QMF Data Service
13.1

Solution Configuration Guide

SC28-3266-00

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" at the end of this information.

May 31, 2022 edition

This edition applies to Version 13 Release 1 IBM® Db2 Query Management Facility (QMF) Classic Edition and Enterprise Edition, which are features of IBM Db2 13 for z/OS (5698-DB2). It also applies to Version 13 Release 1 of IBM Db2 QMF for z/OS (5698-QMF), which is a stand-alone IBM Db2 for z/OS tool. This information applies to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation .
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
© Rocket Software Inc. 2016, 2022.

Contents

About IBM® DB2 QMF® Data Service

IBM® DB2 QMF® Data Service (Data Service) provides integration facilities for both mainframe data sources and other off-platform data sources. Data Service provides optimized, virtualized, and parallelized access to a wide variety of data.

This information describes how to configure the Data Service for access to your mainframe data.

What's new in IBM® DB2 QMF® Data Service Solution Configuration Guide

This section describes recent technical changes to IBM® DB2 QMF® Data Service.

New and changed information is marked like this paragraph. Editorial changes that have no technical significance are not marked.

Description	Related APARs
Virtual table rule support is provided for overriding data buffer and index buffer values for VSAM files for individual requests. See "Modifying the data and index buffer values for VSAM files" on page 50.	PI99385
Virtual table rule support is provided for specifying the number of tracks to read ahead (MULTACC) when reading sequential data sets for individual requests. See "Reading ahead tracks for sequential file access" on page 52.	PH01433
You can control whether native Db2 database subsystems appear in ISPF and the Data Service Studio and if attempts to connect to native Db2 subsystems are allowed. See "Controlling display and access for native Db2 subsystems " on page 39.	PH00610
When streaming SMF data, the requester can use a SQL SELECT statement to stream SMF data in real time, directly from the SMF in-memory buffer. The connection to the SMF in-memory resource is made at the time of the request, and the SQL statement does not reach end of data until the server is stopped or the request is canceled. See "Configuring access to SMF data for IT Operational Analytics" on page 53 and "Configuring access to System Management Facility (SMF) files" on page 54.	PI97277
Db2 Direct is a new Data Service server access method used to access Db2 data by reading the data in the underlying Db2 VSAM linear data sets directly. See "Db2 for z/OS data access methods" on page 22 and "Configuring Db2 Direct" on page 24.	PI95733
The process of creating maps to access VSAM and sequential data has been simplified by support of the following methods: • Querying information in the IBM Application Discovery and Delivery Intelligence (ADDI) dictionary. See "Configuring access to ADDI" on page 58. • Querying information in the IBM Rational Asset Analyzer (RAA) dictionary. See "Configuring access to RAA" on page 65.	PI94393
SQL query access to DB2 unload data sets is now provided. See "Configuring access to DB2 unload data sets" on page 20.	PI94374
The Data Service server can now listen for ENF 55 auxiliary storage shortage signals and throttle storage utilization when an auxiliary storage shortage is signaled. The point at which the Data Service server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility is controlled by the server parameter DSCLIENTAUXSTGCUTOFF. See "Modifying the client auxiliary storage cut-off parameter" on page 100.	PI94026

Chapter 1. Overview

The QMF Data Service server supports a broad range of data sources, including mainframe relational/non-relational databases and file structures, distributed databases running on Linux, UNIX, and Windows platforms, Cloud based relational and non-relational data, and NoSQL databases. IBM® DB2 QMF® Data Service solutions have a range of connectivity options for data consumers, including QMF Workstation/WebSphere, QMF Vision, and QMF for TSO/CICS.

Table 1. Supported data source types

Data support	**Data source type**
Mainframe relational/non-relational databases	• IBM® Db2 for z/OS • IBM® Information Management System (IMS) • IBM® MQ • CA IDMS • Software AG Adabas
Mainframe file structures	• Sequential files • Native VSAM files • z/OS File System (zFS)
Mainframe applications and screens	• IBM® CICS® • VSAM via IBM® CICS® Transaction Server • IBM® InfoSphere™ Federation Server • IBM® Query Management Facility (QMF) • IBM® System Management Facility (SMF) • Software AG Natural • z/OS system logging (SYSLOG)
Distributed data stores running on Linux, Unix, and Windows platforms	• IBM® BigInsights Hadoop • IBM® dashDB • IBM® Db2 • IBM® Db2 Big SQL • IBM® Informix • Apache Derby • Microsoft SQL Server • Oracle

Chapter 2. The SQL solution

To enable ANSI SQL access to mainframe data sources, configure IBM® DB2 QMF® Data Service.

Configuring a solution can include one or more of the following tasks:

- Configure the started task JCL by modifying the `CQD1PROC` member that is in the *hlq*.SCQDCNTL library.
- Configure the server member that is included in data set member *hlq*.CQDS.SCQDEXEC(CQDSIN00).
- Make definition changes in the data provider interface.

Before configuring the SQL solution, the Data Service server installation must be successfully completed.

You can also use the Data Service Studio to get SQL access to your data.

For information about configuring the SQL solution, see the following topics.

Configuring Data Service with DB2 QMF for TSO and CICS

Before using Data Service with DB2 QMF for TSO and CICS, verify that the configuration is complete. The instructions for setting up Data Service are described in the *Installing and Managing DB2 QMF for TSO and CICS* book.

Procedure

1. Ensure that the IBM® DB2 QMF® Data Service DB2 package, CQDHLI, has been bound to QMF, see QMF job in `QMF1210.SDSQSAPE(DSQ1BPKQ)`.
2. For TSO QMF, ensure that the Data Service dataset `CQD.SCQDLOAD` has been allocated to all TSO users STEPLIB or ISPLLIB.
3. For QMF CICS ensure that the `QMF1210.SDSQSAPE(DSQ1ECSQ)` job was run to update the CICS region's CSD that includes the Data Service modules CQDHLI and CQDCLAPI.
4. For QMF CICS ensure that the Data Service load library, `CQD.SCQDCLOD`, has been allocated to the CICS DFHRPL.

Configuring access to data sources

The server supports access to many mainframe and distributed data sources. You can find detailed information for the specific data interface you want to use.

Before you begin

The server must be installed.

About this task

To configure a data source, see the following tasks:

- "Configuring access to data in Adabas" on page 5
- "Configuring access to data in relational database management systems (RDBMS)" on page 8
- "Configuring access to data in IBM IMS databases" on page 42
- "Configuring access to VSAM" on page 50
- "Configuring access to sequential files" on page 52

Chapter 3. Configuring access to data sources

Configure Data Service server to enable access to mainframe data sources.

Configuring access to data in Adabas

To access Adabas, you need to configure the started task JCL and the server configuration member.

Before you begin

The server must be installed.

Procedure

To configure access to data in Adabas, perform the following tasks:

a) Configure the server started task. See "Configuring the server started task JCL" on page 5.

b) Modify the server configuration member. See "Modifying the server configuration member" on page 5.

c) Configure security access to Adabas data. See "Configuring Adabas security" on page 7.

Configuring the server started task JCL

Make the ADALNKR module available.

Before you begin

All LOAD library data sets allocated to the Data Service server in the server started task JCL must be APF-authorized.

About this task

Note: You can skip this task if the ADALNKR module is in the z/OS linklist.

Procedure

1. Add the Adabas LOAD library to the server started task JCL. Uncomment the ADALOAD parameter and set it to the correct Adabas load library name.

   ```
   ADALOAD='ADABAS.LOAD'
   ```

2. Uncomment the reference to the LOAD library in the STEPLIB.

Modifying the server configuration member

Enable the Adabas parameters in the server configuration member.

About this task

The server configuration member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

1. In the CQDSIN00 member, locate the comment "ENABLE ADABAS DATABASE SERVER SUPPORT."
2. Enable the Adabas parameters by changing the syntax `if DontDoThis` to `if DoThis`.

Set the ADABAS parameter to YES. The following example shows the section in the configuration member to enable:

```
if DoThis then
  do
        “MODIFY PARM NAME(ADABAS)VALUE(YES)”
        “MODIFY PARM NAME(ADABASUBINFOSIZE)VALUE(256)”
        “MODIFY PARM NAME(ADABASAUTOCOMMITBIND)VALUE(YES)”

        “MODIFY PARM NAME(ACIMAPREDUCEADAB)VALUE(64000)”
        “MODIFY PARM NAME(ACIMAPREDUCEADAISN)VALUE(64000)”
end
```

The following table lists the parameters for configuring support for Adabas data stores:

Parameter	Description	Valid values
ACIMAPREDUCEADAB	Map Reduce Adabas Record Buffer Size - Allows Adabas Multi-Fetch used to read records via L1 commands. If the Adabas ADARUN limits are exceeded, an Adabas response code 53 is issued.	Buffer size in bytes. 64000 (default value)
ACIMAPREDUCEADAISN	Map Reduce Adabas ISN Buffer Size - When a Key Descriptor is used in a Search query, an Adabas S1 search is performed. The resulting internal sequence number (ISN) Record number list is divided up into separate Map Reduce threads.	Buffer size in bytes. 64000 (default value)
ADABAS	Activates support for Adabas data stores.	**NO** Support is not active. (default value) **YES** Activate support.
ADABASAUTOCOMMITBIND	Activates support for the AUTOCOMMIT BIND option.	**YES** Activate support. (default value) **NO** Support is not active.
ADABASUBINFOSIZE	Specifies the total amount of space to allocate for user information and review information in the Adabas user block. Review the maximum user information size in the ADALNKR, and increase the value of this parameter to be equal to or greater than the maximum user information size.	256 KB (default value)

Configuring Adabas security

Configure security to access Adabas data at a DBID or file number level.

About this task

Securing Adabas files at a DBID or file number level requires the use of the following Data Service server parameters:

- RESOURCETYPE
- SQLVTRESOURCETYPE
- ADABASSECURITY

The following sample jobs for defining Adabas security-related definitions are provided in the *hlq*.SCQDCNTL library:

- CQDRAADA for RACF

  Note: When using job CQDRAADA, make the following changes for file ID security:

  ```
  RDEFINE FACILITY ADAxxxxx.FILyyyyy UACC(NONE)
  PERMIT ADAxxxxx.FILyyyyy CLASS(FACILITY) ID(<USERID>)
  ACCESS(aaaa)
  SETROPTS REFRESH RACLIST(FACILITY)
  ```

  – Change *xxxxx* to the Adabas database ID.

  – Change *yyyyy* to the Adabas file ID.
- CQDA2ADA for CA ACF2
- CQDTSADA for CA Top Secret

Procedure

1. Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Customization Guide*.
2. Ensure the following settings are set in the CQDIN00 file:

```
MODIFY PARM NAME(RESOURCETYPE) VALUE(RAVZ)
MODIFY PARM NAME(SQLVTRESOURCETYPE) VALUE(RAVZ)
MODIFY PARM NAME(ADABASSECURITY) VALUE(YES)
```

Parameter name	Parameter description	Value
RESOURCETYPE	RESOURCE TYPE FOR RESOURCE RULES Specify the name of the security server's class (or resource type for ACF2) that is used to perform resource access authorization checks. When using RACF, the corresponding class name within RACF must start with R, for example, RCQD.	For RACF: RCQD
SQLVTRESOURCETYPE	RESOURCE TYPE FOR SQL ACCESS TO VIRTUAL TABLES Specify the name of the security server's class (or resource type for ACF2) that is used to perform authorization checks for SQL access to metadata and virtual tables in the SQL Engine. When using RACF, the corresponding class name within RACF must start with R, for example, RCQD.	For RACF: RCQD

Parameter name	Parameter description	Value
ADABASSECURITY	ADABAS SECURITY ACTIVATED Set this parameter to indicate that a resource rule is to be constructed consisting of DBID and file. **Note:** Both RESOURCETYPE and SQLVTRESOURCETYPE must be set in order for ADABASSECURITY to be in effect.	YES

Configuring access to data in relational database management systems (RDBMS)

You can access data on DB2 for z/OS and distributed databases IBM Big SQL, IBM dashDB, DB2 LUW (Linux, UNIX, and Windows), Microsoft SQL Server, Oracle, and QMF DRDA.

Before you begin

The server and relational database management system (RDBMS) must already be installed.

About this task

To configure and verify access to data in a RDBMS, perform the following tasks.

Procedure

1. Enable the RDBMS access method in the Data Service configuration member.

   See "Modifying the server configuration member for DRDA" on page 8.
2. Configure access to the database.
   - IBM DB2 for z/OS

     Configure DB2 to use the Distributed Relational Database Architecture (DRDA) access method.

     See "Configuring access to IBM DB2 for z/OS" on page 14.
   - Distributed databases, including Big SQL, dashDB, DB2 LUW, Microsoft SQL Server, Oracle, and QMF DRDA.

     Configure the RDBMS to use the Distributed Relational Database Architecture (DRDA) access method.

     See "Configuring access to distributed databases" on page 26.

Modifying the server configuration member for DRDA

If you are using a zIIP specialty engine, enable the RDBMS access method for Distributed Relational Database Architecture (DRDA) in the server configuration member.

About this task

Configure the server to use Distributed Relational Database Architecture (DRDA) when accessing a RDBMS.

Modify the server configuration member in data set *hlq*.CQDS.SCQDEXEC(CQDSIN00). The server configuration member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

1. Verify that the Unicode translation of the Coded Character Set Identifier (CCSID) used in the `DEFINE DATABASE` statement and the CCSID used by the target RDBMS are defined for your z/OS environment.

   a) You should identify the CCSID of the RDBMS.

   For example, Oracle may use *ccsid1*. In your `DEFINE DATABASE` statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1-ccsid2* and *ccsid2-ccsid1* defined in your Unicode translation table on z/OS using the command **`D UNI,ALL`**.

   b) If the entry is not present, you need to add the entry to your Unicode translation table and refresh.

   Please refer to the IBM z/OS documentation on how to add the entry.

   Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

   ```
   "DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE)"
   "DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE)"
   ```

2. In the CQDSIN00 member, locate the section that contains the comment `Enable DRDA access to DB2 database subsystems`.
3. Enable the DRDA parameters by changing the syntax `if DontDoThis` to `if DoThis`, and then set the DRDASKIPZSERVICES parameter to YES. The following example shows the section in the configuration member to enable:

   ```
   /*------------------------------------------------------------------*/
   /* Enable DRDA access to DB2 database subsystems                    */
   /*------------------------------------------------------------------*/
   if DoThis then
     do
         “MODIFY PARM NAME(TRACEOEDRDARW)        VALUE(YES)”
         “MODIFY PARM NAME(CLIENTMUSTELECTDRDA)  VALUE(NO)”
         “MODIFY PARM NAME(DRDASKIPWLMSETUP)     VALUE(NO)”
         “MODIFY PARM NAME(DRDAFORLOGGINGTASK)   VALUE(NO)”
         “MODIFY PARM NAME(DRDASKIPZSERVICES)    VALUE(YES)”
   ```

   The following table describes these parameters:

Parameter	Description	Valid values
TRACEOEDRDARW	If set to YES (recommended), TCP/IP communications via DRDA are traced. If set to NO, DRDA receive and send operations are not traced.	**YES** **NO** Default value.
CLIENTMUSTELECTDRDA	If set to YES, JDBC clients must explicitly opt in for DRDA to be used by setting the user parameter connection variable to 'DRDA'. **Note:** JDBC clients can always opt out of DRDA processing by setting the user parameter to 'NODRDA'. If set to NO, DRDA processing is used for access all configured RDBMSs.	**YES** **NO** Default value.

Parameter	Description	Valid values
DRDASKIPWLMSETUP	If set to YES, WLM information is not collected and sent to DRDA during JDBC logon processing. If captured, the DRDA equivalent to SET_CLIENT_ID calls is issued after logon to establish these values on the DRDA connection. If not captured, the transmission that is used to set these WLM-related values is bypassed. If set to NO, the client user ID, application name, workstation name, and accounting token that were sent in the initial client buffer are collected and sent separately after logon processing to DRDA.	**YES** **NO** Default value.
DRDAFORLOGGINGTASK	If set to YES, DRDA processing is used for the Db2 on z/OS logging subtask. If set to NO, SAF or RRSAF connections are used. **Note:** Passticket support must be enabled for the target DDF server. If passticket support is not configured, set the parameter to NO.	**YES** **NO** Default value.
DRDASKIPZSERVICES	Prevents DRDA from being used for z/Service Db2 processing. If set to YES, z/Services client tasks do not use DRDA processing for Db2 requests. If set to NO, DRDA will be used when configured for a particular Db2 connection. **Note:** Passticket support must be enabled for all target DDF servers.	**YES** **NO** Default value.

4. Define DRDA RDBMSs by entering a definition statement. Provide your local environment values for all the parameters. The following example shows the section in the configuration member to enable:

```
"DEFINE DATABASE TYPE(type_selection)"         ,
                 "NAME(name)"                  ,
                 "LOCATION(location)"          ,
                 "DDFSTATUS(ENABLE)"              ,
                 "DOMAIN(your.domain.name)"    ,
                 "PORT(port)"                  ,
                 "IPADDR(1.1.1.1)"             ,
                 "CCSID(37)"                   ,
                    "APPLNAME(DSN1LU)"              ,
                    "IDLETIME(110)"                 ,
```

Where *type_selection* is either GROUP, MEMBER, or ZOSDRDA.

The previous example shows only a subset of the available parameters. The following table lists all available parameters for defining DDF endpoints:

Parameter	Description	Valid values
APPLNAME	Application name. The APPLNAME used by the target endpoint for passticket generations. (*Optional*)	A valid value is 1 - 8 characters. If APPLNAME is not specified in the definition statement, no default value is provided and passticket access is disabled. **Note:** APPLNAME is not required when connecting from the JDBC driver.
AUTHTYPE	Authentication type. This can be either DES (Diffie Hellman Encryption Standard) or AES (Advanced Encryption Standard). When AUTHTYPE is not supplied, the default is DES. To force AES, the option must be added to the DEFINE DATABASE statement. Each server can be different in what is supported as to AES/DES. For this setting to have effect, you must specify a security mechanism (SECMEC) that requests encryption.	**DES** Diffie Hellman Encryption Standard (default value) **AES** Advanced Encryption Standard.
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier) configured for this RDBMS subsystem on the RDBMS installation panel DSNTIPF, option 7. (*Optional*)	Refer to the RDBMS vendor documentation for a list of valid CCSID.
DDFSTATUS	The DDF activation status can be altered online by using the ISPF 4-DB2 dialog panels. (*Required*)	**ENABLE** To make this DDF definition active within Data Service server. **DISABLE** DDF endpoint is not used.
DOMAIN	The part of a network address that identifies it as belonging to a particular domain.	No default value.

Parameter	Description	Valid values
IDLETIME	If Db2 ZPARM parameter IDTHTOIN is set to a non-zero value set IDLETIME to a value slightly less (10 secs.) than IDTHTOIN. This will also allow product DRDA threads to become inactive. (*Db2 for z/OS only*)	0-9999 seconds.
IPADDR	Specify the dot-notation IPV4 address of the DDF endpoint. (*Optional*)	If this parameter is not specified, the value 127.0.0.1 (local host) is the default. For group director definitions, use the DVIPA IP address of the group director.
LOCATION	For Db2: The Db2 location name. For LUW: The LUW database. For Oracle: The Oracle SSID as defined to the Oracle Database Provider (Gateway) (*Required*)	A valid value is a string 1 - 16 characters.
NAME	The database name as known to the server. (*Required*)	A valid value consists of 1 - 4 characters. Clients use this ID when they request access to a specific Db2 subsystem.
PORT	The TCP/IP port at which the server is listening. (*Required*)	If this keyword is not entered, the default DRDA port number 443 is used.
SECMEC	The DRDA security mechanism in force. (*For GROUP and MEMBER types.*)	**USERIDPWD** User ID and password are sent as is. No encryption is used. **USRIDONL** User ID is sent as is. No encryption is used for the user ID only (client security). **USRENCPWD** Encrypt the password only. **EUSRIDPWD** Encrypt the user ID and password.

Parameter	Description	Valid values
SYSTEMVCAT	The VCATNAME for the Db2 system catalog tables (in the DSNDB06 database). The VCATNAME for system catalog tables is a system bootstrap value and not available using the data discovery query. Use this parameter if you intend to access the system catalog tables using Db2 Direct or if the VCATNAME for database DSNDB06 is different from the subsystem name.	A valid value is 1 - 8 characters. If this parameter is not specified, the 4-character Db2 subsystem name is used by default as the high-level qualifier for Db2 data sets.
TYPE	For Db2 for z/OS: **GROUP** DDF endpoint is a Db2 group director. **MEMBER** DDF endpoint is a Db2 instance or group member for z/OS. **ZOSDRDA** DDF endpoint is a remote z/OS Db2 on another LPAR. This setting allows you to use SEF ATH rules when z/OS Pass Ticket passwords cannot be used or the server administrator has the requirement to manage the authentication credentials for remote z/OS systems.	For Db2 for z/OS: GROUP MEMBER ZOSDRDA

Verifying Coded Character Set Identifier (CCSID) values and definitions

Verify that the Unicode translation of the CCSID used in the `DEFINE DATABASE` statement and the CCSID used by the target RDBMS are defined for your z/OS environment.

Procedure

1. You should identify the CCSID of the RDBMS.

   For example, Oracle may use *ccsid1*. In your `DEFINE DATABASE` statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1-ccsid2* and *ccsid2-ccsid1* defined in your Unicode translation table on z/OS using the command **`D UNI,ALL`**.
2. If the entry is not present, you need to add the entry to your Unicode translation table and refresh. Please refer to the IBM z/OS documentation on how to add the entry.

   Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

   ```
   "DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE)"
   "DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE)"
   ```

Configuring access to IBM DB2 for z/OS

About this task

Before you issue DB2 requests, you must bind DRDA into packages within each DB2 subsystem.

Procedure

1. "Configuring security" on page 14
2. Configure for DRDA (Distributed Relational Database Architecture) or for RRSAF (Resource Recovery Services attachment facility) access method.
   - If you are using a zIIP specialty engine, enable the RDBMS access method for DRDA:
     a. "Modifying the server configuration member for DRDA" on page 8
     b. "Configuring DB2 for DRDA" on page 16
   - If you are not using a zIIP specialty engine, enable the RDBMS access method for RRSAF:
     a. "Modifying the server configuration member for RRSAF" on page 16
     b. "Configuring DB2 for RRSAF" on page 18

Configuring security

Configure security to provide user access to DB2.

About this task

If the DB2 being accessed does not have the `DSNZPARM DDF` option `TCPALVER` set to either YES or `CLIENT`, then a passticket is needed for certain DB2 on z/OS DRDA operations. These operations may include:

- Refreshing in-memory metadata catalog information at server startup for DB2 on z/OS defined virtual tables. Catalog information is refreshed at every server startup by the Data Service server connecting to each DB2 where virtual tables have been defined.
- Any SQL statement coming from the dsClient interface, dsSpufi or application APIs using the dsClient interface. This may also include running administrative tasks in batch using dsClient that accesses DB2 on z/OS such as updating MapReduce information using the `DRDARange` command.

Procedure

1. This step only applies to DB2 for z/OS. To grant users access to the DB2 subsystem and to enable passticket logon processing, you must define one RACF PTKTDATA resource for each unique DRDA APPLNAME. To define each PTKTDATA resource, customize and run the appropriate job.
   - `CQDRADB2` is for IBM Resource Access Control Facility (RACF) security.
   - `CQDA2DB2` is for CA ACF2 (Access Control Facility) security.
   - `CQDTSDB2` is for CA Top Secret Security (TSS).
2. Assign users READ authority.
   - For DRDA, assign users READ authority to the `*ssid*.DIST` profile.

Related concepts

Db2 for z/OS data access methods
Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring the server started task JCL

If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Configuring DB2 for DRDA
If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Configuring access to DB2 unload data sets
To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Configuring the server started task JCL

If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Before you begin

All LOAD library data sets allocated to the Data Service server in the server started task JCL must be APF-authorized.

Procedure

Edit the JCL in the *hlq*.SCQDCNTL(CQD1PROC) member to include in the PROC statement the DB2LIB parameter with the Db2 library name assigned, as shown in the following example:

```
DB2LIB='DSNX10'
```

The Db2 library must contain the Db2 interface modules, such as DSNALI and DSNHLI, and must be in uppercase and enclosed in quotation marks.

