DB2 QMF Data Service
Version 12 Release 1

Studio User's Guide

SC27-8886-00

DB2 QMF Data Service
Version 12 Release 1

Studio User's Guide

SC27-8886-00

Note

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

This edition applies to Version 12 Release 1 of IBM DB2 Query Management Facility (QMF) Enterprise Edition, which is a feature of IBM DB2 12 for z/OS (5650-DB2) and IBM DB2 11 for z/OS (5615-DB2). It also applies to Version 12 Release 1 of IBM DB2 QMF for z/OS (5697-QM2), which is a stand-alone IBM DB2 for z/OS tool. This information applies to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. About IBM® DB2 QMF® Data Service

IBM® DB2 QMF® Data Service provides mainframe-resident data virtualization for real-time, universal access to data, regardless of location or interface.

Use the Data Service Studio (DS Studio) user interface to create the virtual libraries and virtual tables that maps to and represents the data that you want to access.

Chapter 2. Overview

The Data Service Studio (DS Studio) is built on Eclipse technology and is available with IBM® DB2 QMF® Data Service.

Use the DS Studio to get ANSI SQL access to your data without having to first extract and write data to a separate file, and then transform that data. IBM® DB2 QMF® Data Service gives you real-time access to your data because the data is read directly from the mapped data source, and then transformed during the load process. Depending on your business needs, you can choose the parallel-data loading feature, MapReduce, to further optimize load performance and gain even faster access to your most critical data.

IBM® DB2 QMF® Data Service supported data sources include DB2, IMS DB, VSAM data sets, physical sequential data sets, Oracle and DB2 LUW. This includes data joins between the supported data sources.

The following illustrates the basic IBM® DB2 QMF® Data Service architecture:

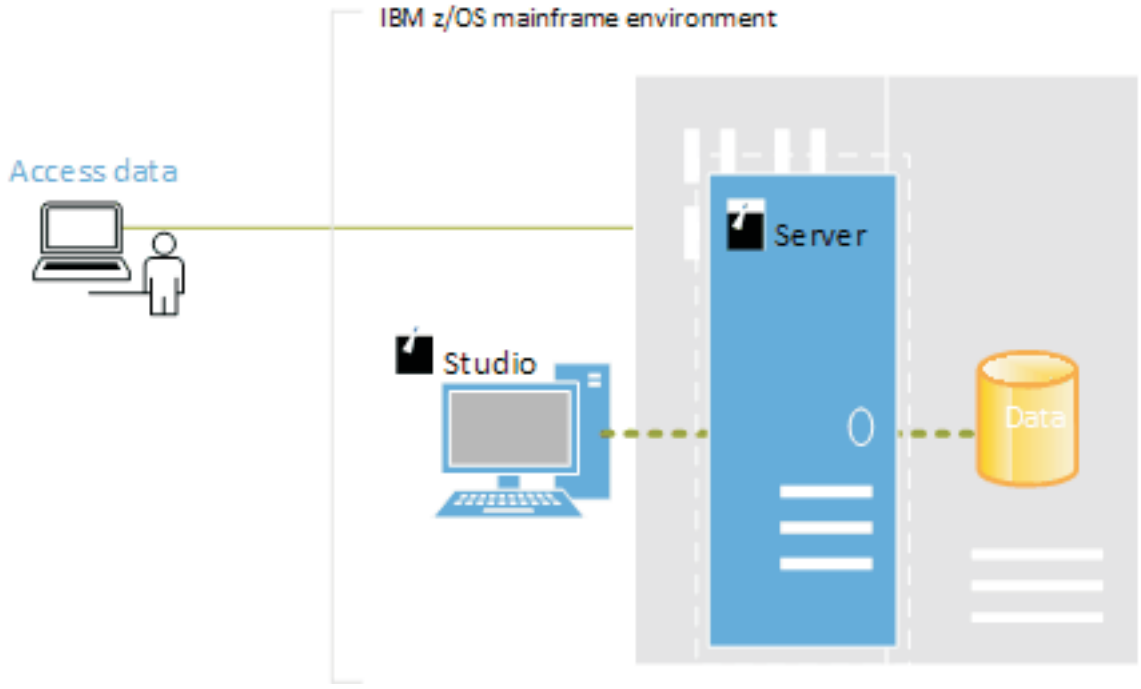

Figure 1. Architecture

To get access to data using the Studio, you need to complete the following high-level steps:

1. From the Studio, open the Data Service perspective.
2. Connect to the Data Service server. To access other DB2 subsystems, you need to create the necessary Bind packages and grant the appropriate privileges.
3. Create a virtual source library that references existing libraries on the mainframe.

4. For SQL access to data, create virtual tables to represent the data you want to access.

Chapter 3. Data Service perspective

The Data Service perspective provides the default views and editors that you use to perform tasks that are associated with managing and loading data.

Views

The following views are available with this perspective:

Views	Description
Active Connections	Lists the open JDBC connections between the Studio and one or more servers. The current Active Connections is used by the SQL to issue SQL queries over the JDBC connection.
Explorer	Lists data resources, stored procedures, and meta data. You can perform tasks on selected objects in the tree. Explorer views include the following tabs: • Client: lists information that is related to data sources and application development on your local machine. • Server: lists the Data Service server to which you want to connect, view resources, or perform tasks. • Network: lists host and server connections within your network. You can choose to view or modify existing host and server settings. • Favorites: lists shortcuts to the mainframe resources that you frequently access.
Labels	Applies labels to Server Trace messages for use when searching within the **Server Trace** view.
Lists	Use to display details for each tree node or object that is selected in an Explorer view.
Search	Use to search for a text string within Server Trace results.
Server Trace	Use to set and gather server diagnostic information for support purposes.
Server Trace Import	Use to import Server Trace (**.isx**) files.
SQL Results	Use to display the results returned from a SQL query in the **SQL Results** tab, and resulting system messages in the **SQL Messages** tab.
Studio Navigator	Use to list shortcuts to key task views and editors for this plug-in.
Properties	Use to display the properties of a selected object on the Server, Network, or Client navigation tabs.

