Updates that apply to IBM® DB2® Analytics Accelerator Loader for z/OS® V2R1 User's Guide (SC27-6777-00)

Date of change: January 2018
Topic: Multiple
Change description: Documentation changes made in support of PTF UI53496 APAR PI89079 – Enhancements for Accelerator Loader server

The following topics have been updated:

In chapter "Overview":
Topic "What's new"

In chapter "Customizing DB2 Analytics Accelerator Loader":
Topic "APF-authorizing the load libraries (required)"
Topic "Modifying the server configuration member for IMS Direct"
Topic "Configuring generation data set retrieval"

In chapter "Loading data from non-DB2, remote DB2, and remote system sources":
Topic "Creating virtual tables for IMS data"
Topic "Creating virtual tables for sequential data"
Topic "Generating JCL"

Chapter "Overview"

Topic: "What's new"
Add the following descriptions:

When generating JCL in the Accelerator Loader studio, you can specify an alternate authorization ID under which the DROP TABLE and CREATE TABLE statements will be executed. The new **SET CURRENT SQLID** field appears on the final page of the **Generate JCL to Load Accelerator** wizard. See "Generating JCL".

IMS Direct supports access to multiple IMS subsystems. For updated procedures, see "Modifying the server configuration member for IMS Direct" and "Creating virtual tables for IMS data".

You can specify a generation data group base name when defining a virtual table, which will cause all active generations of the group to be read. Using a virtual table rule, you can also read only a subset of a generation data group. See "Creating virtual tables for sequential data" and "Configuring generation data set retrieval".

You must APF-authorize the *hlq*.SHLVRPC library. See "APF-authorizing the load libraries (required)".

In the Accelerator Loader studio, when generating JCL to load the accelerator, the option **Enable Loader Parallelism** now defaults to enabled. The default degree of parallelism is set to 4. See "Generating JCL".

Chapter "Customizing DB2 Analytics Accelerator Loader"

Topic "APF-authorizing the load libraries (required)"
Add the following item to the list of libraries that must be APF-authorized:

- *hlq*.SHLVRPC (Any data set allocated to ddname HLVRPCLB needs to be APF-authorized.)

Topic "Modifying the server configuration member for IMS Direct"
Replace this topic with updated content. The updated topic is located at the end of this document.

Topic "Configuring generation data set retrieval"
Add this new topic. The topic is located at the end of this document.

Chapter "Loading data from non-DB2, remote DB2, and remote system sources"

Topic "Creating virtual tables for IMS data"
Replace this topic with updated content. The updated topic, which includes three new subtopics, is located at the end of this document. Organize the topics as follows:
- "Creating virtual tables for IMS data"
  - "Using the IMS DBD Metadata wizard"
  - "Using the IMS PSB Metadata wizard"
  - "Using the IMS Virtual Table wizard"

Topic "Creating virtual tables for sequential data"
Replace this topic with updated content. The updated topic is located at the end of this document.

Topic "Generating JCL"
Replace this topic with updated content. The updated topic is located at the end of this document.

Modifying the server configuration member for IMS Direct

To optionally configure IMS Direct, configure IMS Direct parameters in your Accelerator Loader server configuration file.

About this task

Use this procedure to configure optional IMS Direct parameters in your Accelerator Loader server configuration file.

IMS Direct supports access to multiple IMS subsystems and calls to compression exits, each of which requires additional configuration.

Using exits:

If you use compression exits, you can configure the server to call these exits, providing optimization.

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

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

Procedure

1. In data set *hlq*.SHLVEXEC, locate member *hlvid*IN00, where *hlvid* represents the name of the Accelerator Loader server started task that was customized by using Tools Customizer.
2. In the *hlvid*IN00 member, locate the comment "Enable IMS Direct Map Reduce."
3. (Optional) To access additional IMS subsystems with IMS Direct, add a DEFINE IMSDBINFO statement for each additional IMS subsystem.

