IBM Data Virtualization Manager for z/OS
Version 1 Release 1

Installation and Customization Guide

GC27-8874-00

Contents

Tables

About this information

This information supports IBM Data Virtualization Manager for z/OS (5698-DVM) and contains information about installing and configuring Data Virtualization Manager.

Purpose of this information

This information describes how to prepare for installation, install, customize, and verify IBM Data Virtualization Manager for z/OS in your environment.

Who should read this information

This information is intended for z/OS system programmers and system administrators who are responsible for installing and customizing IBM Data Virtualization Manager for z/OS. The customization information is also of interest to application developers who want to understand how various customization and tuning actions might affect the performance of their applications.

How to send your comments to IBM

We appreciate your input on this documentation. Please provide us with any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

Important: If your comment regards a technical problem, see instead "If you have a technical problem" on page ix.

Send an email to comments@us.ibm.com.

Include the following information:

- Your name and address
- Your email address
- Your phone or fax number
- The publication title and order number:

  IBM Data Virtualization Manager for z/OS Installation and Customization Guide
  GC27-8874-00
- The topic and page number or URL of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM®, you grant IBM a nonexclusive right to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are listed for sending comments. Instead, take one or more of the following actions:

- Visit the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Chapter 1. Overview

Data Virtualization Manager server is a mainframe-resident server that uses the IBM System z Integrated Information Processor (zIIP) for all of its computationally intensive processing, supporting ANSI SQL-92 functions and NoSQL data sources using the MongoDB query language. Data Virtualization Manager server provides seamless, real-time access to enterprise data regardless of operating system, location, or interface.

The server supports real-time data virtualization solutions for enabling mainframe relational and non-relational data to seamlessly integrate with data analytic, Big Data, mobile, and web solutions. All underlying data structures are abstracted from the user.

Before you can use IBM Data Virtualization Manager for z/OS solutions to access your enterprise data, you must install and configure the Data Virtualization Manager server.

After the server is installed, you can install the Data Virtualization Manager studio. Using the studio, you can create virtualized tables to run SQL queries or virtualized collections to run JSON queries.

Once the Data Virtualization Manager server is installed, you can configure the solutions that you want to use.

To ensure the highest levels of performance, Data Virtualization Manager server includes several query optimization features, such as parallel input/output and MapReduce. Data Virtualization Manager server employs multiple, parallel threads to handle input request for data and output delivery of information to the client (data consumer). Once data is transformed from non-relational to a relational format, the powerful data virtualization engine continually streams and buffers data to the client.

Data Virtualization Manager server has a wide range of connectivity options for data consumers, including:

- ANSI 92-SQL (JDBC/ODBC/DRDA)
- NoSQL (JSON)

What's new in IBM Data Virtualization Manager for z/OS Installation and Customization Guide

This section describes recent technical changes to IBM Data Virtualization Manager for z/OS.

New and changed information is marked like this paragraph, with a vertical bar to the left of a change. Editorial changes that have no technical significance are not marked.

Description	**Related APARs**
The JDBC Gateway is a Data Virtualization Manager distributed application server that allows direct connectivity to JDBC data sources. See Chapter 7, “JDBC Gateway,” on page 91	PH01001
Db2 Virtualization is a new feature that provides single-point access to various data source types. See Using Db2 for z/OS to access multiple data source types.	PH03533
You can control whether native Db2 database subsystems appear in ISPF and the Data Virtualization Manager studio and if attempts to connect to native Db2 subsystems are allowed. See “Controlling display and access for native Db2 subsystems ” on page 49.	PH00606
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 59.	PI98000

Description	Related APARs
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 61.	PI97991
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 66 and "Configuring access to System Management Facility (SMF) files" on page 67.	PI97239
Db2 Direct is a new Data Virtualization Manager 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 32 and "Configuring Db2 Direct" on page 34.	PI95751
IMS Direct now supports calls to Guardium encryption and decryption exits. See "Modifying the Data Virtualization Manager configuration member for IMS Direct" on page 52.	PI94740
Data Virtualization Manager server parameter SQLENGDBCSLTFMT, which was previously used for setting the format for DBCS Latin characters, is obsolete. Use of this parameter has been removed from "Configuring support for code pages and DBCS " on page 10.	PI94474
SQL query access to DB2 unload data sets is now provided. See "Configuring access to Db2 unload data sets" on page 31.	PI94369
SQL access to IBM MQ is now provided. See "Configuring access to IBM MQ" on page 56.	PI92252
Delimited data can now be used with virtual tables. See "Configuring delimited data support" on page 14.	PI92252
IMS Direct now supports access to multiple IMS subsystems. See "Modifying the Data Virtualization Manager configuration member for IMS Direct" on page 52.	PI90971
Using a virtual table rule, you can read a subset of a generation data group. See "Configuring generation data set retrieval" on page 13.	PI90302
You must APF-authorize the LOAD library AVZ.SAVZRPC. See "APF-authorizing LOAD library data sets" on page 10.	PI90301
A checklist for customizing the Data Virtualization Manager server is added. See "Preparing to customize" on page 7.	

System requirements

You need to consider the system requirements before you install Data Virtualization Manager server.

IBM hardware / software minimum requirements

IBM zEnterprise 114 (z114), IBM zEnterprise 196 (z196), or more recent IBM zEnterprise system.

IBM z/OS version 1.13 or later.

Load library

The load library for Data Virtualization Manager server version 1.1 is allocated as a partitioned data set extended (PDSE). When a PDSE is used to store load modules, it stores them in structures called program objects.

You cannot copy a PDSE to a partitioned data set (PDS). For more information, see the IBM z/OS documentation.

Program Temporary Fixes (PTFs)

When you use VSAM data access with Data Virtualization Manager server version 1.1, you need to add the following PTFs:

- RLS OA44111:
  - z/OS 1.13 UA75045
  - z/OS 2.1 UA75046
- VSAM:
  - z/OS 1.13 UA75272
  - z/OS 2.1 UA75273

Chapter 2. Installing the Data Virtualization Manager server

Install the Data Virtualization Manager server using IBM SMP/E for z/OS.

About this task

Use the information in *Program Directory for IBM Data Virtualization Manager for z/OS* to install the Data Virtualization Manager server on your system.

Installing server maintenance

To install server maintenance, use the IBM SMP/E for z/OS program. Download the latest PTFs to the z/OS system where you want to apply the PTFs.

Chapter 3. Customizing the Data Virtualization Manager server

After you install IBM Data Virtualization Manager for z/OS using SMP/E, customize the server for use.

Before you begin
You must install Data Virtualization Manager and apply all available maintenance before customizing the server. To apply server maintenance, you should acquire available PTFs and apply them to the server so you will have the most current available code for your installation.

Preparing to customize

Before you start to customize IBM Data Virtualization Manager for z/OS, familiarize yourself with the customization tasks.

The following table describes each significant customization task. Use this checklist to guide you through the customization process.

Table 1. Customization checklist

Step	Task description	For more information
1	Review the required naming conventions that must be followed when configuring the server subsystem ID and the server initialization member.	See "Required naming conventions" on page 8.
2	Create the server data sets using the *hlq*.SAVZCNTL members AVZDFDIV, AVZGNMP1 and AVZEXSWI.	See "Creating server data sets" on page 8.
3	Set up the security application to use with the server using one of the following *hlq*.SAVZCNTL members: AVZRAVDB, AVZA2VDB, AVZTSVDB.	See "Defining security authorizations" on page 9.
4	Configure Workload Manager (WLM) for optimum performance of the server.	See "Configuring Workload Manager (WLM)" on page 9.
5	APF-authorize the product LOAD library data sets.	See "APF-authorizing LOAD library data sets" on page 10.
6	Create a copy of the product libraries (optional).	See "Copying target libraries" on page 10.
7	Configure the server to support DBCS (optional).	See "Configuring support for code pages and DBCS " on page 10.
	Customize the server to access your data sources in *hlq*.SAVZEXEC(AVZSIN00).	See "Customizing the server initialization member" on page 11. Then, see Chapter 4, "Configuring access to data sources ," on page 19 for the specific types of data sources the server should access.
	Configure the started task JCL located in *hlq*.SAVZCNTL(AVZ1PROC) before you can start the server.	See "Configuring the started task JCL" on page 12.

Table 1. Customization checklist (continued)

Step	Task description	For more information
	Configure the CLIST that invokes the ISPF panels by using *hlq*.SAVZEXEC(AVZ).	See "Configuring the ISPF application" on page 13.
	Verify the installation by creating a virtual table and accessing its underlying VSAM file (optional).	See "Verifying the Data Virtualization Manager server installation" on page 16.

Required naming conventions

You must follow the Data Virtualization Manager server naming conventions when configuring the server subsystem ID and the server initialization member.

The server subsystem name must follow the pattern *x*VZ*y*, where *x* is any alphabetic character A - Z and *y* is any alphanumeric character A-Z or 0-9.

Depending on what you name the server subsystem, the server initialization member must follow the same naming convention as the server subsystem name, for example, *x*VZ*y*IN00.

Note: The default server naming conventions used throughout this guide are AVZS for the server subsystem name and AVZSIN00 for the server initialization member.

Creating server data sets

The AVZDFDIV and AVZGNMP1 members of *hlq*.SAVZCNTL create data sets for the Trace Browse, the global variable checkpoint, and the data-mapping facility (DMF) that are used by the Data Virtualization Manager server. The AVZGNMP1 member also copies distributed data sets into user-modifiable data sets. The AVZEXSWI member builds the Web interface objects.

Procedure

1. Customize the AVZDFDIV member in *hlq*.SAVZCNTL to meet your requirements. The AVZDFDIV member contains comments that describe how to customize the variables.
2. Submit the AVZDFDIV member.
3. Customize the AVZGNMP1 member in *hlq*.SAVZCNTL to meet your requirements. The AVZGNMP1 member contains comments that describe how to customize the variables.
4. Submit the AVZGNMP1 member.

   Note: The map data set created in this step should be the first concatenated data set in the DD statement AVZMAPP located in the server started task. See *hlq*.SAVZCNTL(AVZ1PROC). The user and server should have read and write permissions to this data set. The system-provided data set (*hlq*.SAVZSMAP) should be the last data set in the AVZMAPP concatenation. The user and server should only have read access to the data set. The administrator will need read and write permissions.
5. Customize the AVZEXSWI member in *hlq*.SAVZCNTL to meet your requirements. The AVZEXSWI member contains comments that describe how to customize the variables.

   Note: The data set named on the RECEIVE command in the AVZEXSWI member is later used in the server initialization member AVZSIN00 for the **SWICNTLDSN** parameter definition, as follows:

   ```
   swiobj = SHLQ2||".SAVZOBJ"
   "MODIFY PARM NAME(SWICNTLDSN) VALUE("||swiobj||")"
   ```
6. Submit the AVZEXSWI member.

Defining security authorizations

To use an external security product, such as RACF, ACF2, or Top Secret, define the started task name to the security product and authorize the data set.

Procedure

To define the server and other required permissions for your security product, customize the appropriate security option located in the *hlq*.SAVZCNTL library, and submit the job:

- AVZRAVDB is for IBM Resource Access Control Facility (RACF) security.
- AVZA2VDB is for CA ACF2 (Access Control Facility) security.
- AVZTSVDB is for CA Top Secret Security (TSS).

Results

The following table summarizes the access requirements by data definition name:

Table 2. Access requirements by data definition name

Data definition name	Access	Data set name
STEPLIB	READ, EXECUTE	*hlq*.SAVZLOAD
AVZRPCLB	READ, EXECUTE	*hlq*.SAVZRPC
SYSEXEC	READ	*hlq*.SAVZEXEC
AVZTRACE	READ, WRITE	*hlq*.TRACE
AVZCHK1	READ, WRITE	*hlq*.SYSCK1
AVZMAPP	READ, WRITE	*hlq*.SAVZMAP. The user id and the server id should have the access to this dataset.

Make sure that your z/OS Security Administrator reviews the security definitions. You might need to change definitions to meet requirements at your site.

Configuring Workload Manager (WLM)

To get optimum performance from the server, define the server to WLM. The Data Virtualization Manager server should be prioritized slightly below the data provider in your WLM environment. It is not sufficient to simply add the STC to a WLM service class as the server will create independent enclaves for each connection.

About this task

The server should be configured to use a medium to high performing WLM velocity goal as its default service class.

Procedure

1. Create a WLM Classification rule.
   a) Go to the WLM ISPF application, and select option **6** (Classification Rules).
   b) Select option **1** to Create.
   c) Set the Subsystem Type to AVZ, and provide an optional description.

   d) Under the Class/Service Column next to DEFAULTS, set the desired default service class name. If a desired service class does not exist, then create one using option 4 (Service Classes) under the **Primary WLM** menu. Press enter and PF3 to save.

2. Define the Data Virtualization Manager started task AVZ1PROC to a WLM service class.

   a) Go to the WLM ISPF application, and select option 6 (Classification Rules).

   b) For the STC WLM-subsystem type, select **Modify**.

   c) Add an entry for AVZ1PROC.

   d) Add an appropriate service class for the started task and define it relative to existing workload resource management objectives.

   e) Add a unique Report class for the started task.

3. Activate the new WLM policy definition.

APF-authorizing LOAD library data sets

You must authorize for APF (Authorized Program Facility) all LOAD library data sets allocated to the Data Virtualization Manager server.

About this task

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

These LOAD library data sets are allocated to the following ddnames:

- STEPLIB

  You must authorize the LOAD library `hlq.SAVZLOAD`.

