

IBM Data Virtualization Manager for z/OS
1.1

Installation Guide



Note

Before using this information and the product it supports, read the information in [“Product legal notices” on page 45.](#)

This edition applies to Version 1 Release 1 of IBM Data Virtualization Manager for z/OS (5698-DVM) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2022-07-26

© **Copyright International Business Machines Corporation 2017, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© **Rocket Software, Inc. 2017, 2022.**

Contents

- Tables..... V**

- About this information..... vii**

- How to send your comments to IBM.....ix**
 - If you have a technical problem..... ix

- Chapter 1. System requirements..... 1**

- Chapter 2. Prerequisites..... 3**
 - Required naming conventions..... 3
 - Creating server data sets..... 3
 - Defining security authorizations..... 4
 - Configuring Workload Manager (WLM)..... 5
 - APF-authorizing LOAD library data sets..... 5
 - Copying target libraries..... 6

- Chapter 3. Installing and configuring the server..... 7**
 - Installing the Data Virtualization Manager server.....7
 - DVM installation using z/OSMF workflow (Automated manner)..... 7
 - Downloading and untarring files..... 7
 - Configuring workflows..... 8
 - Creating server..... 8
 - Configuration file (wkAVZconfig.xml)..... 10
 - Templates used for variable file..... 12
 - Installing server maintenance..... 17
 - Configuring support for code pages and DBCS 17
 - Customizing the server initialization member..... 18
 - Configuring the started task JCL..... 19
 - Configuring the ISPF application..... 20
 - Configuring generation data set retrieval..... 21
 - Configuring delimited data support..... 22
 - Configuring alternate started task ID..... 23
 - Verifying the Data Virtualization Manager server installation..... 24
 - Invoking the ISPF application using the command shell..... 24
 - Invoking the ISPF application using the subsystem manager..... 25
 - Support for the IBM i series as a DRDA or JDBC data source..... 26

- Chapter 4. Installing client drivers..... 27**
 - Installing the ODBC driver..... 27
 - Installing OBDC docker file for zCX..... 28
 - Sample python programs..... 29
 - Executing a query using scodbcdm64 tool..... 30
 - Installing the JDBC driver..... 30

- Chapter 5. Installing Data Virtualization Manager studio.....31**
 - Changing the workspace in studio..... 32
 - Verifying the studio installation..... 33
 - Changing the language used in studio..... 33

 Chapter 6. Installing JDBC Gateway.....	35
 Chapter 7. Installing DVM interface for ACI.....	39
Accessibility features.....	43
Product legal notices.....	45
Index.....	49

Tables

- 1. Data virtualization prerequisites..... 3
- 2. Access requirements by data definition name..... 4

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 \(support.ibm.com\)](https://support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Chapter 1. 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 OA441111:
 - z/OS 1.13 UA75045
 - z/OS 2.1 UA75046
- VSAM:
 - z/OS 1.13 UA75272
 - z/OS 2.1 UA75273

Chapter 2. Prerequisites

This section covers the prerequisites for installing IBM Data Virtualization Manager for z/OS.

Before you begin to install and configure IBM Data Virtualization Manager for z/OS, familiarize yourself with the tasks that enable you to successfully install and configure the the data virtualization solution that satisfies your requirements. The following table describes each significant prerequisite task.

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 3.
2	Create the server data sets using the <i>hlq.SAVZCNTL</i> members AVZDFDIV, AVZGNMP1 and AVZEXSWI.	See “Creating server data sets” on page 3.
3	Set up the security application to use with the server using one of the following <i>hlq.SAVZCNTL</i> members: AVZRAVDB, AVZA2VDB, AVZTSVDB.	See “Defining security authorizations” on page 4.
4	Configure Workload Manager (WLM) for optimum performance of the server.	See “Configuring Workload Manager (WLM)” on page 5.
5	APF-authorize the product LOAD library data sets.	See “APF-authorizing LOAD library data sets” on page 5.
6	Create a copy of the product libraries (optional).	See “Copying target libraries” on page 6.

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 *xVZy*, 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, *xVZyIN00*.

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.

Note:

AVZGNMP1 creates a copy of the RULES outside of SMP/E. Do not edit the SMP/E copy of these rules. Should a new maintenance touches these rules, see HOLD ACTION for next step about updating the rules created from running AVZGNMP1job.

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:

<i>Table 2. Access requirements by data definition name</i>		
Data definition name	Access	Data set name
STEPLIB	READ, EXECUTE	<i>hlq.SAVZLOAD</i>
AVZRPCLB	READ, EXECUTE	<i>hlq.SAVZRPC</i>
SYSEXEC	READ	<i>hlq.SAVZEXEC</i>
AVZTRACE	READ, WRITE	<i>hlq.TRACE</i>
AVZCHK1	READ, WRITE	<i>hlq.SYSCK1</i>

Data definition name	Access	Data set name
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 h1q . SAVZLOAD.

- AVZRPCLB

You must authorize the LOAD library h1q . 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.

Chapter 3. Installing and configuring the server

This section contains installation information about the server, studio and drivers.

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 the [program directory](#) to install the Data Virtualization Manager server on your system .

DVM installation using z/OSMF workflow (Automated manner)

The automated way of DVM installation uses a series of z/OS Management Facility (z/OSMF) workflow files and templates to invoke z/OSMF for configuring the DVM server and its components.

Workflows simplify and streamline the configuration of the Data Virtualization Manager by taking inputs from the user and populate the panel fields with the values. The templates can also be submitted as batch jobs on z/OS.

Follow the instructions mentioned in the document to install Data Virtualization Manager using automated z/OSMF automated workflow.

Downloading and untarring files

Follow the instructions in this section to download and untar the files.

Before you begin

Note: The minimum PTF level required is UI77011 for configuring workflows to install DVM using z/OSMF.

About this task

After applying the PTF, AVZZOSMF member should be available in hlq.SAVZCNTL library. AVZZOSMF is a TAR file containing zOSMF files.

To untar the zOSMF files carry out the steps mentioned in this task.

Procedure

1. Go to the ISPF panel, enter option 3.4 or 3.17 and then enter the path where you want to save the files. For example, - /user/name/.
2. On the command line enter a forward slash “/”.

The command entry panel appears.

3. Issue `/bin/tar -xvf '//hlq.savzcntl(avzzosmf)'` .

Note: The quotes should be “double” ‘single’ ‘single’ “double”.

Results

This command will untar the zOSMF files to your current path within a sub directory/zosmf. The untar files should be copied from /zosmf directory to the path which will be referred to while proceeding with further steps.

Configuring workflows

This section contains detailed information about creating and configuring workflow to install DVM.

Before you begin

Ensure that the following conditions are met before start configuring DVM.

- The z/OSMF server must be installed, configured and running.
- The user must have login credentials to access the z/OSMF server.
- The **SAVZXOBJ** member must be present.
- The workflow definition and workflow variable files must be ready.