Related concepts

Db2 for z/OS data access methods
Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring DB2 for DRDA
If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Configuring access to DB2 unload data sets

To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Configuring DB2 for DRDA

If you are using a zIIP specialty engine, configure DB2 to use DRDA.

About this task

Before you can successfully issue DRDA requests, you must bind IBM® DB2 QMF® Data Service DBRMs into packages within each target DB2 subsystem.

Procedure

1. Set the DEFAULTDB2SUBSYS parameter in the server configuration member to a valid DB2 subsystem name.
2. Edit the CQDBINDD job that is supplied in the *hlq*.SCQDCNTL data set.

   Follow the instructions that are provided in the JCL.
3. Run the CQDBINDD job.

Related concepts

Db2 for z/OS data access methods
Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring the server started task JCL
If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Configuring access to DB2 unload data sets
To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Modifying the server configuration member for RRSAF

If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

About this task

This task is only applicable for DB2 for z/OS.

Modify the server configuration member in data set *hlq*.CQDS.SCQDEXEC(CQDSIN00). The server configuration member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

1. Verify that the Unicode translation of the Coded Character Set Identifier (CCSID) used in the `DEFINE DATABASE` statement and the CCSID used by the target RDBMS are defined for your z/OS environment.

   a) You should identify the CCSID of the RDBMS.

   For example, Oracle may use *ccsid1*. In your `DEFINE DATABASE` statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1-ccsid2* and *ccsid2-ccsid1* defined in your Unicode translation table on z/OS using the command **`D UNI,ALL`**.

   b) If the entry is not present, you need to add the entry to your Unicode translation table and refresh.

   Please refer to the IBM z/OS documentation on how to add the entry.

   Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

```
"DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE)"
"DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE)"
```

2. Set the DEFAULTDB2SUBSYS parameter in the server configuration member CQDSIN00 to a valid DB2 subsystem name.
3. In the CQDSIN00 member, locate the comment “ENABLE DB2 RRSAF SUPPORT” section.
4. Enable the RRSAF parameters by changing the syntax `if DontDoThis` to `if DoThis`. The following example shows the section in the configuration member to enable:

```
if DoThis then
  do
        “MODIFY PARM NAME(RRS)                VALUE(YES)”
        “MODIFY PARM NAME(DB2ATTACHFACILIT)  VALUE(RRS)”
        “MODIFY PARM NAME(TRACERSSDATA)      VALUE(NO)”
        “MODIFY PARM NAME(TRACERSSEVENTS)    VALUE(YES)”
        “MODIFY PARM NAME(TRACERSSAF)        VALUE(YES)”
end
```

   The following table lists the parameters for configuring support for RRSAF:

Parameter	Description	Valid values
DB2ATTACHFACILITY	Specifies the DB2 attach facility. The Resource Recovery Services attachment facility (RRSAF) uses the DSNRLI interface module and allows for 2–phase commit actions. The Call Attach Facility (CAF) uses the DSNALI interface module.	The default value is RRS. Valid values are RRS and CAF.
RRS	Activates RRS support. This parameter must be set to YES to activate RRS.	**YES** Default value. **NO**
TRACERSSDATA	Specifies whether to trace RRS data.	**YES** Default value. **NO**
TRACERSSEVENTS	Specifies whether to trace RRS events.	**YES** Default value. **NO**

Parameter	Description	Valid values
TRACERSSAF	Creates an entry in the server trace for each call to DSNRLI for RRSAF requests.	**YES** Default value. **NO**

Related concepts

Db2 for z/OS data access methods
Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring the server started task JCL
If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Configuring DB2 for DRDA
If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Configuring access to DB2 unload data sets
To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Configuring DB2 for RRSAF

If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

About this task

This task only applies to DB2 for z/OS.

Procedure

1. Run the CQDBINDC member of the *hlq.*SCQDCNTL data set to bind the following server product plans:
   - CQDC1010 is bound using cursor stability.
   - CQDR1010 is bound using repeatable read.
   - CQDS1010 is bound using read stability.
   - CQDU1010 is bound using uncommitted read.

   Use CQDC1010 as the default server plan, and use the other product plans for operations that require those levels of isolation. To change the default plans, edit the BIND member and replace the default plan names with new names. You must run the BIND job of the *hlq*.SCQDCNTL data set against each DB2 subsystem that you want to access. Use the instructions in the JCL to customize the job.
2. Install the `DSN3@SGN` exit in the DB2 master task (normally placed in the SDSNEXIT data set). Installing this exit enables the server to use DB2 authority that was granted through secondary DB2 authorization IDs.

Related concepts

Db2 for z/OS data access methods

Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring the server started task JCL
If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Configuring DB2 for DRDA
If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Configuring access to DB2 unload data sets
To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Disabling query acceleration

You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

About this task

By default, the server sends the command **SET CURRENT QUERY ACCELERATION = ENABLE WITH FAILBACK** to a DRDA server if it is DB2 for z/OS. This setting allows access to accelerator tables but does not prevent access to non-accelerator tables. Sending the command can be suppressed using the virtual table rule CQDMDTBL by setting the field **OPTBNOQA** to **1** in the rule. If sending the command is suppressed and the table is an accelerator only table, the query will fail. This setting has effect only when the table is owned by a DB2 for z/OS subsystem and the table is an accelerator table; otherwise, there is no impact to the processing.

Use the following procedure to set up the rule.

Procedure

1. Customize the server configuration member (CQDSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

2. Access the VTB rules, as follows:
   a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.
   b) Specify option 2, **SEF Rule Management**.
   c) Enter VTB for **Display Only the Ruleset Named**.
3. Customize the CQDMDTBL rule, as follows:
   a) Specify S next to CQDMDTBL to edit the rule.
   b) Find the **VTB.OPTBNOQA** variable and set to 1 to turn query acceleration off.
   c) Save your changes and exit the editor.
4. Enable the rule by specifying E next to CQDMDTBL and pressing Enter.
5. Set the rule to Auto-enable by specifying A next to CQDMDTBL and pressing Enter.
   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Related concepts

Db2 for z/OS data access methods
Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring the server started task JCL
If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Configuring DB2 for DRDA
If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Configuring access to DB2 unload data sets
To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Configuring access to DB2 unload data sets

To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

About this task

To configure access to a DB2 unload data set, you must add the DB2 unload data set name to the DB2 virtual table in a Server Event Facility (SEF) virtual table rule. With this access, you can issue SQL queries directly against DB2 unload data sets using existing DB2 virtual tables.

Switching a DB2 virtual table to read an unload data set is done by assigning a data set name to the table in a virtual table rule. The VTB variable **vtb.optbdsna** is used to redirect access from DB2 to reading the sequential file named in the variable. The named sequential file must contain unload data created by the DB2 UNLOAD utility. A model VTB rule, CQDMDLDU, is provided to demonstrate redirecting a DB2 virtual table to a DB2 unload data set.

As an example, consider a virtual table named DSNA_EMPLOYEES that maps the EMPLOYEES table in DB2 subsystem DSNA. By activating the model rule CQDMDLDU, you can query an unload sequential dataset named EMPLOYEE.UNLOAD.SEQ by issuing the following query:

```
SELECT * FROM MDLDU_DSNA_EMPLOYEES__EMPLOYEE_UNLOAD_SEQ
```

The CQDMDLDU rule performs the following steps:

1. Extracts the table name DSNA_EMPLOYEES and sets the VTB variable **vtb.optbmtna**.
2. Extracts the data set name EMPLOYEE_UNLOAD_SEQ, converts the underscores to periods, and sets the VTB variable **vtb.optbdsna**.

The following restrictions and considerations apply when using this feature:

- SQL access to DB2 unload files is limited to SQL queries only.
- The columns in DB2 virtual table definition must exactly match the table unloaded in DB2.

Use the following procedure to configure the sample rule CQDMDLDU.

Note: Sample rule CQDMDLDU is intended to be used as a model and may require customization. When customizing this rule, additional logic may need to be added if different unload data sets require different VTB variable settings for CCSID or internal/external format.

Procedure

1. Customize the server configuration member (CQDSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

2. Access the VTB rules, as follows:
   a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.
   b) Specify option 2, **SEF Rule Management**.
   c) Enter VTB for **Display Only the Ruleset Named**.
3. Customize the CQDMDLDU rule, as follows:
   a) Specify S next to CQDMDLDU to edit the rule.
   b) Find the **vtb.optbdsna** variable and specify the name of the DB2 unload data set to process.
   c) Update additional rule options as needed. The following table describes the VTB rule options that support DB2 unload data set access.

VTB variable	Description
vtb.optbdlcv	If the data was unloaded with a DELIMITED statement, set **vtb.optbdlcv** to 1 to declare the data is in delimited format. It may also be necessary to declare the delimiters if the default column delimiter (,) and character string delimiter (“) were overridden when the data was unloaded.
vtb.optbdsna	Specifies the name of the sequential unload data set created by the DB2 UNLOAD utility to access.
vtb.optbduif	By default, the DB2 unload utility writes data in external format. If FORMAT INTERNAL is used when unloading data, **vtb.optbduif** must be set to 1 to declare that the data was unloaded in internal format.
vtb.optbmtna	Specifies the map name of the DB2 virtual table describing the unload file.
vtb.optbtbcc	If the table CCSID is not compatible with the CCSID defined for the SQL engine (CQDSIN00 SQLENGDFLTCCSID parameter), **vtb.optbtbcc** can be used to declare the CCSID of the data. This is particularly important for Unicode tables and tables containing GRAPHIC columns.

   d) Save your changes and exit the editor.
4. Enable the rule by specifying E next to CQDMDLDU and pressing Enter.
5. Set the rule to Auto-enable by specifying A next to CQDMDLDU and pressing Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Related concepts

Db2 for z/OS data access methods
Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring the server started task JCL
If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Configuring DB2 for DRDA

If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Db2 for z/OS data access methods

Db2 for z/OS data can be accessed by the Data Service server using different data access methods.

The following Db2 for z/OS data access methods are available:

- Traditional Db2 access. This method accesses Db2 data through traditional Db2 APIs. This access method allows for reading and writing of the data and provides transactional integrity.
- Db2 Direct. This method accesses Db2 data by reading the underlying Db2 VSAM linear data sets directly. This access method allows read-only access to the data and provides high performance, bulk data access.

The Db2 data access method is specified when creating virtual tables in the Data Service Studio for access to Db2 data.

The following topics provide more information about the Db2 for z/OS data access methods.

Using traditional Db2 access

Traditional Db2 access methods access Db2 data through APIs such as Distributed Relational Database Architecture (DRDA), Call Attachment Facility (CAF), and Resource Recovery Services attachment facility (RRSAF). Using traditional Db2 access allows for reading and writing of the data and provides transactional integrity.

Traditional DB2 access methods provide MapReduce and Virtual Parallel Data support. MapReduce is an algorithm that enables the Data Service server to streamline how it accesses Db2 data, thereby reducing the processing time required to virtualize Db2 data. Statistics about the Db2 database are collected and stored within a metadata repository from which the SQL engine optimizes the MapReduce process.

In order to exploit MapReduce for Db2 when using traditional Db2 access, the Data Service server must collect information about the Db2 database. This information is collected using the **DRDARange** command and is stored within the Data Service server metadata repository.

Traditional Db2 access is used automatically when Db2 Direct access is not available.

Using Db2 Direct

Db2 Direct is a Data Service server access method that reads the data in the Db2 VSAM linear data sets directly instead of accessing the data through traditional Db2 APIs. Using Db2 Direct, large data pulls can be performed in service request block (SRB) mode, and MapReduce and Virtual Parallel Data features can by exploited without any prerequisite processing, such as the collection of statistics using the **DRDARange** command. Db2 Direct access provides a significant increase in performance and reduced elapsed time in processing analytical type queries.

Db2 Direct allows read-only access to the data. When using Db2 Direct, there is no locking involved when accessing the data, so updates may not be captured and deleted records may have been captured. Results from Db2 Direct queries may be out of sync with the current state of a Db2 table due to recent table updates not being flushed to the linear data sets.

Security is managed using Db2 table authorization.

Restrictions and considerations:

Consider the following points when using Db2 Direct:

- The Db2 subsystem hosting a Db2 table must be active when Db2 Direct-enabled tables are loaded or refreshed in the data server. The map build process requires Db2 system access to identify data set information in the Db2 system catalog.
- The Data Service server requires read access to the Db2 VSAM linear data sets. The linear data sets containing the Db2 rows must be available to the data server processing SQL requests for Db2 data. If the data sets are unavailable or archived, Db2 Direct will be disabled during map load or refresh for the virtual table.
- Virtual tables enabled for Db2 Direct must include all the columns defined in the base Db2 table. This is necessary because the columns describe the internal format of the Db2 data.
- If Db2 is not available or some other error occurs during map build or map refresh processing, Db2 Direct is automatically disabled for the table and a message is written to the trace log:

  ```
  DB2 direct processing disabled for map map-name
  ```

- If Db2 Direct processing is disabled, processing will continue with traditional Db2 APIs when possible.
- To determine if Db2 Direct is active, the following messages appear in the server trace:
  - At startup and map refresh, the following message is issued:

    ```
    DB2 direct processing enabled for map map-name
    ```

  - When DB2 Direct is used in a query, the following message is issued:

    ```
    Processing table map-name using DB2 direct
    ```

- If Db2 Direct table security is enabled, the Db2 subsystem must be available to check security at SQL query time.
- If Db2 Direct table security is disabled, unauthorized users who would normally receive a -551 SQLCODE attempting to access data through traditional APIs may gain access to Db2 data.
- Db2 Direct does not support tables with edit procedures or SQL statements containing joins, LOB columns, or key columns.
- If Db2 Direct security is disabled, the CCSIDs of table columns will be assumed based on the ENCODING_SCHEME (EBCDIC, Unicode, ASCII) of the table.

Related tasks

Configuring security
Configure security to provide user access to DB2.

Configuring the server started task JCL
If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Configuring DB2 for DRDA
If you are using a zIIP specialty engine, configure DB2 to use DRDA.

Modifying the server configuration member for RRSAF
If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the server configuration member.

Configuring DB2 for RRSAF
If you are not using a zIIP specialty engine, configure RRSAF for access to local DB2.

Disabling query acceleration
You can use a Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

Configuring access to DB2 unload data sets

To be able to access a DB2 unload data set directly with an SQL query, you must configure a virtual table rule to define the DB2 unload data set name to the DB2 virtual table.

Configuring Db2 Direct

Configure Db2 Direct options or disable Db2 Direct.

Before you begin

Review the restrictions and considerations when using Db2 Direct. See “Using Db2 Direct” on page 22.

About this task

By default, Db2 Direct is enabled in the Data Service server. Use the information in this topic to perform the following optional tasks:

- Disable the Db2 Direct feature for a virtual table by using a Virtual Table (VTB) rule.
- Define the VCATNAME for the DB2 system catalog tables (in the DSNDB06 database) by modifying the `DEFINE DATABASE` statement. The VCATNAME for system catalog tables is a system bootstrap value and is not available using the data discovery query. This task is required only in the following situations:
  - Access to system catalog tables using Db2 Direct is intended.
  - The VCATNAME for database DSNDB06 is different from the subsystem name.
- Configure Db2 Direct options, such as the number of pages to allocate for Db2 segment information, whether to enforce Db2 SQL table security authorizations, and disabling Db2 Direct for the server, by modifying server parameters.
- Specify what Db2 Direct information to display in the server trace by modifying server parameters.

Procedure

1. To disable the Db2 Direct feature for a virtual table, in a VTB rule, set the variable **OPTBDIDD** to 1. For additional information, see the generic sample rule CQDMDTBL.
2. To define the VCATNAME for the Db2 system catalog tables, perform the following steps:
   a) Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization.
   b) In the `DEFINE DATABASE` statement, use the SYSTEMVCAT parameter to define the VCATNAME for the system catalog tables, as shown in the following example:

```
"DEFINE DATABASE TYPE(MEMBER)"                     ,
               "NAME(DBA9)"                       ,
               "LOCATION(RS28DDS9)"               ,
               "DDFSTATUS(ENABLE)"                ,
               "PORT(3725)"                       ,
               "IPADDR(127.0.0.1)"                 ,
               "CCSID(37)"                        ,
               "APPLNAME(DBA9DB2)"                ,
               "SYSTEMVCAT(DDS9)"                 ,
               "IDLETIME(110)"
```

3. To modify server parameters, perform the following steps:
   a) Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization.
   b) Use the **`MODIFY PARM`** command to change a parameter value. For example, the following command disables Db2 Direct for the Data Service server:

```
"MODIFY PARM NAME(DISABLEDB2DIRECT)    VALUE(YES)"
```

   The parameters in the following tables are available for use with Db2 Direct.

Table 2. SQL parameters in group PRODSQL

Parameter name	Parameter description	Default value
DB2DIRECTSEGTBLPAGES	DB2-DIRECT SEGMENT TABLE PAGES Defines the number of 4K pages to be allocated for Db2 segment information. The default value is 8, which should be enough for most Db2 Direct queries. This parameter should only be changed if a query fails because the Db2 Direct segment table was exhausted.	8
DISABLEDB2DIRECT	DISABLE DB2-DIRECT PROCESSING Disables Db2 Direct processing in the server.	NO
DISABLEDB2DIRSEC	DISABLE DB2-DIRECT TABLE SECURITY Disables SQL table security checking when Db2 Direct is selected to process Db2 data. Disabling table security checking will allow access to Db2 data when the target Db2 subsystem is not active. **Important:** Unauthorized users who would normally receive a -551 SQLCODE attempting to access data through traditional APIs like DRDA may gain access to Db2 data.	NO

Table 3. SQL parameters in group PRODTRACE

Parameter name	Parameter description	Default value
TRACEDB2DIRSTATS	TRACE DB2-DIRECT STATISTICS Enables tracing of a summary report to the system trace after each Db2 Direct query. Included in the trace are statistics about read and point operation in the Db2 linear data set(s) processed.	NO
TRACEDB2DIROPEN	TRACE DB2-DIRECT OPEN CONTROL BLOCKS Enables tracing of control blocks created at the open of each linear data set for Db2 Direct processing.	NO

Table 3. SQL parameters in group PRODTRACE (continued)

Parameter name	Parameter description	Default value
TRACEDB2DIRSEGP	TRACE DB2-DIRECT SEGMENT PAGES Enables tracing if Db2 pages containing segmented map information.	NO
TRACEDB2DIRDICTP	TRACE DB2-DIRECT DICTIONARY PAGES Enables tracing of the compression dictionary used to compress and expand rows stored in Db2 linear data sets.	NO
TRACEDB2DIRDATAP	TRACE DB2-DIRECT DATA PAGES Enables tracing of data pages in a linear data set containing Db2 rows.	NO
TRACEDB2DIRROWS	TRACE DB2-DIRECT ROWS Enables tracing of rows extracted from data pages in a Db2 linear data set. If rows are compressed, an additional trace is created of the uncompressed row data.	NO

Configuring access to distributed databases

You can configure access to data on Big SQL, dashDB, DB2 LUW (Linux, UNIX, and Windows), Microsoft SQL Server, Oracle, and QMF DRDA.

Before you begin

If you are connecting to a Big SQL or DB2 LUW database, then you must install and configure the IBM DB2 Federated Server. For additional information, refer to the documentation on the IBM website.

If you are connecting to an Oracle database, then you must install and configure the Oracle Database Provider for DRDA. For additional information, refer to the documentation on the Oracle website.

If you are connecting to a 2016 Microsoft SQL Server database, then you must install and configure the Host Integration Server for HIS DRDA Service. For additional information, refer to the documentation on the Microsoft website. The SYSIBM Views from Microsoft must be installed.

About this task

Configure access to distributed databases by modifying the configuration member, configuring Server Event Facility (SEF) rules, and optionally setting up alternate authentication information.

Procedure

1. "Modifying the server configuration member" on page 27.
2. Configure the Server Event Facility rules and set up authentication for the appropriate database.
   - "Configuring rules and authentication for Big SQL" on page 32.
   - "Configuring rules and authentication for dashDB" on page 33.

- “Configuring rules and authentication for LUW databases” on page 34.
- “Configuring rules and authentication for Microsoft SQL Server” on page 35.
- “Configuring rules and authentication for Oracle DRDA” on page 37.
- “Configuring rules and authentication for QMF DRDA Server” on page 38.

Modifying the server configuration member

Enable the RDBMS access method in the Data Service server configuration member.

About this task

Configure the server to use Distributed Relational Database Architecture (DRDA) when accessing a RDBMS.

Modify the server configuration member in data set *hlq*.CQDS.SCQDEXEC(CQDSIN00). The server configuration member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

1. Verify that the Unicode translation of the Coded Character Set Identifier (CCSID) used in the `DEFINE DATABASE` statement and the CCSID used by the target RDBMS are defined for your z/OS environment.
   a) Identify the CCSID of the RDBMS.

      For example, Oracle may use *ccsid1*. In your `DEFINE DATABASE` statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1*-*ccsid2* and *ccsid2*-*ccsid1* defined in your Unicode translation table on z/OS using the command **`D UNI,ALL`**.

   b) If the entry is not present, add the entry to your Unicode translation table and refresh.

      Refer to the IBM z/OS documentation on how to add the entry.

      Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

      ```
      "DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE)"
      "DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE)"
      ```

2. In the CQDSIN00 member, locate the section that contains the comment “Enable DRDA access to DB2 database subsystems.”
3. Enable the DRDA parameters by changing the syntax `if DontDoThis` to `if DoThis` and then set the `DRDASKIPZSERVICES` parameter to YES. The following example shows the section in the configuration member to enable:

   ```
   if DoThis then
     do
         “MODIFY PARM NAME(TRACEOEDRDARW)        VALUE(YES)”
         “MODIFY PARM NAME(CLIENTMUSTELECTDRDA)  VALUE(NO)”
         “MODIFY PARM NAME(DRDASKIPWLMSETUP)     VALUE(NO)”
         “MODIFY PARM NAME(DRDAFORLOGGINGTASK)   VALUE(NO)”
         “MODIFY PARM NAME(DRDASKIPZSERVICES)    VALUE(YES)”
   ```

   The following table lists the parameters for configuring support for DRDA:

Parameter	Description	Valid values
CLIENTMUSTELECTDRDA	If set to YES, JDBC clients must explicitly opt in for DRDA to be used by setting the user parameter connection variable to 'DRDA'. **Note:** JDBC clients can always opt out of DRDA processing by setting the user parameter to 'NODRDA'. If set to NO, DRDA processing is used for access to all configured RDBMSs.	**YES** **NO** Default value.
DRDAFORLOGGINGTASK	If set to YES, DRDA processing is used for the DB2 on z/OS logging subtask. If set to NO, SAF or RRSAF connections are used. **Note:** Passticket support must be enabled for the target DDF server. If passticket support is not configured, set the parameter to NO.	**YES** **NO** Default value.
DRDASKIPWLMSETUP	If set to YES, WLM information is not collected and sent to DRDA during JDBC logon processing. If captured, the DRDA equivalent to SET_CLIENT_ID calls is issued after logon to establish these values on the DRDA connection. If not captured, the transmission that is used to set these WLM-related values is bypassed. If set to NO, the client user ID, application name, workstation name, and accounting token that were sent in the initial client buffer are collected and sent separately after logon processing to DRDA.	**YES** **NO** Default value.

Parameter	Description	Valid values
DRDASKIPZSERVICES	Prevents DRDA from being used for z/Service DB2 processing. If set to YES, z/Services client tasks do not use DRDA processing for DB2 requests. If set to NO, DRDA will be used when configured for a particular DB2 connection. **Note:** Passticket support must be enabled for all target DDF servers.	**YES** **NO** Default value.
TRACEOEDRDARW	If set to YES (recommended), TCP/IP communications via DRDA are traced. If set to NO, DRDA receive and send operations are not traced.	**YES** **NO** Default value.

