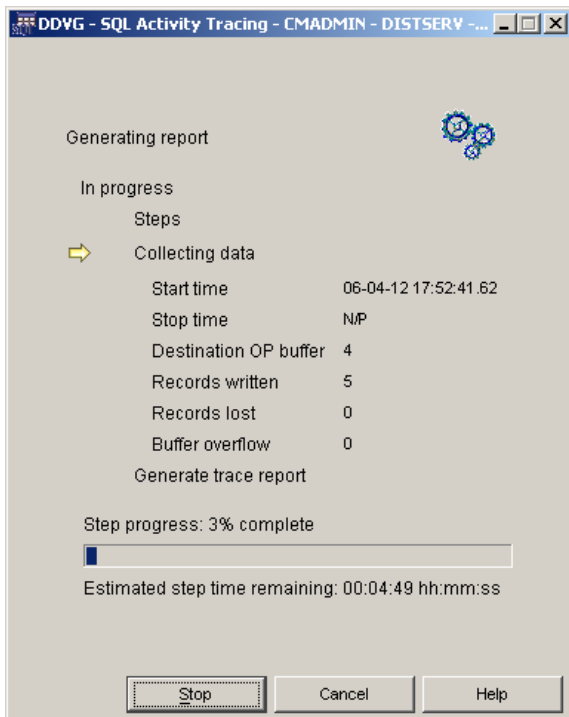


Figure 15: SQL activity Report - Progress

While OMEGAMON XE for DB2 Performance Expert on z/OS collects SQL activity trace data, you can run the search against the Library Server. To start the query, click *OK* in the CM Client for Windows *Basic Search* window (Figure 12).

The following figure shows the result of the query in the database of the Library Server:

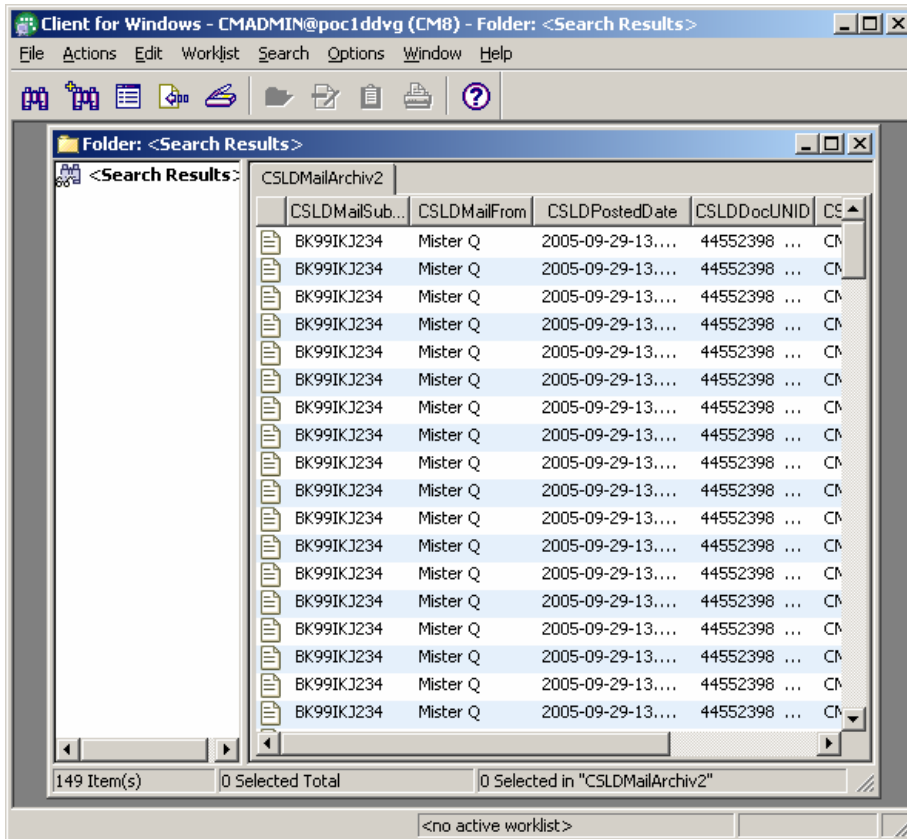


Figure 16: CM Client for Windows - Search complete

The SQL that has been processed for this query can be analyzed in the SQL activity trace. Figure 17 is a small excerpt of the SQL activity trace report. The report comprises several pages. The excerpt shows the part of the activity trace where the lookup of item table *ICMUT01047001* takes place.

In addition to the SQL statements and their execution details, you can also see the scan activity in DB2 databases.



NL	EVENT	TIMESTAMP	ELAP.TIME	TCB	TIME	DETAIL																	
--- SCAN ACTIVITY ---																							
	DATABASE MEMBER	PAGESET TYPE	SCANS	PROCESS	EXAMINE	STAGE 1	STAGE 2	INSERTS	UPDATES	DELETES	DELETES	SCANNED	SCANS	DELETES									
	271	297	1	1	0	1	1	0	0	0	1	2	0	0									
	N/P	INDX																					
1	PREPARE	15:48:10.18	0.010751																				
						STMT#	11669	CURSOR: QUERY_CURSOR1				SQLST:00000	SQLCO:	0									
						TEXT:	SELECT T.* FROM (SELECT DISTINCT CSLDMailArchiv2_1.ITEMID, CSLDMailArchiv2_1.COMPONENTID, CSLDMailArchiv2_1.VERSIONID, 1047 AS COMPONENTTYPEID, 1029 AS ITEMYPEID FROM ICMUT01047001 CSLDMailArchiv2_1 WHERE ((CSLDMailArchiv2_1.ATTR0000001045 = 'BK99IKJ234') AND (CSLDMailArchiv2_1.ATTR0000001046 = 'Mister Q')) AND ((CSLDMailArchiv2_1.SEMANTICTYPE BETWEEN 1 AND 2)) AND (((EXISTS (SELECT 1 FROM ICMSTCOMPILEDACL C , ICMSTITVIEWDEFS V WHERE 1029=V.ITEMYPEID AND V.ITEMYPEVIEWID IN (1029) AND (C.ACL = V.ACLCODE OR C.ACL = -1) AND C.UNUM=2 AND C.RPRIV='1'))) T , ICMSTITEMS001001 I WHERE T.ITEMID = I.ITEMID AND T.VERSIONID = I.VERSIONID OPTIMIZE FOR 500 ROWS FOR READ ONLY WITH UR																
--- SCAN ACTIVITY ---																							
	DATABASE MEMBER	PAGESET TYPE	SCANS	PROCESS	EXAMINE	STAGE 1	STAGE 2	INSERTS	UPDATES	DELETES	DELETES	SCANNED	SCANS	DELETES									
	DSNDB06	47	4	4	0	4	0	0	0	0	4	8	0	0									
	N/P	INDX																					
	DSNDB06	47	4	4	4	0	0	0	0	0	4	4	0	0									
	N/P	SEQD																					
	DSNDB06	61	4	4	0	4	0	0	0	0	4	8	0	0									
	N/P	INDX																					
	DSNDB06	61	4	4	4	0	0	0	0	0	4	4	0	0									
	N/P	SEQD																					
	DSNDB06	66	4	4	0	4	0	0	0	0	4	8	0	0									
	N/P	INDX																					
	TOTAL		20	20	8	12	0	0	0	0	20	32	0	0									
1	OPEN	15:48:10.19	0.053840																				
						STMT#	11700	CURSOR: QUERY_CURSOR1				ISO(UR)	SQLST:00000	SQLCO:	0								
						REOPTIMIZED(NO)	KEEP UPDATE LOCKS(NO)																
1	LOCATION: S1DDVG					OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V3)						PAGE: 1-7											
	GROUP: N/P											REQUESTED FROM: ALL	15:48:00.00										
	MEMBER: N/P											TO: DATES	15:48:45.00										
	SUBSYSTEM: DVVG											ACTUAL FROM: 04/10/06	15:48:10.14										
	DB2 VERSION: V7																						
SUMMARIZED BY OCCURRENCE, WITH SCAN WORKLOAD																							
ORREQUESTER:	9.152.195.50			PRIMAUTH:	CMADMIN	CONNECT :	SERVER	CORRNAME:	ICMClie	CONNTYPE:	DRDA												
				ORIGAUTH:	CMADMIN	PLANNAME:	DISTSERV	CORRNMBR:	t.ex	THRDTYPE:	DBAT												
				ENDUSER :	CMADMIN	WSNAME :	TESTWIN2K	TRANSACTION:	ICMClient.exe														
TRACE #	1.2			DB2 LUWID:	COA80003.L30A.X'0080C0134808'	ACE ADDRESS:	X'1A019388'																
				CICS LUWID:	COA80003.L30A @(.X'134808404040'																		
START TIME:	04/10/06 15:48:10.16			START ELAPSED:	N/A	START REASON:	IN PROGRESS																
STOP TIME:	04/10/06 15:48:38.67			STOP ELAPSED :	N/A	STOP REASON :	END OF FILE																
NL	EVENT	TIMESTAMP	ELAP.TIME	TCB	TIME	DETAIL																	
--- SCAN ACTIVITY ---																							
	DATABASE MEMBER	PAGESET TYPE	SCANS	PROCESS	EXAMINE	STAGE 1	STAGE 2	INSERTS	UPDATES	DELETES	DELETES	SCANNED	SCANS	DELETES									
	DSNDB07	DSN4K01	1047	1546	1546	1546	1546	0	0	0	1047	11	0	0									
	N/P	SEQW																					
	271	264	1	1	0	1	1	0	0	0	1	2	0	0									
	N/P	INDX																					
	271	302	1	1	0	1	1	0	0	0	1	2	0	0									
	N/P	INDX																					
	271	302	1	1	1	0	0	0	0	0	1	1	0	0									
	N/P	SEQD																					
	TOTAL		1050	1549	1547	1548	1548	0	0	0	1050	16	0	0									
PACKAGE						S1DDVG.NULLID.SYSSTAT.X'5359534C564C3031'																	
						ACQUIRE(USE)	REOPT(N)	RELEASE(COMMIT)				ISO(UR)	DYNAMICRULES(RUN)										
						PREPARE(NODEFER)	KEEPDYNAMIC(NO)	PROTOCOL(DRDA)					OPHTINT(N/P)										

Figure 17: SQL activity report output

You can also trace SQL activity by using the Performance Warehouse client. The advantage is that the SQL activity trace can be collected without user interaction, for example, based on a schedule. In addition, you can collect SQL activity trace information for more than one thread that is based on filter criteria, such as plan name or authorization ID. You can then use the Batch reporting functions on the mainframe to create detailed SQL activity reports in various ways.

Performance Warehouse is explained in more detail in Section 6.4 and in the Redbook *A Deep Blue View of DB2 Performance*.

6.2.4 Dynamic SQL

A Content Manager environment processes lots of dynamic SQL. DB2 caches dynamic statements in the Dynamic Statement Cache (if ZPARM *CACHEDYN=YES*). OMEGAMON XE for DB2 Performance Expert on z/OS can display the statements together with their performance metrics. Click *Statistic Details* → *Dynamic SQL Statements*, to get an overview of the effectiveness of the dynamic statement cache.