About this task

z/OSMF workflows requires the following files to be present within the HFS directory, which is a new directory created to store the z/OSMF workflow.

- **Var** file: A var or variable file contains a list of all the variables used in the configuration process. Each variable resolves to a value supplied by the user when the batch templates are submitted. Variables are used in both the config file and the templates. The var file is imported in the Workflow variable field.
- **Config** file: A config or configuration file is the XML schema file that defines the workflow. This file includes the steps within the workflow, the templates that the workflow customizes, and the variables that are used in each step and in each template. The config file references the variable file, so all variables in the config file must be defined in the variable file. The config file is imported in Workflow definition field.

Procedure

1. Click **Workflows** and select **Actions->Create Workflow...**
2. Provide the path to the configuration file and the variable file in the fields **Workflow definition file:** and **Workflow variable input file:** respectively.
 - a) For the **Workflow definition file** field: enter the full qualified path to the wfAVZconfig.xml file.
 - b) For the **Workflow variable input file** field: enter the full qualified path to the AVZvarfile.txt file.

This field is optional, and specifying the AVZvarfile.txt file will automatically populate values for the required variables in the configuration process. If you do not specify the AVZvarfile.txt file, you must complete all mandatory fields manually. It is recommended that you specify the AVZvarfile.txt file.
3. Click **Next**.
4. In the continued **Create Workflow** page, provide the following details.
 - **Workflow name:** Enter a name to the created workflow.
 - **Owner user ID:** Verify the user ID shown and modify if required.
 - Check **Assign all steps to owner user ID**.
5. Click **Finish**.

Creating server

This section contains the information required to declare variables and trigger the JCL jobs.

Procedure

After the workflow is created, perform the following tasks to create a server.

1. Click **Declare parameters** and then **Perform**. Provide the following values:
 - **High level qualifier: - Data Virtualization Manager product library:** Name of the product library e.g. **CSD.QA.AVZ**.

- **High level qualifier for IN00: - Data Virtualization Manager server initialization library:** Name of the IN00 file e.g. **HLQ.SAVZEXEC**.
- **High level qualifier for INEF: - Data Virtualization Manager server initialization library:** Name of the INEF file e.g. **HLQ.SAVZEXEC**.
- **High level qualifier for PROC: - Data Virtualization Manager server procedure library:** Name of the procedure library e.g. **HLQ.PROCLIB**.
- **APF Authorization DSN LIB 1: - Data Virtualization Manager Server APF for 1st DSN LIB name:** APF authorization library name e.g. **HLQ.SAVZLOAD**.
- **APF Authorization DSN LIB 2: - Data Virtualization Manager Server APF for 2nd DSN LIB name:** APF authorization library name e.g. **HLQ.SAVZRPC**.
- **Server started task identifier: - Started task ID:** Name of the created server e.g. **AVZA**
- **UNIT: - DASD unit for work data set:** The type of unit to use for allocating temporary data sets. The valid values are **SYSDA, SYSALLDA, VIO**, and the default value is **SYSDA**.
- **UNIT: - DASD unit for CREATE data set:** The type of unit to use for allocating temporary data sets. The valid values are **SYSDA, SYSALLDA, VIO**, and the default value is **SYSALLDA**.
- **Region size: - Region size the server will use:** The region size of the server. The default value is **0M**.
- **Virtual storage: - Virtual storage above the bar:** The virtual storage in megabytes, for the server. The default value is **200M**.

Click **Next**, and provide the following values.

- **Trace file primary quantity: - Trace file primary space allocation:** Trace file primary quantity in cylinders.
- **Trace file secondary quantity: - Trace file secondary space allocation:** Trace file secondary quantity in cylinders.
- **Global variable checkpoint file primary quantity: - Global variable checkpoint file primary space allocation:** Global variable checkpoint file primary quantity in cylinders.
- **Global variable checkpoint file secondary quantity: - Global variable checkpoint file secondary space allocation:** Global variable checkpoint file secondary quantity in cylinders.
- **Maximum number of messages: - Maximum messages:** Trace file message size.
- **IN00 configuration member: - Started task IN00:** Product initialization REXX program. The default value is **AVZSIN00**.
- **Server started task identifier: - Started task INEF:** Sample SEF initialization REXX program. The default value is **AVZ1INEF**.
- **Proc configuration member: - Started task PROC:** Creates a member to generate proc for the user.
- **Server proc name: - Server started task proc name:** Server task proc name. This value is mandatory, and the default value is **AVZ1PROC**.
- **TCP/IP OE Port: - OE Port Number:** Listening TCP/IP port number for the server. This value is required.
- **Web Services Port: - Web Services Port:** Web Service port number that the studio client uses to communicate with the server. This value is required. The default value is **1201**.
- **SMF record number: - SMF Record Number:** Indicates the SMF record type to be created. This value is optional, and the default value is **249**.
- **Enable IAM support: - Enable support for IAM VSAM files:** This parameter specifies whether to call the interface module for IAM support to analyze keys and set ranges for MapReduce. Valid values are **Yes** and **No**, and the default value is **No**.

Click **Next**, and provide the following values.

- **Db2 high level qualifier: - Db2 subsystem library:** Library for Db2 subsystem.
- **Db2 SSID: - Db2 Subsystem Identifier :** Db2 subsystem name. The default value is **DSN1**.

- **Subsystem port number: - Db2 subsystem port number:** Port number for DRDA access. This value is required, and the default value is 443.
- **Db2 DDF Status: - Db2 Subsystem Status :** DDF status to Db2 database subsystems. The default value is ENABLE.
- **Db2 DDF Location: - Db2 Subsystem Location :** Location of the Db2 subsystem, and the default value is DEV1DSN1.
- **Domain name: - Subsystem domain name:** Domain name of the Db2 subsystem being configured. This value is optional, and the default value is DB2DOMAIN.
- **CCSID value: - Db2 subsystem CCSID value:** CCSID value of the Db2 subsystem being configured. The CCSID value can be found in the Db2 installation panel DSNTIPF, option 7. The default value is 37.
- **LU Name: - Dbe subsystem LU name:** LU name of the Db2 subsystem to be used for RACF pass ticket generation. This value can be found in the master address space of the subsystem configured or by issuing the command **DISPLAY DDF**. The default value is DSN1LU.
- **Bind owner: - Db2 plan BIND owner:** Db2 user ID used as the owner in the BIND job. The default value is DB2USER.
- **Bind collection ID: - Db2 collection identifier:** Collection ID for the DRDA packages used by the server. Use this value in the DVM studio when setting up the connection to the server. The default is DB2USER.
- **Db2 connection ID: - Db2 connection identifier:** Connection ID to use in the BIND job for connecting to Db2 subsystems on z/OS. The default value is DSN1.
- **Db2 SDSNCONNID: - Db2 SDSNCONNID library:** SDSNEXIT library to use while binding packages. This value is required if Db2 is the data source. The default value is DSN1.
- **Db2 SDSNSNETP2: - Db2 SDSNSNETP2 library:** Db2 plan name. The default value is DSNTIAD.
- **JOBLIB DSNSYS: - JOBLIB DSNSYS library:** Db2 subsystem name. The default value is DSN1.