4. Define DRDA RDBMSs by entering a definition statement. Provide your local environment values for all the parameters. The following example shows the section in the configuration member to enable:

```
"DEFINE DATABASE TYPE(type_selection)"         ,
                 "NAME(name)"                  ,
                 "LOCATION(location)"          ,
                 "DDFSTATUS(ENABLE)"              ,
                 "DOMAIN(your.domain.name)"    ,
                 "PORT(port)"                  ,
                 "IPADDR(1.1.1.1)"             ,
                 "CCSID(37)"                   ,
                 "APPLNAME(DSN1LU)"            ,
                 "IDLETIME(110)"               ,
```

This is an example for dashDB:

```
"DEFINE DATABASE TYPE(DASHDB)"                 ,
                 "NAME(name)"                  ,
                 "LOCATION(location)"          ,
                 "AUTHTYPE(AES)"               ,
                 "SECMEC(EUSRIDPWD)"           ,
                 "DDFSTATUS(ENABLE)"              ,
                 "DOMAIN(your.domain.name)"    ,
                 "PORT(port)"                  ,
                 "CCSID(37)"                   ,
```

The following table lists the parameters for defining DDF endpoints:

Parameter	Description	Valid values
APPLNAME	Application name. The APPLNAME used by the target endpoint for passticket generations. (*Optional*)	A valid value is 1 - 8 characters. If APPLNAME is not specified in the definition statement, no default value is provided and passticket access is disabled. **Note:** APPLNAME is not required when connecting from the JDBC driver.

Parameter	Description	Valid values
AUTHTYPE	Authentication type. This can be either DES for Diffie Hellman Encryption Standard or AES for Advanced Encryption Standard. When AUTHTYPE is not supplied, the default is DES. To force AES, the option must be added to the DEFINE DATABASE statement. Each server can be different in what is supported as to AES/DES.	**DES** Diffie Hellman Encryption Standard (default value) **AES** Advanced Encryption Standard.
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier) configured for this RDBMS subsystem on the RDBMS installation panel DSNTIPF, option 7. (*Optional*)	Refer to the RDBMS vendor documentation for a list of valid CCSIDs.
DDFSTATUS	The DDF activation status can be altered online by using the ISPF 4-DB2 dialog panels. (*Required*)	**ENABLE** Make this DDF definition active. **DISABLE** DDF endpoint is not used.
DOMAIN	The part of a network address that identifies it as belonging to a particular domain.	No default value.
IDLETIME	If DB2 ZPARM parameter IDTHTOIN is set to a non-zero value set IDLETIME to a value slightly less (10 secs.) than IDTHTOIN. This will also allow product DRDA threads to become inactive. (*DB2 for z/OS only*)	0-9999 seconds.
IPADDR	Specify the dot-notation IPV4 address of the DDF endpoint. (*Optional*)	If this parameter is not specified, the value 127.0.0.1 (local host) is the default. For group director definitions, use the DVIPA IP address of the group director.

Parameter	Description	Valid values
LOCATION	For DB2: The DB2 location name. For dashDB: This is the database name of the dashDB database or alias name for the database. For LUW: The LUW database. For Oracle: The Oracle SSID as defined to the Oracle Database Provider (Gateway). (*Required*)	A valid value is a string 1 - 16 characters.
NAME	The database name as known to the server. (*Required*)	A valid value consists of 1 - 4 characters. Clients use this ID when they request access to a specific DB2 subsystem.
PORT	The TCP/IP port at which the server is listening. (*Required*)	A valid 1-5 numeric string.
SECMEC	The DRDA security mechanism in force for standard dashDB services requires an authentication method setting. Define as either USRENCPWD, which informs the server to encrypt the PASSWORD or EUSRIDPWD, which informs the server to encrypt the USERID and PASSWORD during the initial connection to dashDB. (*Except QMFDRDA*)	**USRENCPWD** Encrypt password only. **EUSRIDPWD** Encrypt userid and password.

Parameter	Description	Valid values
TYPE	For distributed databases: **BIGSQL** DDF endpoint is a Big SQL engine. **DASHDB** DDF endpoint is a dashDB database. **LUW** DDF endpoint is a DB2 instance or group member for Linux, UNIX, or Windows. **MSSQL** DDF endpoint is a DB2 instance or group member for Microsoft SQL Server. **ORACLE** DDF endpoint is an Oracle instance. The parameter informs DRDA AR and supportive tooling that the remote server is an Oracle Database Provider which supports DRDA AS. The Oracle DRDA AS must be in z/OS simulation mode. **QMFDRDA** DDF endpoint is a QMF DRDA AS Object Server instance.	For distributed databases: BIGSQL DASHDB LUW MSSQL ORACLE QMFDRDA

Configuring rules and authentication for Big SQL

Configure Server Event Facility (SEF) rules and set up authentication to provide access to Big SQL databases.

About this task

To complete configuration for access to Big SQL databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to Big SQL. By default, the server will attempt to log on to Big SQL with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing Big SQL.

This task uses the following tools:

CQDSBIGC
An SQL rule that allows Meta discovery on Big SQL databases.

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDEBIGG
An ATH rule that switches credentials when connecting to a Big SQL database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SCQDXSQL(CQDSBIGC) to allow Data Service Studio Meta discovery on Big SQL databases.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or SQL to display only SQL rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
   Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

   When adding the SYSIN statements that define the alternate credentials for logging in to your Big SQL database, as instructed in the JCL, make sure to specify the correct DBTYPE. For Big SQL, specify `DBTYPE=BIGSQL`.
   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SCQDXATH(CQDEBIGG) to provide the logon credentials to each Big SQL instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or ATH to display only authentication rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
   Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for dashDB

Configure Server Event Facility (SEF) rules and set up authentication to provide access to IBM dashDB databases.

About this task

To complete configuration for access to dashDB databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to dashDB. By default, the server will attempt to log on to dashDB with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing dashDB.

This task uses the following tools:

CQDSDDBC
An SQL rule that allows Meta discovery on dashDB databases.

CQDDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDEDDBG

An ATH rule that switches credentials when connecting to a dashDB database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SCQDXSQL(CQDSDDBC) to allow Data Service Studio Meta discovery on dashDB databases.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or SQL to display only SQL rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
   Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

   When adding the SYSIN statements that define the alternate credentials for logging in to your dashDB database, as instructed in the JCL, make sure to specify the correct DBTYPE. For dashDB, specify DBTYPE=DASHDB.

   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SCQDXATH(CQDEDDBG) to provide the logon credentials to each dashDB instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or ATH to display only authentication rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
   Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for LUW databases

Configure Server Event Facility (SEF) rules and set up authentication to provide access to LUW (Linux, UNIX, and Windows) databases, including databases connected via IBM Federated Server.

About this task

To complete configuration for access to LUW databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to LUW databases. By default, the server will attempt to log on to the LUW database with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing an LUW database.

This task uses the following tools:

CQDSLUWC
An SQL rule that allows Meta discovery on LUW databases.

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDELUWG
An ATH rule that switches credentials when connecting to an LUW database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SCQDXSQL(CQDSLUWC) to allow Data Service Studio Meta discovery on LUW databases.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or SQL to display only SQL rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.

      Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

      When adding the SYSIN statements that define the alternate credentials for logging in to your LUW database, as instructed in the JCL, make sure to specify the correct DBTYPE. For LUW databases, specify `DBTYPE=LUW`.
   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SCQDXATH(CQDELUWG) to provide the logon credentials to each LUW instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or ATH to display only authentication rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.

      Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for Microsoft SQL Server

Configure Server Event Facility (SEF) rules and set up authentication to provide access to Microsoft SQL Server via the 2016 Host Integration Server for HIS DRDA Service.

About this task

To complete configuration for access to Microsoft SQL Server, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to SQL Server. By default, the Data Service server will attempt to log on to SQL Server with the same user ID that was presented for logon to z/OS. A facility is provided in the Data Service server to optionally change the logon credentials for a user when accessing SQL Server.

This task uses the following tools:

CQDSMSSC
An SQL rule that allows Meta discovery on SQL Server databases.

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDEMSSG
An ATH rule that switches credentials when connecting to a SQL Server database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SCQDXSQL(CQDSMSSC) to allow Data Service Studio Meta discovery on SQL Server databases.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter `*` to display all rules, or SQL to display only SQL rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
      Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

      When adding the SYSIN statements that define the alternate credentials for logging in to your Microsoft SQL Server database, as instructed in the JCL, make sure to specify the correct DBTYPE. For SQL Server databases, specify `DBTYPE=MSSQL`.
   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SCQDXATH(CQDEMSSG) to provide the logon credentials to each SQL Server instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter `*` to display all rules, or ATH to display only authentication rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
      Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for Oracle DRDA

Configure Server Event Facility (SEF) rules and set up authentication to provide access to Oracle databases via the Oracle Database Provider for DRDA.

About this task

To complete the configuration for access to Oracle databases via the Oracle Database Provider for DRDA, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to Oracle AS. By default, the Data Service server will attempt to log on to Oracle with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing Oracle.

This task uses the following tools:

CQDSORAC
An SQL rule that allows Meta discovery on Oracle databases.

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDEORAG
An ATH rule that switches credentials when connecting to an Oracle database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SCQDXSQL(CQDSORAC) to allow Data Service Studio Meta discovery on Oracle databases.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or SQL to display only SQL rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.

      Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

      When adding the SYSIN statements that define the alternate credentials for logging in to your Oracle database, as instructed in the JCL, make sure to specify the correct DBTYPE. For Oracle, specify `DBTYPE=ORACLE`.
   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SCQDXATH(CQDEORAG) to provide the logon credentials to each Oracle instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or ATH to display only authentication rules.
   d) Enable the rule by specifying E and pressing Enter.

e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for QMF DRDA Server

Configure Server Event Facility (SEF) rules and set up authentication to provide access to QMF DRDA Server databases.

About this task

To complete the configuration for access to QMF DRDA Server databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to QMF DRDA Server. By default, the Data Service server will attempt to log on to QMF DRDA Server with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing QMF DRDA Server.

This task uses the following tools:

CQDSQMFC
An SQL rule that allows Meta discovery on Oracle databases.

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDEQMFG
An ATH rule that switches credentials when connecting to a QMF DRDA Server database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SCQDXSQL(CQDSQMFC) to allow Data Service Studio Meta discovery on QMF DRDA Server databases.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or SQL to display only SQL rules.
   d) Enable the rule by specifying E and pressing Enter.
   e) Set the rule to Auto-Enable by specifying A and pressing Enter.
      Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

      When adding the SYSIN statements that define the alternate credentials for logging in to your QMF DRDA Server database, as instructed in the JCL, make sure to specify the correct DBTYPE. For QMF DRDA Server databases, specify `DBTYPE=QMFDRDA`.
   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SCQDXATH(CQDEQMFG) to provide the logon credentials to each QMF DRDA Server database. Global variables are used to define alternate authentication credential mapping for the SEF ATH Rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.

c) Enter * to display all rules, or ATH to display only authentication rules.

d) Enable the rule by specifying E and pressing Enter.

e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Controlling display and access for native Db2 subsystems

You can control whether native Db2 database subsystems appear in ISPF and the Data Service Studio and if attempts to connect to native Db2 subsystems are allowed.

About this task

The server parameter **DISABLEATTACH** controls whether native Db2 database subsystems appear in the ISPF and Data Service Studio applications and if attempts to connect to native Db2 subsystems are allowed.

The following table describes the settings for this parameter:

Parameter	Description	Valid values

DISABLEATTACH	Controls whether native Db2 database subsystems appear in the ISPF and Data Service Studio applications and if attempts to connect to native Db2 subsystems are allowed.
	YES
	Only data sources defined as DRDA endpoints appear in the **ISPF DB2 Interface Facility (Database Control)** and the Data Service Studio interface.
	An attempt to connect to a subsystem that does not have a DRDA configuration will be rejected. Trace Browse will show the following message:
	`DB SUBSYSTEM xxxx IS NOT DEFINED`
	For an attempt to connect to a DRDA data source that is disabled, Trace Browse will show the following message:
	`DB SUBSYSTEM xxxx IS NOT OPERATIONAL`
	NO
	(Default) All Db2 subsystems appear in the ISPF and Data Service Studio interfaces.
		YES
		NO

The default setting for server parameter **DISABLEATTACH** is NO; however, the following statement is included in the server configuration file, which changes the setting to YES:

```
“MODIFY PARM NAME(DISABLEATTACH) VALUE(YES)”
```

If this override is omitted from the server configuration file, the setting will default to NO.

To review or update the **DISABLEATTACH** parameter setting, use the following procedure:

Procedure

1. Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Customization Guide*.
2. Review the following statement in your CQDSIN00 member, and update the setting if necessary:

```
“MODIFY PARM NAME(DISABLEATTACH) VALUE(YES)”
```

Configuring access to CA IDMS data

To access CA IDMS data, you must configure the Data Service server started task JCL. You can then optionally verify access to the data.

Data Service server started task JCL changes are required to access CA IDMS software and define default CA IDMS settings.

Restrictions

The following restrictions and considerations apply when accessing CA IDMS data:

- SELECT-only support is provided.
- CA IDMS Logical Record Facility (LRF) is not supported. Virtual views provide many of the same capabilities as LRF and can be used in place of LRF.
- Data access uses CA IDMS network DML only. The CA IDMS SQL product is not required.

Note:

Server configuration parameters control the following behaviors and can be modified if necessary:

- CA IDMS run-unit management, specifically maximum run-units and a timeout value for inactive run-units
- CA IDMS access tracing

Configuring the server started task JCL

Modify the server started task JCL to access CA IDMS and define default CA IDMS settings.

Before you begin

All LOAD library data sets allocated to the Data Service server in the server started task JCL must be APF-authorized.

About this task

Modify the server started task JCL to access CA IDMS and define default IDMS settings.

Procedure

1. Add the CA IDMS load libraries to the STEPLIB, which are required for CA IDMS central version access.
2. Add the SYSCTL DD statement identifying the CA IDMS central version to access.
3. Add the SYSIDMS statement with additional environment parameters. Minimally, this data set should include a CVRETRY=OFF statement to prevent an WTOR message when the CA IDMS central version is not active.
4. Add the CA IDMS system message data set to DCMSG.

Verifying access to CA IDMS data

To verify access to CA IDMS data, you can optionally install a set of maps to the sample database EMPDEMO and run queries using the installed maps.

Before you begin

The CA IDMS sample database EMPDEMO must be installed in the central version you plan to access.

About this task

You can customize and run the provided IVP job CQDISIV1 to install maps to the EMPDEMO database and network schema maps to the SYSTEM database.

The following maps are installed for verification testing using the sample EMPDEMO database:

Table 4. CA IDMS EMPDEMO database maps

Map	Description
EMPSS01_EMPLOYEE	Enables SQL access to EMPLOYEE record.
EMPSS01_OFFICE	Enables SQL access to the OFFICE record.
EMPSS01_DEPARTMENT	Enables SQL access to the DEPARTMENT record.
EMPSS01_OFFICE_EMPLOYEE	Enables SQL access to the OFFICE-EMPLOYEE set for joining the EMPSS01_OFFICE and EMPSS01_EMPLOYEE tables.
EMPSS01_DEPT_EMPLOYEE	Enables SQL access to the DEPT-EMPLOYEE set for joining the EMPSS01_DEPARTMENT and EMPSS01_EMPLOYEE tables.

The network schema maps can be used for verification purposes if the EMPDEMO database is not installed in your central version. These maps access records and sets in the CA IDMS network schema IDMSNTWK, providing SQL access to application metadata. The following table provides a subset of the installed network schema maps that can be used for verification purposes:

Table 5. CA IDMS network schema IDMSNTWK maps

Map	Description
IDMSNWKA_S_010	Enables SQL access to the S-010 network schema record. S-010 records describe application schemas defined to your IDMS central version.
IDMSNWKA_SS_026	Enables SQL access to the SS-026 network schema record. SS-026 records describe application subschemas defined to your IDMS central version.
IDMSNWKA_SSR_032	Enables SQL access to the SSR-032 network schema record. SSR-32 records describe application subschema records defined to your IDMS central version.
IDMSNWKA_S_SS	Enables SQL access to the S-SS set for joining the IDMSNWKA_S_010 and IDMSNWKA_SS_026 tables.
IDMSNWKA_SS_SSR	Enables SQL access to the SS-SSR set for joining the IDMSNWKA_SS_026 and IDMSNWKA_SSR_032 tables.

Procedure

1. Locate the CQDISIV1 member in the *hlq*.SCQDCNTL data set.
2. Modify the JCL according to the instructions provided in the CQDISIV1 member.
3. Submit the job.
4. If the server is active, use the following instructions to refresh maps and make the maps available for use:

a) From the Primary Option Menu, specify option D, **Data Mapping**, and press Enter.

b) From the Data Mapping Facility menu, specify option 3, **Map Refresh**, and press Enter.

Results

CQDISIV1 installs CA IDMS EMPDEMO and network schema maps into the server map data set.

Configuring access to data in IBM IMS databases

To access an IMS database, you need to configure the server started task JCL and the server configuration member.

Before you begin

The server must already be installed.

About this task

IBM® DB2 QMF® Data Service provides seamless, real-time controlled access to IMS database data.

Procedure

To configure and verify access to data in an IMS database, complete the following tasks.

Configuring the server started task JCL

Add `IMS.SDFSRESL` to the server started task JCL.

Before you begin

All LOAD library data sets allocated to the Data Service server in the server started task JCL must be APF-authorized.

About this task

You can omit this task if the IMS resident library (SDFSRESL) module is in the z/OS linklist.

Procedure

Modify the server started task JCL. If the IMS SDFSRESL is not already in the link pack area or linklist, add it to the STEPLIB.

Modifying the server configuration member for DBCTL

Enable the IMS database control (DBCTL) parameters in the server configuration member.

About this task

In order to exploit MapReduce for DBCTL, the server must have information regarding the IMS database to be used by the SQL engine optimizer. This is done by the following command query:

```
SELECT IMSRange('IMS database name')
```

Note: This command should be periodically run if the size of the database changes significantly or the index on the database changes. When this command is issued, data is gathered for each segment in the database for which a virtual table exists and is stored within the server metadata repository.

You can use a batch job to schedule this command to refresh the statistics on a specified schedule. For example:

```
//DSCLIENT  EXEC  PGM=xxxXMAPD,PARM='SSID=VDBS'
//STEPLIB  DD DISP=SHR,DSN=loadlibrary
//OUT      DD SYSOUT=*
//IN       DD *
SELECT IMSRANGE(‘<IMS DBD Name>’);
```

The server configuration member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

1. In the CQDSIN00 member, locate the comment “Enable IMS CCTL/DBCTL support.”
2. Enable the IMS DB parameters by changing the syntax `if DontDoThis` to `if DoThis`, and then set the parameter DBCTL to YES. The following example shows the section in the configuration member to enable:

```
if DoThis then
  do
  "MODIFY PARM NAME(DBCTL)              VALUE(YES)"
  "MODIFY PARM NAME(IMSID)              VALUE(IVP1)"
  "MODIFY PARM NAME(IMSDSNAME)          VALUE(IMSX10.SFDSRESL)"
  "MODIFY PARM NAME(IMSMINTHREADS)      VALUE(5)"
  "MODIFY PARM NAME(IMSMAXTHREADS)      VALUE(10)"
  "MODIFY PARM NAME(IMSNBABUFFERS)      VALUE(0)"
  "MODIFY PARM NAME(IMSFPBUFFERS)       VALUE(0)
  "MODIFY PARM NAME(IMSFPOVERFLOW)      VALUE(0)"
  "MODIFY PARM NAME(TRACEIMSDLIEVENTS)  VALUE(NO)"
```

The following table lists the parameters for configuring support for IMS DB data stores:

Parameter	Description	Valid values
DBCTL	Initialize DBCTL support.	**YES** **NO** (default value)
IMSID	IMS SSID of the DBCTL region.	Four-character name
IMSDSNAME	The name of the data set for the IMS residence library.	Data set name
IMSMINTHREADS	Minimum number of threads.	Numeric value. Default is 5.
IMSMAXTHREADS	Maximum number of threads.	Numeric value. Default is 10.
IMSNBABUFFERS	Total number of NBA buffers.	Numeric value. Default is 0.
IMSFPBUFFERS	Fast path buffers per thread.	Numeric value. Default is 0.
IMSFPOVERFLOW	Fast path overflow buffers.	Numeric value. Default is 0.
TRACEIMSDLIEVENTS	Trace IMS DLI events.	**YES** **NO** (default value)

Modifying the server configuration member for IMS Direct

Enable and configure the IMS Direct parameters in the server configuration member.

About this task

The IMS Direct feature provides map reduce and parallelism support for accessing native IMS files. This support bypasses the requirement of having to use native IMS API calls by reading the IMS database files directly, similar to how an unload utility may work. This method provides a significant improvement in performance and reduced elapsed time in processing analytical type queries.

When an IMS SQL query is run, the SQL engine for the server will determine if the request is best executed using IMS Direct (native file support) or if IMS APIs are required. The determination is based on database and file types supported as well as the size of the database. Virtual tables of the IMS segments are required.

The following types of IMS databases are currently supported by IMS Direct:

- Hierarchical direct access method (HDAM) - VSAM and OSAM
- Hierarchical indexed direct access method (HIDAM) - VSAM and OSAM
- Partitioned HDAM (PHDAM) - VSAM and OSAM
- Partitioned HIDAM (PHIDAM) - VSAM and OSAM
- Fast Path data entry database (DEDB)

When using IMS Direct, there is no locking involved when accessing the data, so updates may not be captured and deleted records may have been captured. Security is managed on the IMS native data set itself when IMS Direct is used. The user ID of the client connection must have the necessary security permissions for reading the IMS database data set(s).

IMS Direct supports access to multiple IMS subsystems and calls to compression exits and Guardium encryption and decryption exits.

Using exits

If you use compression exits or Guardium encryption and decryption exits, you can configure the server to call these exits, providing optimization.

For compression exits, the default mode of operation is to call them in TCB mode with a serialization latch held and a PST address of 0. This can be inefficient since most of the IMS Direct processing takes place in SRB mode on a zIIP. If you know enough about your compression exit, you can optimize performance of the exit by specifying it in either the `IMSDIRCMPTCB`*n*, or `IMSDIRCMPSRB`*n* statements, which are described in the procedure below. All exits are called for INIT and TERM in TCB mode.

- Decompression calls may be made in TCB mode, without serialization by specifying the name in an `IMSDIRCMPTCB`*n* statement. This will allow parallel threads to run without serialization, improving performance.
- Decompression calls may also be made in SRB mode, without serialization, by specifying the name in an `IMSDIRCMPSRB`*n* statement. This will avoid a task switch for each compressed segment, improving performance. Note that the supplied IMS compression DFSCMPX0 exits and DFSKMPX0 will run in SRB mode.

Guardium decryption exits require a PST and PST work area. A dummy PST with a PST work area is passed to these exits when they are specified in an `IMSDIRDECXIT`*n* statement, which is described in the procedure. Guardium decryption exits can run in SRB mode, without serialization.

Procedure

1. Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization.
2. In the CQDSIN00 member, locate the comment “Enable IMS Direct Map Reduce.”
3. Enable the IMS Direct parameters by changing the syntax `if DontDoThis` to `if DoThis`, and then set the parameter IMSDIRECTENABLED to YES. The following example shows the section in the configuration member to enable:

```
if DoThis then
  do
  "MODIFY PARM NAME(IMSDIRECTENABLED)    VALUE(YES)"
  "MODIFY PARM NAME(IMSDIRECTBUFFERSIZE) VALUE(1024)"
  "MODIFY PARM NAME(ACIINTSEGMP256)      VALUE(200)"
  "MODIFY PARM NAME(TRACEIMSDBREFRESH)   VALUE(YES)"
  "MODIFY PARM NAME(TRACEIMSDIRSTATS)    VALUE(YES)"
```

```
   "DEFINE IMSDBINFO",
          .
          .
          .
 end
```

The following table lists the parameters for configuring support for IMS Direct:

Parameter	Description	Valid values
ACIINTSEGMP256	The 256K ACI buffer pool. Required for IMS Direct.	Numeric value. Default is 200.
IMSDIRECTBUFFERSIZE	Specified in KB, and should be greater than the size of the largest complete IMS database record (root + all dependent segments).	Numeric value.
IMSDIRECTENABLED	Enable IMS Direct support.	**YES** **NO** (default value)
TRACEIMSDBREFRESH	Generate trace message when IMS Direct map reduce discovery processing is performed.	**YES** **NO** (default value)
TRACEIMSDIRSTATS	Produce runtime statistics at the end of IMS Direct processing of a data set.	**YES** **NO** (default value)

4. Define your IMS subsystem using the `DEFINE IMSDBINFO` statement. Provide one statement for each IMS subsystem that will be used by IMS Direct.

```
   "DEFINE IMSDBINFO",
          "IMSID(xxxx)",
          "SUFFIX(x)",
          "MODBLKS(your.MODBLKS)",
          "ACBLIB(your.ACBLIB)",
          "DFSRESLB(your.SDFSRESL)",
          "IMSDALIB(your.dynamic.allocation.lib)",
          "RECON1(your.RECON1)",
          "RECON2(your.RECON2)",
          "RECON3(your.RECON3)"
 end
```

The following table lists the parameters used to define the IMS database:

Parameter	Description	Valid values
IMSID	The IMS subsystem identification.	Up to 4-character ID.
SUFFIX	The setting of the SUF= keyword used in the IMS Control Region.	One character. Default value is I.