Views	Description
Virtualization Facility	Displays virtual table mapping details.

Editors

The following text editors are available with this perspective:

Editors	Description
Data Source	Use to edit connection definitions which are used to open active connections.
SQL	Use to compose SQL statements and to invoke queries against the server.
Virtualization Facility	Use to edit meta data settings related to virtual tables and virtual views.

Wizards

This perspective includes wizards that guide you through tasks, such as:

- Setting the server connection.
- Creating virtual source libraries.
- Creating virtual tables for SQL access to data.
- Generating the SQL class.

Chapter 4. Connecting to the Data Service server

To access data on the mainframe, connect Data Service Studio to a Data Service server that is running on an z/OS mainframe instance.

Connecting to the Data Service server

Connect to a Data Service server on the mainframe by using the Data Service Studio.

Before you begin

Before connecting to the Data Service server, verify that the following prerequisite is met:

- When starting the Data Service Studio, right-click Data Service Studio, and select to **Run as administrator**.

Procedure

1. From the **Studio** menu, select **Window** > **Open Perspective** > **Data Service**.
2. On the Server tab, click **Set Server**.
3. In the Set Current Server dialog box, complete the following fields:

Option	Description
Host	Enter the TCP/IP host name or IP address of the mainframe system on which Data Service server is deployed.
Port	Enter the port number that the Data Service server is using. The default is 1200.
Userid	Enter the Mainframe user ID.
User Password	Enter the password associated with the mainframe user ID.

4. Click **OK**.

Configuring access to DB2 subsystems

To allow the Data Service server to access DB2 subsystem resources, you must execute bind packages and grant user authority.

Before you begin

Before you begin this task, you must know the host name and the port number of the Data Service server, and your log on credentials; which must have the authority to process bind and grant operations on the DB2 subsystem.

About this task

Perform this task once for each DB2 subsystem that you want to access.

Procedure

1. From Studio, click **Window** > **Open Perspective** > **Data Service**.
2. On the **Server** tab, click **Set Server**.
3. On the Set Current Server dialog box, complete the following fields:

Option	Description
Host	Enter the TCP/IP host name or IP address of the mainframe system.
Port	Enter the port number that is used to communicate with the Data Service server. The default is 1200.
Userid	Enter the mainframe user ID.
User Password	Enter the password for the mainframe user ID.

4. Click **OK**.
5. On the **Server** tab, expand **SQL** > **Data** > **Other Subsystems**.
6. Right-click the subsystem and select **BIND/GRANT Packages**.
7. Complete the following fields:

Option	Description
Package Prefix	Enter the two character prefix to assign to the package. The package prefix must match the prefix that is defined on the mainframe server. If you change the default prefix (DV), you must also change it in the .SCQDEXEC(CQDSIN00) file, where `server` is the customized server name.
Number of Cursors	Enter the number of cursors to use to process results. The default is 200.
Collection	Enter the value to use to bind packages. The default is NULLID. This value is normally determined by the DB2 Administrator.
Table Qualifier	Enter the value to use to qualify unqualified SQL. This value is normally determined by the DB2 Administrator.
Owner UserId	Enter the user ID of the package owner. This value is normally determined by the DB2 Administrator.
Grant to	Set only when granting authority for the target DB2 server. The default is `PUBLIC`.

8. Select one or more of the following package processing options, and then click **Execute**:

Option	Description
Bind Package	Binds the product packages.
Grant Execute	Grants access permission to the user ID specified, if any, in the **Grant to** field.

Option	Description
Replace Packages	Replaces an existing package, if one exists, for the specified subsystem. Select this option only if the package already exists. Some subsystems require the replacement of existing packages.

9. Depending on the options that you select, additional dialog boxes and messages might be displayed.
10. Review results in the **Results** text box, and click **BIND/GRANT**.

Locale considerations

The default Studio uses code page US/English IBM 1047 to perform character data translations between the native Java character encoding (UTF-8) and the mainframe EBCDIC. You can modify data source connection definitions to use different local code pages.

Procedure

To configure the data source connection definition:

1. In the **Active Connections** view, close all open connections.
2. On the **Client** tab, expand **Data Virtualization** > **Data Sources** > **JDBC**, and then browse for the data source that you want to modify.
3. Right-click the data source that you want to modify, and then click **Edit**.
4. In the **Data Source Editor**, click the **Connection String** tab.
5. Add or modify the `Charset` setting to use the appropriate EBCDIC based code page. For example, IBM037.
6. If `LGID=ENC` exists in the connection string, delete it to avoid potential interference with the `Charset` setting.
7. Save the data source definition.

To change the default `Charset` that the studio uses when creating connection definitions:

8. From the **Window** menu, select **Preferences**, expand Data Service, and then expand **Driver**.
9. Add the new `Charset` setting in **Connection Overrides**, and then click **OK**.
10. Create a new connection definition (DSN), and then confirm that the new setting displays in the connection string. When running queries using the new data source definition, the correct character data (including language specific glyphs) should now display in the SQL Results view.

Chapter 5. Creating virtual source libraries

A *virtual source library* is a reference to a library that already exists on the mainframe, which contains the information that is required to virtualize the source data.

About this task

For example, for a VSAM file, the library must contain the copybook that describes the structure of the records in the VSAM file. For an IMS database, you need to provide the Database Definition (DBD) and Program Specification Block (PSB) files, and a copybook structure for each segment of the IMS database that you want to virtualize. You might need to create multiple virtual source libraries, depending on how the files or various types are organized in the z/OS® file system.

The following source library options are available:

Data set: A PDS or PDSE on z/OS. Members in the data set contain the structure definitions of the source files you want to virtualize in a 80 byte record format. Data set is the default selection.