- AVZRPCLB

  You must authorize the LOAD library `hlq.SAVZRPC`.

If any data sets allocated to these ddnames are not APF-authorized, the Data Virtualization Manager server will issue the error message AVZ0051S during startup identifying the ddname and data set name of each unauthorized library. Startup processing will discontinue and the server will shut down.

Procedure

The APF authorize should be done dynamically using the SETPROG APF command, and then made permanent for the next IPL (initial program load) by updating the appropriate system PARMLIB member.

Copying target libraries

It is recommended that copies be made of the target libraries to preserve any prior customization, as applying new maintenance often replaces existing PDS members.

Configuring support for code pages and DBCS

You can configure the server to support Japanese code pages and double-byte character sets (DBCS).

About this task

To support different code pages and double-byte character sets, you must manually customize the server initialization member.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" on page 10.
2. In the member, locate the `DEFINE DATABASE` statement for your subsystem, and verify that the CCSID value is set correctly for the subsystem.
3. Locate the comment `Set CCSID for non-DB2 data`, as shown in the following example:

```
/*-------------------------------------*/
/* Set CCSID for non-DB2 data          */
/*-------------------------------------*/

if DoThis then
  do
              "MODIFY PARM NAME(SQLENGDFLTCCSID)      VALUE(1047)"
```

4. Change `DontDoThis` to `DoThis` to enable the parameters.
5. Update the following parameter:

Parameter	Description	Valid values
SQLENGDFLTCCSID	Specifies the CCSID to use for SQL engine tables. All host tables except for Db2 are assumed to be stored in this CCSID. Where possible, this CCSID should match the client CCSID used when connecting.	CCSID value Sample values: • 1047 (LATIN OPEN SYS EB) • 931 (JAPAN MIX EBCDIC) • 1390 (JAPAN MIX EBCDIC)

Customizing the server initialization member

The server initialization member `AVZSIN00` is a REXX program that you use to set product parameters and define links and databases. You must customize the server initialization member for your installation environment.

About this task

The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" on page 10.

As you go through the installation, you accept or set parameter values in the server initialization member.

If you are installing the server for the first time, it is recommended that all the default values be accepted. You can change the values as needed later.

If you are installing a new version of the server over a previous version, the previous server member might contain parameter values that you modified to meet specific requirements in your environment. In this case, you should review the initialization member for the previous version for any customizations that need to be made to the initialization member for the new version.

Procedure

1. Find the line that contains "SHLQ1" and provide your own high-level qualifier to define the ISPF data sets. For example: "SHLQ1=AVZ"
2. If you created copies of your target libraries to preserve customizations, find the line that contains "SHLQ2" and provide your own high-level qualifier to define the Event Facility (SEF) data sets. Ensure that the HLQ results in proper data set references for these features.

   For example: "SHLQ2=AVZ.AVZS". If you did not create copies of the target libraries, then "SHLQ2" should contain the same value as "SHLQ1".

3. Review the following default values for the TCP/IP parameters and change the values as necessary. The following example shows the section of the initialization member in which to make the changes:

   ```
   “MODIFY PARM NAME(OEPORTNUMBER) VALUE(1200)”
   “MODIFY PARM NAME(WSOEPORT) VALUE(1201)”
   “MODIFY PARM NAME(TRACEOERW) VALUE(YES)”
   “MODIFY PARM NAME(OEKEEPALIVETIME) VALUE(30)”
   "MODIFY PARM NAME(PARALLELIO) VALUE(YES)"
   “MODIFY PARM NAME(OEPIOPORTNUMBER) VALUE(1204)”
   ```

Configuring the started task JCL

To configure the started task JCL, modify the AVZ1PROC (subsystem default ID) member that is in the *hlq*.SAVZCNTL library.

About this task

The AVZ1PROC member contains the JCL procedure that is required to run the main address space (started task).

Procedure

1. Add the HLQ name of the libraries to the *hlq* parameter.

   This parameter sets the server data set allocations to the correct data set names.
2. Confirm that the SYSEXEC DD statement allocates the correct data set name that contains the customized server initialization member AVZSIN00. This data set was created in job AVZGNMP1 previously in the step “Creating server data sets” on page 8. The default name is *hlq*.SAVZEXEC(AVZSIN00).
3. Ensure that the DD AVZMAPP concatenation points to the *hlq*.AVZMAPP data set created in the previous installation job AVZGNMP1. This data set should be first in the concatenation and is used for storing user-defined virtual table maps. The *hlq*.AVZMAPP data set, which contains the default virtual table maps that are part of the product distribution, should be placed last.
4. The server runs as a z/OS started task. Under normal circumstances, the server starts at system startup and stops before the system shuts down. To start the server on demand, use the following console command:

   ```
   S AVZS
   ```

   where *AVZS* is the subsystem name of the server instance you defined.

   Note: If you use a procedure name other than the SSID provided in the example, then you issue the start command using that procedure name.
5. If you use an automation package to start the system, associate the **START** command with the VTAM initialization complete message (IST020I), the TCP/IP initialization complete message (EZB6473I), or both messages.
6. To verify that the startup is successful, look for the following entries in the server Job Entry Subsystem (JES) log.

   ```
   SD74391I OE stack binding port 1200 to IP address 0.0.0.0
   SD74391I OE stack binding port 1201 to IP address 0.0.0.0
   SD74391I OE stack binding port 1202 to IP address 0.0.0.0
   ```

What to do next

If you want to stop the server, issue the following console command:

P AVZS

If you issue a **CANCEL** command, all available connections terminate with an abend, and the server shuts down immediately.

Configuring the ISPF application

Configure and invoke the ISPF application.

Before you begin

The Data Virtualization Manager server must be started before you can invoke the ISPF application.

Procedure

1. Edit the *hlq*.SAVZEXEC(AVZ) member, and replace the data set name in the following statement with the data set name that you chose for the *hlq*.SAVZLOAD library:

   ```
   llib='hlq.SAVZLOAD'
   ```

2. Copy the *hlq*.SAVZEXEC(AVZ) member to a data set that is allocated to the SYSPROC allocation for all TSO users.

   Before starting the ISPF application, you must configure and start your server. See "Configuring the started task JCL" on page 12

   When the server starts, the ISPF data sets are dynamically allocated.
3. To invoke the ISPF application, go to the ISPF command shell and enter the following command: `EX ‘hlq.SAVZEXEC(AVZ)’ ‘SUB(AVZS)’`

   Where:

   - *hlq* is the high level qualifier.
   - *AVZS* is the subsystem name of the server instance you defined.

   All ISPF clients will communicate with the specified subsystem.

Configuring generation data set retrieval

You can configure the server to read only a subset of generation data sets (GDSs) by activating a VTB rule.

About this task

To read only a subset of generation data sets in a generation data group (GDG), you must enable virtual rule AVZGDGS1 and use the prefix GDG__ in your SQL statement.

A VTB rule is provided that allows a subset of the GDG to be read. VTB rule AVZGDGS1 is invoked by the SEF every time a table with the prefix GDG__ is found in the SQL statement.

The table name in the SQL statement must be of the form:

```
GDG__NumGens_RelGen_MapName
```

Where:

- GDG__ is a constant indicating a generation data set request.
- *NumGens* is a required number 0 through 999 indicating the number of generations to read.
- *RelGen* is an optional number 0 through 999 indicating the relative generation at which to start reading. A value of 0 is equivalent to a suffix of (0) in a JCL allocation; a value of 1 is equivalent to (-1), and so on.
- *MapName* is the table defined in the map data set.

For example, the following request will result in generations HLQ.GDG.STAFF(-3) through HLQ.GDG.STAFF(-6) being retrieved:

```
SELECT * FROM GDG__4_3_STAFF
```

Where the STAFF table specifies a base data set name of HLQ.GDG.STAFF. In other words, with this request, four generations will be read in descending generation order beginning with relative generation 3 (that is, generations 3, 4, 5, and 6).

Use the procedure in this task to enable sample rule AVZGDGS1.

Additional details:

When a request is made to allocate a data set, it will first be determined if the data set name represents a GDG base name. If so, a CSI lookup call will be made to return the associated GDS data set names. If a VTB rule does not specify the number of generations to read and MapReduce is disabled, or if there is a single generation, the GDG will be allocated using its base data set name, and normal system concatenation of generation data sets will occur. If MapReduce is enabled and there are multiple active generation data sets, a number of I/O processing tasks will be created. The number of I/O tasks is determined as follows:

1. If VPD is in use, the number of VPD I/O threads specified.
2. If MRC is in use, the number of active Client threads defined in the MRC request.
3. If neither VPD nor MRC is in use, the number of I/O threads will be equal to the lesser of the following:
   - The number of active generation data sets in the GDG
   - The number of generations requested by a VTB rule
   - The number of MapReduce tasks specified in the ACIMAPREDUCETASKS configuration

When the number of I/O tasks is equal to or less than the number of generation data sets, each task will read one or more complete data sets. When the number of I/O tasks exceeds the number of generation data sets, some tasks will be idle.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) 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 Data Virtualization Manager server - 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 AVZGDGS1 and pressing Enter.
4. Set the rule to Auto-enable by specifying A next to AVZGDGS1 and pressing Enter.
   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring delimited data support

To be able to process delimited data using virtual tables, you must configure a virtual table rule to activate delimited data processing and optionally define delimiter values.

About this task

Data Virtualization Manager provides the ability to process delimited data from files, MQ data, and log streams using virtual tables mapped to MQ or z/OS files. The most common form of delimited data is comma separate value files (.csv).

When delimited data processing is activated, processing occurs in column order, so the delimited data must include a value for each column in the map in the correct order to prevent errors. Data conversion errors will occur if the delimited data is not compatible with the host types of the columns. If conversion fails, diagnostic information related to the error is automatically logged for troubleshooting problems.

Delimited processing is supported through virtual table rules only. Using virtual table rule options, you can enable delimited data processing, set column and string delimiter values, and control header record processing.

A sample rule, AVZMDDLM, is provided that documents these settings. Use the following procedure to configure the sample rule.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) 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 Data Virtualization Manager server - 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 AVZMDDLM rule, as follows:
   a) Specify S next to AVZMDDLM to edit the rule.
   b) Find the **vtb.optbdlcv** variable and set to 1 to activate delimited processing for a map.
   c) Update additional rule options as needed. The following table describes the VTB rule options that support delimited data processing.

VTB variable	Description
vtb.optbdlcv	Set to 1 to activate delimited processing for a map.
vtb.optbdlco	Set the column delimiter. The default value is the comma character (,). For example, if you use the colon character (:) as the column delimiter, specify `vtb.optbdlco = ':'`.
vtb.optbdlch	Set the character field or string delimiter. The default value is the quotation mark character ("). For example, if you use the hash character (#) as the string delimiter, specify `vtb.optbdlch = '#'`.
vtb.optbdlhr	Set to 1 to identify and remove the header record containing column names. If specified without a header prefix, the system compares the first token in each line to the first column name in the table to recognize and discard the header. The default is no header checking.
vtb.optbdlhp	Define prefix data that identifies the beginning of a header line to be discarded. The specified value can contain a maximum of 32 bytes. This value is compared to the beginning of each delimited line of data before any tokenization is performed. For example, `vtb.optbdlhp = '"NAME","ADDRESS"'`. **Note:** If an `optbdlhp` value is defined, it supersedes any `optbdlhr` setting and the `optbdlhr` value is ignored.

   d) Save your changes and exit the editor.
4. Enable the rule by specifying E next to AVZMDDLM and pressing Enter.

5. Set the rule to Auto-enable by specifying A next to AVZMDDLM and pressing Enter.

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

Configuring ALTSTARTEDTASKID

About this task

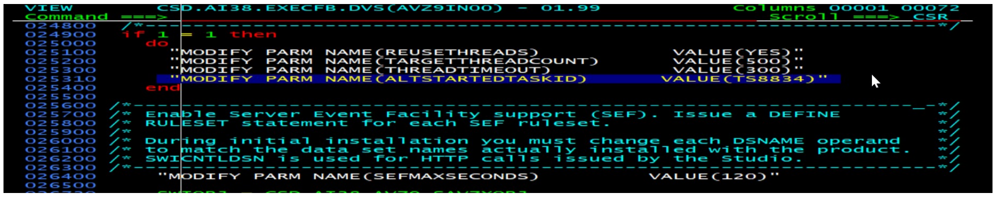
Configure the parameter ALTSTARTEDTASKID to be used by the Data Virtualization Manager server to build or update the metadata cache. When this USER ID is configured, the server will always use this USER ID during startup for any metadata cache refresh queries. Normal user queries that use DRDA will use the primary user id and password.

Procedure

1. Add the following line to the IN00 file.

```
MODIFY PARM NAME(ALTSTARTEDTASKID)              VALUE(ID)
```

   In the above line, *ID* is the alternate started task id.