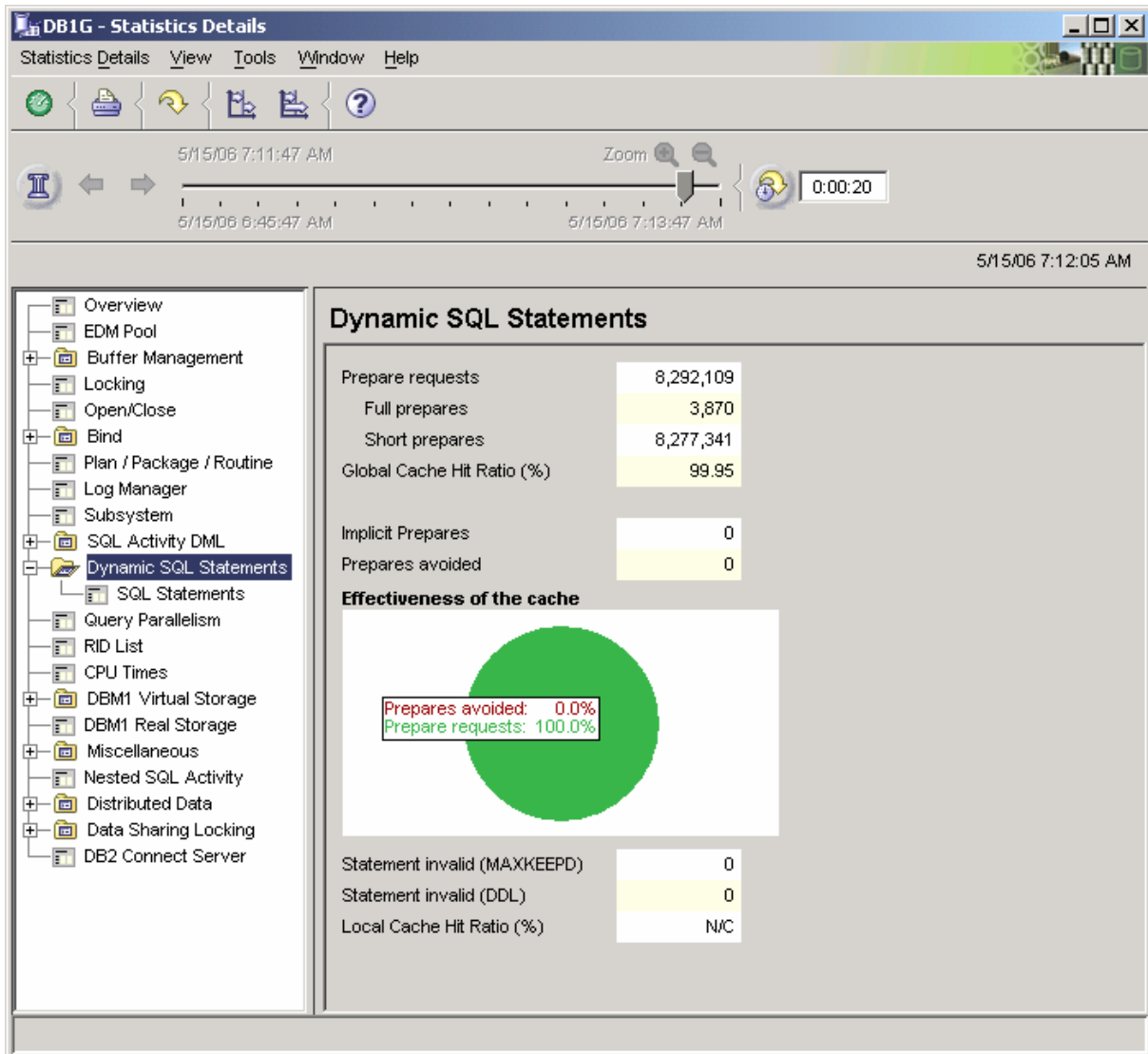


Figure 18: Performance Expert Client - Effectiveness of Dynamic Statement Cache

Field	Description
Prepare request	Shows the total number of explicit and implicit prepare requests. An explicit prepare occurs when an application requests an SQL PREPARE or EXECUTE IMMEDIATE. An explicit prepare always results in a short or a full prepare. An implicit prepare occurs when the user copy of the prepared SQL statement no longer exists in the local dynamic SQL cache. An implicit prepare always results in a short or a full prepare.
Full prepare	A full prepare occurs for explicit and implicit prepare requests when the skeleton copy of the prepared SQL statement is not found in global dynamic SQL cache of the EDM pool.
Short prepare	A short prepare occurs for explicit and implicit prepare requests when the skeleton copy of the prepared SQL statement exists in the global dynamic SQL cache in the EDM pool.

You should monitor the dynamic statement cache and analyze its characteristics over time. If the effectiveness decreases considerably, have a closer look at the cached statements. A good hit ratio depends on your specific environment. A rule-of-thumb states that a hit ratio is good if it is greater than 90%.

The *SQL Statements* panel shows statements together with their performance metrics that are stored in the dynamic statement cache. Use this screen to get information about most frequently used statements.

Figure 19 shows meaningful performance metrics as columns:

Field	Description
Executions	Shows the number of executions of this statement. For a cursor statement, this is the number of OPENS.
Index Scan	Shows the number of index scans performed for this statement.
Tablespace Scan	Shows the number of table space scans performed for the statement.
Getpage	Shows the number of Getpage operations performed for this statement.
Elapsed Time	Shows the accumulated elapsed time used for the statement.
...	

For a complete list of available performance metrics, refer to the Performance Expert Help.

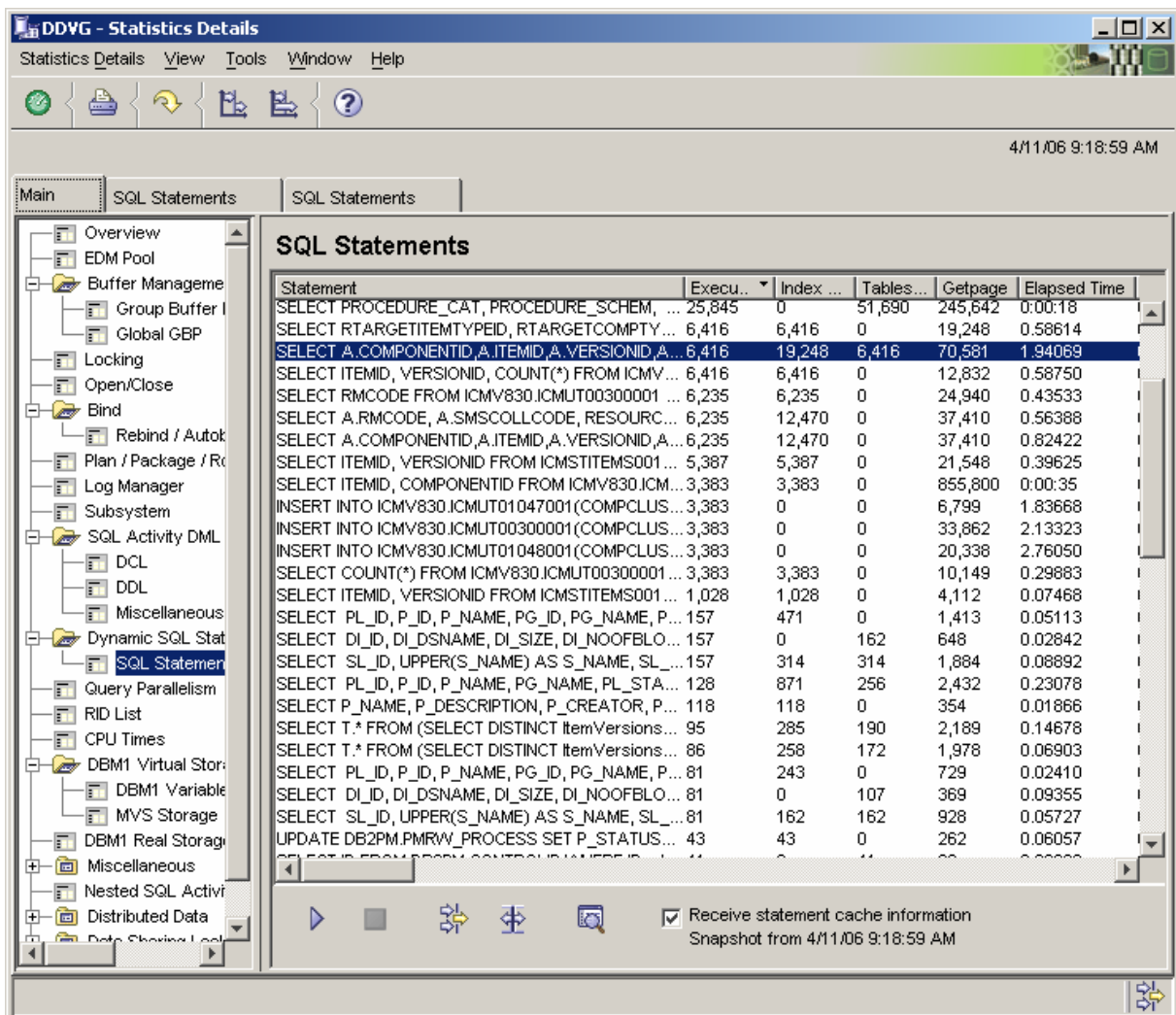


Figure 19: Performance Expert Client - SQL Statements in Dynamic Statement Cache

To get detailed information about a specific statement, you can select it from the list and browse the details. The following statement is a *SELECT* from table *ICMUT01047001*. The table represents a specific item type in the Library Server as explained in Section 5.4.