- COBOL copybooks for IMS/DB segments, VSAM, and sequential files.
- IMS PSB and IMS DBD definitions for IMS databases.
- DDM Views (Data Definition Module) for Adabas. Run your Natural job to create a DDM View listing, and then use the output to create a data set member in the source library.

Natural: An ACI service that provides real-time access to the Natural source library (FUSER file) for DDM Views is required. This option is valid if you are virtualizing data for use with Adabas, and have configured an ACI service. If you cannot create or choose a source library and accept the 2 byte Adabas column names, you can choose from the following options:

Note: This method may result in naming collisions. For example, if the column name is also a reserved word such as, 'AS', the column name must be changed before you can successfully query the referenced column.

- Directly access the DDM View of the Adabas source definitions.
- Run the Natural job to create a DDM View listing and place the output in your Data set Source Library (also supported if ACI is not configured).

Procedure

1. On the **Server** tab, under **Admin**, expand **Source Libraries**.
2. Right-click **Create Virtual Source Library**, and select **Create Virtual Source Library**.
3. From the list of wizards, select the type of data to reference, **Data Set** or **Natural**, and then click **Next**.
4. For the **Data Set** wizard, complete the following fields to identify the PDS on the mainframe where source members exist:

Field	Action
Name	Enter a name for the source library.

Field	Action
Description	Enter an optional description for the source library.
Library Name	Enter the name of the mainframe data set that the source library references.

5. If you select the Natural source library wizard, enter the information that follows, and then click **Test Natural Library** to test your new source library.

Field	Action
Name	Enter a name for the source library.
Description	Enter an optional description for the source library.
Natural Library	Enter the name of the Natural library.
DBID	Enter the ID of the database where the Natural source exists.
File Number	Enter the Adabas file number of the Natural source library (FUSER file). Typically, this number is set to 9.
Service Type	Select the ACI service to use to browse the library for source members and run requests: • **CICS** • **Batch**
From	Enter the name of the starting object to return. This information is optional and used to limit the list of objects returned.
To	Enter the name of the ending object to return. This information is optional and used to limit the list of objects returned.
DDM	If the natural library contains only dictionaries, select to enable the DDM to disable **Object types** options.
Object types	If DDM is not enabled, select from the following object types to return: • **Parameter Data Area** • **Global Data Area** • **Map** • **Program** • **Text** • **Copycode** • **Local Data Area** • **Subprogram** • **Subroutine**

6. Click **Finish**.

Results

The new source library is displayed in the **Source Libraries** folder.

Chapter 6. Creating virtual tables

Use virtual table wizards to create virtual tables from which you can generate the SQL used to read and extract the data from a mainframe.

The virtual table wizard that you select is determined by the data type from which you want SQL access.

Creating virtual tables for Adabas data

Create a virtual table to get SQL access to your Adabas data.

Before you begin

Have the following information available: Adabas DB ID and password, the file number to use, and the subsystem name.

Procedure

1. On the **Server** tab, verify that you are connected to the correct server.
2. Expand **Admin** > **Source Libraries** and then verify that the required source libraries exist.
3. Expand **SQL** > **Data**, and then expand the server from which you want to create the virtual table.
4. Right-click **Virtual Tables**, and then select **Create Virtual Table**.
5. Under **Wizards**, select the wizard to use, and then click **Next**.
6. Complete the following fields, and then click **Next**:

Option	Description
Name	Enter a unique name. The name can contain a maximum of 30 characters. Uppercase alphanumeric characters are allowed as well as numbers 0-9. The underscore (_) character is allowed; however, the initial character in the name must be an alphanumeric character.
Target	Select the target data set to store the metadata (for example: hlq.USER.MAP). Data sets are defined in the server configuration file.
Description	Enter an optional description.
Arrays Handling	Enable one of the following array management options: • **Flatten arrays into a single fixed table at runtime**: This relates to multiple occurring (MU) fields and periodic (PE) groups. • **Return arrays into separate tables at runtime**: This relates to multiple occurring (MU) fields and periodic (PE) groups. A subtable is generated for each array. Subtables only support SQL read access.

7. Complete the Adabas table parameters fields, then click **Next**:

Option	Description
DB ID	Enter the Adabas database ID.
File Number	Enter the number of the file to use.
Adabas Password	If the file is password-protected, enter the password. This password is stored in the virtual table so that future queries use the same password to access the data.
Database	Enter the name of the Adabas database.
SubSystem	Enter the name of the Adabas subsystem.
Max MU Count	Enter the maximum number of times to repeat the MU field. The default is 10.
Max PE Count	Enter the maximum number of times to repeat the PE field. The default is 10.
Create Count Field	Enable if you want to index every MU or PE field so that the index (count) field created precedes the repeating field. This index field tells the caller how many repeating fields are being used.
Secure	Enable if you want to choose the Adabas file ID number to be used for file name security.
DE Search only	Enable if you want the utility to generate control definitions that allow the client to only use WHERE columns that are Adabas descriptors (such as superde, subde, and hyperde).
Search by PE index	Enable to allow the client to target rows that match a particular occurrence of the PE field when searching rows using the WHERE clause. If this parameter is not specified, all rows where any occurrence of that PE field match the value specified will be targeted.
Unpacked to Packed	Informs the extract to convert all unpacked format fields to packed format.
Binary to Integer	Informs the extract to convert all 2-byte and 4-byte binary fields to short integer and integer formats, respectively.
Advanced	To divide the data into logical partitions and process the partitions in parallel, click **Advanced**. Enter a **Thread Count** value for **MapReduce (Server Parrallelism Settings)**. The number of zIIP processors is checked at runtime, and one thread is used for each processor discovered. The value you specify overrides the default value (2) and the discovered value.