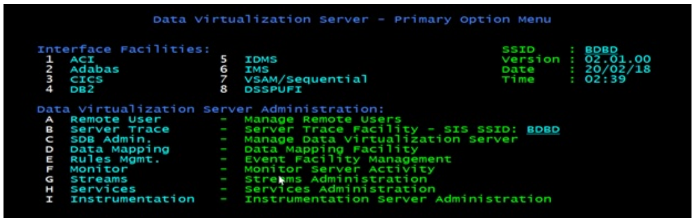


```
VIEW       CSD.AI38.EXECFB.DVS(AVZ9IN00) - 01.99              Columns 00001 00072
Command ===>                                                     Scroll ===> CSR
       /*----------------------------------------------------------------------*/
        if 1 = 1 then
          do
           "MODIFY PARM NAME(REUSETHREADS)                   VALUE(YES)"
           "MODIFY PARM NAME(TARGETTHREADCOUNT)              VALUE(500)"
           "MODIFY PARM NAME(THREADTIMEOUT)                  VALUE(300)"
           "MODIFY PARM NAME(ALTSTARTEDTASKID)              VALUE(TS8834)"
          end

       /*----------------------------------------------------------------------*/
       /* Enable Server Event Facility support (SEF). Issue a DEFINE            */
       /* RULESET statement for each SEF ruleset.                               */
       /*                                                                       */
       /* During initial installation you must change each DSNAME operand       */
       /* to match the data set names actually installed with the product.      */
       /* SWICNTLDSN is used for HTTP calls issued by the Studio.               */
       /*----------------------------------------------------------------------*/
          "MODIFY PARM NAME(SEFMAXSECONDS)                  VALUE(120)"
```

2. Restart the server to make the changes effect.

   You can verify the configuration of the parameter in the server trace. From the ISPF primary menu, select option **B Server Trace** to open the server trace.

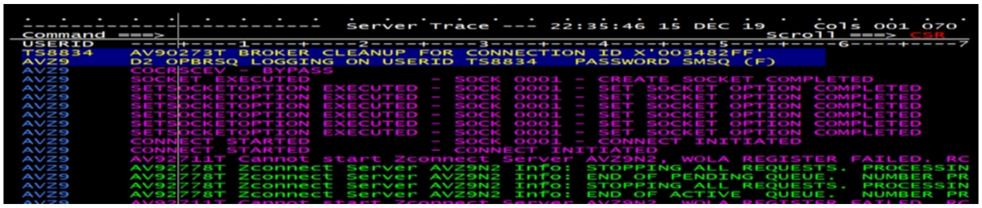


```
              Data Virtualization Server - Primary Option Menu

Interface Facilities:                                    SSID    : BDBD
 1  ACI                 5  IDMS                          Version : 02.01.00
 2  Adabas              6  IMS                           Date    : 20/02/18
 3  CICS                7  VSAM/Sequential               Time    : 02:39
 4  DB2                 8  DSSPUFI

Data Virtualization Server Administration:
 A  Remote User         -  Manage Remote Users
 B  Server Trace        -  Server Trace Facility - SIS SSID: BDBD
 C  SDB Admin.          -  Manage Data Virtualization Server
 D  Data Mapping        -  Data Mapping Facility
 E  Rules Mgmt.         -  Event Facility Management
 F  Monitor             -  Monitor Server Activity
 G  Streams             -  Streams Administration
 H  Services            -  Services Administration
 I  Instrumentation     -  Instrumentation Server Administration
```

   The following trace contains the configured ALTSTARTEDTASKID



```
-------------------------  Server Trace ---  22:35:46 15 DEC 19     Cols 001 070
Command ===>                                                    Scroll ===> CSR
USERID   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7
TS8834   AV90273T BROKER CLEANUP FOR CONNECTION ID X'003482FF'
AVZ9     D2 OPBRSQ LOGGING ON USERID TS8834    PASSWORD SMSQ (F)
AVZ9     COCRSCEV - BYPASS
AVZ9     SOCKET EXECUTED             - SOCK 0001 - CREATE SOCKET COMPLETED
AVZ9     SETSOCKETOPTION EXECUTED    - SOCK 0001 - SET SOCKET OPTION COMPLETED
AVZ9     SETSOCKETOPTION EXECUTED    - SOCK 0001 - SET SOCKET OPTION COMPLETED
AVZ9     SETSOCKETOPTION EXECUTED    - SOCK 0001 - SET SOCKET OPTION COMPLETED
AVZ9     SETSOCKETOPTION EXECUTED    - SOCK 0001 - SET SOCKET OPTION COMPLETED
AVZ9     SETSOCKETOPTION EXECUTED    - SOCK 0001 - SET SOCKET OPTION COMPLETED
AVZ9     SETSOCKETOPTION EXECUTED    - SOCK 0001 - SET SOCKET OPTION COMPLETED
AVZ9     CONNECT STARTED             - SOCK 0001 - CONNECT INITIATED
AVZ9     CONNECT STARTED             - CONNECT INITIATED
AVZ9     AV92711T Cannot start Zconnect Server AVZ9N2, WOLA REGISTER FAILED. RC
AVZ9     AV92778T Zconnect Server AVZ9N2 Info: STOPPING ALL REQUESTS. PROCESSIN
AVZ9     AV92778T Zconnect Server AVZ9N2 Info: END OF PENDING QUEUE.  NUMBER PR
AVZ9     AV92778T Zconnect Server AVZ9N2 Info: STOPPING ALL REQUESTS. PROCESSIN
AVZ9     AV92778T Zconnect Server AVZ9N2 Info: END OF ACTIVE  QUEUE.  NUMBER PR
```

Verifying the Data Virtualization Manager server installation

To verify the server installation, create a sample VSAM file and a virtual table, and then run a query that accesses the VSAM data.

Procedure

1. Create the sample VSAM file on the mainframe that hosts the server. Run the AVZGNSTF member in the *hlq*.SAVZCNTL 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. Run the AVZIVVS1 member in the *hlq*.SAVZCNTL data set to perform a batch extract of the sample VSAM file listing and create a virtual table that formats the result set that is returned from the VSAM file.

   This step runs a query against the sample VSAM file. The job should complete with a condition code of 0.

Chapter 4. Configuring access to data sources

Configure the Data Virtualization Manager 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 Data Virtualization Manager configuration member.

Before you begin

The server must be installed.

About this task

The SQL interface for Adabas provides seamless, real-time controlled access to Adabas data. It allows ODBC, JDBC, and web clients to access Adabas data in a relational model using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to Adabas. The result is a relational result set, with no host programming required.

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 19.

b) Modify the server configuration member. See "Modifying the Data Virtualization Manager configuration member" on page 20.

c) Configure security access to Adabas data. See "Configuring security access to Adabas data" on page 21.

Configuring the server started task JCL

Make the ADALNKR module available.

Before you begin

All LOAD library data sets allocated to the Data Virtualization Manager 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 Data Virtualization Manager configuration member

Enable the Adabas parameters in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 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.

Parameter	Description	Valid values
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 security access to Adabas data

Configure security to provide access to Adabas.

Procedure

See "Configuring Adabas security" in the *Administration Guide*.

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

The SQL interface for DB2 provides seamless, real-time controlled access to RDBMS data. It allows ODBC, JDBC, and web clients to access RDBMS data in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to RDBMS. The result is a relational result set, with no host programming required.

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

Procedure

1. Enable the RDBMS access method in the Data Virtualization Manager configuration member.

   See “Modifying the Data Virtualization Manager configuration member for DRDA” on page 23.
2. Configure access to the database.
   - IBM DB2 for z/OS

     Configure DB2 to use either the Distributed Relational Database Architecture (DRDA) access method or the Resource Recovery Services attachment facility (RRSAF) access method.

     See “Configuring access to IBM Db2 for z/OS” on page 22.
   - 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 36.

Configuring access to IBM Db2 for z/OS

Configure Db2 to use either the Distributed Relational Database Architecture (DRDA) access method or the Resource Recovery Services attachment facility (RRSAF) access method.

About this task

Using DRDA might yield a lower total cost of ownership than RRSAF because DRDA allows a higher percentage of Db2 work to run in SRB mode and offloaded to a zIIP specialty engine.

If you have a zIIP specialty engine, use DRDA. If you do not have a zIIP specialty engine, use RRSAF.

Before you issue Db2 requests, you must bind DRDA, RRSAF, or both into packages within each Db2 subsystem. Binding both access methods is recommended.

Configure access to Db2 for z/OS databases as follows.

Procedure

1. "Configuring security" on page 22
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 Data Virtualization Manager configuration member for DRDA" on page 23
     b. "Configuring DB2 for DRDA" on page 28
   - If you are not using a zIIP specialty engine, enable the RDBMS access method for RRSAF:
     a. "Modifying the Data Virtualization Manager configuration member for RRSAF" on page 29
     b. "Configuring Db2 for RRSAF" on page 30

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 Virtualization Manager 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.
   - `AVZRADB2` is for IBM Resource Access Control Facility (RACF) security.
   - `AVZA2DB2` is for CA ACF2 (Access Control Facility) security.
   - `AVZTSDB2` is for CA Top Secret Security (TSS).

2. Assign users READ authority.
   - For DRDA, assign users READ authority to the *ssid*.DIST profile.
   - For RRSAF, assign users READ authority to the *ssid*.RRSAF profile.

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 Virtualization Manager server in the server started task JCL must be APF-authorized.

About this task

This task is not applicable for Db2 UDB (Linux, UNIX, and Windows).

Note: Skip this task if the Db2 interface modules, DSNALI and DSNHLI, are in the z/OS linklist.

Procedure

Edit the JCL in the *hlq*.SAVZCNTL(AVZ1PROC) 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.

Modifying the Data Virtualization Manager 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 Data Virtualization Manager configuration member.

About this task

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

Modify the Data Virtualization Manager configuration member in data set *hlq*.AVZS.SAVZEXEC(AVZSIN00). The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 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 AVZSIN00 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, the ODBC and JDBC clients must explicitly opt in for DRDA to be used by setting the user parameter connection variable to 'DRDA'. **Note:** ODBC and 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 ODBC/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 ODBC/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 Virtualization Manager 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.*)	**USRIDPWD** 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

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 Data Virtualization Manager for z/OS DBRMs into packages within each target DB2 subsystem.

Procedure

1. Set the DEFAULTDB2SUBSYS parameter in the Data Virtualization Manager configuration member to a valid DB2 subsystem name.
2. Edit the AVZBINDD job that is supplied in the *hlq*.SAVZCNTL data set.

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

Modifying the Data Virtualization Manager 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 Data Virtualization Manager configuration member.

About this task

This task is only applicable for Db2 for z/OS.

Modify the Data Virtualization Manager configuration member in data set *hlq*.AVZS.SAVZEXEC(AVZSIN00). The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 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 Data Virtualization Manager configuration member AVZSIN00 to a valid Db2 subsystem name.
3. In the AVZSIN00 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.

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

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 AVZBINDC member of the *hlq*.SAVZCNTL data set to bind the following server product plans:
   - AVZC1010 is bound using cursor stability.
   - AVZR1010 is bound using repeatable read.
   - AVZS1010 is bound using read stability.
   - AVZU1010 is bound using uncommitted read.

   Use AVZC1010 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*.SAVZCNTL 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.

Disabling query acceleration

You can use a Data Virtualization Manager 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 AVZMDTBL 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 Data Virtualization Manager configuration member (AVZSIN00) 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 Data Virtualization Manager server - 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 AVZMDTBL rule, as follows:
   a) Specify S next to AVZMDTBL 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 AVZMDTBL and pressing Enter.
5. Set the rule to Auto-enable by specifying A next to AVZMDTBL and pressing Enter.
   Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

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 Data Virtualization Manager 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, AVZMDLDU, 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 AVZMDLDU, 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 AVZMDLDU 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 AVZMDLDU.

Note: Sample rule AVZMDLDU 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 Data Virtualization Manager configuration member (AVZSIN00) 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 Data Virtualization Manager server - 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 AVZMDLDU rule, as follows:
   a) Specify S next to AVZMDLDU 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 (AVZSIN00 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 AVZMDLDU and pressing Enter.
5. Set the rule to Auto-enable by specifying A next to AVZMDLDU and pressing Enter.

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

Db2 for z/OS data access methods

Db2 for z/OS data can be accessed by the Data Virtualization Manager 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 Virtualization Manager 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 Virtualization Manager 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 Virtualization Manager server must collect information about the Db2 database. This information is collected using the **DRDARange** command and is stored within the Data Virtualization Manager server metadata repository.

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

Using Db2 Direct

Db2 Direct is a Data Virtualization Manager 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 Virtualization Manager 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.

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 33.

About this task

By default, Db2 Direct is enabled in the Data Virtualization Manager 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 AVZMDTBL.
2. To define the VCATNAME for the Db2 system catalog tables, perform the following steps:
   a) Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) 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 Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) 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 Virtualization Manager server:

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

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

Table 3. 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 4. 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
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 Data Virtualization Manager Server Event Facility (SEF) rules, and optionally setting up alternate authentication information.

Procedure

1. "Modifying the Data Virtualization Manager configuration member" on page 37.
2. Configure the Data Virtualization Manager Server Event Facility rules and set up authentication for the appropriate database.
    - "Configuring rules and authentication for Big SQL" on page 42.
    - "Configuring rules and authentication for dashDB" on page 43.
    - "Configuring rules and authentication for LUW databases" on page 44.
    - "Configuring rules and authentication for Microsoft SQL Server" on page 45.
    - "Configuring rules and authentication for Oracle DRDA" on page 46.
    - "Configuring rules and authentication for QMF DRDA Server" on page 47.

Modifying the Data Virtualization Manager configuration member

Enable the RDBMS access method in the Data Virtualization Manager Data Virtualization Manager configuration member.

About this task

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

Modify the Data Virtualization Manager configuration member in data set *hlq*.AVZS.SAVZEXEC(AVZSIN00). The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 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 AVZSIN00 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, the ODBC and JDBC clients must explicitly opt in for DRDA to be used by setting the user parameter connection variable to 'DRDA'. **Note:** ODBC and 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.

Parameter	Description	Valid values
DRDASKIPWLMSETUP	If set to YES, WLM information is not collected and sent to DRDA during ODBC/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.
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 ODBC/JDBC driver.
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.