Click **Next** and then **Finish**

2. For the steps 2-7, perform the following.

- Click the step name and then click **Perform**.
- Review the JCL and modify it if required. Check the box **All steps in workflow '*workflow_name*'**. if the changes should be reflected in all steps or check the box **This step '*step_name*' only**. if the changes should be reflected only in this step. *workflow_name* is the name of the created workflow, and the *step_name* is the name of the current step.
- Click **Next**. Edit the JCL if required.
- Click **Next**.
- Check **Submit JCL** to submit the JCL file and/or check **Save JCL** to save the JCL file as z/OS UNIX file or z/OS data set.
- Click **Finish** to submit and/or save the JCL file.

Results

Once you complete the steps 1-7, the JCL files are submitted and the server with the given settings is created.

Configuration file (wkAVZconfig.xml)

This section describes how to configure the workflow definition file (wkAVZconfig.xml).

1. Declare variables: This is the first step that the user needs to perform. This step allows the user to enter the parameter value in the categorized fields. The fields will be populated with default values. User can replace it based on their needs.
2. Data Virtualization Manager Server Files: This step customizes the AVZFiles.template. This template creates and runs a batch job that creates and customizes the global variable and the system check

data sets that the server uses. The customized batch job is copied to the hlq.SAVZCNTL product data set, where 'hlq' is the high level qualifier of the product data sets. As a user, you need to place the map at hlq.avzid.savzmap. This job will delete if hlq.avzid.SAVZ* data set exists and create the following data sets:

- HLQ.AVZID.SAVZMAP
 - HLQ.AVZID.SAVZXATH
 - HLQ.AVZID.SAVZXCMD
 - HLQ.AVZID.SAVZXEXC
 - HLQ.AVZID.SAVZXOBJ
 - HLQ.AVZID.SAVZXSQL
 - HLQ.AVZID.SAVZXTOD
 - HLQ.AVZID.SAVZXVTB
 - HLQ.AVZID.SAVZXAPF
 - HLQ.AVZID.SYSCHK1
 - HLQ.AVZID.SYSCHK1.DATA
 - HLQ.AVZID.TRACE
 - HLQ.AVZID.TRACE.DATA
3. Server Initialization: This step customizes the AVZIN00.template. On submitting this template, it will generate AVZxIN00 and AVZxINEF files. This template customizes the server parameter file AVZxIN00, where AVZx is the four-character DVM ID used in customizing the server. This file contains parameters that the server uses to access the data sources. The customized batch job is copied to the hlq.SAVZEXEC product data set, where hlq is the high level qualifier of the product data sets.
 4. Data Virtualization Manager APF Authorization: This step performs APF authorization of load and RPC libraries that are required for the Server. APF Authorization is done dynamically using SETPROG command. This step also creates a member called APFATH in HLQ.AVZID.SAVZXAPF dataset. The content of APFATH must be copied to SYSIBM.PARMLIB by system administrator to retain APF authorization of LOAD and RPC libraries even after an IPL is completed.
 5. Data Virtualization Manager server proc: This step customizes the AVZServer.template. This template customizes the server started task proc avzidPROC, where avzid is the four-character DVM ID used in customizing the server. This is the server that is started when using the Data Virtualization Manager. The customized batch job is stored in the hlq.SAVZCNTL product data set, where hlq is the high level qualifier of the product data sets.
 6. Data Virtualization Manager bind job for CAF/RRSAF: This is an optional step. This step customizes the AVZbindc.template. This template creates and runs a bind job when using Db2 for CAF/RRSAF. This customized bind job is stored in the hlq.SAVZCNTL product data set, where hlq is the high level qualifier of the product data sets. It generates the data set hlq.SAVZDBRM.
 7. Data Virtualization Manager bind job for DRDA: This is an optional step. This step customizes the templateAVZbindd.template. This template creates and runs a bind job when using Db2 for DRDA. This customized bind job is stored in the hlq.SAVZCNTL product data set, where hlq is the high level qualifier of the product data sets. Note that Db2 Bind and DRDA job expects plan or packages to be available during plan or package free step, otherwise a return code of 08 is shown for the first time.

Templates used for variable file

This section contains the information about the various templates used to assign the value of the variables.

No	Variables	Description	Template Used
1	AVZHLQ	<p>This parameter is High level qualifier of SMPE product library. This value is required.</p> <p>It is expected that after the SMPE install, the following libraries are available:</p> <ul style="list-style-type: none"> • AVZHLQ.SAVZBIN • AVZHLQ.SAVZCNTL • AVZHLQ.SAVZDBRM • AVZHLQ.SAVZEXEC • AVZHLQ.SAVZLOAD • AVZHLQ.SAVZMAP • AVZHLQ.SAVZMENU • AVZHLQ.SAVZOBJX • AVZHLQ.SAVZPENU • AVZHLQ.SAVZRPC • AVZHLQ.SAVZSAMP • AVZHLQ.SAVZSLIB • AVZHLQ.SAVZSMAP • AVZHLQ.SAVZTENU • AVZHLQ.SAVZXATH • AVZHLQ.SAVZXCMD • AVZHLQ.SAVZXEXC • AVZHLQ.SAVZXSQL • AVZHLQ.SAVZXTOD • AVZHLQ.SAVZXVTB • AVZHLQ.SAVZXOBJ 	<ul style="list-style-type: none"> • AVZFiles.template • AVZbindc.template • AVZbindd.template • AVZIN00.template
2	DASDUNIT	<p>This parameter specifies the type of unit to use for allocating temporary data sets. This value is required. The default value is SYSDA.</p>	<ul style="list-style-type: none"> • AVZbindc.template • AVZbindd.template • AVZFiles.template • AVZServer.template

No	Variables	Description	Template Used
3	AVZTRCP	This parameter defines the primary quantity in cylinders for the trace file. The trace file must be large enough to contain the number of messages specified in the parameter Maximum rows to contain.	AVZFiles.template
4	AVZTRCS	This parameter defines the secondary quantity in cylinders for the trace file.	AVZFiles.template
5	AVZAPPL	This parameter specifies location of the tDb2 subsystem that the server is to use as a data source.	AVZIN00.template
6	AVZCHKP	This parameter defines the primary quantity in cylinders for the trace file. The trace file must be large enough to contain the number of messages specified in the parameter Maximum rows to contain.	AVZFiles.template
7	AVZCHKS	This parameter defines the secondary quantity in cylinders for the global variable checkpoint file.	AVZServer.template, AVZFiles.template
8	AVZBMAX	This parameter specifies the maximum number of messages to retain in the trace file. As the server processes data, messages are written to the trace file. Older messages roll off as newer messages are written. This value is required. The default value is 300000	AVZIN00.template

