IBM Db2 Analytics Accelerator for z/OS
Version 7.1.9

User's Guide

SH12-7082-09

Note

Before you use this information and the product it supports, read the information in "Notices" on page 177.

Ninth Edition, October 2019

This edition applies to version 7.1.9 of IBM® Db2® Analytics Accelerator for z/OS® (product number 5697-DA7), and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces SH12-7082-08. Changes to this edition are marked with a vertical bar.

Contents

Figures

About this book

This book provides information about the following subjects:

- General information on components and objects used by IBM Db2 Analytics Accelerator for z/OS
- Conditions for query acceleration with IBM Db2 Analytics Accelerator for z/OS (supported queries, data types, functions, limitations, and so on)
- Connecting to a data server
- Adding accelerators
- Adding and maintaining tables for query acceleration with IBM Db2 Analytics Accelerator for z/OS
- Configuring and activating system traces
- Adding, removing, activating, and deactivating individual accelerators
- Component updates

Who should read this book

This book is intended for database professionals and database administrators who need to complete the following tasks with IBM Db2 Analytics Accelerator for z/OS:

- Accelerate Db2 for z/OS queries with IBM Db2 Analytics Accelerator for z/OS
- Complete administration tasks with the help of IBM Db2 Analytics Accelerator Studio

What's new in version 7?

IBM Db2 Analytics Accelerator for z/OS Version 7.1.0 is a newly designed product rather than a major upgrade. It runs on different, more powerful hardware, which allows unprecedented acceleration rates. It also comes with a new deployment option, which does not require external hardware, but runs exclusively in a dedicated LPAR on your existing IBM Z hardware.

- Common Db2 in-memory, columnar SQL engine with BLU acceleration and SQL compatibility improvements, for example:
  - Native support for the EBCDIC MBCS and GRAPHIC data types (instead of a conversion to UTF-8 as in earlier versions)
  - Acceleration of all types of correlated sub-queries (only small subset was offloaded in version 5), including table expressions with sideway references
  - Support for timestamps of precision 12 (these were truncated to precision 6 in earlier versions
  - Native support for the FOR BIT DATA subtype, for all types of table encoding (EBCDIC, UNICODE, ASCII). This was available for EBCDIC only in earlier versions).
  - Native support for the TIMESTAMP value 24:00:00 (this was mapped to 23:59:59 in earlier versions)
  - Improved support for scalar functions that were previously not offloaded when applied to certain data types
  - Support for MIN, MAX, DAY, LAST_DAY, BIT*, TIMESTAMP_ISO, VARIANCE, and STDDEV with the UNIQUE clause
  - Support for the HEX function
  - Improved support for tables in mixed encoding
  - EBCDIC tables can be added to accelerator even if UNICODE tables are already present
  - Better time synchronization through use of system heartbeat
  - Improved accuracy for CURRENT_TIME, CURRENT_TIMESTAMP, and CURRENT_DATE special registers
- Smart Load, meaning that the SYSPROC.ACCEL_LOAD_TABLES stored procedures optimizes the workload and the degree of parallelism for load jobs.
- New software update approach: Updates for the accelerator are not part of the z/OS installation package, but can be downloaded individually from IBM FixCentral, which significantly reduces the installation effort.
- New and changed stored procedures and interfaces:
  - SYSPROC.ACCEL_UPDATE_SOFTWARE2 replaces SYSPROC.ACCEL_UPDATE_SOFTWARE
  - The GetActivationLog sub-function extends SYSPROC.ACCEL_CONTROL_SYSPROC.ACCELERATOR, which allows you to monitor software activations.
  - New monitoring interface extensions for query details
  - SYSPROC.ACCEL_GET_QUERIES2 extends the SYSPROC.ACCEL_GET_QUERIES interface by returning query statements in a result set rather than in an output parameter of limited size. Query statements in the result set are not truncated if they exceed 128 bytes in length.
  - SYSPROC.ACCEL_GET_QUERY_DETAILS2 retrieves query plan information from the remote database management system in separate result sets. The information is not combined with the query text anymore.
  - SYSPROC.ACCEL_GET_QUERY_EXPLAIN returns the Db2 EXPLAIN information in a compressed, base64-encoded format.
  - The following stored procedures can write IFCID392 audit traces: SYSPROC.ACCEL_ADD_TABLES, SYSPROC.ACCEL_ALTER_TABLES, SYSPROC.ACCEL_REMOVE_TABLES

Enhancements in version 7.1.9

See the What's new document.

Enhancements in version 7.1.8

See the What's new document.

Enhancements in version 7.1.7

See the What's new document.

Enhancements in version 7.1.6

See the release notes.

Enhancements in version 7.1.5

See the release notes.

Enhancements in version 7.1.4

See the release notes.

Enhancements in version 7.1.3

See the release notes.

Enhancements in version 7.1.2

See the release notes.

Enhancements in version 7.1.1

See the release notes.

Accessibility features

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Db2 Analytics Accelerator Studio:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers

The IBM Db2 Analytics Accelerator for z/OS Information Center, and its related publications, are accessibility-enabled. The accessibility features of the information center are described in the topic Accessibility features in Eclipse.

Keyboard navigation

In addition to the standard Microsoft Windows navigation keys, this product uses keyboard shortcuts and controls of the Eclipse framework. For more information about these features, see the following website:

Navigating the user interface using the keyboard).

Interface information

The Eclipse framework supports the JAWS screen reader software. If you are facing difficulties in getting JAWS to read out table column headers, see the following websites:

- Configuring accessibility options for textual editors
- Tips for making user interfaces accessible

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Chapter 1. Introduction

IBM Db2 Analytics Accelerator for z/OS is a query accelerator capable of reducing response times for Db2 for z/OS queries by an order of magnitude.

IBM Db2 Analytics Accelerator for z/OS integrates deeply into your existing Db2 for z/OS environment and can process very large amounts of data. The addition of IBM Db2 Analytics Accelerator for z/OS makes Db2 for z/OS a truly universal database management system with optimum performance characteristics for transactional and analytical applications.

IBM Db2 Analytics Accelerator Studio is the graphical frontend of IBM Db2 Analytics Accelerator for z/OS. It offers functions for the configuration and administration, as well as analytical tools that collect performance metrics or troubleshooting information.

This chapter provides an overview of the basic tasks that can be completed with IBM Db2 Analytics Accelerator Studio.

Connecting to a data server

IBM Db2 Analytics Accelerator for z/OS needs access to your data servers to read the database catalogs and invoke the stored procedures for the accelerators. To connect to a database server, you must complete the tasks that are described in this section.

Creating a database connection profile

IBM Db2 Analytics Accelerator Studio is the user interface for administering accelerators. It connects to your Db2 for z/OS data server using Java™ Database Connectivity (JDBC), like other database clients. The configuration parameters for accessing a Db2 for z/OS data server are stored in database connection profiles. Create a database connection profile for each Db2 for z/OS data server with an accelerator.

Related tasks

Creating a database connection profile
Create a database connection profile to gain access to a Db2 subsystem on a database server. A Db2 subsystem houses one or more databases, in which the source data for query acceleration (schemas and tables) is kept.

Connecting to a database

To connect IBM Db2 Analytics Accelerator Studio to a Db2 for z/OS database, double-click one of the database connection profiles in the **Administration Explorer**.

Adding accelerators

When a connection to a database has been established through one of your database connection profiles, you can see all currently deployed accelerators in the **Object List Editor**. You access these interface elements from the **Accelerator Perspective** in IBM Db2 Analytics Accelerator Studio. To connect a newly installed accelerator, use the **Add Accelerator** wizard in IBM Db2 Analytics Accelerator Studio.
Related tasks

Completing the authentication using the Add Accelerator wizard

To complete the authentication, you specify the IP address, the port number, and the pairing code in the **Add Accelerator** wizard.

Defining the data to load into an accelerator

Defining the data to load into an accelerator basically means selecting the proper tables. However, you also need to complete a number of related sub-tasks. This section lists all required tasks in the correct order.

Related concepts

Distribution keys
A distribution key determines how tables rows are distributed among the available working nodes. IBM Db2 Analytics Accelerator for z/OS selects a suitable distribution key automatically when tables are loaded. The distribution varies between the two extremes: an even distribution (called random distribution) without the use of a key and a maximally skewed distribution that uses a distribution key.

Organizing keys
Organizing key also have an impact on the query response time. When choosing an organizing key, you select columns by means of which you group the rows of an accelerator-shadow table within the data slices on the worker nodes. This creates grouped segments or blocks of rows with equal or nearby values in the columns selected as organizing keys. If an incoming SQL query references one of the organizing key columns in a range or equality predicate, the query can run much faster because entire blocks can be skipped. It is not necessary to scan the entire accelerator-shadow table. Thus the time needed for disk output operations related to the query is drastically reduced.

How to select tables for query acceleration
Selecting the proper tables for a query to be accelerated is basically simple: You need to define the tables that are referenced by the query. Slightly more thought needs to be spent on the choice of proper distribution keys and organizing keys.

Related tasks

Specifying or changing a distribution key or organizing keys
Distribution and organizing keys have a considerable impact on the query response time.

Selecting the proper tables

In choosing the proper tables, you determine the data to be queried. For each Db2 for z/OS table that you select, a corresponding table is created on one or more accelerators. These tables are called accelerator-shadow tables. Accelerator-shadow tables are not necessarily copies of their originals. Often, they contain fewer columns than the tables they stem from because columns with unsupported data types are excluded. Accelerator-shadow tables must be filled with data before they can be queried. This is done in a later step (see loading accelerator-shadow tables).

Related tasks

Defining tables on an accelerator
Using the **Add Tables to Accelerator** wizard, you can define accelerator-shadow tables on an accelerator.

Checking the table size

When you select a table for an accelerator in the **Add Tables** wizard, IBM Db2 Analytics Accelerator Studio displays the table size. This table size is calculated on the basis of the values in the CARDF and AVGROWLEN columns of the SYSIBM.SYSTABLES table. Thus the metrics come from Db2 for z/OS. Comparing this size with the available disk space on your accelerator, you can roughly assess how much space the table will occupy on your accelerator. However, Db2 for z/OS and the accelerator use completely different compression algorithms, so the size of a table in Db2 might differ from the size of the same table on the accelerator.

Working with accelerator-shadow tables

To populate, maintain, and control accelerator-shadow tables, you complete the tasks that are described here.

Loading data into selected tables

To enable users to run accelerated queries against selected database tables, you must load the empty tables that you previously added to the accelerator with data. In doing so, you create a snapshot copy of your existing Db2 for z/OS data on the accelerator.

The Db2 Optimizer calculates the expected response times for incoming queries. If an accelerator can process a query against the selected tables faster than the database management system, the query is routed to the accelerator and evaluated against the populated accelerator-shadow tables.

You can reload accelerator-shadow tables, and, under certain conditions, also reload selected partitions of such tables. Doing so replaces the current data with a more recent snapshot, resulting in an update. When you reload tables or partitions, the accelerator automatically detects the tables and partitions that changed since the last load. This way, you can avoid loading the same data twice. The load wizard shows the load status for each table or partition and recommends maintenance activities.

Related concepts

Updating accelerator tables continually
The incremental update function of IBM Db2 Analytics Accelerator for z/OS allows you to update accelerator-shadow tables continually. Changes to the data in original Db2 for z/OS tables are thus propagated to the corresponding target tables with a high frequency and just a brief delay. This way, query results from an accelerator are always extracted from recent, close-to-realtime data.

Related tasks

Loading tables
Successful queries against accelerator-shadow tables are possible only if the tables contain data. Therefore, you must load the tables after their definition (empty structure) has been copied to the accelerator.

Enabling tables for query acceleration

Query acceleration can be switched on and off for each accelerator-shadow table. This allows you to do maintenance work on a particular table while query acceleration remains active for other tables on the same accelerator.

Disabling query acceleration for a table

Disabling a table, you can prevent accelerated queries against this table. This might be necessary when a query must be run against very recent data, which has not yet been copied to the accelerator.

Enabling tables for incremental updates

The incremental update function allows you to automatically update accelerator-shadow tables in an ongoing process. Updates to DB2® tables are then propagated to the corresponding tables on the accelerator with little delay, so that accelerated queries will always “hit” very recent data. You have to mark the tables that you want to include in this process.

Disabling incremental updates

Disabling incremental updates excludes tables from the incremental update process.

Archiving partition or table data

If table partitions in Db2 for z/OS contain data that is no longer updated (historical data), you can use the high-performance storage saver (HPSS) to move this data to an accelerator to free up costly storage on your Db2 for z/OS data server. A positive side-effect is that query processing and table maintenance become faster because the remaining data consists of fewer table rows and indexes require fewer entries.

Removing tables

If tables are no longer needed for queries, you can use the removal function in IBM Db2 Analytics Accelerator Studio to remove the accelerator-shadow tables from the accelerator.

Related tasks

Removing tables from an accelerator
Follow the steps in this topic to remove accelerator-shadow tables from an accelerator.

Order in which to complete tasks

See a diagram that shows the order of the most common tasks.

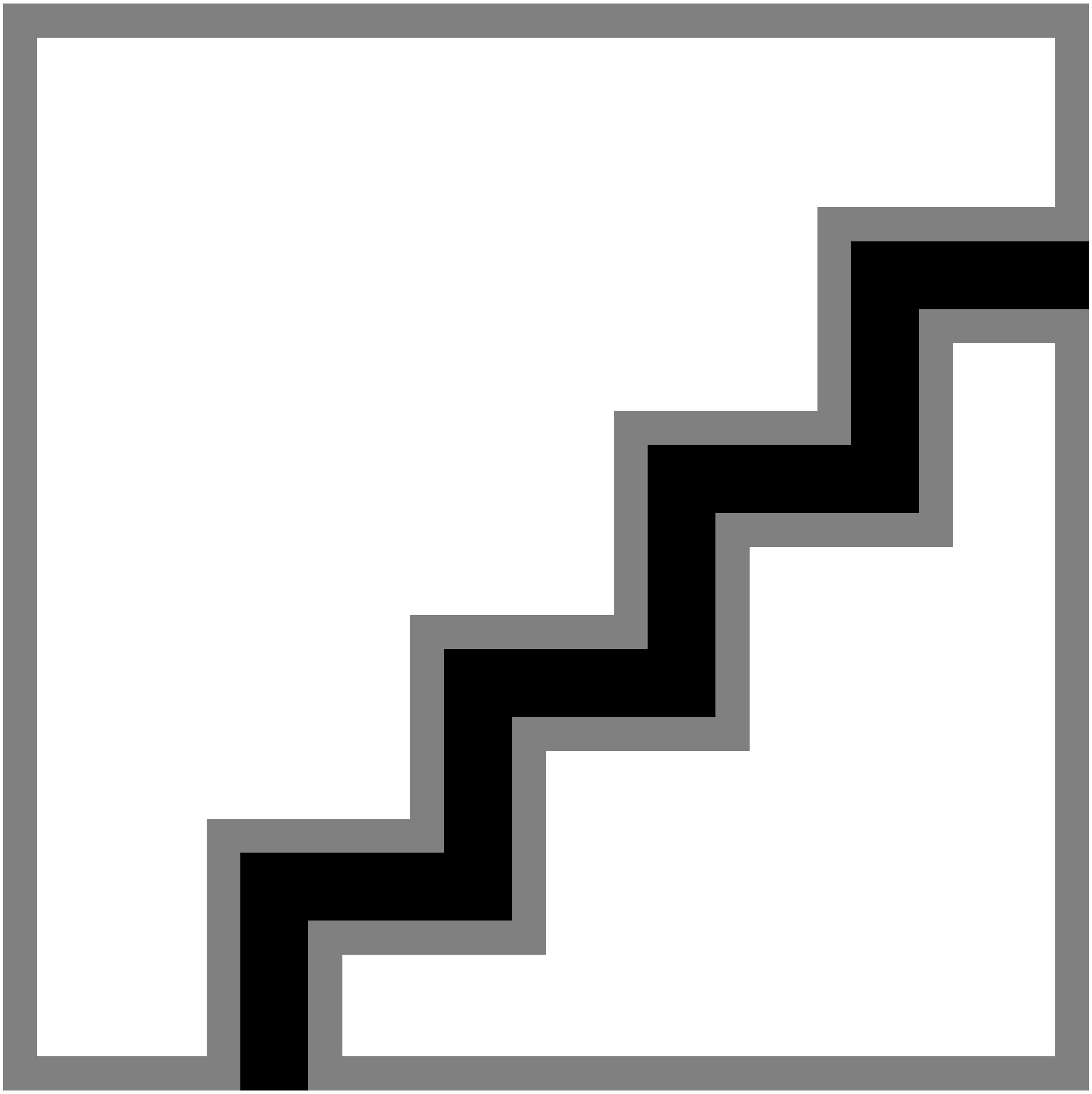


Figure 1. IBM Db2 Analytics Accelerator Studio tasks in the overall process flow

Chapter 2. Basic concepts

Get familiar with the basic concepts of IBM Db2 Analytics Accelerator for z/OS.

Accelerator

IBM Db2 Analytics Accelerator for z/OS consists of a high-performance hardware platform and an optimized database query engine. These components work together to support a variety of data analysis and business reporting tasks.

The high-performance query processing capabilities are embedded in a “shared nothing” architecture, which means that each worker node is able to process data independently, without network traffic or communication between the nodes. Hence there is hardly a contention for shared resources.

Each query is processed in parallel by all worker nodes, with each node processing a portion of each accelerator-shadow table. These portions are called *data slices*. The accelerator-shadow tables are thus horizontally partitioned because each worker node processes a data slice of each table.

Static SQL query support

Static SQL query support means that you can accelerate SQL queries that are embedded within Db2 for z/OS application programs.

Static SQL queries are prepared at the time the application program is bound as a package in Db2 for z/OS, that is, before the execution of the application in Db2. To accelerate static SQL queries within these application programs, you must BIND or REBIND the Db2 package for that application and specify a new bind option that requests static query acceleration.

Accelerator-shadow table

An accelerator-shadow table represents a table in your database. When you define accelerator-shadow tables in IBM Db2 Analytics Accelerator Studio, you instruct IBM Db2 Analytics Accelerator for z/OS to copy the definition of that table (an empty table) to an accelerator.

Queries including this table are then routed to the accelerator provided that the requirements for query redirection are met. A query can only be routed to an accelerator if the query refers to a subset or all of the tables on the accelerator.

Restriction: A table can be added to multiple accelerators, but an accelerated query using that table can run only on one accelerator at a time.

In addition, a query can only be accelerated if all tables that are referenced by the query reside on the same accelerator, if the accelerator-shadow tables are loaded, and if query acceleration is enabled for these tables. A query can reference tables indirectly if the table references a view or a table alias. For example, if the query references a view that joins two tables, both tables must be present on the accelerator.

When tables are added to an accelerator, Db2 for z/OS keeps track of these accelerator-shadow tables in its system catalog. The system table entries enable the optimizer of the database management system to complete the following tasks:

- Query matching, that is, comparing an incoming query with the information stored in the system tables to check whether the query can be handled by IBM Db2 Analytics Accelerator for z/OS.
- Query evaluation, that is, calculating the estimated query response time. If a performance gain can be predicted, the query is sent to IBM Db2 Analytics Accelerator for z/OS. If not, the query is handled by the database management system itself.

If a table contains columns with unsupported data types, these columns will not be included in the corresponding accelerator-shadow table. All other columns will be copied. So the job of defining a table will be completed even if some columns do not fulfill the criteria. However, queries that address columns with unsupported data types will not be accelerated.

Related tasks

Defining tables on an accelerator
Using the **Add Tables to Accelerator** wizard, you can define accelerator-shadow tables on an accelerator.

How to select tables for query acceleration

Selecting the proper tables for a query to be accelerated is basically simple: You need to define the tables that are referenced by the query. Slightly more thought needs to be spent on the choice of proper distribution keys and organizing keys.

Suppose that you want to accelerate the following query, which extracts the total number of items and the total amount of money paid by each customer from a set of sales tables. The query does not reach infinitely into the past; rather, it considers just the orders that were received after a certain date. The results will be grouped by customer number.

Note: The `O_TOTALPRICE` column denotes the total price per order. It is not the total price that is calculated by the query.

Query:

```
SELECT SUM(L_QUANTITY), SUM(O_TOTALPRICE), O_CUSTKEY
FROM TPCH.LINEITEM L JOIN TPCH.ORDERS O on L.L_ORDERKEY = O.O_ORDERKEY
WHERE O.O_ORDERDATE > '01.01.1994' GROUP BY O.O_CUSTKEY;
```

The query references the following tables:

- `LINEITEM (L)`
- `ORDERS (O)`

You would thus define these tables on the accelerator.

The query contains the following join predicate (join condition):

```
L.L_ORDERKEY = O.O_ORDERKEY
```

Suppose that `O.O_ORDERKEY` is a primary key and that the L table uses this key as a foreign key (`L.L_ORDERKEY`). To enable colocated joins, which lead to shorter query response times, the columns of accelerator-shadow tables must be on the same processing node. In order to achieve this, you would thus use `O.O_ORDERKEY` and `L.L_ORDERKEY` as distribution keys. Hence you would specify `O.O_ORDERKEY` as the distribution key for O and `L.L_ORDERKEY` as the distribution key for L.

Furthermore, the query contains the condition

```
WHERE O.O_ORDERDATE > '01.01.1994'
```

The `O.O_ORDERDATE` column makes a good candidate for an organizing key because its selection as a key of this type will group table rows with the same shipment date together. All rows with the same shipment date will be organized in blocks or clusters. When the query “encounters” a value that does not fulfill the condition, it can skip an entire block rather than individual rows. This decreases the time that is needed for scans of accelerator-shadow tables, and thus further decreases the query response time.

Access plan graph

If you are unsure whether the query refers to views or aliases rather than actual tables, you can display the access plan graph of the query.

In the access plan graph for this example, you can see that TPCH.LINEITEM and TPCH.ORDERS are tables because they appear in the access plan graph. Views and aliases are not represented in access plan graphs. See Figure 2 on page 7.

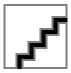


Figure 2. Access plan graph of the query in the example

Related concepts

Defining the data to load into an accelerator
Defining the data to load into an accelerator basically means selecting the proper tables. However, you also need to complete a number of related sub-tasks. This section lists all required tasks in the correct order.

Related information

Graphing the access plan for a query

Distribution keys

A distribution key determines how tables rows are distributed among the available working nodes. IBM Db2 Analytics Accelerator for z/OS selects a suitable distribution key automatically when tables are loaded. The distribution varies between the two extremes: an even distribution (called random distribution) without the use of a key and a maximally skewed distribution that uses a distribution key.

Random distribution

A distribution that is based on random numbers, which serve as input for a distribution algorithm. The output of this algorithm are numbers that identify data slices. Hence the rows to be processed can be assigned to certain data slices, which can be seen as chunks of data to be processed by the individual working nodes. The final result is a more or less even distribution across the working nodes.

Use of a distribution key (hash partitioning)

The accelerator uses a hash function on the key columns to determine the working node that receives a table row. Rows with equal hash values are distributed to the same working node.

Impact of the distribution key on the query performance

Parallel processing works best if the table rows to be processed are evenly distributed across the worker nodes. If the rows of a table that is referenced by a query are unevenly distributed, the worker nodes with fewer rows have to wait for the node with the biggest number of rows to finish processing. However, if your queries involve joins of big tables (100 million to billions of rows), and the rows of these accelerator-shadow tables are scattered across the working nodes even though they are evenly distributed in terms of quantity, you can easily spend much more time on data redistributions and broadcasts than you actually gain by the advantage of an even distribution.

Impact of distribution key on join performance

When two tables are joined, the query runs fastest if the matched rows of both accelerator-shadow tables reside on the same working node (collocated join). A collocated join is instigated by selecting the join columns as the distribution key for both tables.

However, if an accelerator-shadow table F1 is joined with two or more other tables D1, ... D*n* using different join keys, you can only locate one of the pairs (F1-D1, F1-D2, ..., F1-D*n*) on the same node because you can only select one of the join columns in table F1 as the distribution key for F1. In this case, the product will colocate the biggest tables.

Best practices for selecting distribution keys

Follow these guidelines when using distribution keys:

- For fact tables and the largest dimension tables, specify a distribution key. For all other tables, start with the default (random distribution).

- Prefer single-column distribution keys to multiple-column keys.
- Select a column as the distribution key that distributes the table rows as evenly as possible across the worker nodes, that is, select primary keys or columns with a unique index. Do not specify columns that only have a small number of distinct values. For example, Boolean values like *male* and *female* would place the table rows on two of the worker nodes only.
- When joining two large tables, try to colocate the tables by specifying the join columns as the distribution key for both tables. Note that to instigate a collocated join, the data types of the join columns in both tables must match.
- In general, do not specify a date, time, or timestamp column as the distribution key. Such keys would also distribute the table rows evenly across the worker nodes, but many queries restrict the number of result rows by a range predicate on the date, time, or timestamp column. For example, a query retrieving revenue figures from last month only would be processed on a single worker node if a column MONTH was selected as the distribution key because rows with equal values in this column would all be assigned to the same data slice.
- Although adverse to the goal of faster query acceleration, a distribution key on the primary key or unique constraint of a table helps if you must update tables continually. Therefore, decide what is more important to you: query performance or the latency of incremental updates. The accelerator automatically picks a distribution key when you enable incremental updates for a table unless you have defined a distribution key by yourself.

Organizing keys

Organizing key also have an impact on the query response time. When choosing an organizing key, you select columns by means of which you group the rows of an accelerator-shadow table within the data slices on the worker nodes. This creates grouped segments or blocks of rows with equal or nearby values in the columns selected as organizing keys. If an incoming SQL query references one of the organizing key columns in a range or equality predicate, the query can run much faster because entire blocks can be skipped. It is not necessary to scan the entire accelerator-shadow table. Thus the time needed for disk output operations related to the query is drastically reduced.

Organizing keys allow the query engine to identify the disk regions (addressable storage blocks on the disk) that contain the clusters of rows relevant to a query.

Best practices for selecting organizing keys

- In general, the accelerator should be able to process your queries with adequate performance so that organizing keys are not needed. However, the use of an organizing key, particularly on large fact tables can result in table scan performance gains by multiple orders of magnitude.
- An organizing key has no effect if the table is too small. The **Organized** column in the **Accelerator** view reflects this by not showing a value for the degree of organization (percentage).
- It is recommended to define an organizing key for accelerator-shadow tables that you want to update continually by using the incremental update function. Use the primary key columns of the table for the organizing key. When the incremental update function applies DELETE or UPDATE statements, the accelerator has to search for the rows that must be deleted or updated. Defining an organizing key accelerates these operations.
- Organizing keys work best on large tables (with millions or more records), under the condition that your queries restrict on column values that are rather scattered across the table. For example, if a query restricts on a set of product IDs and customer IDs from a table of sales records in which the records are presorted by the date on which a record was added, the customer ID column and the product ID column would make good candidates for an organizing key. These keys would have the effect that records with the same customer ID and product ID are grouped together, leading to the table scan performance benefits described earlier.