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

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

Parameter	Description	Valid values
IMSID	The IMS subsystem identification.	Up to 4-character ID.
SUFFIX	The setting of the SUF= keyword used in the IMS Control Region.	One character. Default value is I.
ACBLIB	ACBLIB data sets contain the application control blocks (ACBs), which describe IMS applications, and data management blocks (DMBs), which describe databases and the applications that can access them.	your.ACBLIB
DFSRESLB	Load library that contains the major IMS modules.	your.SDFSRESL
IMSDALIB	Dynamic Allocation Library for IMSDBs and RECONs.	your.dynamic.allocation.lib
MODBLKS	Used to support dynamic resource definition. Contains the APPLCTN, DATABASE, RTCODE, and TRANSACT macros.	your.MODBLKS
RECON1	Primary RECONciliation dataset, which holds all of the resource information and event tracking information that is used by IMS.	your.RECON1
RECON2	An active copy of RECON1.	your.RECON2
RECON3	Spare RECON to be used when RECON1 or RECON2 are not useable.	your.RECON3

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

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

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

5. To call a compression exit, perform one of the following steps as appropriate:
   - If your compression exit must be called in TCB mode but can run properly without serialization, specify your exit name in the following statement:

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

     where *n* is a number from 1 to 10 and *exitname* is the name of the compression exit routine.
   - If your exit can run properly in SRB mode without serialization, specify your exit name in the following statement:

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

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

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

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

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

Configuring generation data set retrieval

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

About this task

To read only a subset of a generation data set (GDG), you must enable virtual rule HLVGDGS1 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 HLVGDGS1 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 dataset.

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 HLVGDGS1.

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 Map Reduce 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 Map Reduce 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 Map Reduce 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 server configuration member (*hlvid*IN00) 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 Accelerator Loader - 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 HLVGDGS1 and pressing Enter.

4. Set the rule to Auto-enable by specifying A next to HLVGDGS1 and pressing Enter. Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Creating virtual tables for IMS data

Create a virtual table that maps to the IMS data that you want to access, and from which the SQL used to access the data is generated and executed.

Before you begin

The Program Specification Block (PSB) and Database Definition (DBD) source members, and the copybooks for each segment must exist in the virtual source libraries defined to the server. For details, see Creating virtual source libraries .

To use the IMS Direct feature, the `IMSDIRECTENABLED` parameter must be enabled in the *hlvid*IN00 file.

About this task

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

Procedure

1. Expand the **SQL** > **Data** > *SSID* node, where *SSID* is the name of your server.
2. Right-click **Virtual Tables** and select **Create Virtual Table(s)**.
3. Under **Wizards**, select the **IMS** wizard and click **Next**.
4. On the **New IMS virtual Table(s)** page, create metadata for an IMS virtual table by completing the following steps:
   a. Choose a DBD by doing one of the following steps:
      - Select a **DBD** from the drop-down list.
      - If your DBD does not appear in the drop-down list, click **Extract DBD** to create the requisite metadata. The **New IMS DBD Metadata Wizard** launches. See "Using the IMS DBD Metadata wizard" on page 20.
   b. Choose a PSB by doing one of the following steps:
      - Select a **PSB** from the drop-down list.
      - If your PSB does not appear in the drop-down list, click **Extract PSB** to create the requisite metadata. The **New IMS PSB Metadata Wizard** launches. See "Using the IMS PSB Metadata wizard" on page 21.
   c. Click **Create Virtual Table** to create a virtual table for an IMS segment in the selected DBD and PSB. The **New Virtual Table Wizard** launches. See "Using the IMS Virtual Table wizard" on page 22.

      Note: Both the DBD and PSB must be defined for this button to be enabled.
5. Click **Finish**.

What to do next

Use the studio to easily compose and execute SQL queries using your new virtual tables. See Generating and executing SQL queries.

Using the IMS DBD Metadata wizard

Use the **New IMS DBD Metadata Wizard** to create DBD server metadata.