Parameter	**Description**	**Valid values**
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 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 Data Virtualization Manager 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:

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

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

AVZEBIGG
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 SAVZXSQL(AVZSBIGC) to allow Data Virtualization Manager studio Meta discovery on Big SQL databases.
   a) On the Data Virtualization Manager server - 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEBIGG) 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 Data Virtualization Manager server - 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 Data Virtualization Manager 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:

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

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

AVZEDDBG

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 SAVZXSQL(AVZSDDBC) to allow Data Virtualization Manager studio Meta discovery on dashDB databases.
   a) On the Data Virtualization Manager server - 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEDDBG) 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 Data Virtualization Manager server - 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 Data Virtualization Manager 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:

AVZSLUWC

An SQL rule that allows Meta discovery on LUW databases.

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

AVZELUWG
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 SAVZXSQL(AVZSLUWC) to allow Data Virtualization Manager studio Meta discovery on LUW databases.
   a) On the Data Virtualization Manager server - 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZELUWG) 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 Data Virtualization Manager server - 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 Data Virtualization Manager 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 Virtualization Manager 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 Virtualization Manager server to optionally change the logon credentials for a user when accessing SQL Server.

This task uses the following tools:

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

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

AVZEMSSG
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 SAVZXSQL(AVZSMSSC) to allow Data Virtualization Manager studio Meta discovery on SQL Server databases.
   a) On the Data Virtualization Manager server - 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEMSSG) 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 Data Virtualization Manager server - 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 Data Virtualization Manager 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 Virtualization Manager 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:

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

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

AVZEORAG
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 SAVZXSQL(AVZSORAC) to allow Data Virtualization Manager studio Meta discovery on Oracle databases.
   a) On the Data Virtualization Manager server - 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEORAG) 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 Data Virtualization Manager server - 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 Data Virtualization Manager 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 Virtualization Manager 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:

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

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

AVZEQMFG
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 SAVZXSQL(AVZSQMFC) to allow Data Virtualization Manager studio Meta discovery on QMF DRDA Server databases.
   a) On the Data Virtualization Manager server - 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEQMFG) 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 Data Virtualization Manager server - 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 Virtualization Manager 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 Virtualization Manager 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 Virtualization Manager 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 Virtualization Manager 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 Virtualization Manager 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 Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization in the step “Copying target libraries” on page 10.
2. Review the following statement in your AVZSIN00 member, and update the setting if necessary:

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

Configuring access to data in IBM IMS databases

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

Before you begin

The server must already be installed.

About this task

IBM Data Virtualization Manager for z/OS provides seamless, real-time controlled access to IMS database data. It allows ODBC, JDBC, Data Virtualization Manager client, and web clients to access IMS database data in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to an IMS database. The result is a relational result set with no host programming required.

For a key field that allows using either an index (HIDAM) or a randomizer (HDAM), IBM Data Virtualization Manager for z/OS uses DBCTL. For unkeyed fields, IBM Data Virtualization Manager for z/OS uses IMS Direct as both IMS Direct and IMS DBCTL need to scan the 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 Virtualization Manager 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.

Configuring ODBA Support

You can configure the Data Virtualization Manager to support the Open Database Access (ODBA) interface. Perform the following tasks to enable ODBA support.

Procedure

1. Turn off IMS DBCTL support in the IN00.
2. Under **Enable IMS CCTL/DBCTL support**, choose **DontDoThis**. This sets the value of the parameter DBCTL to **NO**
3. Add the following to the IN00:

```
/*------------------------------------------------------------------*/
/* Enable IMS ODBA support                                          */
/*------------------------------------------------------------------*/
if DoThis then
do
```

```
   "MODIFY PARM NAME(IMSODBA)             VALUE(YES)"
   "MODIFY PARM NAME(RRS)                 VALUE(YES)"
   "MODIFY PARM NAME(MAXODBACONNECT)      VALUE(12)"
   "MODIFY PARM NAME(TRACEIMSDLIEVENTS)   VALUE(NO)"
   "MODIFY PARM NAME(IMSID)               VALUE(IMS1)"

   "DEFINE IMSODBA",
        "NAME(IMS1)",
        "CONTROLREGIONID(IMS1)",
        "MAXTHREADS(10)",
        "MINTHREADS(3)",
        "USERID(AVZ9)"
end
```

4. Confirm that the virtual table uses **IMS/DBCTL** on Studio
5. If IMS is a FASTPATH DB, add the following.

```
"FASTPATHNBA(100)",
"FASTPATHOVERFLOW(100)",
"FASTPATHALLOCATED(100)",
```

Modifying the Data Virtualization Manager configuration member for DBCTL

Enable the IMS database control (DBCTL) parameters in the Data Virtualization Manager configuration member.

About this task

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 *
```

The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 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)

Parameter	Description	Valid values
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 Data Virtualization Manager configuration member for IMS Direct

Enable and configure the IMS Direct parameters in the Data Virtualization Manager 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 Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization.
2. In the AVZSIN00 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.
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

Parameter	Description	Valid values
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 Data Virtualization Manager configuration member, and set virtual table options.

Data Virtualization Manager 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 Data Virtualization Manager configuration member for IBM MQ

To enable support for MQ data, you must update your Data Virtualization Manager 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 Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization.
2. Add the following statement to your AVZSIN00 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 Data Virtualization Manager 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 AVZMDLMQ documents these settings.

When accessing MQ data with sample rule AVZMDLMQ (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 AVZMDLMQ.

Note: Sample rule AVZMDLMQ 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 Data Virtualization Manager configuration member (AVZSIN00) 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 Data Virtualization Manager server - 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 AVZMDLMQ rule, as follows:

a) Specify S next to AVZMDLMQ 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.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 AVZMDLMQ and pressing Enter.

5. Set the rule to Auto-enable by specifying A next to AVZMDLMQ and pressing Enter.

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

Configuring access to native VSAM

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

Before you begin

The server must already be installed.

About this task

The SQL interface for native VSAM provides seamless, real-time controlled access to native VSAM data. It allows ODBC, JDBC, and web clients to access native VSAM data in a relational model using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to native VSAM. The result is a relational result set, with no host programming required.

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 Virtualization Manager server.

   Run the AVZGNSTF member in the *hlq*.SAVZCNTL 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 AVZIVVS1 member in the *hlq*.SAVZCNTL 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 AVZIVVS1 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 Virtualization Manager 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 AVZBUFND and AVZBUFNI 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 AVZBUFND. The AVZBUFND 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:

  ```
  AVZ1000I VTB.OPTBVSND set to 30
  ```

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

  Use sample rule AVZBUFNI. The AVZBUFNI 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:

  ```
  AVZ1000I VTB.OPTBVSNI set to 30
  ```

Procedure

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

   a) Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization in the step "Copying target libraries" on page 10.

   b) Add the following statements to your AVZSIN00 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 Data Virtualization Manager configuration member (AVZSIN00) 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:

      1) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt**.

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

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

   c) Enable each rule as follows:

   - Specify E next to AVZBUFND and press Enter.
   - Specify E next to AVZBUFNI and press Enter.

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

   - Specify A next to AVZBUFND and press Enter.
   - Specify A next to AVZBUFNI 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.

VSAM Record Level Sharing

Record-level sharing (RLS) is an access mode for VSAM data sets to enable the VSAM data to be shared, with full update capability, between many applications. Setting the VSAMOPENRLS parameter to yes makes the Access method Control Block (ACB) opened indefinitely. This helps to respond to frequent separate requests for map reduce processing against a single shared file.

Setting the VSAMRLSQUIESCE to yes causes the shared VSAM files to be closed when the use count reaches zero where the use count is calculated based on the OPEN and CLOSE requests made by a map reduce thread.

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.

Before you begin

The server must already be installed.

About this task

The SQL interface for sequential files provides seamless, real-time controlled access to sequential files. It allows ODBC, JDBC, Data Virtualization Manager client, and web clients to access sequential files in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to native VSAM. The result is a relational result set, with no host programming required.

Reading ahead tracks for sequential file access

You can use a Data Virtualization Manager 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 AVZMLTAC is provided.

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

The AVZMLTAC 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:

```
AVZ1000I VTB.OPTBMACC set to 15
```

Use the following procedure to set up the rule.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) 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 Data Virtualization Manager server - 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 AVZMLTAC and pressing Enter.
4. Set the rule to Auto-enable by specifying A next to AVZMLTAC and pressing Enter.

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

Configuring access to IBM CICS

For VSAM data access via the CICS Transaction Server (TS), you need to configure the Data Virtualization Manager configuration member and CICS TS.

Before you begin

The server must already be installed.

About this task

The server connects to CICS TS, via the IBM EXCI (External CICS Interface).

CICS provides logging and recovery facilities that are required if VSAM updates are being applied. When accessing VSAM files that are owned by CICS TS, recovery is provided by CICS TS.

Modifying the Data Virtualization Manager configuration member

Enable the VSAM data access via CICS TM parameters in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 member, locate the comment "ENABLE CICS TRANSACTION SERVER SUPPORT."
2. Enable the CICS TS parameters by changing `if DontDoThis` to `if DoThis`.

   ```
   if DoThis then
     do
      "MODIFY PARM NAME(EXCI)                    VALUE(YES)"
      "MODIFY PARM NAME(EXCICONNECTIONNAME)      VALUE(CICA)"
      "MODIFY PARM NAME(TRACEEXCIDPLEVENTS)      VALUE(YES)"
      "MODIFY PARM NAME(CICSSENDABCODE)          VALUE(YES)"
      "MODIFY PARM NAME(RRSCICS)                 VALUE(YES)"
   ```

The following table lists the parameters for configuring support for a CICS TS:

Parameter	Description	Valid values
EXCI	Initialize EXCI support.	**NO** **YES** Default value.
EXCICONNECTIONNAME	EXCI Default Connection Name. Specifies the default CICS Connection Name for EXCI support.	EXCW
TRACEEXCIDPLEVENTS	Trace EXCI DPL Events	**NO** Default value. **YES**
CICSSENDABCODE	Send ABEND Code to Clients. Controls the sending of the CICS ABEND code to the client. If set to YES, the ABEND code is returned to the client as part of the error message.	**NO** Default value. **YES**
RRSCICS	Specifies whether RRS CICS support is active.	**NO** Default value. **YES**

3. Create a `DEFINE CONNECTION` statement for each CICS region. Include the following parameters in the statement:

```
"DEFINE CONNECTION  NAME(CICA)",
                   "GROUP(CICA)",
                   "ACCESSMETHOD(IRC)",
                   "NETNAME(CICADBVS)",
                   "INSERVICE(YES)",
                   "PROTOCOL(EXCI)",
                   "APPLID(XXXXXXXX)",
                   "LOADBALGROUP(LBG1)",
                   "SECURITYNAME( )",
end
```

Parameter	Description	Valid values
NAME	Specify a four-character name for the connection to the CICS region.	Four-character name
GROUP	Specify the same name as the connection name.	Eight-character name
ACCESSMETHOD	Specify IRC.	IRC
NETNAME	Specify the network name of the remote system. To identify these connections in CICS, use a name that is a combination of the connection name and the server subsystem name.	Eight-character name

Parameter	Description	Valid values
INSERVICE	Specify YES to open the connection at server startup. Specify NO to open the connection manually.	**NO** Default value. **YES**
PROTOCOL	Specify EXCI.	EXCI
APPLID	Specify the VTAM APPLID of the target CICS.	No restriction on the APPLID name
LOADBALGROUP	Specify the name of the group that is used to balance the CICS workload across multiple CICS regions. Specify the same group name in each DEFINE CONNECTION statement that you create. (*Optional*)	Eight-character name
SECURITYNAME	Specify a valid security name from the remote system.	Eight-character name

4. Create a DEFINE SESSION statement for each CICS region. Include the following parameters in each statement:

```
"DEFINE SESSION   NAME(CICA)",
                   "GROUP(CICA)",
                   "CONNECTION(CICA)",
                   "PROTOCOL(EXCI)",
                   "RECEIVERFX(XD)",
                   "RECEIVERCOUNT(0)",
                   "SENDPFX(SD)",
                   "SENDCOUNT(20)",
                   "IOAREALEN(4096)",
end
```

Parameter	Description	Valid values
NAME	Specify the same name that you specified for the NAME when you defined the connection for this CICS region.	Four-character name
GROUP	Specify the same name that you specified for the NAME when you defined the connection for this CICS region.	Eight-character name
CONNECTION NAME	Specify the same name that you specified for the NAME when you defined the connection for this CICS region.	Four-character name
PROTOCOL	Specify EXCI.	EXCI
RECEIVEPFX	Not applicable.	This field should be blank its included to provide complete list of connection parameters.
RECEIVECOUNT	Not applicable.	This field should be blank its included to provide complete list of connection parameters.

Parameter	Description	Valid values
SENDPFX	Specify a one- or two-character prefix for the session name. The session name, which is limited to four characters, is composed of the prefix and the session number. Therefore, if you define more than 99 sessions, specify a one-character prefix.	A one- or two-character prefix
SENDCOUNT	Specify the maximum number of concurrent transactions. This value should match the RECEIVECOUNT value minus one set in the DEFINE SESSIONS definition in the AVZCICSD job.	Any number up to 255
IOAREALEN	Specify the length, in bytes, of the terminal input/output area to use to process transmitted messages.	Any value up to 4096K

Configuring CICS

Configure CICS by modifying the CICS started tasks JCL, the System Initialization Table (SIT), and the DFHCSD file.

Procedure

1. Add the *hlq*.SAVZCLOD library to the DFHRPL concatenation in each CICS region that you want to connect to server.
2. Use the CEMT INQUIRE IRC command to verify that the CICS interregion communication (IRC) facility is open.

   To start IRC at CICS system startup, ensure that the IRCSTRT=YES parameter is in the SITPARM for the CICS region.
3. Update the DFHCSD file by performing the following steps:

   a) For each CICS region, modify and submit the AVZCICSD job that is in *hlq*.SAVZCNTL data set:

   - Update the DEFINE CONNECTION and DEFINE SESSION values to match the definitions that you specified in the Data Virtualization Manager configuration member. The GROUP value is the CICS GROUPNAME and does not need to match the GROUP name that is defined for the server. By default, 21 sessions are defined. Set this value to the maximum number of concurrent transactions for a single instance of server. The maximum value is 250.
   - Change the name of the *hlq*.FILEA data set to the FILEA VSAM data set name. This VSAM file is used when you verify access to CICS data.

   The member contains additional information about modifying the job.

   b) Update LIST(*YOURLIST*) to match the startup group list for the CICS region.

   c) Review more comments in the JCL notes section for additional considerations. Define all of the definitions in the *hlq*.SAVZCNTL(AVZCICSD) member.

Configuring security

Configure security to provide user access to CICS TS.

About this task

See "CICS security" in the *Administrator's Guide*.

Configuring access to zFS files

The Data Virtualization Manager server is already configured to support zFS files. No modifications are needed to configure access to zFS files. However, you should verify 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 Virtualization Manager 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*.SAVZSMAP 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 67

- “Configuring access to SYSLOG files” on page 70
- “Configuring access to OPERLOG files” on page 71

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*.SAVZSMAP data set to the AVZMAPP 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 AVZSIN00 member, as follows:

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

   Verify the VTB ruleset name:

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

   If there were any changes to AVZSIN00, recycle the server started task.
3. To enable real-time access to SMF data, add the following statements to the AVZSIN00 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 AVZSIN00 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. To use SMF_1100P* maps, add the following statements to the AVZSIN00 member:

```
"MODIFY PARM NAME(ACIMAPREDUCEBUFF)  VALUE(16383K)"
"MODIFY PARM NAME(ACIMAPREDUCESPACE) VALUE(64)"
```

6. Enable reading SMF data from GDG data sets and access to SMF data using dynamic data set names by enabling Data Virtualization Manager Server Event Facility rule AVZSMFT1 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 AVZSMFT1 in the VTB ruleset:

      1) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt**.
      2) Specify option 2, **SEF Rule Management**.
      3) Enter VTB for **Display Only the Ruleset Named**.
      4) Enable the rule by specifying E and pressing Enter.
      5) 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:

        1) Change Global Prefix to `GLOBAL2`.
        2) Select SMFTBL2 by entering S next to the SMFTBL2 data set.
        3) 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

To configure access to system log (SYSLOG) files, you need to configure the Data Virtualization Manager configuration member and the server virtual table rules.

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:

AVZSYSLG

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

AVZSYSL2

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)
```

AVZSYSL3

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. Customize the Data Virtualization Manager configuration member.

   To enable virtual table rule events, configure the SEFVTBEVENTS parameter in the AVZSIN00 member, as follows:

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

2. Access the VTB rules, as follows:
   a) In the Data Virtualization Manager server - 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. For AVZSYSLG, customize the rule, as follows:
   a) Specify S next to AVZSYSLG to edit the rule.
   b) Customize the rule with the SYSLOG data set name.
   c) Save your changes and exit the editor.

   Note: For AVZSYSL2 and AVZSYSL3, no customization of the rule is needed.
4. Enable each rule by specifying E next to the member name and pressing Enter.
5. 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.
6. If global variables are needed, set up the SYSLOG global variable.

Configuring access to OPERLOG files

No modifications are needed to configure the Data Virtualization Manager 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 CA IDMS data

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

Data Virtualization Manager 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 Virtualization Manager 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.

Modifying the Data Virtualization Manager configuration member for CA IDMS

To optionally configure server parameters for CA IDMS, you can update your Data Virtualization Manager server configuration file.

About this task

The CA IDMS server parameters can assist you in configuring CA IDMS data access. In most typical environments, the default settings for these parameters will not need modification.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and may have been copied to a new data set for customization.
2. Add the following statements to your AVZSIN00 member:

   The following table lists the parameters for configuring CA IDMS data access:

Parameter	Description	Valid values
MAXIDMSRUNUNITS	MAXIMUM IDMS RUN UNITS This parameter limits the number of concurrent IDMS run units that a server will start to access a CA IDMS central version. Limiting concurrent IDMS run units will prevent storage related user 3134 abends when creating run units with CA IDMS.	Positive numeric value. Default value is 4.
SQLENGIDMSRUTIMOUT	IDMS RUN UNIT INACTIVITY TIMEOUT Specifies the length of time in seconds to keep a run unit active for reuse by subsequent SQL queries in a client connection.	Positive numeric value. Default value is 60 seconds.

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 AVZISIV1 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 5. 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 6. 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 AVZISIV1 member in the *hlq*.SAVZCNTL data set.
2. Modify the JCL according to the instructions provided in the AVZISIV1 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

AVZISIV1 installs CA IDMS EMPDEMO and network schema maps into the server map data set.

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 Virtualization Manager 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 75.
2. Define ADDI project in the server configuration member. See "Modifying the configuration member for ADDI access" on page 75.
3. Activate virtual table rules. See "Configuring virtual table rules for ADDI" on page 79.
4. Define credentials for target database(s). See "Configuring authentication for ADDI" on page 79.

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 Virtualization Manager 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 SAVZSAMP member AVZIAMPD:

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 Virtualization Manager 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 AVZIAMPS member in the *hlq*.SAVZCNTL data set.
2. Modify the JCL according to the instructions provided in the AVZIAMPS 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 Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager 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 78.

The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 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
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`.

Parameter	Description	Valid values
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 Virtualization Manager 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
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.

Parameter	Description	Valid values
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 75.
- 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 75.

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 Virtualization Manager 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 AVZDRATH batch utility.

Configuring virtual table rules for ADDI

Configure Data Virtualization Manager 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 AVZIADTB and AVZIADVW provide support to process tables starting with `IAD_` and views starting with `IADV_`.

AVZIADTB
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.

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

With the rules activated, the Data Virtualization Manager 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 AVZIADTB and AVZIADVW in the VTB ruleset:

a) In the Data Virtualization Manager server - 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 Virtualization Manager 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:

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. Use this utility to define alternate logon information for the Data Virtualization Manager 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.

AVZEMSSG

An ATH rule that swaps z/OS user information with SQL Server authentication information defined using the AVZDRATH 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 AVZDRATH and enable ATH rule AVZEMSSG
- Add authentication information for specific mainframe users using sample job AVZDRATH and enable ATH rule AVZEMSSG

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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.
2. Auto-enable the SEF ATH rule SAVZXATH(AVZEMSSG) 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 Data Virtualization Manager server - 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 AVZEMSSG 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 Virtualization Manager 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 81.
2. Define RAA database in the server configuration member. "Modifying the configuration member for RAA access" on page 82.
3. Activate virtual table rules. See "Configuring virtual table rules for RAA" on page 84.
4. Define credentials for target database(s). See "Configuring authentication for RAA" on page 85.

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 Virtualization Manager 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 SAVZSAMP member AVZRAMPD.

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 Virtualization Manager 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 AVZRAMPS member in the *hlq*.SAVZCNTL data set.
2. Modify the JCL according to the instructions provided in the AVZRAMPS 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 Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager 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 84.

The server member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

In the AVZSIN00 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
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.

Parameter	Description	Valid values
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 Virtualization Manager 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.
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 ODBC/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 81.
- 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 82.
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 Virtualization Manager 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 AVZDRATH batch utility.

Configuring virtual table rules for RAA

Configure Data Virtualization Manager 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 AVZRAATB and AVZRAAVW provide support to process tables starting with RAA_ and views starting with RAAV_.

AVZRAATB

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.

AVZRAAVW

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

With the rules activated, the Data Virtualization Manager 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 AVZRAATB and AVZRAAVW in the VTB ruleset:

a) In the Data Virtualization Manager server - 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 Virtualization Manager should also be usable for the z/OS system where Db2 resides. By default, the Data Virtualization Manager 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 Virtualization Manager 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:

AVZDRATH

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

AVZEDB2G

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 AVZDRATH and enable ATH rule AVZEDB2G
- Add authentication information for specific mainframe users using sample job AVZDRATH and enable ATH rule AVZEDB2G

If z/OS user IDs and passwords used to connect to the Data Virtualization Manager 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 AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:

   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.

   b) Modify the JCL according to the instructions provided in the AVZDRATH 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 AVZDRATH member and submit the job.

2. Auto-enable the SEF ATH rule SAVZXATH(AVZEDB2G) 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 Data Virtualization Manager server - 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 AVZEDB2G rule member by entering A and pressing Enter.

Chapter 5. Installing the Data Virtualization Manager client drivers

Install the drivers on your development workstation that are required to access data made available through Data Virtualization Manager.

IBM Data Virtualization Manager for z/OS supports the following drivers:

- JDBC
- ODBC

Installing the JDBC driver

Java-based applications and tools use the JDBC driver to access data that is made available through Data Virtualization Manager.

About this task

The JDBC driver is a Type 4 driver that is written in Java and implements the network protocol for the Data Virtualization Manager.

Procedure

1. From the mainframe, transfer the driver installation member *hlq*.SAVZBIN(AVZBIN2) to your development workstation using the File Transfer Protocol (FTP) in binary mode.
2. Rename the file to JDBCdriver.zip.
3. On your development workstation, create a directory, and then extract the contents of the JDBCdriver.zip archive into that directory.

What to do next

For details about the JDBC driver, see the *Developer's Guide*.

Installing the ODBC driver

For non-Java based applications and tools, use the ODBC driver to access data that is made available through Data Virtualization Manager.

About this task

The installation process installs the driver, utilities, and sample programs as shown in the following table. Installation requires approximately 15 MB disk space.

Install set	Components installed
Default	ODBC driver and samples.
Minimal	ODBC driver only, installed onto local computer.
Custom	Allows customization of installed features
Network Admins	Only installs driver definitions on local machine. No physical drivers are installed locally. Drivers must already be installed on another network machine using the Network Install option.

As part of the installation procedure, you are able to specify the destination folder and whether environment variables required by the driver are available to all users of the computer or only to the user who performs the installation.

Procedure

1. From the mainframe, transfer the driver installation member *hlq*.SAVZBIN(AVZBIN3) to your development workstation using the File Transfer Protocol (FTP) in binary mode.
2. Rename the file to ODBCdriver.zip.
3. On your development workstation, create a directory, and then extract the contents of the ODBCdriver.zip archive into that directory.

   ODBC installers for the following platforms are extracted to the directory:

   - Red Hat Enterprise Linux (RHEL)
   - SUSE Linux Enterprise Server (SLES)
   - UNIX System Services (USS)
   - Ubuntu
   - Windows
4. Run either the 32-bit or 64-bit version of the installer as appropriate for your target ODBC applications, and follow the installation wizard.
5. Optional: To uninstall the ODBC driver, go to **Control Panel** > **Programs and Features**, select the Data Virtualization Manager ODBC Driver item, and click **Uninstall/Change**.

What to do next

For details the ODBC driver, see the *IBM Data Virtualization Manager Developer's Guide*.

Chapter 6. Installing the Data Virtualization Manager studio

The Data Virtualization Manager studio is an Eclipse-based user interface that allows you to create and manage metadata on the Data Virtualization Manager server that is required to provide access to your mainframe and non-mainframe data.

Before you begin

Before installing the Data Virtualization Manager studio, verify that all installation prerequisites are met:

System component	Requirement
Permissions	You have appropriate user logon credentials and user privileges on your client system to install the Data Virtualization Manager studio. For example, to install the studio on Windows, you need administrator authority; ensure that your user profile has the appropriate privileges to write to the target system location.
Supported operating systems	Windows 7, 8, 10 Linux – Red Hat Enterprise Linux 6.7 or higher; Ubuntu 16 or higher macOS (Sierra)
System memory	A minimum of 4 GB is recommended.
Hard disk space	A minimum of 1 GB is recommended.
Software	Installing the Data Virtualization Manager studio as a plug-in to your existing Eclipse environment requires Eclipse Kepler 4.3 (or higher) and Java 1.7 or Java 1.8.

About this task

You can choose to install the Data Virtualization Manager studio software in a new Eclipse environment (a *full install*) or as a plug-in within an existing Eclipse environment (a *plug-in install*):

Full install

A *full install* installs the Data Virtualization Manager studio software in a new Eclipse environment on Windows. This method includes the installation of Windows Eclipse (64-bit), JRE 1.7, and the Data Virtualization Manager plug-in. This installation method is recommended for Windows 64-bit users who are installing the studio for the first time.

Plug-in install

A *plug-in install* installs only the Data Virtualization Manager plug-in. This installation method is recommended for the following users:

- Users on all supported platforms other than Windows 64-bit
- Existing Eclipse users that want to reuse their own Java runtime and Eclipse installation
- Users wanting to upgrade their existing Data Virtualization Manager studio installation with a newer version of the Data Virtualization Manager plug-in

Procedure

1. From the z/OS mainframe, transfer the installation member *hlq*.SAVZBIN(AVZBIN1) to a folder on your workstation using the File Transfer Protocol (FTP) in binary mode.
2. Rename the file to `avz-studio.zip`.