  Since restrictions on summary columns in dimension tables are, in many cases, automatically pushed down to the join column of a fact table, organizing keys on such columns in the fact table can be very beneficial.

An organizing key is also recommended if your history of data records reaches back into the past for an extended period, but the majority of your queries, in using a range predicate on a fact-table timestamp column or a parent attribute in a joined dimension, requires a constrained range of dates.

Example:

```
SELECT ... FROM ... WHERE TRANSACTION_DATE BETWEEN (<date1>, <date2>)
```

In this example, the TRANSACTION_DATE column would make a good organizing key.

As additional columns are chosen as organizing keys, the benefit of predicates on column subsets is reduced. Four keys are the allowed maximum. However, there is hardly a need to select more than three.

- Organizing keys are also useful the more frequently the columns that you specified as keys are used in query predicates, alone or in combination, and if the column cardinality is high (that is, if the columns have many different values).
- For organizing keys to have a positive effect on the table-scan performance, a query does not have to reference all the columns that have been defined as organizing keys. It is enough if just one of these columns is addressed in a query predicate. However, the benefit is higher if all columns are used because this means that the relevant rows are kept in a smaller number of disk regions.
- There is no preference for any of the columns that you specify and the order in which columns are selected does not matter either.
- Bear in mind, though, that clustered table rows cause a processing overhead when you load or update these tables on the accelerator.

Tables states

A table can go through various states while it is being processed by IBM Db2 Analytics Accelerator for z/OS. The meaning of these states is explained here. In addition, it is shown which operation you can run on a table in a particular state. Finally, a diagram shows the possible shifts from one state to another (state transitions).

Table states are displayed in IBM Db2 Analytics Accelerator Studio, in the **Accelerator** view for regular accelerators (**Tables** section, **Last Load** column). A table can be in one of the following states:

InitialLoadPending
State after the definition of an accelerator-shadow table on the accelerator, which indicates that the table must be loaded with data.

InitialLoadInProgress
State that indicates the loading of an empty accelerator-shadow table or the reloading of an accelerator-shadow table that was in the *Error* state.

Loaded
The load process has been finished.

SchemaModificationInProgress
The schema of a table is being modified. This happens when you change distribution key or organizing key columns, or if background jobs create foreign-key relationships with columns of the table.

UpdateInProgress
The data of the table is being updated. For example, an accelerator-shadow table is in this state if a reload has been enforced by various circumstances.

ReplicationInProgress
The table data is currently being synchronized by the incremental update function.

ArchivingInProgress
Db2 table data (data partitions) is being archived on an accelerator.

Error

State that indicates a processing error, after which an accelerator-shadow table is reset to the *InitialLoadPending* state. That is, a reload is required.

Possible operations

Table 1 on page 10 shows which operations you can run on a table in a particular state.

Table 1. Possible operations on a table depending on its state

Table state	Enable or disable query acceleration	Load or update	Schema modification	Remove
InitialLoadPending	No	Yes. The entire accelerator-shadow table is loaded or reloaded. Partition lists are ignored.	Yes	Yes
InitialLoadInProgress	No	No	No	Yes
Loaded	Yes	Yes	Yes	Yes
SchemaModificationInProgress	Yes	No	No	Yes
UpdateInProgress	Yes	No	No	Yes
ReplicationInProgress	Yes	Yes	No	Yes
ArchivingInProgress	Yes	No	No	No
Error	No	Yes. The entire accelerator-shadow table is loaded or reloaded. Partition lists are ignored.	Yes	Yes

State transitions

Figure 3 on page 11 shows the state transitions that are possible.

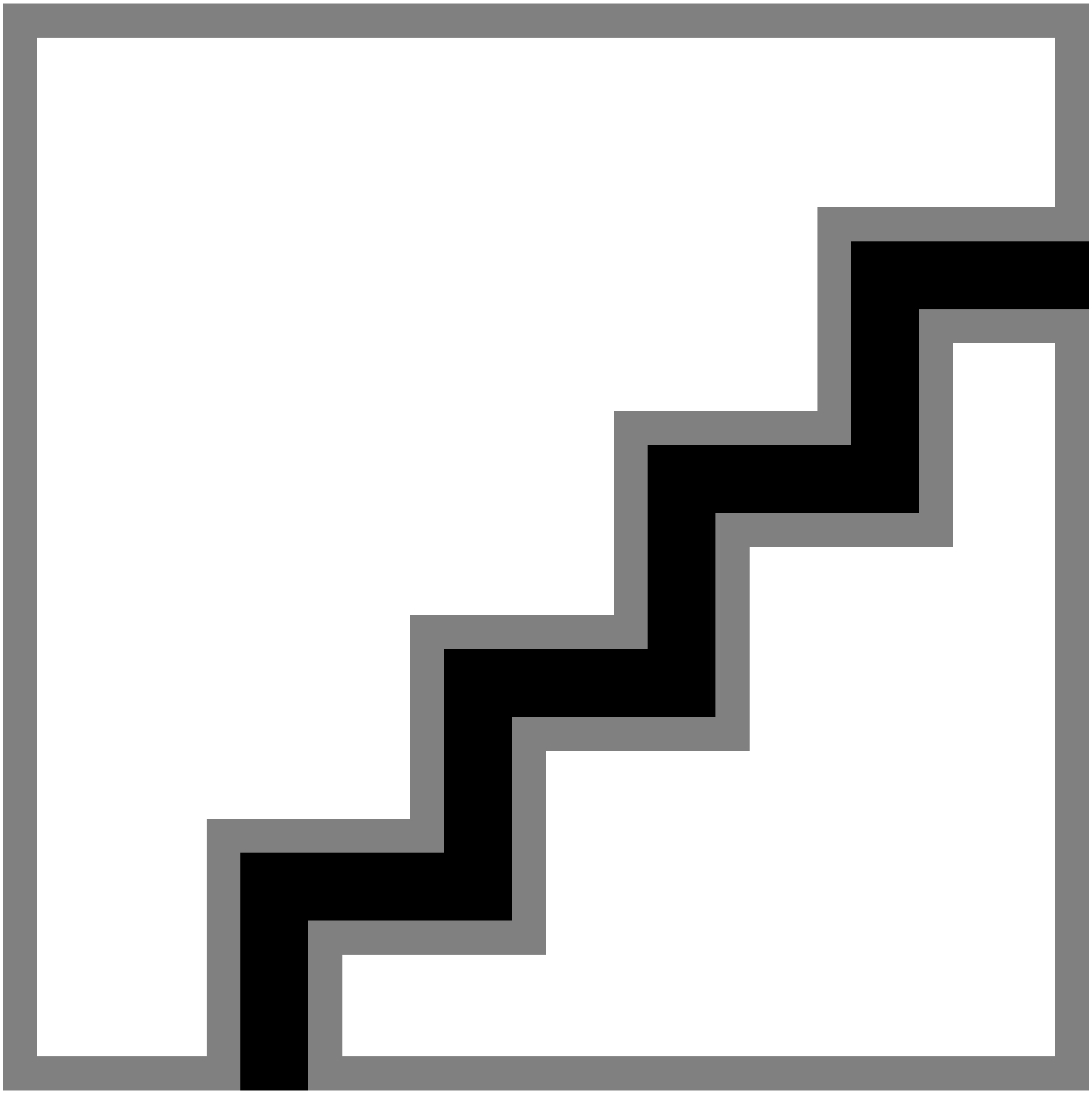


Figure 3. Possible table state transitions

Related concepts

Accelerator view

To show the tables on an accelerator, view the query history, or access various other functions, double-click an accelerator name in the **Object List Editor**, which opens the related **Accelerator** view.

Moved tables or partitions

These are table partitions or entire tables whose data was moved to an archive on an accelerator by the High-Performance Storage Saver. Moving table data saves storage space on System z®.

You can move partitions or entire tables (all partitions of a table) to an archive on an accelerator if the tables were previously defined on the accelerator (that is, if accelerator-shadow tables of these tables exist). As part of the move, the original Db2 data is deleted. The table space containing the tables or partitions in Db2 is set to a persistent (permanent) read-only state so that the original tables can no longer be updated or otherwise modified.

To be able to recover the data in an emergency case, an image copy is also created as part of the move. The recovery function of the High-Performance Storage Saver uses the image copies to restore the data to its original location in Db2. The High Performance Storage Saver is mainly used for historical data that is no longer actively used or maintained.

Normally, the data in moved tables and partitions is not included in accelerated queries. However, you can set the parameter `GET_ACCEL_ARCHIVE = YES` to include it. This can be done at multiple levels: You can set `GET_ACCEL_ARCHIVE = YES` as a ZPARM, as a special register, as a bind option for static SQL, or as a connection property for Java Database Connectivity (JDBC) or Open Database Connectivity (ODBC) connections.

Mind that the `CURRENT QUERY ACCELERATION` special register must also be set, or otherwise the moved data cannot be accessed.

Related concepts

Freeing up storage in Db2 for z/OS
The high-performance storage saver (HPSS) archives data of table partitions in Db2 for z/OS to an accelerator. In moving data that is no longer actively used (historical data) to a less expensive storage device, you can free up costly storage space on your database server.

Accelerator-only tables and in-database transformation

Accelerator-only tables are tables that do not originate from Db2 base tables. Their data exists only on an accelerator. Data in accelerator-only tables can only be modified by data-manipulation (DML) statements (INSERT, UPDATE, and DELETE). With accelerator-only tables, you can do in-database transformation (IDT) in a more efficient way.

Queries and data-manipulation statements against accelerator-only tables are always routed to an accelerator. If a SQL statement does not satisfy the conditions for acceleration, the statement fails. Furthermore, DML statements that are run on an accelerator can only manipulate data in accelerator-only tables.

Accelerator-only tables are useful for statistics and analytic tools that employ temporary data for reports because the high velocity of execution allows these tools to gather all required data quickly. Since the data in these tables can be modified fast, they are also ideal for data-preparation tasks that must be completed before the data can be used for predictive modeling.

Use accelerator-only tables if you want to accelerate queries that access previously calculated, interim result sets. To take full advantage of the product's high-speed capabilities, store all your interim results in accelerator-only tables. This allows subsequent queries or data transformations to process all relevant data on the accelerator.

To create an accelerator-only table, you submit a `CREATE TABLE` statement in Db2 that uses the `IN ACCELERATOR <acceleratorname>` clause. Accelerator-only tables are registered in the Db2 catalog with a type of `'D'`.

Accelerator-only tables cannot be loaded by the SYSPROC.ACCEL_LOAD_TABLES stored procedure or by the corresponding function in IBM Db2 Analytics Accelerator Studio. You have to use `INSERT` statements. These can be submitted from interactive client interfaces (SPUFI, IBM Data Studio), or from client application programs using ODBC, JDBC or static SQL. It is possible to transfer many rows in a single batch or transaction (multi-row insert). However, the multi-row insert support is limited to client application programs and cannot be used interactively.

Important:

The use of accelerator-only tables brings extra workloads to your accelerators that compete for the same resources, such as CPU, disk, and system memory. This might impact the stability of your production environment.

Conventional in-database transformation versus in-database transformation with accelerator-only tables

A conventional approach to in-database transformation might consist of the following steps:

1. `CREATE TABLE T1 …;`
2. `INSERT INTO T1 SELECT … FROM T0 …;`
3. `SELECT … FROM T1 …;`
4. `DROP TABLE T1;`

Step “1” on page 13 creates an interim table. Interim results that were calculated on the basis of the data in T0 are stored in table T1 (see step “2” on page 13). In step “3” on page 13, a second query reads the interim table T1 to calculate the final result. In step “4” on page 13, the interim table T1 is dropped because it is no longer needed.

To accelerate the steps above, create table T1 in step “1” on page 13 as an accelerator-only table. This accelerates the final (second) query and avoids the movement of data in Db2, which occupies considerable processing resources if T1 is large. If T1 is an accelerator-only table, the INSERT statement in step “2” on page 13 is run on the accelerator rather than in Db2. You also omit the duplication of table T1 in Db2 because you do not need to run IBM Db2 Analytics Accelerator stored procedures to define, load, and enable table T1 on the accelerator. This is a big advantage because many extract-transform-and-load (ETL) tools that generate data transformation statements automatically do not support stored procedure calls. Using accelerator-only tables, you have to modify the statements generated by these tools only slightly to end up with accelerated in-database transformation steps that do not require stored procedure calls.

Before running the following modified steps, you must define, load and enable acceleration for table T0 on the accelerator by using the IBM Db2 Analytics Accelerator stored procedures or the corresponding functions in IBM Db2 Analytics Accelerator Studio. This is just a one-time preparation step. Then you continue with the following steps:

1. `CREATE TABLE Q1.T1 … IN ACCELERATOR … IN DATABASE …;`
2. `INSERT INTO Q1.T1 SELECT … FROM T0 …;`
3. `SELECT … FROM Q1.T1 …;`
4. `DROP TABLE Q1.T1;`

Notes:

1. Depending on your situation, you might want to keep the accelerator-only table for later use and run step “4” on page 13 at a later time.
2. In contrast to temporary tables, accelerator-only tables are much more like normal Db2 tables. It is therefore recommended that you create these tables in an authorized database by using a `CREATE TABLE` statement with the `IN DATABASE` clause. Otherwise, accelerator-only tables are created in an implicit database, just like normal Db2 tables. You might also want to add an explicit schema qualifier to the table name.

The following examples consist of extended data transformations that use more than one interim table. They show how to change conventional Db2 statements to arrive at an accelerated solution to in-database-transformation. The left column shows the conventional approach. The right column shows the in-database-transformation steps that include accelerator-only tables.

Example 1

The Db2 tables T1, T2 and T3 already exist. The conventional approach uses two declared global temporary tables to store the interim results.

Table 2. Example 1: Declared global temporary tables versus accelerator-only tables

<table>
<tr><th></th><th>Conventional approach (Db2)</th><th>Accelerator-only tables</th></tr>
<tr><td>One-time preparation</td><td>N/A</td><td>Define and load the tables T1, T2 and T3 on accelerator ACCEL1 by using stored procedures or the corresponding functions in IBM Db2 Analytics Accelerator Studio.</td></tr>
<tr><td>Step 2</td><td><pre>
DECLARE GLOBAL TEMPORARY TABLE
   SESSION.TEMP1 (
   C11 CHAR(2), ...)
   ON COMMIT PRESERVE ROWS;

INSERT INTO SESSION.TEMP1
SELECT ... FROM T1 ...;

DECLARE GLOBAL TEMPORARY TABLE
   SESSION.TEMP2 (
   C21 CHAR(2), ...)
   ON COMMIT PRESERVE ROWS;

INSERT INTO SESSION.TEMP2
   SELECT ... FROM T2 ...;

SELECT ... FROM
   FROM SESSION.TEMP1 ...
        SESSION.TEMP2 ...
        T3 ...;

DROP TABLE SESSION.TEMP1;
DROP TABLE SESSION.TEMP2;
</pre></td><td><pre>
SET CURRENT QUERY ACCELERATION = ALL;

CREATE DATABASE DBTEMP;

CREATE TABLE TEMP1 (
   C11 CHAR(2), ...)
   IN ACCELERATOR ACCEL1
   IN DATABASE DBTEMP;

INSERT INTO TEMP1
   SELECT ... FROM T1 ...;

CREATE TABLE TEMP2 (
   C21 CHAR(2), ...)
   IN ACCELERATOR ACCEL1
   IN DATABASE DBTEMP;

INSERT INTO TEMP2
   SELECT ... FROM T2 ...;

SELECT ... FROM
   FROM TEMP1 ...
        TEMP2 ...
        T3 ...;

DROP TABLE TEMP1;
DROP TABLE TEMP2;
DROP DATABASE DBTEMP;
</pre></td></tr>
</table>

Example 2

The Db2 tables T1 and T2 already exist. The final transformation result is stored in the existing Db2 table T3. The conventional approach uses two declared global temporary tables to store the interim transformation results.

Table 3. Example 2: Declared global temporary tables versus accelerator-only tables

	Conventional approach (Db2)	Accelerator-only tables
One-time preparation	N/A	Define and load the tables T1 and T2 on accelerator ACCEL1 by using stored procedures or the corresponding functions in IBM Db2 Analytics Accelerator Studio.

Table 3. Example 2: Declared global temporary tables versus accelerator-only tables (continued)

	Conventional approach (Db2)	**Accelerator-only tables**

Step 2	DECLARE GLOBAL TEMPORARY TABLE
	   SESSION.TEMP1 (
	   C11 CHAR(2), ...)
	   ON COMMIT PRESERVE ROWS;
	
	INSERT INTO SESSION.TEMP1
	   SELECT ... FROM T1 ...;
	
	DECLARE GLOBAL TEMPORARY TABLE
	   SESSION.TEMP2 (
	   C21 CHAR(2), ...)
	   ON COMMIT PRESERVE ROWS;
	
	INSERT INTO SESSION.TEMP2
	   SELECT ...
	   FROM SESSION.TEMP1 ...;
	
	INSERT INTO T3
	   SELECT ... FROM
	   FROM SESSION.TEMP2 ...
	   T2 ...;
	
	DROP TABLE SESSION.TEMP1;
	DROP TABLE SESSION.TEMP2;
		SET CURRENT QUERY ACCELERATION = ALL;
		
		CREATE DATABASE DBTEMP;
		CREATE TABLE TEMP1 (
		   C11 CHAR(2), ...)
		   IN ACCELERATOR ACCEL1
		   IN DATABASE DBTEMP;
		
		INSERT INTO TEMP1
		SELECT ... FROM T1 ...;
		
		CREATE TABLE TEMP2 (
		   C21 CHAR(2), ...)
		   IN ACCELERATOR ACCEL1
		   IN DATABASE DBTEMP;
		
		INSERT INTO TEMP2
		   SELECT ... FROM TEMP1 ...;
		
		INSERT INTO T3
		   SELECT ... FROM
		   FROM TEMP1 ...
		   T2 ...;
		
		DROP TABLE TEMP1;
		DROP TABLE TEMP2;
		
		DROP DATABASE DBTEMP;

Notes:

- Bear in mind that alternative constructs, such as declared global temporary tables, which can exist in Db2, but not on an accelerator, cannot be accelerated by IBM Db2 Analytics Accelerator.
- It is not possible to create image copies of accelerator-only tables because these tables do not hold any data in Db2. You can only recover the data from their original sources, which is why accelerator-only tables should be used only for temporary data.
- An accelerator-only table can reside on just a single accelerator. Copies on other accelerators cannot exist (no high availability).
- For information on how to create an accelerator-only table using the `CREATE TABLE` command or the restrictions governing the use of these tables, follow the **Related information** link at the end of this topic.
- When you submit a `CREATE TABLE` statement to create an accelerator-only table, the statement is committed in Db2 for z/OS before the corresponding action is completed on the accelerator. That is, there is a brief period during which the table is listed in the Db2 catalog, but not in the accelerator catalog. If you run the SYSPROC.ACCEL_GET_TABLES_INFO stored procedure during that time, the integrity status `ViolatedOnAccelerator` is reported for that table in the result set. (In IBM Db2 Analytics Accelerator Studio, you see the error icon next to the table name in the **Tables** section of the **Accelerator** view. If you move the mouse pointer over the icon, the integrity status is displayed in a fly-over help window.)

Related concepts

Configuring Db2 for z/OS for the acceleration of dynamic SQL queries
To make Db2 for z/OS route dynamic SQL queries to an accelerator, you must enable acceleration by using the CURRENT QUERY ACCELERATION special register. Alternatively, you can set the QUERY_ACCELERATION ZPARM in Db2 for z/OS. This also applies to queries against accelerator-only tables.

Configuring Db2 for z/OS for the acceleration of static SQL queries

To accelerate static SQL queries, use the QUERYACCELERATION and GETACCELARCHIVE bind option for Db2 packages.

Related reference

Restrictions of accelerator-only tables
Here is a summary of the restrictions that exist for accelerator-only tables.

Impact of special register, ZPARM, or bind option settings on accelerator-only tables
SQL statements that reference accelerator-only tables can only be executed on an accelerator. However, certain combinations of special register and ZPARM settings might prohibit the execution of a statement on an accelerator. The same is true for the QUERYACCELERATION bind option for static SQL queries. The table in this section shows how these settings affect the statement execution.

Related information

CREATE TABLE statement

Workload balancing

Workload balancing can be described as follows: If more than one accelerator is connected to a Db2 subsystem, and the tables that are referenced in a query are defined on more than one of these accelerators, IBM Db2 Analytics Accelerator for z/OS automatically picks the accelerator with the lowest utilization for processing.

Product behavior

Workload balancing is possible between both types of version 7.1.0 accelerators: accelerators on an IBM Integrated Analytics System and Db2 Analytics Accelerator on Z accelerators. However, workload balancing does not occur if different product versions are used. For example, if a version 5.1.0 and a version 7.1.0 accelerator are attached to the same Db2 subsystem, IBM Db2 Analytics Accelerator will always favor the version 7.1.0 accelerator.

IBM Db2 Analytics Accelerator for z/OS system table entries

When you select tables for query acceleration, information about these tables is added to the IBM Db2 Analytics Accelerator for z/OS system tables in the Db2 for z/OS catalog of the connected subsystem. Each entry (system table row) serves as a link between Db2 and the table on the accelerator. An entry contains, among other information, the table name.

Chapter 3. Conditions for query routing to an accelerator

Queries can be routed to an accelerator only if certain conditions are met. There are also adverse conditions that prohibit query acceleration. Both types of conditions are discussed in this section.

A query can be routed to an accelerator if the following general conditions are met:

1. The accelerator is in an operational mode and has been started.
2. Db2 for z/OS has been configured so that the query will be routed to a connected accelerator provided that this query meets all other conditions.
3. The accelerator supports the SQL expressions that are used in the query.
4. All data that is referenced by the query, such as the contents of tables and views, is available on one and the same accelerator. To satisfy this condition, the accelerator must have been loaded with the data in question.

Related information

DB2 11 for z/OS: SYSIBM.DSN_PROFILE_ATTRIBUTES

Configuring Db2 for z/OS for the acceleration of dynamic SQL queries

To make Db2 for z/OS route dynamic SQL queries to an accelerator, you must enable acceleration by using the CURRENT QUERY ACCELERATION special register. Alternatively, you can set the QUERY_ACCELERATION ZPARM in Db2 for z/OS. This also applies to queries against accelerator-only tables.

The value of the QUERY_ACCELERATION ZPARM provides the default setting for the CURRENT QUERY ACCELERATION special register. Both, the ZPARM and the special register accept the following values:

1 (NONE)

No routing of dynamic SQL queries to an accelerator. Queries will be processed by Db2 for z/OS only (inhouse query processing).

2 (ENABLE)

A dynamic SQL query will be routed to an accelerator if it fulfills all required conditions. An incoming query is tested against a set of heuristics, which include the table size and a response time estimate based on cost information from the SYSIBM.DSN_PROFILE_ATTRIBUTES table. Both tests ensure that a query will only be routed to an accelerator if the query can be expected to run faster than in Db2 for z/OS. However, if an error occurs while the query is being processed by the accelerator, Db2 for z/OS will return a negative SQLCODE to the application and query processing will stop.

3 (ENABLE WITH FAILBACK)

Dynamic queries are accelerated only if Db2 for z/OS determines that it is advantageous to do so. If an accelerator returns an error during the PREPARE phase or when first opening (OPEN) the query, the query is processed by Db2 for z/OS rather than sent to the accelerator. If the accelerator returns an error during a FETCH operation or a subsequent OPEN operation, Db2 for z/OS returns an error to the user and the query ends abnormally.

4 (ELIGIBLE)

Dynamic queries are accelerated if they are eligible for acceleration. Db2 for z/OS does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are executed by Db2 for z/OS. If an accelerator fails while a query is running, or if the accelerator returns an error, Db2 for z/OS returns a negative SQL code to the application.

5 (ALL)

A dynamic query will always be routed to an accelerator, no matter if it fulfills the conditions or not. If processing cannot start or continue because an incoming query fails to fulfill all the conditions for accelerated query processing, Db2 for z/OS returns a negative SQLCODE to the application and query processing ends abruptly. That is, the query will not be processed at all.

A query against an accelerator-only table can only succeed if the CURRENT QUERY ACCELERATION special register is set to a value other than NONE. There is no difference to accelerator-shadow tables in this respect. However, this does not affect data-definition (DDL) statements, such as CREATE TABLE or DROP TABLE. DDL statements are executed regardless of the special register or ZPRAM setting.

To enable query routing to an accelerator on the Db2 side, you issue the following SQL statement:

```
SET CURRENT QUERY ACCELERATION = ENABLE
```

FAILBACK processing
If you specify `SET CURRENT QUERY ACCELERATION = ENABLE WITH FAILBACK`, the query is returned to Db2 for processing if an error occurs during the PREPARE phase. This is the time before the query is actually routed to the accelerator, and during which the heuristics are tested. A query is also returned if an error was caused by an accelerator failure, a network failure, a network connection timeout, or a similar error. Errors immediately following the initial OPEN request of a query usually fall into this category.

Restrictions:

FAILBACK processes are not started if one of the following conditions applies:

- Query results have already been returned. This holds true even if the results are not complete.
- Query routing to an accelerator fails although the same query could be routed successfully before.
- Queries do not qualify for routing to an accelerator. Such queries are prepared and executed in Db2, regardless of the setting of the `CURRENT QUERY ACCELERATION` special register.

Settings resulting from the preparation phase are stored in the Db2 Dynamic Statement Cache. This allows the query engine to skip the preparation phase when a query comes in for the second time. However, this is also often the reason for not returning a query to Db2 for z/OS in case of an error. If you assume that this has happened, invalidate the Dynamic Statement Cache and rerun the query.

Configuring Db2 for z/OS for the acceleration of static SQL queries

To accelerate static SQL queries, use the QUERYACCELERATION and GETACCELARCHIVE bind option for Db2 packages.

Important:

- The acceleration bind option QUERYACCELERATION applies to cursor queries and the SELECT portion of the SQL INSERT from SELECT statement. The SQL SELECT INTO statement can also be bound for acceleration.
- The functions and behavior of the acceleration bind option QUERYACCELERATION for static queries is the same as the comparable existing acceleration special register CURRENT QUERY ACCELERATION and CURRENT GET ACCEL ARCHIVE that is used for dynamic query acceleration, except that the Db2 decision to accelerate a static query and bind it for acceleration occurs at BIND or REBIND PACKAGE time, and not at query execution time when the application is run. The special registers do not affect the binding or the execution of static queries; instead, the new bind option is used to determine acceleration behavior for static queries.
- The accelerator need not be started at BIND or REBIND PACKAGE time for the static query to be bound for acceleration. However, you must start the accelerator before you run the application that executes the static query.