About this task

This wizard is used to create server metadata containing information extracted from the selected DBD source. This DBD metadata is a prerequisite for creating IMS virtual tables. The name of each DBD map will be determined from the contents of the DBD source.

Procedure

1. On the **New DBD Metadata** page, complete the following fields and click **Next**:

Field	Action
Metadata Library	From the drop-down list, select the target library where the DBD metadata will be stored (for example, *hlq*.USER.MAP). The target libraries are specified in the server's started task JCL.
Description	Enter an optional description.

2. On the **Source Download** page, complete the following fields and click **Next**:

Field	Action
Available Source Libraries	From the list of **Available Source Libraries**, select the virtual source library that contains the DBD source member.
Source Library Members	Select the DBD that you want to use and click **Download** to copy the member from the mainframe to your desktop. Use **Filter patterns** to filter the list.
Downloaded Source Files	Review the list of downloaded members and ensure that the check box for the DBD that you want to use has been selected.

3. On the **Data Layout** page, complete the following fields and click **Next**:

Field	Action
Source	Expand the source file to verify that it displays the expected database definition (DBD).
Start Field	Accept the default root start field, or if multiple DBD nodes are present in the source tree, you can click on one of the DBD nodes to indicate that you only want to map that one DBD.
End Field	**End Field** selection is disabled when extracting DBD source.

4. On the **IMS Server configuration** page, complete the following fields:

Field	Action
• **Use IMS/DBCTL (read/write, transactional integrity)** • **Use IMS-Direct (read-only, high performance bulk data access)**	Select the IMS protocol to use. Choose **Use IMS/DBCTL (read/write, transactional integrity)** to use IMS API calls. Choose the default option **Use IMS-Direct (read-only, high performance bulk data access)** to enable IMS Direct for the DBD. To use this feature, IMS Direct must also be enabled in the *hlvid*IN00 file. You must select this option for the DBD to be able to enable IMS Direct for a virtual table.
IMS ID Override (used with IMS-Direct only)	Specify the IMS ID of the IMS subsystem to use when multiple IMS subsystems are defined for use with IMS Direct. This value will override the default IMS ID in the DBD map.
Advanced	When reading large volumes of data from tables, click **Advanced** to display and configure the **MapReduce** feature. The **MapReduce** feature enables you to divide the data into logical partitions and process those partitions in parallel using the **Thread Count** value. At runtime, the number of zIIP processors is verified and one thread is used for each zIIP processor; resulting in improved performance. The **Thread Count** value you specify overrides the default value (2) and the discovered value. To disable **MapReduce**, select the **Disable MapReduce** check box.

5. Click **Finish**.

What to do next

Return to the **New IMS Virtual Table(s)** page and define the IMS PSB. See
Chapter 5, "Creating virtual tables for IMS data," on page 19.

Using the IMS PSB Metadata wizard

Use the **New IMS PSB Metadata Wizard** to create PSB server metadata.

About this task

This wizard is used to create server metadata containing information extracted
from the selected PSB source. This PSB metadata is a prerequisite for creating IMS
virtual tables. The name of each PSB map will be determined from the contents of
the PSB source.

Procedure

1. On the **New PSB Metadata** page, complete the following fields and click **Next**:

Field	Action
Metadata Library	From the drop-down list, select the target library where the PSB metadata will be stored (for example, *hlq*.USER.MAP). The target libraries are specified in the server's started task JCL.
Description	Enter an optional description.

2. On the **Source Download** page, complete the following fields and click **Next**:

Field	Action
Available Source Libraries	From the list of **Available Source Libraries**, select the virtual source library that contains the PSB source member.
Source Library Members	Select the PSB that you want to use and click **Download** to copy the member from the mainframe to your desktop. Use **Filter patterns** to filter the list.
Downloaded Source Files	Review the list of downloaded members and ensure that the check box for the PSB that you want to use has been selected.