Parameter	Description	Valid values
ACBLIB	ACBLIB data sets contain the application control blocks (ACBs), which describe IMS applications, and data management blocks (DMBs), which describe databases and the applications that can access them.	your.ACBLIB
DFSRESLB	Load library that contains the major IMS modules.	your.SDFSRESL
IMSDALIB	Dynamic Allocation Library for IMSDBs and RECONs.	your.dynamic.allocation.lib
MODBLKS	Used to support dynamic resource definition. Contains the APPLCTN, DATABASE, RTCODE, and TRANSACT macros.	your.MODBLKS
RECON1	Primary RECONciliation dataset, which holds all of the resource information and event tracking information that is used by IMS.	your.RECON1
RECON2	An active copy of RECON1.	your.RECON2
RECON3	Spare RECON to be used when RECON1 or RECON2 are not useable.	your.RECON3

5. (Optional) Add the following statements to configure additional IMS Direct parameters:

```
"MODIFY PARM NAME(IMSDIRECTCYLBUF) VALUE(3)"
"MODIFY PARM NAME(IMSDIRECTOSAMRECSRD) VALUE(2)"
```

Parameter	Description	Valid values
IMSDIRECTCYLBUF	Specifies the number of cylinders of data to buffer for each file processed in an IMS Direct task.	1-50. Default value is 3.
IMSDIRECTOSAMRECSRD	Specifies the number of records to read in each OSAM I/O operation. For random reads, a large number may lead to unnecessary blocks read. For sequential reads, small numbers may give decreased performance.	1-50. Default value is 2.

6. To call a compression exit, perform one of the following steps as appropriate:

   - If your compression exit must be called in TCB mode but can run properly without serialization, specify your exit name in the following statement:

     ```
     "MODIFY PARM NAME(IMSDIRCMPXITTCBn) VALUE(exitname)"
     ```

     where *n* is a number from 1 to 10 and *exitname* is the name of the compression exit routine.

- If your exit can run properly in SRB mode without serialization, specify your exit name in the following statement:

  ```
  "MODIFY PARM NAME(IMSDIRCMPXITSRBn) VALUE(exitname)"
  ```

  where *n* is a number from 1 to 10 and *exitname* is the name of the compression exit routine.

If neither of these conditions apply, do not specify the name of your compression exit.

Note: Review "Using exits" for more information about configuring calls to compression exits.

Parameter	Description	Valid values
IMSDIRCMPXITTCB*n*	Specifies the name of a compression exit that can be safely called without serialization. Up to 10 exit names can be specified, where *n* is a number from 1 to 10. Since the server runs multiple threads in parallel, this feature provides optimization by eliminating the possible serialization conflicts between threads.	Name of compression exit routine
IMSDIRCMPXITSRB*n*	Specifies the name of a compression exit that can be safely called without serialization and in SRB mode. Up to 10 exit names can be specified, where *n* is a number from 1 to 10. Since multiple exit names can be called without serialization and without switching off the zIIP (SRB mode) into TCB mode (GP processor), this feature provides optimization by eliminating the need to switch tasks for each exit call. The IBM supplied compression exits DFSCMPX0 and DFSKMPX0 will run safely in SRB mode. They can be specified in IMSDIRCMPXITSRB1 and IMSDIRCMPXITSRB2.	Name of compression exit routine

7. To call Guardium encryption and decryption exits, add the following statement:

   ```
   "MODIFY PARM NAME(IMSDIRDECXITSRBn) VALUE(exitname)"
   ```

   where *n* is a number from 1 to 20 and *exitname* is the name of the Guardium exit routine.

   Note: Review "Using exits" for more information about configuring calls to Guardium encryption and decryption exits.

Parameter	Description	Valid values
IMSDIRDECXITSRB*nn*	Specifies the name of the Guardium encryption and decryption exit routine. Up to 20 exit names can be specified, where *nn* is a value from 1 to 20.	Name of Guardium exit routine

Configuring access to IBM MQ

For access to IBM MQ (MQ) data, you must modify the server started task, configure the server configuration member, and set virtual table options.

Data Service provides SQL-only query access to MQ queues using virtual tables. Data in MQ queues is described using COBOL or PLI data descriptions taken from copybooks or programs.

IBM MQ for z/OS Versions 7.5 and newer are supported.

Note: Server configuration parameters control MQ tracing and can be modified if necessary.

Configuring the server started task JCL

Modify the server started task JCL to access IBM MQ data.

Before you begin

All data sets that you add to the server started task JCL STEPLIB must be APF-authorized.

About this task

Modify the server started task JCL to access IBM MQ data. You can skip this task if the IBM MQ load module is in the z/OS linklist or link pack area.

Procedure

Add the IBM MQ load library to the server started task JCL STEPLIB.

Modifying the server configuration member for IBM MQ

To enable support for MQ data, you must update your Data Service server configuration file.

About this task

To be able to access MQ data in virtual tables, enable the feature in the server configuration file, as described in the following procedure.

Procedure

1. Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization.
2. Add the following statement to your CQDSIN00 member:

   The following table describes this parameter:

Parameter	Description	Valid values
MQACTIVE	Initialize IBM MQ support. This parameter must be set to YES to access MQ queues.	**YES** **NO** (default value)

Configuring virtual table rules for IBM MQ

Configure Server Event Facility (SEF) rules to support IBM MQ data.

About this task

You can configure VTB rule options to control the MQ data access feature. These options control inclusion of the MQ message descriptor meta data fields in the virtual tables, how to handle truncated messages, and whether to perform destructive reads. Sample VTB rule CQDMDLMQ documents these settings.

When accessing MQ data with sample rule CQDMDLMQ (or equivalent options) enabled, tables prefixed with MDLMQ_* are filtered, and the map name is extracted by removing the MDLMQ_ prefix. For example, the following query will execute the rule and query virtual table MQ_CSQ7_TRADE:

```
SELECT * FROM MDLMQ_MQ_CSQ7_TRADE
```

Use the following procedure to configure the sample rule CQDMDLMQ.

Note: Sample rule CQDMDLMQ is intended to be used as a model and may require customization. When customizing this rule, additional logic may need to be added if different VTB variable settings are required for different MQ queues.

Procedure

1. Customize the server configuration member (CQDSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

2. Access the VTB rules, as follows:
   a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.
   b) Specify option 2, **SEF Rule Management**.
   c) Enter VTB for **Display Only the Ruleset Named**.
3. Customize the CQDMDLMQ rule, as follows:
   a) Specify S next to CQDMDLMQ to edit the rule.
   b) Update the rule options as needed. The following table describes the VTB rule options that support MQ data access.

VTB variable	Description	Valid values
vtb.optbmqdg	Delete messages during retrieval. When set to 1, SQL queries will remove messages from the queue if ALL messages in the queue are successfully retrieved by the server. Retrieval of MQ messages will use non-browse (destructive) MQGET calls with syncpoint control. Once all messages are delivered to the server, they will be deleted from the queue. If a failure occurs before all messages are retrieved, an MQBACK call will be issued to restore messages to the queue that have been retrieved so far. Note that an MQCMIT will be issued and messages deleted if the IBM MQ syncpoint limit is reached. A failure after MQCMIT will not be able to restore messages as they have been permanently deleted.	0 (Default) 1

VTB variable	Description	Valid values
vtb.optbmqim	When set to 1 for an MQ virtual table, the MQ Series Message Descriptor (MQMD) meta data fields will be added to the virtual table as columns and returned with each result row. These columns are prefixed with the value MQMD_.	0 (Default) 1
vtb.optbmqtc	By default, a truncation error reading an IBM MQ message will result in a query failure. When set to 1, MQ Series access ignores truncated message warnings and returns data received.	0 (Default) 1

c) Save your changes and exit the editor.

4. Enable the rule by specifying E next to CQDMDLMQ and pressing Enter.
5. Set the rule to Auto-enable by specifying A next to CQDMDLMQ and pressing Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring access to VSAM

No modifications are required to configure the SQL interface for native VSAM. However, you should verify that the server has access to VSAM. Optionally, you can control the data buffer (BUFND) and the index buffer (BUFNI) values for VSAM files either globally or for individual requests.

Verifying access to native VSAM

Verify native VSAM data access by creating a sample VSAM file and a corresponding virtual table and running a query that accesses the VSAM data.

Procedure

1. Create the sample VSAM file on the mainframe that hosts the Data Service server.

   Run the `CQDGNSTF` member in the *hlq*.SCQDCNTL data set to allocate and load the sample VSAM file.

   The job should complete with a condition code of 0.
2. Create the `staffvs` virtual table, and run a query that returns a result set.

   Run the `CQDIVVS1` member in the *hlq*.SCQDCNTL data set to perform a batch extract of the sample VSAM file listing and create a virtual table that is used to format the result set that is returned from the VSAM file.

   The job should complete with a condition code of 0.
3. Verify that the SQL results contained in the `CQDIVVS1` member are valid.

Modifying the data and index buffer values for VSAM files

You can change the data and index buffer values for VSAM files.

About this task

You can control the data buffer (BUFND) and the index buffer (BUFNI) values for VSAM files either globally or for individual requests, as follows:

- To change the values globally, you must add the required parameters to your Data Service server configuration file. The following table lists these parameters:

Parameter	Description	Valid values
SQLENGVSAMDATABUFF	Specifies the number of data buffers for VSAM files. Default: 20	Numeric value.
SQLENGVSAMINDEXBUFF	Specifies the number of index buffer for VSAM files. Default: 30	Numeric value.

- To change the values for individual requests, you can use virtual table (VTB) rules. Sample VTB rules CQDBUFND and CQDBUFNI are provided.

  To override your index buffer or data buffer values, you must enable the respective rule and use the appropriate BUF prefix for table names in your SQL statement, as follows.

  – **To override the data buffer (BUFND) value:**

    Use sample rule CQDBUFND. The CQDBUFND rule is invoked every time a table with the prefix BUFND_ is found in the SQL statement. The following format is expected:

    ```
    BUFND_nn_virtualtablename
    ```

    Where:

    - *nn* is the number of data buffers (BUFND) for the VSAM data sets
    - *virtualtablename* is the name of the virtual table

    For example:

    ```
    SELECT * from BUFND_30_STAFF_VSAM ;
    ```

    The following message is displayed in the Server Trace:

    ```
    CQD1000I VTB.OPTBVSND set to 30
    ```

  – **To override the index buffer (BUFNI) value:**

    Use sample rule CQDBUFNI. The CQDBUFNI rule is invoked every time a table with the prefix BUFNI_ is found in the SQL statement. The following format is expected:

    ```
    BUFNI_nn_virtualtablename
    ```

    Where:

    - *nn* is the number of index buffers (BUFNI) for the VSAM data sets
    - *virtualtablename* is the name of the virtual table

    For example:

    ```
    SELECT * from BUFNI_30_STAFF_VSAM ;
    ```

    The following message is displayed in the Server Trace:

    ```
    CQD1000I VTB.OPTBVSNI set to 30
    ```

Procedure

1. To change the values globally, perform the following steps:

   a) Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Customization Guide*.

   b) Add the following statements to your CQDSIN00 member:

```
"MODIFY PARM NAME(SQLENGVSAMDATABUFF) VALUE(20)"
"MODIFY PARM NAME(SQLENGVSAMINDEXBUFF) VALUE(30)"
```

2. To change the values for individual requests, perform the following steps:

   a) Customize the server configuration member (CQDSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

   b) Access the VTB rules, as follows:

      i) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.

      ii) Specify option 2, **SEF Rule Management**.

      iii) Enter VTB for **Display Only the Ruleset Named**.

   c) Enable each rule as follows:

   - Specify E next to CQDBUFND and press Enter.
   - Specify E next to CQDBUFNI and press Enter.

   d) Set each rule to Auto-enable as follows:

   - Specify A next to CQDBUFND and press Enter.
   - Specify A next to CQDBUFNI and press Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

   e) Use the appropriate BUF prefix for table names in your SQL statement.

Configuring access to sequential files

No modifications are needed to configure the SQL interface to access sequential files. However, you should verify access to sequential files. Optionally, you can specify the number of tracks to read ahead when reading sequential data sets for individual requests.

Reading ahead tracks for sequential file access

You can use a Server Event Facility (SEF) rule to specify the number of tracks to read ahead (MULTACC) when reading sequential data sets for individual requests.

About this task

Using a virtual table (VTB) rule, you can specify the number of tracks to read ahead (the MULTACC parameter value) for MapReduce sequential file access for individual requests. This support overrides the value in the server parameter **ACIMAPREDUCETRACKS (NUMBER OF MAP REDUCE TRACKS TO READ)** for individual requests. Sample VTB rule CQDMLTAC is provided.

To override the MULTACC value, you must enable the CQDMLTAC rule and use the MACC_*nn*_ prefix for table names in your SQL statement.

The CQDMLTAC rule is invoked every time a table with the prefix MACC_*nn*_ is found in the SQL statement. The following format is expected:

```
MACC_nn_virtualtablename
```

Where:

- *nn* is the number of tracks to read ahead (the MULTACC value) when reading sequential data sets
- *virtualtablename* is the name of the virtual table

For example:

```
SELECT * from MACC_15_STAFF_SSEQ ;
```

The following message is displayed in the Server Trace:

```
CQD1000I VTB.OPTBMACC set to 15
```

Use the following procedure to set up the rule.

Procedure

1. Customize the server configuration member (CQDSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

2. Access the VTB rules, as follows:
   a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.
   b) Specify option 2, **SEF Rule Management**.
   c) Enter VTB for **Display Only the Ruleset Named**.
3. Enable the rule by specifying E next to CQDMLTAC and pressing Enter.
4. Set the rule to Auto-enable by specifying A next to CQDMLTAC and pressing Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring access to zFS files

The Data Service server is already configured to support zFS files. No modifications are needed to configure access to zFS files.

Configuring access to SMF data for IT Operational Analytics

IT Operational Analytics (ITOA) allows you to retrieve, analyze, and report data for IT operations. System information can be logged using the IBM System Management Facility (SMF) and the native Data Service server logging feature. Logging allows you to collect various system and operations-related information.

Before you begin

Verify that the following IBM APARs have been applied:

- APAR OA49263. This APAR provides real-time SMF support and is a requirement for the configuration of real-time SMF data access. (The closed date for this APAR is 2016-08-31.)
- APAR OA48933. This APAR is required to address accessing log streams. SMF log stream configuration is required for in-memory resource support. (The closed date for this APAR 2015-11-24.)

About this task

Virtual tables for SMF are provided in the *hlq*.SCQDSMAP data set.

The following options are available to access the SMF data:

- Reading data from SMF data sets - SMF information is recorded in MAN*x* data sets. When a data set gets full, the data is processed via IFASMFDP. When defining global variables for accessing SMF data in data sets, the output of IFASMFDP is used.
- Reading data from log streams - SMF information is recorded in multiple log streams and data can be read directly from the log streams. Log stream recording is determined by the data set name beginning with IFASMF that is used in the VTB rule for SMF.
- Reading SMF data from in-memory (real-time) - SMF information is read directly from the system buffer. SMF information is read in real time. There are two interfaces to real-time SMF data, which connect to the in-memory resource at different times, as follows:

- At product initialization. This interface connects to the in-memory resource at product initialization and continuously reads from the API to maintain a buffer of recent SMF activity. This buffer can be queried, and its contents will be returned, followed by an end-of-data indication.
- At the time of the request. This interface connects to the in-memory resource at the time of the request and streams the SMF data to the requester in real time. A request to this named stream is considered non-ending, and data will continue to flow until the request is canceled or the server is stopped.

When defining the global variables for SMF, the data set can be either a log stream or a SMF dump data set from IFASMFDP. The log stream data set is recommended for access to near real-time data.

To configure access to IT Operational Analytics data, see the following topics:

- "Configuring access to System Management Facility (SMF) files" on page 54
- "Configuring access to SYSLOG files" on page 57
- "Configuring access to OPERLOG files" on page 58

Configuring access to System Management Facility (SMF) files

To configure access to System Management Facility (SMF) files, you need to configure the server started task JCL, the server configuration member, and the server virtual table member. To enable reading SMF data real-time using log streams, you must have the **SMFPRMxx** member in the system PARMLIB data set configured to use both log streams and in-memory resources. Follow the steps in this section to use SMF GDG data set names, or to use dynamic data set names.

About this task

SMF data set names are dynamic in local environments and require SEF rules enablement and optionally Global Variables set to specific values to provide data set names to the virtual tables and views when using SMF data set or log stream configurations.

You can choose either GDG data set name support or dynamic data set name support, or both, to quickly access your SMF data. These two options are provided for your convenience to help you start accessing your SMF data. Custom rules may need to be developed to use your local naming convention to access your SMF files.

Procedure

1. Configure the server started task JCL by concatenating the *hlq*.SCQDSMAP data set to the CQDMAPP DD statement to add all maps for SMF.
2. Customize the server configuration member.

   To enable virtual table rule events, configure the SEFVTBEVENTS parameter in the CQDSIN00 member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

   Verify the VTB ruleset name:

   ```
   "DEFINE RULESET NAME(VTB)"
          "RULETYPE(VTB)"
          "DSNAME('"||SHLQ2||".SCQDXVTB')"
   ```

   If there were any changes to CQDSIN00, recycle the server started task.
3. To enable real-time access to SMF data, add the following statements to the CQDSIN00 member after the GLOBAL PRODUCT OPTIONS statement.

   ```
   IF DoThis
     THEN DO
       "DEFINE SMF NAME(IFASMF.INMEM)",
       "STREAM(IFASMF.INMEM.STREAM)",
       "BUFSIZE(500)",
   ```

```
    "TIME(0)"
END
```

Note: You must have the **SMFPRMxx** member in the system PARMLIB data set configured to use log streams and in-memory resources.

Parameter	Description	Valid values
NAME	Specifies the name of the in-memory resource. This value must match the name of a resource defined to SMF with the **INMEM** parameter. If this parameter is included, the in-memory API will be read continuously and a buffer of the most recent records will be maintained. Either this parameter or the **STREAM** parameter, or both, must be specified.	This parameter must contain the name of an in-memory resource defined to SMF with the INMEM statement. The format of the name is defined by SMF configuration, which is 1-26 characters and must begin with IFASMF.
STREAM	Specifies the name of the streaming in-memory feature. If this name is specified on a SELECT statement, a dynamic connection will be made to the SMF in-memory API and records will be streamed to the caller in real time. Either this parameter or the **NAME** parameter, or both, must be specified.	If a **NAME** parameter is also supplied, the in-memory resource named in that parameter will be connected to and the value of this parameter can be any name, 1-26 characters. If the **NAME** parameter is not supplied, this parameter must contain the name of an in-memory resource defined to SMF with the **INMEM** parameter. If both **NAME** and **STREAM** are provided, the names must be different.
BUFSIZE	Indicates how much SMF data (megabytes) will be retained in memory for queries. If the buffer fills up, the oldest data will be discarded. In parallel, SMF is recording these records to a log stream. This parameter applies to the resource named in the **NAME** parameter.	1-10,000
TIME	Indicates how long (in minutes) to keep SMF data in memory. Older data will be discarded. Specifying 0 indicates no time limit and data will be retained until the buffer fills up. This parameter applies to the resource named in the **NAME** parameter.	0-1440

4. To use SMF data in compressed log streams, add the following statement to the CQDSIN00 member:

```
"MODIFY PARM NAME(ZEDCCOMPRESSION)      VALUE(YES)"
```

Note: You must have the **SMFPRMxx** member in the system PARMLIB data set configured to use compressed log streams, and the zEDC Express hardware feature must be installed.

5. Enable reading SMF data from GDG data sets and access to SMF data using dynamic data set names by enabling Server Event Facility rule CQDSMFT1 in the VTB ruleset. You can select from a GDG data set, any SMF dump data set, a log stream data set, or the in-memory stream. Activate your options by customizing the rule.

   a) Use the following steps to enable rule CQDSMFT1 in the VTB ruleset:

      i) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.

      ii) Specify option 2, **SEF Rule Management**.

      iii) Enter VTB for **Display Only the Ruleset Named**.

      iv) Enable the rule by specifying E and pressing Enter.

      v) Set the rule to Auto-enable by specifying A and pressing Enter.

      Setting the rule to Auto-enable activates the rule automatically when the server is re-started.

   b) Configure the access method using one or more of the following methods:

      - Review the information in the rule for the instructions on setting Global Variables that will be used by the rule. Navigate one screen back on the ISPF panel, or start over by going to option E, **Rules Mgmt.**, and then option 1, **Global Variables**. In the Global Variables display, perform the following steps:

        i) Change Global Prefix to `GLOBAL2`.

        ii) Select SMFTBL2 by entering S next to the SMFTBL2 data set.

        iii) Configure the SMF data access option. DEFAULT should have corresponding SMF dump data set names if used. This option can be used to specify the source SMF, such as GDGBASE, INMEM, and LOGSTREAM.

        Note:

        VTB rules and global variables may be used to reference a GDG data set, any SMF dump data set, a log stream data set, or the in-memory stream. For example:

        ```
        GLOBAL2.SMFGBL2.YESTERDAY = "YOUR.DATASET.SMFDUMP(-1)"
        GLOBAL2.SMFGBL2.M2 = "YOUR.DATASET.SMFDUMP(-2)"
        GLOBAL2.SMFGBL2.M3 = "YOUR.DATASET.SMFDUMP(-3)"
        GLOBAL2.SMFGBL2.M4 = "YOUR.DATASET.SMFDUMP(-4)"
        GLOBAL2.SMFGBL2.M5 = "YOUR.DATASET.SMFDUMP(-5)"
        GLOBAL2.SMFGBL2.IM = "IFASMF.INMEM"
        GLOBAL2.SMFGBL2.IM2 = "IFASMF.INMEM2"
        GLOBAL2.SMFGBL2.LOG = "LOGSTREAM.dataset.name"
        ```

      - Pass a dynamic data set name for SMF tables using the following format for the table name in the SQL statement:

        ```
        TableMapName__DataSetName
        ```

        Where `DataSetName` is prefixed by two underscores (__) and the periods in the data set name are replaced with single underscores (_).

        For example, `SELECT * FROM SMF_01400__DATA_SET_NAME` would translate into an SQL query of `SELECT * FROM SMF_14000` and access the data set `DATA.SET.NAME`.

      - Pass a dynamic data set name for SMF virtual views using the following format for the virtual view name in the SQL statement:

        ```
        ViewMapName__DataSetName
        ```

        Where `DataSetName` is prefixed by two underscores (__) and the periods in the data set name are replaced with single underscores (_).

For example, SELECT * FROM SMFV_01400__DATA_SET_NAME would translate into an SQL query of SELECT * FROM SMFV_01400 and access the data set DATA.SET.NAME.

Configuring access to SYSLOG files

The Data Service server is enabled to support access to SYSLOG files. Use these steps to enable the rule.

About this task

Virtual table rules are provided that support the processing of SYSLOG files and vary based on the type of file name used for your SYSLOG data sets. Each of the rules for SYSLOG processing requires that the table names in the SQL begin with SYSLOG. The following rules are provided:

CQDSYSLG

This rule uses a global variable to specify the name of the data set to use for the SYSLOG data.