No	Variables	Description	Template Used
9	AVZID	This parameter identifies the server started task and is also known as the AVZID. Many servers can be running, and each can communicate with different data sources. This value identifies the servers from each other. This value is required. The default value is AVZ1.	AVZServer.template, AVZIN00.template
10	AVZREGIONSZ	This parameter specifies the region size that the server will use as it processes data. This value is required. The default value is 0M.	AVZIN00.template
11	AVZMEM	This parameter specifies the virtual storage above the bar, in megabytes, for the server. This value is required. The default value is 200.	AVZIN00.template
12	AVZOEPORT	This parameter defines the Open Edition TCP/IP port number on the host on which the server listens to service ODBC/JDBC requests. OE sockets can run over OE TCP/IP, z/OS TOPCP/IP, and other TCP/IP implementations. This value is required. The default value is 1200.	AVZIN00.template
13	AVZWSOE	This parameter defines the Web Service port number that the studio client will use to communicate with the server. This value is required. The default value is 1201.	AVZIN00.template

No	Variables	Description	Template Used
14	AVZSMFR	This parameter specifies the SMF number for the server to use as it creates SMF records. This value is optional. Leaving this value blank will result in writing no SMF records by the server. For more information in SMF subtype records, refer to the user guide. There is no default value.	AVZIN00.template
15	AVZIAM	This parameter specifies whether to call the interface module for IAM support to analyze keys and set ranges for MapReduce. Valid values are Yes and No. The default value is No.	AVZIN00.template
16	AVZDB2SSID	This parameter specifies the Db2 subsystem name that the server is to use as a data source.	AVZIN00.template
17	AVZDB2PORT	This parameter specifies the port number to use for DRDA access for this subsystem. This value is required. The default value is 443.	AVZIN00.template
18	AVZDB2DDF	This parameter specifies whether DDF status to DB2 database subsystems is enabled.	AVZIN00.template
19	AVZDB2DN	This parameter specifies the fully-qualified name that identifies the domain name of the Db2 subsystem being configured. The domain name is used instead of the IP address. This value is optional. There is no default value.	AVZIN00.template

No	Variables	Description	Template Used
20	AVZDB2CCSID	This parameter specifies the CCSID of the value of the Db2 subsystem being configured. The CCSID value of this subsystem can be found in the Db2 installation panel DSNTIPF, option 7. This value is required. The default value is 37.	AVZIN00.template
21	AVZDB2APPL	This parameter specifies the LU name of the Db2 subsystem being configured and is used for RACF pass ticket generation. This value can be found in the master address space of the subsystem being configured or by issuing the command DISPLAY DDF. This value is required. The default value is DSN1LU.	AVZIN00.template
22	AVZBNDO	This parameter specifies the Db2 user ID that will be used as the OWNER in the BIND job. This parameter is required. The default is DB2USER.	AVZbindd.template
23	AVZCOLL	This parameter defines the collection ID to use for the DRDA packages used by the server. Use this value in the Data Virtualization Manager studio when setting up the connection to the server. This value is required. The default is DB2USER.	AVZbindd.template
24	AVZCONNID	This parameter defines the connection ID to use in the BIND job for connecting to Db2 subsystems on z/OS. This value is required.	AVZbindc.template, AVZbindd.template
25	DB2LIB	This parameter specifies the library for Db2 subsystem.	AVZbindc.template, AVZbindd.template

No	Variables	Description	Template Used
26	AVZIN00	This variable will create the member to specify the product initialization REXX program.	AVZIN00.template
27	AVZINEF	This variable will create the member to specify the sample SEF initialization REXX program.	AVZIN00.template
28	AVZSERV	This parameter will create a member to generate the proc for the user.	AVZServer.template
29	AVZDB2CONNID	This parameter specifies the Db2 connection ID.	AVZbindd.template
30	AVZDSNTEP2	This parameter specifies the Db2 plan name.	AVZbindd.template
31	DSNSYS	This parameter specifies the Db2 subsystem name.	AVZbindd.template, AVZbindc.template
32	UNIT1	This parameter specifies the type of unit to use for allocating create data sets. This value is required. The default value is SYSALLDA.	AVZFiles.template
33	APFLIB1	LOAD file for APF Authorization	AVZfiles.template AVZAPFAUTH.template
34	APFLIB2	RPC file for APF Authorization	AVZfiles.template AVZAPFAUTH.template

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.

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 6](#).
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: <ul style="list-style-type: none">• 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 6](#).

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".

- 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

- 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.

- Include a job step that invokes BPXTCAFF. BPXTCAFF is invoked as a job step in front of an existing program in a started procedure or submitted job stream. For example:

```
//STEP0 EXEC PGM=BPXTCAFF,PARM=TPNAME
//REALSTEP EXEC PGM=MYPGM,PARM='MyParms'
```

The desired transport is specified with the PARM= keyword and must be 1 to 8 uppercase characters. If PARM= is not supplied, or is blank, then the address space's transport affinity will be reset to no transport selected. This can also be specified as PARM=&VAR, where VAR is a PROC keyword that is passed in from the Start command or is a static system symbol.

BPXTCAFF sets transport affinity for an address space for the duration of that address space or job. This affinity persists over job steps within the job, persists over UNIX process termination and redubbing, and applies to all UNIX processes running within that address space. BPXTCAFF is intended for use with non-C or POSIX(OFF) programs where the `_BPXK_SETIBMOPT_TRANSPORT` environment variable is not effective. It is also intended for programs that do not make their own calls to `setibmopt()` or `BPX1PCT` or that cannot be modified to do so. BPXTCAFF exits with one of the following return codes in register 15.

0 - Successful. TPNAME matched an AF_INET socket transport.

2 - Minor failure. TPNAME did not match any transport but CINET is not configured so transport affinity is moot.

8 - Failure. CINET is configured and TPNAME did not match any transport running under CINET.

12 - Failure. The interface to the routine was not valid.

- 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 3. The default name is *hlq.SAVZEXEC(AVZSIN00)*.

Note:

AVZGNMP1 creates a copy of the SEF RULESET (ATH, CMD, EXC, RPC, SQL, TOD, VTB) dataset outside of SMP/E for entry into the configuration of the server IN00 under the DEFINE RULESET section. Do not edit the SMP/E copy for these rules. If a new maintenance affects these rules, see HOLD ACTION for next step about updating the rules created from running AVZGNMP1job.

4. 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.
5. 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.