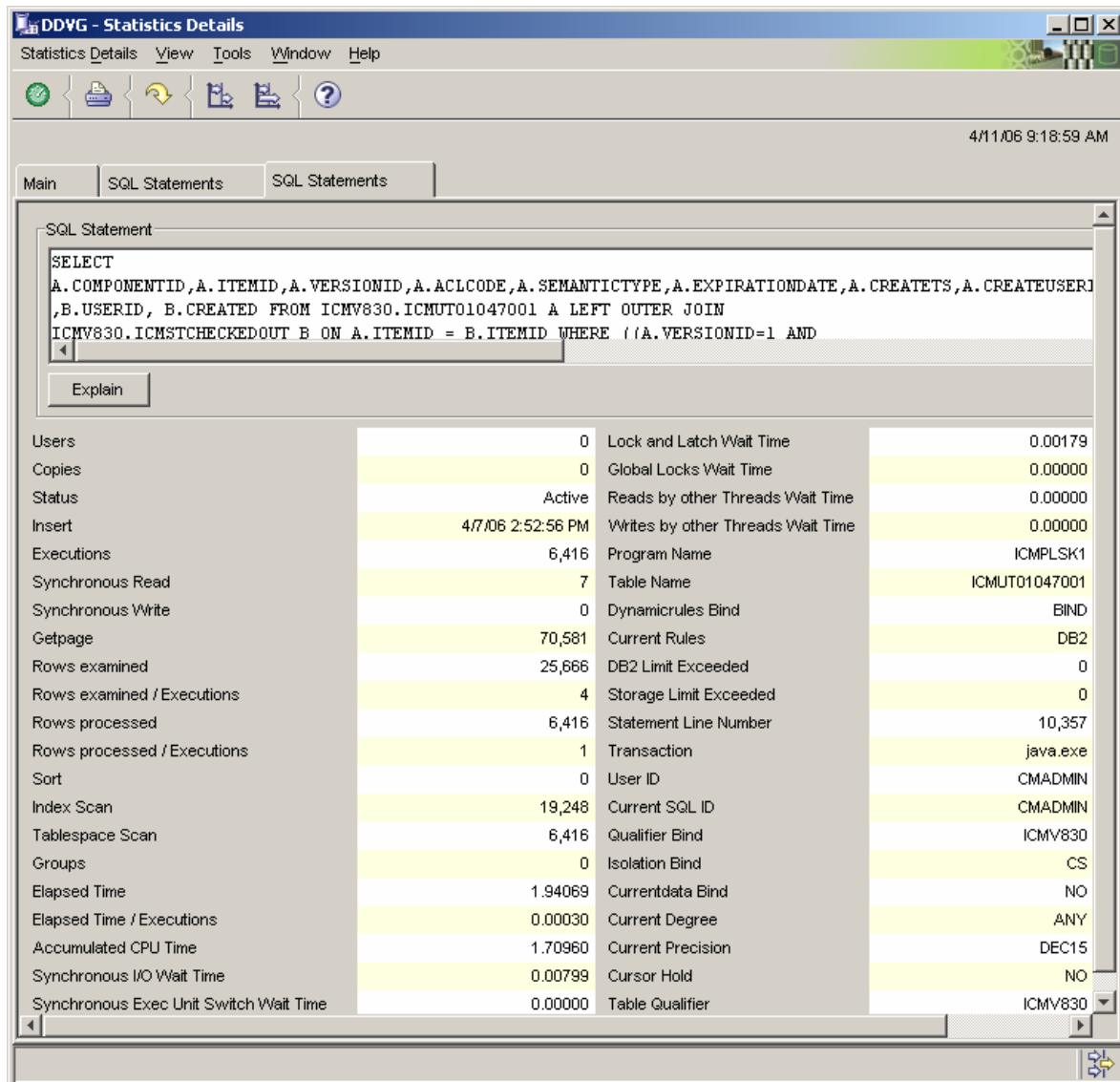


Figure 20: Performance Expert Client - Details about a statement in Dynamic Statement Cache

OMEGAMON XE for DB2 Performance Expert on z/OS provides a description of all fields in a window by pressing the F1 key. Sample field descriptions:

Field	Description
Users	Shows the number of current users of the statement. These are the users who have prepared or executed the statement during their current unit of work.
Copies	Shows the number of copies of the statement owned by all threads in the system.
Status	Shows the status of the statement. If one of the following values is shown, the statement has been removed from the cache but current users might still have an active copy. DB2 continues to track the statement until the use count and copy count are zero. The values are: INVALIDATED BY DROP OR ALTER INVALIDATED BY REVOKE REMOVED FROM CACHE BY LRU INVALIDATED BY RUNSTATS
Rows processed	Shows the number of rows processed for this statement. For example, the number of rows returned for a SELECT, or the number of rows affected by an INSERT, UPDATE, or DELETE.
Accumulated CPU time	Shows the accumulated CPU time. Data for the following columns is only collected when accounting trace class 3 is active.

If this information is not detailed enough, you can receive more information from Visual Explain. Visual Explain provides a graphical view of the access paths that DB2 chooses for an SQL query. This helps to interpret the plan table output without manual efforts.

You can use this information to tune queries based on their specific relationships to different database objects, such as tables and indexes. The graphs show table scans, sorts and similar access path concepts.

Performance Expert Client integrates Visual Explain. To analyze a statement of the Dynamic Statement Cache, click *Explain* in Figure 20.

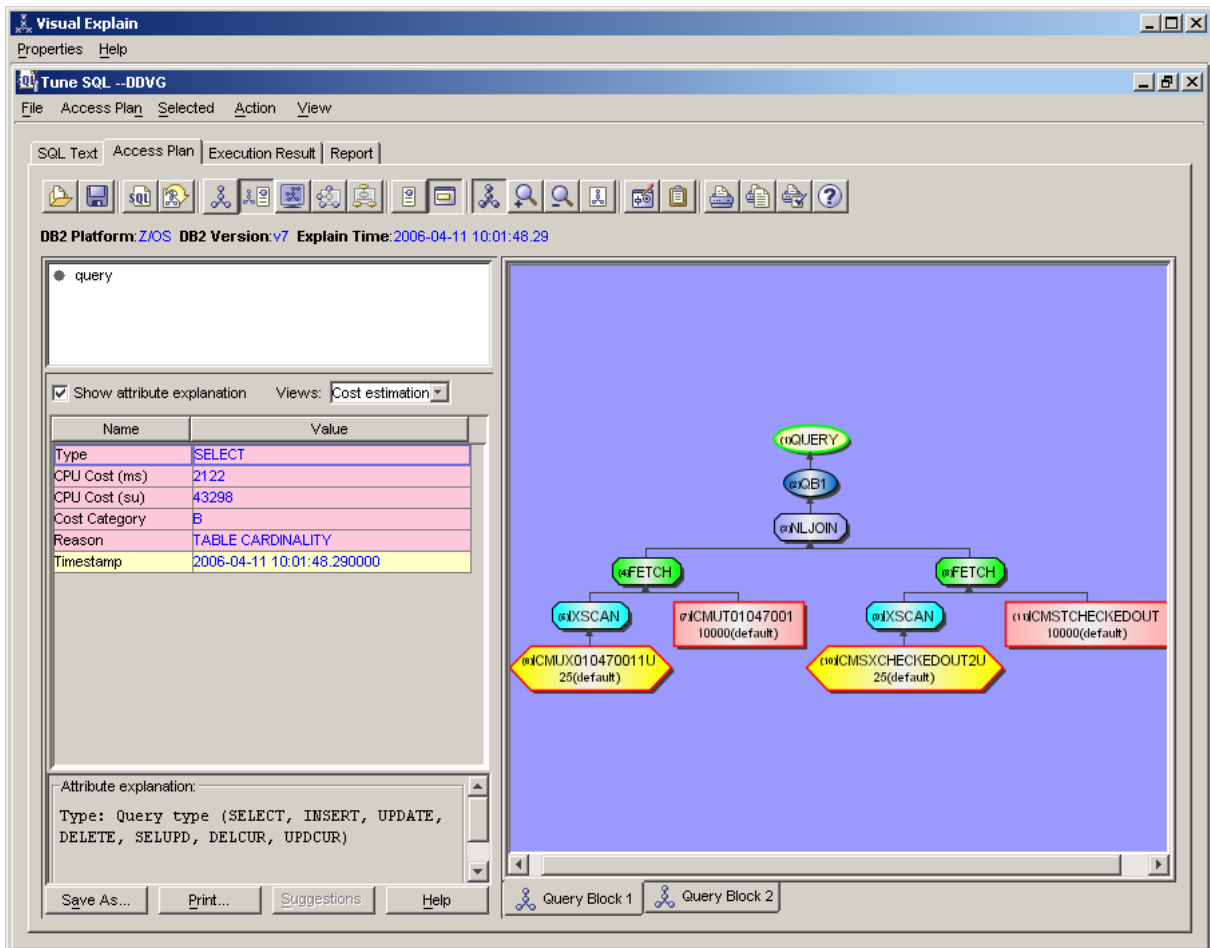


Figure 21: Visual Explain - Access Path

Figure 21 illustrates the access path of a statement from the Dynamic Statement Cache. In this case, it displays a Join between table *ICMUT01047001* (see Section 5.4) and *ICMSTCHECKEDOUT* table. The *ICMSTCHECKEDOUT* table maintains a list of all items that are checked out. When a user checks in a locked item, the row is deleted from this table.

The visual access path displays two index scans and provides additional information, such as CPU cost analysis for this specific statement.

Visual Explain also verifies DB2 statistics and indicates if the last statistics update (*RUNSTATS* run) happened too long ago. In this case, you get a message to run *RUNSTATS* more frequently. In DB2 for z/OS Version 8, the optimizer needs more statistics to calculate good access paths. Therefore, it is important to run *RUNSTATS* more often if the Library Server runs on DB2 for z/OS Version 8.



6.2.5 Object Analysis and I/O

Tuning your buffer pools and SQL statements to ensure good access paths does not always satisfy the needs of your intended performance improvement, whereas analyzing DB2 objects often helps to decrease I/O load.

OMEGAMON XE for DB2 Performance Expert on z/OS offers the *Object Analysis* component through the OMEGAMON Classic interface. Content Manager benefits from this performance analysis feature because the CM environment consists of various DB2 objects that are dynamically created, changed, or removed.

Adding new item types to a CM system results in creating new DB2 tables. Adding attributes to an item type results in altering tables and adding columns to the specified item type table.

If the administrators of the CM environment are not familiar with DB2, they might not create appropriate indexes for attributes that are frequently used in their particular CM workload scenarios.

For more details on access path issues, refer to Section 6.2.2.

From that point of view, it is important to monitor the objects in DB2 from time to time to detect critical situations regarding object allocation, object activity, volume activity, and data set extent activity.

Figure 22 shows the *Object Activity* panel, which allows high-level analysis of Getpage and I/O activity of DB2 databases. The environment that is used in this document contains the following relevant databases:

- ICML83DB Library Server
- ICMR83DB Resource Manager
- GROUP02 OAM (storage for archived documents)

```

_____ ZOJOS     VTS     O2     V310./C DDVG 04/12/06 10:32:44     2
> Help PF1     Back PF3     Up PF7     Down PF8     Sort PF10     Zoom PF11
> O.B
>     OBJECT ANALYSIS:     Enter a selection letter on the top line.
>
> A-OBJECT ALLOCATIONS     *-OBJECT ACTIVITY     C-VOLUME ACTIVITY
> D-EXTEND ACTIVITY     E-DISPLAY OPTIONS     F-COLLECTOR CONTROL
=====
>     OBJECT ACTIVITY SUMMARY
OJOS
+ Interval Time     = 00:15:00     Interval Elapsed = 00:08:25
+ Total Getpage     = 3123059     Total I/O     =     119
+ *
+     % of     % of     Getp     Sync     Pre     Async     Other
+ Database     Getp     I/O     per RIO     Getpage     Read     Fetch     Write     Write
+ -----
+ DB2PM     .2%     .0%     7044.0     7044     0     0     0     0
+ DSNDB06     68.3%     .0%     2133759.0     2133759     0     0     0     0
+ DSNDB07     15.4%     1.6%     241702.0     483404     0     2     0     0
+ GROUP02     .8%     92.4%     26555.0     26555     0     0     110     0
+ ICML83DB     13.9%     2.5%     434873.0     434873     1     0     2     0
+ ICMR83DB     .9%     3.3%     7045.0     28180     4     0     0     0
+ OAMADMIN     .2%     .0%     9244.0     9244     0     0     0     0
=====

```

Figure 22: VTAM - Object Activity

This activity report shows a lot of Getpages in the Library Server (13.9%) but a low I/O rate. The Library Server is a catalog file where lots of searches, simple inserts, updates and deletes take place. On the contrary, the OAM database denotes high I/O rates (92,4%) because these table spaces contain the real documents that are stored and retrieved. This also results in high write rates. It seems that a lot of store workload occurred during analyzing the CM system.