8. Optional: If you have a Natural Data Definition Module (DDM) listing of the file, you can complete the following to get additional metadata information:

Option	Description
Available Source Libraries	From the list of **Available Source Libraries**, select the source library that contains the data structure definition that you want to use when virtualizing data.
Source Library Members	Select the names of each source library member that represents the data structure that you want to include. The green arrow next to a DDM indicates that it is a suggested member, not that it is selected.

9. Complete the following data layout fields, and then click **Next**:

Option	Description
Source	Expand the source file to verify that it correctly displays the source (member).
Start Field	Accept the default root start field, or expand the file and select a different start field.
End Field	Accept the default root end field, or expand the file and select a different end field. By default, **End Field** is disabled.

10. Click **Finish**.

What to do next

You can use virtual tables to generate SQL queries.

Creating virtual tables for RDBMS data

Create a virtual table for SQL access to data from an existing RDBMS data source. RDBMS data sources include DB2 LUW (Linux, UNIX, and Windows) and Oracle.

Before you begin

You need to have the following information available: The RDBMS subsystem name and optionally, the plan name.

Procedure

1. On the **Server** tab, verify that you are connected to the correct server.
2. Expand **SQL** > **Data**, and then expand the server from which you want to create the virtual table.
3. Right-click **Virtual Tables**, and then select **Create Virtual Table**.
4. Under **Wizards**, select the wizard to use, and then click **Next**.
5. Complete the following fields, and then click **Next**:

Option	Description
Name	Enter a unique name. The name can contain a maximum of 30 characters. Uppercase alphanumeric characters are allowed as well as numbers 0-9. The underscore (_) character is allowed; however, the initial character in the name must be an alphanumeric character.

Option	Description
Target	Select the target data set to store the metadata (for example: hlq.USER.MAP). Data sets are defined in the server configuration file.
Description (optional)	Enter an optional description.

6. Complete the following data collection fields, and then click **Next**:

Option	Description
Table Browser	Browse the DBMS source subsystem, and then choose the table or the view that has the data that you want to access.
Table Columns	Displays the columns in the selected table. Select the columns to include.
Virtual Target System	Accept the previously selected target system, select a different target system, or click **Create Target System**. If you choose to create a new target system, on the **New Virtual Target System** dialog, enter the new target system name, the connection to use, and optionally the DBMS default plan name to use.
Advanced	To divide the data into logical partitions and process the partitions in parallel, click **Advanced**. Enter a **Thread Count** value for **MapReduce (Server Parrallelism Settings)**. The number of zIIP processors is checked at runtime, and one thread is used for each processor discovered. The value you specify overrides the default value (2) and the discovered value.

7. Click **Finish**.

What to do next

You can choose to generate a SQL query from the virtual table.

Creating virtual tables for IMS data

Create virtual tables for SQL access to IMS data.

Before you begin

The PSB, DBD, and the copybooks for each segment must exist in the source library.

Procedure

1. On the **Server** tab, verify that you are connected to the correct server.
2. Expand **Admin** > **Source Libraries** and then verify that the required source libraries exist.
3. Expand **SQL** > **Data**, and then expand the server from which you want to create the virtual table.

4. Right-click **Virtual Tables**, and then select **Create Virtual Table**.
5. Under **Wizards**, select the wizard to use, and then click **Next**.
6. Complete the following data layout fields in the order that follows, and then click **Next**:

Option	Description
DBD	To download new members from DBD files in your source libraries, click **Extract DBD**, or select a previously downloaded DBD definitions from the **DBD** drop-down.
PSB	To download new members from PSB files in your source libraries, click **Extract PSB**, or select a previously downloaded PSB definitions from the **PSB**.
Create Virtual Table	After selecting a DBD and PSB, you can create a virtual table on a per IMS segment basis by selecting **Create Virtual Table**, and then completing the wizard for each segment.

7. Complete the following data set fields, and then click **Next**:

Option	Description
Target	Select the target data set to store the metadata (for example: hlq.USER.MAP). Data sets are defined in the server configuration file.
Description	Enter an optional description.

8. Complete the following source library fields, and then click **Next**:

Option	Description
Available Source Libraries	From the list of **Available Source Libraries**, select the source library that contains the data structure definition that you want to use when virtualizing data.
Source Library Members	Select PDS members that represent the data structures to include, and then click **Download** to copy the members from the mainframe to your desktop.
Download Source Files	Select previously downloaded members to include.

9. Complete the following data layout fields, and then click **Next**:

Option	Description
Source	Expand the source file to verify that it correctly displays the source (member).
Start Field	Accept the default root start field, or expand the file and select a different start field.
End Field	Accept the default root end field, or expand the file and select a different end field. By default, **End Field** is disabled.

10. Select the **DBD Name**, **Segment Name**, and optionally the **PSB Name** to use. You can optionally add new names.
11. Click **Finish**.

What to do next

You can use virtual tables to generate SQL queries.

Creating virtual tables for sequential data

Create virtual tables for SQL access to sequential data.

Before you begin

Before creating the virtual table, add the copybook to the source library and have the data set name available.

Procedure

1. On the **Server** tab, verify that you are connected to the correct server.
2. Expand **Admin** > **Source Libraries** and then verify that the required source libraries exist.
3. Expand **SQL** > **Data**, and then expand the server from which you want to create the virtual table.
4. Right-click **Virtual Tables**, and then select **Create Virtual Table**.
5. Under **Wizards**, select the wizard to use, and then click **Next**.
6. Complete the following fields, and then click **Next**:

Option	Description
Name	Enter a unique name. The name can contain a maximum of 30 characters. Uppercase alphanumeric characters are allowed as well as numbers 0-9. The underscore (_) character is allowed; however, the initial character in the name must be an alphanumeric character.
Target	Select the target data set to store the metadata (for example: hlq.USER.MAP). Data sets are defined in the server configuration file.
Description	Enter an optional description.
Convert VAR* fields to True VAR* fields	To convert existing **VAR*** fields to **True VAR*** fields, enable **Convert VAR* fields to True VAR* fields**.