Table 4. Bind options for static SQL queries

QUERYACCELERATION	**GETACCELARCHIVE**
Binds static SQL queries for execution on an accelerator	Includes or excludes partition data that was archived by the High Performance Storage Saver (HPSS) from accelerated static SQL queries
Settings: **1 (NONE)** No static query in the application is bound for acceleration or will be accelerated when the application is run. **2 (ENABLE)** A static query is bound for acceleration if it satisfies the acceleration criteria, including the cost and heuristics criteria. The query is routed to an accelerator when the application is run. If the static query does not satisfy the acceleration criteria, the query is bound for an execution in DB2. In the following cases, DB2 returns a negative SQL code for the query to the application: • If a failure occurs while executing the accelerated static query on the accelerator • If the accelerator returns an error for the query • If the accelerator is not started and DB2 cannot route the static query to the accelerator for execution **3 (ENABLEWITHFAILBACK)** Same as ENABLE, except for the following cases: • The accelerator returns an error. • The accelerator fails on the first OPEN request for the accelerated static query. • The accelerator is not active or started when the first OPEN occurs. In either of these cases, DB2 does not return a negative SQL code (accelerator failure) to the application, but executes a "failback" operation. That is, a temporary statement-level incremental bind of the query is performed, and the query is run in DB2. No "failback" to DB2 is possible after a successful OPEN request.	Settings: **NO** No static query is bound to retrieve archived partition data from the accelerator. If the static query is not bound for acceleration, it will be be bound to run in DB2. If the static query is bound for acceleration because the bind option **QUERYACCELERATION** was specified, the query is routed to the accelerator when the application is run, but the query will only retrieve *active* data from the accelerator and ignore any HPSS data. **YES** If the bind option QUERYACCELERATION is also specified and the static query references a table from which data has been archived by the HPSS, one of the following courses of action is taken: • If the static query satisfies the acceleration criteria (as specified by QUERYACCELERATION), the query is bound for acceleration, and when the application is run, the query includes the archived data in the query. • If the static query cannot be bound for acceleration because it does not satisfy the criteria, DB2 fails to complete the BIND or REBIND PACKAGE operation and returns an error message for that query.

Table 4. Bind options for static SQL queries (continued)

QUERYACCELERATION	GETACCELARCHIVE
4 (ELIGIBLE) A static query is bound for acceleration if the query meets the basic acceleration criteria, regardless of cost or heuristics, and the query will be routed to the accelerator when the application is run. **5 (ALL)** All static queries in the application are to be bound for acceleration and routed to the accelerator when the application is run. If DB2 determines that a static query cannot be bound to run on the accelerator, and the query references a base table or view, DB2 fails to complete the BIND or REBIND PACKAGE operation with an error message for that query. No error message is returned for Declared Global Temp Tables (DGTTs) and Created Global Temp Tables (CGTTs) because these tables cannot be accelerated.	

The bind option is supported for:

- BIND, BIND COPY, and REBIND PACKAGE (local and remote)
- BIND DEPLOY (for SQLPL procedures)
- ALTER PROCEDURE and ALTER FUNCTION for native SQLPL procedures and SQLPL scalar functions
- CREATE PROCEDURE and CREATE FUNCTION for native SQLPL procedures and SQLPL scalar functions

The bind option is not supported for:

- ALTER TRIGGER
- CREATE TRIGGER
- REBIND TRIGGER PACKAGE

Important: The default value of the bind option is *option not specified*. The default is not taken from the comparable Db2 installation system parameter (ZPARMs) QUERY_ACCELERATION.

- The default value for both new bind options is *option not specified*. The default for the bind options is not taken from the comparable DB2 installation system parameters (ZPARMs) QUERY_ACCELERATION and GET_ACCEL_ARCHIVE.
- For INSERT, UPDATE, DELETE, and SELECT statements that address accelerator-only tables, the QUERYACCELERATION bind option must be set to ENABLE or ELIGIBLE. If you specify ALL, the operation fails and an SQLCODE is returned.
- If you submit a CREATE, DROP, or COMMIT statement against an accelerator-only table together with an INSERT, UPDATE, DELETE, or SELECT statement (against the same table) in the same thread, the table does not come into existence during the bind phase. However, the current implementation requires that all referenced tables are defined and enabled for query acceleration during the bind phase.

  As a workaround, you can use the VALIDATE(RUN) bind option, which binds the INSERT, UPDATE, DELETE, or SELECT statement incrementally at execution time. However, this usually does not give you the optimum performance. Therefore, consider running the static CREATE statement in a separate thread before running the INSERT, UPDATE, DELETE, or SELECT statement.

The bind option can also be used for the acceleration of dynamic queries. If specified, it sets the initial value in the same manner as the comparable special register CURRENT QUERY_ACCELERATION when the application is run, provided that the special register has not already been set explicitly before the invocation of that application. The value of the bind option is then used as the special register value for dynamic queries in that application, so that you do not have to add an explicit SET statement to the application for the acceleration special register.

Mind that you cannot use the bind options for dynamic queries under the following conditions:

- Dynamic and static queries are started from the same application or script.
- Both types of queries work on the same or share a number of tables.
- You want to accelerate the dynamic queries, but not the static queries.

In this case, you must use explicit SET statements to set the special registers in your application or script. This is because the bind options would also lead to an acceleration of the static queries. Special registers, on the other hand, have no effect on static queries.

1. Db2 installation system parameter QUERY_ACCELERATION and GET_ACCEL_ARCHIVE
2. Bind option value if specified (QUERYACCELERATION and GETACCELARCHIVE)
3. Explicit SET statement for the special register

As with the acceleration of dynamic queries and the use of the acceleration special register, the use of the bind option requires you to set ACCEL (Db2 ZPARM) to a value other than NONE to enable the Db2 subsystem for query routing to an accelerator.

Supported queries

A set of conditions must be fulfilled before an accelerator can handle a query.

The following conditions must apply:

- The principal expression used by the query is a SELECT statement or an “INSERT FROM SELECT” statement. In this case, only the SELECT part of the query is accelerated.

  Note: To enable support for “INSERT FROM SELECT” statements, set the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 2. For more information, see *Installing Db2 libraries with IBM Db2 Analytics Accelerator for z/OS support* in the *IBM Db2 Analytics Accelerator for z/OS: Installation Guide*.
- The query is defined as read-only. The criteria for read-only queries are documented in the subsection *Read-only cursors* of the topic *DECLARE CURSOR* in the Db2 10 for z/OS information center or the IBM Knowledge Center. A link is provided under **Related information** at the end of this topic.

  Tip: If a query is not considered a read-only query by Db2 for z/OS, but you want the query to run on an accelerator, you can add the `FOR READ ONLY` clause to the SQL of the query.
- If the query is submitted by an application and introduces a cursor definition by means of the DECLARE CURSOR statement, then this cursor must be a read-only cursor that is neither a scrollable cursor, nor a rowset cursor that is declared by a remote application, such as an SQL statement submitted from the Db2 CLI for Windows. However, read-only rowset cursors are supported if they are declared by local applications (such as SPUFI or local stored procedure calls) in the following contexts:
  - As part of a DELCARE CURSOR statement, for example:

    ```
    EXEC SQL DECLARE DYN1 CURSOR WITH ROWSET POSITIONING FOR STMT1;
    ```
  - As an attribute in a PREPARE statement, for example:

    ```
    STMTSTR = 'SELECT COLI2,COLI8,COLI4,COLI2'
                                       || 'FROM NTZTB01'
                                       || ' WHERE 1=1'
                                       || ' ORDER BY COLI2'
                                       || ' FOR FETCH ONLY';

    ATTRSTR = 'WITH ROWSET POSITIONING';
    ```

```
EXEC SQL PREPARE STMT1 ATTRIBUTES :ATTRSTR FROM :STMTSTR;
```

- In a rowset-positioned FETCH operation, for example:

```
DCL AINT2A(07) BIN FIXED(15);
DCL AINT2B(07) BIN FIXED(15);
DCL AINT4B(07) BIN FIXED(31);
DCL AINT8B(07) BIN FIXED(63);

EXEC SQL FETCH NEXT ROWSET  FROM  DYN1 FOR :NROWS ROWS INTO
  :AINT2A,:AINT8B, :AINT4B, :AINT2B;
```

- When #SET MULT_FETCH is used as an instruction for the DSNTEP4 program, for example:

```
//SYSIN    DD *
--#SET MULT_FETCH 7
SET CURRENT QUERY ACCELERATION = ALL;
SELECT COLI2, COLI8, COLI4, COLI2 FROM ADMF001.NTZTB01
WHERE 1=1 FOR FETCH ONLY;
```

- Plans for query processing must use packages rather than database request modules (DBRMs).
- Based on its own set of heuristics, Db2 classifies an acceleration of the query as favorable. That is, Db2 “expects” a much shorter query response time by routing the query to an accelerator.
- DRDA® is used for the transmission of the query from Db2 to the accelerator. Therefore, DRDA must be configured and started from any Db2 subsystem connected to the accelerator.

Related information

DECLARE CURSOR

Supported data types

See this overview of supported Db2 data types, corresponding dashDB® types, ranges, and hints at restrictions. The section also lists the unsupported data types at the end.

Numeric data types

Table 5. Numeric data types

Db2 type	dashDB type	Range Db2	Range dashDB	Comment
SMALLINT	SMALLINT (INT2)	-32768 to +32767	-32768 to +32767	
INTEGER	INTEGER (INT, INT4)	-2147483648 to +2147483647	-2147483648 to +2147483647	
BIGINT	BIGINT (INT8)	-9223372036854775808 to +9223372036854775807	-9223372036854775808 to +9223372036854775807	
REAL	REAL	~ -7.2 * 10^{+75} to -5.4 * 10^{-79} or ~ +5.4 * 10^{-79} to +7.2 * 10^{+75}	~ -3.4 * 10^{+38} to -1.18 * 10^{-38} or ~ 1.18 * 10^{-38} to +3.4 * 10^{+38}	For details, see “Differences in floating-point error tolerances” on page 57

Table 5. Numeric data types (continued)

Db2 type	dashDB type	Range Db2	Range dashDB	Comment
DECFLOAT	DECFLOAT	Precision of 34 or 16 digits (default is 34) 10^{-6143} to 10^{+6144} or 10^{-383} to 10^{+384}	Precision of 34 or 16 digits (default is 34) 10^{-6143} to 10^{+6144} or 10^{-383} to 10^{+384}	See Note under this table.
DOUBLE	DOUBLE	~ $-7.2 * 10^{+75}$ to $-5.4 * 10^{-79}$ or ~ $+5.4 * 10^{-79}$ to $+7.2 * 10^{+75}$	~ $-1.79 * 10^{+308}$ to $-3.40 * 10^{-308}$ or ~ $+3.4 * 10^{-308}$ to $+1.79 * 10^{+308}$	For details, see "Differences in floating-point error tolerances" on page 57
DECIMAL(*precision, scale*)	NUMERIC(*precision, scale*)	Max. precision 31	Max. precision 31	

Note on DECFLOAT: During tests, the extreme values at either end of the range could not be loaded into accelerator-only tables, which hints at a general problem with very low and very high values in this context. However, there was no problem loading the extreme values into regular accelerator-shadow tables-

See also the entry for the ROUND function in Restrictions.

.

Date, time, and timestamp data types

Table 6. Date, time, and timestamp data types

Db2 type	dashDB type	Range Db2	Range dashDB	Comment
DATE	DATE	0001-01-01 to 9999-12-31	0001-01-01 to 9999-12-31	The use of DATE = LOCAL is not supported.
TIME	TIME	00:00:00.000000000000 to 24:00:00.000000000000	00:00:00.000000000000 to 24:00:00.000000000000	The use of TIME = LOCAL is not supported. The dashDB database on the accelerator supports the same precision as Db2 for z/OS (up to 12 fractional digits). A truncation of values is therefore not necessary.

Table 6. Date, time, and timestamp data types (continued)

Db2 type	dashDB type	Range Db2	Range dashDB	Comment
TIMESTAMP	TIMESTAMP	**Date portion:** 0001-01-01 to 9999-12-31 **Time portion:** 00:00:00.000000000000 to 24:00:00.000000000000	**Date portion:** 0001-01-01 to 9999-12-31 **Time portion:** 00:00:00.000000000000 to 24:00:00.000000000000	Some timestamp formats are supported by Db2 for z/OS, but not by dashDB: • `2004-02-15T11.34.56` • `2004-02-15 11.34.56` **Important:** If values are formatted in any of the formats shown above and stored as characters, Db2 for z/OS cannot prevent these values from being sent to an accelerator. Errors will be the result. Timestamps with timezone specifications are not supported, to the effect that: • SQL queries that require TIMESTAMP with a timezone specification as the result data type are not routed to an accelerator. • Timestamps with timezone constants are not accepted. • Functions and special registers that are related to timestamps with timezone specifications are not accepted.

String data types

Only the following combinations of coded character set identifiers (CCSIDs) and subtypes are allowed in string expressions:

Table 7. Supported combinations of CCSIDs and subtypes

CCSID	Allowed character subtypes	Comment
ASCII (for example 819; there are numerous others)	SBCS	An ASCII table cannot be added to an accelerator if it is part of a Db2 subsystem with a setting of `mixed=yes`. This holds true even if the table does not contain any multi-byte (or mixed) data.
EBCDIC (for example 37; there are numerous others)	SBCS MIXED (partially supported) DBCS (graphic, partially supported)	
UNICODE (367, 1208, 1200)	SBCS (367) MIXED (1208) DBCS (1200)	

Table 8. String data types

Db2 type	**dashDB type**	**Supported CCSIDs**	**Comment**
BINARY	BINARY	N/A	
VARBINARY	VARBINARY	N/A	
CHAR FOR SBCS DATA	CHAR	ASCII EBCDIC UNICODE (367)	
VARCHAR FOR SBCS DATA	VARCHAR	ASCII EBCDIC UNICODE (367)	The maximum length in Db2 for z/OS is 32704. In dashDB, it is 32592. A table load fails and message `AQT20052E` is displayed if a string exceeds the dashDB limit.
CHAR FOR MIXED DATA	CHAR	UNICODE (1208) EBCDIC (partially supported)	
VARCHAR FOR MIXED DATA	VARCHAR	UNICODE (1208) EBCDIC (partially supported)	
CHAR FOR BIT DATA	CHAR FOR BIT DATA	ASCII (single-byte only) EBCDIC UNICODE (1208)	
VARCHAR FOR BIT DATA	VARCHAR FOR BIT DATA	ASCII (single-byte only) EBCDIC UNICODE (1208)	
GRAPHIC	GRAPHIC	UNICODE (1200) EBCDIC	This type is fully supported for EBCDIC and UNICODE tables. **Important:** If your Db2 subsystem is set to `MIXED=NO`, and you use the GRAPHIC() function on a single-byte (SBCS) CHAR column, the query will fail when it is routed to an accelerator.
VARGRAPHIC	VARGRAPHIC	UNICODE (1200) EBCDIC	This type is fully supported for EBCDIC and UNICODE tables. The maximum length in Db2 for z/OS is 16352. In dashDB, it is 16296. An accelerator table load operation aborts when a string exceeds the dashDB limit. **Important:** If your Db2 subsystem is set to `MIXED=NO`, and you use the GRAPHIC() function on a single-byte (SBCS) VARCHAR column, the query will fail when it is routed to an accelerator.

ROWID data type

The ROWID data type is to uniquely and permanently identify rows in a table. A ROWID value can be used to access a table row directly. The ROWID data type is supported by IBM Db2 Analytics Accelerator. However, consider the following limitation:

Two formats of ROWIDs exist: an *external* format and an *internal* format. The external format consists of the internal format plus several hexadecimal strings. Db2 for z/OS can handle both formats, but stores just the internal format. Since the internal format is stored, this is also what is loaded into the corresponding accelerator-shadow tables.

ROWID columns can be used like any other column, as, for example, in queries like:

```
SELECT <RWID> FROM <table>;
```

, where `<RWID>` is supposed to be a column containing ROWIDs. In cases like this, Db2 for z/OS returns the longer, external format. The query would work in the same way if it were accelerated, but the accelerator would return the internal format because it cannot translate or otherwise infer the external format from the internal values.

In addition, the values in ROWID columns can be used in WHERE predicates for direct row access, as in queries like:

```
SELECT * FROM <table> WHERE RWID = ROWID(X'...')
```

Inside the parentheses of the `ROWID(X'...')` expression, you can use both, the external or the internal format, if the query is processed by Db2 for z/OS.

However, the same query can be accelerated only if the internal format is used, for the same reason as in the previous example: The accelerator "understands" just the internal format.

Notes:

- For some complex queries (including OUTER JOIN expressions and views, for example), Db2 for z/OS requires the external ROWID format. Such queries cannot be accelerated.
- ROWID expressions can also be used in `INSERT INTO … SELECT FROM` statements, where just the SELECT portion of the query is accelerated.
- You cannot add columns with the ROWID data type to accelerator-only tables. If you try to include a column definition of that sort in a `CREATE TABLE … IN ACCELERATOR` statement, Db2 for z/OS returns SQL Code `-4747`.
- To use ROWID expressions in accelerated static SQL queries, first rebind the static application or package with the Db2 bind option QUERYACCELERATION. Use a value other than NONE. For more details, see *Configuring Db2 for z/OS for the acceleration of static SQL queries* (link a the end of this topic) and the Db2 for z/OS documentation.

Unsupported Db2 data types

- BLOB
- CLOB
- DBCLOB
- XML

Related concepts

Conditions that prevent query routing to an accelerator
Db2 for z/OS does not route a query to an accelerator if any of the following conditions applies.

Configuring Db2 for z/OS for the acceleration of static SQL queries

To accelerate static SQL queries, use the QUERYACCELERATION and GETACCELARCHIVE bind option for Db2 packages.

Supported SQL functions and expressions

See which aggregate functions and scalar functions are supported by IBM Db2 Analytics Accelerator.

Supported aggregate functions

IBM Db2 Analytics Accelerator supports all aggregate functions, except for the XMLAGG function.

The following aggregate functions are supported even if they are used in combination with the DISTINCT or UNIQUE keyword:

- STDDEV
- STDDEV_SAMP
- VARIANCE
- VARIANCE_SAMP

Note: Restrictions exist if you use the DISTINCT or UNIQUE keyword. See Table 10 on page 29.

Supported scalar functions

In addition, IBM Db2 Analytics Accelerator supports a variety of scalar functions and the CAST specification.

If restrictions exist for a function in Table 9 on page 27, the function is marked with an asterisk(*). The restrictions are described in Table 10 on page 29.

Table 9. Supported scalar functions and specifications

Function name	Function name	Function name	Function name
• ABS	• DEGREES	• MOD	• SUBSTR
• ADD_MONTHS	• DIGITS	• MONTH	• SUBSTRING (see Table 10 on page 29)
• BINARY	• DOUBLE_PRECISION (also DOUBLE, FLOAT)	• MONTHS_BETWEEN	• TIME (see Table 10 on page 29)
• BIGINT	• EXP	• NEXT_DAY	• TIMESTAMP (see Table 10 on page 29)
• BITAND (see Table 10 on page 29)	• EXTRACT (see Table 10 on page 29)	• NEXT_WEEK	• TIMESTAMPDIFF
• BITANDNOT	• FLOAT	• NEXT_MONTH	• TIMESTAMP_FORMAT (also TO_DATE; see Table 10 on page 29)
• BITNOT	• FLOOR	• NEXT_QUARTER	• TIMESTAMP_ISO (see Table 10 on page 29)
• BITOR	• GRAPHIC (see Table 10 on page 29)	• NORMALIZE_DECFLOAT	• TO_NUMBER (DECFLOAT_ FORMAT)
• BITXOR	• HOUR (see Table 10 on page 29)	• NULLIF	• TRANSLATE (see Table 10 on page 29)

Table 9. Supported scalar functions and specifications (continued)

Function name	Function name	Function name	Function name
• CEILING	• HEX (see Table 10 on page 29)	• NVL	• TRIM (also with two arguments; (see Table 10 on page 29)
• CAST specification (see Table 10 on page 29)	• IFNULL	• POSITION	• TRUNCATE(see Table 10 on page 29)
• CHAR (see Table 10 on page 29)	• DOUBLE_PRECISION (also DOUBLE, FLOAT)	• POSSTR	• UPPER (UCASE) (see Table 10 on page 29)
• CHARACTER_LENGTH	• INTEGER (INT)	• POWER® (see Table 10 on page 29)	• VALUE
• COALESCE	• JULIAN_DAY (see Table 10 on page 29)	• QUARTER (see Table 10 on page 29)	• VARBINARY
• COMPARE_ DECFLOAT (see Table 10 on page 29)	• LAST_DAY	• RADIANS	• VARCHAR
• CONCAT (see Table 10 on page 29)	• LEFT (see Table 10 on page 29)	• RAND (see Table 10 on page 29)	• VARCHAR9 (see Table 10 on page 29)
• CORR	• LENGTH	• REAL	• VARCHAR_FORMAT (TO_CHAR; see Table 10 on page 29)
• COVAR	• LN	• REPEAT	• VARGRAPHIC (see Table 10 on page 29)
• COVAR_SAMP	• LOCATE (see Table 10 on page 29)	• REPLACE	• WEEK (see Table 10 on page 29)
• DATE (see Table 10 on page 29)	• LOCATE_IN_STRING (see Table 10 on page 29)	• RIGHT (see Table 10 on page 29)	• WEEK_ISO (see Table 10 on page 29)
• DAY	• LOG10	• ROUND (see Table 10 on page 29)	• YEAR
• DAYOFMONTH	• LOG	• ROWID (see Table 10 on page 29)	
• DAYOFWEEK	• LOWER (LCASE) (see Table 10 on page 29)	• RPAD	
• DAYOFWEEK_ISO (see Table 10 on page 29)	• LPAD (see Table 10 on page 29)	• RTRIM (also with two arguments)	
• DAYOFYEAR (see Table 10 on page 29)	• LTRIM (also with two arguments)	• SECOND (see Table 10 on page 29)	
• DAYS	• MAX (GREATEST)	• SIGN 4see Table 10 on page 29)	

Table 9. Supported scalar functions and specifications (continued)

Function name	Function name	Function name	Function name
• DECFLOAT (see Table 10 on page 29)	• MEDIAN	• SMALLINT	
• DECFLOAT_FORMAT (TO_NUMBER)	• MICROSECOND	• SPACE	
• DECIMAL	• MIDNIGHT_SECONDS (see Table 10 on page 29)	• SQRT	
• DECODE (also with two arguments)	• MIN (also LEAST)	• STRIP (see Table 10 on page 29)	
	• MINUTE (see Table 10 on page 29)	• SMALLINT	

Table 10. Restrictions

Function, keyword, or expression	Restriction
• BITAND	The lowest possible BIGINT value (-9223372036854775808) at any argument position returns a result data type of DECFLOAT(34) rather than BIGINT, as in Db2 for z/OS). For any other valid input value, the result data type is BIGINT.
• CAST specification	Restrictions exist if you want to use CAST expressions to convert string data types and, by specifying the string unit, determine the number of bytes to be used for the representation of characters in the resulting strings: **CAST TO CHAR** If the string unit is CODEUNITS16 or CODEUNITS32, IBM Db2 Analytics Accelerator might return a different number of trailing blanks than Db2 for z/OS. **CAST TO GRAPHIC** Allowed string units are CODEUNITS16 and CODEUNITS32. When compared with the output from Db2 for z/OS, IBM Db2 Analytics Accelerator might return a different number of trailing blanks. **CAST TO VARGRAPHIC** Allowed string units are CODEUNITS16 and CODEUNITS32. See also “Strings as time values and their conversion” on page 49.

Table 10. Restrictions (continued)

Function, keyword, or expression	Restriction

• CHAR	• If the string unit is CODEUNITS16 or CODEUNITS32, IBM Db2 Analytics Accelerator might return a different number of trailing blanks than Db2 for z/OS.
	• If a string of space characters is requested in a query, the CHAR function always returns strings with a length of 255, no matter which string length is specified in the query. Suppose, for example, that a table with the name TMP contains strings consisting of (subsequent) space characters. If you run a query like
	`SELECT CHAR(SPACE(20)) FROM TMP;`
	you will find blank strings with a length of 255 rather than a length of 20 in the results.
	• If the function is used to retrieve a substring of a multibyte string, so that it will split a multibyte-character, Db2 for z/OS returns the correct byte-code of the split character, whereas IBM Db2 Analytics Accelerator replaces this portion of the byte-code with blanks. You can avoid this by using the *length* argument (OCTETS, CODEUNITS16, or CODEUNITS32).
	Suppose, for example, that you run the following query to retrieve the first five bytes of the string ＭＡＭＡ, consisting of full-width Latin capital letters. Each character is represented by three bytes. The mode of representation is UTF-8.
	`SELECT HEX(CHAR('ＭＡＭＡ',5));`
	To illustrate the differences in the outputs, the HEX function is used:
	Result in Db2 for z/OS:
	EFBCAD**EFBC**
	This value represents the first letter 'M' (in full) and the first two bytes of the first letter 'A'.
	Result returned by accelerator:
	EFBCAD**2020**
	This value represents the first letter 'M' (in full) and two blanks instead of the first two bytes of the letter 'A'.
	You could specify OCTETS as the length argument to avoid this problem:
	`SELECT HEX(CHAR('ＭＡＭＡ',5,OCTETS));`
	The result would then be EFBCAD**EFBC** in both cases.

• CONCAT	If the result data type is VARCHAR, the maximum length of a result string is limited to 4000 characters.

Table 10. Restrictions (continued)

Function, keyword, or expression	**Restriction**
- DATE	- Db2 accepts an argument in the format `yyyynnn` as input for the DATE function. IBM Db2 Analytics Accelerator does not accept this format and therefore issues an error. - If the DATE function is used on numeric expressions, it is converted to `DATE(to_date('0000-12-31','YYYY-MM-DD') +trunc(numeric_exp))` during query routing to an accelerator. If the `numeric_exp` result is a negative number, IBM Db2 Analytics Accelerator adds BC to the date. An error is returned when the converted scalar function is evaluated in Db2.
- DAYOFWEEK_ISO - DAYOFYEAR - JULIAN_DAY - MIDNIGHT_SECONDS - QUARTER - TIMESTAMP_ISO - WEEK - WEEK_ISO	- If Db2 for z/OS is configured to use a local DATE format (`DATE=LOCAL` on DSNTIP4 panel) and a query uses a DATE constant as an argument for any of these functions, the query is not accelerated, and `SQLCODE -4742` is returned. - Queries containing the DAYOFWEEK_ISO, DAYOFYEAR, JULIAN_DAY, QUARTER, TIMESTAMP_ISO, or WEEK_ISO function fail with `SQLCODE=-443` under the following conditions: 1. The query includes the clause `FETCH FIRST n ROWS`. 2. Any of the functions mentioned before uses string representations of U.S. or European date formats as arguments. The same applies to queries containing MIDNIGHT_SECONDS, except that string representations of European date formats are allowed. - Queries containing the TIMESTAMP_ISO function fail during the attempt to route them to an accelerator if string representations of TIME values are used as arguments.