Move the cursor over a database and press the F11 key to zoom in to display more details in the *Database Activity* panel. The panel shows Getpage and I/O activity for spaces that are owned by a single database. The following figure shows spaces of the Library Server:



```

_____ ZOJ01      VTS      O2      V310./C DDVG 04/12/06 10:57:46  2
> Help PF1      Back PF3      Up PF7      Down PF8      Sort PF10      Zoom PF11
>
>      OBJECT ANALYSIS:  Enter a selection letter on the top line.
>
> A-DATABASE ALLOCS  *-DATABASE ACTIVITY  C-THREAD ACTIVITY  D-DISPLAY OPTIONS
=====
>
>      DATABASE ACTIVITY
OJ01
+ Interval Time    = 00:15:00      Interval Elapsed = 00:03:26
+ Total Getpage    = 59787          Total I/O        = 42
+
+ Database = ICML83DB
+
+      *
+      % of      % of      Getp      Sync      Pre      Async      Other
+ Spacenam  Getp      I/O      per RIO   Getpage  Read      Fetch      Write      Write
+ -----
+ ICMU1LPQ   39.2%    4.7%    23455.0   23455    0         0         2         0
+ ICMS1E1Y   4.4%     .0%     2664.0    2664    0         0         0         0
+ ICMSTITE   3.5%     .0%     2122.0    2122    0         0         0         0
+ COMDLSTS   3.0%     .0%     1824.0    1824    0         0         0         0
+ ICMS1$R1   3.0%     4.7%     1853.0    1853    0         0         2         0
+ ICMS1051   2.8%     4.7%     1709.0    1709    0         0         2         0
+ ICMU1D4Q   2.6%     4.7%     1567.0    1567    0         0         2         0
+ ICMS1GP8   2.6%     4.7%     1610.0    1610    0         0         2         0
+ ICMSTCOM   2.2%     .0%     1348.0    1348    0         0         0         0
+ ICMS1E23   2.2%     .0%     1322.0    1322    0         0         0         0
+ ICMSTITV   2.0%     .0%     1254.0    1254    0         0         0         0
+ ICMLFQ32   1.8%     4.7%     1077.0    1077    0         0         2         0

```

Figure 23: VTAM - Database activity

This report gives a detailed view of the table spaces that are used by the Library Server of the sample environment. Table space *ICMLFQ32* for example holds all *ICMUT** tables that represent the item types.

You should again monitor these I/O rates and identify the spaces that do not follow the overall trend. If, for example, the *ICMLFQ32* changes its I/O behavior, you should analyze if new item types have been added to the system and if indexes have been created for all attributes that are frequently used in queries.

DB2 provides the concept of partitioned table spaces for very large tables. Table rows are distributed over several partitions based on a partition key. In Content Manager, the attribute *ITEMID* is an applicable partitioning key because it is built on a time stamp. The *ICMSTITEMSnnnsss* table (where *nnn* is LibraryID and *sss* is SysSegmentID) identifies the items within Content Manager. If you are working with many items and you experience high I/O rate for the corresponding table space, you should partition the table space.

Another sample is the *ICMSTLINKSnnnsss* table, which maintains relationships between items. Links can be used to model various functionalities. For example, you can support folder and document relationships. To support folder and document relationships, each link represents a folder relationship or a document relationship. The *TargetItemID* represents the document item, and the *SourceItemID* represents the folder item.

If most of your documents are stored in folders, the *LINKS* table increases. Therefore, you should also partition it, probably based on the system-defined attribute *SOURCEITEMID*.

For more details about partitioning specific table spaces, refer to the *IBM Content Manager v8.2 Performance Tuning Guide*.

To get detailed allocation-related information about a single table space, move the cursor over the table space name and press F11 to zoom in:

```

_____ ZOJA2      VTS      O2      V310./C DDVG 04/12/06 11:43:37  2
> Help PF1      Back PF3      Up PF7      Down PF8      Sort PF10      Zoom PF11
>
>      OBJECT ANALYSIS:  Enter a selection letter on the top line.
>
> *--SPACENAME ALLOCS                                     B--SPACENAME ACTIVITY
=====
>                                     SPACENAME ALLOCATIONS
OJA2
+ Database = ICML83DB      Spacename = ICMLFQ32      Type = TABLESPACE
+ Datasets = 1            Volumes = 1            Exts = 1
+ Space = 420 Trks
+ *
+
+ DSN      Vols      Exts      High      High      %
+ ----      ----      ----      Fmt Pg      Alloc Pg      In Use      Dataset Space
+ -----
+ 001      1      1      0001E3      000275      76.7%      420 Trks
=====

```

Figure 24: VTAM - Spacename allocations

The panel *SPACENAME ALLOCATIONS* displays data sets that are currently allocated by DB2 and provides the following fields:

<i>Field</i>	<i>Description</i>
Type	simple, partitioned or segmented table space index or partitioned index
Datasets	Number of data sets used by the space
Volumes	Number of volumes used by the data sets
Exts	Number of extents allocated by all data sets
Space	Total amount of space
DSN /dataset	Data set sequence number
Vols /dataset	Number of volumes that the data set resides on
Exts /dataset	Number of extents allocated by this data set
High Fmt Pg /dataset	Highest page number formatted in the data set (in hex)
High Alloc Pg /dataset	Highest page number allocated in the data set (in hex)
% in use / dataset	Percentage of pages in use for the data set
Dataset space /dataset	Total amount of space allocated by this data set

A high-level overview of allocation related information is also provided by the *Object Allocation Summary* panel. This is a less granular report on database level as displayed in Figure 25.



```

_____ ZOJAS   VTS   O2       V310./C DDVG 04/12/06 11:56:44  2
> Help PF1   Back PF3   Up PF7   Down PF8   Sort PF10   Zoom PF11
> O.A
>
>   OBJECT ANALYSIS:  Enter a selection letter on the top line.
>
> * -OBJECT ALLOCATIONS          B-OBJECT ACTIVITY          C-VOLUME ACTIVITY
> D-EXTEND ACTIVITY             E-DISPLAY OPTIONS          F-COLLECTOR CONTROL
=====
>
>                               OBJECT ALLOCATION SUMMARY
OJAS
+ Total Databases = 8          Total Spaces = 202          Total Datasets = 191
+ *
+ Database   Spaces   Tblsp   Indxs   DSNs   Extents   Exts/DSN   Max Exts
+ -----
+ DB2PM      29         7        22      29     34        1.1         4
+ DSNDB01    11         5         6       11     50        4.5        23
+ DSNDB06    43        13        30      42     49        1.1         7
+ DSNDB07     2         2         0        2     10        5.0         8
+ GROUP02    6         2         4        6      6         1.0         1
+ ICML83DB   94        34        60      85    160        1.8        12
+ ICMR83DB   9         5         4         8      8         1.0         1
+ OAMADMIN   8         3         5         8      8         1.0         1

```

Figure 25: VTAM - Object allocation summary

The panel *Volume Activity Summary* helps to observe volumes that are used by DB2 databases. Again, you can zoom into volumes to get more details about I/O rates and allocations on volume level. If most of the I/O is processed by a specific volume, you should have a closer look at your table space distribution.

```

_____ ZOJVS   VTS   O2       V310./C DDVG 04/12/06 12:18:20  2
> Help PF1   Back PF3   Up PF7   Down PF8   Sort PF10   Zoom PF11
> O.C
>
>   OBJECT ANALYSIS:  Enter a selection letter on the top line.
>
> A-OBJECT ALLOCATIONS          B-OBJECT ACTIVITY          *-VOLUME ACTIVITY
> D-EXTEND ACTIVITY             E-DISPLAY OPTIONS          F-COLLECTOR CONTROL
=====
>
>                               VOLUME ACTIVITY SUMMARY
OJVS
+ Interval Time   = 00:15:00          Interval Elapsed   = 00:09:01
+ *
+
+ Volume Unit  Util%   Vol   Serv   Total  DB2    % DB2  Alloc  Alloc  Exts/
+ -----
+ DB2DVG 5412   .0%    .6    139    4      3.5%   78    134    1.7
+ DB2DV2 5413   .0%    1.7    12     5      4.4%   13     38    2.9
+ DB2DV3 520B   .0%    3.2    86    50     44.6%  91    124    1.3
+ OAMOM1 5A1E   .0%    1.0   249    1       .8%    9      9     1.0
+ OAMOM2 5A1F   .0%    1.0    58    52     46.4%  8     27    3.3

```

Figure 26: VTAM - Volume activity



6.3 Monitoring DB2 Connect

Content Manager applications usually run on distributed platforms and connect to a Library Server or Resource Manager on z/OS. DB2 for z/OS accepts inbound-distributed connections through its Distributed Data Facility (DDF) address space and the Distributed Relational Database Architecture (DRDA) protocol.

Clients such as the CM Client for Windows or any other CM application can establish such a connection to DB2 through a DB2 Connect instance running on Linux, UNIX, or Windows systems. OMEGAMON XE for DB2 Performance Expert on z/OS can monitor DB2 Connect instances and analyze the performance of their DB2 connection. You can use this information to evaluate the overall health of a DB2 Connect server, or to make changes to improve the throughput.

The performance data is collected by a DB2 Connect Monitoring Agent that runs on the DB2 Connect machine. Then it is stored into Performance Warehouse database tables. The OMEGAMON Server uses this data to provide an overall view of DB2 Connect health and connection status. This status can be viewed from Performance Expert Client, the OMEGAMON Classic interface, and the Tivoli Enterprise Portal.

For more information about DB2 Connect, refer to *DB2 Connect User's Guide Version 8*, SC09-4835.

6.3.1 Connection Pooling and Connection Concentrator

Connection Pooling and Connection Concentrator are important features of DB2 Connect servers. Content Manager environments that handle many requests to the Library Server from different users or high query rates require many concurrent connections.

DB2 Connect Monitoring Agents handle these connections and forward them to DB2 for z/OS where a new thread is created. Without using Connection Pooling and Connection Concentrator, opening a new connection results in creating a new thread in DB2.

Assembling and disassembling threads in DB2 is an expensive procedure and consumes significant resources without actually processing CM data. This means that the system performance decreases.

Connection Pooling lets you reuse an established thread for further incoming connections. This means that a thread remains open in DB2 after the client closes its connection. If the connection is reopened, it is handled by the same thread.

Connection Concentrator lets you share physical connections to DB2 for z/OS by multiple clients. Clients open logical connections to DB2 Connect and DB2 Connect decides at transaction boundaries (COMMIT, ROLLBACK) about the actual mapping of a logical connection to a physical host connection.

The following parameters of DB2 Connect customize Connection Pooling and Connection Concentrator:

MAX_CONNECTIONS	Maximum number of logical agents
MAX_COORDAGENTS	Maximum number of coordinator agents. Connection Concentrator is on if MAX_CONNECTIONS > MAX_COORDAGENTS
MAXAGENTS	Maximum of overall agents in the system
MAXCAGENTS	Maximum number of concurrent coordinator agents
NUM_POOLAGENTS	Maximum number of idle and inactive agents in the pool
NUM_INITAGENTS	Number of idle agents DB2 Connect creates at startup

The Tivoli Enterprise Portal allows you to monitor these settings and hereby supports you to customize these parameters in your environment.