CQDSYSL2

This rule supports the use of generation data group (GDG) data set names. One of the following formats is expected:

-
  ```
  SYSLOG_GDG_nnnn
  ```

  Where *nnnn* is a relative GDG number (between 0 and 9999) that is appended to the GDG base name value that is obtained from the GLOBAL2.SYSLOG.GDGBASE variable. For example, if the table name as specified in the SQL statement is SYSLOG_GDG_1, then the data set name returned by this rule is HLQ.SYSLOG(-1), depending on the value in GLOBAL2.SYSLOG.GDGBASE.

-
  ```
  SYSLOG_DSN_suffix
  ```

  Where *suffix* is used as the last part of a global variable of the form GLOBAL2.SYSLOG.*suffix* in order to look up the name of the data set to be used. If this variable does not exist, the data set name specified in GLOBAL2.SYSLOG.DEFAULT is used to read the SYSLOG records.

By using global variables, you do not need to modify the code in the rule. The following are some examples of global variables that can be set up to be used in conjunction with this rule:

```
Global Prefix: GLOBAL2.SYSLOG
S Subnode Name   Nodes         Subnode Value
- -------------- ----- ---------------------------
  GDGBASE            0 HLQ.SYSLOG
  DEFAULT            0 HLQ.SYSLOG(0)
  TODAY              0 HLQ.SYSLOG(0)
  YESTERDAY          0 HLQ.SYSLOG(-1)
```

CQDSYSL3

This rule lets you dynamically specify in your SQL the name of the data set to use when processing SYSLOG files. In the SQL, the table name must begin with the prefix SYSLOG; the rest of the table name is used by the rule to determine the actual data set name to use for processing the SYSLOG records.

The following format is expected:

```
SYSLOG__DataSetName
```

Where *DataSetName* is preceded by two underscores (__) and the periods in the data set name are replaced with single underscores (_). For example, SELECT * FROM SYSLOG__DATA_SET_NAME would translate into an SQL query of SELECT * FROM SYSLOG and access the data set DATA.SET.NAME.

To use one of the rules, you must enable the rule and use the prefix SYSLOG for table names in your SQL statement. The enabled rules are invoked every time a table with the prefix SYSLOG is found in the SQL statement.

Use the following procedure to set up the rules.

Procedure

1. Access the VTB rules, as follows:
   a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.
   b) Specify option 2, **SEF Rule Management**.
   c) Enter VTB for **Display Only the Ruleset Named**.
2. For CQDSYSLG, customize the rule, as follows:
   a) Specify S next to CQDSYSLG to edit the rule.
   b) Customize the rule with the SYSLOG data set name.
   c) Save your changes and exit the editor.

   Note: For CQDSYSL2 and CQDSYSL3, no customization of the rule is needed.
3. Enable each rule by specifying E next to the member name and pressing Enter.
4. Set each rule to Auto-enable by specifying A next to the member name and pressing Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.
5. If global variables are needed, set up the SYSLOG global variable.

Configuring access to OPERLOG files

No modifications are needed to configure the Data Service server to access OPERLOG data; however, OPERLOG must be active in a system logger log stream.

About this task

Use the following procedure to verify that OPERLOG is active in a system logger log stream.

Procedure

To display the active medium where messages are recorded, enter the following command:

```
D C,HC
```

The following results are expected:

```
CNZ4100I 15.19.16 CONSOLE DISPLAY 056
 CONSOLES MATCHING COMMAND: D C,HC
 MSG:CURR=0    LIM=9000 RPLY:CURR=0    LIM=9999  SYS=P02      PFK=00
 HARDCOPY  LOG=(SYSLOG,OPERLOG)  CMDLEVEL=CMDS
       ROUT=(ALL)
 LOG BUFFERS IN USE: 0       LOG BUFFER LIMIT: 9999
```

Configuring access to ADDI

To use IBM Application Discovery and Delivery Intelligence (ADDI) information for creating virtual maps that access VSAM and sequential data, you must configure the server for ADDI access.

System requirements

The following system requirements apply:

- IBM Application Discovery Suite Version 5.0 or newer
- Microsoft Host Integration Server (HIS) 2016 or higher. The SYSIBM views that are part of the Microsoft HIS Software Development Kit must be installed as part of the HIS installation.
- Microsoft SQL Server 2012 Enterprise or Express or higher

Restrictions

The following restrictions and considerations apply when using ADDI to access VSAM and sequential data sets:

- Virtual table creation is restricted to data sets in the ADDI project that are processed by COBOL programs using JCL. Data sets accessed using CICS as well as other databases (such as IMS, CA IDMS, or Adabas) are not supported.
- Virtual table mapping is only supported through the Data Service Studio. No batch utilities or ISPF interfaces are provided to map tables.

Configuration steps

The following configuration steps are required to use ADDI to access VSAM and sequential data:

1. Install virtual tables. See "Installing virtual tables and virtual target maps for ADDI access" on page 59.
2. Define ADDI project in the server configuration member. See "Modifying the configuration member for ADDI access" on page 60.
3. Activate virtual table rules. See "Configuring virtual table rules for ADDI" on page 63.
4. Define credentials for target database(s). See "Configuring authentication for ADDI" on page 64.

Installing virtual tables and virtual target maps for ADDI access

Install virtual tables and virtual target maps for IBM Application Discovery and Delivery Intelligence (ADDI) access.

About this task

The Data Service Studio reads the ADDI project using virtual tables and views installed as part of server set up. The following maps are distributed in XMIT format in the SCQDSAMP member CQDIAMPD:

ZIADTSPR
Virtual target system TSIAD_PROJECT1 for external subsystem named IAD1.

ZIADT001-ZIADT021
Virtual tables that map tables in the ADDI project. Each virtual table uses the name of the corresponding ADDI project table with the added prefix `IAD_`. For example, SQL Server table `dbo.Variables` has a virtual table name of `IAD_VARIABLES`.

ZIADV001-ZIADV002
Virtual views on the IAD_ virtual tables used by the Data Service Studio to read ADDI data. These views are all prefixed with `IADV_` (for example, `IADV_DATASETS`). All data access from the studio is performed using virtual views.

These maps are not installed by default. Use the following procedure to install these maps.

Procedure

1. Locate the CQDIAMPS member in the *hlq*.SCQDCNTL data set.
2. Modify the JCL according to the instructions provided in the CQDIAMPS member.
3. Submit the job.

   The virtual tables and virtual target maps are installed.

Modifying the configuration member for ADDI access

Enable and configure the parameters for IBM Application Discovery and Delivery Intelligence (ADDI) in the server configuration member.

About this task

The server configuration member contains a sample DATABASE definition that defines the first ADDI project. The initial definition is named IAD1 and is disabled.

When enabling the database definition for the first ADDI project, the LOCATION and IPADDR parameters must be set to the correct project name and IP address of the Microsoft HIS DRDA Provider Service for SQL Server. The LOCATION provides the name of the SQL Server project, and IPADDR(...) PORT(...) provide the TCP/IP information for the HIS DRDA Service. DOMAIN(...) can be used instead of IPADDR to provide the DNS of the HIS DRDA Service. The subsystem NAME(IAD1) should not be changed because a target subsystem map is configured to use this name for the virtual tables accessing the ADDI project.

For multiple ADDI projects, see "Adding an ADDI project" on page 62.

The server configuration member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

1. In the CQDSIN00 member, locate the comment "Sample IBM Application Discovery configuration".
2. Enable the ADDI parameters by changing the syntax `if DontDoThis` to `if DoThis`. The following example shows the section in the configuration member to enable:

```
/*---------------------------------------------------------------*/
/* Sample IBM Application Discovery configuration using DRDA to  */
/* communicate with a Microsoft SQLServer database.              */
/*---------------------------------------------------------------*/
if DoThis then do
"DEFINE DATABASE TYPE(MSSQL)"                  ,
                 "NAME(IAD1)"                  ,
                 "LOCATION(EZ_Project1)"       ,
                 "DDFSTATUS(ENABLE)"           ,
                 "SECMEC(USRIDPWD)"            ,
                 "IPADDR(::FFFF:0.0.0.0)"      ,
                 "PORT(446)"                   ,
                 "CCSID(37)"                   ,
                 "IDLETIME(0)"
end
```

The following table lists the parameters for configuring support for ADDI:

Parameter	Description	Valid values
TYPE	Database type. Because ADDI stores information in Microsoft SQL Server, this value must be MSSQL.	MSSQL

Parameter	Description	Valid values
NAME	The database name as known to the server. The first definition must be `IAD1` because the target system map names this as the subsystem to access for ADDI. For additional ADDI projects, subsystems can have any name since you must also create a virtual target system to point to it; however, it recommended that the name start with `IAD`. (*Required*)	A valid value consists of 1 - 4 characters. For example, `IAD1`.
LOCATION	Name of the database for the ADDI project. The LOCATION parameter must be set to the correct database name of the target MSSQL server. (*Required*)	A valid value is a string 1 - 16 characters.
DDFSTATUS	The DDF activation status (*Required*)	**ENABLE** Make this DDF definition active within Data Service server. DDFSTATUS should always be ENABLE for TYPE(MSSQL). **DISABLE** DDF endpoint is not used. This value disables the MSSQL database. This value should only be used if the database is off-line or otherwise not available for access.
SECMEC	Security mechanism. The DRDA security mechanism for authentication with the HIS DRDA Service for SQL Server. The SECMEC setting for TYPE(MSSQL) must match the HIS DRDA Service configuration.	**USRIDPWD** User ID and password **USRIDONL** User ID only **USRENCPWD** Encrypt the password only **EUSRIDPWD** Encrypt the user ID and password

Parameter	Description	Valid values
IPADDR	Specify the IPV4 or IVP6 address of the target MSSQL server. Use DOMAIN instead of IPADDR to supply the DNS of the target HIS DRDA Server for SQL Server. Use DOMAIN if the IPADDR or the HIS DRDA Service Provider can change. Either DOMAIN or IPADDR is required, but not both.	A valid IPV4 or IVP6 address set to the correct remote IP address for the system running Microsoft SQL Server.
DOMAIN	The part of a network address that identifies it as belonging to a particular domain. Use DOMAIN instead of IPADDR to supply the DNS of the target HIS DRDA Server for SQL Server. Use DOMAIN if the IPADDR or the HIS DRDA Service Provider can change. Either DOMAIN or IPADDR is required, but not both.	No default value.
PORT	The TCP/IP port defined for Microsoft HIS DRDA Service Provider. For TYPE(MSSQL), the standard HIS default is 446. (*Required*)	A valid 1-5 numeric string.
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier). (*Required*)	Refer to the Microsoft SQL Server documentation for a list of valid CCSIDs. Refer to the ISV documentation on HIS DRDA Service to SQL Server. For USA, this value is `037`.
IDLETIME	This setting is not used for TYPE(MSSQL).	0

Adding an ADDI project

Perform required configuration steps to add an ADDI project.

About this task

For multiple ADDI projects, you must perform configuration steps to define each additional ADDI project. The following requirements apply when maintaining multiple ADDI projects:

- For the first instance of an ADDI project:
  - The database name in the must be IAD1.

- The target system for the name IAD1 is automatically installed with the ADDI maps, as described in "Installing virtual tables and virtual target maps for ADDI access" on page 59.

- For subsequent ADDI projects:
  - It is recommended that the database name start with IAD.
  - The target system must start with TSIAD.

Perform the following procedure for each additional ADDI project.

Procedure

1. Repeat the database definition in the configuration member and make the following changes:
   a) Change the NAME value to a unique name (for example, IAD2).
   b) Change the LOCATION value to match the Microsoft SQL Server project name containing the ADDI project you need to access.

   For information about the database definition parameters, see "Modifying the configuration member for ADDI access" on page 60.
2. Define a new virtual target system using the studio. The name of the virtual target system must start with TSIAD. This can be done in the Data Service Studio by selecting the **Create Virtual Target System** in the **Server** tab under the **SQL** > **Target Systems** > **DBMS** node of the tree. The connection value in each definition must match the NAME value defined in the DATABASE definition in the configuration member.
3. If required, create authentication information using the CQDDRATH batch utility.

Configuring virtual table rules for ADDI

Configure Server Event Facility (SEF) rules to support multiple projects using common virtual table and view definitions.

About this task

To support multiple projects using common virtual table and view definitions, VTB rules CQDIADTB and CQDIADVW provide support to process tables starting with `IAD_` and views starting with `IADV_`.

CQDIADTB
This table rule looks at the base view of a query for double underscores "__" and uses the data after the underscores to update the target subsystem for the query.

CQDIADVW
This view rule looks for the double underscores and removes them from the view name to process.

With the rules activated, the Data Service Studio can suffix the view names with __SSID for all calls and process multiple ADDI projects using a single set of maps.

These rules must be activated regardless of the number of ADDI projects to be enabled.

Use the following procedure to set up these rules.

Procedure

Use the following steps to enable rules CQDIADTB and CQDIADVW in the VTB ruleset:

a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.
b) Specify option 2, **SEF Rule Management**.
c) Enter VTB for **Display Only the Ruleset Named**.
d) Enable the rules by specifying E and pressing Enter.
e) Set the rules to Auto-enable by specifying A and pressing Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring authentication for ADDI

Configure authentication for communicating with the IBM Application Discovery and Delivery Intelligence (ADDI) project.

About this task

It is common for data centers to assign different user IDs for access to z/OS and for access to SQL Server. By default, the server will attempt to log on to SQL Server with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing SQL Server.

When communicating between the Data Service server and the ADDI project, you must define what credentials to use in MSSQL connections if z/OS users are not defined as users to SQL Server. To accomplish this, the following tools are provided:

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. Use this utility to define alternate logon information for the Data Service server started task and z/OS users. This utility places SQL Server authentication information in GLOBALU system variables for connecting to ADDI projects. You can also use this utility to list existing credential information.

CQDEMSSG
An ATH rule that swaps z/OS user information with SQL Server authentication information defined using the CQDDRATH utility. This rule uses AES encrypted passwords stored as GLOBALU system variables.

You can use any of the following options for authentication:

- Use z/OS IDs for authentication
- Add a global default user definition using sample job CQDDRATH and enable ATH rule CQDEMSSG
- Add authentication information for specific mainframe users using sample job CQDDRATH and enable ATH rule CQDEMSSG

Network administrators may need to open ports for DRDA communication between the z/OS host and the Microsoft SQL Server machine(s) hosting ADDI projects. The default port for Microsoft SQL Server access is 446.

If z/OS user IDs are not defined to Microsoft SQL Server, use the following procedure to define alternate authentication information for the started task and z/OS users requiring access to this feature:

Procedure

1. Use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

   When adding the SYSIN statements that define the alternate credentials for logging in to your ADDI project, as instructed in the JCL, make sure to specify the correct DBTYPE. For ADDI projects, specify `DBTYPE=MSSQL`.

   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
2. Auto-enable the SEF ATH rule SCQDXATH(CQDEMSSG) to switch credentials when connecting to ADDI using DRDA. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.

c) Enter * to display all rules, or ATH to display only authentication rules.

d) Set Auto-Enable for the CQDEMSSG rule member by entering A and pressing Enter.

Configuring access to RAA

To use IBM Rational Asset Analyzer (RAA) information for creating virtual maps that access VSAM and sequential data, you must configure the server for RAA access.

System requirements

The following system requirement applies:

- IBM Rational Asset Analyzer for System z 6.1 PID5655-W57

Restrictions

The following restrictions and considerations apply when using RAA to access VSAM and sequential data sets:

- Virtual table creation is restricted to data sets in the RAA database that are processed by COBOL programs using JCL. Data sets accessed using CICS as well as other databases (such as IMS, CA IDMS, or Adabas) are not supported.
- Virtual table mapping is only supported through the Data Service Studio. No batch utilities or ISPF interfaces are provided to map tables.

Configuration steps

The following configuration steps are required to use RAA to access VSAM and sequential data:

1. Install virtual tables. See “Installing virtual tables and virtual target maps for RAA access” on page 65.
2. Define RAA database in the server configuration member. “Modifying the configuration member for RAA access” on page 66.
3. Activate virtual table rules. See “Configuring virtual table rules for RAA” on page 69.
4. Define credentials for target database(s). See “Configuring authentication for RAA” on page 69.

Installing virtual tables and virtual target maps for RAA access

Install virtual tables and virtual target maps for IBM Rational Asset Analyzer (RAA) access.

About this task

The Data Service Studio reads the RAA database using virtual tables and views installed as part of server set up. The following maps are distributed in XMIT format in the SCQDSAMP member CQDRAMPD.

ZRAATSPR
Virtual target system TSRAA_PROJECT1 for external subsystem named RAA1.

ZRAAT001-ZRAAT010
Virtual tables mapping tables in the RAA database. All tables use the same name as the corresponding RAA database table with a prefix of RAA_ (for example, “DMH”.”DMH_DATA_RECORD” in DB2 has a virtual table name of RAA_DATA_RECORD).

ZRAAV001-ZRAAV003
Virtual views on the RAA_ virtual tables used by the Data Service Studio to read RAA data. These views are all prefixed with RAAV_ (for example, RAAV_DATASETS). All data access from the studio is performed using virtual views.

These maps are not installed by default. Use the following procedure to install these maps.

Procedure

1. Locate the CQDRAMPS member in the *hlq*.SCQDCNTL data set.
2. Modify the JCL according to the instructions provided in the CQDRAMPS member.
3. Submit the job.

   The virtual tables and virtual target maps are installed.

Modifying the configuration member for RAA access

Enable and configure the parameters for IBM Rational Asset Analyzer (RAA) in the server configuration member.

About this task

The server configuration member contains a sample DATABASE definition that defines the first RAA database.

When enabling the database definition for the first RAA instance, the LOCATION and IPADDR parameters must be set to the database information for the DB2 on z/OS subsystem hosting the RAA database. The subsystem NAME(RAA1) should not be changed because a target subsystem map is configured to use this name for the virtual tables accessing the RAA database.

For multiple RAA databases, see “Adding an RAA database” on page 68.

The server member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and copied to *hlq*.CQDS.SCQDEXEC(CQDSIN00) by the job in the CQDGNMP1 member for you to make your local modifications.

Procedure

In the CQDSIN00 member, locate the comment “IBM Rational Asset Analyzer location". The following example shows the section in the configuration member to locate:

```
/*----------------------------------------------------------------*/
/* DRDA definition for IBM Rational Asset Analyzer location. RAA */
/* database definitions must have a NAME() starting with RAA      */
/*----------------------------------------------------------------*/
"DEFINE DATABASE TYPE(ZOSDRDA)"                    ,
                "NAME(RAA1)"                       ,
                "LOCATION(DRDAZOS)"                ,
                "DDFSTATUS(ENABLE)"                ,
                "PORT(443)"                        ,
                "IPADDR(127.0.0.1)"                ,
                "CCSID(37)"                        ,
                "APPLNAME(DSN1LU)"                 ,
                "IDLETIME(100)"

end
```

The following table lists the parameters for configuring support for RAA:

Parameter	Description	Valid values
TYPE	Database type. Because RAA stores information in DB2 for z/OS, this value must be ZOSDRDA.	ZOSDRDA

Parameter	Description	Valid values
NAME	The database name as known to the server. The first definition must be RAA1 because the target system map names this as the subsystem to access for RAA. For additional RAA databases, subsystems can have any name since you must also create a virtual target system to point to it; however, it recommended that the name start with RAA. (*Required*)	A valid value consists of 1 - 4 characters, starting with RAA. For example, RAA1.
LOCATION	Name of the database. The LOCATION parameter must be set to the database information for the DB2 on z/OS subsystem hosting the RAA database. (*Required*)	A valid value is a string 1 - 16 characters.
DDFSTATUS	The DDF activation status, which can be altered online by using the ISPF 4-DB2 dialog panels. (*Required*)	**ENABLE** Make this DDF definition active within Data Service server. **DISABLE** DDF endpoint is not used.
PORT	The TCP/IP port at which the server is listening. (*Required*)	A valid 1-5 numeric string.
IPADDR	Specify the dot-notation IPV4 address of the DDF endpoint. For the first RAA instance, the IPADDR parameter must be set to the database information for the DB2 on z/OS subsystem hosting the RAA database. (*Optional*)	If this parameter is not specified, the value 127.0.0.1 (local host) is the default. For group director definitions, use the DVIPA IP address of the group director.
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier) configured for this RDBMS subsystem on the RDBMS installation panel DSNTIPF, option 7. (*Optional*)	Refer to the RDBMS vendor documentation for a list of valid CCSIDs.

Parameter	Description	Valid values
APPLNAME	Application name. The APPLNAME used by the target endpoint for passticket generations. (*Optional*)	A valid value is 1 - 8 characters. If APPLNAME is not specified in the definition statement, no default value is provided and passticket access is disabled. **Note:** APPLNAME is not required when connecting from the JDBC driver.
IDLETIME	If DB2 ZPARM parameter IDTHTOIN is set to a non-zero value set IDLETIME to a value slightly less (10 secs.) than IDTHTOIN. This will also allow product DRDA threads to become inactive. (*DB2 for z/OS only*)	0-9999 seconds.

Adding an RAA database

Perform required configuration steps to add an RAA database.

About this task

For multiple RAA databases, you must perform configuration steps to define each additional RAA database. The following requirements apply when maintaining multiple RAA databases:

- For the first instance of an RAA database:
  - The database name in the must be RAA1.
  - The target system for the name RAA1 is automatically installed with the RAA maps, as described in "Installing virtual tables and virtual target maps for RAA access" on page 65.
- For subsequent RAA databases:
  - It is recommended that the database name start with RAA.
  - The target system must start with TSRAA.

Perform the following procedure for each additional RAA database.

Procedure

1. Repeat the database definition in the configuration member and make the following changes:
   a) Change the NAME value to a unique name (for example, RAA2).
   b) Change the LOCATION value to reference the DB2 subsystem hosting the RAA database.

   For information about the database definition parameters, see "Modifying the configuration member for RAA access" on page 66.
2. If the schema (table owner) used by RAA is not 'DMH', update the system global variable GLOBAL2.RAA.*database-name*.SCHEMA to the correct schema name for the RAA database tables.
3. Define a new virtual target system using the studio. The name of the virtual target system must start with TSRAA. This can be done in the Data Service Studio by selecting the **Create Virtual Target System** in the **Server** tab under the **SQL** > **Target Systems** > **DBMS** node of the tree. The connection value in each definition must match the NAME value defined in the DATABASE definition in the configuration member.
4. If required, create authentication information using the CQDDRATH batch utility.

Configuring virtual table rules for RAA

Configure Server Event Facility (SEF) rules to support multiple instances of the IBM Rational Asset Analyzer (RAA) schema using common virtual table and view definitions.

About this task

To support multiple instances of the RAA schema using common virtual table and view definitions, VTB rules CQDRAATB and CQDRAAVW provide support to process tables starting with RAA_ and views starting with RAAV_.

CQDRAATB
This table rule looks at the base view of a query for double underscores "__" and uses the data after the underscores to update the target subsystem for the query. This rule will also change the schema (or table owner) name of RAA tables from DMH to another value if the global system variable GLOBAL2.RAA.*database-name*.SCHEMA is set with an alternate schema name.

CQDRAAVW
This view rule looks for the double underscores and removes them from the view name to process.

With the rules activated, the Data Service Studio can suffix the view names with __SSID for all calls and process multiple instances of the RAA schema using a single set of maps.

These rules must be activated regardless of the number of RAA databases to be enabled.

Use the following procedure to set up these rules.

Procedure

Use the following steps to enable rules CQDRAATB and CQDRAAVW in the VTB ruleset:

a) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.

b) Specify option 2, **SEF Rule Management**.

c) Enter VTB for **Display Only the Ruleset Named**.

d) Enable the rule by specifying E and pressing Enter.

e) Set the rules to Auto-enable by specifying A and pressing Enter.

Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring authentication for RAA

Configure authentication for communicating with the IBM Rational Asset Analyzer (RAA) database.

About this task

Since RAA is hosted on a z/OS DB2 database, the z/OS credentials that are used to connect to Data Service should also be usable for the z/OS system where DB2 resides. By default, the Data Service server will attempt to use the same user ID that was presented for logon to z/OS for access to the RAA database. To use these credentials, the user ID must have SELECT access on the RAA tables in DB2.