Option	Description
Arrays Handling	Enable one of the following array management options: • **Flatten arrays into a single fixed table at runtime**: This supports both **OCCURS** and **OCCURS DEPENDING ON** statements. • **Return arrays into separate tables at runtime**: This supports both **OCCURS** and **OCCURS DEPENDING ON** statements. A subtable is generated for each array. Subtables only support SQL read access. • **Flatten arrays now**: If you select this option, you cannot change array-handling after you save the virtual table.

7. Complete the following source library fields, and then click **Next**:

Option	Description
Available Source Libraries	Select the source library that contains the data structure to use.
Source Library Members	Select PDS members that represent the data structures to include, and then click **Download** to copy the members from the mainframe to your desktop.
Download Source Files	Select previously downloaded members to include.

8. Complete the following data layout fields, and then click **Next**:

Option	Description
Source	Expand the source file to verify that it correctly displays the source (member).
Start Field	Accept the default root start field, or expand the file and select a different start field.
End Field	Accept the default root end field, or expand the file and select a different end field. By default, **End Field** is disabled.

9. Optional: Accept the default table redefines or expand **Redefine** to modify your selection, and click **Next**.
10. Complete the following data source fields, and then click **Next**:

Option	Description
Data set Name	Enter a name for the data set. To use a PDS member as the data source, specify the partitioned data set name. Otherwise, you may specify a sequential data set or a Generation Data Groups (GDG) data set using the GDG syntax such as hlq.DATA.SEQ(-1). Click **Validate** to verify that the data set name exists on the host.

Option	Description
Member	If applicable, enter the PDS member name to use. Click **Validate** to verify that the member name exists on the host. The DSN is not validated.
Post-Read Exit Name	To manipulate the data after reading it from the source file, enter the name of the post-read exit to use. This is the custom exit routine that is installed on the Server and that is used to perform additional processing after a record is read from the data source.
Advanced	To divide the data into logical partitions and process the partitions in parallel, click **Advanced**. Enter a **Thread Count** value for **MapReduce (Server Parrallelism Settings)**. The number of zIIP processors is checked at runtime, and one thread is used for each processor discovered. The value you specify overrides the default value (2) and the discovered value. You can also choose to disable MapReduce.

11. Click **Finish**.

What to do next

You can use virtual tables to generate SQL queries.

Creating virtual tables for VSAM and IAM data

Create virtual tables for SQL access to VSAM data, and IAM files.

Before you begin

Have the following information available: VSAM cluster name (*sourcelibrary.copybook.filename*).

Procedure

1. On the **Server** tab, verify that you are connected to the correct server.
2. Expand **Admin** > **Source Libraries** and then verify that the required source libraries exist.
3. Expand **SQL** > **Data**, and then expand the server from which you want to create the virtual table.
4. Right-click **Virtual Tables**, and then select **Create Virtual Table**.
5. Under **Wizards**, select the wizard to use, and then click **Next**.
6. Complete the following fields, and then click **Next**:

Option	Description
Name	Enter a unique name. The name can contain a maximum of 30 characters. Uppercase alphanumeric characters are allowed as well as numbers 0-9. The underscore (_) character is allowed; however, the initial character in the name must be an alphanumeric character.
Target	Select the target data set to store the metadata (for example: hlq.USER.MAP). Data sets are defined in the server configuration file.
Description	Enter an optional description.
Convert VAR* fields to True VAR* fields	To convert existing **VAR*** fields to **True VAR*** fields, enable **Convert VAR* fields to True VAR* fields**.
Arrays Handling	Enable one of the following array management options: • **Flatten arrays into a single fixed table at runtime**: This supports both **OCCURS** and **OCCURS DEPENDING ON** statements. • **Return arrays into separate tables at runtime**: This supports both **OCCURS** and **OCCURS DEPENDING ON** statements. A subtable is generated for each array. Subtables only support SQL read access. • **Flatten arrays now**: If you select this option, you cannot change array-handling after you save the virtual table.

7. Complete the following source library fields, and then click **Next**:

Option	Description
Available Source Libraries	From the list of **Available Source Libraries**, select the source library that contains the data structure definition that you want to use when virtualizing data.
Source Library Members	Select PDS members that represent the data structures to include, and then click **Download** to copy the members from the mainframe to your desktop.
Download Source Files	Select previously downloaded members to include.

8. Complete the following data layout fields, and then click **Next**:

Option	Description
Source	Expand the source file to verify that it correctly displays the source (member).
Start Field	Accept the default root start field, or expand the file and select a different start field.

Option	Description
End Field	Accept the default root end field, or expand the file and select a different end field. By default, **End Field** is disabled.

9. Optional: Accept the default table redefines or expand **Redefine** to modify your selection, and click **Next**.
10. Complete the following fields, and then click **Next**:

Option	Description
Cluster Name	Enter the cluster name for the VSAM data set, and then click **Validate**.The server searches the catalog on the mainframe to confirm that the data set exists. If the data set exists, a dialog displays the data set type.
Post-Read Exit Name	To manipulate the data after reading it from the source file, enter the name of the post-read exit to use. This is the custom exit routine that is installed on the Server and that is used to perform additional processing after a record is read from the data source.
Alternate Indexes	If the VSAM file has been defined to include alternate indexes, you can click **Get** to add index information to the virtual table, or you can click **Delete** to remove the information. Alternate indexes are used to improve query performance when the search criteria includes columns that are not part of the primary index. Alternate indexes have an indirect relationship to the cluster name, but they must be defined separately. If you are using a KSDS VSAM or ESDS cluster, you can specify alternative indexes that are associated with the cluster.
Advanced	To divide the data into logical partitions and process the partitions in parallel, click **Advanced**. Enter a **Thread Count** value for **MapReduce (Server Parrallelism Settings)**. The number of zIIP processors is checked at runtime, and one thread is used for each processor discovered. The value you specify overrides the default value (2) and the discovered value.