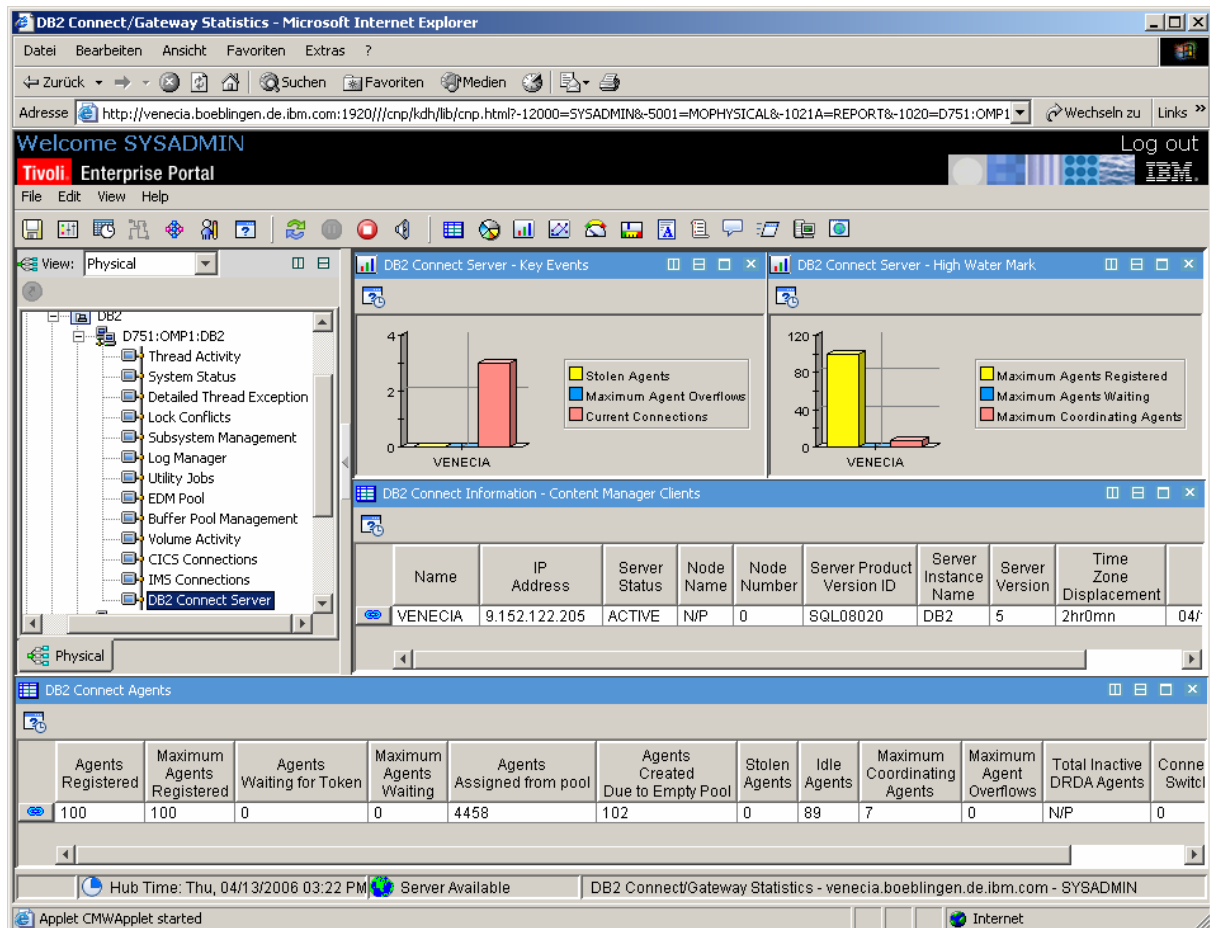


Figure 27: TEP - DB2 Connect Monitoring

TEP graphically displays the agents of the CM applications that are currently connected. It also shows how many agents are waiting to get connected. The blue chart bar in the *High Water Mark* component in the upper right corner displays the number of agents that wait for a connection because the physical connections are currently used.

If too many agents are waiting, CM clients experience a bad performance for their connections to DB2.

6.3.2 TCP/IP & networking

Another option of DB2 Connect Monitoring is the display of the execution time of a sample SQL statement. Monitoring the sample statement helps you identify network problems between DB2 Connect and DB2 for z/OS. If the network part of the execution time increases considerably, you should contact the network administrators. The reason for the increase might be changes in the network infrastructure.

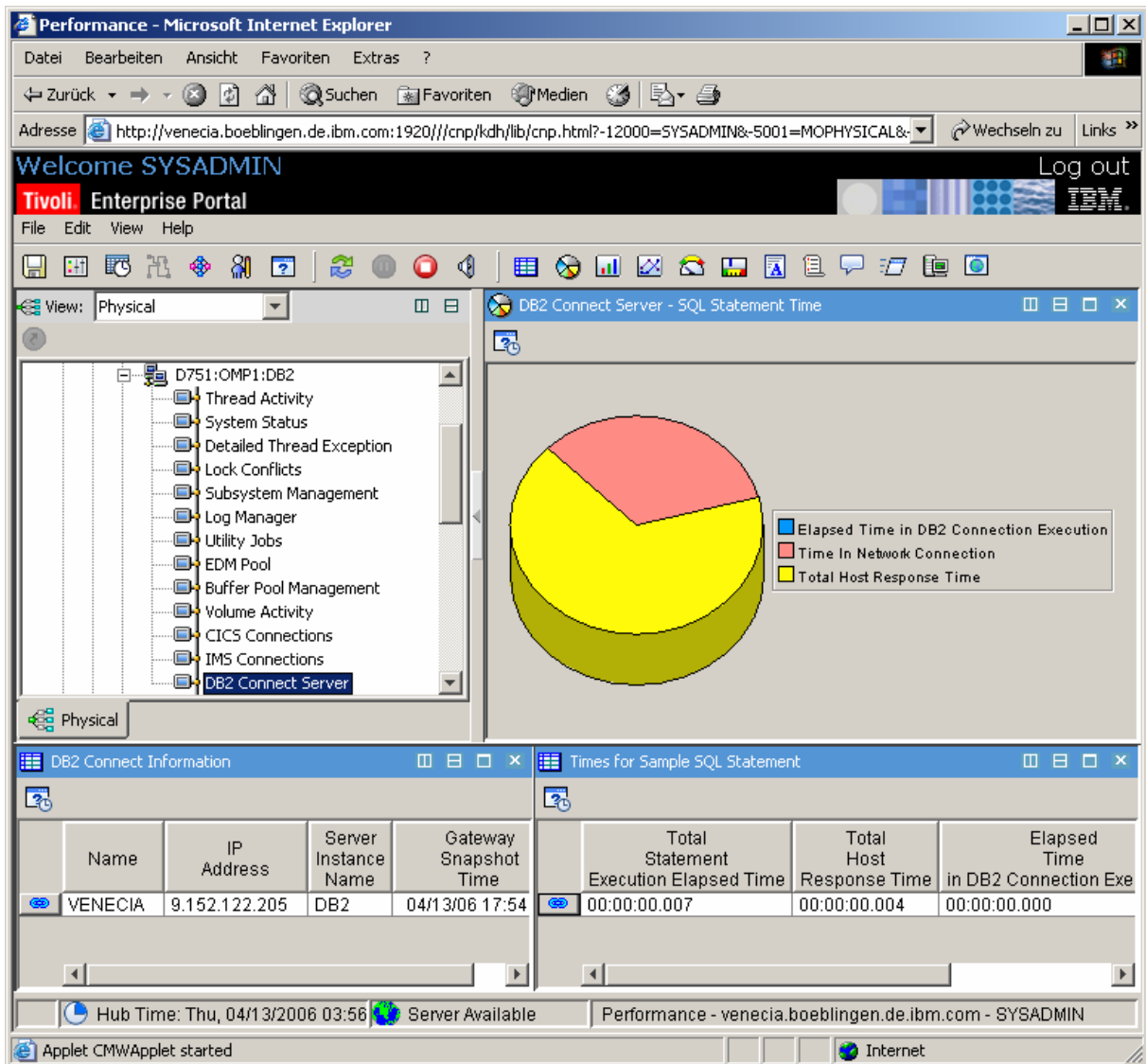


Figure 28: TEP - DB2 Connect Monitoring, SQL statement time

6.4 Performance Warehouse

Performance Warehouse provides an infrastructure to:

1. Automate performance analysis tasks
2. Save historical monitoring data over a long period of time in the Performance Warehouse database
3. Perform analysis of Performance Warehouse data by using SQL queries or predefined rules-of-thumb.

The infrastructure consists of a Performance Warehouse server and a Performance Warehouse client. The server component implements a kind of workflow engine to automate performance analysis tasks. Moreover, it manages and maintains a Performance Warehouse database to store performance data. The client component enables you to define Performance Warehouse processes, Performance Warehouse queries, and Performance Warehouse rules-of-thumb. These Performance Warehouse objects are also stored in the Performance Warehouse database.



A Performance Warehouse process is the concept to automate a performance analysis task. This process consists of one or more steps that you can use to:

- Collect DB2 performance data,
- Load DB2 performance into the Performance Warehouse database
- Generate performance reports

You can organize Performance Warehouse processes in process groups by using the Performance Warehouse client, which is similar to the organization of files by using directories.

A Performance Warehouse query is an SQL query to analyze performance data that is stored in the Performance Warehouse database. The Performance Warehouse client provides query editors that support you in defining Performance Warehouse queries by accessing the meta data model of the Performance Warehouse database. Like Performance Warehouse processes, queries can also be organized in groups.

A Performance Warehouse rule-of-thumb is a DB2 expert rule, such as an arithmetic expression, that consists of DB2 performance counters that are associated with problem and warning thresholds. You can use rules-of-thumb to analyze performance data that is stored in the Performance Warehouse database. For more information about rules-of-thumb, refer to the Redbooks, which are listed in the Appendix.

OMEGAMON XE for DB2 Performance Expert on z/OS provides templates for Performance Warehouse objects. Templates are product-supplied samples for processes, queries, and rules-of-thumb, which you can copy and customize according to your needs.

Currently, Performance Warehouse supports the scheduled loading of statistics and accounting performance data into the Performance Warehouse database and the scheduled generation of statistics and accounting reports. Supported load formats are SMF data sets, GTF data sets, converted SAVE data sets, or FILE data sets. Note that the automatically generated statistics and accounting reports are also stored in the Performance Warehouse database for your reference.

Statistics and accounting information helps you proactively tune the databases in a CM environment. It provides general performance information about the subsystem and detailed workload information on transaction level at low cost because collecting this information has only a low impact on the monitored subsystem itself.

Storing statistics and accounting data over time in the Performance Warehouse database and analyzing this data on a regular basis, helps you identify potential performance problems or performance trends and determine the times of heavy workload on your system.

You can find detailed information about the Performance Warehouse infrastructure and the Performance Warehouse client in the following references:

- Manual: Monitoring Performance from the Performance Expert Client, SC18-9640
- Redbook: A Deep Blue View of DB2 Performance, SG 247224

The following section shows how to use the Performance Warehouse infrastructure to monitor threads of an CM application client that are connected to your Library Server and to the Resource Manager databases.

You get the following information:

1. The times at which the most threads are connected to your Library Server and Resource Manager databases. This helps you determine the peak times and normal workload times of your CM system. It also helps you find out if the workload on your Content Manager system is evenly spread or if it changes over time.
2. How busy these threads are and how many resources they consume.