If you choose to specify alternate credentials when communicating between the Data Service server and the RAA database, you must define what credentials to use. A facility is provided in the server to optionally change the logon credentials for a user when accessing the RAA database. To accomplish this, the following tools are provided:

CQDDRATH
A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

CQDEDB2G
An ATH rule that switches credentials when connecting to an RAA database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

You can use any of the following options for authentication:

- Use z/OS IDs for authentication
- Add a global default user definition using sample job CQDDRATH and enable ATH rule CQDEDB2G
- Add authentication information for specific mainframe users using sample job CQDDRATH and enable ATH rule CQDEDB2G

If z/OS user IDs and passwords used to connect to the Data Service server are not authorized for the Db2 database hosting the RAA tables, you must define the credentials to use. Use the following procedure.

Procedure

1. Use the sample job CQDDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the CQDDRATH member in the *hlq*.SCQDCNTL data set.
   b) Modify the JCL according to the instructions provided in the CQDDRATH member.

      When adding the SYSIN statements that define the alternate credentials for logging in to your RAA database, as instructed in the JCL, make sure to specify the correct DBTYPE. For RAA databases, specify `DBTYPE=ZOSDRDA`.
   c) Submit the job.
   d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the CQDDRATH member and submit the job.
2. Auto-enable the SEF ATH rule SCQDXATH(CQDEDB2G) to switch credentials when connecting to RAA using DRDA. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the IBM DB2 QMF Data Service - Primary Option Menu, select option **E** for Rules Mgmt.
   b) Select option **2** for SEF Rule Management.
   c) Enter * to display all rules, or ATH to display only authentication rules.
   d) Set Auto-Enable for the CQDEDB2G rule member by entering A and pressing Enter.

Chapter 4. Administering the Data Service server

You can perform tasks to manage the Data Service server.

Protected resources

System programmers typically configure advanced security during Data Service server customization. Data Service server provides protection for its resources by using RACF classes, CA Top Secret classes, and CA ACF2 generalized resource rules.

The overall RACF class (or resource type for ACF2) for Data Service is specified with the server parameter RESOURCETYPE. Classes can be shared among multiple instances of servers and either share the authorization rules or keep them separate.

Important: If the RESOURCETYPE parameter is not explicitly specified, the setting defaults to NON, which disables all product authorization checking.

When a user invokes a Data Service resource, the user's ID and the class of the resource are passed to the security program for authorization. The security program uses rules that you specify to determine whether to grant access to the resource.

To expedite future authorization checks of an identical request, Data Service server keeps the results of all security checks in protected storage.

The “look-aside” security check information is saved on a Task Control Block (TCB) basis and remains in effect until the TCB terminates. If you are initially denied access, but later have your security profile that is changed to allow access, you must exit the ISPF/SDF application to terminate its TCB. Depending on the security package, you may have to take other actions. Under ACF2, for example, you must issue the **ACFRESET** command. All security authorization events are logged in the Server Trace facility, and if access is denied, a message is produced.

The type of access you request — ADD/ALTER, READ, or UPDATE — depends on which resource you are using. The ACF2 ADD is equivalent to the RACF ALTER. See “Access requirements” on page 73 for the type of access that is required to use Data Service facilities.

Enabling security parameters for resource rules

To enable the security parameters, change `if DontDoThis` to `if DoThis`.

```
if DoThis then
do
  "MODIFY PARM NAME(RESOURCETYPE) VALUE(RCQD)"
end
```

Parameter name	Parameter description	Default value
RESOURCETYPE	RESOURCE TYPE FOR RESOURCE RULES Contains the name of the security server s class (or resource type for ACF2) that is used to perform resource access authorization checks. If not explicitly specified, this parameter defaults to NON. Valid values: **NON** Disables all product authorization checking. **Important:** If you leave generalized resource checking disabled, a security exposure may exist. Anyone with a valid TSO user ID can gain access to the Data Service ISPF control application, where they are fully authorized to perform the functions that are provided by the interface. This assumes, however, that the user has sufficient information at hand to log on to TSO/E and then gain access to the ISPF/SDF application. ***classname*** RACF class name or ACF2 resource type. When using RACF, the corresponding class name within RACF must start with R, for example, RCQD.	NON

List of protected resources

The following table describes the resources that are protected by the Data Service security mechanism.

Note: You cannot modify the resource names.

Table 6. Protected resources

Resource name	Description
ACI.aci-mapname	Access to an ACI (Advanced Communication Interface) service definition.
ADA.ADABAS-file-name	Access to an Adabas file name.
ADATRACE	Authority to issue Adabas TRACE ON and TRACE OFF commands.
ADAxxxxx.FILyyyyy	Access to an Adabas file ID number.
ATHZOOM	Access to Server Trace authorization event PF4 Zoom information.
CICSCONNECTIONS	Access to monitor and control CICS connections.
CONTROLBLOCKS	Data Service internal data structures.
CQD	Access to the ISPF/SDF interactive control facility.
DATABASES	Access databases that are defined to Data Service.
DATAMAP	Access to the Data Mapping Facility.
FILE	Access to shared files that are defined to Data Service.
FILETYPE	Access to the Data Service file-suffix/MIME-type control table.
GLOBALS	Access to global variables.
IMSLTERM	Tables correlating user IDs or TCP/IP addresses to LTERM to legacy LTERM security can be supported using an APPC interface.

Table 6. Protected resources (continued)

Resource name	Description
LINKS	Access to communication links that are defined to Data Service.
PARMS	Access to the ISPF/SDF parameter display.
RPC.<rpc_name>	RPC-based security.
SEF	Access to the Event Facility dialogs.
SIS	Access to the Instrumentation Server.
TOKENS	Access to the Data Service tokens display.
TRACEBROWSE	Access to the Server Trace facility.
TRACEDATA	Access to all trace data, including SQL and underlying binary file trace records.
USERS	Access to the attached/remote users applications.

Access requirements

The following table provides the type of access that is required to use each Data Service facility.

Table 7. Data Service access requirements

Resources	Action	Suggested user	Access required
ADATRACE	Issuing the **ADABASTRACE ON** and **OFF** commands.	DBA, Program Products, VTAM, Operations	READ
ATHZOOM	Viewing Server Trace authorization event PF4 zoom information.	DBA, Program Products, VTAM, Operations	READ
CONTROLBLOCK	Using the Data Service command.	DBA, Program Products, VTAM, Operations	READ
CONTROLBLOCK, CQD	Viewing product control blocks using the ISPF/SDF option CQD.	DBA, Program Products	READ
CONTROLBLOCK, CQD	Modifying product control blocks using a future facility.	DBA, Program Products	UPDATE
CQD	Defining links using the **ADDRESS CQD DEFINE LINK** command.	DBA, Program Products, VTAM, Operations	ADD/ALTER
DATABASES	Viewing databases using the **ADDRESS CQD DISPLAY DATABASE** command.	DBA, Program Products, VTAM, Operations	READ
DATABASES, CQD	Modifying databases using the **ADDRESS CQD MODIFY DATABASE** command.	DBA, Program Products	UPDATE

Table 7. Data Service access requirements (continued)

Resources	**Action**	**Suggested user**	**Access required**
GLOBALS	Viewing global variables.	All (DBA, Program Products, Operations, Developers, End-Users)	READ
GLOBALS	Updating global variables.	DBA, Administrator, Developers	UPDATE
IMSLTERM, CQD	Correlating user IDs or TCP/IP addresses to LTERMs.	DBA, Administrator	READ, UPDATE
LINKS	Viewing links using the **ADDRESS CQD DISPLAY LINK** command.	DBA, Program Products, VTAM, Operations	READ
LINKS, CQD	Modifying links using either the **ADDRESS CQD MODIFY LINK** command.	DBA, Program Products, VTAM, Operations	UPDATE
LINKS, CQD	Defining databases using the **ADDRESS CQD DEFINE DATABASE** command.	DBA, Program Products	ADD/ALTER
PARMS, CQD	Modifying the product parameters the **ADDRESS CQD MODIFY PARM** command.	DBA, Program Products, VTAM, Operations	UPDATE
PARMS, CQD	Viewing all Server Trace data.	DBA, Program Products, VTAM, Operations	READ
SEF, DATAMAP	Refreshing Data Maps	DBA, Admin	READ access to SEF; UPDATE access to DATAMAP.
TRACEBROWSE, TRACEDATA, CQD	Issuing SQL statements via CQDSPUFI.	DBA, Program Products, VTAM, Operations	READ
USERS, CQD	Viewing remote users the **ADDRESS CQD DISPLAY REMOTE** command.	DBA, Program Products, VTAM, Operations	READ
USERS, CQD	Killing remote users using the ISPF/SDF option **CQD Admin / CQD Group**	DBA, Operations, Developers, End-Users	READ, UPDATE
USERS, CQD	Viewing product Data Service parameters using the **ADDRESS CQD DISPLAY PARM** command.	DBA, Program Products, VTAM, Operations	READ

Defining resources to RACF

Procedure

1. Use the following JCL as a model for defining a new RACF class to the RACF class descriptor table for RCQD.

   ```
   //STEP1 EXEC ASMHCL
   //C.SYSLIB DD DSN=SYS1.MODGEN,DISP=SHR
   //C.SYSIN DD *
   RCQD ICHERCDE CLASS=RCQD,
          ID=128,
          MAXLNTH=39,
          FIRST=ALPHANUM,
          OTHER=ANY,
          POSIT=25,
          OPER=NO
        ICHERCDE
   /*
   //L.SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR
   //L.SYSIN DD *
        INCLUDE SYSLMOD(ICHRRCDE)
          ORDER RCQD
          ORDER *** Previous user-defined classes ***
          ORDER *** Previous user-defined classes ***
          ORDER ICHRRCDE
        NAME ICHRRCDE(R)
   /*
   ```

   Restart the Data Service server so that RACF recognizes the new class.
2. Perform an IPL to change the RACF class descriptor table. This procedure is necessary for RACF to recognize the new class.
3. Define all RACF resource types to class RCQD with the following command:

   ```
   RDEFINE RCQD CONTROLBLOCKS UACC(NONE)
   ```

   Repeat the RDEFINE command for each RACF resource type.
4. Provide access to the resource according to the following example:

   ```
   PERMIT CONTROLBLOCKS CLASS(RCQD) ID(USERID) ACCESS(READ)
   ```

   Where USERID is the ID of the user to whom you want to grant READ permissions access.

   If you do not want the FACILITY class to be used, the *hlq*.SCQDCNTL(CQDRADF2) member can be used as a sample for how to define the RACF class descriptor and router table.

   You can edit and submit the job in *hlq*.SCQDCNTL(CQDRARES) to define and add permissions for the resource required by your site.
5. Activate the class to RACF with the following command:

   ```
   SETROPTS CLASSACT(RCQD)
   ```

What to do next

These members must be updated every time a new security resource name such as ATHZOOM or USERS is added.

Defining resources to CA Top Secret

Procedure

1. Define an entry in the RDT, as shown in the following example:

```
TSS ADDTO(RDT) RESCLASS(CQD) RESCODE(nn)-
       ATTR(LONG,PRIV,LIB,DEFPROT,GENERIC)-
       ACLST(NONE,ALL,ALTER=1COO,UPDATE,READ)DEFACC(READ)
```

Where *nn* is a hexadecimal code between 01 and 3F.

2. Add all the resources to an owner with the following commands:

   `TSS ADDTO(owner) CQD(CONTROLBLOCKS)`

   Repeat this `TSS ADDTO` command for all resource types.
3. Permit the resources to profiles or users as follows:

```
TSS PERMIT(userid) CQD(TRACEDATA) ACC(READ)
```

4. You can edit and submit the job in *hlq*.`SCQDCNTL(CQDTSRES)` to define and add permissions for the resource required by your site.

What to do next

These members must be updated every time a new security resource name such as ATHZOOM or USERS is added.

Defining resources to ACF2

Procedure

1. Define a generalized resource class named CQD.
2. Define resource rules for each of the resource class. Member *hlq*.`SCQDCNTL(CQDA2RES)` can be used as an example.
3. Use the following ACF2 command to allow users access to the resource rule:

```
ACFNRULE KEY(TRACEBROWSE) TYPE(CQD) ADD(UID(*********userid) ALLOW
```

4. You can edit and submit the job in *hlq*.`SCQDCNTL(CQDA2RES)` to define and add permissions for the resource required by your site:

Optional security jobs

The following table lists the jobs that are in the *hlq*.`SCQDCNTL` library. The jobs can be edited and submitted for the purpose that is specified in the Description column.

Table 8. Optional security jobs

Description	**RACF**	**CA ACF2**	**CA Top Secret**
ACI persistent connection security	RACFACI	ACF2ACI	TSSACI
ADABAS file name or file ID	CQDRAADA	CQDA2ADA	CQDTSADA
BPEL role-based security	RACFBPEL	ACF2BPEL	TSSBPEL
CICS transaction security	RACFCICS	ACF2CICS	TSSCICS
DB2 RRSAF security	RACFDB2	ACF2DB2	TSSDB2
IDMS transaction security	RACFIDMS	ACF2IDMS	TSSIDMS
IMS OTMA and transaction	RACFIMS	ACF2IMS	TSSIMS
Permissions that are required for JVM installation	RACFJVM	ACF2JVM	TSSJVM
MQ security for Streams	RACFMQ	ACF2MQ	TSSMQ
IBM® DB2 QMF® Data Service Resource security	CQDRARES	CQDA2RES	CQDTSRES

Table 8. Optional security jobs (continued)

Description	RACF	CA ACF2	CA Top Secret
IBM® DB2 QMF® Data Service RPC security	RACFRPC	ACF2RPC	TSSRPC
Defining IBM® DB2 QMF® Data Service started task to security product	CQDRAVDB	CQDA2VDB	CQDTSVDB
RRS XA-2PC security	RACFXA	ACF2XA	TSSXA
Streams security	RACFZEV	ACF2ZEV	TSSZEV

ISPF load modules

If you use TSO Command to restrict access to TSO commands, you must define the IBM® DB2 QMF® Data Service ISPF load modules to your security product.

Table 9. IBM® DB2 QMF® Data Service load modules

Load module	Description
CQD	TSO command to invoke S__ interactive application.
CQD2RU	Routine to invoke IBM® DB2 QMF® Data Service ISPF application.
CQDI	REXX Implicit Interpreter TSO Command processor.
CQDICOMP	REXX Implicit Interpreter TSO Command processor.
CQDIDB	REXX Implicit Interpreter TSO Command processor.
CQDIMEX	REXX Implicit Interpreter TSO Command processor.
CQDOB	Alias for CQDOCP.
CQDOCP	Trace Browse routine.
CQDORU	Trace Browse routine.
CQDX	REXX Implicit Interpreter TSO Command processor (Server REXX).
CQDXCOMP	REXX Implicit Interpreter TSO Command processor.
CQDXDB	REXX Implicit Interpreter TSO Command processor.
CQDXSCAN	REXX Implicit Interpreter TSO Command processor.
SDHOCM	Host command environment for address CQD.
SDISCBRU	Display product control blocks.
SDISSTRU	Display product statistics.
SDISTBRU	General-purpose table display routine.
SDISVARU	ISPF product variables display.
SDLINK	Main product module.
SDRXBR	Browse routine for REXX S__ line variables.
SDRXDM	A REXX function to call new DMF parser.
SDRXID	A REXX function for issuing commands to IDCAMS.
SDRXIN	Initialize the REXX environment.
SDRXLELK	Bridge REXX TO LE/370 main routine.

Table 9. IBM® DB2 QMF® Data Service load modules (continued)

Load module	Description
SDRXPC	Product-related control block function.
SDRXSG	REXX function for examining storage in another address space.
SDRXST	Product-related control block function.
SDRXTE	Terminate REXX environment.
SDRXTK	REXX function for parsing strings into token.
SDRXVA	REXX function for manipulating variables in a calling REXX exec.
SDSLSVMD	SSL
SDSLUTCC	SSL
SDSLUTCK	SSL
SDSLUTDE	SSL
SDSLUTKY	SSL
SDSLUTPA	SSL
SDSLUTRQ	SSL

RACF PassTickets

The RACF PassTicket can be used instead of a user logon password.

About this task

When you use a RACF PassTicket, the default application name that is passed is the three-character subsystem ID code (for example, CQD for IBM® DB2 QMF® Data Service) appended with the system SMFID. This application name must match a PTKTDATA profile name for PassTicket generation and authentication to work. For example, if the system SMFID is DEV1, the application name is CQDDEV1, and you must define a PKTDATA profile for IBM® DB2 QMF® Data Service with the name CQDDEV1. The default application name can be changed by using the PASSTICKETAPPNAME parameter.

Also, a PTKTDATA profile name can be further qualified by RACF user ID and/or RACF connect group (for example, CQDDEV1.CQDS or CQDDEV1.SYS1.CQDS). This allows different instances of an application to have unique single sign-on keys.

For more information on defining profiles in the PTKTDATA class, see the *z/OS Security Server RACF Security Administrator's Guide*.

Defining security for RPCs

About this task

Use the following procedure if you need to restrict access to RPCs:

Procedure

1. Use the MODIFY PARM command to add the following parameters that are located in the CQDSIN00 configuration member:

```
“MODIFY PARM NAME(ACF/2SAFCALL) VALUE(NO)”
“MODIFY PARM NAME(CHECKRPCAUTHORITY) VALUE(YES)”
```

```
“MODIFY PARM NAME(RESOURCETYPE) VALUE(RCQD)”
“MODIFY PARM NAME(TRACEAUTHEVENTS) VALUE(YES)”
```

Parameter	Description	Valid values
ACF/2SAFCALL (*ACF2 Only*)	To use RPC security with ACF2, you must run a version of ACF2 that supports SAF calls.	**YES** Enables Data Service server to use SAF calls for Resource Rules. **NO** (default) All users are allowed to run all RPCs. The RPC can always provide its own security.
CHECKRPCAUTHORITY	Controls whether the SEF and ACF2/RACF should be used to check whether each user has the authority to run each RPC. If set to YES, the SEF and ACF2/RACF are used to verify RPC execution authority.	**YES** The SEF and ACF2/RACF is used to verify RPC execution authority. **NO** (default) All users are allowed to run all RPCs. The RPC can always provide its own security.
RESOURCETYPE	Contains the name of the security server's class (or resource type for ACF2) that is used to perform resource access authorization checks.	
TRACEAUTHEVENTS (*Optional*)	Turn on authorization event tracing (this allows you to trace the RPC security checks).	**YES** **NO** Default value is NO.

2. If you want to define all RPCs to your security product and grant RPC access to the specific users, you must edit and submit one of the following sample jobs that are located in the *hlq*.SCQDCNTL library.

   - RACFRPC for RACF security
   - ACF2RPC for CA ACF2 security
   - TSSRPC for CA Top Secret security

   Note: If you enable the CHECKRPCAUTHORITY parameter, you must define each RPC to your security product.

Information access with the TRACEDATA resource

The TRACEDATA resource controls access to information in the trace log.

About this task

The two types of information that are contained within the Data Service server trace log:

- SQL source statements (the real SQL source statements, as taken from database request modules or prepared strings, which may contain objects such as table names or column names).
- Binary data that underlies the trace log.

Users who have READ authority for the TRACEDATA resource and READ authority for CQD and TRACEBROWSE can view the entire trace log. Users who do not have READ authority have only restricted access to this information.

For SQL events, if your user ID matches the user ID associated with the event, you are permitted to look at an uncensored log of the SQL event. Otherwise, you can only see a censored representation of the SQL statement. The censored version includes the SQL verb but does not include objects, such as table names or column names.

The TRACEDATA resource restricts data differently, depending on the type of event:

- SQL Events: If your user ID matches the user ID associated with the event, you are permitted to look at an uncensored log of the SQL event. Otherwise, you can only see a censored representation of the SQL statement. The censored version includes the SQL verb but does not include objects, such as table names or column names.
- Non-SQL Events: If your user ID matches the user ID associated with the event, you are permitted to see an uncensored view of the underlying binary data for event. Otherwise, you are not allowed to see the binary data at all; no data is displayed and a message is written to the terminal.

Resource security for test versions of Data Service server

All resource security is simulated for test versions of the Data Service server running in a TSO session. The z/OS security subsystem is not consulted, because a test TSO copy of the product is not authorized to perform this type of security check. All work is performed using the TSO user’s existing z/OS authorizations.

In this environment, all security checks are assumed to complete successfully. If you are running test copies of the Data Service server under TSO, you should find this feature helpful in deploying new applications, because you can review the security checks that occur when the application is deployed in a production environment.

Workload Manager (WLM)

Using the IBM Workload Manager for z/OS, you can define performance goals and assign a level of importance to each goal in business terms. The system matches its resources to the work and determines whether goals are being met by monitoring and adapting its processing. This allows you to make the best possible use of the server's resources, while achieving the best possible response times.

Goals are specified for the WLM services in IBM® DB2 QMF® Data Service in the same way they are specified for z/OS-managed work, by associating work with a service class. The assigned service class informs the operating system about the performance goal and importance level that is associated with the work, as well as the address spaces involved in processing the work request.

Support for the Workload Manager (WLM) is available for the SQL data access. For information about planning for and using workload management, refer to the IBM Knowledge Center for the *MVS Planning: Workload Management* and *MVS Workload Management Services* documents.

WLM enclaves

To facilitate implementation of transaction management, WLM uses enclaves. An enclave is a group of one or more logically related z/OS task control blocks (TCB) and service request blocks (SRB) that manage the work in entities.

Using enclaves provides the following benefits:

- Work running in enclave SRBs can be offloaded to a zIIP processor. The Data Service server runs in enclave SRB mode, when possible, to allow CPU offloading.
- The resources that are used to process the transaction can be accounted to the transaction rather than to the address space in which the transaction runs. Service class performance goals are inherited by the enclave.

The Data Service server establishes a logical dispatchable unit (LDU) for each process and thread in its address space. This LDU consists of a TCB/SRB pair that is dispatched in SRB mode in a WLM enclave, if possible, switching to TCB mode only if required by system or database interfaces. The SRB mode execution is eligible for offloading to a zIIP based on the definitions in the WLM service policy.

During installation, the Data Service server establishes two long-running enclaves. One is the service class `CQD_SCNM`, and the other is the service class `CQD_SCHI`. Dispatchable units join these enclaves as appropriate. Unique enclaves are created as needed for the processes and threads for SQL data access.

Configuring Workload Manager (WLM)

You use WLM to define performance goals and assign a level of importance to each goal in business terms.

The system then matches its resources to the work, as well as monitors the goals and makes necessary processing adoptions accordingly.

This section explains several ways that you can configure WLM support and provides the definitions that are required to use the support.

WLM definitions

A service definition is the name that is given to the combination of service policies, workloads, service classes, resource groups, classification rules, and application environments. It is based on the performance objectives in a service level agreement (SLA). The following is a list of WLM definitions:

Workload

A named group of work, or service classes, that is reported as a unit.

Service Class

A named group of work that has similar performance goals, resource requirements, or importance. In the service class, you assign each goal and its relative importance, and associate the service class with a specific workload and resource group. IBM® DB2 QMF® Data Service requires the following service classes.

- `CQD_SCHI ZIIPCLASS=CQD` High priority. This service class is for IBM® DB2 QMF® Data Service critical work. Assign this class goal as close to SYSSTC as possible.
- `CQD_SCNM ZIIPCLASS=CQD` Normal work. This class is for IBM® DB2 QMF® Data Service administrative work. Assign this class the same goals as those used for DB2 master or the IMS control region.
- `CQD_SCTX ZIIPCLASS=CQD` Client work. This service class is for client requests. Assign this class the same goals as those supporting the data source. This would most likely be the CICS, IMS/TM, or DB2 WLM address space.

Classification Rules

A classification rule maps work coming into the system to a specific service class and report class. A classification is based on the subsystem type and work qualifiers in the subsystem type. The work qualifiers define and associate service classes to the type of work.

Report Class

A named group of work that is for reporting purposes only. Use report classes to distinguish among types of work that run in the same service class.

Providing WLM definitions via Data Service

Before you begin

Before you start this procedure, it is important to understand the following requirements:

- Data Service must have proper access to the MVSADMIN.WLM.POLICY resource.