11. Click **Finish**.

What to do next

You can use virtual tables to generate SQL queries.

Chapter 7. Accessing data in System Management Files

Get access to data in System Management Files (SMF) using SMF virtual tables.

About this task

To get access to SMF data, the option to include pre-created SMF virtual tables must be selected during the product software installation.

Procedure

1. From the Server view, expand **SQL** > **Data** > **server name** > **Virtual Tables**.
2. Right-click the SMF virtual table from which you want to access the data.
3. Right-click **Generate Query**, and then review the resulting SQL statement. If necessary, you can modify the statement to meet your needs. The following example shows a generated SQL statement:

```
-- ---------------------------------------------------------------
-- This statement will return all rows and all columns from the
-- following table:
-- Name         : SMF_03000
-- Catalog      : null
-- Schema       : DVSQL
-- Remarks      : DATA - SMFDATA
-- Tree Location: DEV1/1410/SQL/Data/ADBV/Virtual Tables/SMF_03000
-- The sql statement:
SELECT SMF_LEN, SMF_ZERO, SMF_FLAG, SMF_RTY, SMF_TIME, SMF_SID, SMF_SSI,
  SMF_STY, SMF_SEQN, SMF30SOF, SMF30SLN, SMF30SON, SMF30IOF, SMF30ILN, SMF30ION,
  SMF30UOF, SMF30ULN, SMF30UON, SMF30TOF, SMF30TLN, SMF30TON, SMF30COF, SMF30CLN,
  SMF30CON, SMF30AOF, SMF30ALN, SMF30AON, SMF30ROF, SMF30RLN, SMF30RON, SMF30POF,
  SMF30PLN, SMF30PON, SMF30OOF, SMF30OLN, SMF30OON, SMF30EOF, SMF30ELN, SMF30EON,
  SMF30EOR, SMF30RVD, SMF30EOS, SMF30DRO, SMF30DRL, SMF30DRN, SMF30ARO, SMF30ARL,
  SMF30ARN, SMF30OPO, SMF30OPL, SMF30OPN, SMF30OPM, SMF30UDO, SMF30UDL, SMF30UDN,
  SMF30UDS, SMF30RMO, SMF30RML, SMF30RMN, SMF30RMS, SMF30MOF, SMF30MLN, SMF30MNO,
  SMF30MOS, SMF30CDO, SMF30CDL, SMF30CDN, SMF30USO, SMF30USL, SMF30USN,
  CHILD_KEY, BASE_KEY
FROM SMF_03000 LIMIT 1000;
```

4. Optional: Execute the SQL statement to view, test, or save the resulting data.

What to do next

Get the code to use in your programs and applications by creating a SQL class from the virtual table.

Chapter 8. Generating and executing SQL queries

To test SQL access to your data, generate the SQL query from an existing virtual table, and then execute the query to view the results.

Before you begin

To view or change how SQL results are displayed in the Studio SQL Results view, from the **Window** menu, select **Preferences** > **Data Service**. The following SQL preferences determine how SQL results display:

- **SQL Generate Query Behavior**: Determines whether you are prompted to execute SQL, or if SQL executes automatically.
- **SQL Results View Max Rows**: Maximum number of rows to return in the SQL Results view. The default value is 1000.
- **SQL Results Max Bytes**: Maximum Bytes size amount of data to return in the SQL Results view.
- **SQL Results values accessed as**: String or object.

You can choose to view SQL results in the SQL Results view or, for large results sets, you can choose to save the results to a `.csv` file, and then open and view results using Microsoft Excel.

Avoid returning large result sets that are memory intensive by editing the SQL statement prior to executing the statement.

Procedure

1. On the **Server** tab, right-click the virtual table, and select **Generate Query**. The generated query selects all columns that do not have a **`WHERE`** predicate. The SQL **`SELECT`** statement that displays lists the columns that you can use to create the data definition language (DDL) statements, and that are used to define the DB2 table and the target table.
2. Optional: In the **Generated.sql** view, modify the SQL to select only the data that you want. Any ANSI compliant SQL is acceptable.
3. Optional: To view or test the data that the SQL statement returns, right-click the highlighted **`SELECT`** statement, and then click either **Execute SQL** to view results in the SQL Results view, or **Execute SQL and File results** to save the results in a `.csv` file.
4. Optional: To create a virtual view of the SQL, highlight the **`SELECT`** statement, right-click and select **Create a virtual view**. Virtual views are helpful for more complex queries and for using JOINs.

Results

In the SQL results view:

- Double-click a row to view additional details about that row.
- Select the **Export Result Set** view option to export SQL results to a `.csv` file.
- Click **SQL Messages** to view query-related system messages.

By default, if a result set includes 25 or more columns, each set of 25 columns are displayed incrementally as groups. You can choose which group you want to view

using the **Columns Group** field. You can set the number of columns that you want to include in each group, ranging from 25-200, in the **Columns per group** field.

Chapter 9. Creating virtual views

If columns in your table are missing, or if you want to join columns from different tables, consider creating a virtual view.

Before you begin

The virtual tables representing the data that you want to access or join must already exist.

About this task

A *virtual view* is the SELECT statement that contains the columns from the source data that are used to read data directly from the data source. In some cases, creating virtual views is more convenient than regenerating and editing SQL each time. To create a virtual view, complete the steps that follow.

Procedure

1. In the Server view, expand **SQL** > **Data** > **Data Service server** > **Virtual Tables**.
2. Right-click the virtual table that represents the data that you want to access, and then select **Create Virtual View**.
3. In the **Name** field, enter a name for the virtual view.
4. From the Target drop-down, select the target to use for this virtual view.
5. Optional: In the **Description** field, enter a description.
6. Click **Next**.
7. In the Table Browser, expand the Virtual Tables folder, and then select an existing virtual table to use to compose the SQL Statement.
8. Click **Next**.
9. Optional: Review the resulting SQL statement and make any necessary modifications.
10. Click **Validate** to validate the SQL.
11. If valid, on the SQL Validation message that displays, click **OK**.
12. Click **Finish**.