6.4.1 Determine peak workload times in your CM system

To determine the peak times and normal workload times, you must load statistic records into the Performance Warehouse database on a regular basis. To perform this, you can setup a Performance Warehouse process that is scheduled to start after the SMF switches take place. During process execution, the statistics data is read from the SMF data set and stored in the Performance Warehouse database. Optionally, you can generate a statistics report afterwards.

To implement this, you can use the process template *DB2PM.Templates.Load Warehouse 5* in process group *Public*. The process template uses the reporting function of OMEGAMON XE for DB2 Performance Expert on z/OS to load statistics data from SMF data sets into the Performance Warehouse database.

Copy the process and use the Performance Warehouse client interface to interactively modify the report command stream. For example, you can modify the way in which the statistics data is loaded into the Performance Warehouse database, or you can add a report command to generate an additional statistics report. Schedule and activate the process to load the statistics data into the Performance Warehouse database on a regular basis.

The report step of the process is configured in the Performance Warehouse GUI as shown in Figure 29.

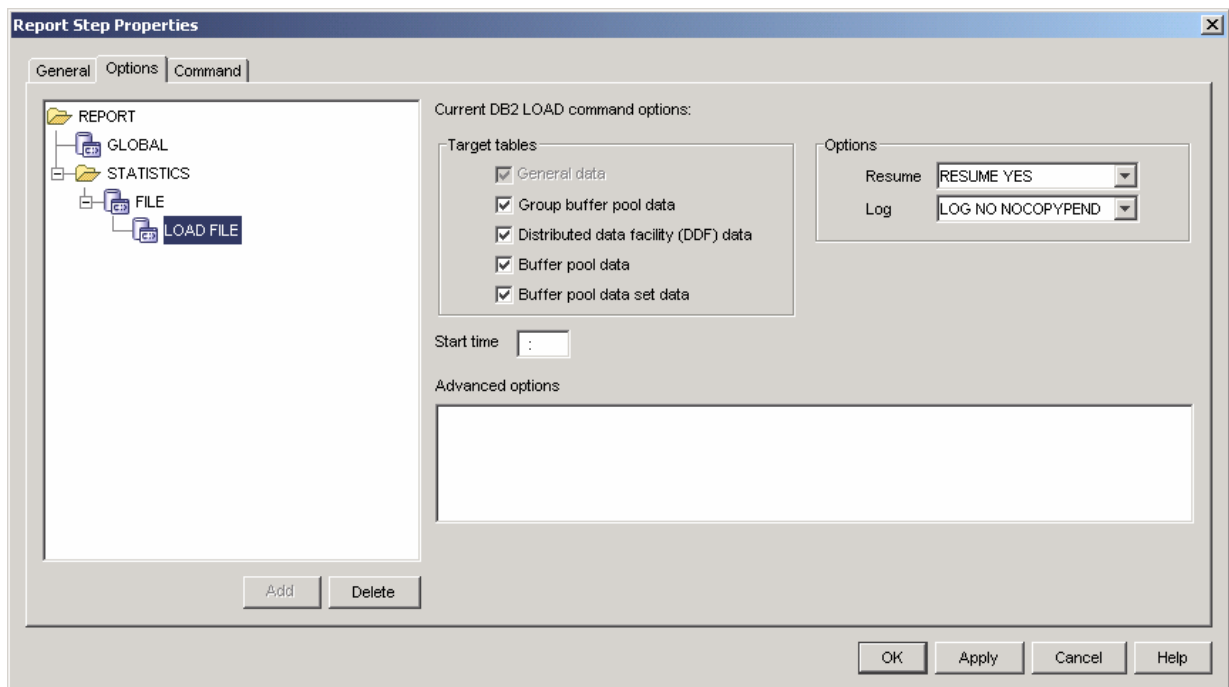


Figure 29: Performance Warehouse - Report Step Properties

Next, create a new query group and create the following queries:

1. *SQL Characteristics*
2. *STP Throughput*

The query *SQL Characteristics* shows the percentage of SQL DML operations over time. It can be used to analyze changes in workload characteristics. The query is as follows:



```

SELECT
  G.END_REC_TSTAMP,
  DECIMAL( (G.SELECT/C.TOTAL)*100, 5, 2)          AS "% SELECT",
  DECIMAL( (G.INSERT/C.TOTAL)*100, 5, 2)          AS "% INSERT",
  DECIMAL( (G.UPDATE/C.TOTAL)*100, 5, 2)          AS "% UPDATE",
  DECIMAL( (G.DELETE/C.TOTAL)*100, 5, 2)          AS "% DELETE",
  DECIMAL( (G.PREPARE/C.TOTAL)*100, 5, 2)         AS "% PREPARE",
  DECIMAL( (G.DESCRIBE/C.TOTAL)*100, 5, 2)        AS "% DESCRIBE",
  DECIMAL( (G.DESCRIBE_TABLE/C.TOTAL)*100, 5, 2)  AS "% DESCRIBE TABLE",
  DECIMAL( (G.OPEN/C.TOTAL)*100, 5, 2)            AS "% OPEN",
  DECIMAL( (G.CLOSE/C.TOTAL)*100, 5, 2)           AS "% CLOSE",
  DECIMAL( (G.FETCH/C.TOTAL)*100, 5, 2)           AS "% FETCH"
FROM
  DB2PM.DB2PM_STAT_GENERAL G,
  (SELECT A.LOCAL_LOCATION, A.GROUP_NAME, A.SUBSYSTEM_ID, A.MEMBER_NAME, A.BEGIN_REC_TSTAMP,
    (A.SELECT + A.INSERT + A.UPDATE + A.DELETE + A.PREPARE + A.DESCRIBE +
    A.DESCRIBE_TABLE + A.OPEN + A.CLOSE + A.FETCH) AS TOTAL
  FROM DB2PM.DB2PM_STAT_GENERAL A) AS C
WHERE G.LOCAL_LOCATION=C.LOCAL_LOCATION
AND   G.GROUP_NAME=C.GROUP_NAME
AND   G.SUBSYSTEM_ID=C.SUBSYSTEM_ID
AND   G.MEMBER_NAME=C.MEMBER_NAME
AND   G.BEGIN_REC_TSTAMP=C.BEGIN_REC_TSTAMP
AND G.LOCAL_LOCATION = :loc
AND G.GROUP_NAME = :group
AND G.SUBSYSTEM_ID = :subsystem
ORDER BY G.END_REC_TSTAMP

```

This screenshot shows an example query result in HTML format. It indicates that the workload mainly performs data retrieval and that there is a peak interval between 1:30 PM and 1:49 PM.

END_REC_TSTAMP	% SELECT	% INSERT	% UPDATE	% DELETE	% PREPARE	% DESCRIBE	% DESCRIBE TABLE	% OPEN	% CLOSE	% FETCH
2006-04-11 13:29:02.206986	18.66	6.29	1.34	0.81	3.13	0.00	0.00	6.82	5.97	56.92
2006-04-11 13:34:03.152636	21.62	7.54	1.02	0.93	3.39	0.00	0.00	7.39	6.51	51.57
2006-04-11 13:39:04.0971	21.67	7.51	1.08	0.94	3.51	0.00	0.00	7.69	6.81	50.74
2006-04-11 13:44:05.040839	21.46	7.12	1.02	0.92	3.38	0.00	0.00	7.37	6.47	52.22
2006-04-11 13:49:05.984169	22.29	6.81	1.00	0.94	3.39	0.00	0.00	7.34	6.41	51.78
2006-04-11 13:54:06.929138	11.13	1.10	3.52	1.27	7.37	0.00	0.00	18.84	18.70	38.03
2006-04-11 13:59:07.874118	9.04	0.00	4.11	1.28	8.36	0.00	0.00	21.65	21.65	33.87
2006-04-11 14:04:08.81892	9.04	0.00	4.11	1.28	8.36	0.00	0.00	21.65	21.65	33.87
2006-04-11 14:09:09.762609	8.57	0.00	3.90	1.21	7.96	0.06	0.00	20.55	20.55	37.17
2006-04-11 14:14:10.707525	8.93	0.00	4.06	1.27	8.36	0.19	0.00	21.48	21.48	34.20
2006-04-11 14:19:11.652024	8.97	0.02	4.07	1.29	8.35	0.00	0.00	21.51	21.51	34.25
2006-04-11 14:24:12.595917	8.88	0.08	4.00	1.33	8.15	0.00	0.00	21.08	21.08	35.35

A good starting point for the SQL Characteristics query is the query template *DB2PM.Statistic.SQL DML.Quantity*.

The stored procedure call throughput is a key performance metric in an IBM Content Management V8.3 environment. The Library Server uses stored procedures to process any user request to the Library Server database.

Monitoring the stored procedure throughput over time helps you detect trends regarding changing throughput or changing workload mix in your Content Management environment.

If you detect degraded stored procedure throughput over time, then run *RUNSTATS* and *REBIND* first to make sure that the database statistics are up-to-date. You might see a throughput improvement after running these utilities.

Monitoring the stored procedure throughput by using DB2 statistics traces is very straightforward. Throughput is the number of event occurrences per time interval. The time interval during which DB2 writes statistics records to SMF is defined by using the *Statistics Time Interval ZPARM (STATIME)*. In the sample environment, the value of *STATIME* is set to 5, which is 5 minutes.

The following query calculates the stored procedure throughput per minute. Moreover, the thread creation throughput per minute and the number of bytes sent and received per minute through DDF is calculated. Each CM application client that connects to the Library Server database and the Resource Manager database on z/OS causes that a DB2 thread is created or reused on the DB2 subsystem. Thus, the thread creation throughput is an indicator for the transaction rate in your Content



Management environment. Most of the documents that are managed by a Content Management system are stored to or retrieved from workstation clients. Thus, the number of sent and received bytes helps you determine if the size characteristics of the managed documents changes over time.

```
SELECT
  G.END_REC_TSTAMP,
  DECIMAL(G.STPROC_CALL/INTEGER(G.INTERVAL_ELAPSED/60),31,2) AS CALL_STP_PER_MIN,
  DECIMAL((G.CREATE_THREAD + G.DBATS_CREATED)/INTEGER(G.INTERVAL_ELAPSED/60),31,2) AS
CRT_THRDS_PER_MIN,
  DECIMAL(D.BYTES_SENT/INTEGER(D.INTERVAL_ELAPSED/60),31,2) AS BYTES_SENT_PER_MIN,
  DECIMAL(D.BYTES_RCD/INTEGER(D.INTERVAL_ELAPSED/60),31,2) AS BYTES_RCD_PER_MIN
FROM DB2PM.DB2PM_STAT_GENERAL G, DB2PM.DB2PM_STAT_DDF D
WHERE G.LOCAL_LOCATION = D.LOCAL_LOCATION
AND G.GROUP_NAME = D.GROUP_NAME
AND G.SUBSYSTEM_ID = D.SUBSYSTEM_ID
AND G.MEMBER_NAME = D.MEMBER_NAME
AND G.BEGIN_REC_TSTAMP = D.BEGIN_REC_TSTAMP
AND G.LOCAL_LOCATION = :loc
AND G.GROUP_NAME = :group
AND G.SUBSYSTEM_ID = :subsystem
order by G.END_REC_TSTAMP
```

The screenshot below shows an example query result displayed in HTML format:

END_REC_TSTAMP	CALL_STP_PER_MIN	CRT_THRDS_PER_MIN	BYTES_SENT_PER_MIN	BYTES_RCD_PER_MIN
2006-04-11 13:29:02.206986	325.00	465.00	612086.00	132459.00
2006-04-11 13:34:03.152636	484.00	726.60	684214.00	185710.00
2006-04-11 13:39:04.0971	329.60	494.60	468208.40	129971.60
2006-04-11 13:44:05.040839	471.80	705.20	696775.00	181868.80
2006-04-11 13:49:05.984169	856.00	1283.60	1204642.00	320002.00
2006-04-11 13:54:06.929138	5.20	5.80	14838.60	12629.00
2006-04-11 13:59:07.874118	0.00	0.00	7190.40	11139.20
2006-04-11 14:04:08.81892	0.00	0.00	7098.00	10986.00
2006-04-11 14:09:09.762609	0.00	0.00	7098.00	10986.00
2006-04-11 14:14:10.707525	0.00	0.00	7098.00	10986.00
2006-04-11 14:19:11.652024	0.40	0.20	8139.20	11545.80
2006-04-11 14:24:12.595917	1.60	0.80	10227.80	11678.60

6.4.2 Analysis of thread resource consumption

So far, statistics data that is stored in the Performance Warehouse gave a high-level overview of the performance of the Library Server in a Content Management system. In this section, the focus is on DB2 thread performance, which means the performance of individual DB2 applications that are connected to the Library Server or the Resource Manager databases. For this process, you need accounting data that is stored in the Performance Warehouse database.