6. 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.
7. 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 19](#)

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, AVZMDDLML, 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 AVZMDDLML rule, as follows:
 - a) Specify S next to AVZMDDLML to edit the rule.
 - b) Find the **vtb.optbd1cv** 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 <code>vtb.optbdlco = ':'</code> .
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 <code>vtb.optbdlch = '#'</code> .
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, <code>vtb.optbdlhp = '"NAME", "ADDRESS"'</code> . Note: If an <code>optbdlhp</code> value is defined, it supersedes any <code>optbdlhr</code> setting and the <code>optbdlhr</code> value is ignored.

- d) Save your changes and exit the editor.
4. Enable the rule by specifying E next to AVZMDDL and pressing Enter.
 5. Set the rule to Auto-enable by specifying A next to AVZMDDL and pressing Enter.
Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring alternate started task ID

About this task

You can configure the parameter ALTSTARTEDTASKID to build or update the metadata cache in the Data Virtualization Manager. When this ID is configured, the server always uses this ID during startup for any metadata cache refresh queries. Normal user queries, that use DRDA, use the primary user id and password.

Procedure

1. Add the following line to the INOO file.

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

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

2. Restart the server to make the changes effect.

Results

Verify the configuration of the parameter ALTSTARTEDTASKID in the server trace. To open the server trace, select option **B Server Trace** from the ISPF primary menu.

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.

Invoking the ISPF application using the command shell

You can invoke the Data Virtualization Manager server administrative menu by using the **ISPF Command Shell**.

Before you begin

- The Data Virtualization Manager server must be running.
- Your security administrator must give your TSO user ID READ, UPDATE, or both authorities. You need READ authority to view resource lists. You need UPDATE authority to modify server information.

Procedure

1. From the **ISPF Primary Option Menu**, enter the option **6** and press ENTER.
2. Enter the following command in the **ISPF Command Shell**:

- If you are only using the Data Virtualization Manager server:

```
'hlq.SAVZEXEC(AVZ)' 'SUB(AVZS)'
```

- If you are using the Instrumentation Server:

```
'hlq.SAVZEXEC(AVZ)' 'SUB(AVZS) TBSSID(AVZS)'
```

where *hlq* is your high-level qualifier, *SUB (AVZS)* specifies the subsystem name for the Data Virtualization Manager server, and *TBSSID(AVZS)* specifies the subsystem name for the Instrumentation Server.

Results

The **Primary Option Menu** panel is displayed. This panel specifies the ID of the subsystem to which you are connected and information about the product version. From this panel, you access all of the functionality in the product.

IBM Data Virtualization Manager for z/OS

```
Interface Facilities:
1 ACI          5 IDMS
2 Adabas      6 IMS
3 CICS        7 VSAM/Sequential
4 DB2         8 DSSPUFI
SSID      : AVZS
Version   : 01.01.00
Date     : 20/03/10
Time     : 08:12
```

```
Data Virtualization Server Administration:
A Remote User      - Manage Remote Users
B Server Trace    - Server Trace Facility - SIS SSID: AVZS
C AVZ 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
```

```
Option ===>
F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT   F11=RIGHT   F12=RETRIEVE
```

Invoking the ISPF application using the subsystem manager

You can use the subsystem manager to quickly navigate to the administration menu of Data Virtualization Manager server.

Procedure

To invoke the Data Virtualization Manager server using the subsystem manager:

1. Enter the command **dv** in the **ISPF Command Shell** to see the available subsystems.
2. Type **A** next to the desired subsystem to open the administrative menu of the subsystem.

In the following example, the administrative menu of the subsystem **AVZ4** is invoked.

```
----- DV Subsystems ----- Row 1 to 12 of 36
PLEX: RSPLEX01 LPAR: RS28      MODE: EDIT      CONFIRM: YES
LCs: A Admin  S Start  P Stop   J JCL   I IN00  O Output  D Details  F Files
DVSS LPAR Jobname  Status  ServCls
ADBJ RS28 ADBJ     UP      SDB_SCNM
ADBV RS28 ADBV     UP      SDB_SCHI
AVZB RS28 AVZB     UP      AVZ_SCNM
AVZC RS28 AVZC     UP      AVZ_SCHI
AVZS RS28 AVZS     UP      AVZ_SCHI
AVZY RS28 AVZY     UP      AVZ_SCHI
AVZ1 RS28 AVZ1     UP      AVZ_SCHI
AVZ3 RS28 AVZ3     UP      SDB_SCNM
A AVZ4 RS28 AVZ4     UP      AVZ_SCNM
AVZ5 RS28 AVZ5     UP      AVZ_SCNM
AVZ9 RS28 AVZ9     UP      AVZ_SCHI
AZKS RS28 AZKS     UP      SDB_SCNM
Command ===>                               Scroll ===> CSR
```

The administrative menu of the subsystem **AVZ4** is displayed.

```

. . . . . IBM Data Virtualization Manager for z/OS . . . . .

Interface Facilities:
1 ACI          5 IDMS          SSID   : AVZ4
2 Adabas      6 IMS           Version: 01.01.00
3 CICS       7 VSAM/Sequential Date   : 20/03/10
4 DB2        8 DSSPUFI      Time   : 08:12

Data Virtualization Server Administration:
A Remote User - Manage Remote Users
B Server Trace - Server Trace Facility - SIS SSID: AVZS
C AVZ 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

Option ==>

```

Support for the IBM i series as a DRDA or JDBC data source

Configure the IBM i series on an IBM AS/400 server as a DRDA or a JDBC data source.

DVM now supports the IBM i series on an IBM AS/400 server as a DRDA or a JDBC data source and configure the AVZSIN00 file to define a database on the i series.

Chapter 4. Installing 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

Note: If you use Python to connect with DVM and retrieve the data, ensure that you use pyodbc instead of JDBC. For a sample python programs, see [“Sample python programs”](#) on page 29.

Installing the ODBC driver

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

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. Go to IBM Fix Central page - <https://www.ibm.com/support/fixcentral/>.
2. Under **Find Product** section, in the **Product Selector** list, select IBM Data Virtualization Manager for z/OS, and then press **Enter**.
3. In the **Installed Version** list, select a required version number.
4. In the **Platform** list, select **All** and then click **Continue**.
The Identify fixes page is displayed.
5. Click **Browse for Fixes** option and then click **Continue**.
6. Select **ODBC** and then click `dvm-odbc-linux-windows-<version>.zip` file to download.
7. On your development workstation, create a directory.
8. Double-click the `dvm-odbc-linux-windows-<version>.zip` file, and then extract the contents to the 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
9. Run either the 32-bit or 64-bit version of the installer as appropriate for your target ODBC applications, and follow the installation wizard.
 10. 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*.

Installing ODBC docker file for zCX

The ODBC driver is built on Ubuntu 18.04 version and delivered as a docker file. You should have atleast Ubuntu 18.04 version or above to use this feature.