- COMPARE_DECFLOAT	Since the database engine on the accelerator does not support DECFLOAT constants (see entry for DECFLOAT), the values to be compared are interpreted as floating-point numbers (DOUBLE), which results in a loss of precision for big numbers. This loss of precision might cause a value shift that leads to deviating comparison results. Consider the following cases:
	`COMPARE_DECFLOAT(0.123456789012345678901E-187,`
	`                 1.234567890123456789010E-188),`
	`COMPARE_DECFLOAT(0.123456789012345678901E-185,`
	`                 1.234567890123456789010E-188),`
	`COMPARE_DECFLOAT(0.123456789012345678901E-187,`
	`                 1.234567890123456789010E-185)`
	The output of these comparisons is different, depending on where they are executed:
	Db2 for z/OS:
	`'2','2','1'`
	IBM Db2 Analytics Accelerator:
	`'0','2','1'`

Table 10. Restrictions (continued)

Function, keyword, or expression	**Restriction**
• DECFLOAT	• The database engine on the accelerator does not check, and therefore does not necessarily preserve the scale of the original Db2 for z/OS values. Db2 for z/OS defines the scale as the number of digits to the right of the decimal point. For example, 0.03E3 might be returned as 0.3E2. As long as the value is numerically accurate according to the IEEE754 specification, it is considered to be correct. • Different representations of the same numerical value (for example 1E2, 100, and 10E1) also have slightly different representations in memory and the database engine on the accelerator rarely normalizes these values. If the database is a row store, the values are not normalized at all. If it is a columnar store, normalization might occur more often, especially if a GROUP BY operation is performed on the values. However, you can never rely on the normalization of differently represented values. • The database engine on the accelerator does not support DECFLOAT constants. If the accelerator receives a DECFLOAT constant from Db2 for z/OS, and the value of this constant is within the range of DOUBLE values accepted by the database engine, the value is interpreted as DOUBLE, and the accelerator might return a value with a different precision. If a received value is not within the range of DOUBLE, SQL code `-405` or `-410` is returned. For example: – `SQLCODE = -405, ERROR:  THE NUMERIC CONSTANT 1.0E-308 CANNOT BE USED AS SPECIFIED BECAUSE IT IS OUT OF RANGE ...` – `SQLCODE = -410, ERROR:  A NUMERIC VALUE 1234.5678901234567890123456E-55 IS TOO LONG, OR IT HAS A VALUE THAT IS NOT WITHIN THE RANGE OF ITS DATA TYPE ...` • Db2 for z/OS and the database engine of IBM Db2 Analytics Accelerator compute different DECFLOAT values if one of the operands is Infinity (∞) and none of the other operands are of the type *not a number* (NaN). Signaling (sNaN) values are also counted as NaN. In the majority of these cases, Db2 for z/OS returns `NaN` as the result, whereas IBM Db2 Analytics Accelerator returns positive or negative Infinity.

Table 10. Restrictions (continued)

Function, keyword, or expression	Restriction
• DISTINCT • UNIQUE	If the DISTINCT or UNIQUE keyword is applied to string values with trailing blanks, IBM Db2 Analytics Accelerator might cut off the trailing blanks from the resulting values. Example: `SELECT DISTINCT N_S_VCHR CONCAT ' ' FROM MATRIX.MBCS_EBCDIC;` The SQL code specifies that a space character or blank is to be added at the end of each value retrieved. Now see some of the rows returned by this query, once by Db2 for z/OS, and once by IBM Db2 Analytics Accelerator: **Db2 for z/OS:** 1. `[    ]` (four blanks) 2. `[  HIJ  ]` (two blanks at end of string) 3. `[!"$%¢&\/ ]` (one blank at end of string) **IBM Db2 Analytics Accelerator:** 1. `[]` (empty string) 2. `[  HIJ]` (no blanks at end of string) 3. `[!"$%¢&\/]` (no blanks at end of string)
• GRAPHIC	If the string unit is CODEUNITS16 or CODEUNITS32, IBM Db2 Analytics Accelerator might return a different number of trailing blanks than Db2 for z/OS.
• HEX	This function returns a CHAR value in Db2 for z/OS. If the query is accelerated by IBM Db2 Analytics Accelerator, the function returns a VARCHAR value.
• LEFT • LPAD • RIGHT	Result deviations can occur if input strings contain multibyte or graphic EBCDIC characters.
• LOCATE • LOCATE_IN_STRING	Result deviations can occur under the following conditions: • Input strings contain multibyte or graphic characters and the search begins in the middle of the string. • A string unit (OCTETS, CODEUNITS16, or CODEUNITS32) is not specified.

Table 10. Restrictions (continued)

Function, keyword, or expression	Restriction

• LCASE
• LOWER
• UCASE
• UPPER
	• If a string argument contains characters with diacritical marks, Db2 for z/OS does not deliver the expected result, but IBM Db2 Analytics Accelerator does. For example:
	`CREATE TABLE XYZ.TMP (A GRAPHIC(64), B GRAPHIC(64) NOT NULL)`
	`CCSID UNICODE;`
	`INSERT INTO XYZ.TMP VALUES ('Ü','Ü');`
	`SELECT LOWER(B) XYZ.TMP;`
	Db2 for z/OS:
	Returns Ü, same as before because characters with diacritical marks are not converted.
	IBM Db2 Analytics Accelerator:
	Returns ü (correct result). Reason: The string argument is converted before the function is executed.
	• Accelerated queries might return different results (including invalid values) or an error with `SQLCODE=-171`, `SQLSTATE=42815` if any of these functions is applied to EBCDIC or ASCII data. For example, take the following query, which returns invalid values:
	`SELECT UCASE('µ~stuvwxyz', 'De_CH') FROM SINGLE_EBCDIC;`
	The query is supposed to find the string `'µ~stuvwxyz'` in a table containing single-byte EBCDIC data and convert the resulting strings to uppercase. The Swiss variant of the German character set is used.
	Depending on where the query is run, the following values are returned:
	Db2 for z/OS:
	`µ˜STUVWXYZ` (correct result)
	Accelerator:
	`<unprintable symbol>˜STUVWXYZ` (incorrect result)

Table 10. Restrictions (continued)

Function, keyword, or expression	**Restriction**

• LCASE
• LOWER
• UCASE
• UPPER
(continued)
	• Result deviations can occur if input strings contain multibyte or graphic EBCDIC characters.
	• Db2 for z/OS and IBM Db2 Analytics Accelerator return different length values if any of these functions is applied to GRAPHIC data. For example:
	`CREATE TABLE TMP (A GRAPHIC(64),`
	`B GRAPHIC(64) NOT NULL)`
	`CCSID UNICODE;`
	`INSERT INTO TMP VALUES ('a','b');`
	After that, you run the following query:
	`SELECT LENGTH(UPPER(B)) FROM TMP;`
	Depending on where the query is run, the following values are returned:
	Db2 for z/OS:
	64
	Accelerator:
	192
	The reason is that the accelerator uses a different result data type, that is, CHAR(192) instead of GRAPHIC(64).

• POWER	If the argument data type is BIGINT, the result data type will be cast to DOUBLE.
• RAND	Supported without a parameter only.
• ROWID	In Db2 for z/OS, two types of ROWID values exist: the longer, external type, and the shorter internal type. If a query references ROWID values and these values are in the external format, the values are converted to the internal format when the query is routed to an accelerator because IBM Db2 Analytics Accelerator can handle the internal type only. The internal ROWID type is also what the accelerator returns in the query results. If you have an application that relies on the external ROWID type to be returned, do not accelerate the query.
• ROUND	If ROUND works on a numerical expression of type DECFLOAT, and the rounding would result in a range underflow or overflow, Db2 for z/OS returns positive or negative Infinity (-∞ or +∞), whereas IBM Db2 Analytics Accelerator returns *not a number* (NaN).
• SIGN	If the data type of an input argument is DECIMAL, the result must be cast to this type.
• STRIP	Result deviations can occur if input strings (in fields or columns) contain graphic EBCDIC characters.
• SUBSTRING	SUBSTRING is supported only if the expression refers to character or graphic data types.

Table 10. Restrictions (continued)

Function, keyword, or expression	**Restriction**
• TIMESTAMP • TIMESTAMP_FORMAT • TO_DATE	Some string literals with a dash (-) or space character ( ) as the separator (delimiter) between the date and the time portion and a period (.) or colon (:) to separate hours, minutes, and seconds are not supported or only partially supported as function arguments: **Examples:** • `yyyy-mm-dd[ hh.mm.ss[.nnnnnn]]` (space character before hh; period to separate hours, minutes, and seconds). This format is not supported. • `yyyy-mm-dd[-hh:mm:ss[.nnnnnn]]` (dash before hh; colon to separate hours, minutes, and seconds). This format is supported by the current version of IBM Db2 Analytics Accelerator, but only in connection with Db2 12 for z/OS or later. See also "Dates in TIMESTAMP expressions" on page 49.
• TRANSLATE	• If the arguments are of data type CHAR or VARCHAR, the corresponding characters in the to-string expression and the from-string expression must have the same number of bytes (except in the case of a zero-length string). This means, for example, that you cannot convert a single-byte character to a multibyte character, or a multibyte character to a single-byte character. The padding character (pad-char-exp argument) cannot be the first byte of a valid multibyte character. If it is, error `-171` is returned. • Result deviations can occur if arguments contain multibyte characters.
• TRUNCATE	You cannot use the TRUNCATE function on accelerator-only tables.
• TRIM	Result deviations can occur if input strings contain multibyte or graphic EBCDIC characters.
• UPPER • UCASE	See LOWER/LCASE.

Table 10. Restrictions (continued)

Function, keyword, or expression	**Restriction**

• VARCHAR9	Result deviations can occur if the VARCHAR9 function is used to convert DECIMAL values to strings:
	• When compared with the results returned by Db2 for z/OS, an accelerated query might show a leading blank in the results. For example:
	`SELECT VARCHAR9(13.1) FROM TABLE_1`
	gives you the following results:
	Db2 for z/OS:
	`13.1`
	IBM Db2 Analytics Accelerator
	`13.1` (leading blank)
	In accordance with these results, the length function also returns different values:
	`SELECT LENGTH(VARCHAR9(13.1)) FROM TABLE_1`
	Db2 for z/OS:
	4
	IBM Db2 Analytics Accelerator:
	5
	• When compared with the results returned by Db2 for z/OS, an accelerated query might show leading zeros (0s) in places where the result value does not require all of the formatted digits. For example, if `1.0` is one of the values returned by the following query:
	`SELECT VARCHAR9(DECIMAL(col1,7,5)) FROM TABLE_1`
	you get the following results:
	Db2 for z/OS:
	`1.00000`
	IBM Db2 Analytics Accelerator
	`01.00000` (leading zero)

Table 10. Restrictions (continued)

Function, keyword, or expression	Restriction
• VARCHAR_FORMAT	The following date and timestamp formats are not supported: • `FF[n]` - fractional seconds (000000-999999) The optional number *n* is used to specify the number of digits to include in the return value. Valid values for *n* are the integers from 1-12. The default is 6. • `ID` - ISO day of the week (1-7) The value 1 represents Monday. The value 7 represents Sunday. • `IYYY` - ISO year (0000-9999) The last four digits of the year based on the ISO week that is returned. • `NNNNNN` - microseconds (000000-999999) This format is equivalent to FF6. • RR - last two digits of the year (00-99) • SSSSS - seconds since the previous midnight

Unsupported scalar functions

Table 11. Unsupported scalar functions

• ACOS	• CCSID_ENCODING	• DIFFERENCE	• MULTIPLY_ALT	• SCORE	• TOTALORDER
• ASCII	• CLOB	• EBCDIC_CHR	• MQ <xxx>	• SIN	• TRUNC_TIMESTAMP
• ASCII_CHR	• COLLATION_KEY	• EBCDIC_STR	• NORMALIZE_STRING	• SINH	• UNICODE
• ASCII_STR	• COS	• ENCRYPT_TDES	• OVERLAY	• SOA <xxx>	• UNICODE_STR
• ASIN	• COSH	• GENERATE_UNIQUE	• QUANTIZE	• SOUNDEX	• VERIFY_GROUP_ FOR_USER
• ATAN	• CONTAINS	• GETHINT	• PACK	• TAN	• VERIFY_ROLE_FOR_USER
• ATAN2	• DBCLOB	• GETVARIABLE	• RAISE_ERROR	• TANH	• VERIFY_TRUSTED_CONTENT_ FOR_USER
• ATANH	• DECFLOAT_SORTKEY	• IDENTITY_VAL_ LOCVAL	• RID	• TIMESTAMPADD	• <xxx>XML<xxx>
• BLOB	• DECRYPT <xxx>	• INSERT	• ROUND_TIMESTAMP	• TIMESTAMP_TZ	

Supported functions of the database engine

The product's database engine offers a range of aggregate functions and online analytical processing (OLAP) functions. The latter are frequently called built-in functions. A subset of these functions is supported. That is, you can use these functions in your SQL queries, and they will be processed if the query can be accelerated. A list of the supported functions is provided here.

General information

- To use any of these functions, the QUERY_ACCELERATION special register must be set to ALL, ENABLE, or ELIGIBLE.
- All functions listed here can be referenced in a database view.
- None of the functions can be specified as the source function of a user-defined function (UDF). Trying to do so results in `SQLCODE -120`.
- Db2 for z/OS verifies just the validity of the parameter data types. All other processing, including the validation, is done by the database engine of the accelerator.
- Some of the functions listed here are available as OLAP functions and as aggregate functions, but most of them are OLAP only. If a function is also available as an aggregate function, you find a link to an external description. All OLAP functions are externally described in a single topic. A link is provided at the bottom.

Note: The following list makes frequent use of the term *OLAP window*. An online analytical processing window (OLAP window) can be seen as the scope or range of values preceding or succeeding the current value. The selected window serves as the basis for the calculation.

CUME_DIST

This is a distribution function that returns a cumulative distribution of a row within an OLAP window, expressed as a value between 0.0 and 1.0.

Result computation

The number of rows preceding or at the same level as the current row in the OLAP window is divided by the total number of rows in the OLAP window. The result cannot be NULL.

Result data type

DECFLOAT(34)

More information

CUME_DIST aggregate function

FIRST_VALUE

This function returns the expression value for the first row in an OLAP window.

Result computation

If `IGNORE NULLS` is specified, all rows where the expression value for the row is the null value are not considered in the calculation. If `IGNORE NULLS` is specified and all values in the OLAP window are null, FIRST_VALUE returns the null value.

LAG

This function returns the expression value for the row at *offset* rows *before* the current row. The offset must be a positive integer constant.

Result computation

An offset value of 0 means the current row. If a *window-partition-clause* is specified, *offset* means offset rows before the current row and within the current partition. If *offset* is not specified, the value 1 is used.

If *default-value* (which can be an expression) is specified, it will be returned if the offset goes beyond the scope of the current partition. Otherwise, the null value is returned.

If `IGNORE NULLS` is specified, all rows where the expression value for the row is the null value are not considered in the calculation. If `IGNORE NULLS` is specified and all rows are null, *default-value* is returned. If *default-value* was not specified, the null value is returned.

LAST_VALUE

This function returns the expression value for the last row in an OLAP window.

Result computation

If IGNORE NULLS is specified, all rows where the expression value for the row is the null value are not considered in the calculation. If IGNORE NULLS is specified and all values in the OLAP window are null, FIRST_VALUE returns the null value.

LEAD

This function returns the expression value for the row at *offset* rows *after* the current row.

Result computation

An offset value of 0 means the current row. If a *window-partition-clause* is specified, *offset* means offset rows before the current row and within the current partition. If *offset* is not specified, the value 1 is used.

If *default-value* (which can be an expression) is specified, it will be returned if the offset goes beyond the scope of the current partition. Otherwise, the null value is returned.

If IGNORE NULLS is specified, all rows where the expression value for the row is the null value are not considered in the calculation. If IGNORE NULLS is specified and all rows are null, *default-value* is returned. If *default-value* was not specified, the null value is returned.

NTH_VALUE

This function returns the expression value for the *n*throw in an OLAP window.

Result computation

Db2 for z/OS does not support the following keywords :

- FROM FIRST
- FROM LAST
- RESPECT NULLS
- IGNORE NULLS

If you specify any of these, SQLCODE -199 is returned. Hence you can only use this function without these keywords. The database engine of the accelerator executes the function with default settings, which equates to specifying FROM FIRST RESPECT NULLS.

NTILE

This is function returns the quantile rank of a row. If the argument can be null, the result can be null. If the argument is null, the result is the null value. The result is the quantile rank, which is associated with the current row. Equivalent sort keys are not considered when rows are divided into quantiles. Rows with equivalent sort keys can be assigned to different quantiles based on the non-deterministic order of these sort keys. Therefore, NTILE is a non-deterministic function.

Result computation

The NTILE function computes the quantile rank of a row by dividing the ordered rows within the OLAP window into *num-tiles* quantiles and returns a value between 1 and *MIN(n,num-tiles)*, where *n* is the number of rows within the OLAP window. If *n* is evenly divisible by num-tiles, the rows in the OLAP window are grouped into *num-tiles* quantiles, each containing *n/num-tiles* rows. Otherwise, each of the quantiles *1* through *MOD(n,num-tiles)* is assigned *n/num-tiles+ 1* rows while each of the quantiles *MOD(n,num-tiles) + 1* through *num-tiles* is assigned *n/num-tiles* rows.

Result data type

DECFLOAT(34)

PERCENT_RANK

This is a distribution function that returns a relative percentile rank of a row within an OLAP window, expressed as a value between 0.0 and 1.0.

Result computation

If the number of rows in the OLAP window is greater than 1, the result is the *RANK of the current row in the OLAP window -1* divided by the *total number of rows in the OLAP window -1*.

If the number of rows in the OLAP window is less than or equal to 1, the result is 0.0.

More information
PERCENT_RANK aggregate function

RATIO_TO_REPORT
This function returns the ratio of an *argument* to the *sum of the arguments* in an OLAP partition.

Result computation
The division is always executed by using the DECFLOAT(34) data type. If the argument can be null, the result can be null; if the argument is null, the result is the null value.

Result data type
DECFLOAT(34)

REGEXP_COUNT
This scalar function returns a number that says how many times a regular expression pattern can be found in a string.

More information
REGEXP_COUNT scalar function

REGEXP_INSTR
This scalar function returns the starting or ending position of the matched substring, depending on the value of the *return_optionargument*.

More information
REGEXP_INSTR scalar function

REGEXP_LIKE
This scalar function returns an INTEGER value indicating if the regular expression pattern is found in a string. The function can be used only where a predicate is supported.

Result data type
The result of this function is cast to an INTEGER value of 0 or 1, where 0 is *false*, and 1 is *true*. This is because Db2 for z/OS currently does not support Boolean functions or data types.

More information
REGEXP_LIKE scalar function

REGEXP_REPLACE
This scalar function returns a modified version of the source string, in which matching occurrences of the regular expression pattern are replaced with the specified replacement string.

Result data type
The length of the result buffer is 32672 bytes if a replacement string is specified. If a replacement string is not specified, the result buffer length is the same as the input buffer length, except for the following data types:

Data type	Result buffer length in Bytes
Numeric	42
DATE	10
TIME	8
TIMESTAMP	32

More information
REGEXP_REPLACE scalar function

REGEXP_SUBSTR
This scalar function returns one occurrence of a substring that matches the regular expression pattern.

Result data type
The result buffer length is the same as the input buffer length, except for the following data types:

Data type	Result buffer length in Bytes
Numeric	42
DATE	10
TIME	8
TIMESTAMP	32

More information

REGEXP_SUBSTR scalar function

Related information

OLAP specification

Supported special registers

IBM Db2 Analytics Accelerator supports most Db2 special registers. During SQL routing to an accelerator, these special registers are replaced with the equivalent dashDB functions or with a literal string containing the register's value at the time of statement execution. In some cases, this leads to different query results.

The following special registers are supported by dashDB function substitution:

- CURRENT ACCELERATOR
- CURRENT APPLICATION COMPATIBILITY
- CURRENT APPLICATION ENCODING SCHEME
- CURRENT CLIENT_ACCTNG
- CURRENT CLIENT_APPLNAME
- CURRENT CLIENT_CORR_TOKEN
- CURRENT CLIENT_WRKSTNNAME
- CURRENT DATE

  Replaced with the dashDB function DATE(CURRENT DATE) on the accelerator.
- CURRENT DEBUG MODE
- CURRENT DECFLOAT ROUNDING MODE
- CURRENT DEGREE
- CURRENT EXPLAIN MODE
- CURRENT GET_ACCEL_ARCHIVE
- CURRENT LOCALE LC_CTYPE
- CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION
- CURRENT MEMBER
- CURRENT OPTIMIZATION HINT
- CURRENT PACKAGE PATH
- CURRENT PACKAGESET
- CURRENT PATH
- CURRENT PRECISION
- CURRENT QUERY ACCELERATION
- CURRENT QUERY ACCELERATION WAITFORDATA
- CURRENT REFRESH AGE
- CURRENT ROUTINE VERSION

- CURRENT RULES
- CURRENT SCHEMA
- CURRENT SERVER
- CURRENT SQLID
- CURRENT TEMPORAL BUSINESS_TIME
- CURRENT TEMPORAL SYSTEM_TIME
- CURRENT TIME

  Replaced with dashDB function TIME(CURRENT TIME) on the accelerator.

  Restriction: Microseconds are not returned in the query results. A zero is displayed in the positions where you would expect microseconds.
- CURRENT TIMESTAMP

  Replaced with the dashDB function TIMESTAMP(CURRENT TIMESTAMP).

  Restriction: Microseconds are not returned in the query results. A zero is displayed in the positions where you would expect microseconds.
- SESSION_USER
- SESSION TIME ZONE

Most of the remaining special registers are supported through literal string substitution in the SQL. For example, a reference to CURRENT SQLID will be replaced in the SQL with the literal string value from that register before the SQL is submitted to dashDB for processing.

Conditions that prevent query routing to an accelerator

Db2 for z/OS does not route a query to an accelerator if any of the following conditions applies.

- You use a distributed data facility (DDF) client, such as an Open Database Connectivity (ODBC) client, that does not support *level 8* of the IBM Distributed Relational Database Architecture™ (DRDA).
- The encoding scheme of an SQL statement is `M` because the tables use different encoding schemes, or the query contains an expression that explicitly relates to a coded character set identifier (CCSID), for example, a cast specification with a CCSID option.
- The FROM clause of the query specifies a data-change-table-reference, or, in other words, the query is selected from a FINAL TABLE or from an OLD TABLE.
- The query contains a string expression (including columns) with an unsupported subtype. See Table 7 on page 24.
- The query contains an expression with an unsupported result data type or subtype. See “Supported data types” on page 22.
- The query refers to a column that uses a field procedure (FIELDPROC).
- The query uses a special register other than:
  - CURRENT DATE
  - CURRENT TIME
  - CURRENT TIMESTAMP
- A LOCAL date format is used, that is, one or more of the following conditions apply:
  1. The query contains a CHAR function in which LOCAL is specified as the second argument.
  2. The query contains a date or time expresssion and the DATE FORMAT field of the DSNTIP4 installation panel specifies LOCAL.
  3. Application programs that process SQL on Db2 for z/OS have been precompiled with the DATE(LOCAL) option.

Exception: You can enable LOCAL date expressions in the format dd/mm/yyyy or yyyymmdd   (with two space characters at the end) by including the QUERY_ACCEL_OPTIONS ZPARM value 4 or 7. For more information, see *Installing Db2 libraries with IBM Db2 Analytics Accelerator for z/OS support* in the *IBM Db2 Analytics Accelerator for z/OS: Installation Guide*.

- The query contains a sequence expression (NEXTVAL or PREVVAL).
- The query contains a user-defined function (UDF)
- The query contains a ROW CHANGE expression
- The query contains a string constant that is longer than 16000 characters.
- A new column name is referenced in a sort-key expression, for example:

```
SELECT C1+1 AS X, C1+2 AS Y FROM T WHERE ... ORDER BY X+Y;
```

- The query uses a DAY function where the argument of the function specifies a duration.
- The query uses a TRANSLATE function with only two arguments.
- The query uses one of the following scalar functions and one or more of its arguments contain a parameter marker or a scalar fullselect:
  - LOCATE
  - MICROSECOND
  - MIDNIGHT_SECONDS
  - TRANSLATE
- The query uses a table function, such as ADMIN_TASK_LIST or ADMIN_TASK_STATUS.
- The query uses an unsupported scalar function. See Table 11 on page 38
- The GET_ACCEL_ARCHIVE special register is set to the value YES (meaning that archived data will be included in the query) and the query references a table that has the value C in the ARCHIVE column of the SYSACCEL.SYSACCELRATEDTABLES catalog table. A value of A or C indicates that the table data has been archived on an accelerator. The value C means that the archived data is not stored on the accelerator that is supposed to process the query. In other words, the active query data and the archived data are not located on the same accelerator. To process the query successfully, the ARCHIVE column must show the value A or no value. The value A indicates that both, the active data and the archived data, are located on the same accelerator.
- The instruction to process “SELECT FROM ... INSERT INTO ... ” statements (ZPARM QUERY_ACCEL_OPTIONS includes the value 2) is ignored if the target table uses an encoding scheme that is different from the encoding scheme of the tables in the SELECT statement.
- A *static* SQL query that includes a FOR SYSTEM_TIME ... or FOR BUSINESS_TIME ... is not routed to an accelerator if one of the following bind options (value YES, default) is used.

Table 12. Unsupported bind options

Clause	Bind option (value YES, which is the default)
– FOR BUSINESS_TIME AS OF – FOR BUSINESS_TIME FROM ... TO ... – FOR BUSINESS_TIME BETWEEN ... AND...	– BUSTIMESENSITIVE – BUSINESS_TIME SENSITIVE
– FOR SYSTEM_TIME AS OF – FOR SYSTEM_TIME FROM ... TO ... – FOR SYSTEM_TIME BETWEEN ... AND ...	– SYSTIMESENSITIVE – SYSTEM_TIME SENSITIVE

In cases like this, you receive error message -4742 with reason code 18.

Related concepts

EXPLAIN information

After creating the necessary EXPLAIN tables, you can analyze queries by invoking the Db2 EXPLAIN function. The analysis shows whether a query can be accelerated, indicates the reason for a failure, and gives a response time estimate. The outcome of the analysis can also be visualized in an access plan graph.

Isolation levels

Db2 isolation levels are ignored by IBM Db2 Analytics Accelerator.

Every table on an accelerator has its counterpart. That is, the table has been derived from a table that resides in Db2 for z/OS. An accelerator-shadow table is thus a copy of a projection of a Db2 table. In most cases, the projection is the Db2 table itself. Columns of the original Db2 table are excluded from the projection only if a column uses an unsupported data type.

You can change a Db2 table practically at any time by means of INSERT, UPDATE, or DELETE operations, mass imports (LOAD), and schema modifications (selected DDL). In this version of IBM Db2 Analytics Accelerator for z/OS, changes to a DB2 table are propagated automatically to the associated accelerator-shadow table only if the incremental update feature is used. If incremental updates are not enabled, you must update the accelerator-shadow tables manually in order to synchronize the table data. You can start the update from IBM Db2 Analytics Accelerator Studio or run the SYSPROC.ACCEL_LOAD_TABLES stored procedure for this purpose.