Results

In the **Server** view, locate the new virtual view by expanding **SQL**> **Data Service serverData**> **Virtual Views**.

Chapter 10. Using Server Trace

Use the Server Trace view to record and view Data Service server messages.

To collect and view diagnostics for the client, run the **Gather Diagnostics** wizard, which saves the information to a `.zip` folder.

Enabling Server Trace for the Studio

You can include Studio calls in your server trace results. The Server Trace preference is enabled by default.

Before you begin

You must be able to connect to the Data Service server from which you want to collect trace information.

Procedure

1. From the **Window** menu, select **Preferences**.
2. In the tree listing preferences, click **Data Service**.
3. To enable tracing, select the **Enable Server Tracing of Studio Calls** check box. Tracing is enabled by default.
4. In the Studio **HTTP Debug Option** drop-down, select one of the following HTTP debug options:

Option	Description
Off	Do not collect HTTP messages. All trace activities are deactivated, including interactive tracing.
Normal	Commands that complete with a failure status and are traced after execution, including the return codes.
All	All instructions are traced before execution.
Commands	All commands are traced before execution. Return codes are also traced for commands that complete with an error or failure status.
Error	Commands that complete with error status are traced after execution, including the return codes.
Failure	Commands that complete with a failure status are traced after execution, including the return codes.
Intermediates	All instructions are traced before execution. All terms, intermediate results, and substituted variable names are traced during expression evaluation. The final results of any expression that is evaluated also displays. Values assigned by **`arg`**, **`parse`**, or **`pull`** instructions are also traced.
Labels	Shows all labels when executed.

Option	Description
Results	All instructions are traced before execution. The final result of any expression that is evaluated also displays. Values assigned by **arg**, **parse**, or **pull** instructions are also traced.

Starting Server Trace

Start tracing Data Service server records in the Server Trace view.

Before you begin

Before running a server trace, you must be able to connect to the Data Service server from which you want to collect trace information.

Procedure

1. From the **Studio Navigator** view, on the **Common Tools** tab, click **Server Trace**.
2. In the **Server Trace** view, verify that the current server is the correct server from which to collect trace messages, or to select a different server, click **Set Server** .
3. To start tracing, click **Play** (the blue arrow). The **Server Trace** table displays trace records.
4. To open the Server Trace Zoom page, double-click a message. The page provides message details and allows you to search for specific details within the message.

Filtering Server Trace results

Use the **Profile** option to filter the records that display in the Server Trace view.

Before you begin

You must be able to connect to the Data Service server from which you want to filter trace information. You can set filtering criteria before or after you run a Server Trace. Your most current filtering selections are automatically saved as your default filtering profile.

Procedure

1. On the **Server Trace** view, click **Profile**.
2. On the Server Trace Profile page, enable the fields that you want to include in the results.
3. For each enabled field, click **Add** to further filter your results. You can either select from the values that display or enter the value when prompted.
4. Click **OK** to save changes to your profile and to apply profile to the results in the **Server Trace** table.

What to do next

Use the **Display** option to select and sort columns that display in the filtered table. You can also choose to export the trace results.

Using Server Trace Zoom

Use Server Trace Zoom to view Server Trace message details.

Before you begin

Server Trace must be running before you can use Zoom.

About this task

Use Zoom to view all details for a selected server trace message in the Server Trace view. The message ID, type, and description are displayed. You can also view existing control block details.

Procedure

1. In the Server Trace view, double-click the message for which you want to view Zoom details.
2. View message details on the Zoom dialog box, and choose from the following options:

Option	Description
Previous	Click **Previous** to search for the previous occurrence of the text string entered.
Next	Click **Next** to search for the next occurrence of the text string entered.
Search	Click **Search**, and then enter a search string. To search for the next occurrence of the text string, click **Search** again.
Close	Click **Close** to close the search dialog.

Searching Server Trace messages

You can search server trace message results for a particular text string or message ID.

Before you begin

You must start the Server Trace before you can begin searching within resulting Server Trace messages.

About this task

Procedure

1. On the **Server Trace** view, click the drop-down view menu, and then click **Search**.
2. On the Search dialog box, under **From**, select one the following options to specify how to search within the results:

Option	Description
First	Search for the first occurrence of the text string.

Option	Description
Last	Search for the last occurrence of the text string.
ID	Search starting from the message ID you enter.

3. Under **For**, enter the text string to search for within the message control blocks. Text strings cannot include spaces or special characters, and wild card searches are not supported.
4. Select **Previous** to find previous occurrences of the text string, or select **Next** to find the next occurrence of the text string.
5. Click **Search** to begin the search.

What to do next

View messages that meet the search criteria in the Server Trace view.

Labeling Server Trace messages

Create labels to bookmark server trace messages that you frequently access.

Before you begin

You must start the Server Trace before you can begin labeling records.

Procedure

1. In the **Server Trace view**, right-click the message that you want to label, and then select **Add Label**.
2. On the Message Label dialog, enter text for the **Label**, and then click **OK**.
3. Optional: In the Labels view, double-click the label to locate the message in the Server Trace view.

Exporting Server Trace messages

Use the Server Trace view to export server trace messages to a file.

About this task

You can export the resulting server trace messages from the Server Trace view. You can limit the number of messages that you can export into a file from Admin preferences.

Procedure

1. In the Server Trace view, click the drop-down view menu, and then select **Export**.
2. Under **Export Type**, select one of the following message export options:

Option	Description
Summary	Exports minimum message information: • Message ID • Date • Time • User ID • Message text
Full	Exports all available message information and all data about that message: • Message ID • Date • Time • User ID • Message text • Zoom
Comma Separated Format	Exports all table information to a `.cvs` file. This file type cannot be imported for viewing in the Server Trace view.