- Your user ID must have UPDATE access or the following error occurs:

```
*SDx0038S INSTALL OF WLM SERVICE DEFINITION FAILED, RC=X'0000000C',
REASON=X'0A3E0C0E', DETECTED AT OPINWM+X'FFC3BF06'
```

- Your user ID for starting the server must have READ access or the following error occurs:

```
SDx3269I WLM administration userid xxxxxxxx logged on to system

SDx0037E WLM EXTRACT SERVICE DEFINITION FAILED, RC=X'00000004', DETECTED AT
OPINWM+X'00000B02'
```

Procedure

1. Add the following statements to your CQDSIN00 configuration member:

```
If DoThis then
  do
    “MODIFY PARM NAME(WLMFORCEPOLICY) VALUE(YES)”
    “MODIFY PARM NAME(WLMTRANNAME) VALUE(APPLNAME)”
    “MODIFY PARM NAME(WLMUSERID) VALUE(CQDS)”
End
```

The following table lists the parameters for WLM definitions:

Parameter	Description	Valid values
WLMFORCEPOLICY	Controls whether the Data Service server enforces service policy requirements.	**YES** The server initialization examines the active policy for required elements and terminates if the elements do not exist and the server is not allowed to add them. The server also examines the policy anytime it is refreshed, and shuts down the server if the new policy is not in compliance with server requirements. **NO** (default) The Data Service server checks the policy for required definitions, and issues an error message if the subsystem type (default CQD, set by WLMSUBSYSTEM) is not defined in the policy. The server is allowed to initialize, and is not shut down for any policy changes.

Parameter	Description	Valid values
WLMTRANNAME (*optional*)	Specifies which value is used as the transaction name when classifying the Data Service server transactions.	**APPLNAME** (default) The application name set in the client ODBC data source is used as the transaction name. **MODNAME** The name of the application that uses the client ODBC driver is used as the transaction name. **INTNAME** The client application executable internal name is used as the transaction name.
WLMUSERID (*optional*)	Specifies a highly privileged user ID under which WLM administration functions are performed. This user ID must be authorized to update the MVSADMIN.WLM.POLICY resource. If WLMUSERID is not specified, the server subsystem ID is used for WLM policy administration.	CQDS (default subsystem ID)

2. Enter WLM from the ISPF/PDF option 6 panel to log on to the IBM TSO/ISPF WLM administration tool.
3. Extract and save a copy of the current service definition. This is for backup purposes only.
4. Optional: Update the `CQDSIN00` configuration member with a valid `WLMUSERID`.
5. Start Data Service server.

   Upon startup, Data Service:

   - Examines the current WLM service policy for the required elements. If the active policy contains the required elements, initialization continues. If the required elements are not found, Data Service messages xDy0706I, and xDy0707I are issued for each missing element.

```
xDy0706I DATA VIRTUALIZATION SERVER CQDS requires the following elements
 missing from WLM Service Policy active_policy_name

xDy0707I Type: WORKLOAD, Data Virtualization Parameter: WLMWORKLOAD,
Value: CQD_WKLD

xDy0707I Type: SUBSYSTEM, Data Virtualization Parameter: WLMSUBSYSTEM,
Value: CQD

xDy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMSERVICECLASS, Value: CQD_SCNM

xDy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMHISERVICECLASS, Value: CQD_SCHI

xDy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMTXSERVICECLASS, Value: CQD_SCTX
xDy0707I Type: REPORT CLASS, Data Virtualization Parameter:
 WLMP1REPORTCLASS, Value: CQD_RCP1

xDy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMP2REPORTCLASS Value: CQD_RCP2

xDy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMP3REPORTCLASS, Value: CQD_RCP3
```

```
xDy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMTRANSACTION, Value: CQD_TNNM

xDy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMHITRANSACTION, Value: CQD_TNHI

xDy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMTXTRANSACTION, Value: CQD_TNTX
```

- Data Service then examines the WLM service definition for the required elements. If `WLMFORCEPOLICY` is set to NO, the following actions are skipped. If `WLMFORCEPOLICY` is set to YES, the following actions are enforced. The default is NO.

  Action 1: If the required elements are found in the service definition, a WTOR is issued, requesting permission to activate the current service policy. If the current policy is no longer in the service definition, the user is asked to select one of the policies in the service definition for activation.

  ```
  *nn xDy0719R Reply 'GO' to activate Policy
  service_policy_name, or 'CANCEL' to terminate
  Server initialization
  ```

  If you reply with CANCEL, the Data Service server shuts down.

  If you reply with GO, the Data Service server automatically activates your WLM Policy *service_policy_name*, and you should see the following message in the system log:

  ```
  IWM001I WORKLOAD MANAGEMENT POLICY service_policy_name NOW IN EFFECT
  ```

- **Action 2**: If the required elements are not found in the service definition, the Server issues message xDy0706I, and then message xDy0707I for each missing element.

  ```
  xDy0706I DATA VIRTUALIZATION SERVER CQDS requires the following elements
  missing from WLM Service Definition service_definition_name.

  xDy0707I Type: WORKLOAD, Data Virtualization Parameter: WLMWORKLOAD,
  Value: CQD_WKLD

  xDy0707I Type: SUBSYSTEM, Data Virtualization Parameter: WLMSUBSYSTEM,
  Value: CQD

  xDy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
  WLMSERVICECLASS, Value: CQD_SCNM

  xDy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
  WLMHISERVICECLASS, Value: CQD_SCHI

  xDy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
  WLMTXSERVICECLASS, Value: CQD_SCTX

  xDy0707I Type: REPORT CLASS, Data Virtualization Parameter:
  WLMP1REPORTCLASS, Value: CQD_RCP1

  xDy0707I Type: REPORT CLASS, Data Virtualization Parameter:
  WLMP2REPORTCLASS, Value: CQD_RCP2

  xDy0707I Type: REPORT CLASS, Data Virtualization Parameter:
  WLMP3REPORTCLASS, Value: CQD_RCP3

  xDy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
  WLMTRANSACTION, Value: CQD_TNNM

  xDy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
  WLMHITRANSACTION, Value: CQD_TNHI

  xDy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
  WLMTXTRANSACTION, Value: CQD_TNTX
  ```

  The preceding messages are followed by a WTOR requesting permission to update the service definition.

  ```
  *nn xDy0708R Reply 'GO' to update the WLM Service Definition, or
  'CANCEL' to terminate server initialization
  ```

If you reply with CANCEL, Data Service server shuts down.

If you reply with GO, the Data Service server automatically makes the proper WLM updates to your WLM policy definition. At the conclusion of the update process, you receive the following message.

```
xDy0709I WLM Service Definition service_definition_name has been updated
with required elements
```

Action 3: A separate WTOR message is presented to activate the policy.

```
*nn xDy0719R Reply 'GO' to activate Policy service_policy_name, or
'CANCEL' to terminate server initialization
```

If you reply with CANCEL, Data Service server shuts down. The user can use the TSO/ISPF WLM administration dialog to extract the service definition and review the additions that are made by the Data Service server.

If you reply with GO, the Data Service server automatically activates your WLM policy *service_policy_name*, and you see the following message:

```
IWM001I WORKLOAD MANAGEMENT POLICY service_policy_name NOW IN EFFECT
```

Note: After the WLM service policy is activated, if you change any IBM® DB2 QMF® Data Service required WLM element in the service definition to an invalid value and activate a service policy, all servers requiring the now invalid definition shut down.

Note: You should have a backup of your existing WLM service policy definitions.

Providing WLM definitions manually

About this task

If you want to manually define the required WLM definitions rather than have the Data Service server automatically install them at startup time, take the following steps:

Procedure

1. Start the WLM administration tool. The IBM TSO/ISPF WLM administration tool is used in the following examples. Other administrative tools can also be used.
   a) Enter **WLM** from the ISPF/PDF option 6 panel to log on to the IBM TSO/ISPF WLM administration tool.
   b) Select **Option 2 Extract Definition from WLM Couple Data Set** from the Choose Service Definition box.
2. Define the workloads.
   a) Select **Option 2 Workloads**. Press Enter.
   WLM displays the Workload Selection List panel.
   b) Create workload CQD_WKLD.
3. Define the service classes.
   a) Select **Option 4 Service Classes**. Press Enter.
   WLM displays the Service Class Selection List panel.
   b) Here you create the following service classes:
      - `CQD_SCHI ZIIPCLASS=CQD` IBM® DB2 QMF® Data Service high priority
      - `CQD_SCNM ZIIPCLASS=CQD` IBM® DB2 QMF® Data Service normal work
      - `CQD_SCTX ZIIPCLASS=CQD` IBM® DB2 QMF® Data Service client work

   Note:

- Do not change service class names.
- ZIIPCLASS=CQD is a required keyword in the description.
- The values that are shown for service class goals are default values that you can modify.

4. Define subsystem type CQD and its classification rules.

   a) Select **Option 6 Classification Rules**. Press Enter.

   WLM displays the Subsystem Type Selection List for Rules panel.

   b) Define subsystem type CQD and associated classification rules.

5. Define the report classes.

   a) Select **Option 7 Report Classes**. Press Enter.

   WLM displays the Report Class Selection List panel.

   b) In this panel, create the following report classes:

   - CQD_RCP1 D1000 P100 PERIOD 1
   - CQD_RCP2 D1500 P100 PERIOD 2
   - CQD_RCP3 P100 PERIOD 3

   Note:

   - Do not change report class names.
   - The terms in the report class descriptions are used to provide CPU offload criteria for Data Service server work as follows:

     D*nnnn*: The number of service units during which the dispatchable units are in the associated period while eligible for offloading to the zIIP processor.

     P*nnn*: The percentage of time in the associated period that Data Service server tries to offload work to the zIIP processor.

6. Activate a Service Policy.

   a) Select **Option 3 Activate Service Policy** from the **Utilities** drop-down menu on the panel.

   b) Follow directions to activate a policy.

Using the WLM Administration Tool

Procedure

1. Enter the following command to start the IBM TSO ISPF administration tool:

```
TSO WLM
```

2. Follow all prompts until the **Choose Service Definition** panel is displayed.
3. Type 2 to select the **Extract definition from WLM couple data set** option.
4. Press **ENTER**. The **WLM Definition** panel appears. You can select the option for the task that you want to perform.

Workload Manager definitions

During initialization, Data Service server connects the server address space to the WLM and ensures that WLM elements are in the current active service policy.

Table 10. WLM Element Types

WLM Element Type	Server Parameter	Default Value
Workload	WLMWORKLOAD	CQD_WKLD
Subsystem	WLMSUBSYSTEM	CQD

Table 10. WLM Element Types (continued)

WLM Element Type	Server Parameter	Default Value
Service Class	WLMSERVICECLASS	CQD_SCNM
Service Class	WLMHISERVICECLASS	CQD_SCHI
Service Class	WLMTXSERVICECLASS	CQD_SCTX
Classification Rule	WLMTRANSACTION	CQD_TXNM
Classification Rule	WLMHITRANSACTION	CQD_TXHI
Classification Rule	WLMTXTRANSACTION	CQD_TXTX
Report Class	WLMP1REPORTCLASS	CQD_RCP1
Report Class	WLMP2REPORTCLASS	CQD_RCP2
Report Class	WLMP3REPORTCLASS	CQD_RCP3

Modifying the workload

The workload, CQD_WKLD, is required by the Data Service server.

About this task

To modify the IBM® DB2 QMF® Data Service workload definition:

Procedure

Select the **Workloads** option from the **WLM Definition** panel (see "Using the WLM Administration Tool"). Press Enter.

The system displays the **Modify a Workload** panel.

Results

Note: You can change the **Workload Name** field by using the WLMWORKLOAD parameter, which is located in the server configuration member, CQDSIN00. Do not change this name unless instructed to do so by IBM Software Support.

Modifying a service class definition

Before you begin

For details about setting up service class definitions, refer to the IBM Knowledge Center for the *MVS Planning: Workload Management* and *MVS Workload Management Services* documents.

- The CQD_SCHI service class is used for high importance server work, such as management tasks of short duration that should not be interrupted, establishing a new thread for a new transaction.
- CQD_SCNM is the default service class for all work that is not explicitly classified, except for the following types of work:
  - Server process LDUs that are assigned CQD_SCHI.
  - SQL transactions are classified according to WLM classification rules. If a Data Service server classification rule is added that assigns SQL transactions to CQD_SCNM or CQD_SCHI, the LDU representing the transaction is joined to one of the long-running enclaves that are established for the transaction task. A new enclave is created for this LDU.
- The CQD_SCTX service class is used for SQL transactions that are not otherwise classified. A new enclave is created for each LDU assigned to CQD_SCTX.

About this task

To modify the service class definition:

Procedure

1. Select the **Service Classes** option from the **WLM Definition** panel. Press Enter.

   The system displays the **Service Class Selection List** panel.
2. Select a definition in the service class selection list. Select Enter.

   The system displays the following panel that shows the default definition for the CQD_SCNM service class.

   The description contains the following information:

   - The service class name can be modified by using the `WLMHISERVICECLASS` parameter, which is located in the `CQDSIN00` configuration member.

     Note: Do not change this name unless you are told to do so by Technical Support.
   - The `ZIIPCLASS=CQD` in the description field is used to construct the names of report classes that have CPU offload criteria that are specified in their descriptions. If the `ZIIPCLASS` keyword is not specified correctly, IBM® DB2 QMF® Data Service work that is dispatched as enclave SRBs assigned to this service class is not off loaded to the zIIP.
   - The workload name is for reporting purposes only and can be changed to any valid workload name.
   - The service class goal is a single period with an execution velocity goal. The percentage and importance can be changed, but set them at a level appropriate to a mission-critical server.

Viewing subsystem and classification rules

View the Data Service server classification rules.

About this task

The following classification rules are required:

- The subsystem type must be `CQD`.
- A rule classifying transaction `CQD_TXHI` to service class `CQD_SCHI`.
- A rule classifying transaction `CQD_TXNM` to service class `CQD_SCNM`.
- A rule classifying transaction `CQD_TXTX` to service class `CQD_SCTX`.

Procedure

1. Select the **Classification Rules** option from the **WLM Definition** panel (see “Using the WLM Administration Tool”).
2. Select **CQD** from the list of rules in the classification rules selection.
3. Press **ENTER**. The system displays the **Modify Rules for the Subsystem Type** panel that shows the default definition for the SDB classification rules.

Results

Do not change the classification rules. They are used internally by the Data Service server. Classification rules for SQL, Streams, and Services can be added to these rules.

Modifying a report class definition

Before you begin

The following report classes are required by the Data Service server:

- CQD_RCP1
- CQD_RCP2
- CQD_RCP3

About this task

To modify a report class definition:

Procedure

1. Select the **Report Classes** option from the **WLM Definition** panel. Press Enter.

   The system displays the **Report Class Selection List** panel.
2. Select a report class name from the list of report classes. Press Enter.

   The system displays the following panel that shows the default definition for the report class.

   The panel shows the following information:

   - The report class definition is used to provide first period CPU offload information for service classes. The report class is not used in the classification rules.
   - The report class name can be modified by using the WLMP1REPORTLASS parameter, which is located in the CQDSIN00 configuration member.

     Note: Do not change this name unless you are told to do so by IBM Software Support.

     The format of the report class name is:

     *xxx*_RCP1

     where *xxx* is a ZIIPCLASS=*xxx* specification on a service class description and _RCP1 is fixed and must not be changed.
   - The D*nnnn* in the description field is the first period duration in service units for CPU offloading. The *nnnn* value can be adjusted by the user.
   - The *P100* in the description field is the percentage of time in the first period that WLM attempts to offload enclave SRBs in the associated service class to the zIIP.

WLM classification rules

WLM classification rules apply to the SQL solution.

Note: Before defining classification rules, make sure that WLM is installed and set up correctly.

SQL

The Data Service server establishes a unique enclave for each transaction. WLM classification rules can assign this enclave to a service class with velocity or response goals and one or more periods.

WLM populates the enclave definition with the following information:

- Client User ID. WLM uses the client user ID to find a classification rule match. The client user ID is mapped to the WLM qualifier type UI.
- DB2 Plan Name. WLM uses the DB2 plan name to find a classification rule match. The DB2 plan name is mapped to the WLM qualifier type PN.
- DB2 Subsystem Name. WLM uses the DB2 subsystem ID to find a classification rule match. The DB2 subsystem name is mapped to the WLM qualifier type SPM.
- Transaction Name. WLM uses the transaction name to find a classification rule match, depending on the following transaction name values. The transaction name is mapped to the WLM qualifier type TN.
  - APPLNAME: (Default) The application name that is specified in the client data source is used as the transaction name.
  - MODNAME: The name of the application by using the Data Driver is used as the transaction name.

– INTNAME: The application executable internal name is used as the transaction name.

Using WLM classifications

You can allow WLM to use their existing service and report classes instead of using the hard-coded IBM® DB2 QMF® Data Service definitions.

Procedure

1. Set the following parameters that are located in the CQDSIN00 configuration file. Set the values of the WLMUSERID and WLMTRANNAME parameters to names already in your policy so that IBM® DB2 QMF® Data Service is correctly classified.

   ```
   if 1 = 1 then
     do
       “MODIFY PARM NAME(WLMUSERID) VALUE(CQDS)”
       “MODIFY PARM NAME(WLMTRANNAME) VALUE(APPLNAME)”
   ```

2. If your server configuration member, CQDSIN00, does not match your existing WLM definitions, add the following parameter to your CQDSIN00 member, and keep the default value NO:

   ```
   if 1 = 1 then
     do
       “MODIFY PARM NAME(WLMFORCEPOLICY) VALUE(NO)”
   ```

   Note: If you set WLMFORCEPOLICY to NO, and the service class and report class descriptions are not correct, the zIIP offload criteria is unavailable and the default value of 100% is used for all IBM® DB2 QMF® Data Service enclaves. The service and report classes to which reference is made are those set (or defaulted) in the server configuration member, CQDSIN00, for the WLMPnREPORTCLASS and WLM*SERVICECLASS parameters.

Activating the WLM service policy

About this task

⚠ **Warning:** If you change a required element to an invalid value or remove a required definition and activate a service policy, all active servers that require that definition are shut down.

Procedure

1. After you edit the service definition, select **Utilities** from the **WLM Definition** panel (see “Using the WLM Administration Tool”).
2. From the **Utilities** menu, select the **Install Definition** option to save the updated service definition.
3. Use the **Activate Service Policy** option to activate a service policy.

Verifying WLM classification

Procedure

1. Make sure the following started task parameter is added to the CQDSIN00 configuration member:

   ```
   "MODIFY PARM NAME(TRACEWLMCALLS)   VALUE(YES)"
   ```

   This activates tracing for Data Service server calls made to the WLM APIs for transaction management.
2. Connect with your application, and run a transaction.
3. Go to the **Data Service server Primary Option** menu, and select the **Trace Browse** option. Press Enter.
4. The system displays a panel that shows the trace (trace lines are wrapped for the purposes of easier viewing).

The panel shows an ODBC connection that is created from Data Service Studio to a Data Service server, and an update that is sent to a DB2 table by using this connection. The CQDSIN00 member contains the following command:

```
"MODIFY PARM NAME(WLMCLASSTRAN)VALUE(YES)"
```

The following classification rule was added to the default rules installed by the Data Service server for subsystem SDB.

```
____ 1 TN * ___ SDH_SCTX ________
```

The Trace Browse shows the following WLM operations that occurred:

- WLM enclave join executed. The LDU for the new connection thread is joined to the long running CQD_SCHI enclave to initialize the thread.
- WLM classify work executed. The LDU for the new connection thread is classified to the CQD_SCTX service class.
- WLM enclave create executed. An enclave is created using the CQD_SCTX service class for the new connection thread.
- WLM offload CPU time executed. This shows the call to WLM with the criteria for offloading this enclaves SRB work to the zIIP. The durations and percentages for the offloading are obtained from the SDB_RPCn report class definitions.
- WLM enclave leave executed. The LDU leaves the CQD_SCHI enclave that it joined to initialize the thread.
- WLM enclave join executed. The LDU is joined to the CQD_SCTX enclave that was created for it in a preceding step. This is where the actual transaction work is done. In this case, an update is made to the DB2 table USERID.STAFF.
- WLM enclave leave executed. Processing of the DB2 update is complete, and the LDU leaves the CQD_SCTX enclave.
- WLM enclave join executed. The LDU rejoins the CQD_SCHI enclave for thread termination.
- WLM enclave delete executed. The CQD_SCTX enclave is deleted.

WLM Health Reporting

Data Service server reports to WLM on its relative "health" by issuing the IWM4HLTH macro with a health indicator between 0% and 100%. Data Service server starts with a health indicator of 100%. This reporting is enabled by using the WLMHEALTHREPORT parameter, which by default is set to YES.

Periodically, Data Service server examines indicators and adjusts its health percentage. If failures, such as ACI timeouts and ACI abends, occur, the health percentage is adjusted down. The higher the failure rate, the larger the adjustment. If no failures occur, the health percentage is adjusted up. To set the interval for this parameter, use the WLMHEALTHINTERVAL.

Configure WLM health reporting by using the following parameters in the CQDSIN00 member.

Parameter	Description	Valid values
CONCURRENTMX	The maximum number of concurrent sessions, which may be open with the server. This limit is enforced such that new connection requests are rejected if the total number of active sessions would exceed this limit. Setting this limit to zero causes all new connections to be rejected, while allowing in-flight sessions to remain active.	2000 (default)

Parameter	Description	Valid values
WLMHEALTHINTERVAL	Controls how often health statistics are reported to WLM. Interval is in seconds.	60 (default)
WLMHEALTHREPORT (*optional*)	Controls whether the Data Service server reports its health percentage to WLM.	**YES** (default) Data Service server uses the current rates of ACI timeouts and ABENDS to compute a change in the health percentage reported to WLM. **NO**
WLMMAXHEALTH (*optional*)	Controls the current health value reported to WLM.	0 – 100 (default value is 100)

You can examine the current level of health by looking at the value for WLMHEALTH by selecting **CQD Admin** > **CQD Parms** > **PRODWLM** from the **IBM® DB2 QMF® Data Service Server — Primary Option** menu.

You can change the current health value by using the WLMMAXHEALTH parameter.

This parameter allows the CONCURRENTMX parameter to work with the SHAREPORTWLM parameter. Setting CONCURRENTMX parameter to zero forces WLMMAXHEALTH to zero. Setting CONCURRENTMX from zero to nonzero forces WLMMAXHEALTH to 100.

Block fetch

Block fetch pre-extracts rows and sends them in blocks to the requesting node. this process improves the performance of most queries by minimizing network traffic and by using data that is already on the node to accommodate subsequent queries.

Data Service server only uses block fetch with read-only queries. This type of query occurs in the following situations:

- The SELECT statement has a FOR FETCH ONLY clause.
- The SELECT statement has an ORDER BY clause.
- The SELECT statement's first FROM clause contains more than one table (or view).
- The SELECT statement has the UNION or UNION ALL operator.
- The SELECT statement has the DISTINCT keyword in the first SELECT clause.
- The SELECT statement has a column function in the first SELECT clause.
- The SELECT statement has a HAVING clause in the outside SELECT statement.
- The SELECT statement has a GROUP BY clause in the outside SELECT statement.
- The SELECT statement contains a subquery where the base object of the SELECT statement and the subquery is the same table.

By default, blocks hold 256 KB of data. This number is set by the Data Service server NETWORKBUFFERSIZE parameter. The number of blocks that are used is set by the Data Service server PREFETCH parameter. If Data Service server evaluates a query and determines that it is eligible for block fetch, it begins fetching rows into the prefetch buffers; however, no transmission of data takes place until the first (real) FETCH statement reaches the server.

Note: The maximum number of bytes that is sent for each transmission (for each VTAM SEND) is limited to 32 KB, although Data Service server's internal prefetch buffers can be larger.

Use block fetch to improve the performance of queries that process many rows in a table.

Note: Using block fetch with a query in which no DESCRIBE (or PREPARE INTO) is performed in advance of fetching rows can degrade performance. Data Service server must internally perform a DESCRIBE to determine the types of data that may be returned.