The update of accelerator-shadow tables is an asynchronous process, that is, there is no guarantee that the data in the Db2 tables and the data in the accelerator-shadow tables are always in sync. Even with incremental updates, a certain degree of latency is involved, which means that tables on an accelerator are updated somewhat later than the original DB2 tables. Therefore, the result set that a query returns when it is processed by Db2 might differ from the result set that the same query produces when it is processed by IBM Db2 Analytics Accelerator. This implies that isolation levels set for Db2 data are ignored or must be ignored by applications that send queries to IBM Db2 Analytics Accelerator. Result sets that are returned by IBM Db2 Analytics Accelerator do not necessarily comply with any isolation level setting.

To ensure data consistency on a relatively safe level, update accelerator-shadow tables immediately before you run the query and choose an appropriate lock scope so that the affected Db2 tables cannot be altered during the update. This, however, still does not guarantee that the results of an accelerated query will show the latest changes because the lock might prevent or postpone an update of the original Db2 table data.

If you use the incremental update function, you can ensure that committed Db2 for z/OS changes are propagated to the accelerator by calling the ACCEL_CONTROL_ACCELERATOR stored procedure with the **`<waitForReplication>`** command.

If the most recent table changes must be reflected in the result set, set the CURRENT QUERY ACCELERATION special register to the value NONE. This means that the query is executed by Db2 and will not be accelerated.

Related concepts

Updating accelerator tables continually

The incremental update function of IBM Db2 Analytics Accelerator for z/OS allows you to update accelerator-shadow tables continually. Changes to the data in original Db2 for z/OS tables are thus propagated to the corresponding target tables with a high frequency and just a brief delay. This way, query results from an accelerator are always extracted from recent, close-to-realtime data.

Related tasks

Loading tables

Successful queries against accelerator-shadow tables are possible only if the tables contain data. Therefore, you must load the tables after their definition (empty structure) has been copied to the accelerator.

Incompatibilities between Db2 for z/OS and IBM Db2 Analytics Accelerator

In the list of incompatibilities, you find explanations for execution errors or different query results that Db2 for z/OS and IBM Db2 Analytics Accelerator might return for the same query.

Data types

See how IBM Db2 Analytics Accelerator handles certain Db2 data types:

FLOAT

The FLOAT data type is an approximate data type. The result can be different due to platform differences (hexadecimal floating-point numbers (HFP) versus IEEE754 floating-point numbers) or other internal handling differences. This applies to expressions that return FLOAT data and to expressions that contain a FLOAT data-type argument.

Related concepts

Differences in floating-point error tolerances
For queries that involve floating-point aggregates, you can expect IBM Db2 Analytics Accelerator for z/OS to produce results that are different from those delivered by inhouse Db2 for z/OS processing.

Division and multiplication of decimal values

During the division or multiplication of decimal values, Db2 for z/OS might truncate operand values if at least one of the operands is defined with a precision greater than 15. If that happens, the result of the calculation will be imprecise or wrong. IBM Db2 Analytics Accelerator does not truncate operand values internally, so the results will be more precise.

Example: Consider the following query:

```
CREATE TABLE TEST_TABLE(
D_DECIMAL_4_2 DECIMAL(4,2),
D_DECIMAL_17_2 DECIMAL(17,2),
D_DECIMAL_30_6 DECIMAL(30,6)
);
INSERT INTO TEST_TABLE VALUES(3.5, 3.5, 0.5);
INSERT INTO TEST_TABLE VALUES(1.5, 1.5, 1.5);

SELECT D_DECIMAL_17_2/D_DECIMAL_30_6, D_DECIMAL_17_2*D_DECIMAL_30_6
FROM TEST_TABLE;

SELECT D_DECIMAL_4_2/D_DECIMAL_30_6, D_DECIMAL_4_2*D_DECIMAL_30_6
FROM TEST_TABLE;
```

Db2 for z/OS returns results as follows:

- For the first calculation in the query:

```
+----------+
| ?  | 1.5 |
| 1  | 1.5 |
+----------+
```

- For the second calculation in the query:

```
+------------+
| ?   | 1.75 |
| 1.5 | 2.25 |
+------------+
```

The question mark (?) means that an overflow has occurred (division by zero) because Db2 for z/OS has internally reduced the value `0.5` to `0`.

IBM Db2 Analytics Accelerator would have returned the correct results, which are the same for both calculations in the query:

```
+------------+
| 7   | 1.75 |
| 1   | 2.25 |
+------------+
```

The truncation problem is discussed in subsections of the *Arithmetic with two decimal operands* topic of the SQL reference in the Db2 11 for z/OS Knowledge Center. A link to this topic is provided under **Related information**.

Related information

Arithmetic with two decimal operands

Encoding issues

Consider the following points when you have to deal with Unicode SQL statements, multibyte (mixed) data in connection with parameter markers, and multibyte or double-byte EBCDIC columns.

Unicode SQL statements

Generally, when a UNICODE-encoded SQL statement contains characters or strings (as a literal, host variable, or parameter marker) that refer to a single-byte-encoded EBCDIC column, then Db2 tries to convert these strings to EBCDIC. Sometimes, such a string cannot be converted properly because it contains invalid UNICODE characters or characters that do not exist in the target EBCDIC code page. In such cases, the results of the SQL operation and the SQLCODE returned by the accelerator might be different from those returned by Db2 for z/OS with regard to the following aspects:

- Queries might return a different number of rows.
- `SQLCODE 0` might be returned instead of the SQL warning codes +331 or +335.
- Different substitution characters might be put into returned or inserted strings.
- Returned or inserted strings or substitution characters might be different if multiple encoding is enabled on the accelerator.

Mixed data and parameter markers

If a Db2 subsystem is set up to contain mixed data (`mixed=yes`), that is, if differently encoded data is allowed to coexist, and the coded character set identifier (CCSID) is not specified for parameter markers in a query, the query might return incorrect results.

The LIKE operator in connection with binary wildcard characters

Depending on the system executing a query (Db2 for z/OS or IBM Db2 Analytics Accelerator, the query might not return any results or wrong results if the LIKE operator is used in combination with a *binary* wildcard character (% or _). The reason for that is that both systems expect a different binary encoding of the wildcard characters.

Db2 for z/OS

Valid binary encodings of the wildcard characters are bx`'6c'` for the percent sign (%) and bx`'6d'` for the underscore character (_).

IBM Db2 Analytics Accelerator

Valid binary encodings of the wildcard characters are `X'25'` for the percent sign (%) and `X'5F'` for the underscore character (_).

This means, for example, that if a query uses `bx'6c'` to refer to the percent sign as a wildcard character, Db2 for z/OS will return the expected result, but IBM Db2 Analytics Accelerator will not.

On the contrary, if an expression like `SELECT COL FROM TABLE WHERE COL LIKE BINARY('A%')` is encountered, IBM Db2 Analytics Accelerator will give you the expected result, but Db2 for z/OS will not.

Exception handling

When an exception, such as a buffer overflow, a division by zero, or an out-of-range value occurs, Db2 returns the value NULL combined with a +802 warning, provided that the faulty expression is in the outermost SELECT statement and affects only one row. This is also called *friendly arithmetic*. IBM Db2 Analytics Accelerator will always issue an error in cases like these.

Maximum row length

The maximum row length for a table on an accelerator is 1048319 bytes. If the combined length of all column fields in a Db2 table exceeds this limit, you cannot define that table on the accelerator.

Predicate evaluation

Db2 tolerates SQL errors during predicate evaluation as long as the error has no impact on the result of the evaluation. IBM Db2 Analytics Accelerator, on the contrary, always returns an error. This rule applies to predicates consisting of multiple terms (an SQL error in a predicate consisting of only one term cannot be tolerated). An SQL error in a predicate does not influence the outcome of the predicate evaluation if it does not matter whether the error makes the predicate logically *true* or *false*. For example, if the predicate is a conjunction of terms as in `P1 AND P2`, and the evaluation of P1 results in an SQL error, then this error does not influence the outcome if P2 evaluates to *false* because both, *true AND P2* and *false AND P2*, evaluate to *false*. Table 13 on page 48 shows all relevant permutations for two-term predicates (E = *error*, T = *true*, F = *false*). Values in parentheses indicate that the P2 term is not evaluated if an evaluation of P1 already yields the result of the entire predicate evaluation.

Table 13. Relevant combinations of two-term predicates

P1	P2	P1 AND P2	P1 OR P2
E	E	E	E
E	T	E	T
E	F	F	E
T	E	E	(T)
F	E	(F)	E

The reason for choosing this evaluation method is that Db2 sometimes changes the order of term evaluation internally. The change is not announced or reported, and therefore goes unnoticed. Depending on the order of term evaluation, Db2 returns an error in one case, but not the other.

The behavior is very similar for the CASE and DECODE functions. These are basically SQL implementations of IF ... THEN ... ELSE constructs and exist in Db2 and IBM Db2 Analytics Accelerator. Db2 will ignore a wrongly defined option as long as it can select a correct one that evaluates to true. IBM Db2 Analytics Accelerator, on the other hand, will first check all available options, and return an error if one of these is not correctly defined. In the latter case, it does not matter whether selecting one of the correctly defined options would have led to a correct result.

SQL issues

In a grouping select, the HAVING clause can only reference expressions that are single-valued within a group. That is, you can only reference group fields, aggregates, or single-valued expressions that are derived from group fields or aggregates (which must include constants).

Time and date values

Time and date constructs in the dashDB database often differ from the constructs used by Db2 for z/OS, so that values and expressions must be converted. The conversion is often accompanied by limitations.

24:00:00 conversion

This product supports the Db2 for z/OS time representation 24.00.00 for midnight (end of day) in TIME and TIMESTAMP columns or SQL expressions. There is no need to set up a conversion.

However, although the time 24:00:00 is accepted as a valid time, it is never returned by the accelerator as the result of a time addition or subtraction. This holds true even if the time added or subtracted is 0 (seconds, minutes, or hours) or 24 hours: The result will always be 00:00:00. Db2 for z/OS will return 24:00:00 in these cases.

Suppose you have a table TEST with a timestamp column C1. You insert just the following value into the column: `9999-12-01-24.00.00.000000000000`. Now you run the following query, which includes a simple calculation:

```
SELECT C1 - 1 MONTH,C1 FROM TEST
```

That is, you want to retrieve the values from C1, but one month is to be deducted from each value. Since there is just one value, you will see only one record in the query results. But note the difference depending on where the query was executed:

Db2 for z/OS
    `9999-11-01-24.00.00.000000000000`

IBM Db2 Analytics Accelerator
    `9999-11-02-00.00.00.000000000000`

As you can see, Db2 for z/OS gives you 24:00, end of day, on the 1st of November, but IBM Db2 Analytics Accelerator gives you 00:00, beginning of the next day (2nd of November). Both values mean exactly the same, but there is a difference in the representation.

Dates in TIMESTAMP expressions

If a date is supplied as the only argument to a TIMESTAMP expression, for example `TIMESTAMP(DATE('2015-12-31'))`, Db2 for z/OS returns a timestamp with a time portion that shows no fractional seconds, whereas IBM Db2 Analytics Accelerator returns fractional seconds with a precision of six digits:

Db2 for z/OS	`2015-12-31 00:00:00`
IBM Db2 Analytics Accelerator	`2015-12-31 00:00:00.000000`

You can circumvent this problem by supplying the date as a string value, for example `TIMESTAMP('2015-12-31')` because this gives the same return value in both systems, that is:

```
2015-12-31 00:00:00.000000
```

Strings as time values and their conversion

For some scalar functions, such as TIME, Db2 accepts a time value, a timestamp, and also a string representation of times and timestamps. When Db2 evaluates the scalar function, it first checks whether a value can be interpreted as a time. If this fails, it tries to interpret the value as a timestamp. The corresponding accelerator functions do not accept string values. During query conversion, the content of a string is not evaluated. Instead, it is always cast to a time value. If the string represents a valid time format on the accelerator, IBM Db2 Analytics Accelerator processes the query. Otherwise, if the string represents a timestamp, it returns an error. A workaround is to cast the string explicitly to a timestamp in the query.

Affected functions:

- CAST (if the operation converts the string to a Db2 TIME value)
- EXTRACT(HOUR FROM <arg>)
- EXTRACT(MINUTE FROM <arg>)
- HOUR

- MIDNIGHT_SECONDS
- MINUTE
- SECOND
- TIME

Trailing blanks

Trailing blanks cause incorrect results under the following conditions:

- The result data type of an SQL expression is VARCHAR or VARGRAPHIC (as with LEFT, LTRIM, REPLACE, RIGHT, or TRIM for example).
- The original values returned as VARCHAR or VARGRAPHIC contain trailing blanks.
- A GROUP BY clause is used to sort the results on the VARCHAR or VARGRAPHIC column.

  Important: If you use the UNION keyword to merge the results of distinct SELECT statements, then this operation is internally realized by using GROUP BY clauses. That is, you might see incorrect results even though your query does not contain a GROUP BY clause.

Queries of this type return incorrect results because the accelerator uses column-organized tables for performance reasons. The columnar engine used for these tables removes trailing blanks from VARCHAR or VARGRAPHIC values that are sorted by a GROUP BY clause. If the original values are of varying length and contain trailing blanks, the blanks are removed when the rewritten query is executed. Hence the grouped values in the column-organized table do not contain trailing blanks, and the length calculation will inevitably lead to deviating results.

GROUP BY example

```
SELECT <varchar_col> || 'a' FROM <table> GROUP BY <varchar_field>
```

In the first part of the query, the letter a is added to each value of a VARCHAR column at the end. The result data type is also VARCHAR. If the original values contain trailing blanks, and the first part of the query is run on its own, you see the following, different outputs depending on where the query is run:

Db2 for z/OS:

`<varchar_value> a`

That is, the trailing blanks are preserved for each value in the column.

IBM Db2 Analytics Accelerator:

`<varchar_value>a`

The trailing blanks are removed by the accelerator.

The second part of the query adds a GROUP BY clause to the first part. The outputs of the entire query differ as follows:

Db2 for z/OS:

- `<varchar_value> a`
- `<varchar_value>a`

Db2 for z/OS returns both possibilities for each grouped value.

IBM Db2 Analytics Accelerator:

`<varchar_value>a`

IBM Db2 Analytics Accelerator returns just the values without the trailing blanks.

UNION example

```
((SELECT "ACC001"."COL1","ACC001"."COL2","ACC001"."COL3"
  FROM "DWAXXYYZ"."TB12345-0136008" ACC001
   WHERE "ACC001"."COL1"<2)
UNION
(SELECT "ACC002"."COL1","ACC002"."COL2","ACC002"."COL3"
 FROM "DWAXXYYZZ"."TB12345-0136008" ACC002
  WHERE "ACC002"."COL1"<2)) WITH UR
```

In this example, the UNION keyword is used to merge the results of two SELECT queries, each of these against a copy of the same table. The copies exist on different accelerators. One of these accelerators is an IBM Db2 Analytics Accelerator for z/OS Version 5.1.x accelerator, the other a version 7.1.x accelerator.

The version 5.1.x accelerator returns column values with the same number of trailing blanks as Db2 for z/OS. However, the version 7.1.0 accelerator returns column values with fewer or no trailing blanks.

Remember: In general, queries are internally rewritten, so that they might contain a GROUP BY clause even though no such clause was given in the original query. In this case, the number of trailing blanks extracted from VARCHAR or VARGRAPHIC columns can be different depending on where the query is run.

Incompatibilities related to IBM InfoSphere Data Replication for z/OS

You might encounter execution errors or deviating query results when you use the incremental update function. Some of these errors or result deviations are due to incompatibilities between Db2 for z/OS, IBM Db2 Analytics Accelerator for z/OS, and IBM InfoSphere Change Data Capture for z/OS. The following section deals with these incompatibilities.

IBM InfoSphere Data Replication for z/OS is the main software component behind the incremental update function of IBM Db2 Analytics Accelerator for z/OS.

All incompatibilities that are listed in the section "Incompatibilities between Db2 for z/OS and IBM Db2 Analytics Accelerator" on page 46 also apply to IBM InfoSphere Data Replication for z/OS.

In addition, incompatibilities that affect the regular load function of the product (the SYSPROC.ACCEL_LOAD_TABLES stored procedure) also affect the incremental update function. However, the following exceptions exist when IBM InfoSphere Data Replication for z/OS transfers data to tables on an accelerator:

- Values of VARCHAR FOR MIXED and VARGRAPHIC columns that exceed 16000 characters are cut off when they are received by IBM Db2 Analytics Accelerator for z/OS. Mind that unlike Db2 for z/OS, IBM InfoSphere Data Replication for z/OS does not issue a warning or error message in cases like this.
- If you run product versions 5 and 7 in parallel, be aware that a column data type supported by one version might not be supported by the other. Since the CDC capture agent is the same for both versions, it is unable to differentiate between version 5 and version 7 accelerators. Hence you might be able to update a table on a version 5 accelerator without problems, but receive an error when you try to update the same table on a version 7 accelerator.

Restrictions of accelerator-only tables

Here is a summary of the restrictions that exist for accelerator-only tables.

Current limitations

The use of accelerator-only tables brings extra workloads to your accelerators that compete for the same resources, such as CPU, disk, and system memory. This might impact the stability of your production environment.

Data-definition (DDL) statements

The following restrictions exist for data-definition statements and accelerator-only tables:

- The `IN ACCELERATOR` clause can be used only once in a statement.
- A `CREATE TABLE` must not contain a materialized query definition if the table is an accelerator-only table.
- A `CREATE MASK` statement must not define a mask for an accelerator-only table.

- A `CREATE PERMISSION` statement must not define a column permission for an accelerator-only table.
- A `DECLARE GLOBAL TEMPORARY TABLE` statement must not reference an accelerator-only table.
- A `CREATE TRIGGER` or `ALTER TRIGGER` statement must not reference an accelerator-only table in the body of the trigger.
- A `CREATE TRIGGER` or `ALTER TRIGGER` statement with an `INSTEAD OF` clause must not reference a view that references an accelerator-only table in the body of the trigger.
- An `ALTER TABLE` statement is not allowed if the table is an accelerator-only table.
- An `ALTER TABLESPACE` statement is not allowed if the table space contains accelerator-only tables.
- A `DROP TABLESPACE` statement fails if the table space contains accelerator-only tables. You must first drop the accelerator-only tables by using a `DROP TABLE` statement.
- The `DROP DATABASE` statement does not behave as expected:
  - If you create an accelerator-only table without specifying a database, the table is created in the default database as usual, but a `DROP DATABASE` statement does not remove that database.
  - A `DROP DATABASE` statement does not remove a database and the tables therein if some of the tables are accelerator-only tables. To remove such a database, first remove all accelerator-only tables by using separate `DROP TABLE` statements, and then the database.
- A `RENAME TABLE` statement fails if the table is an accelerator-only table.
- A `LOCK TABLE` statement must not refer to an accelerator-only table.
- A `CREATE INDEX` statement fails if the table to be indexed is an accelerator-only table.
- A `CREATE SYNONYM` statement fails if the synonym is for an accelerator-only table.
- You cannot create more than one accelerator-only table in a single transaction. The following example does not work:

  ```
  CREATE TABLE AOT1 ... IN ACCELERATOR ...
  CREATE TABLE AOT2 ... IN ACCELERATOR ...
  COMMIT;
  ```

  This example, on the other hand, does work:

  ```
  CREATE TABLE AOT1 ... IN ACCELERATOR ...
  COMMIT;
  CREATE TABLE AOT2 ... IN ACCELERATOR ...
  COMMIT;
  ```

In addition, the following clauses or keywords cannot be used in DDL statements that refer to accelerator-only tables:

Table 14. Unsupported clauses and keywords

APPEND	AS FULLSELECT[1]	AUDIT	BUFFERPOOL
CHECK	COMPRESS	DATA CAPTURE	DSSIZE
EDITPROC	FIELDPROC	FOREIGN KEY	GENERATED
HASH	IMPLICITLY HIDDEN	IN TABLESPACE	INLINE LENGTH
LIKE	LOGGED	MEMBER CLUSTER	NOT LOGGED
PARTITION BY	PERIOD	PRIMARY KEY	TRACKMOD
UNIQUE INDEX	VALIDPROC	VOLATILE	WITH DEFAULT

[1] The restriction for the `AS FULLSELECT` clause is limited to the `CREATE TABLE` statement. It does work with other statements that reference an accelerator-only table, such as the `CREATE VIEW` statement.

Other SQL statements

The following statements are not allowed:

- You cannot run an `INSERT INTO <accelerator-only table> SELECT ... FROM ...` statement if the SELECT part of the statement references a normal Db2 table that has no loaded counterpart on the accelerator. The statements succeeds only if the SELECT part references one or more tables on the same accelerator as the accelerator-only table.
- You cannot submit an UPDATE statement against table columns that serve as distribution keys.
- You cannot insert values directly into a decimal column of an accelerator-only table if these values are the result of a division where the dividend, the divisor, or both are decimals. However, the operation works if the division is part of the SELECT portion in an `INSERT INTO ... SELECT FROM …` statement.

Restrictions on data types, column types, and code pages

Note that restrictions that apply to certain data types in general also apply to the use of these types in accelerator-only tables. In addition to that, the following restrictions exist for data types, column types, and code pages (CCSIDs), or are especially noteworthy:

- The column names in an accelerator-only table must not start with a tilde (~) followed by an underscore (_) character.
- The CHAR FOR BIT DATA and VARCHAR FOR BIT DATA data types are not supported in accelerator-only tables if the environment has been set up for the use of mixed data (ZPARM MIXED=YES).
- You cannot define a large-object (LOB) column for an accelerator-only table.
- You cannot define a security label column for an accelerator-only table.

Stored procedures

Some of the IBM Db2 Analytics Accelerator stored procedures do not work on accelerator-only tables:

SYSPROC.ACCEL_ADD_TABLES
Submit a `CREATE TABLE ... IN ACCELERATOR` statement instead.

SYSPROC.ACCEL_REMOVE_TABLES
Submit a suitable `DROP TABLE` statement instead.

SYSPROC.ACCEL_SET_TABLES_REPLICATION
Accelerator-only tables cannot be enabled for incremental updates.

SYSPROC.ACCEL_ARCHIVE_TABLES
You cannot archive partitions of accelerator-only tables because accelerator-only tables are not partitioned.

SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES
Since you cannot archive partitions of accelerator-only tables, you can neither restore archived partitions.

SYSPROC.ACCEL_SET_TABLES_ACCELERATION
You cannot disable acceleration on accelerator-only tables because they exist on an accelerator only, and thus must be enabled for acceleration.

Database limitations

In general, all limitations of the database on the IBM Integrated Analytics System also apply to accelerator-only tables.

Different results

Results obtained by running the same query against a normal Db2 table and an accelerator-only table might differ under the following conditions:

- One of the tables that are specified in the query contains a CHAR column in Unicode or in a mixed format, and the column values are padded with blank characters.

- An INSERT statement for DECIMAL values fails with a numeric data overflow error if the allowed maximum of 31 digits is exceeded, as in the following example:

  ```
  INSERT INTO table ('decimal(31,0)_column')
  VALUES (9999999999999999999999999999999.9);
  ```

  The value to be inserted here has a total length of 32. Compare the result that IBM Db2 Analytics Accelerator returns with the result that Db2 for z/OS would return if this were an INSERT statement for a regular Db2 table:

  IBM Db2 Analytics Accelerator:
  You receive an out-of-range error message:

  ```
  "The numeric literal "9999999999999999999999999999999.9" is not
  valid because its value is out of range."
  ```

  Db2 for z/OS:
  Db2 for z/OS truncates such values:

  ```
  9999999999999999999999999999999.
  ```

Newer Db2 features

Some newer features, which were introduced with DB2 11 for z/OS, are not supported by IBM Db2 Analytics Accelerator. Other new features in Db2 11 for z/OS are either fully supported, or their use is restricted. Read on to find out about the details.

General

The EXPLAIN reason codes that hint at routing errors in connection with features introduced with newer Db2 for z/OS versions are the codes 11, 15, and 17.

Reason code 11 covers unsupported expressions, data types, and so on, while reason code 15 is a general reason code for all errors that are not covered by reason code 11. Reason code 17 is returned if you try to process a query with an INSERT statement, but have not set the QUERY_ACCEL_OPTIONS parameter accordingly. For more information about the QUERY_ACCEL_OPTIONS parameter, see *Installing Db2 libraries with IBM Db2 Analytics Accelerator support* in the *IBM Db2 Analytics Accelerator for z/OS: Installation Guide*

Access control of columns or rows

IBM Db2 Analytics Accelerator handles columns with access control or Db2 tables with row access control as follows:

- Columns with access control are ignored when the corresponding accelerator-shadow table is loaded.
- Tables for which row access control has been defined cannot be added to an accelerator.
- If a query references a column for which a column mask has been activated, the query is not routed to an accelerator.

Cross-loading of tables

When a LOAD DATA INCURSOR ... INTO statement is used, the query inherits the setting of the CURRENT QUERY ACCELERATION special register. Hence queries are automatically routed to an accelerator if this has been enabled by the CURRENT QUERY ACCELERATION special register.

Cross-loading is not supported if the CURRENT GET_ACCEL_ARCHIVE special register is set. Hence queries with this setting cannot be accelerated in a cross-loading context.

CURRENT EXPLAIN MODE special register

Query routing to an accelerator works for queries that use the CURRENT EXPLAIN MODE special register if the special register is set to the value YES or EXPLAIN. Meaning of these settings:

CURRENT EXPLAIN MODE = YES

Enables the Db2 EXPLAIN facility and causes EXPLAIN information to be inserted into the EXPLAIN tables for eligible dynamic SQL statements after preparing and executing the statement. Dynamic SQL statements are hence compiled and executed normally.

CURRENT EXPLAIN MODE = EXPLAIN

Enables the Db2 EXPLAIN facility and captures EXPLAIN information for eligible dynamic SQL statement after preparing the statement. Note that with setting, a dynamic statement or query will not be executed, except for the SET clauses that it might contain.

New OLAP functions

IBM Db2 Analytics Accelerator supports the following moving aggregate functions:

- AVG
- CORRELATION (see **Notes®**)
- COUNT
- COUNT_BIG
- COVARIANCE (see **Notes**)
- COVARIANCE_SAMP (see **Notes**)
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE (VARIANCE_SAMP)

Notes:

- The result data type that IBM Db2 Analytics Accelerator returns for the CORRELATION, COVARIANCE, and COVARIANCE_SAMP functions is DOUBLE, and not DECFLOAT, as in Db2 for z/OS. This results in a loss of precision.
- Queries that use the previously mentioned moving aggregate functions are not accelerated if the window-aggregation-group clause specifies just a group-end clause, for example:

```
SELECT C1, C2, C3, SUM(C2)
OVER (ORDER BY C3 ROWS UNBOUNDED FOLLOWING)
FROM <table>
ORDER BY 1,2,3,4;
```

Temporal tables

. Queries referencing temporal tables usually contain the following expressions or clauses:

- `FOR SYSTEM_TIME AS OF`
- `FOR SYSTEM_TIME FROM ... TO ...`
- `FOR SYSTEM_TIME BETWEEN ... AND ...`
- `FOR BUSINESS_TIME AS OF`
- `FOR BUSINESS_TIME FROM ... TO ...`
- `FOR BUSINESS_TIME BETWEEN ... AND ...`

Restriction: A *static* SQL query that includes a `FOR SYSTEM_TIME ...` or `FOR BUSINESS_TIME ...` is not routed to an accelerator if one of the following bind options (value YES, default) is used.