3. On the **Data Layout** page, complete the following fields and click **Next**:

Field	Action
Source	Expand the source file to verify that it displays the expected program specification block (PSB).
Start Field	Accept the default root start field, or if multiple PSB nodes are present in the source tree, you can click on one of the PSB nodes to indicate that you only want to map that one PSB.
End Field	**End Field** selection is disabled when extracting DBD source.

4. Click **Finish**.

What to do next

Return to the **New IMS Virtual Table(s)** page and create the virtual table. See Chapter 5, "Creating virtual tables for IMS data," on page 19.

Using the IMS Virtual Table wizard

Use the **New Virtual Table Wizard** to create a new IMS virtual table.

About this task

This wizard is used to map an IMS segment using a copybook representation to produce a new IMS virtual table.

Procedure

1. On the **New IMS Virtual Table** page, complete the following fields and click **Next**:

Field	Action
Name	Enter a unique name. The name can contain a maximum of 50 characters. The name must consist of an uppercase letter followed by zero or more characters, each of which is an uppercase letter, a digit, or the underscore character.
Metadata Library	From the drop-down list, select the target library where the virtual table metadata will be stored (for example, *hlq*.USER.MAP). The target libraries are specified in the server's started task JCL.
Description	Enter an optional description.
Convert VAR* fields to True VAR* fields	This is a deprecated field and should not be selected.
Arrays Handling	Select one of the following options: • **Flatten arrays into a single fixed table at runtime (Y)**: This option supports both OCCURS and OCCURS DEPENDING ON statements. • **Return arrays into separate tables at runtime (N)**: This option supports both OCCURS and OCCURS DEPENDING ON statements. A subtable is generated for each array. Subtables support SQL read access only.

2. On the **Source Download** page, complete the following fields and click **Next**:

Field	Action
Available Source Libraries	From the list of **Available Source Libraries**, select the virtual source library that contains the data structure definition that you want the virtual table to use.
Source Library Members	Select the PDS members that represent the data structures to include and click **Download** to copy the members from the mainframe to your desktop.
Downloaded Source Files	Select one or more previously downloaded members.

3. On the **Virtual Table Layout** page, complete the following fields and click **Next**:

Field	Action
Source	Expand the source file to verify that it correctly displays the source (member).

Field	Action
Start Field	Accept the default root start field, or expand the tree and select a specific start field to be used to map the IMS segment in question. (The IMS segment is chosen in Step 4 below.)
End Field	Accept the default root end field, or expand the file and select a different end field. By default, **End Field** is disabled.

4. On the **IMS Information** page, complete the following fields:

Field	Action
Segment Name	From the drop-down list, select the segment name.
• **Use IMS/DBCTL (read/write, transactional integrity)** • **Use IMS-Direct (read-only, high performance bulk data access)**	Select the IMS protocol to use. Choose the default option **Use IMS/DBCTL (read/write, transactional integrity)** to use IMS API calls. Choose **Use IMS-Direct (read-only, high performance bulk data access)** to enable IMS Direct on the virtual table. To use this feature, IMS Direct must also be enabled for the selected DBD and enabled in the *hlvid*IN00 file.
Advanced	When reading large volumes of data from tables, click **Advanced** to display and configure the **MapReduce** feature. The **MapReduce** feature enables you to divide the data into logical partitions and process those partitions in parallel using the **Thread Count** value. At runtime, the number of zIIP processors is verified and one thread is used for each zIIP processor; resulting in improved performance. The **Thread Count** value you specify overrides the default value (2) and the discovered value. To disable **MapReduce**, select the **Disable MapReduce** check box.

5. Click **Finish**.

What to do next

Return to the **New IMS Virtual Table(s)** page and if necessary create the next virtual table. See Chapter 5, "Creating virtual tables for IMS data," on page 19.

Creating virtual tables for sequential data

Create a virtual table that maps to the sequential data that you want to access, and from which the SQL used to access the data is generated and executed.

Before you begin

Before creating the virtual table, verify that the data set name exists and that the copybook exists in the source library.