3. Create a new installation folder for the Data Virtualization Manager studio.
4. Double-click the `avz-studio.zip`, and then extract the contents to the installation folder.
5. In the installation folder, navigate to the `studio\install` folder that was created, and then select one of the following installation methods:

   - To perform a *full install*, installing the Data Virtualization Manager studio software in a new Eclipse environment, complete the following steps:

     a. In the `studio\install` folder, run the `setup.bat` script.

     b. After the installation completes, launch the Data Virtualization Manager studio using the shortcut created on the desktop or in the **Start** menu.

   - To perform a *plug-in install*, installing the Data Virtualization Manager studio software as a plug-in to an existing Eclipse environment, complete the following steps:

     a. From your Eclipse application, click **Help** > **Install New Software**, and then click **Add**.

     b. On the Add Repository dialog box, click **Archive**.

     c. Locate the `dvm.zip` file in the `studio\install` folder, and then click **Open**.

     d. Enter the software file name, a name for the repository, and then click **OK**.

     e. Select the check box next to the software item, **Data Virtualization Manager for z/OS**, and then click **Next**.

     f. Complete the remaining installation wizard steps, and then restart Eclipse when prompted.
6. To begin using Data Virtualization Manager studio, open the DV Data perspective using the menu option **Window** > **Open Perspective**.

Verifying the studio installation

Verify that you can connect from the studio to the server and browse metadata on the server.

Procedure

1. On the **Server** tab, click **Set Server**.
2. Provide information for the following fields and click **OK**:

Field	Description
Host	The z/OS LPAR name on which the Data Virtualization Manager server is running.
Port	The JDBC port number that the server is using. During customization, the port number is specified in the server configuration file; the parameter name is **OEPORTNUMBER**. To locate this number, use SDSF on the mainframe to browse the server JOB output and search for `OEPORTNUMBER`.
Userid	The user ID that the server will use to authenticate the connection.
User Password	The password that corresponds to the user ID being used to connect to the server.

The **Server** tab displays the new server connection. You can now browse the server metadata and configure the interfaces for the solutions that you want to use.

Chapter 7. JDBC Gateway

Use the JDBC Gateway to virtualize any JDBC 4.0 compliant database.

Topics:

- "Installing the JDBC Gateway" on page 91. This topic provides information about installing the JDBC Gateway component, including system requirements.
- "Using the JDBC Gateway" on page 95. This topic provides information about supported data sources and configuring access to those data sources.

Installing the JDBC Gateway

The *JDBC Gateway* is a Data Virtualization Manager distributed application server that allows direct connectivity to JDBC data sources. Install the JDBC Gateway to connect directly to JDBC data sources.

Before you begin

Before installing the JDBC Gateway, review the following points:

- For an overview of the JDBC Gateway solution, see "Using the JDBC Gateway" on page 95.
- The following terminology is used in the installation procedure:
  - *JDBC Gateway server*. The server is the backend component that allows communication with the Data Virtualization Manager server.
  - *JDBC Gateway administrative console*. The administrative console is the front-end web component that you use to configure your data sources. Only a single user (web client) can access the JDBC Gateway administrative console at a time. When installing the JDBC Gateway, you must specify a specific user ID for this purpose. This user ID is an internal application ID that allows access to the web user interface.
  - *Port for the Web UI*. This port will be used to access the Web-based administrative console and is specified during the installation procedure.

    Note: The JDBC Gateway also uses another port to listen for incoming DRDA requests. This DRDA listener port is set later when configuring the JDBC Gateway.
- Before installing the JDBC Gateway, verify that all installation requirements are met, as follows:

System component	Requirement
Permissions	You have appropriate user logon credentials and user privileges on your client system to install the JDBC Gateway. For example, to install and deploy the JDBC Gateway on Windows, you may need to run with administrator privileges depending on the target location.
Supported platforms	The JDBC Gateway is a pure Java application and therefore can be deployed on any platform that supports Java 8 or higher.
System memory	Minimum of 1 GB
Hard disk space	Minimum of 500 MB
Software	– Java 8 is required to install and deploy JDBC Gateway. – One of the following web browsers (with JavaScript support enabled) must be used to access the JDBC Gateway administrative console: - Google Chrome browser V50.0.2661.102 or later

System component	Requirement
	- Mozilla Firefox V47.0.1 or later - Microsoft Edge V25.10586.0.0 or later - Microsoft Internet Explorer V10 or later - Apple Safari browser V9.0.3 or later – Database connectivity requires an appropriate JDBC driver for each type of data source that is accessed.

About this task

Use the following procedure to install the JDBC Gateway. This installation installs the JDBC Gateway server and administrative console.

During the installation, you must specify a user ID to be used for the JDBC Gateway administrative console. When using the JDBC Gateway administrative console, only a single user can access the administrative console at a time.

As part of the installation, the following actions occur:

- The `jgate.properties` file is created, which contains the site-specific settings.
- Start and stop scripts appropriate to the platform are created. The installer creates `cmd` scripts if you are running on Windows and `sh` scripts if you are running on Unix or Linux.

Considerations for USS installation: For installation in USS, it is recommended that you define the following environment variables:

```
export IBM_JAVA_OPTIONS="-Dfile.encoding=ISO8859-1"
export _BPXK_AUTOCVT=ON
```

When the installer generates start and stop scripts, the following actions occur depending on these variables:

- If you have not set the recommended environment variables, the scripts will be generated in EBCDIC. You can run the gateway as normal for Unix using the following command: `sh startServer.sh`
- If you set the `IBM_JAVA_OPTIONS` variable, the scripts will be generated in ASCII, and you will need to use the following command: `chtag -tc ISO8859-1 <`*`file`*`>`. (Tagging in USS basically means `_BPXK_AUTOCVT` must be `ON` if you want to edit or execute the script in the shell.)

Files generated by the JDBC Gateway, such as log files and the `jgate.properties` file, will be generated in ASCII regardless of the aforementioned environment variable settings (except for `jetty.out`, which is in EBCDIC). In order to browse these files natively in USS, you must use the `chtag` command and set `_BPXK_AUTOCVT=ON`.

Procedure

1. Locate the JDBC Gateway installation file and copy it to your workstation.
2. From the z/OS mainframe, transfer the installation member *hlq*.SAVZBIN(AVZBINJ) to your workstation using the File Transfer Protocol (FTP) in binary mode.
3. Rename the file to `jdbc-gateway.zip`.
4. On your host machine, create a directory to host the JDBC Gateway, and then extract the contents of the installation file into that directory.

   The extracted contents will include the `JDBCGatewaySetup11.jar` file.

   Note: If your host machine does not have an unzip utility, extract the contents of the installation file on a Windows workstation and copy the `JDBCGatewaySetup11.jar` file to the host machine.
5. At a command prompt in the directory, run the following command:

   ```
   java -jar JDBCGatewaySetup11.jar
   ```

The installer launches.

6. Enter the following information at the prompts:

Prompt	**Description**
`You are about to install JDBC Gateway. Do you want to proceed? (Y/n)`	Enter Y to continue with the installation, or enter n to cancel the installation.
`Specify the installation directory (`*`local directory`*`\JDBCGateway):`	Enter the path of the directory where to install the application, or press Enter to use the default value as indicated.
`Set login for JDBC Gateway admin Web page (admin):`	Enter the user ID to be used for the JDBC Gateway administrative console, or press Enter to use the default value `admin`.
`Set password for JDBC Gateway admin Web page:`	Enter the password for the administrative console user ID. The password must be at least five characters in length.
`Confirm your password:`	Re-enter the password for the administrative console user ID.
`Set port for the Web UI (8080):`	Enter the number of an available TCP/IP port for the application, or press Enter to use the default value `8080`. This port number will be used when launching the JDBC Gateway administrative console.
`Installation completed. Do you want to start the JDBC Gateway now? (Y/n)`	Enter Y to start the server, or enter n to exit the installation. **Note:** If you enter Y, the server starts within the same shell.

Results

The JDBC Gateway has been installed and is ready for use. Information about the activity of the JDBC Gateway is available in the Java Console and in the log files.

If you specified to start the server, information about the startup process is displayed.

What to do next

- To start to the server, see "Starting the JDBC Gateway server" on page 93.
- To launch the administrative console, see "Launching the JDBC Gateway administrative console" on page 94.

Starting the JDBC Gateway server

Start the JDBC Gateway server so that you can connect directly to JDBC data sources.

Before you begin

The JDBC Gateway must be installed. See "Installing the JDBC Gateway" on page 91.

About this task

Use the following procedure to start the JDBC Gateway server.

Information about the startup and additional activity of the JDBC Gateway is available in the Java Console and in the following log file:

```
home_dir_for_user_profile\Application Data\IBM\JDBC Gateway\log\jetty.out
```

Procedure

1. At a command prompt in the JDBC Gateway installation directory, run one of the following commands:

   - For Windows: `startServer`
   - For Linux or Unix: `sh startServer.sh`

   Information about the startup process is displayed using the following format:

   ```
   Using settings file: home_dir_for_user_profile\Application Data\IBM\JDBC Gateway\Settings\jgate.properties
   Server is starting. It will be available on: http://localhost:port
   Server process ID: processID
   See home_dir_for_user_profile\Application Data\IBM\JDBC Gateway\log\jetty.out for server status
   information.
   ```

2. Wait for the JDBC Gateway server startup process to complete, which is indicated by the following message in the `jetty.out` log file:

   ```
   date time : JGATE Server started and ready to accept connections on port port_number
   ```

3. Optional: To stop the JDBC Gateway server, run the following command in the JDBC Gateway installation directory:

   - For Windows: `stopServer`
   - For Linux or Unix: `sh stopServer.sh`

Results

The JDBC Gateway server has been started and is ready for use. Information about the activity of the JDBC Gateway is available in the Java Console and in the log files.

What to do next

Start the JDBC Gateway administrative console. See "Launching the JDBC Gateway administrative console" on page 94.

Launching the JDBC Gateway administrative console

Launch the JDBC Gateway administrative console so that you can configure connections to JDBC data sources.

Before you begin

The JDBC Gateway server must be installed and active. See "Installing the JDBC Gateway" on page 91 and "Starting the JDBC Gateway server" on page 93.

About this task

Use the following procedure to start the JDBC Gateway administrative console.

Only a single user (web client) can access the JDBC Gateway administrative console at a time.

Note: The JDBC Gateway does not require an external web application server. It contains its own Jetty web application server.

Procedure

1. In a web browser, launch the JDBC Gateway administrative console using the following URL:

   ```
   http://server:port
   ```

   where:

   - *server* is the machine name or address where the JDBC Gateway server is running
   - *port* is the port specified during the installation
2. Enter the **Username** and **Password** specified during installation.

   The JDBC Gateway administrative console launches.

Results

The JDBC Gateway administrative console is running and ready for use. Information about the activity of the JDBC Gateway is available in the Java Console and in the log files.

What to do next

Configure access to data sources in the JDBC Gateway and the Data Virtualization Manager server. See "Configuring access to data sources using the JDBC Gateway" on page 97.

Using the JDBC Gateway

The *JDBC Gateway* is a Data Virtualization Manager distributed application server that allows direct connectivity to JDBC 4.0 data sources. The use of another federation server is not required.

Data sources

The JDBC Gateway solution is designed to work with any JDBC 4.0 compliant database. The following combinations of JDBC databases and drivers have been tested and verified to be supported by the JDBC Gateway:

- Hadoop 2.9.2 with the Hive 2.0 standalone JDBC driver
- Oracle 12 using the Oracle Thin Driver, version 6
- PostgreSQL version 11.1 using the JDBC driver version 42.2.5

Note:

1. The degree of JDBC compliance can vary across different driver vendor implementations and versions. In some cases, there may be interoperability problems when trying to use a particular JDBC driver to access a particular DBMS.
2. In Postgres, it is recommended not to use double quotes for the table names or the column names in the CREATE script. Using double quotes will make the identifier as case sensitive and result in a failed query if exact match is not given.

Getting started

Use the following procedure to access your first data source using the JDBC Gateway:

1. Install the JDBC Gateway. See "Installing the JDBC Gateway" on page 91.
2. Start the JDBC Gateway server. See "Starting the JDBC Gateway server" on page 93.
3. Launch the JDBC Gateway administrative console in a supported browser using the following URL:

   ```
   http://host:port
   ```

   See "Launching the JDBC Gateway administrative console" on page 94.
4. In the JDBC Gateway administrative console, perform the following steps:

   a. Determine the port that the JDBC Gateway will use for listening for incoming DRDA requests. You can review or change the port using the **Server Status** area of the JDBC Gateway administrative console. See "Using the JDBC Gateway administrative console" on page 96.

   b. Set up access to the data source by performing the following tasks:

      1) Locate and add JDBC driver information for the data source. See "Adding JDBC driver information for a data source" on page 97.

2) Create a data source definition entry, specifying the location name, driver, URL and user information. See “Creating a data source definition entry” on page 99.

5. In the Data Virtualization Manager server, set up access to the data source by performing the following tasks:

   a. Register the connection to the JDBC Gateway by entering the location, host and the port for the data source.

   b. Enable the SEF rules and set global variables for the data source.

   For information about these tasks, see “Configuring the Data Virtualization Manager server for JDBC Gateway sources” on page 100.

6. Use the Data Virtualization Manager studio to create virtual tables and views from the JDBC data source, just as you do for other supported sources, such as VSAM or IMS.

Using the JDBC Gateway administrative console

Use the JDBC Gateway administrative console to create and manage your data source definitions.