Table 15. Unsupported bind options

Clause	Bind option (value YES, which is the default)
• FOR BUSINESS_TIME AS OF • FOR BUSINESS_TIME FROM ... TO ... • FOR BUSINESS_TIME BETWEEN ... AND...	• BUSTIMESENSITIVE • BUSINESS_TIME SENSITIVE
• FOR SYSTEM_TIME AS OF • FOR SYSTEM_TIME FROM ... TO ... • FOR SYSTEM_TIME BETWEEN ... AND ...	• SYSTIMESENSITIVE • SYSTEM_TIME SENSITIVE

In cases like this, you receive error message `-4742` with reason code 18.

Timestamps

The portion that denotes fractional seconds in timestamps can have up to twelve digits in Db2 for z/OS and also in dashDB. Hence IBM Db2 Analytics Accelerator fully supports this format. That is, values are not truncated as in earlier versions of IBM Db2 Analytics Accelerator for z/OS.

Restrictions: Timestamps with timezone specifications are not supported, to the effect that:

- SQL queries that require TIMESTAMP with a timezone specification as the result data type are not routed to an accelerator.
- Timestamps with timezone constants are not accepted.
- Functions and special registers that are related to timestamps with timezone specifications are not accepted.

Note: If a string constant or expression is used in connection with timestamps, such as `string_col = timestamp_col`, the query is routed to IBM Db2 Analytics Accelerator regardless of the content of the string expression because the string constant or expression is not validated. This might cause a run-time error during query execution if the string column contains unsupported timestamp values or formats.

Global variables

Db2 11 for z/OS introduces the concept of global variables, which can be created, instantiated, accessed, and modified by applications. Global variables are named memory variables that you can access and modify through SQL statements. Global variables enable you to share relational data between SQL statements without the need for an application to transfer data. A global variable is associated with a specific application context, and contains a value that is unique to the scope of that context.

Global variables are not supported by IBM Db2 Analytics Accelerator. A query that contains a global variable statement is hence not routed to an accelerator.

Furthermore, the CURRENT_QUERY_ACCELERATION special register cannot be set to the value of a global variable.

Support for the array data type in SQL PL, CALL, and Java

An array consists of an ordered set of elements of a single built-in data type. Starting with Db2 11 for z/OS, Db2 offers limited support for arrays within SQL to provide more options for routines and data access code.

Arrays can be defined as parameters and variables for SQL routines. In addition, arrays can be passed from one procedure to another as arguments for the **`IN`** and **`OUT`** parameters, function parameters, or as a function return value. Db2 11 for z/OS and later versions support both, ordinary arrays and associative arrays. Ordinary arrays have a defined upper limit, while associative array do not have such a limit. When the array data type is used in a query, the query is not routed to an accelerator.

Aliases and synonyms
An alias is an alternative name for an object, such as a Db2 table, a Db2 view, or another Db2 alias. It can be used to reference an object wherever that object can be referenced directly. Synonyms are similar to aliases, but are supported only for compatibility with previous releases. Synonyms will be deprecated in a future release. Db2 11 for z/OS extends the support for aliases, so that these can be defined for sequence objects, and also provides support for *public aliases*. IBM Db2 Analytics Accelerator does not support aliases if these are used in a sequence object.

SQL grouping sets and supergroups

Important: Contrary to the statement in prior versions of this documentation, grouping sets and supergroups are currently not supported by any IBM Db2 Analytics Accelerator for z/OS Version 7 product.

Differences in floating-point error tolerances

For queries that involve floating-point aggregates, you can expect IBM Db2 Analytics Accelerator for z/OS to produce results that are different from those delivered by inhouse Db2 for z/OS processing.

Reason

You know that the same query processed twice by Db2 for z/OS can give you different results. This is due to the fact that the results depend on the order in which the rows or tuples are processed. It is not possible to determine the order that Db2 will choose beforehand.

In the case of IBM Db2 Analytics Accelerator for z/OS, another sort of result deviation occurs, which is caused by the different methods of representation for floating-point numbers. Db2 for z/OS uses hexadecimal floating-point (HFP) numbers, whereas IBM Db2 Analytics Accelerator for z/OS uses IEEE754 floating-point numbers.

When a query is routed from Db2 for z/OS to IBM Db2 Analytics Accelerator for z/OS, all processing takes place on the accelerator.

The main difference between these two methods is the base or radix of the exponential part. HFP uses a base of 16. IEEE754 uses a base of 2. The base of 16 requires the exponent of HFP numbers to be increased in steps of 4 during the calculation of an aggregate. IEEE754 numbers, on the other hand, need to be increased just in steps of 1. The multiplier in front of the exponent (mantissa) of an HFP number loses 4 bits at the right end each time the exponent is increased. The mantissa of IEEE754 numbers loses just 1 bit. The difference in error thus lies in the difference of three bits, and the amount per step depends on the values of these. If the values of the three bits are all zero, the error difference is also zero. If all values are 1, the difference in error is maximized. Because of the lower exponential increase per step, IEEE754 is the more precise method of representation.

Whether the difference in precision matters much mainly depends on the number of rows or tuples (*n*) used for floating-point aggregates. Naturally, if *n* gets higher, the result deviation or error increases, too. How high the total deviation will be is hard to quantify as it further depends on the type of the mathematical function that is used (compare, for example, a simple addition with an operation that takes the square of each number involved).

Important: In contrast to HFP numbers of the type REAL in Db2 for z/OS, which might range from -7.2E+75 to +7.2E+75, IEEE754 is only capable of processing REAL floating point numbers between -3.4E+38 and 3.4E+38. This means that it is not possible to load an accelerator-shadow table with REAL values outside the range of -3.4E+38 to 3.4E+38.

Also bear in mind that HFP and IEEE754 have different minimum and maximum values for REAL and DOUBLE numbers:

Table 16. Minimum and maximum REAL and DOUBLE values (HFP and IEEE754)

	HFP (Db2 for z/OS)		**IEEE754 (IBM Db2 Analytics Accelerator for z/OS)**	
	REAL	DOUBLE	REAL	DOUBLE
Min.				
Max.				
Min. positive				
Max. negative				

Chapter 4. IBM Db2 Analytics Accelerator Studio

You control the product from IBM Db2 Analytics Accelerator Studio, which is the administration client for IBM Db2 Analytics Accelerator.

Instead of using IBM Db2 Analytics Accelerator Studio, you can also control and operate the product by using Db2 commands and the stored procedures that come with the product.

Launching an IBM Db2 Analytics Accelerator Studio function causes the administration client to connect to your data server, which in turn calls a stored procedure that executes the corresponding command on the accelerator. The accelerator is thus always controlled and maintained indirectly via the data server.

In most cases, you do not have to change existing database applications to use IBM Db2 Analytics Accelerator. If an accelerator has been properly deployed and tables have been added to the accelerator, queries that come from database applications are usually routed automatically from your data server to the accelerator provided that this results in shorter response times.

See the following diagram:

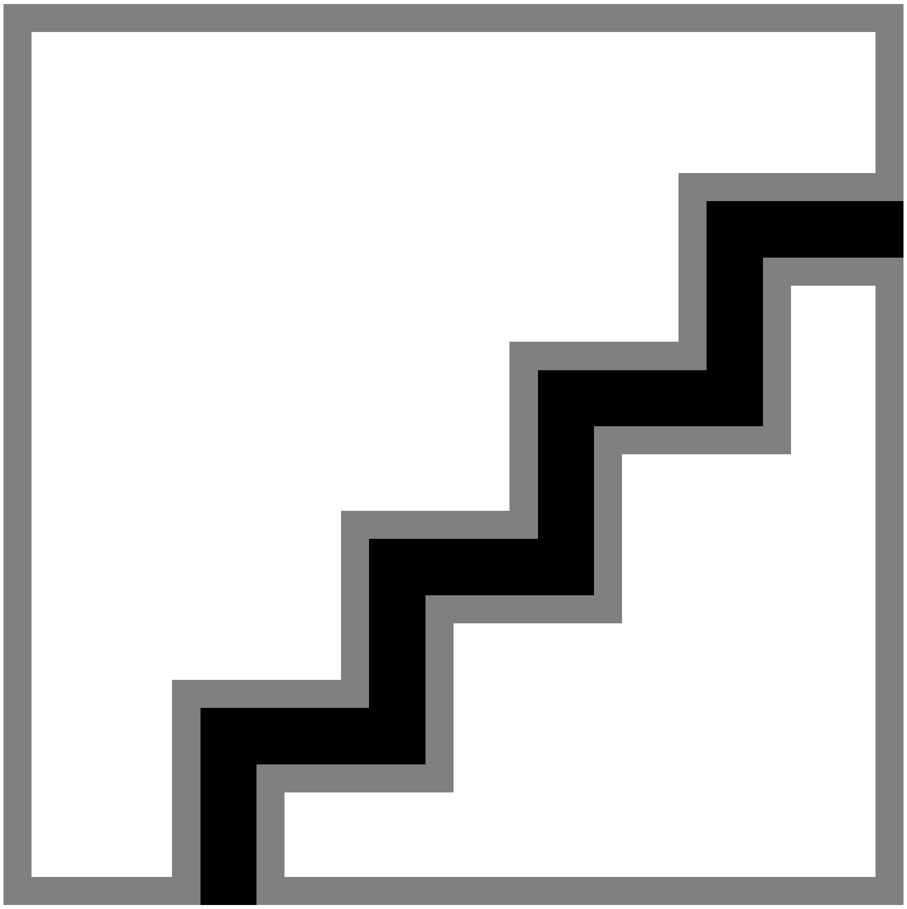


Figure 4. IBM Db2 Analytics Accelerator setup

Chapter 5. Interface elements of IBM Db2 Analytics Accelerator Studio

Here is a brief description of the interface elements that show IBM Db2 Analytics Accelerator Studio functionality.

Note: When you start IBM Db2 Analytics Accelerator Studio for the first time, you see the **Welcome** screen. To see the main window of IBM Db2 Analytics Accelerator Studio with the principal functions and controls, just close the **Welcome** screen by clicking the cross-shaped icon on the tab of the **Welcome** screen (usually positioned on the upper left).

Menu bar

When IBM Db2 Analytics Accelerator Studio has been opened, specific menus or menu choices are added to the menu bar of the Accelerator perspective. In addition, some standard menu choices of the Data perspective might have a different effect when IBM Db2 Analytics Accelerator Studio is loaded.

Edit > Select All
Selects all elements in the currently active view.

Window > Open Perspective > Other > Accelerator
Opens the Accelerator perspective, which shows IBM Db2 Analytics Accelerator Studio (if it has been installed).

Window > Reset Perspective
Restores all default views of IBM Db2 Analytics Accelerator Studio to their default size and position in the IBM Data Studio main window.

Window > Show View
Opens a submenu that allows you to reopen a currently closed IBM Db2 Analytics Accelerator Studio view.

Help > Welcome > Overview > IBM Db2 Analytics Accelerator Studio
Opens the *Introduction* topic in the online help system.

Help > Welcome > First Steps > Demonstration Videos
Opens the *Demonstration videos* topic in the online help system, which lists the available videos. Clicking a link starts the playback of a video.

Help > Welcome > Workbench
Closes the **Welcome** screen and displays the main view of IBM Db2 Analytics Accelerator Studio.

Administration Explorer

The **Administration Explorer** shows your database connection profiles and various objects like schemas, tables, and stored procedures that are defined in the database catalogs of your data servers.

The **Administration Explorer** is the navigation tree that you can see in Figure 5 on page 62. It is usually located on the left-hand side.

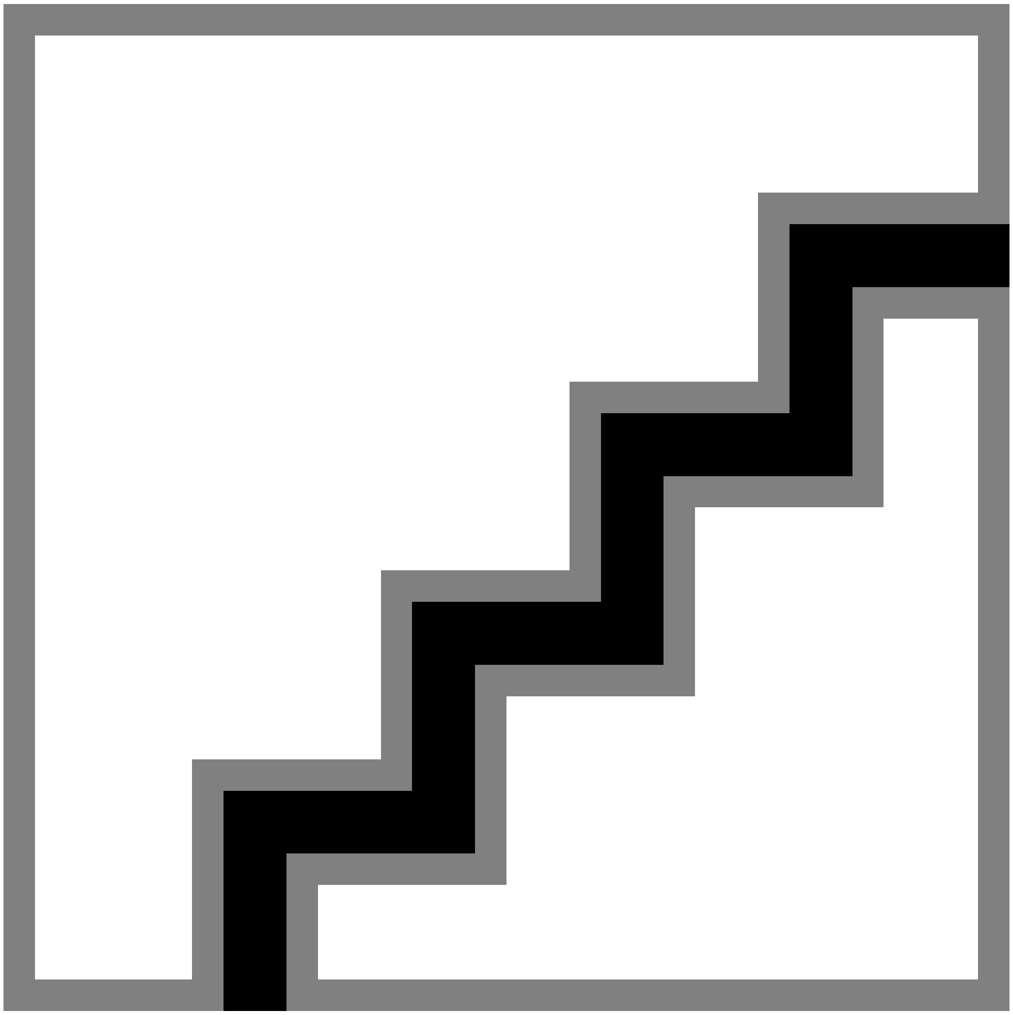


Figure 5. Administration Explorer

Menus

Depending on your selection in the **Administration Explorer**, the menu (right-click action) offers the following menu choices specific to IBM Db2 Analytics Accelerator for z/OS:

An existing database connection profile

When you right-click an existing database connection profile, you can select the following items from the menu:

Connect

Connects to the database that is referred to in the profile. You are asked for the password of the logon user if the option **Save password** is not selected in the profile.

Disconnect

Disconnects an active database connection.

Ping

Issues a **`ping`** command to check whether the network connection to the database server works.

Manage Connection

Gives you access to a submenu from which you can rename, delete, or copy (duplicate) the selected profile.

Properties

Opens the **Properties** sheet of the existing profile so that you can edit it.

Folders

When a database server has been connected to, you can see the various objects on that server, which are organized in folders. Most folders pertain to database objects that are created or configured in Db2 for z/OS. An exception is the **Accelerators** folder, which pertains to IBM Db2 Analytics Accelerator for z/OS only.

Accelerators folder

Selecting the **Accelerators** folder opens the **Object List Editor** in the main view (pane on the right).

The context menu of the **Accelerators** folder offers the following choices:

Add Accelerator
Starts the **Add Accelerator** wizard, which allows you to add an accelerator to the system configuration.

Add Virtual Accelerator
Starts the **Add Virtual Accelerator** wizard, which allows you to add a virtual accelerator to the system configuration.

Important: Virtual accelerators are a feature of earlier versions of IBM Db2 Analytics Accelerator for z/OS. They are not supported by IBM Db2 Analytics Accelerator for z/OS Version 7.

Save Trace
Opens the **Save Trace** window, in which you can select the trace information that you want to save and either save it to a file or submit it to IBM.

Refresh
Refreshes the list of accelerators in the Object List Editor (right pane).

Object List Editor

Apart from various other functions, the Object List Editor contains a table with an entry for each accelerator that has been configured. The **Object List Editor** opens when you select the **Accelerators** folder in the **Administration Explorer**.

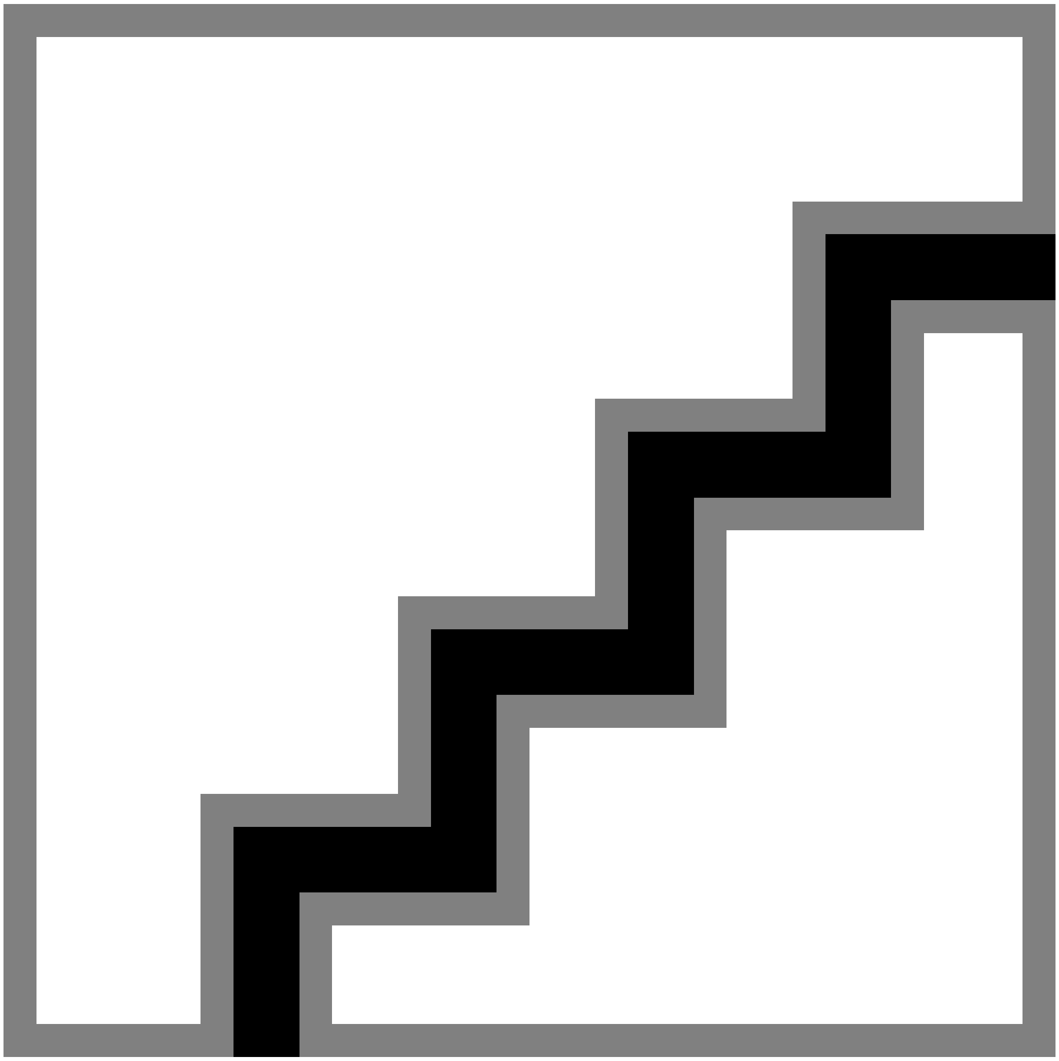


Figure 6. Object List Editor

Buttons in the Object List Editor
On top of the table in the **Object List Editor**, you find the following buttons:

The arrow pointing to the left returns to the previous display in the **Object List Editor**.

The downward-pointing arrow allows you to select a particular earlier display from a list.

To switch forward to a display in the Object List Editor that was opened later than the current one.

The downward-pointing arrow allows you to select a particular later display from a list.

To change the type of object that is currently displayed in the Object List Editor. The currently selected type is displayed to the right of the twistie (here: accelerators).

To enable the selected accelerator in Db2 for z/OS.

Notes:

- Multiple selections are not possible here. You can enable only one accelerator at a time.
- If the accelerator is attached to a Db2 data sharing group, acceleration is enabled for all members of the group.

To disable the selected accelerator in Db2 for z/OS.

The downward-pointing arrow on the right opens a menu from which you can select the following choices:

Stop
Same as clicking the button on the toolbar

Stop Force
A function that disables the selected accelerator immediately, that is, without waiting for the completion of not yet finished queries.

Notes:

- Multiple selections are not possible here. You can disable only one accelerator at a time.
- If the accelerator is attached to a Db2 data sharing group, acceleration is disabled for all members of the group.

To remove the selected accelerator from the system configuration.

Note: Multiple selections are not possible here. You can remove only one accelerator at a time.

To open the **Accelerator** view for the selected accelerator, in which you can define, remove, enable, or disable accelerator-shadow tables and invoke various other functions.

Clicking this button has the same effect as double-clicking an accelerator name in the table of the **Object List Editor**.

Note: Multiple selections are not possible here. You can only open the **Accelerator** view for one accelerator at a time.

The search field makes it easier to find particular accelerators if the list is long. Type the names of accelerators in this field, either fully or partially and press Enter. The table will display just those accelerators bearing or starting with that name.

Column information

The table in the **Object List Editor** provides various information about the accelerators, such as:

Name
Name of an accelerator

Size

Size of an accelerator, that is, its number of worker nodes (3, 6, or 12).

Acceleration

Whether acceleration has been *started* or *stopped* for the Db2 subsystem that is associated with an accelerator.

Note: The status of an accelerator cannot not been determined if acceleration has been stopped for a Db2 subsystem.

Status

Current® status or state that an accelerator is in. The following states are possible:

Cluster Is Initializing

Interim state that is shown when a new software version is applied. When this process has finished, the accelerator goes to *Fully Operational*.

Fully Operational

Ready to be enabled in Db2 for z/OS.

Online

Ready for processing.

Running

The accelerator is processing queries.

Stopping

Interim state that is shown when you click the **Stop** button or issue the **`STOP -ACCEL`** command from Db2 for z/OS. During the *Stopping* phase, already started queries are completed.

Stopped

The accelerator is currently disabled in Db2 for z/OS.

Unknown

The current status of the accelerator cannot be determined.

Used Space

Overall size of the accelerator-shadow tables on an accelerator

Active Queries

The number of queries that is currently being processed on an accelerator

Replication

Indicates whether the incremental update function is enabled for this accelerator in the DB2 subsystem. Possible values:

Disabled

The incremental update function has not been configured for this accelerator and is therefore not available.

Error

The incremental update function has been configured, but an error occurred. In most cases, this means that a component cannot be reached. Therefore, it cannot be started or controlled. The event log might provide useful information about the root cause of the error.

Started

The incremental update function is running and updates are applied to accelerator-shadow tables that have been included in the process.

Stopped

The incremental update function is available and accelerator-shadow tables have been included in the process. However, the function is not running right now.

Accelerator view

To show the tables on an accelerator, view the query history, or access various other functions, double-click an accelerator name in the **Object List Editor**, which opens the related **Accelerator** view.

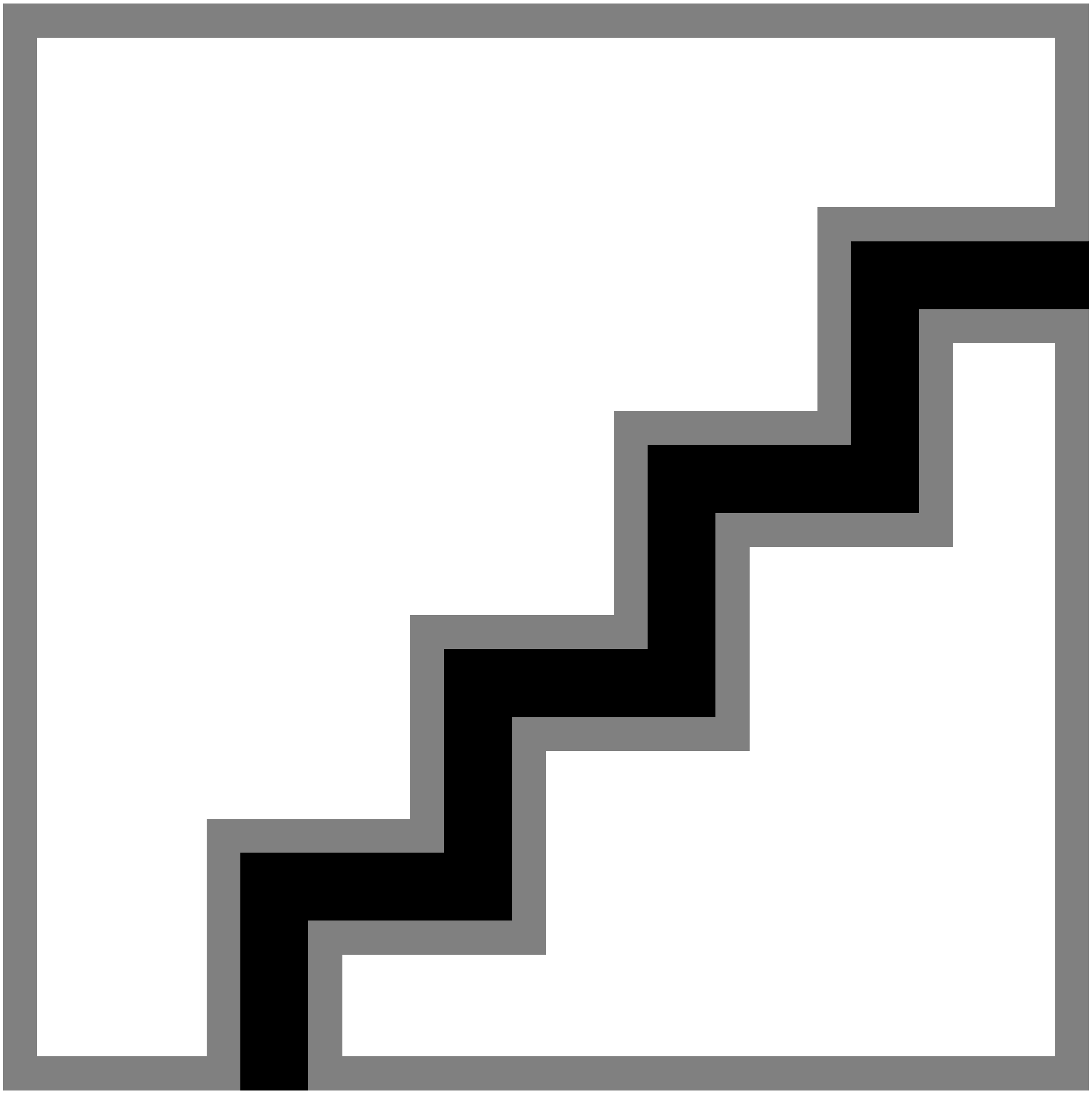


Figure 7. ***Accelerator*** *view*

Status information and functional links

The header of the **Accelerator** view contains status information and links to related functions.