In addition, depending on the type of isolation level that is used, remember the following considerations:

- If the plan is bound with the Repeatable Read (RR) option and block fetch is used, many more pages can be locked for update than without block fetch, especially if the number of rows that are normally extracted by the query is small.
- If the plan is bound with the Cursor Stability (CS) option and block fetch is used, data changes can take place between the time the data is extracted and the time that it is used by the application.

Enabling block fetch

Using block fetch improves performance of certain types of SQL queries by asynchronously pre-extracting rows (on the server node) ahead of the current row. The pre-extracted rows are then sent back to the requesting node in blocks that contain multiple rows of data.

Procedure

To enable block fetch, use the `MODIFY PARM` command to add the following parameter to the `CQDSIN00` configuration member:

```
“MODIFY PARM NAME(PREFETCH) VALUE(3 BLOCKS)”
```

Parameter	Description	Valid values
PREFETCH	Controls how many blocks of rows should be fetched from DB2. These blocks of rows are used to build the compressed row buffers that are sent to an ODBC application from the server. This value should only be changed if the buffers that are being transmitted from the server to an ODBC client application are not full. **Note:** This parameter value should be changed only when recommended by technical support.	3 (default)

Configuring DB2 for z/OS Continuous Block Fetch

You can configure support for DB2 for z/OS Continuous Block Fetch (CBF) using DRDA for high performance.

About this task

This task applies only to IBM DB2 for z/OS using DRDA.

Procedure

1. Configure the CQDSIN00 member.
   a) Set the DRDA configuration for DB2.

b) In the DRDA Define, the default for the QRBLKSZ parameter is set to 128K. Modify QRBLKSZ if a larger block if needed. Recommendation is to keep the default. As an example, to add 512K:

```
"DEFINE DATABASE TYPE(MEMBER)"              ,
           "NAME(DB3A)"                      ,
           "LOCATION(ZOS3DB3A)"              ,
           "DDFSTATUS(ENABLE)"               ,
           "DOMAIN(MYHOST)"        ,
           "PORT(3740)"                      ,
           "CCSID(37)"                       ,
           "QRBLKSZ(524288)                  ,
           "IDLETIME(160)"
```

2. Add the DRDAMAXBLKEXT parameter. Start with value 8:

```
"MODIFY PARM NAME(DRDAMAXBLKEXT)        VALUE(8)"
```

3. In the SQL query, estimate the number of rows in the RESULT SET and use it in the SQL query as follows:

   a) Assuming the SQL query is `SELECT * FROM CBFTABLE`, and there are 50000000 rows.

   b) Append the following to the end: `OPTIMIZE FOR 50000000 ROWS FOR FETCH ONLY`

   For example:

```
SELECT * FROM CBFTABLE OPTIMIZE FOR 50000000 ROWS FOR FETCH ONLY
```

4. To verify the functionality, turn on the following `TRACE BROWSE` parameter:

   a) `TRACE DRDA CODEPOINT READ/WRITE/FLOW YES`

   b) `TRACE DRDA CODEPOINT WRITE BUFFER YES`

   The trace should look like the following example. The number of corresponding "CodePoint(READ)" equates to the value of DRDAMAXBLKEXT set in the CQDSIN00:

```
18:56:43 0301869847                LEN=02A7,CPT=241B,ELEN=00
18:56:43 0301869848 DSNHLI INTERNAL     OPEN-CURSOR - DSNT400I
                                   SQLCODE = 000,  SUCC
18:56:43 0301869849                LEN=0221,CPT=241B,ELEN=00
18:56:43 0301869850 DSNHLI BLOCK FETCH  (41490) - RC 0 REASON
                                   00000000 SQLCODE 0
18:56:44 0301869851                LEN=01A0,CPT=241B,ELEN=00
18:56:44 0301869852                LEN=007C,CPT=241B,ELEN=00
```

MapReduce

This section provides information on MapReduce features for performance enhancement.

You should also refer to the *IBM® DB2 QMF® Data Service User's Guide* for additional information on using MapReduce features.

Virtual Parallel Data

Virtual Parallel Data (VPD) allows you to group multiple simultaneous requests against the same data source and run them in parallel, while doing the input and output (I/O) only once. VPD also allows single or multiple requests to run with asymmetrical parallelism, separately tuning the number of I/O threads and the number of client or SQL engine threads.

To use this feature you must provide a VPD group name when submitting request(s). All requests submitted to the same Data Service server with the same group name within a time period will be placed into a VPD group. One or more I/O threads will be started to read the data source and write it to a wrapping buffer. Group members will share the data in the buffer(s), without having to read the data source directly.

A group is created when the first member request arrives. The group is closed either when all members (and all their parallel MRC threads) have joined, or when a timeout has expired. The I/O threads are started as soon as the group is created, and data begins to flow to the buffer. If the buffer fills before the group is closed, the I/O thread(s) will wait. Once the group is closed and active members begin consuming data, the buffer space is reclaimed and I/O continues.

VPD supports MapReduce Client (MRC), and group members can use different levels of MRC parallelism. For example, a single VPD group might have six members, three members using 5 MRC threads, and the other three using 9 MRC threads. The group will consist of six members and 42 client threads. The number of I/O threads is determined separately. VPD supports a group of a single member, thus supporting asymmetrical parallelism for single requests when using MRC.

VPD is currently supported for the following data sources:

- Adabas files
- Physical sequential data sets on disk, tape, or virtual tape
- Log streams
- IBM MQ
- VSAM KSDS, RRDS, and ESDS files
- IAM files
- zFS/HFS files

Configuring Virtual Parallel Data

To configure Virtual Parallel Data, specify a group name and appropriate parameters.

Procedure

1. Configure the following parameters in the CQDSIN00 member:

```
/------------------------------------------------------------------/
/* Enable Virtual Parallel Data for asymmetrical parallelism      */
/------------------------------------------------------------------/
if DoThis then
  do
  "MODIFY PARM NAME(VPDGROUPTIMEOUT) VALUE(60)"
  "MODIFY PARM NAME(VPDBUFFERSIZE) VALUE(40)"
  "MODIFY PARM NAME(VPDTRACEDB) VALUE(NO)"
```

The following table lists the VPD parameters:

Parameter	Description	Valid values
VPDBUFFERSIZE	Specifies the default buffer size, in megabytes above the bar, for a Virtual Parallel Data buffer.	Numeric value in megabytes. Default is 40.
VPDGROUPTIMEOUT	Specifies the maximum time, in seconds, from the time a group is formed until it is closed. Default: 60 seconds	Numeric value in seconds. Default is 60.
VPDTRACEDB	Controls whether Virtual Parallel Data processing will trace debugging messages.	**NO** Do not trace debugging messages (default). **YES** Trace debugging messages.

Parameter	Description	Valid values
VPDTRACEREC	Causes Virtual Parallel Data to trace at the record level. (*Optional*) **Note:** Setting this to YES will produce a large amount of trace output.	**NO** Do not trace record level messages (default). **YES** Trace record level messages.

2. Supply the group name.
3. Optional: Specify the number of members in the group. Although optional, this parameter is recommended.

   When this parameter is provided, the group is closed as soon as all members have joined. If the number is not provided, the group is not closed until the timeout expires. There is no default.
4. Optional: Specify a timeout value for the group formation.

   When the first group member request arrives at the server, the timer is started. If the group remains open when the request expires, it is closed. Any members/threads arriving after the timeout will be placed in a new group. The default is 60 seconds, and can be overridden in the CQDSIN00 file.
5. Optional: Specify the number of I/O threads to use when reading the data source. If this value is not provided, the number of threads is determined as follows:

   a) If the data source is a tape data set and the number of volumes can be determined, the same number of I/O threads will be started.

   b) Otherwise, if a Map Reduce thread count is provided in the data map, that number is used.

   c) Otherwise, if a value is configured for ACIMAPREDUCETASKS in the CQDSIN00 configuration member, that number is used.

   d) Otherwise, a single I/O thread will be started.

Innovation Access Method (IAM)

Innovation Access Method (IAM) is a VSAM optimization product distributed by Innovation Data Processing. Enable MapReduce for IAM by setting the MAPREDUCEIAMKEYMOD parameter to YES.

MapReduce is implemented by analyzing the file to be retrieved and dividing it up into parts for simultaneous parallel retrieval. For VSAM, this is done by referencing information kept by VSAM about a file. This is supported for key-sequenced data sets (KSDS), entry-sequenced data sets (ESDS), and relative record data set (RRDS) VSAM files. For sequential files, this is done by analyzing information about the extents and volumes of the file. However, for IAM a different approach must be taken because there is no information about the internal structure of an IAM file.

To implement MapReduce for IAM, contact Innovation Data Processing and request module IAMRKTEX. This module will perform the analysis of the internal structure of the IAM file and allow implementation of MapReduce technology. This module will be provided free of charge on request to Innovation Data Processing.

Configuring MapReduce for IAM

Enable MapReduce for IAM by configuring the Data Service server.

Before you begin

The Data Service server must already be installed.

About this task

To enable MapReduce for IAM, you must configure the server configuration file.

Procedure

1. Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Customization Guide*.
2. Locate the parameter `MAPREDUCEIAMKEYMOD`.
3. Use the **`MODIFY PARM`** command to change the MAPREDUCEIAMKEYMOD parameter value, as follows:

   ```
   "MODIFY PARM NAME(MAPREDUCEIAMKEYMOD) VALUE(YES)"
   ```

Metadata repository

The metadata repository for MapReduce stores statistics about virtual tables that are used to enhance performance in conjunction with MapReduce and parallelism. This support applies to DRDA and IMS data sources, including those accessed via the IBM Federated Server (such as Terradata and Sybase), as well as data sources accessed via direct DRDA support (DB2 LUW and Oracle) provided by the Data Service server. The gathered metadata persists across server restarts.

Populating the metadata repository

You can periodically run the **DRDARange** or **IMSRange** command to gather metadata repository information about the backend virtual tables.

About this task

You can run the metadata repository command for DRDA or IMS either using the ISPF panels or a batch job.

Note: When using MapReduce support, **DRDARange** is required for a relational database management system (RDBMS).

The following restrictions and considerations apply when using this feature:

- Current support does not contain any optimizer enhancements for processing complex queries or joins other than what may be used to enhance MapReduce.
- If a table does not contain enough rows to properly calculate a DRDA Range, then the following error is also returned for this condition:

  ```
  Table <schema>.<table_ name> not eligible for range processing
  ```

  An additional error message can be found in the tracebrowse for this error. For example:

  ```
  22:10:53 Row count 14 too small for range processing
  22:10:53 SELECT DRDARANGE('virtual_table.DBLIDX') FOR FETCH ONLY - SQLCODE 0
  22:10:53 SQL ENGINE HPO OPEN-CURSOR - SQLCODE 0
  22:10:53 SQL ENGINE HPO FETCH - SQLCODE 100
  ```

Procedure

Run the appropriate command as follows:

- Using the ISPF panels:
  - For DRDA data sources, use the SELECT statement at the virtual table level.

    ```
    SELECT DRDARANGE('<TABLE NAME>',MAX_SCAN,'OPTION1','OPTION2',...);
    ```

    Note: It is recommended to use option PARTONLY for partitioned tables. Using this option will force the use of partition boundaries when determining parallelism.
  - For the IMS data source, use the SELECT statement at the database level.

```
SELECT IMSRANGE('IMS database name')
```

- Using a batch job, which you can use to schedule the commands to refresh the statistics on a specified schedule. A sample job is provided in *hlq*.SCQDCNTL(CQDRANGE). Instructions for required edits to the job are provided in the member.

```
//RANGE  EXEC  PGM=CQDXMAPD,PARM='SSID=CQDS,,MXR=30000000'
//STEPLIB  DD DISP=SHR,DSN=loadlibrary
//RPT      DD SYSOUT=*
//FMT      DD SYSOUT=*,DCB=LRECL=4096
//OUT      DD SYSOUT=*
//IN       DD *
SELECT DRDARANGE('<TABLE NAME>',MAX_SCAN,'OPTION1','OPTION2',...);
SELECT IMSRANGE('<IMS DBD Name>');
```

Modifying the data and index buffer values for VSAM files

You can change the data and index buffer values for VSAM files.

About this task

You can control the data buffer (BUFND) and the index buffer (BUFNI) values for VSAM files either globally or for individual requests, as follows:

- To change the values globally, you must add the required parameters to your Data Service server configuration file. The following table lists these parameters:

Parameter	Description	Valid values
SQLENGVSAMDATABUFF	Specifies the number of data buffers for VSAM files. Default: 20	Numeric value.
SQLENGVSAMINDEXBUFF	Specifies the number of index buffer for VSAM files. Default: 30	Numeric value.

- To change the values for individual requests, you can use virtual table (VTB) rules. Sample VTB rules CQDBUFND and CQDBUFNI are provided.

  To override your index buffer or data buffer values, you must enable the respective rule and use the appropriate BUF prefix for table names in your SQL statement, as follows.

  – **To override the data buffer (BUFND) value:**

    Use sample rule CQDBUFND. The CQDBUFND rule is invoked every time a table with the prefix BUFND_ is found in the SQL statement. The following format is expected:

    ```
    BUFND_nn_virtualtablename
    ```

    Where:

    - *nn* is the number of data buffers (BUFND) for the VSAM data sets
    - *virtualtablename* is the name of the virtual table

    For example:

    ```
    SELECT * from BUFND_30_STAFF_VSAM ;
    ```

    The following message is displayed in the Server Trace:

    ```
    CQD1000I VTB.OPTBVSND set to 30
    ```

  – **To override the index buffer (BUFNI) value:**

Use sample rule CQDBUFNI. The CQDBUFNI rule is invoked every time a table with the prefix `BUFNI_` is found in the SQL statement. The following format is expected:

```
BUFNI_nn_virtualtablename
```

Where:

- *nn* is the number of index buffers (BUFNI) for the VSAM data sets
- *virtualtablename* is the name of the virtual table

For example:

```
SELECT * from BUFNI_30_STAFF_VSAM ;
```

The following message is displayed in the Server Trace:

```
CQD1000I VTB.OPTBVSNI set to 30
```

Procedure

1. To change the values globally, perform the following steps:

   a) Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Customization Guide*.

   b) Add the following statements to your CQDSIN00 member:

   ```
   "MODIFY PARM NAME(SQLENGVSAMDATABUFF) VALUE(20)"
   "MODIFY PARM NAME(SQLENGVSAMINDEXBUFF) VALUE(30)"
   ```

2. To change the values for individual requests, perform the following steps:

   a) Customize the server configuration member (CQDSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

   ```
   "MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES)"
   ```

   b) Access the VTB rules, as follows:

      i) In the IBM DB2 QMF Data Service - Primary Option Menu, specify option E, **Rules Mgmt**.

      ii) Specify option 2, **SEF Rule Management**.

      iii) Enter VTB for **Display Only the Ruleset Named**.

   c) Enable each rule as follows:

   - Specify E next to CQDBUFND and press Enter.
   - Specify E next to CQDBUFNI and press Enter.

   d) Set each rule to Auto-enable as follows:

   - Specify A next to CQDBUFND and press Enter.
   - Specify A next to CQDBUFNI and press Enter.

   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

   e) Use the appropriate BUF prefix for table names in your SQL statement.

Modifying the client auxiliary storage cut-off parameter

You can specify at what point the Data Service server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility.

About this task

The Data Service server listens for ENF 55 auxiliary storage shortage signals and throttles storage utilization when an auxiliary storage shortage is signaled.

The Accelerator Loader server will perform the following actions depending on the received ENF 55 signal:

- When signal ENF55QLF_AUX_WARNING is received:
  1. Issue the following message:

     ```
     CQD4265W Data Server Client buffer expansion disabled due to auxiliary storage
     warning
     ```

  2. Disable Data Service server buffer expansion for two hours and ten minutes.
  3. Issue the following message:

     ```
     CQD4266I Data Server Client services resumed
     ```

- When signal ENF55QLF_AUX_SHORTAGE is received:
  1. Disable Data Service server buffer expansion.
  2. Issue the following message:

     ```
     CQD4265W Data Server Client buffer expansion disabled due to auxiliary storage
     shortage
     ```

- When signal ENF55QLF_AUX_CRITICAL_SHORTAGE is received:
  1. Disable Data Service server buffer expansion.
  2. Issue the following message:

     ```
     CQD4265W Data Server Client buffer expansion disabled due to auxiliary storage
     critical shortage
     ```

  3. Disable new Data Service server requests.
  4. Issue the following message:

     ```
     CQD4267W Data Server Client refusing new requests due to critical auxiliary
     storage shortage.
     ```

- When signal ENF55QLF_AUX_SHORTAGE_RELIEVED is received:
  - Re-enable all Data Service server functions.
  - Issue the following message:

    ```
    CQD4266I Data Server Client services resumed.
    ```

The point at which the Data Service server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility is controlled by the **DSCLIENTAUXSTGCUTOFF** parameter.

To change the value, complete the following steps.

Procedure

1. Locate the server configuration member. The server initialization member is shipped in data set member *hlq*.SCQDEXEC(CQDSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Customization Guide*.
2. Use the **MODIFY PARM** command to change the **DSCLIENTAUXSTGCUTOFF** parameter value:

```
"MODIFY PARM NAME(DSCLIENTAUXSTGCUTOFF) VALUE(WARNING)"
```

Parameter name	Parameter description	Default value
DSCLIENTAUXSTGCUTOFF	DSCLIENT AUX STORAGE NEW CONNECTION CUTOFF Specifies at what point the Data Service server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility. **WARNING** New Data Service server connections will be rejected when an auxiliary storage warning is received. This signal is issued when message IRA205I occurs. **SHORTAGE** New Data Service server connections will be rejected when an auxiliary storage shortage is signaled. This signal is issued when message IRA200E occurs. **CRITICAL** New Data Service server connections will not be rejected until an auxiliary storage critical shortage is signaled. This signal is issued when message IRA201E occurs.	WARNING

Virtual table SAF security

A single Data Service server environment can provide data virtualization to multiple independent tenants or application groups. The virtual table SAF (system authorization facility) security feature provides a SAF mechanism to secure virtual tables so that each tenant can only access tables authorized for members of the tenant group.

Activating this security feature will prevent using virtual table names in metadata queries (such as, **SQLENG.TABLES**, **SQLENG.COLUMNS**), as well as querying or updating application data mapped using unauthorized table names.

Server interface parameter

The SQLVTRESOURCETYPE parameter in the PRODSECURITY parameter group defines a security class name for virtual table resource checking. By default, this system parameter defaults to the value 'NON' indicating that security checking is disabled.

When activated with a class name, the SQLVTRESOURCETYPE parameter will enable SAF resource checking on metadata queries (such as, **SQLENG.TABLES**, **SQLENG.COLUMNS**) as well as virtual table queries using the resource name *resource_class.table_owner.table_name* where:

- *resource_class* is the class name define for the RESOURCETYPE parameter in the PRODSECURITY parameter group (for example, RCQD)
- *table_owner* is the SQL TABLE OWNER NAME (SQLENGTABLEOWNER) as defined in the PRODSQL parameter group (for example: 'DVSQL')
- *table_name* is the map (or virtual table) name as defined in the map data set

For improved performance in SAF calls, RACROUTE REQUEST=FASTAUTH provides general resource checking. A separate INTRNLONLY parameter named 'DISABLE FASTAUTH SECURITY CHECKS' disables use of FASTAUTH if security problems are encountered. Disabling FASTAUTH will switch to RACROUTE REQUEST=AUTH checking on all resource rules which can degrade query performance on metadata tables.

When securing metadata tables, READ access is required to query rows in the following tables.

- SQLENG.COLUMNS
- SQLENG.COLUMNPRIVS
- SQLENG.ERRORMSGS
- SQLENG.FOREIGNKEYS
- SQLENG.PRIMARYKEYS
- SQLENG.ROUTINES
- SQLENG.SPECIALCOLS
- SQLENG.STATISTICS
- SQLENG.TABLES
- SQLENG.TABLEPRIVS

Securing tables using the generic profile SQLENG.* is also an option if preferred.

Securing specific virtual tables is also required when activating this feature. Securing virtual tables by specific or generic rules activates two security checks:

1. When querying metadata tables (SQLENG.*), users must minimally have READ access to the virtual tables in order for rows related to a table to be returned. In this case, there are no errors returned. Instead, the information about a specific table is omitted from the result set and the user has no indication that the table exists.
2. When querying virtual tables, the user must have READ access to each table in the SQL SELECT statement and UPDATE access to any table that is the target of an SQL INSERT, UPDATE, or DELETE statement.

Restrictions and Considerations

Virtual table authorization checking is built on general resource checking and is impacted by the following product parameter in the PRODSECURITY group:

- ALLOWUNPROT – The ALLOWUNPROT parameter allows access to unprotected resources. When set to YES, this parameter allows access to resource names that have no matching resource definition in the SAF database. ALLOWUNPROT should be set to NO to insure resource rules are correctly processed.

  Note: ALLOWUNPROT=NO will automatically activate numerous resource checks unrelated to this feature.

The *table_owner.map_name* resource name is internally restricted to 44 bytes. While internal map names larger than 44 bytes are still allowed, resource checking will only pass the first 44 bytes of the *table_owner.map_name* string in the SAF call for validation. Generic resource rules will be necessary if map names exceed this limitation.

Because all maps are limited to a single table owner as defined in the `SQLENGTABLEOWNER` system parameter, users should consider a standard prefix for all map names they want to secure for application groups. This simple generic resource rules can be defined to protect these names. For example, if the `SQLENGTABLEOWNER` is configured as `'DVSQL'` and an application group uses AG01 as a prefix on all table names, a generic resource `'DVSQL.AG01*'` will control access to all tables starting with AG01 as a map name.

All SQL queries are automatically secured when this feature is activated. This means that resource rules must exist to allow READ access to the metadata tables SQLENG.*.

This feature is limited to SQL access to virtual tables. Users authorized to create tables can create tables which may not be accessible due to SQL access rules implemented using this feature.

Migrating maps

Use the Map Migration utility to move your virtual table maps from a development environment to a test or production environment or from one release to another.

Before you begin

Before using the Map Migration utility, make sure that the following prerequisites have been met:

- **Data Service Studio requirements**

  If migrating DB2 virtual tables, target systems used by each table must be defined in the target server using one of the following definitions:

  - If you want to use the same target system name, define the target system name on the target server.
  - If you want to use a different target system name, then define the new target system name, and use the `TSYS=OLD_TSYS,NEW_TSYS` parameter in the CQDGNMPM batch migration utility.

- **Data Service server requirements**

  Make sure that both the origin and destination servers have been started.

- **Data Service server security requirements**

  The following table summarizes the security permissions required to use the migration utility:

Table 11. Security permissions required to use the migration utility

	JCL library	**Map export PDS**	**Server map data set**
	The location where the JCL resides.	The PDS library to which the exported metadata objects are unloaded.	The `CQDMAPP DD` data set, which must be the first data set in the concatenation if the parameter `NEW MAP DSN` is not set.
Batch user ID	UPDATE	CREATE READ	N/A
Server user ID	N/A	UPDATE	UPDATE READ

About this task

The Map Migration utility facilitates change control of the virtual table maps. Change control is the process of moving the virtual table maps defined in a development environment to a test or production environment or from one release to another.

You can use the CQDGNMPM member located in your *hlq*.SCQDCNTL data set for migrating virtual table maps. See the CQDGNMPM member for a list of parameters available for use when migrating virtual table maps.

You can use the CQDGNMPM member to perform the following tasks:

- Migrate one or multiple virtual table maps from one server to another.
- Change the virtual table map definition using the optional parameters. See the comments in the sample job for more details.

Procedure

1. Customize the migration utility job, CQDGNMPM, for the requirements at your site.
2. Submit the CQDGNMPM batch job. Utility job CQDGNMPM extracts the contents of the maps, stores the metadata objects in the map export PDS library, and creates the batch job that is used to rebuild the maps on the target server.
3. Submit the batch JCL that was created in the previous step to rebuild the maps on the target server.

Results

The utility extracts the content of the map export PDS and rebuilds the map on the target server.

Notices

This information was developed for products and services offered in the US. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below.

© (*your company name*) (*year*).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. (*enter the year or years*).

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at http://www.ibm.com/legal/copytrade.shtml.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM®.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Index

Product Number: 5698-DB2
5698-DB2
5698-QMF

SC28-3266-00