Before you begin

The JDBC Gateway must be installed, the JDBC Gateway server must be active, and the JDBC Gateway administrative console must be launched. See “Installing the JDBC Gateway” on page 91.

Procedure

Use the JDBC Gateway administrative console to create and manage your data source definitions. The following table describes the areas of the default JDBC Gateway view:

Field/Element	Action
Add New Data Source	Click this button add a new data source. For details, see “Creating a data source definition entry” on page 99.
Location **JDBC URL**	Displays a list of defined data sources. Select an entry to display properties and location information. • **Location**: Location name of the data source. **Note:** This value corresponds to the LOCATION parameter defined for the data source in the Data Virtualization Manager server. • **JDBC URL**: The URL that points to the data source.
Server Status	Displays and controls the JDBC Gateway server status and the DRDA listener port. • **Status**: JDBC Gateway server status. Click **Start** or **Stop** to control the server. • **Port**: The port on which the JDBC Gateway is listening for incoming DRDA requests. Click **Edit** to change the port number. This setting also allows you to control whether the server is started automatically when the JDBC Gateway `startServer` script is run. **Note:** This port value will be used when adding a JGATE database definition statement to the Data Virtualization Manager server configuration file (*AVZS*IN00).
Location Information	Displays the following details for selected data source entry: • **Domain**: Domain name of the JDBC Gateway. • **Location**: Name of the target data source. • **Port**: Port on which the JDBC Gateway is listening for incoming DRDA requests.

Field/Element	Action
	Note: These values will be used when adding a JGATE database definition statement to the Data Virtualization Manager server configuration file (*AVZS*IN00). Click **Test Connection** to test the connection to the data source. If you have specified any information incorrectly you will not be able to connect.

Configuring access to data sources using the JDBC Gateway

Configure access to JDBC data sources that will be accessed using the JDBC Gateway.

To configure access for a data source, you must complete the following steps:

1. Add the compliant JDBC driver for the data source to the JDBC Gateway. See "Adding JDBC driver information for a data source" on page 97.
2. Create the data source definition entry in the JDBC Gateway, specifying the location name, driver, URL, and user information. See "Creating a data source definition entry" on page 99.
3. Configure the Data Virtualization Manager server for the data source. See "Configuring the Data Virtualization Manager server for JDBC Gateway sources" on page 100.

Adding JDBC driver information for a data source

Add JDBC driver information to the JDBC Gateway.

Before you begin

The JDBC Gateway must be installed, the JDBC Gateway server must be active, and the JDBC Gateway administrative console must be launched. See "Installing the JDBC Gateway" on page 91.

About this task

The JDBC Gateway requires a compliant JDBC driver for each data source to be accessed. You must locate and add JDBC driver information for each data source. The driver files must be accessible to the JDBC Gateway. The JDBC Gateway retains the defined JDBC driver information, and you would only repeat this specification process to add new drivers or make changes to the properties of an existing driver.

In preparation for this task, obtain the following driver information for the data source from the data source vendor or from the driver documentation:

- Driver class name. For example: `org.postgresql.Driver`
- Driver JAR files
- URL format. Each data source type has a unique URL format that is used to access the data and is specific by vendor. For example, for Postgres: `jdbc:postgresql://{host}:{port}/{database}`

To add JDBC driver information to the JDBC Gateway, using the JDBC Gateway administrative console, you will define the driver library for the data source, and then add the driver files to the library. Use the following procedure to add JDBC driver information for a data source.

Procedure

1. In the JDBC Gateway administrative console, select **Preferences** > **JDBC Libraries**.

   The following table describes the areas of the page:

Area	Description
JDBC driver libraries	JDBC driver libraries that are already set up. Use the search bar to quickly locate information in the table.

Area	Description
Driver files	JAR files associated with selected driver library.
Details	Additional information about the selected driver library

2. Add a driver library by performing the following steps:
   a) Click the **Add Driver** button.
   b) In the **Add New Driver Library** window, provide the following information:

Field	Action
Enter new library name	Enter a name for the library. The JDBC driver information for each type of database is organized by libraries. It is recommended that the name that you specify describes the JDBC information that will be included in the library. For example, if you are adding JDBC driver information for accessing Postgres databases, you might call the library `Postgres`. However, this is a descriptive field and can include any text.
Driver class name	Specify the actual name of the driver class that will be used. This information can be found in your JDBC driver documentation. For example: `org.postgresql.Driver`
URL templates	Optional: Specify a generic example of a correctly formatted URL that could be used to connect to the database. For example, if you are adding JDBC driver information for accessing Postgres databases, you might specify the following JDBC URL template: `jdbc:postgresql://{host}:{port}/{database}`. The generic information as specified in the template is presented when you are adding data sources, where you will replace the generic information with the specific database information.

   Note: The **Validate** and **JDBC Driver Properties** options are not applicable until the driver files have been added.

   c) Click **OK**.
3. Add JDBC driver files to the library by performing the following steps:
   a) Click the **Add Driver Files** button.
   b) In the **Add Files** dialog, click **Add** and specify the path to the JDBC driver files to add.
   c) Click **OK**.
4. Optional: Update JDBC driver information as follows:

   - To edit the JDBC driver library information, validate the drivers, or add connection keywords, select an existing JDBC driver library from the list and click **Edit Driver**. The **Edit Driver Library** window opens where you can make changes to the library name, class name, and URL templates. You can also use the **Validate** option to validate the driver files, and the **JDBC Driver Properties** option to enter driver-specific connection keywords.
   - To remove a JDBC driver library, select an existing JDBC driver library from the list and click **Remove Driver**. The library, including all the JAR files that it contains, is removed.

- To remove a JAR file from a JDBC driver library, select an existing file from the list and click **Remove Driver File**. The file is removed.

5. Click **OK**.

Results

The JDBC driver information is saved.

Note: You must repeat this process for each JDBC driver that will be used to access a data source type.

What to do next

Create the data source definition entry, specifying the location name, driver, URL, and user information. "Creating a data source definition entry" on page 99.

Creating a data source definition entry

Configure the JDBC Gateway for access to data sources.

Before you begin

The JDBC Gateway must be installed, the JDBC Gateway server must be active, and the JDBC Gateway administrative console must be launched. See "Installing the JDBC Gateway" on page 91.

Also, the compliant JDBC driver should be added to the JDBC Gateway. See "Adding JDBC driver information for a data source" on page 97.

About this task

Use the following procedure to create a data source definition entry. This data source definition entry is made in the JDBC Gateway administrative console and is used for access to the data source by the JDBC Gateway.

Procedure

1. In the JDBC Gateway administrative console, click the **Add New Data Source** button.
2. In the **JDBC Gateway** dialog, complete the following fields.

Field	Action
Location	Enter the location name. A valid value is a string 1 - 16 characters. For example: ORCL. **Note:** This value must match the LOCATION value that will be specified for the corresponding data source definition in the Data Virtualization Manager server configuration file.
Connection Parameters	Enter the JDBC connection information, as follows: • **JDBC Driver**: Specify the library for the JDBC driver that will be used to access the data source. Select a library from the drop-down list, or click the ellipsis (...) option to the right of the field to open the **Select JDBC Driver** dialog where you can create additional JDBC driver libraries. (For more information, see "Adding JDBC driver information for a data source" on page 97.) • **JDBC URL**: Specify the URL that points to the data source to which you want to connect. The format for the URL can be displayed in the drop-down list if a JDBC URL template was supplied when the driver was configured. **Note:** You can also use the **Build URL by URL-Template** dialog box to form the correct string. Click **Build URL** to open the **Build URL by URL-Template** dialog box. From the JDBC URL drop-down list, select the template. In the table, specify the server, port, and database information and click OK. The

Field	Action
	result URL string is added to the JDBC URL list. This feature is available if a JDBC URL template was provided when the driver was configured. • **Advanced**: Click **Advanced** to specify any driver-specific connection string keywords and their values that will be used for the data source. The list of available advanced properties will change depending on both the type of driver being used, and the version of the driver. For information on any keywords that are required by a selected database driver, see the documentation for the driver.
Set User Information	Click **Set User Information** to provide authorization information used when accessing the data source. Provide the following information on the **User Information** dialog: • **User ID and password are required**: Select this option to require the use of a user ID and password when accessing the data source. If the data source allows access without a user ID and password, selecting this option will override that allowance. • **Allow users to save password**: Select this option to allow users to save passwords. • **Allow users to change password**: Select this option to allow users to change passwords. (**Note:** This option is for Db2 only.) • **User name** and **Password**: Specify the user ID and password that will be used to access the data source. The user ID and password that you specify when connecting to the data source are used to authorize the user.
Test Connection	Click **Test Connection** to test the connection to the data source. If you have specified any information incorrectly, you will not be able to connect.

3. Click **Finish**.

Results

The connection to the data source is validated. If successful, the data source location is added to the list of available data sources.

What to do next

Configure the Data Virtualization Manager server for the JDBC Gateway source.

Configuring the Data Virtualization Manager server for JDBC Gateway sources

Configure the Data Virtualization Manager server for use with the JDBC Gateway.

Before you begin

Configure access to the data source using the JDBC Gateway. See “Creating a data source definition entry” on page 99.

About this task

To use the JDBC Gateway to connect to your data source, the following changes must be made to the Data Virtualization Manager server:

- The DEFINE DATABASE TYPE value must be set, as follows:

```
"DEFINE DATABASE TYPE(JGATE)"
```

  JGATE
  DDF endpoint is the JDBC Gateway.

- Optionally, the following utility and SEF procedure can be configured in support of TYPE(JGATE):

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

AVZEJGAG
An ATH rule that switches credentials when connecting to a JGATE data source using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. In the Data Virtualization Manager server configuration file **(xVZyIN00)**, register the connection to the JDBC Gateway using a definition statement, such as the following example:

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

The following table lists the parameters:

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. 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 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.

Parameter	Description	Valid values
DOMAIN	The domain name or hostname on which the JDBC Gateway server is running. Either DOMAIN or IPADDR is required, but not both.	No default value.
IPADDR	The dot-notation IPV4 address of the host on which the JDBC Gateway server is running. Either DOMAIN or IPADDR is required, but not both.	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 JGATE: The location name specified in the JDBC Gateway data source definition entry. See "Creating a data source definition entry" on page 99. (*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 downstream database server.
PORT	The TCP/IP port on which the JDBC Gateway server is listening. (*Required*)	A valid 1-5 numeric string. If this keyword is not entered, the default DRDA port number 443 is used.
SECMEC	The DRDA security mechanism in force.	**EUSRIDPWD** Encrypt the user ID and password. **USRIDPWD** 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 password only.
TYPE	Defines the DDF endpoint type. **JGATE** DDF endpoint is the JDBC Gateway.	When using the JDBC Gateway, JGATE is the valid value.

2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:

   a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.

b) Modify the JCL according to the instructions provided in the AVZDRATH member.

   When adding the SYSIN statements that define the alternate credentials for logging in to your JDBC Gateway source, as instructed in the JCL, make sure to specify the correct DBTYPE. For JDBC Gateway sources, specify DBTYPE=JGATE.

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 AVZDRATH member and submit the job.

3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEJGAG) to provide the logon credentials to each JDBC Gateway data source instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
   a) On the Data Virtualization Manager server - 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.
4. Restart the Data Virtualization Manager server.

Results

The connection between the JDBC Gateway and the Data Virtualization Manager server for the JDBC data source has been defined.

What to do next

Use the studio to create virtual tables and views from the JDBC data source.

Example: Configuring access to Oracle data

Configure the JDBC Gateway for access to Oracle data.

Before you begin

The JDBC Gateway must be installed, the JDBC Gateway server must be active, and the JDBC Gateway administrative console must be launched. See "Installing the JDBC Gateway" on page 91.

About this task

Use the following procedure to configure access to Oracle data.

Procedure

1. Download the Oracle Thin Driver from the Oracle website. For example, `ojdbc8.jar`.
2. In the JDBC Gateway administrative console, select **Preferences** > **JDBC Libraries**, and then complete the following steps:
   a) Select the row for the **Driver Library Name** `Oracle Thin Driver` in the table, and click **Add Driver Files**.
   b) Use the **Add Files** dialog to add the Oracle Thin Driver file.
   c) Click **OK** to close the **JDBC Libraries** preference page.
3. Create a JDBC Gateway data source for Oracle as follows:
   a) Select **File** > **New** > **Other**, and then in the **New** wizard dialog, select **Data Source** and click **Next**.
   b) Complete the following fields:

Field	Action
Location	Enter the location name. For example, `Oracle`.

Field	Action
Connection Parameters	Enter the connection parameters: • **JDBC Driver**: From the drop-down list, select `Oracle Thin Driver`. • **JDBC URL**: Enter the JDBC URL as follows: `jdbc:oracle:thin:@//oracle-host:1521/ORCL`
Set User Information	Click **Set User Information**, and enter the credentials for accessing the Oracle database, as follows: • **User name**: *OracleUser* • **Password**: *OraclePwd*

c) Click **Test Connection**.

d) Click **Finish**.

4. In the Data Virtualization Manager server configuration file, register the connection to the JDBC Gateway data source using a definition statement, such as the following example:

```
"DEFINE DATABASE TYPE(JGATE)"                    ,
               "NAME(ORCL)"                      ,
               "LOCATION(Oracle)"                  ,
               "DDFSTATUS(ENABLE)"               ,
               "SECMEC(USRIDPWD)"                ,
               "PORT(1527)"                      ,
               "IPADDR(10.26.4.125)"             ,
               "CCSID(37)"
               "IDLETIME(110)"
```

   For details about this statement, see“Configuring the Data Virtualization Manager server for JDBC Gateway sources” on page 100.