Before you begin

Install an Ubuntu docker image. The information provided in this topic are tested with the following configurations.

```
cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.4 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.4 LTS"
VERSION_ID="18.04"
```

```
Linux abcd 4.15.0-99-generic #100-Ubuntu SMP Wed Apr 22 20:31:47 UTC 2020 s390x s390x s390x
GNU/Linux
```

Procedure

1. Download and install the Docker desktop application from <https://www.docker.com/products/docker-desktop>.
2. Place the provided ODBC *Dockerfile* file and the ODBC driver file (*DVM_ODBC-3.x.x-zLinux-x64.tar.gz* file) in a folder and run the following docker build command to build the ODBC driver docker image:

```
docker build . -t odbcdriver
```

3. After the docker build command execution is completed, execute the following command to view the available docker images:

```
docker images
```

4. Execute the following docker run command to run the created docker image:

```
docker run -it odbcdriver bash
```

5. Edit the **.odbc.ini** file using the command **vi /root/.odbc.ini**.
6. Under the **ABCD_DEBUG** DSN section, update the HOST, PORT, UID and PWD information. If you are using the Db2 subsystem, add the relevant details for the HOST, PORT, UID, PWD and SUBSYS fields under the **ABCD** section. Here is an example configuration.

```
[ABCD_DEBUG]
```

```

Driver=Data Virtualization Manager 64-bit Debug Driver 3.2
HOST=
PORT=
LINK=TCPIP
DBTY=DVS
SUBSYS=NONE
UID=
PWD=
TRACEFLAG=DEBUG SQL DATA BUFFER NETWORK LOCK SYS
TRACEPATH=/root/errlog.txt

```

```

[ABCD]
Driver=Data Virtualization Manager 64-bit Driver 3.2
HOST=
PORT=
LINK=TCPIP
DBTY=DRDAorDB2
SUBSYS=DSN1
UID=
PWD=
TRACEFLAG=DEBUG SQL DATA BUFFER NETWORK LOCK SYS
TRACEPATH=/root/errlog.txt

```

The ODBC driver traces are stored in the **/root/errlog.txt** file.

7. Execute the `isql -v ABCD_DEBUG` command to connect to the server that you configured in the **.odbc.ini** file.
8. Run a query to confirm the connection.

Sample python programs

During ODBC docker image installation, Python versions 2.7.17 and 3.6.9 and pyodbc library are also installed. This section contains a sample program that you can use to connect to the DVM and execute a query.

Sample program for Python 2.7.17

```

import pyodbc

conn = pyodbc.connect('DSN=ABCD_DEBUG')
conn.setencoding(unicode,encoding='utf-8', ctype=pyodbc.SQL_CHAR)
conn.setdecoding(pyodbc.SQL_CHAR,encoding='utf-8', ctype=pyodbc.SQL_CHAR)

cursor = conn.cursor()
cursor.execute("select ZIP,CITY,STATE from us_zipcodes limit 10")

for row in cursor:
    print(row)

```

Note: Python 2.7.17 has issues handling in decimal values other than 0-9. For handling decimal values, use Python 3.6.9.

Sample program for Python 3.6.9

```

import pyodbc

conn = pyodbc.connect('DSN=ABCD_DEBUG')
conn.setencoding(encoding='utf-8', ctype=pyodbc.SQL_CHAR)
conn.setdecoding(pyodbc.SQL_CHAR,encoding='utf-8', ctype=pyodbc.SQL_CHAR)

cursor = conn.cursor()
cursor.execute("select * from us_zipcodes limit 10")

for row in cursor:
    print(row)

```

Executing a query using scodbcdm64 tool

Using the tool **scodbcdm64** provided in the ODBC package, you can execute a query.

Procedure

1. Provide the SQL query in the **input.txt** file.
2. Execute the query using the following command.

```
scodbcdm64 ABCD_DEBUG input.txt output.txt
```

Where **ABCD_DEBUG** is the DSN name, **input.txt** is the input file that contains the query, and the **output.txt** is the file where the output must be stored.

Example

For the following example, the query given in the input file is `select * from us_zipcodes limit 10`.

```
scodbcdm64 ABCD_DEBUG input.txt output.txt

vi output.txt
01001 AGAWAM           MA 15338      72.622739    42.070206
01002 CUSHMAN          MA 36963      72.515650    42.377017
01005 BARRE            MA 4546       72.108354    42.409698
01007 BELCHERTOWN     MA 10579      72.410953    42.275103
01008 BLANDFORD        MA 1240       72.936114    42.182949
01010 BRIMFIELD        MA 3706       72.188455    42.116543
01011 CHESTER          MA 1688       72.988761    42.279421
01012 CHESTERFIELD    MA 177        72.833309    42.381670
01013 CHICOPEE        MA 23396      72.607962    42.162046
01020 CHICOPEE        MA 31495      72.576142    42.176443
```

Installing the JDBC driver

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

About this task

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

Procedure

1. Go to IBM Fix Central page - <https://www.ibm.com/support/fixcentral/>.
2. Under **Find Product** section, in the **Product Selector** list, select IBM Data Virtualization Manager for z/OS, and then press **Enter**.
3. In the **Installed Version** list, select a required version number.
4. In the **Platform** list, select **All** and then click **Continue**.
The Identify fixes page is displayed.
5. Click **Browse for Fixes** option and then click **Continue**.
6. Select **JDBC** and then click `dvm-jdbc-<version>.zip` file to download.
7. On your development workstation, create a directory.
8. Double-click the `dvm-jdbc-<version>.zip` file, and then extract the contents to the directory.

What to do next

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

Chapter 5. Installing Data Virtualization Manager studio

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 IBM Data Virtualization Manager for z/OS installation with a newer version of the IBM Data Virtualization Manager for z/OS plug-in

Procedure

1. Go to IBM Fix Central page - <https://www.ibm.com/support/fixcentral/>.

2. Under **Find Product** section, in the **Product Selector** list, select IBM Data Virtualization Manager for z/OS, and then press **Enter**.
3. In the **Installed Version** list, select a required version number.
4. In the **Platform** list, select **All** and then click **Continue**.
The Identify fixes page is displayed.
5. Click **Browse for Fixes** option and then click **Continue**.
6. Select **Studio** and then click `dvm-studio-<version>.zip` file to download.
7. Create a new installation folder for the Data Virtualization Manager studio.
8. Double-click the `dvm-studio-<version>.zip` file, and then extract the contents to the installation folder.
9. 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. A Studio Setup dialog box appears, with the following prompt: **Would you like studio to prompt for workspace selection each time (Enter Y/N - Default is N)?**
Enter **Y**, to select a workspace each time you launch the studio.
Note: On entering **Y**, each time you launch the Data Virtualization Manager studio, a dialog box appears to select a workspace. If you enter **N** or do not enter anything in the Studio Setup dialog box, you can always change the workspace when required. See the topic [“Changing the workspace in studio”](#) on page 32.
 - c. 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.
10. To begin using Data Virtualization Manager studio, open the DV Data perspective using the menu option **Window > Open Perspective**.