Table 17. Header information

Label	Description
Acceleration	Whether the accelerator is currently enabled in Db2 for z/OS. To change the current status, click **Start** or **Stop**. If the accelerator is attached to a Db2 data sharing group, the behavior is different. The status *Started* is displayed if acceleration has been started for at least one member of the data sharing group. Other statuses are shown only if the status is the same for all members of the data sharing group. If the status of individual members is different, a warning icon is displayed. If you move the mouse pointer over the displayed status or the warning icon, a pop-up window shows the acceleration status for all members of the data sharing group. An additional functional link labeled **Change** is displayed for data sharing groups. Clicking it opens a window in which you can change the acceleration status of each member in the data sharing group.
Status	The status of the accelerator hardware. If the accelerator is attached to a Db2 data sharing group, you can move the mouse pointer over the status information to display the status of each member in the data sharing group.
Used space	The amount of disk space that is taken up by the tables, in proportion to the available space on the accelerator.
Replication	Only visible if incremental updates are configured. A status of **Started** indicates that the incremental update process is running for tables included in the process. **Stopped** indicates that the incremental update process has been suspended. To change the current status, click **Start** or **Stop**.
Credentials valid since	Shows the date on which the last authentication of the accelerator took place. To renew the authentication, click **Update**.
Encryption details	If you click the **Encyption details**, the **Encyption Details** window opens, which shows information about the use of encryption for the selected accelerator. This includes the status of hard-disk encryption, the status of encryption of data in motion, a list of encrypted IPsec tunnels (IP connections), and a list of the certificates that are used.
Trace	Shows the trace profiles for the accelerator and the stored procedures that are currently in use. To change a profile, click **Configure**. To change the settings for the saving of trace information to a file, click **Save**.
Active queries	The number of accelerated queries that are currently being processed. In parentheses, you see the number of queued queries.

Table 17. Header information (continued)

Label	Description
Replication latency	Only visible if incremental updates are configured. This is not the case in Figure 7 on page 67. The value indicates how much the incremental update function is lagging behind. Possible values: **Stopped** Incremental updates have been stopped for the entire accelerator. **Low** It will take less than three minutes to propagate recent changes to the corresponding accelerator-shadow tables. **Medium** Recent changes to the original DB2 tables will be propagated to the corresponding accelerator-shadow tables within three and six minutes. **High** It will take six minutes or longer to apply recent changes to DB2 tables to the corresponding accelerator-shadow tables. See Note on replication latency. When you move the mouse pointer over the latency value, a pop-up window shows the following information: **Target Agent** Operations completed by the incremental update function. **Inserts** Number of row inserts in accelerator-shadow tables that have been included in the incremental update process. **Deletes** Number of row deletions from accelerator-shadow tables that have been included in the incremental update process. **Remember:** • An UPDATE is a DELETE followed by an INSERT, so these two figures account for all incremental updates that have been applied. • Since an UPDATE in DB2 is a DELETE followed by an INSERT on the accelerator side, the values can be different from those in Db2 for z/OS (see **Source Agent**). Furthermore, multiple operations on a single row within a short time might be reported as just one single operation. • If a load operation is in progress and lock mode **Row** has been selected, the original DB2 tables are not locked during the load. Hence it is possible to INSERT new rows into the DB2 tables even though a load is taking place. If incremental updates are enabled for the same tables, the incremental update function stores these new rows in the spill queue until the load has finished, and applies these changes thereafter (draining the spill queue). In the course of a drain, a DELETE and an INSERT operation is carried out for each row in the affected accelerator-shadow tables. The DELETE deletes the row that was inserted by the previous load operation, and the INSERT adds this row again (applies the latest change) . This has the effect that the number **Deletes** increases in cases like this. Clicking **Show events** opens the event viewer, which keeps a record of all transactions that are related to incremental updates.

Note on replication latency: The *replication latency* is disclosed through different channels, at different points in time. Sometimes, it is calculated by different systems. Values are reported:

- In the **Accelerator** view as described in the previous table
- When you run the Db2 command `-DIS ACCEL(<name>) DETAIL`
- In trace files:

```
<Event id="61098" timestamp="2018-11-28 18:07:51" level="W" addressee="0" >
        The current replication latency of XXXXXs  on DB2 location ADB2K04 (subscription
DWE9PLVY)
        has exceeded the threshold of 3600s.
</Event>
```

- In the status monitoring counter Q8STCRL
- As a result of running the SYSPROC.ACCEL_CONTROL_ACCELERATOR stored procedure in connection with the `<getAcceleratorInfo>` parameter.

These systems or reporting channels are not synced, so deviating values are inevitable. Deviations can be considerable, depending the channel that is tapped and on the time when this is done. Stick to one reporting channel if you need to compare latency values.

Table 18. Header information (continued)

Label	Description
Replication latency	**Source Agent** Number of data changes to relevant Db2 for z/OS tables (tables for which accelerator-shadow tables have been defined and that are included in the incremental update process). **Inserts** Number of row inserts in the relevant Db2 for z/OS tables. **Updates** Number of updates to rows in the relevant Db2 for z/OS tables. **Deletes** Number of row deletions from the relevant Db2 for z/OS tables. **Latency** Repetition of the **Replication latency** value. Clicking **Show events** opens the event viewer, which keeps a record of all transactions that are related to incremental updates.

Monitoring

The **Monitoring** section provides the following information:

*Table 19. Information in the **Monitoring** section*

Label	Description
Queries	Metrics related to queries since the start of the currently viewed accelerator, divided into the following categories: **Successful** Number of queries that ran successfully **Failed** Number of queries that failed
Queue	Metrics related to queuing since the start of the currently viewed accelerator: **Avg. wait time** Average time that queries had to stay in the queue. The basis for the calculation is all queries that were processed during the last minute.

Table 19. Information in the ***Monitoring*** *section (continued)*

Label	Description
CPU Cost	CPU time that was used since the start of the currently viewed accelerator, with regard to the connected Db2 subsystem. The total CPU time is divided into the following categories: **Query execution** CPU time needed for query processing **Data maintenance** CPU time needed for load and High Performance Storage Saver (HPSS) operations **Replication** CPU time needed for incremental updates

About

The following information is displayed in the **About** section:

Table 20. Information in the ***About*** *section*

Label	Description	Information blocks	Description
z/OS	Information about components on the z/OS data server	**Stored procedures**	The currently installed version of the IBM Db2 Analytics Accelerator stored procedures
Server	Information about software components that are installed on the IBM Integrated Analytics System or as part of your Db2 Analytics Accelerator on Z	**Accelerator server**	The version of the accelerator software that is currently in use.
		Level	The current version of the database run by your accelerator.
Access Server	Only visible if incremental updates are configured. This is the version of the CDC Access Server, the central component in the incremental update process, which manages the list of subscriptions (incremental update requests) and controls the agents involved in the process.		
Replication engine	Only visible if incremental updates are configured. This is the version of the CDC Apply Agent, which propagates data changes to the IBM Integrated Analytics System, according to the instructions received from the CDC Access Server.	**Unicode sort order**	Only visible if the connected DB2 subsystem or data sharing group is a DB2 11 for z/OS database. The value reflects the sort order that was selected for tables on the accelerator. Possible values are **UTF-8 binary** and **UTF-16 binary**. The choice has an impact on how supplementary Unicode characters are sorted in query results.

*Table 20. Information in the **About** section (continued)*

Label	Description	Information blocks	Description
Client	Information about your IBM Db2 Analytics Accelerator Studio installation	**Studio**	The currently installed version of IBM Db2 Analytics Accelerator Studio. Click **Check for Updates** to look for a newer version.

Tables

The **Tables** section in the **Accelerator** view for contains buttons and a listing of the tables on the accelerator.

*Table 21. Buttons in the **Tables** section*

Button	Description
Add	Starts the **Add Tables** wizard.
Remove	Removes the selected tables from the accelerator.
Load	Opens the **Load Tables** window, in which you can select accelerator-shadow tables for loading and set the lock scope for the load process.
Acceleration	Opens a pull-down menu, from which you can enable or disable the selected accelerator-shadow tables so that these are included or excluded from query acceleration.
Replication	Only visible if incremental updates are configured. This is not the case in Figure 7 on page 67. Clicking the button opens a pull-down menu from which you can enable or disable incremental updates for selected accelerator-shadow tables.
List Tasks	Opens the **List Tasks** window, which displays the running tasks on the currently viewed accelerator, such as table load processes. To cancel a running task, select it in the list and click **Cancel Selected Task**.

Table 22. Icons and columns in the table of tables

Item	Description
	Icon representing the schema of a table
	Icon representing an accelerator-only table
	Icon representing an accelerator-shadow table
Name	Column containing the table names
Size	Column containing the table sizes
Acceleration	Column that shows whether acceleration is enabled for a table
Last Load	Column that shows the timestamp of the last successful load for each table or the current table state. Because accelerator-only tables are not loaded by IBM Db2 Analytics Accelerator functions, the column always shows the table state (for example *AOT ready*) for this type of table.

Table 22. Icons and columns in the table of tables (continued)

Item	Description
Storage Saver Partition	Column that provides information about the archived partitions of a table. The column is visible only if tables with archived partitions exist. At the schema level, the column shows how many partitions out of the total number of partitions were archived. At the table level, it shows the compressed data size of the archived partitions and the number of rows in these partitions. If archiving is in progress for a partition, the columns shows *In Progress*.
Replication Since	Column that shows the last incremental-update capture-point (a timestamp) for incrementally updated accelerator-shadow tables or *Disabled* for the tables that are not updated in that way. The column shows *Suspended* when a table load has started and incremental updates are therefore on halt, and *Refreshing* when table changes are being applied, be this through the ongoing load process or the draining of incremental updates from the spill queue.
Distribution Key	Column that shows *Random* if random distribution is used or the names of distribution-key-columns if a distribution key is used.
Skew	Column that shows the skew value for each table on the accelerator. The skew value is a metric for the distribution imbalance of table rows across the snippet-processing units (SPUs) or worker nodes. The value is the difference between the smallest and the largest data slice for a table in megabytes (MB).

Query Monitoring

The **Query Monitoring** section shows an extensive number of additional buttons, information columns, and other controls. For a description, see "Selecting and ordering query monitoring columns" on page 101.

Stored procedure call

To display or refresh the list of schemas and tables, IBM Db2 Analytics Accelerator Studio invokes the **SYSPROC.ACCEL_GET_TABLES_INFO** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Related concepts

Tables states

A table can go through various states while it is being processed by IBM Db2 Analytics Accelerator for z/OS. The meaning of these states is explained here. In addition, it is shown which operation you can run on a table in a particular state. Finally, a diagram shows the possible shifts from one state to another (state transitions).

How to select tables for query acceleration

Selecting the proper tables for a query to be accelerated is basically simple: You need to define the tables that are referenced by the query. Slightly more thought needs to be spent on the choice of proper distribution keys and organizing keys.

Related tasks

Defining tables on an accelerator

Using the **Add Tables to Accelerator** wizard, you can define accelerator-shadow tables on an accelerator.

Loading tables

Successful queries against accelerator-shadow tables are possible only if the tables contain data. Therefore, you must load the tables after their definition (empty structure) has been copied to the accelerator.

Enabling tables for query acceleration
You can permit or prevent the sending of queries to an accelerator by enabling or disabling the corresponding accelerator-shadow tables.

Specifying or changing a distribution key or organizing keys
Distribution and organizing keys have a considerable impact on the query response time.

Including or excluding tables from incremental updates
Incremental updates can be enabled or disabled on the table level. So you can individually select the tables that take part in the process. When a subsystem has been configured for incremental updates, the toolbar of the accelerator view for regular accelerators shows an extra button, which works like a drop-down menu.

Selecting and ordering query monitoring columns
You can select the columns that are displayed in the **Query Monitoring** section of the **Accelerator** view and also change their order of appearance.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Chapter 6. Tasks

See an overview of the tasks that you can complete by using IBM Db2 Analytics Accelerator Studio.

Prerequisites
Before you go about a task, check whether you meet the requirements for running IBM Db2 Analytics Accelerator Studio:

- Network connections must exist between the database server and the machine on which IBM Db2 Analytics Accelerator Studio is installed. In addition, network connections must exist between the database server and the accelerators.
- At least one accelerator (hardware) must be installed and connected to your database server.
- The appropriate Program Temporary Fixes (PTFs) must be installed so that IBM Db2 Analytics Accelerator for z/OS can connect to your database management system.
- IBM Db2 Analytics Accelerator for z/OS stored procedures must exist on the database server.
- IBM Db2 Analytics Accelerator for z/OS jobs have been run to configure and integrate the accelerator hardware and software into your database environment.

Creating a database connection profile

Create a database connection profile to gain access to a Db2 subsystem on a database server. A Db2 subsystem houses one or more databases, in which the source data for query acceleration (schemas and tables) is kept.

About this task
In IBM Db2 Analytics Accelerator Studio, the connection information is stored in profiles for reuse. Having created a profile, you can reconnect to a database by double-clicking the icon representing the profile in the **Administration Explorer**.

Procedure

1. On the header of the **Administration Explorer** on the left, click the downward-pointing arrow next to **New** and select **New Connection to a Database**.
2. In the **New Connection** window, decide how to name the database connection profile:
   - To use the name of the database server that you want to connect to, leave **Use default naming convention** selected.
   - To choose a different name, clear **Use default naming convention**, and type the name in the **Connection Name** field.
3. From the **Select a database manager** list, select **Db2 for z/OS**.
4. Make sure that in the **JDBC driver** drop-down list, **IBM Data Server Driver for JDBC and SQLJ (JDBC 4.0) Default** is selected.
5. In the **Location** field, type the name of the database server that you want to connect to.

   Tip: To determine the **Location**, **Host** name, and **Port number**, a Db2 for z/OS systems programmer or database administrator can issue a `DIS DDF` command.
6. In the **Host** field, type the host name or IP address of the data server on which the database server is located.
7. In the **Port number** field, you see that port number 446 is selected by default. Leave this setting unless the database server uses another port.
8. Select **Retrieve objects created by this user only** if you want to restrict database access to the databases, schemas, tables, and other objects that were created by the logon user.

If you do not select this option (default), IBM Db2 Analytics Accelerator Studio will show and make selectable all databases, schemas, and tables that the logon user has access to, including those to which this user might have only read access.

9. In the **User name** field, type the user ID that you want to use to log on to the database server.

   Note that you can only use IBM Db2 Analytics Accelerator Studio successfully if this user has sufficient rights to run the stored procedures behind the IBM Db2 Analytics Accelerator Studio functions. The section *Appendix C. Required access rights* in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference* lists the privileges that are required to run a particular stored procedure. If you are uncertain, use an ID with SYSADMIN authority.

   In many organizations, it is a common practice to have personal user IDs with restricted authority and special-purpose user IDs (groups in most cases) with extensive privileges in a certain field. IBM Db2 Analytics Accelerator Studio supports this practice in that you can specify a secondary user ID, which might have the privileges that your logon user ID lacks, such as the privilege to run stored procedures. If the secondary ID is a group user ID, the logon user must of course be a member of that group. To specify a secondary user ID, follow these steps:

   a. On the **Connection Parameters** page, click the **Optional** tab.

   b. In the **Property** field, type the following statement:

   ```
   currentSQLID
   ```

   c. In the **Value** field, type the secondary user ID.

   d. Click **Add**.

   e. Click the **General** tab to return to that page and complete the logon.

10. In the **Password** field, type the password belonging to the logon user ID.

11. Leave the **Save password** check box deselected.

    ⚠ **Attention:** You can select **Save password** to avoid having to enter the password each time that you want to work with the database server. This, however, is not recommended because only a lightweight encryption is applied when the password is stored on your local hard disk.

12. Leave the **Default schema** field blank.

13. Click **Test Connection** to check if you can log on to the database server.

14. Click **Finish**.

Results

After creating the profile, IBM Db2 Analytics Accelerator Studio automatically connects to the Db2 subsystem.

What to do next

If it takes too long to load all objects of the Db2 subsystem into the **Administration Explorer** (more than one minute), you can set a filter to limit the number of schemas to be loaded:

1. In the **Administration Explorer**, right-click the icon representing the Db2 subsystem (database symbol).
2. Select **Properties** from the menu.
3. In the **Properties for ...** window, select **Default Schema Filter**.
4. Clear the **Disable filter** check box. This activates the filter controls.
5. From the **Name** drop-down list, select a suitable filter mask. In the adjacent text field, type the filter string. For example, to exclude all schemas whose names starts with the characters BLU:

   a. From the **Name** drop-down list, select **Does not start with the characters**.

   b. In the text field, type BLU.

6. Click **Apply**.

Connecting to a database server

To connect to a Db2 subsystem, you just double-click the icon representing the subsystem. The icon is the result of successfully creating a database connection profile.

Before you begin

If it takes too long to load all objects of the Db2 subsystem into the **Administration Explorer** (more than one minute), you can set a filter to limit the number of schemas to be loaded:

1. In the **Administration Explorer**, right-click the icon representing the Db2 subsystem (database symbol).
2. Select **Properties** from the menu.
3. In the **Properties for ...** window, select **Default Schema Filter**.
4. Clear the **Disable filter** check box. This activates the filter controls.
5. From the **Name** drop-down list, select a suitable filter mask. In the adjacent text field, type the filter string. For example, to exclude all schemas whose names starts with the characters BLU:
   a. From the **Name** drop-down list, select **Does not start with the characters**.
   b. In the text field, type BLU.
6. Click **Apply**.

Procedure

In the **Administration Explorer**, double-click the icon representing the Db2 subsystem.

For example:

Results

When the connection was successful, the icon representing the subsystem changes and the database object types, such as table spaces or tables, are displayed in a folder hierarchy in the **Administration Explorer**.

Example

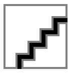


*Figure 8. Database server connection in the **Administration Explorer***

Exploring a database server

When a connection to a database server has been established successfully by double-clicking the corresponding profile icon, the **Administration Explorer** shows the database object types in a folder hierarchy. You can navigate this folder hierarchy to obtain information about the objects, for example, to find out which schemas and tables the connection gives you access to. Selecting a folder shows the corresponding objects in the **Object List Editor**.

Procedure

1. If necessary, expand the node representing the database server connection.
2. Expand the single node underneath, which represents the database server.
3. Navigate downwards in the folder hierarchy to see the objects on the database server.

Results

When IBM Db2 Analytics Accelerator for z/OS has been set up correctly, the **Administration Explorer** shows a folder called **Accelerators**. You can open this folder to display the list of configured accelerators. Double-clicking an accelerator name opens a view that shows the details of this accelerator.

Adding accelerators

Adding an accelerator to your configuration is a two-step process. First, you must obtain a pairing code from the IBM Db2 Analytics Accelerator Console. Second, you enter this paring code along with the IP address and the name of the accelerator in the **Add Accelerator** wizard of IBM Db2 Analytics Accelerator Studio.

Obtaining the pairing code for authentication

Communication between an accelerator and a Db2 subsystem requires both components to share credentials. These credentials are generated after you submit a temporarily valid pairing code. This step is required each time you add a new accelerator. The following steps describe how to obtain the pairing code.

About this task

Note: You can renew the authentication for an existing accelerator without having to use a new pairing code. To do so, click the **Update** link in the **Accelerator** view.

The steps *Obtaining the pairing code for accelerator authentication* and *Adding Accelerators* (next topic) belong together, but are seldom carried out by the same person. Since the pairing code obtained from the IBM Db2 Analytics Accelerator Console is only valid for a limited time (30 minutes by default), the persons operating the console and IBM Db2 Analytics Accelerator Studio must coordinate the steps.

To obtain the pairing code, the **`SYSPROC.ACCEL_UPDATE_CREDENTIALS`** stored procedure is run on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. Ask the network administrator or the person who did the TCP/IP setup for the IP address of the accelerator. Make a note of this information. You need to enter it as you complete the steps that follow.

   For IBM Db2 Analytics Accelerator on an IBM Integrated Analytics System, this is the virtual IP or wall IP address.

   For Db2 Analytics Accelerator on Z, this is the IP address of the network that you labeled DB2 in the **Appliance Installer**.
2. Log on to the IBM Db2 Analytics Accelerator Console by using **`telnet`** or **`ssh`**. The preferred method is **`ssh`**.

   For more information:

   - Appendix B, “Using telnet to log on to the IBM Db2 Analytics Accelerator Console,” on page 173
   - Appendix C, “Using ssh to log on to the IBM Db2 Analytics Accelerator Console,” on page 175
3. Press the Pause key, then Enter to display the following screen:

```
*****************************************************************************
*         Welcome to the IBM Db2 Analytics Accelerator Console
*****************************************************************************

You have the following options:
 (0) - (Menu) Manage Configuration Console
Users
 (1) - (Menu) Run Accelerator Functions
 (2) - (Menu) Manage Incremental Updates
 (3) - (Menu) Manage Encryption of Data in Motion
 (4) - (Menu) Manage Call Home

 ----------------------------------------------------------------------
 (x) - Exit the Configuration Console
```

4. Type 1 and press Enter to display the submenu:

```
 main -> 1
 --------------------------------------------------------
 You have the following options:

 (0) - Go back one level
 (1) - Obtain pairing code, IP address, and port
 (2) - List paired Db2 subsystems
 (3) - Clear query history
 (4) - Restart accelerator process
 (5) - Dump extensive diagnostic information
 (6) - Stop backend database
 (7) - Start backend database
 (8) - Transparently convert Db2/Z REAL columns into DOUBLE columns when loading a table
 (9) - Set the DB2 subsystem for time synchronization
 (10) - Rotate key used to generate service password
 (11) - Accelerator Workload Management
```

5. Type 1 and press Enter:
6. When the message `Specify for how long you want the pairing code to be valid.` is displayed, enter an appropriate integer to specify the validity period in minutes.

   The time that you choose must be sufficient for you or a coworker to go to the workstation that runs IBM Db2 Analytics Accelerator Studio, start the **Add New Accelerator** wizard, and enter the information that is returned by the console. Values from 5 to 1440 are allowed. If you just press Enter, you accept the default of 30 minutes.

   ```
   Press <return> to accept the default of 30 minutes.
    Cancel the process by entering 0.



    Accelerator pairing information:
      Pairing code  : 6048
      IP address    : 203.0.113.8
      Port          : 1400
      Port (AT-TLS) : 11400
      Valid for     : 30 minutes

    Press <return> to continue
   ```

   Important: A pairing code is valid for a single try only. Furthermore, the code is bound to the IP address that is displayed on the console.
7. Make a note of the following information on the **console**:
   - `Pairing code`
   - `IP address`
   - `Port` (for unencrypted network communication). Use this port if you are not sure.
   - `Port (AT-TLS)` (for encrypted network communication).

     The use of encryption requires extra configuration steps on the accelerator and on the participating z/OS L Pars.
8. Press Enter to return to the main menu of the **console**.

9. Type x and press Enter to exit the **console** and close the telnet session.

Related tasks

Completing the authentication using the Add Accelerator wizard
To complete the authentication, you specify the IP address, the port number, and the pairing code in the **Add Accelerator** wizard.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Completing the authentication using the Add Accelerator wizard

To complete the authentication, you specify the IP address, the port number, and the pairing code in the **Add Accelerator** wizard.

Before you begin

Make sure that the following conditions apply:

- You need privileges to run Db2 administration commands and stored procedures on z/OS. If you created a power user as suggested in the *Installation Guide*, the power user will have the required privileges.
- You have a valid pairing code. The pairing code, which is of temporary validity, can be obtained by using the **IBM Db2 Analytics Accelerator Console**.

⚠ **Attention:** Do not give ordinary users SELECT authorization on the SYSIBM.USERNAMES table because this allows the users to see the authentication information in the SYSIBM.USERNAMES.NEWAUTHID column.

About this task

You can renew the authentication for an existing accelerator without having to use a new pairing code. To do so, click the **Update** link in the **Accelerator** view.

⚠ **Attention:** Making a new backup of your Db2 catalog tables is strongly recommended after each authentication update because restoration processes in your Db2 subsystem can make an accelerator unusable. This happens if you must restore your Db2 catalog and the backup of the catalog was made before the last update of the accelerator credentials. In this case, the latest authentication information will not be in the catalog tables of the backup, and so the accelerator can no longer be used.

For the completion of this task, the following stored procedures are run on your data server:

- SYSPROC.ACCEL_TEST_CONNECTION
- SYSPROC.ACCEL_ADD_ACCELERATOR

For information about the privileges that are required to run these procedures and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related information** at the end of this section.

Procedure

1. Select the **Accelerators** folder in the **Administration Explorer**.
2. On the menu bar of the **Object List Editor**, click the downward-pointing arrow next to the green plus sign.
3. From the drop-down menu, select **Add Accelerator**.
4. In the **Name** field, type a name for the accelerator.

   This name is automatically copied to the **Location** field.

   The location name is the unique name of the accelerator in the SYSIBM.LOCATIONS table. Mostly, this is the same name as the accelerator name.

Restriction: An accelerator cannot be shared between two or more Db2 subsystems if the subsystems use the same location name. If you copy an entire subsystem, make sure to change the location name of the copy afterwards.

5. In the **Pairing code** field, type the pairing code.
6. In the **IP address** field, type the IP address of the accelerator.
7. In the **Port** field, type `1400`. This is the fixed port for network communication between the z/OS data server and the accelerator.
8. Click **Test Connection** to check whether the accelerator with the given address can be connected to.
9. Click **OK**.

   A connection test is carried out.

Related tasks

Obtaining the pairing code for authentication
Communication between an accelerator and a Db2 subsystem requires both components to share credentials. These credentials are generated after you submit a temporarily valid pairing code. This step is required each time you add a new accelerator. The following steps describe how to obtain the pairing code.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

DRDA connection does not work
You can ping the accelerators, but you cannot establish a distributed relational database access (DRDA) connection between your database management system and the accelerator.

Displaying accelerator networking details

To display the networking details of connected accelerators, such as the host names, IP addresses, and ports used, you can issue an SQL query.