Procedure

1. Expand the **SQL** > **Data** > *SSID* node, where *SSID* is the name of your server.
2. Right-click **Virtual Tables** and select **Create Virtual Table(s)**.
3. Under **Wizards**, select the **Sequential** wizard and click **Next**.
4. On the **New Virtual Table Wizard** page, complete the following fields and click **Next**:

Field	Action
Name	Enter a unique name. The name can contain a maximum of 50 characters. The name must consist of an uppercase letter followed by zero or more characters, each of which is an uppercase letter, a digit, or the underscore character.
Metadata Library	From the drop-down list, select the target library where the virtual table metadata will be stored (for example, *hlq*.USER.MAP). The target libraries are specified in the server's started task JCL.
Description	Enter an optional description.
Convert VAR* fields to True VAR* fields	This is a deprecated field and should not be selected.
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.

5. On the **Source Download** page, complete the following fields and click **Next**:

Field	Action
Available Source Libraries	Select the source library that contains the data structure to use.
Source Library Members	Select the PDS members that represent the data structures to include and click **Download** to copy the members from the mainframe to your desktop.
Download Source Files	Select one or more previously downloaded members.

6. On the **Virtual Table Layout** page, complete the following fields and click **Next**:

Field	Action
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. Generally, you want to map all the data. However, if the source file is a program listing from which you want to select specific information, select the start and end fields to only map to that information.
End Field	Accept the default root end field, or expand the file and select a different end field. By default, **End Field** is disabled.

7. Optional: On the **Virtual Table Redefines** page, accept the default table redefines or expand **Redefine** to modify your selection, and click **Next**.

8. On the **Data Source Details** page, complete the following data source fields and click **Next**:

Field	Action
Data Set Name	Enter the data set name you want to use. The following data set types are supported: • PDS or PDSE: Specify the partitioned data set name. This requires that you also enter a **Member** name prior to validating that the member name exists on the host. • Physical sequential: Specify the sequential data set name and click **Validate** to verify that the data set name exists on the host. • Generation Data Groups (GDG): Specify the GDG data set using the GDG syntax. For example: *hlq*.DATA.SEQ(-1). You can also specify a base GDG name so that all generations of the GDG will potentially be accessed. Click **Validate** to verify that the data set name exists on the host.
Member	If you selected a PDS or PDSE for the **Data Set Name**, you must also enter the member name to use. Click **Validate** to verify that the member name exists on the host.

Field	Action
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 is used to perform additional processing after a record is read from the data source.
Advanced	When reading large volumes of data from tables, click **Advanced** to display and configure the **MapReduce** feature. The **MapReduce** feature enables you to divide the data into logical partitions and process those partitions in parallel using the **Thread Count** value. At runtime, the number of zIIP processors is verified and one thread is used for each zIIP processor; resulting in improved performance. The **Thread Count** value you specify overrides the default value (2) and the discovered value. To disable **MapReduce**, select the **Disable MapReduce** check box.

9. Click **Finish**.

What to do next

Use the studio to easily compose and execute SQL queries using your new virtual tables. See Generating and executing SQL queries.

Generating JCL

Generate the JCL that loads the data to the accelerator from a virtual table, virtual view, or selected SQL statement.

Before you begin

- Have the following information available before starting this task:
  - Target server name
  - Target DB2 subsystem name
  - DB2 load library names
  - Accelerator name
  - Product data set names
  - JCL library name
  - Table creator name
  - Table name
- To enable the **Load Resume** feature, you must have IBM DB2 Analytics Accelerator for z/OS V4 PTF5 installed on the accelerator.
- If the source data is not hosted on the same LPAR as the accelerator, two accelerator servers are required:
  - The *data server* is the Accelerator Loader server that hosts the virtual table or view.
  - The *target server* is the Accelerator Loader server that has access to the target DB2 subsystem and the accelerator. The target server's *hlvid*IN00 file must contain a configuration entry for the data server. When you run the wizard to generate the JCL, the name of this configuration definition is referred to as the *Data Server Name*.