5. In the Data Virtualization Manager server, enable rule AVZEJGAG. For more information, see “Configuring the Data Virtualization Manager server for JDBC Gateway sources” on page 100..

Results

The following connections have been established:

- The connection from the JDBC Gateway to the Oracle data source
- The connection between the JDBC Gateway and the Data Virtualization Manager server for the Oracle data source

What to do next

Use the studio to create virtual tables and views to access the Oracle data.

Example: Configuring access to Hadoop data

Configure the JDBC Gateway for access to Hadoop data.

Before you begin

The JDBC Gateway must be installed, the JDBC Gateway server must be active, and the JDBC Gateway administrative console must be launched. See “Installing the JDBC Gateway” on page 91.

About this task

Configuring access to Hadoop data requires both the standalone Hive 2.0 JDBC jar and the Hadoop Common jar driver files.

Use the following procedure to configure access to Hadoop data.

Procedure

1. Download the Apache Hive and Apache Hadoop driver files.
2. In the JDBC Gateway administrative console, select **Preferences** > **JDBC Libraries**, and then complete the following steps:

   a) Click **Add Driver**, complete the following fields, and click **OK**:

Field	**Action**
Enter new library name	Enter HADOOP
Driver class name	Enter `org.apache.hive.jdbc.HiveDriver`

   b) Select the row for the **Driver Library Name** HADOOP in the table, and click **Add Driver Files**.

   c) Use the **Add Files** dialog to add the driver files. You need to include both the standalone Hive 2.0 JDBC jar and the Hadoop Common jar.

   d) Click **OK** to close the **JDBC Libraries** preference page.
3. Create a JDBC Gateway data source for Hadoop as follows:

   a) Select **File** > **New** > **Other**, and then in the **New** wizard dialog, select **Data Source** and click **Next**.

   b) Complete the following fields:

Field	**Action**
Location	Enter the location name. For example, Hadoop.
Connection Parameters	Enter the connection parameters: • **JDBC Driver**: From the drop-down list, select HADOOP. • **JDBC URL**: Enter the JDBC URL as follows: `jdbc:hive2://hadoop-host:10000/default`
Set User Information	Click **Set User Information**, and enter the credentials for accessing the Hadoop database, as follows: • **User name**: *HadoopUser* • **Password**: *HadoopPwd*

   c) Click **Test Connection**.

   d) Click **Finish**.
4. In the Data Virtualization Manager server configuration file, register the connection to the JDBC Gateway data source using a definition statement, such as the following example:

```
/*---------------------------------------------------------*/
/*      HADOOP                                             */
/*---------------------------------------------------------*/
"DEFINE DATABASE TYPE(JGATE)"                     ,
               "NAME(HIVE)"                       ,
               "LOCATION(Hadoop)"                   ,
               "DDFSTATUS(ENABLE)"                ,
               "SECMEC(USRIDPWD)"                 ,
               "PORT(1527)"                       ,
               "IPADDR(10.26.4.125)"              ,
               "CCSID(37)"
               "IDLETIME(110)"
```

   For details about this statement, see“Configuring the Data Virtualization Manager server for JDBC Gateway sources” on page 100.
5. In the Data Virtualization Manager server, enable rule AVZEJGAG. For more information, see “Configuring the Data Virtualization Manager server for JDBC Gateway sources” on page 100..

Results

The following connections have been established:

- The connection from the JDBC Gateway to the Hadoop data source
- The connection between the JDBC Gateway and the Data Virtualization Manager server for the Hadoop data source

What to do next

Use the studio to create virtual tables and views to access the Hadoop data.

Setting preferences

The **Preferences** dialog is used to set user preferences and add necessary drivers.

The **Preferences** window consists of two panes. The left pane displays the list of preferences groups and the right pane displays the page for the selected group. The following groups of preferences are displayed in the **Preferences** window:

- JDBC Libraries
- Log
- Output

Setting JDBC driver preferences

Use the **JDBC Libraries** preferences to set up and manage JDBC driver information for your data sources.

About this task

You can use the **JDBC Libraries** preferences page to review, define or update JDBC driver information for each type of database (such as Db2®, Informix®, Oracle) that will be accessed.

Use the following procedure to access the **JDBC Libraries** preferences page. For details about adding new driver definitions, see “Adding JDBC driver information for a data source” on page 97.

Procedure

1. To access the **JDBC Libraries** page, select **Preferences** > **JDBC Libraries**.

   All of the JDBC driver libraries that you have already set up are listed in the **JDBC driver libraries** area. The JAR files associated with selected driver library are listed in the **Driver files** area. Additional information about the selected driver library is displayed on the **Details** panel.
2. For information about adding or editing driver definitions, see “Adding JDBC driver information for a data source” on page 97.

Setting log preferences

Use the **Log** page of the **Preferences** window to activate a log file that will track JDBC Gateway processing information.

About this task

The log file information can be useful in debugging.

It is recommended to leave the log level at the default setting of `error`. Only increase the level at the direction of IBM Software Support.

Use the following procedure to specify the log file preferences.

Procedure

1. Click **Preferences** > **Log**.
2. Check **Enable log** to activate the log file for debugging purposes. If this check box is selected, the log file option fields are enabled.

3. Check one or more of the log file options to indicate what information should be gathered. It is recommended that all options remain checked. The available log file options are as follows:
   - Print stack trace for log exceptions
   - Print log class and method
   - Print log user token
4. Click **Edit Log Categories** to modify the category level.

   The following levels are available: `none`, `emergency`, `alert`, `critical`, `error`, `warning`, `notice`, `info`, `debug`, `all`.
5. Click **Apply** to save your preferences choices.
6. Click **Restore Defaults** to restore the default preference values.
7. Click **OK** to close the **Preferences** window.

You can collect the generated log file and save it as a zip file to be able to send it for IBM Software Support.

To collect the generated log file:

8. Click **Help** > **Collect Support Data...**. In the **Collect Support Data** window, you can choose the option **All dates** if you want all the files generated from the beginning or specify a date range.
9. Click **Save Report**. This will save the log file as a zip file on your local system.

Setting output preferences

You can use the **Output** page of the **Preferences** window to activate the **Output** view that tracks the information about errors and connections in the JDBC Gateway.

About this task

The information from the **Output** view can be useful for debugging. It can be delivered as a report in the **Output** view and automatically added to the log file.

Use the following procedure to specify the output file preferences:

Procedure

1. Click **Preferences** > **Output**.
2. On the **Output** page, you can specify the following options:

   Show errors

   This option displays all error texts in the **Output** view.

   Show connection status

   This option displays the statuses of connections to data sources in the **Output** view.

   Automatically activate Output view

   When an error occurs or a message appears, this option automatically opens the **Output** view.
3. Click **Apply** to save your preferences choices.
4. Click **Restore Defaults** to restore the default preference values.
5. Click **OK** to close the **Preferences** window.

Troubleshooting

Collect troubleshooting data to provide to technical support.

About this task

Use the following procedure to collect troubleshooting data.

Procedure

1. Set the log level to `debug`. See "Setting log preferences" on page 106.
2. Reproduce the issue.

3. Set the log level to the previous value.
4. Select **Help** > **Collect Support Data**.
5. Complete the fields and click **Save Report**.

Chapter 8. Data sets

The following table lists the data sets that installation member INSTPAC creates.

The installation data sets have the format *hlq.data_set_name_suffix*, where *hlq* is the high level qualifier and *data_set_name_suffix* is as listed in the table.

Any data set prefixed with the subsystem ID (*SSID*) is a data set created during post-installation to allow user modification. These post-installation data sets have the format *hlq.SSID.data_set_name_suffix*.

Table 7. Data sets created by INSTPAC

Data set name suffix	SMP/E data type	SMP/E data definition	Data set organi- zation	Record format	Logical record length	Block size	tracks
AAVZCNTL	USER2	ASDBCNTL	PO-E	FB	80	32720	180
AAVZDBRM	USER1	ASDBDBRM	PO-E	FB	80	32720	300
AAVZEXEC	EXEC	ASDBEXEC	PO-E	FB	80	32720	450
AAVZHTML	TEXT	ASDBHTML	PO-E	VB	32765	32760	15
AAVZHTX	PRODXML	ASDBHTX	PO-E	VB	19036	19040	15
AAVZJLOD	PROGRAM	ASDBJLOD	PO-E	U	0	6000	15
AAVZLIST	UTOUT	ASDBLIST	PO-E	FBA	133	32718	45
AAVZMAP	DATA	ASDBMAP	PO-E	FB	1024	31744	15
AAVZMLIB	MSG	ASDBMLIB	PO-E	FB	80	32720	15
AAVZMOD	MOD	ASDBMOD	PO-E	U	0	6144	1800
AAVZOBJ	USER3	ASDBOBJ	PO-E	FB	80	32720	1275
AAVZPLIB	PNL	ASDBPLIB	PO-E	FB	80	32720	105
AAVZRPC	PROGRAM	ASDBRPC	PO-E	U	0	6000	750
AAVZSAMP	SAMP	ASDBSAMP	PO-E	FB	80	32720	270
AAVZSLIB	SKL	ASDBSLIB	PO-E	FB	80	32720	15
AAVZSWI	DATA	ASDBSWI	PO-E	FB	80	32720	30
AAVZTLIB	TBL	ASDBTLIB	PO-E	FB	80	32720	15
AAVZXATH	EXEC	ASDBXATH	PO-E	FB	80	32720	15
AAVZXCMD	EXEC	ASDBXCMD	PO-E	FB	80	32720	15
AAVZXEXC	EXEC	ASDBXEXC	PO-E	FB	80	32720	15
AAVZXGLV	EXEC	ASDBXGLV	PO-E	FB	80	32720	15
AAVZXPUB	EXEC	ASDBXPUB	PO-E	FB	80	32720	15
AAVZXRPC	EXEC	ASDBXRPC	PO-E	FB	80	32720	15
AAVZXSQL	EXEC	ASDBXSQL	PO-E	FB	80	32720	15
AAVZXTOD	EXEC	ASDBXTOD	PO-E	FB	80	32720	15
AAVZXVTB	EXEC	ADVSXVTB	PO-E	FB	80	32720	15

Table 7. Data sets created by INSTPAC (continued)

Data set name suffix	**SMP/E data type**	**SMP/E data definition**	**Data set organi-zation**	**Record format**	**Logical record length**	**Block size**	**tracks**
SAVZXATH	EXEC	SSDBXATH	PO-E	FB	80	32720	15
SAVZCLOD	MOD	SSDBCLOD	PO-E	U	0	6000	270
SAVZXCMD	EXEC	SSDBXCMD	PO-E	FB	80	32720	15
SAVZCNTL	USER2	SDBCNTL	PO-E	FB	80	32720	180
SAVZDBRM	USER1	SSDBDBRMX	PO-E	FB	80	32720	15
SAVZEXEC	EXEC	SSDBXEXC	PO-E	FB	80	32720	15
SAVZEXEC	EXEC	SSDBEXEC	PO-E	FB	80	32720	450
SAVZXGLV	EXEC	SSDBXGLV	PO-E	FB	80 3	2720	15
SAVZHTML	TEXT	SSDBHTML	PO-E	VB	32756	32760	15
SAVZHTX	PRODXML	SSDBHTX	PO-E	VB	19036	19040	15
SAVZINST	SMP/E does not maintain this data set.		PO-E	FB	80	32720	18
SAVZJLOD	PROGRAM	SSDBJLOD	PO-E	U	0	6000	15
SAVZLIST	UTOUT	SSDBLIST	PO-E	FBA	133	32718	45
SAVZLOAD	MOD	SSDBLOAD	PO	U	0	6000	4500
SAVZMAP	DATA	SSDBMAP	PO-E	FB	1024	31744	15
SAVZOBJ	USER3	SSDBOBJ	PO-E	FB	80	32720	1125
SAVZXPUB	EXEC	SSDBXPUB	PO-E	FB	80	32720	15
SAVZXRPC	EXEC	SSDBXRPC	PO-E	FB	80	32720	15
SAVZRPC	PROGRAM	SSDBRPC	PO-E	U	0	6000	375
SAVZSAMP	SAMP	SSDBSAMP	PO-E	FB	80	32720	300
SAVZMDL1	SMP/E does not maintain this data set.	SDBLMOD	PO-E	U	0	6000	18
SAVZMDL2	SMP/E does not maintain this data set.	SDBLMOD2	PO-E	U	0	6000	18
SAVZMENU	MSG	SHDWMLIB	PO-E	FB	80	32720	15
SAVZPENU	PNL	SHDWPLIB	PO-E	FB	80	32720	210
SAVZSLIB	SKL	SHDWSLIB	PO-E	FB	80	32720	15
SAVZTENU	TBL	SHDWTLIB	PO-E	FB	80	32720	15
SAVZXSQL	EXEC	SSDBXSQL	PO-E	FB	80	32720	15
SAVZXWWW	DATA	SSDBSWI	PO-E	FB	80	32720	30

Table 7. Data sets created by INSTPAC (continued)

Data set name suffix	SMP/E data type	SMP/E data definition	Data set organi-zation	Record format	Logical record length	Block size	tracks
SWI.OBJ1	SMP/E does not maintain this data set.		PO-E	VB	4092	12288	30
SAVZXTOD	EXEC	SSDBXTOD	PO-E	FB	80	32720	15

Note: SMP/E does not maintain this data set.

Index

GC27-8874-00