You must set up a Performance Warehouse process that is scheduled to start after the SMF switches take place. The accounting data is then read from the SMF data set and stored in the Performance Warehouse database.

You can use the Performance Warehouse process template *DB2PM.Templates.Load Warehouse 6* to implement scheduled loading of accounting data into the Performance Warehouse database. The Performance Warehouse process template uses the reporting function to load the SMF data. The report step of the process is configured in the Performance Warehouse GUI as shown in Figure 30.

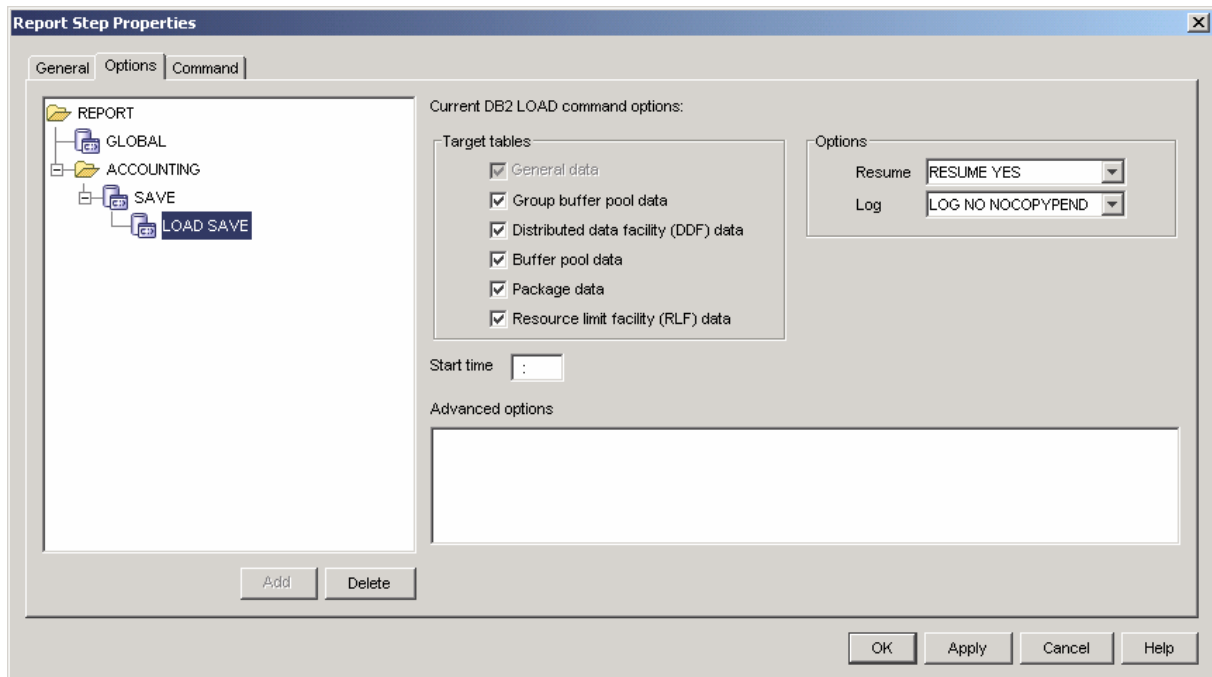


Figure 30: Performance Warehouse GUI - Report Step Properties

To analyze the accounting data that is stored in the Performance Warehouse database, you can use the following queries:

1. Identify long running applications
2. Identify long running applications that spent most of their time in DB2 either working or waiting for resources
3. Identify DB2 packages of CM applications that are candidates for tuning

Note: A DB2 package is an object that contains a set of statically bound SQL statements and is stored in the DB2 system catalog. A CM application client that performs static SQL statements, for example in a stored procedure call, deploys DB2 packages.

Identify long running applications

To identify long running applications, you must calculate the average elapsed time of each application. Because the majority of CM applications connect through DRDA to the Library Server, you must specify a GROUP BY clause by using the thread attributes *PLAN_NAME*, *CLIENT_TRANSACTION*, *CLIENT_ENDUSER*, *CLIENT_WSNAME*. This identifies if the elapsed time of the application depends on the transaction, the user, the workstation, or any combination of these attributes. In this example, an application is assumed to be long running if the elapsed time (*CLASS1_ELAPSED*) exceeds five seconds. You can adapt this threshold value to your Content Management environment. The query is as follows:

```
SELECT
  PLAN_NAME ,
  CLIENT_TRANSACTION ,
  CLIENT_ENDUSER ,
  CLIENT_WSNAME ,
  SUM(CLASS1_ELAPSED) / SUM(OCCURRENCES) AS AVG_ELAPSED_TIME
FROM DB2PM.DB2PMSACCT_GENERAL
GROUP BY PLAN_NAME, CLIENT_TRANSACTION, CLIENT_ENDUSER, CLIENT_WSNAME
HAVING (SUM(CLASS1_ELAPSED) / SUM(OCCURRENCES)) >= 5
ORDER BY AVG_ELAPSED_TIME DESC
```

This screenshot shows an example query result in HTML format:



PLAN_NAME	CLIENT_TRANSACTION	CLIENT_ENDUSER	CLIENT_WSNAME	AVG_ELAPSED_TIME
java.exe	java.exe	cmadmin	OBSTLER	1898.662444
DSNESPRR				538.872289
ICMClient	ICMClient.exe	CMADMIN	TESTWIN2K	7.236451

Note that accounting data is stored in the Performance Warehouse database in a pre-aggregated way. The column *OCCURRENCES* tracks the number of DB2 threads, which are accumulated in one row in the Performance Warehouse accounting tables. You can use them to calculate average times and events.

Identify long running applications, which spent most of their time in DB2

The query above can be modified to display only DB2 applications that spent most of their time in DB2, which means the ratio of *CLASS2_ELAPSED_TIME/CLASS1_ELAPSED_TIME* is greater than or equal to 50%.

Note: DB2 applications that spent the biggest part of their processing time in DB2 are the best candidates for database tuning.

```
SELECT
  PLAN_NAME,
  CLIENT_TRANSACTION,
  CLIENT_ENDUSER,
  CLIENT_WSNAME,
  SUM(CLASS1_ELAPSED) / SUM(OCCURRENCES) AS AVG_ELAPSED_TIME,
  SUM(CLASS2_ELAPSED) / SUM(OCCURRENCES) AS AVG_IN_DB2_TIME,
  DOUBLE(SUM(CLASS2_ELAPSED))/DOUBLE(SUM(CLASS1_ELAPSED)) AS IN_DB2_PERCENT
FROM DB2PM.DB2PMSACCT_GENERAL
GROUP BY PLAN_NAME, CLIENT_TRANSACTION, CLIENT_ENDUSER, CLIENT_WSNAME
HAVING SUM(CLASS1_ELAPSED) / SUM(OCCURRENCES) >= 5
AND DOUBLE(SUM(CLASS2_ELAPSED))/DOUBLE(SUM(CLASS1_ELAPSED)) >= 0.5
ORDER BY AVG_ELAPSED_TIME DESC
```

This screenshot shows an example query result in HTML format:

PLAN_NAME	CLIENT_TRANSACTION	CLIENT_ENDUSER	CLIENT_WSNAME	AVG_ELAPSED_TIME	AVG_IN_DB2_TIME	IN_DB2_PERCENT
ICMClient	ICMClient.exe	CMADMIN	TESTWIN2K	7.236451	7.132686	0.9856607205893662

Identify packages that are candidates for tuning

You should check by using EXPLAIN, if the DB2 packages that are identified by the following query benefit from additional indexes. As a rule of thumb, all packages that perform more than 100 get pages per statement execution are tuning candidates.

```
SELECT
  SUM( P.SQL_STMTS_ISSUED) AS #STMT_ISSUED,
  SUM( DOUBLE(B.BP_GETPAGES))/SUM( CASE WHEN P.SQL_STMTS_ISSUED=0 THEN NULL
                                     ELSE P.SQL_STMTS_ISSUED END) AS GET_PAGES_PER_STMT,
  P.PLAN_NAME,
  P.PCK_ID,
  P.CONNECT_TYPE
FROM DB2PM.DB2PMSACCT_PROGRAM P, DB2PM.DB2PMSACCT_BUFFER B
WHERE P.PRIMAUTH = B.PRIMAUTH
AND P.MAINPACK = B.MAINPACK
AND P.PLAN_NAME = B.PLAN_NAME
AND P.PLAN_NAME IN ( SELECT PLAN_NAME
                     FROM DB2PM.DB2PMSACCT_GENERAL
                     GROUP BY PLAN_NAME, CLIENT_TRANSACTION, CLIENT_ENDUSER, CLIENT_WSNAME
                     HAVING SUM(CLASS1_ELAPSED) / SUM(OCCURRENCES) >= 5
                     AND DOUBLE(SUM(CLASS2_ELAPSED))/DOUBLE(SUM(CLASS1_ELAPSED)) >= 0.5 )
AND P.CLIENT_TRANSACTION=B.CLIENT_TRANSACTION
AND P.SUBSYSTEM_ID = B.SUBSYSTEM_ID
GROUP BY P.PLAN_NAME, P.PCK_ID, P.CONNECT_TYPE
HAVING SUM( DOUBLE(B.BP_GETPAGES)) / SUM( CASE WHEN P.SQL_STMTS_ISSUED=0 THEN NULL ELSE
P.SQL_STMTS_ISSUED END) > 100
ORDER BY 2 DESC
```



This screenshot shows an example query result in HTML format:

#STMT_ISSUED	GET_PAGES_PER_STMT	PLAN_NAME	PCK_ID	CONNECT_TYPE
2	1166.5	ICMCIen	ICMPLSAM	DRDA
4	583.25	ICMCIen	ICMPLSCW	DRDA
4	583.25	ICMCIen	ICMPLSPC	DRDA
12	194.41666666666666	ICMCIen	SYSSTAT	DRDA

Note that this application analysis strategy is a short version of the more explicit version that Performance Warehouse offers by using the *Expert Analysis Queries*. These queries are part of the OMEGAMON XE for DB2 Performance Expert on z/OS query templates.