About this task

In the steps that follow, the information that you enter is only required the first time that you generate the JCL.

Procedure

1. On the **Server** tab, expand **SQL** > **Data**.
2. Expand **Virtual Views** or **Virtual Tables** to navigate to the virtual view or virtual table that represents the source data that you want to load.
3. Right-click the virtual view or table, and select **Generate JCL to Load Accelerator** to open the **Generate JCL to Load Accelerator** wizard.
4. On the **Source Information** page, review the source subsystem and the SQL query. If you choose to modify the SQL query, you must click **Validate** before you can proceed to the next page. If the query is not valid, an error message displays showing the source of the problem. The SQL query must be corrected before you can continue.
5. Click **Next**.
6. On the **Target Information** page, specify the following information:

Field	Action
Target Server Selection	- Choose **Use Current Server** if the server that is hosting the virtual table or virtual view resides on the same LPAR as the accelerator. This is the default setting. - Choose **Use Alternate Target Server** if the server that is hosting the virtual table or virtual view does not reside on the same LPAR as the accelerator. If you choose this option, use **Set Server** to select the alternate server.
Target DB2 Subsystem	Accept the default DB2 subsystem name or choose a different subsystem to use from the drop-down list. This name must match the data server name that is configured in the target server *hlvid*IN00 file. The subsystem that you choose will display as the default setting the next time that you run the wizard.
Table Creator	Accept the **Table Creator** name that is displayed, or select a different name from the drop-down list.
Table Name	Accept the **Table Name** that is displayed, or select a different name from the drop-down list.
Available Accelerators	From the list of **Available Accelerators**, select the accelerators to use. If an accelerator is associated with a group, the **Group** label is displayed next to the accelerator name. You can select up to eight accelerators to load data simultaneously.

Field	Action
CREATE TABLE DDL Options	Choose to include any of the following optional table DDL options: • **Database Name** – Enter the name of the database to use when loading data to the accelerator. • **Table Space Name** – Enter the name of the table space to use when loading data to the accelerator. • **CCSID** – Use this option to override the default CCSID of the target database. This option generates a CCSID clause on the CREATE TABLE DDL statement for the target table. DB2 has a default CCSID set in the ZPARM ENSCHEME that is used if the encoding scheme or CCSID is not specified in the DDL. The CCSID clause is required when the desired encoding scheme of the target table is different from the default value in the ZPARM. Select one of the values from the drop-down list: `ASCII`, `EBCDIC`, `UNICODE`. Or, leave the option blank to omit the CCSID clause from the generated DDL. • **Accelerator Only Table** – If this table is used to only load data to the accelerator and not to simultaneously load data to DB2 for z/OS, select **Accelerator Only Table**. • **Enable Unicode Column Expansion** – When converting EBCDIC to Unicode, select this option to increase the precision of maximum length columns by a factor of 3. This option ensures that conversion of EBCDIC to Unicode cannot cause an overflow condition due to character expansion. – When this option is clear (default), the DDL for the target table on the source character columns is VARCHAR(n), where n is the maximum size of the source EBCDIC column. – When this option is selected, the DDL for the target table on the source character columns is VARCHAR($n*3$), where n is the maximum size of the source EBCDIC column. The SELECT statement (specified on the **Source Information** page) is adjusted accordingly in the generated JCL. A cast will be applied to each source character column in the SELECT statement, as follows: `... CAST(`*source-column-name* `AS VARCHAR(`*n*`*3)) ...` Where n is the maximum size of the source EBCDIC column.