3. Under **Export Content**, select one of the following message content options:

Option	Description
Message ID Range	Select a range of messages to export by entering the first message ID in **From**, and the last message ID to include in **To**.
Transaction ID	Exports only those messages with the RRS transaction ID value that you specify.
Global Transaction ID	Exports only those messages with the RRS global transaction ID that you specify.
Connection ID	Exports only those messages that are associated with a specific client that is currently connected to the server.
Message ID List	Lists message IDs. This option is only available if the **Full** export type option is selected.

4. Click **Next**.
5. Click **Browse** to specify a file name and export location.
6. Click **Finish** to save the file.

Importing Server Trace messages

Using the **Import File Viewer** tab, you can import and view Server Trace messages.

Before you begin

Server Trace must be running before you can import a file.

About this task

You can import the `.isx` files server trace messages and view those messages on the **Import File Viewer** tab.

Procedure

1. On the **Import File Viewer** tab, click **Import**.
2. Navigate to the `.isx` file that you want to import, and then double-click the file. Messages and message details display on the **Import File Viewer** tab.
3. Optional: To view more details about a message, right-click on the message, and then select **Zoom**.
4. Optional: To change how the messages display, click **Display**.

Chapter 11. Preferences

Preferences allow you to customize several IBM® DB2 QMF® Data Service settings.

Preferences are categorized as follows:

Admin preferences

On the Admin screen, you can set the maximum number of Server Trace messages that you want to export.

From the **Window** menu, select **Preferences**>**Data Service> Admin**.

The default value for **Server Trace Export size limit** is 5000. Specifying a value greater than 5000 can cause a MAX CPU TIME EXCEEDED error to occur.

Console preferences

Use **Console** preferences to view or modify console display settings.

From the **Window** menu, select **Preferences**> **Data Service> Console**, and then enter the following information:

Fixed width console
Allows you to specify a fixed width for characters to display in the console.

Maximum character width
Specify the maximum character width for a fixed width console. The default setting is 80.

Limit console output
Allows you to limit the console buffer size by number of characters.

Console buffer size (characters)
Specify the number of characters to limit the buffer size. The default setting is 80000.

Console entry size limit (characters)
Specify console entry size number limit of characters. The default setting is 500.

Dictionary preferences

Use Dictionary preferences to add or delete reserved words in dictionaries, and add or delete dictionaries, based on languages being used.

From the **Window** menu, select **Preferences**> **Data Service> Dictionary**, and then enter the following information:

Dictionary
Lists the default dictionaries. You can add new dictionaries or delete existing dictionaries.

Reserved word
Lists reserved words for each dictionary. You can add new words to the list, or delete existing words.

Driver preferences

Driver preferences allow you to specify the default location of the driver configuration files.

JDBC Driver Settings

The installer attempts to identify the location of the driver configuration files. You can choose to specify a new location.

Note: You can also access data sources that are stored in other configuration files, by adding those configuration files from the Client view.

From the **Window** menu, select **Preferences**> **Data Service> Driver**, and then enter the following information:

Enable Tracing
Enables tracing for the JDBC driver. This setting is disabled by default. If you change this option, you must restart the studio to complete the change.

Default DSN File
Specifies the default location of the DSN file. This file is used to store the JDBC connection definitions that are generated for use in the **Active Connections** view.

Connection Overrides
Allows you to override the connection settings that the studio uses when it creates JDBC connection definitions. Specify a single name-value pair or a semicolon-delimited list.

SSL preferences

Enable SSL to secure JDBC and HTTP network communications between the Data Service Studio and the Data Service server.

Note: See the Administrator Guide for Data Service server SSL requirements and configuration details.

To enable SSL, from the **Window** menu, select **Preferences** > **Data Service** > **SSL**, and then select **Use SSL for Studio-Server communications (JDBC and HTTP)**.

After enabling SSL, choose the protocol to use for communications and specify the server and client authentication requirements.

Communication

Protocol
Select the protocol version to use for communications between the Data Service Studio and the Data Service server. The default is TLS 1.2.

Server Authentication strategies

Select the authentication strategy to use:

Require Server Validation

To require that all Data Service server certificates be authenticated, enable **Require Server Validation**, and then enter the following information:

- **Truststore**: The path name of the file on the local machine. The file must contain the Data Service server CA (certificate authority) certificate.
- **Password**: The password for the truststore file.
- **Type**: The truststore file type. For example: JKS, PKCS12, BKS, UBER.

Allow Self-Signed Certificates

To allow the Data Service server use of self-signed certificates, enable **Allow Self-Signed Certificates**.

- **Truststore**: The path name of the file on the local machine. The file must contain the self-signed server CA (certificate authority) certificate.
- **Password**: The password for the truststore file.
- **Type**: The truststore file type. For example: JKS, PKCS12, BKS, UBER.

Trust All

To allow all Data Service server certificates, enable **Trust All**. If enabled, the Data Service Studio does not validate the server certificate.

Client Authentication

To enable client authentication by the Data Service server, select **Enable Client Authentication**, and then enter the following information:

Keystore

The path name of the file on the local machine. The file must contain a client certificate which has been signed by the server CA.

Password

The password for the keystore.

Type The keystore file type. For example: JKS, PKCS12, BKS, UBER.

Alias Click **Refresh** to confirm that the password is valid and that the alias (label) appears.

To save SSL preferences, click **Apply**, and then click **OK**. When setting the Data Service server, ensure that you enter the secure port number.

Notices

This information was developed for products and services offered in the US. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as shown below.

© (*your company name*) (*year*).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. (*enter the year or years*).

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at http://www.ibm.com/legal/copytrade.shtml.

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

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

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

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

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

Terms and conditions for product documentation

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

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

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

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

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

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

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

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

Privacy policy considerations

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

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

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

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

Index

Product Number: 5697-QM2
5650-DB2
5615-DB2

Printed in USA

SC27-8886-00