Changing the workspace in studio

You can change the workspace in the Data Virtualization Manager studio from default workspace to any workspace. This topic outline the steps to change the workspace.

Procedure

On installing, if you enter **N** when prompted, a default workspace will appear, but you can change the workspace anytime. Perform the following steps to change the workspace.

1. In the **File** menu, select **Switch Workspace > Other**. The Select a Workspace as Directory dialog box appears.
2. In the **Workspace** list, select the workspace or Browse the required workspace.
3. Click **Launch**.

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 OEPORNUMBER . To locate this number, use SDSF on the mainframe to browse the server JOB output and search for OEPORNUMBER.
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.

Changing the language used in studio

You can change the language used in the user interface of the Data Virtualization Manager studio from English to Japanese. This topic covers the steps to change the language used in Data Virtualization Manager studio.

Procedure

By default, English language is used in Data Virtualization Manager studio. Perform the following steps to change the language to Japanese:

1. Edit the <studio_install_directory>\eclipse\eclipse.ini configuration file to add the following parameters:
 - Set the -Duser.language parameter as ja.
 - Set the -Duser.country parameter as JP.

Here's a sample eclipse.ini configuration file for your reference:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.0.v20130327-1440.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.200.v20140116-2212
-product
org.eclipse.epp.package.java.product
--launcher.defaultAction
openFile
--launcher.XXMaxPermSize
256M
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vmargs
-Dosgi.requiredJavaVersion=1.6
-Xms40m
-Xmx512m
```

```
-Duser.language=ja  
-Duser.country=JP
```

2. Save and close the `eclipse.ini` file.
3. Restart Data Virtualization Manager studio.

The Data Virtualization Manager studio user interface now uses the Japanese language.

If you wish to revert to English, comment the `-Duser.language` and `-Duser.country` entries in the `eclipse.ini` file using `#` and restart the Data Virtualization Manager studio.

Here's a sample of the commented entries in the `eclipse.ini` configuration file:

```
...  
...  
-Xms40m  
-Xmx512m  
#-Duser.language=ja  
#-Duser.country=JP
```

Chapter 6. Installing JDBC Gateway

The *JDBC Gateway* is a IBM Data Virtualization Manager for z/OS 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 following terminology is used in the installation procedure:
 - *JDBC Gateway server*. The server is the backend component that allows communication with the IBM Data Virtualization Manager for z/OS 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	<ul style="list-style-type: none">– 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:<ul style="list-style-type: none">- Google Chrome browser V50.0.2661.102 or later- 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: `ctag -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 `ctag` command and set `_BPXK_AUTOCVT=ON`.

Procedure

1. Go to IBM Fix Central page - <https://www.ibm.com/support/fixcentral/>.
2. Under **Find Product** section, in the **Product Selector** list, select IBM Data Virtualization Manager for z/OS, and then press **Enter**.
3. In the **Installed Version** list, select a required version number.
4. In the **Platform** list, select **All** and then click **Continue**.

The Identify fixes page is displayed.

5. Click **Browse for Fixes** option, enter the value for the selected option and then click **Continue**.
6. Select JDBC Gateway and then click `dvm-jdbc-gateway-<version>.zip` file to download.
7. 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.

8. At a command prompt in the directory, run the following command:

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

The installer launches.

9. 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 (<i>local directory</i> \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.
Set port for accepting DRDA connections (1527):	Enter the number of an available TCP/IP port for the application, or press Enter to use the default value 1527. This port number will be used to accept DRDA connections.
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

See [Using](#).

Chapter 7. Installing DVM interface for ACI

This section provides the installation requirements and installation steps for the DVM interface for ACI when used with Software AG's Natural via CICS and Batch.

Before you begin

Consider the following when installing the DVM Interface for ACI:

- If using the Natural language with DVM ACI support, the minimum recommended version of Natural is V 4.2.5.
- If you are using Natural security, you must create the DVM Natural Library before adding Natural source distribution. If you are not using Natural Security, the library is created automatically.
- For CICS, the DVM interface for CICS/TS must be installed.

About this task

Perform the following steps to install the DVM interface for ACI:

Procedure

1. Install the DVM server. See Chapter 3, “Installing and configuring the server,” on page 7.
2. Modify the DVM initialization EXEC.

Use the MODIFY PARM command to set following parameters located in the DVM initialization EXEC, xVZyIN00:

```
"MODIFY PARM NAME(ACIPREALLOC VALUE(100)"
"MODIFY PARM NAME(ACIDEFAULTCONNNAME) VALUE(EXCS)"
"MODIFY PARM NAME(ACIPERSISTTIMEOUT) VALUE(CLIENT)"
"MODIFY PARM NAME(ACICONNPW) VALUE(NO)"
"MODIFY PARM NAME(ACISRVCONNINT) VALUE(25)"
"MODIFY PARM NAME(ACISETSRVPRINTTIME) VALUE(YES)"
"MODIFY PARM NAME(ACICICSGENERIC) VALUE(ACICSRV)"
"MODIFY PARM NAME(ACIBATCHGENERIC) VALUE(ACIBSRV)"
"MODIFY PARM NAME(ACITRACE) VALUE(YES)"
```

3. Assemble the ACI translation/migration table.

You must assign a default DV Server subsystem to route all service application ACI requests. You can do this with the ACI translation/migration table (AVZRTX table) provided with the DVM interface for ACI. The AVZRTX table is initially loaded by AVZCITRUI (the ACI Task Related User Exit Initialization program), and is viewed by AVZCITRU (the ACI Task Related User Exit).

The minimal assembled module must contain the TYPE=INITIAL card with a default defined DVM subsystem and a TYPE=FINAL card to set the default subsystem.

Note: The default DVM server subsystem can be overridden by using the "#ETBCB-ACI-ID" field in the REGISTER function to specify a DVM server subsystem.

If you are using Natural via CICS, the AVZRTX table can also be used by AVZCINAT, the Natural “front end” program, to obtain information to start the Natural session. AVZCIFEN is a CICS module that allows the passing of the COMMAREA to the invoked ACI server, if the CICS transaction is associated with the AVZCIFEN program. If you associate the transaction name with the program AVZCIFEN when defining the transaction to CICS, you can pass Natural startup parameters to the transaction using the COMMAREA. In turn, AVZCIFEN transfers control (EXEC CICS XCTL) to a program of your choice. If AVZCINAT is used as the program to which AVZCIFEN transfers control, the COMMAREA must be formatted with the Natural nucleus program name (eight bytes, space padded if necessary), followed by a comma, followed by the CICS transaction ID for Natural (four bytes, space padded if necessary), followed by a comma, followed by the Natural startup parameters.

Note: If the SDCIFEN program is used to specify the Natural startup parameters, as a minimum, the AVZRTX table must be assembled with the TYPE=INITIAL card with a default DVM subsystem defined and a TYPE=FINAL card to set the default subsystem.

To assemble the AVZRTX table:

- a) Use data set member AVZACJBH, located in HLQ.SAVZSAMP. This member contains the source that needs to be assembled for the ACI translation/migration table.
- b) Install the ACI translation/migration table into a load library that is concatenated to the DFHRPL DD statement for CICS which may include the SAVZCLOD dataset or for batch include the AVZRTX table within the STEPLIB of the batch application.

The following example shows an example AVZRTX entry with the required minimum:

```
ACI TYPE=INITIAL,SUBS=xVZy DEFINITION START / DEFAULT
ACI TYPE=FINAL DEFINITION END
```

4. Link the stubs:

- a) Link the AVZCICIS stub into your Natural CICS nucleus, using the following example:

```
//AVZLIB DD DSN=HLQ.SAVZCLOD,DISP=SHR
.
.
CHANGE BROKER(DVSACI) <--CHANGING THE DEFAULT NAME
INCLUDE AVZLIB(SDCICIS) FOR AVZ ACI STUB
INCLUDE NATLIB(NATLAST) FINAL INCLUDE
```

You must add DVSACI as a CSTATIC parameter, as shown in the following example:

```
CSTATIC=(CMMSG,DVSACI)
```

- b) Link the AVZRBIS stub into your Natural Batch nucleus, using the following example:

```
//AVZLIB DD DSN=HLQ.SAVZLOAD,DISP=SHR
.
.
CHANGE BROKER(DVSACI) <--CHANGING THE DEFAULT NAME
INCLUDE AVZLIB(AVZRBIS) FOR AVZ ACI STUB
INCLUDE NATLIB(NATLAST) FINAL INCLUDE
```

You must add DVSACI as a CSTATIC parameter, as shown in the following example:

```
CSTATIC=(CMMSG,DVSACI)
```

5. Create CICS resource definitions.

The DVM interface for ACI requires several resources that include programs, a transaction, and an intrapartition transient data queue (TDQ) to be defined in all CICS regions that will be using this component.

Note: These resources should have been defined during the installation of the DVM interface for CICS/TS, a prerequisite for this component. AVZCICSD (in HLQ.SAVZCNTL library) is the name of the member used to define these resources. This member defines SDCITRUI and SDCITRU to CICS for the loading and viewing of the AVZRTX table (see [Step 3](#)).

6. Assemble CICS PLTPI for ACI task related user exit.

- a) Add the following entry to the program list table/program initialized (PLTPI) table.

```
DFHPLT TYPE=ENTRY, PROGRAM=SDCITRUI
```

This entry should follow the first of the following statements to ensure that the SDCITRUI will be executed in phase 2 of the CICS PLTI process:

```
DFHPLT TYPE=ENTRY, PROGRAM=DFHDELIM
```

- b) Code the appropriate parameter in CICS startup.

```
PLTPI=xx
```

where xx is the suffix value given to the DFHPLTI table.

7. In order to associate the data set modules with CICS, add *HLQ.SAVZCLOD* data set to the CICS DFHRPL data set concatenation.
8. Restart the CICS region.

Because the CICS installation for the DVM interface for ACI requires use of a CICS task related user exit, you should restart your CICS region. The following message appears on the console if the CICS Task Related User Exit correctly installs and is enabled:

```
SDB0364I SDCITRU is started and enabled.
```

9. Add Natural source distribution.

Customize and submit *AVZACNLD*, located in *HLQ.SAVZCNTL*. This is a Natural Systems utility that will load the Natural environment objects into the AVZ Natural library. These objects include Natural samples and ACI Server programs.

Note: If you are using Natural security, you must create the AVZ Natural Library before performing this step. If you are not using Natural Security, the library is created automatically.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in this product enable users to perform the following activities:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS® interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*
 - *z/OS TSO/E Primer*
 - *z/OS TSO/E User's Guide*

These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Product legal notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive

Armonk, NY 10504-1785
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.html>.

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java[™] and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux[®] is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

A

- accessibility
 - overview [43](#)
- ACF2 [4](#)
- APF (Authorized Program Facility)
 - LOAD library [5](#)
- AVZDFDIV member [3](#)
- AVZSIN00 member [18](#)
- AVZSMUTC member [17](#)

C

- checklist, customization [3](#)
- code page [17](#)
- configuring
 - APF-authorizing [5](#)
 - Data Virtualization Manager server file [18](#)
 - ISPF client [20](#)
 - security authorizations [4](#)
 - started task JCL [19](#)
 - Workload Manager (WLM) [5](#)
- cookie policy [45](#), [47](#)
- copying
 - target libraries [6](#)
- creating
 - SWIOBJ data set [3](#)
 - system data sets [3](#)
- customization checklist [3](#)

D

- Data Studio
 - verifying installation [33](#)
- Data Virtualization Manager server
 - installing [7](#)
 - naming conventions [3](#)
 - starting [19](#)
 - stopping [19](#)
- Data Virtualization Manager server file [18](#)
- Data Virtualization Manager studio
 - download [31](#)
 - installing [31](#)
- DBCS [17](#)
- default subsystem name [18](#)
- delimited data, configuring [22](#)
- double-byte character set [17](#)
- downloading
 - components [7](#)
- Downloading
 - studio component [31](#)
- DVM interface for ACI [39](#)

G

- generation data set, configuring access [21](#)

I

- installation
 - DVM interface for ACI [39](#)
 - system requirements [1](#)
- installing
 - JDBC Gateway [35](#)
 - server component [7](#)
 - studio component [31](#)
- ISPF application [24](#)
- ISPF client [20](#)

J

- JDBC Gateway
 - installing [35](#)

L

- legal notices
 - cookie policy [45](#), [47](#)
 - notices [45](#)
 - programming interface information [45](#)
 - trademarks [45](#), [46](#)
- LOAD library [5](#)

M

- maintaining
 - Data Virtualization Manager server [17](#)

N

- naming conventions
 - Data Virtualization Manager server [3](#)
- notices [45](#)

P

- programming interface information [45](#)

R

- RACF [4](#)

S

- screen readers and magnifiers [43](#)
- security authorizations
 - ACF2 [4](#)
 - RACF [4](#)
 - Top Secret Security [4](#)
- sending comments to IBM [ix](#)
- server event facility (SEF)
 - configure delimited data [22](#)

- server event facility (SEF) *(continued)*
 - configure GDG access [21](#)
- SMP/E
 - maintaining Data Virtualization Manager server [17](#)
- started task JCL [19](#)
- Starting
 - Instrumentation Server [24](#)
 - server [24](#)
- system data sets
 - creating [3](#)
- system requirements [1](#)

T

- target libraries [6](#)
- Top Secret Security [4](#)
- trademarks [45](#), [46](#)

V

- verifying [24](#)
- verifying installation
 - Data Studio [33](#)

W

- Workload Manager (WLM)
 - configuring [5](#)



GC27-8874-02