Field	Action
Enable Loader Parallelism	To use parallel processing (MapReduce), select **Enable Loader Parallelism** and either enter the number of parallel tasks to use, or click **Fetch Existing Table Attributes** if you are using an existing accelerated table name and you want to reuse the same parallelism settings. This setting is optional and is enabled by default. The default degree of parallelism is 4.
Join Virtual Parallel Data Group (VPD)	Select **Join Virtual Parallel Data Group (VPD)** to join an existing group of target servers, and specify the following: • **Group Name** – Enter the name of the group to join. • **Number of members** – Enter the number of members (Accelerator Loader jobs) that are in the group. Although this setting is optional, it is recommended that you enter the number of Accelerator Loader jobs that will be used. • **Group Timeout** – Enter the timeout duration, in seconds. This controls the timing window starting from the point-in-time when the first VPD member query is made to the server, to the time that the VPD group is closed to further queries for this VPD group session. This setting is optional. • **I/O Task Count** – Enter the number of tasks to use for VPD. This setting is optional and is disabled by default.
DDL Preview	Click **DDL Preview** to preview the SQL CREATE statement that will be used to create the accelerated table. This step is optional.

7. Click **Next**.
8. On the **JCL Generation Details** page, specify the following information:

Field	Action
Target Server Group	Enter the name of the **Target Server Group** to use. This setting is optional.
Utility ID	Enter the target server **Utility ID** to use. This setting is optional.
STEPLIB DD Concatenation Libraries	In the following fields, enter the data set names of the Accelerator Loader load libraries that you want to use. • **Accelerator Loader Library (1)** • **Accelerator Loader Library (2)**

Field	Action
DB2 Load Libraries	To add a DB2 load library, click **Add** and enter the name of the DB2 load library. You can also choose an existing load library and click **Modify** or **Delete** to modify or delete the DB2 load library. This setting is optional.
DB2 Dynamic SQL Program (DSNTEP2)	Enter the required details for running the DSNTEP2 sample DB2 program: • **DB2 Load Library (RUNLIB)** • **Plan Name** (the default is DSNTEP2)
JCL Settings	Click **JCL Settings** to view or modify the JCL generation preferences in the **JCL Preferences** dialog and click **OK**. This setting is optional.

9. Click **Next**.
10. On the **Job Creation** page, specify the following information:

Field	Action
JCL Library	Select or enter the name of the JCL library to use. The JCL library that you choose is displayed the next time you open this page. If you do not plan to save the job on the host or if you do not know the location, leave this field blank. This setting is optional.
Member Name	Enter the name of the member to use to store your generated JCL.
Local File Name	Depending on the JCL library that you chose, you can choose to enter the name of a local file to use.
Job Name	Accept or modify the **Job Name**.

Field	Action

Job Step Details	Specify the following job options:
	
	• **DROP TABLE** – Select this option to drop the existing target table definition. Do not select this option if the accelerated table does not already exist.
	
	• **CREATE TABLE** – Select this option to create a new target table definition.
	
	• From the **LOAD REPLACE** drop-down list, select a load option to use:
	
	– **LOAD REPLACE** – Replaces the data in the table.
	
	– **LOAD RESUME** – Loads new data to an existing table. Selecting LOAD RESUME disables the DROP TABLE and CREATE TABLE options.
	
	• **SET CURRENT SQLID** – Specify an alternate authorization ID under which the DROP TABLE and CREATE TABLE statements will be executed in the generated Accelerator Loader job. This setting is optional. If this field is set, the generated JCL will include a statement to assign a value to the DB2 CURRENT SQLID special register. This statement appears as the first line of SYSIN in both the DROP and CREATE steps of the job, as follows:
	
	`//SYSIN    DD  *`
	`   SET CURRENT SQLID = 'MYSQLID';`
	`   . . .`
	`/*`

Additional Actions	Choose from the following actions: • **Open in Editor** – Select **Open in Editor** to review the JCL in the **JCL** editor. • **Auto-run using JCL View** – Select **Auto-run using JCL View** to automatically submit the JCL after it is generated. If the JCL library is specified in the JCL view, a copy of the JCL is saved in a PDS on the host. You can use this copy for future data loads. These settings are optional.

11. Click **Generate** to generate the JCL.