For details about the Expert Analysis Queries, refer to the Redbook: *IBM DB2 Performance Expert for z/OS Version 2, SG 246867*.



7 Appendix A: Installation tokens

Token name	Description	Example
?ACLPLAN?	DB2 plan for ICMPCACL.	ICMACLPL
?BP32?	Buffer pool for table space ICMLFQ32 and ICMLNF32.	BP32K
?BPS4?	Buffer pool for library server tables.	BP4
?BPV4?	Buffer pool for library server indexes.	BP4
?CREATOR?	ID of the creator defining the DB2 objects.	ICMCRTOR
?DB2CONNECTID?	User ID of the DB2 connect ID. It must be a TSO user ID with DB2 user privileges. You can use this user ID as the DB2 connect ID for DB2 Content Manager Version 8 users who do not have a TSO user ID on z/OS.	ICMCONT
?DB2PKGCOLLID?	Collection name of the DB2 package	ICMMLS
?ICM?	High-level qualifier of the DB2 Content Manager Version 8 library.	ICMV830
?ICMADMIN?	User ID of the default system administrator; the ID must be a TSO user ID with DB2 system administration privileges.	ICMADMN
Token name	Description	Example
?ICMMLSP?	Customized JCL for the user table access program preparation: ICMMLSP1, ICMMLSP2, or ICMMLSP3	This token is no longer used.
?ICMMLSQ?	Customized JCL for the user table creation SQL job: ICMMLSQ1, ICMMLSQ2, or ICMMLSQ3.	ICMMLSQ3
?NUMTCB?	The number of task control blocks (TCBs) used to process end user requests.	32
?PREFETCHCOLL?	Default resource manager prefetch collection name if the default resource manager does not use any device other than a disk. Then, the following SQL can be removed from SYSIN input data of this job: <pre>INSERT INTO ?CREATOR?.ICMSTCo11Name- VALUES (1, 0, '?ICMADMIN?', CURRENT TIMESTAMP-CURRENT TIMEZONE, 1, 2, '?PREFETCHCOLL?') ;</pre>	OAM.CLLCT002
?RMLOGINID?	Default resource manager user ID. Assign a user ID for this resource manager as follows: <ul style="list-style-type: none"> • RACF[®] (ACF/2 or Top Secret) user ID for DB2 Content Manager for z/OS • LAN logon ID for LAN base (This job loads a dummy password for the user ID into the DB2 Content Manager Version 8 table ICMSTResourceMgr. Upon completion of installation, change the password of the user ID for this resource manager using the system administration client.)	RMADMIN
?RMPORNUM?	Default resource manager port number.	8080
?SAPPLAN?	DB2 plan for ICMPLSAP.	ICMSAPPL
?SMSCOLL?	Default resource manager collection name.	OAM.CLLCT001

For more information please refer to <http://www-1.ibm.com/support/docview.wss?uid=swg21220345>



8 Appendix B: Space Requirements

During the installation of the Content Manager on a DB2 on z/OS subsystem, several tables are created by the installation job *ICMMLSCR*.

When planning the installation, you should consider table sizes, record lengths of the data in the tables, and if the table is subject to a large number of queries and updates. You can customize your installation by updating the *ICMMLSCR* installation job to configure your storage volumes and the buffer pools of your DB2 subsystem.

8.1.1 Library Server

Table Sizes

The default DDL to create the Library Server and Resource Manager tables creates simple DB2 tables, which means that the DDL does not use segmented or partitioned table spaces. Segmented and partitioned table spaces are typically used in a production environment to handle large sized tables. DB2 Content Manager DB2 table sizes are categorized as follows:

S	Small	< 500 rows
M	Medium	500 – 9999 rows
L	Large	> 10000 rows

Table name	Record length	Table size	(High query)	(High update)
ICMSTTHELOG ¹	-	-		
ICMSTACCESSCODES	12	S		
ICMSTACCESSLISTS	50	S		
ICMSTACTIONLISTSCM	8	S		
ICMSTACTIONSCM	5938	S		
ICMSTALISTCDCM	4	S		
ICMSTADMINDOMAINS	12	S		
ICMSTATTRDEFS	26	S	Y	
ICMSTATTRGROUP	18	S	Y	
ICMSTCHECKEDOUT	92	S	Y	Y
ICMSTCOLLNAME	306	S	Y	
ICMSTCOMPATTRS	549	S		
ICMSTCOMPATTRSFK	344	S		
ICMSTCOMPDEFS	71	S		
ICMSTCOMPILEDACL	87	M-L	Y	
ICMSTCOMPILEDPERM	48	M-L	Y	
ICMSTCOMPVIEWATTRS	283	S		
ICMSTCOMPVIEWDEFS	106	S		
ICMSTDOMAINACCESS	16	S		
ICMSTDOMAINPRIVSET	16	S		
ICMSTITEMAUTOLINK	38	S		
ICMSTITEMEVENTS	1365	L ²		
ICMSTITEMSTODELETE	609	S-M		Y ³
ICMSTITEMS001001	84	L	Y	Y
ICMSTITEMTYPEDEFS	138	S	Y	
ICMSTITEMTYPEREL	28	S		
ICMSTITEMVER001001	84	L ⁴	Y	Y
ICMSTITVIEWDEFS	62	S		
ICMSTITVIEWID	16	S		



Table name	Record length	Table size	(High query)	(High update)
ICMSTREPLICAS	72	L		
ICMSTRESOURCEMGR	568	S		
ICMSTRI001001	144	L ⁵	Y	
ICMSTRMACCESSTYPES	149	S		
ICMSTSYSADMEVENTS	1339	L ²		
ICMSTSYSCONTROL	152	S		
ICMSTTEXTINDEXCONF	1504	S		
ICMSTUSERGROUPS	72	S-M		
ICMSTUSERS	1241	S-M		
ICMSTXDOOBJECTS	786	S		
ICMSTXDOOBJFACTORY	273	S		
ICMST390CONTROL	528	S		
ICMUT00200001 (routing process)	463	S		
ICMUT00201001 (routing steps)	213	S		
ICMUT00202001 (work step)	1643	M-L	Y	Y
ICMUT00203001 (collection resume list)	139	M-L	Y	Y
ICMUT00204001 (work package)	348	M-L	Y	Y
ICMUT00205001 (item resume list)	135	S-M	Y	Y
ICMUT00206001 (work list)	475	M	Y	
ICMUT00207001 (work step list)	155	M-L	Y	
ICMUT00208001 (container data)	417	S		
ICMUT00209001	1,000,114	S		
ICMUT00210001	548	S	Y	
ICMUT00300001 (BASE)	347	L	Y	Y
ICMUT00301001 (BASETEXT)	908	L	Y	Y
ICMUT00302001 (BASESTREAM)	597	L	Y	Y
ICMUT00303001 (NOTELOG)	908	L	Y	Y
ICMUT00304001 (ANNOTATION)	347	L	Y	Y
ICMUT00400001 (CLIENTEXIT)	301	S		
ICMSTLINKS001001	101	L	Y	Y
ICMSTMAXKEYWORD	9	S		
ICMSTMIMETYPES	1818	S		
ICMSTNLSKEYWORDS	307	S		
ICMSTNLSLANGUAGES	45	S		
ICMSTPRIVDEFS	12	S-M		
ICMSTPRIVGROUPCODE	12	S		
ICMSTPRIVGROUPS	16	S		
ICMSTPRIVSETCODES	12	S		
ICMSTPRIVSETS	16	S		
ICMSTREPLICARULES	18	S		

In Section 6.2.5 you see how the OMEGAMON Classic Interface can be used to monitor I/O traffic in your DB2 environment. This function of OMEGAMON XE for DB2 Performance Expert on z/OS helps you analyze your table space design to improve the I/O rate. If you experience performance problems regarding I/O, you should adjust the DDL of the created CM objects (Section 5.2).



Customization

Depending on CM usage, tables might be populated differently. Some ICM tables are created to support special operations, such as versioning and logging. If your CM applications use these kinds of operations, consider the following:

If event logging is turned on, tables ICMSTITEMEVENTS and ICMSTSYSADMEVENTS might become very large. A CM system that uses versioning requires a very large ICMSTITEMVER001001 table. Thus, turning versioning on and off can change your overall response time. Deleting many items at the same time, causes high update processing on the ICMSTITEMSTODELETE table.

8.1.2 Resource Manager

Table Sizes

Table name	Record length	Table size	High query	High update
ICMRMASYNCH	1108	L	Y	Y
ICMRMCONTROL	182	S	Y	
ICMRMKEY	198	S	Y	
ICMRMPREFETCH	488	L	Y	Y
ICMRMTRK	706	S-M	Y	Y
ICMRMCOLLECTION	206	S-M	Y	Y
ICMRMSERVER	334	S-M	Y	Y
ICMRMMGMTCLASS	17	S	Y	



9 Appendix C: References

DB2 Content Manager

DB2 Content Management Version 8.3 Information Center

<http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0/topic/com.ibm.cmgmt.hlp/welcome.htm>

White paper Ute Baumbach: Monitoring IBM DB2 Content Manager V8.3 Enterprise Edition with DB2 Performance Expert V2.2 for MP

<http://www-1.ibm.com/support/docview.wss?uid=swg27006673>

DB2 Content Manager System Administration Guide

<http://www.elink.ibm.com/public/applications/publications/cgi-bin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC27-1335-07>

IBM Content Manager v8.2 Performance Tuning Guide

<http://www-1.ibm.com/support/docview.wss?uid=swg27003894&aid=1>

IBM DB2 Content Manager Enterprise Edition V8.3 Performance Tuning Guide

<http://www.ibm.com/support/docview.wss?uid=swg27006452>

IBM DB2 Content Manager Enterprise Edition V8.3 Performance Monitoring and Maintenance Guide

<http://www-1.ibm.com/support/docview.wss?uid=swg27006451>

OMEGAMON XE for DB2 Performance Expert on z/OS

Redbook: A Deep Blue View of DB2 Performance: IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS

<http://www.redbooks.ibm.com/redpieces/abstracts/sg247224.html>

Redbook: IBM DB2 Performance Expert for z/OS Version 2

<http://www.redbooks.ibm.com/abstracts/sg246867.html>

Developer Works Article 'Part 1,2,3 Simplify Performance Management and Tuning with DB2 Performance Expert'

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0409schuetz/>

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0409schuetz2/index.html>

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0507baumbach/>

IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS: Monitoring Performance from the Performance Expert Client, SC18-9640

<http://publibz.boulder.ibm.com/epubs/pdf/fpempc10.pdf>

IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS: Monitoring Performance from the OMEGAMON Classic Interface, SC18-9659-00

<http://publibz.boulder.ibm.com/epubs/pdf/fpecic10.pdf>

Additional Resources

IBM z/OS: DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support, SC35-0426

<http://publibz.boulder.ibm.com/epubs/pdf/dgt2o230.pdf>

DB2 Connect User's Guide Version 8, SC09-4835

<http://users.sdsc.edu/~jrowley/db2/DB2%20Connect%20Users%20Gd.pdf>