About this task

The information can be extracted from certain Db2 catalog tables and system tables:

- `SYSACCEL.SYSACCELERATORS`
- `SYSIBM.IPNAMES`
- `SYSIBM.LOCATIONS`

Procedure

1. In the header of the **Administration Explorer** in IBM Db2 Analytics Accelerator Studio, click the downward-pointing arrow and select **New SQL script**.
2. Copy and paste the following query in the script window on the right:

```
SELECT A.ACCELERATORNAME, A.LOCATION, I.IPADDR, L.PORT
FROM SYSACCEL.SYSACCELERATORS AS A, SYSIBM.IPNAMES AS I, SYSIBM.LOCATIONS AS L
WHERE A.LOCATION = L.LOCATION AND I.LINKNAME = L.LINKNAME;
```

3. Click
4. If no connection profile is selected in the **Administration Explorer**, the **Select Connection Profile**window comes up. If so, select the proper database connection profile (database server) in the **Connections** list.
5. Click **Finish**.
6. Click the **Result1** tab in the lower right corner.
   The requested information is displayed in a table.

Example

```
ACCELERATORNAME LOCATION  IPADDR       PORT
--------------- --------  ------       ------
BC6             BC6       203.0.113.7  1277
TF12            TF12      203.0.113.8  1400
TF3             TF3       203.0.113.9  1400
```

Defining tables on an accelerator

Using the **Add Tables to Accelerator** wizard, you can define accelerator-shadow tables on an accelerator.

Before you begin

- Make sure that at least one accelerator has been configured.
- Make sure that a database connection is associated with this accelerator.
- Make sure that your resource database contains tables.
- The incremental update function requires that tables have a unique key. Therefore, you might need to add unique keys to some of your source tables. If the chosen key column is neither a primary key, nor a primary index, you must explicitly specify it as an informational unique constraint. This cannot be done in IBM Db2 Analytics Accelerator Studio. You must run the SYSPROC.ACCEL_ADD_TABLES stored procedure for this purpose.

About this task

- The maximum number of tables that you can define on a single accelerator is 27132.
- Table columns of the type CHAR and VARCHAR can be encoded in UNICODE, EBCDIC, or ASCII.

  You can add tables with differently encoded columns (EBCDIC, ASCII, and Unicode) to a single accelerator. However, a single query can only reference columns in tables that all have the same encoding.
- IBM Db2 Analytics Accelerator for z/OS does not support EDITPROC encryption. SQL encryption and decryption functions, such as DECRYPT_CHAR(), require binary table columns, which cannot be used on an accelerator. Furthermore, data that is stored in Db2 with the EDITPROC encryption tool will not be encrypted on the hard disk of the accelerator. However, the data on an accelerator is encoded with a special algorithm for efficient compression and therefore cannot be easily extracted from the media.

This task is carried out by the SYSPROC.ACCEL_ADD_TABLES stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In IBM Db2 Analytics Accelerator Studio, in the **Administration Explorer**, go to the **Accelerators** folder.
2. In the **Object List Editor**, double-click the accelerator that you want to define tables on.
3. In the **Accelerator** view, click **Add** on the toolbar, which is located above the list of tables.
4. In the **Add Tables to Accelerator** window, you see a list of all available schemas and tables under **Available tables**.

   If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.
5. Specify the tables.

   The list of selectable items is organized hierarchically. At first, you might only see the available schemas. If you click the twistie in front of a schema name, the tables belonging to the schema come into view.

You cannot select accelerator-only tables because these tables already exist on an accelerator. Entries for these tables are listed, but they cannot be selected. Apart from the different icon, you find the information *already added* in parentheses to identify the table as an accelerator-only table. Remember that accelerator-only tables are created by a `CREATE TABLE ... IN ACCELERATOR` SQL statement.

The **Unsupported Columns** column lists the unsupported columns in each Db2 table (if any). If possible, do not select tables with unsupported columns because such columns are not loaded on the accelerator. Queries referencing such columns cannot be accelerated. In most cases, a column is not supported because its data type cannot be handled by IBM Db2 Analytics Accelerator for z/OS.

Note: Detecting unsupported columns takes a long time if IBM Db2 Analytics Accelerator Studio has to connect to a database with numerous tables. To save this time, you can suppress the display of unsupported columns. To do so, go to **Window** > **Preferences** > **IBM Db2 Analytics Accelerator** > **Accelerator View** and clear the **Check for unsupported columns** check box.

- To define a single table on the selected accelerator, select its name under **Available tables**, in the **Name** column, and click to move the column to the **Tables to be added** list. To select multiple tables, hold down the CTRL key and click the names of the tables that you want to add. To move all visible tables to the **Tables to be added** list, click .
- To correct your choice and remove a table from the **Tables to be added** list, select it in the **Name** column of that list list and click to move the table back to the **Available tables** list.

  To remove all tables from the **Tables to be added** list, click .

Note: If a Db2 source table was created after IBM Db2 Analytics Accelerator Studio connected to a Db2 subsystem on the data server, you might be unable to locate this table in the list of available tables. In such a case, click .

The problem can also occur if you have defined a filter in the **Administration Explorer** or in the **Properties** section of a database connection profile. If so, proceed as follows:

a. Close the **Add Tables** window.

b. Remove the filter.

c. Reopen the **Add Tables** window.

d. Click .

6. Click **Finish**.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Determining the status of an accelerator

See how to check the status of an accelerator.

Before you begin

- A database connection profile must exist for the database that is associated with the accelerator.
- You (your user ID) must be connected to the database.

About this task

This task is carried out by the **SYSPROC.ACCEL_CONTROL_ACCELERATOR** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click an accelerator to open the **Accelerator** view.

Results

The status is displayed on top of the page that is displayed.

Related tasks

Enabling an accelerator

Queries can only be routed to an accelerator if the accelerator has been enabled in Db2 for z/OS.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Enabling an accelerator

Queries can only be routed to an accelerator if the accelerator has been enabled in Db2 for z/OS.

Before you begin

- You need a connection to a database that has an accelerator attached to it.
- You (your user ID) must be connected to the database and have the rights to control the accelerator.
- The **Acceleration** status of the accelerator must be *Stopped*.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click an accelerator to open the **Accelerator** view.
3. Click on the toolbar.

   This invokes the Db2 command **-START ACCEL** on your data server.

Results

When an accelerator has been enabled, its **Acceleration** status changes to *Started*.

Related tasks

Disabling an accelerator

You can disable an accelerator to prevent the routing of queries to this accelerator. Disabling an accelerator is recommended, for example, before you update software on the data server or on the accelerator. You can also disable an accelerator temporarily to compare response time. During the disablement, all queries are processed by Db2. You might also want to disable an accelerator for maintenance tasks. It is not necessary to disable an entire accelerator for maintaining accelerator-shadow tables; each table can be disabled individually.

Disabling an accelerator

You can disable an accelerator to prevent the routing of queries to this accelerator. Disabling an accelerator is recommended, for example, before you update software on the data server or on the accelerator. You can also disable an accelerator temporarily to compare response time. During the disablement, all queries are processed by Db2. You might also want to disable an accelerator for maintenance tasks. It is not necessary to disable an entire accelerator for maintaining accelerator-shadow tables; each table can be disabled individually.

Before you begin

- You need a connection to a database that has an accelerator attached to it.
- You (your user ID) must be connected to the database and have the rights to control the accelerator.
- The status of the accelerator must be *Online*.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click an accelerator to open the **Accelerator** view.
3. Complete one of the following steps:

   - Click on the accelerator to disable the selected accelerator without canceling running queries. The accelerator status first changes to *Stopping*, then to *Stopped*. During the *Stopping* phase, running queries are completed.
   - Click the button on the right of the button and select **Force** from the menu to disable the selected accelerator and cancel all running queries. The status of the accelerator changes to *Stopped* immediately.

   Note: Disabling does not make an accelerator unusable or remove it from the configuration. It just deactivates it. To re-enable an accelerator, click the button on the toolbar.

Enabling or disabling an accelerator for members of a data sharing group

If an accelerator is attached to a Db2 data sharing group, you can enable or disable acceleration individually for each member of the group.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click an accelerator to open the **Accelerator** view.
3. In the header of the Accelerator view, next to the **Acceleration** status, click the **Change** link.
4. The **Start / Stop Acceleration** window lists all members of the data sharing group and shows their acceleration status. To enable or disable the accelerator for members of the data sharing group, select the appropriate check boxes in front of the member names.

   If the group contains many members, you can speed up the process by using the **Select All** and **Deselect All** buttons.
5. Click one of the following buttons on top of the list:

   Start
   To enable the accelerator for the selected members

   Stop
   To disable the accelerator for the selected members

   Stop Force
   To disable the accelerator immediately, that is, without waiting for the completion of not yet finished queries.
6. Click **Close** when finished or **Cancel** to close the window without making changes.

Loading tables

Successful queries against accelerator-shadow tables are possible only if the tables contain data. Therefore, you must load the tables after their definition (empty structure) has been copied to the accelerator.

Before you begin

- A database connection profile must exist for the database that is associated with the accelerator.
- You (your user ID) must be connected to the database. Your user ID requires privileges to:
  - Control the accelerator
  - Read the data to be loaded into the accelerator
- The accelerator-shadow tables that you want to load or update must exist on the accelerator.

About this task

- Loading accelerator-shadow tables can take a long time (for example minutes or hours) depending on the amount of data in the original Db2 tables.
- An external product can be used to speed up and facilitate the load process: the Db2 Analytics Accelerator Loader, which can also load external, non-Db2 for z/OS data into accelerator-shadow tables and comes with extra load options.
- The disk throughput can be volatile, depending on the disk configuration, the placement of data sets, the number of parallel I/O operations, and so on. To a large extent, the throughput depends on the following factors:
  - Partitioning of tables in your Db2 subsystem. In Db2 for z/OS, this determines the degree of parallelism that can be used internally by the Db2 Unload Utilities.
  - Number of DATE, TIME, and TIMESTAMP columns in your original table. The conversion of the values in such columns is CPU-intensive.
  - Compression of data in Db2 for z/OS.
  - Number of available processors.
  - Workload Manager (WLM) configuration.
  - Workload on the IBM Z server.
  - Workload on the accelerators.
- During a load, IBM Db2 Analytics Accelerator automatically selects a suitable distribution key.

Restrictions:

- IBM Db2 Analytics Accelerator for z/OS does not support EDITPROC encryption. SQL encryption and decryption functions, such as DECRYPT_CHAR(), require binary table columns, which cannot be used on an accelerator. Furthermore, data that is stored in Db2 with the EDITPROC encryption tool will not be encrypted on the hard disk of the accelerator. However, the data on an accelerator is encoded with a special algorithm for efficient compression and therefore cannot be easily extracted from the media.
- During a load process, you cannot close IBM Db2 Analytics Accelerator Studio or the database connection before the loading has finished.
- The maximum number of tables on a single accelerator is 27132. Naturally, that also means that a load job cannot load more than this number of tables.
- Incremental updates are not restarted automatically if you enable these for just one table and complete the enablement steps in this order:
  1. Define the accelerator-shadow table.
  2. Enable incremental updates for this table.
  3. Load the accelerator-shadow table.

4. Start incremental updates (**Replication** > **Start**).
5. Reload the accelerator-shadow table.

Normally, when this is done for more than one table, incremental updates continue or are restarted for all tables that are not being loaded or where loading has been finished, which finally results in an automatic restart for all tables. In this special case (just one table), however, you must restart incremental updates manually after the reload.

- If you change the partitioning type of the table space of an accelerator-shadow table in Db2 for z/OS (for example, from *segmented* to *partitioned by growth*), you can no longer load the accelerator-shadow table. The operation will fail. Therefore, you must remove the table from the accelerator and redefine it after changing the partitioning type.

You can also integrate the loading and updating of accelerator-shadow tables into your scheduled ETL processes by invoking the SYSPROC.ACCEL_LOAD_TABLES stored procedure directly from a job that is run by your preferred scheduler (for example, Tivoli® workload scheduler).

Table loading is carried out by the **SYSPROC.ACCEL_LOAD_TABLES** stored procedure on your data server. In addition, the **SYSPROC.ACCEL_SET_TABLES_ACCELERATION** stored procedure is invoked to enable acceleration for the loaded table.

For information about the privileges that are required to run these procedures and further details, see *Appendix C. Required access rights* in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related information** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click the accelerator containing the tables that you want to load.
3. In the **Accelerator** view, you see a list of the tables on the accelerator. Select the accelerator-shadow tables that you want to load.

   To see the tables, you might have to expand the schema nodes first, by clicking the plus sign in front of a schema name. Selecting an entire schema will select all accelerator-shadow tables belonging to that schema for loading.
4. Click **Load**.

   The list in the **Load Tables** window contains a column labeled **Load Recommended**. The information in this column is supposed to help you with the selection. Because table loads are long-running processes, you should only load or reload an accelerator-shadow table or partition when necessary.

   Important:

   - You can only select table partitions if the tables are partitioned by ranged or partitioned by growth, and if they were already loaded in the past. In all other cases, you can only load entire tables.
   - The information in the **Load Recommended** column might be inaccurate if Db2 realtime statistics are not up-to-date (RUNSTATS). or if incremental updates have been applied to the accelerator-shadow table since the last load.

   The **Load Recommended** column contains one of the following statuses for each table or partition:

   No - data unchanged
   Table or partition data has not changed in Db2 for z/OS. Therefore, you need not load or reload the accelerator-shadow table or partition.

   Yes - changed data
   A load or reload of the accelerator-shadow table or partition is recommended because data changes have been applied to the original Db2 for z/OS tables.

   Unknown - no statistics
   A load or reload of the accelerator-shadow table or partition is recommended because the current status of the table or partition cannot be determined. It must therefore be assumed that the table is out of sync. The reason might be that access to Db2 statistics is not available or that the last successful invocation of RUNSTATS was some time ago.

Unknown - replication enabled

Data changes have been detected since the last load of the accelerator-shadow table, but the table is currently enabled for incremental updates. The data changes are propagated asynchronously by the incremental update function, but it can not be determined whether the table is already in sync. A reload is usually not required under this condition.

Yes - enforced

A partition with a value of **Yes - enforced** in the **Load Recommended** column will always be loaded if the entire accelerator-shadow table is selected for loading. A possible reason is that the partition is in *InitialLoadPending* state because it has never been loaded before.

If the list of tables and partitions is long, you can apply a filter to hide certain items from the display. To do so, select a different value from the **Show recommended** drop-down list. The following choices are available:

All (default)

Displays all accelerator-shadow tables and partitions.

All recommended

Displays only the accelerator-shadow tables and partitions for which a load or reload is recommended.

No - data unchanged

Displays only the accelerator-shadow tables and partitions that need not be loaded or reloaded.

Yes - changed data

Displays only the accelerator-shadow tables and partitions for which a load or reload is recommended because data in the original tables has changed.

Unknown - no statistics

Displays only the accelerator-shadow tables and partitions for which a load or reload is recommended because the status cannot be determined.

Yes - enforced

Displays only the accelerator-shadow tables and partitions for which a load or reload will be enforced.

5. Select the appropriate tables or partitions.

   Tick the check boxes in front of the table or partition names, or use the buttons on the right:

   Select All

   Selects all tables and partitions that are shown in the **Load Tables** window.

   Deselect All

   Deselects all tables and partitions.

   Select Recommended

   Selects all accelerator-shadow tables and partitions for which a load or reload is recommended in the **Load Recommended** column.

   Expand All

   Expands all table nodes so that the partitions come into view.

   Collapse All

   Collapses all table nodes so that the table partitions are hidden from view.

   Next Unknown

   Jumps to and selects the next accelerator-shadow table or partition with a status of **Unknown - no statistics** in the **Load Recommended** column. This might prove useful if the list is crowded.

6. Select **Lock Db2 tables while loading** if you want to protect (lock) the original tables or table partitions in the database against changes during the load operation:

   All Tables

   Protects the Db2 source tables of to-be-loaded accelerator-shadow tables against changes during the load operation.

Data availability

Newly loaded data becomes available for queries when the load job has been finished for the entire table set.

Error behavior

If an error occurs while loading one of the tables in the set, the load job for that table is aborted, and the job proceeds with the next table in the set. Message AQT20121E is issued in case of a load error.

Current Table

Protects just the Db2 source table of the accelerator-shadow tables that is currently being loaded.

Data availability

Newly loaded data in a table becomes available for queries when the load job for that table has been finished and when the entire operations has moved on to the next table in the set.

Error behavior

If an error occurs while loading one of the tables in the set, the load job for that table is aborted, and the job proceeds with the next table in the set. Messages:

- Message AQT20121E is returned if the load of all tables in the set failed.
- Message AQT10420W is issued for each single table after a failure to load that table

Current Partitions

Protects the original Db2 table-space partition of the accelerator-shadow table partition to be loaded against changes during the load. With this setting, a non-partitioned table is always locked completely.

Note: Locks at partition level are not supported for Db2 for Linux, UNIX, and Windows.

Data availability

Newly loaded data in a table partition becomes available for queries when the load job for that partition has been finished and when the entire operations has moved on to the next partition or table.

Error behavior

If an error occurs while loading a table partition, the load job for that partition is aborted, and the job proceeds with the next partition to be loaded. An error has no effect on other partitions or tables to be loaded; the job continues. Messages:

- Message AQT20121E is returned if the load of all tables or all partitions of a partitioned table in the set failed.
- Message AQT10421W is issued for each partition after a failure to load that partition.

Special cases

To avoid partially loaded tables, lock mode TABLE is applied during an initial load, even if you specified lock mode PARTITIONS. For details, see the description of lock mode TABLE.

If the reload of a partition that is enabled for incremental updates fails, incremental updates are disabled for the entire table even if the reload of the other table partitions succeeded.

NONE

No locking at all. However, only committed data is loaded into the accelerator-shadow tables because the Db2 data is unloaded with isolation level CS and SKIP LOCKED DATA.

Data availability

Data availability is the same as for lock mode PARTITIONS.

Error behavior

The error behavior is the same as for lock mode PARTITIONS.

7. The option **After the load, enable acceleration for disabled tables** ensures that you can run accelerated queries against all accelerator-shadow tables immediately after the completion of the load operation. It is selected by default. Deselect it only if there is a good reason to do so.

   The following scenario is thinkable: the table is big and one or more columns could not be loaded because their data type is not supported. These columns, however, would have held the crucial data.

In this case, it makes perfect sense to switch the option off to prevent a waste of processing resources.

8. Click **OK**.

   Note: If the load fails, first check whether DSNUTIL was started in Db2 for z/OS.

Related concepts

Updating accelerator tables continually
The incremental update function of IBM Db2 Analytics Accelerator for z/OS allows you to update accelerator-shadow tables continually. Changes to the data in original Db2 for z/OS tables are thus propagated to the corresponding target tables with a high frequency and just a brief delay. This way, query results from an accelerator are always extracted from recent, close-to-realtime data.

Distribution keys
A distribution key determines how tables rows are distributed among the available working nodes. IBM Db2 Analytics Accelerator for z/OS selects a suitable distribution key automatically when tables are loaded. The distribution varies between the two extremes: an even distribution (called random distribution) without the use of a key and a maximally skewed distribution that uses a distribution key.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter)

Enabling tables for query acceleration

You can permit or prevent the sending of queries to an accelerator by enabling or disabling the corresponding accelerator-shadow tables.

Before you begin

- You need a connection to a database that has an accelerator attached to it.
- You (your user ID) must be connected to the database and have the rights to control the accelerator.
- The tables must have been defined on the accelerator.

About this task

After its creation, an accelerator-shadow table is in *load pending* state. Query acceleration is disabled for tables in this state. The *load pending* state is also reached when an error has led to an operation failure. Also, when data has been loaded into a table, the table is not automatically enabled. Therefore, you must enable the table so that it becomes *active* and will thus be used in accelerated queries. You can disable or re-enable a loaded accelerator-shadow table by switching query acceleration for this table on or off. To enable a table, the table must be in *loaded* or *update in progress* state.

If partitions of a table have been archived by the High-Performance Storage Saver (HPSS), these partitions no longer belong to the regular accelerator-shadow table. Enabling this table for accelerated queries will thus not enable the archived partitions. To include archived partitions in accelerated queries, you must therefore use the GET_ACCEL_ARCHIVE special register in your queries or set this option as a ZPARM.

Tip: If you want to disable all tables, it is better to stop the entire accelerator. For more information, see the link under **Related tasks** at the end.

The task of enabling tables for query acceleration is carried out by the **SYSPROC.ACCEL_SET_TABLES_ACCELERATION** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

When an accelerator-shadow table has been enabled, the corresponding entry in the ENABLED column of the SYSACCEL.SYSACCELERATEDTABLES is set to ON. Although an accelerator-shadow table can exist on more than one accelerator, it can be enabled on only one accelerator. If a table is in a state that does not permit enabling, the operation ends abnormally and the enablement is canceled.

Important:

- Query acceleration can be always disabled regardless of the state an accelerator-shadow table is in. This is useful if you need to recover from an error situation in which an accelerator is down or unreachable.
- You might have to reload or even remove tables from an accelerator after the execution of an ALTER TABLE or ALTER TABLESPACE statement in Db2 for z/OS. For more information, see *What to do after changing a table or table space in Db2 for z/OS*. You find a **Related reference** link at the end of this topic.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click the accelerator containing the tables that you want to enable.
3. In the list on the lower part of the view, select the tables that you want to enable.
4. Click **Acceleration** > **Enable** on the toolbar.

Results

Queries against these tables are accelerated if the associated accelerator has been started.

Related tasks

Disabling query acceleration for tables
You can disable accelerated queries against selected tables without entirely disabling the associated accelerator.

Loading tables
Successful queries against accelerator-shadow tables are possible only if the tables contain data. Therefore, you must load the tables after their definition (empty structure) has been copied to the accelerator.

Disabling an accelerator
You can disable an accelerator to prevent the routing of queries to this accelerator. Disabling an accelerator is recommended, for example, before you update software on the data server or on the accelerator. You can also disable an accelerator temporarily to compare response time. During the disablement, all queries are processed by Db2. You might also want to disable an accelerator for maintenance tasks. It is not necessary to disable an entire accelerator for maintaining accelerator-shadow tables; each table can be disabled individually.

Related reference

What to do after changing a table or table space in Db2 for z/OS
In a number of cases, an ALTER TABLE or ALTER TABLESPACE statement in Db2 for z/OS does not involve maintenance work on your accelerator-shadow tables. There are cases, however, in which it does. See which steps you might have to take.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Disabling query acceleration for tables

You can disable accelerated queries against selected tables without entirely disabling the associated accelerator.

Before you begin

- You need a connection to a database that has an accelerator attached to it.
- You (your user ID) must be connected to the database and have the rights to control the accelerator.

- The tables that you want to disable must have been added to the accelerator.

About this task

The task of disabling tables is carried out by the **`SYSPROC.ACCEL_SET_TABLES_ACCELERATION`** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click the accelerator containing the tables that you want to disable.
3. In the list of tables in the **Accelerator** view, select the tables that you want to disable.
4. Click **Acceleration** > **Disable** on the toolbar.

Results

This disables query acceleration for the selected tables only. Other tables that are associated with the same accelerator are not affected.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Specifying or changing a distribution key or organizing keys

Distribution and organizing keys have a considerable impact on the query response time.

About this task

Distribution keys and organizing keys can influence the query response time positively as well as negatively. Therefore, make sure that you only select such keys when necessary, and that you select a proper key.

Distribution key: The default is not to use a distribution key, but to distribute the data (table rows) evenly among the processing nodes. This is what the **Random distribution without a distribution key** check box signifies. If **Random distribution without a distribution key** is selected, all table rows are distributed sequentially to the processing nodes (random distribution) so that each node receives an equal share of the workload. To distribute the data evenly is usually the best choice for small and medium-sized tables, so do not change the setting unless incoming queries demand big tables to be joined.

To execute a join, data might have to be transferred over the network, which slows down the process. In such a case, it might be better to use a distribution key. If you select a join column as the distribution key for all tables involved, you assign the rows of these tables to the same processing node. This has the effect that no data needs to be transferred to execute the join.

In general, unique key columns, such as primary keys or columns with a unique index, are a good choice for a distribution key because all values are distinct. Columns that allow NULL as a value are not a good choice if many NULL values can be expected. The reason is that all rows with a NULL in the specified column will end up on the same processing node.

Tip: If you cannot assess the distribution of values in table columns, start with random distribution. If you witness a slow performance, try using a few of the recommended columns as the distribution key. Always use as few columns as possible in a distribution key.

Organizing keys: An organizing key further enhances the performance in that it sorts table rows into blocks with equal values in the selected columns. The query engine can thus process incoming rows more quickly, as less time is required to scan the rows. Since the rows are distributed for processing in blocks, entire blocks can be skipped if the value in a particular column does not match a selection criterion specified in the query. It is not necessary to scan the entire accelerator-shadow table on disk.

Organizing keys must be columns that can be referenced in zone maps. By default, an accelerator creates zone maps for columns of the following data types:

- Integers (SMALLINT, INT, BIGINT)
- DATE
- TIMESTAMP

In addition, an accelerator creates zone maps for columns of the following data types if columns of this type are used as organizing keys:

- CHAR (all sizes, but only the first eight bytes are used in the zone map)
- VARCHAR (all sizes, but only the first eight bytes are used in the zone map)
- DECIMAL (up to 18 digits)
- REAL
- DOUBLE
- TIME

Important:

- An organizing key has no effect if the table is too small. The **Organized** column in the **Accelerator** view reflects this by not showing a value for the degree of organization (percentage).
- Do not change the keys of more than one table at a time if tables with sizes of more than 1 GB are involved.

This task of altering or changing distribution keys is carried out by the **`SYSPROC.ACCEL_ALTER_TABLES`** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. Double-click the accelerator containing the tables for which you want to specify distribution keys.
3. In the list of schemas and tables in the **Accelerator** view, select a table that contains the columns to be used as a distribution key.
4. Click **Alter Keys** on the toolbar.
5. Specify a distribution key:

   To use a distribution key instead of even distribution:

   a. Clear the **Random distribution without a distribution key** check box. This enables the controls for the definition of a distribution key.

   b. In the **Alter Distribution Keys** window, you see a list of the columns in the selected table. To specify a column as the distribution key or as part of it, select the column in the list and click the right-arrow button.

      The list on the left has a **Recommendation** column, which suggests certain columns to be used in a distribution key. Suggestions are made on the basis of a previous run of Db2 RUNSTATS. So if no recommendations can be seen, use RUNSTATS and then return to the **Alter Distribution Keys** window.

      RUNSTATS also calculates the column cardinality (**Cardinality** in the selection list). This is the number of distinct values in a column. The higher this number, the better a column is suited as a distribution key.

      Important: The RUNSTATS utility cannot collect statistics for accelerator-only tables because the data of this type of table does not reside in Db2. Hence key recommendations that rely on Db2 statistics cannot be given. You only find key recommendations for accelerator-only tables if these can be derived from the column data type.

The **Name like** filter field makes it easier to find particular columns if the list is long. Type a column name in this field, either fully or partially, to display just the columns bearing or starting with that name. The names of columns that are currently selected as keys are grayed out.

Selected columns appear in the upper list box on the right. You can remove a selected column by first selecting it in this box, and then clicking the left-arrow button.

Using the buttons with the upward-pointing and downward-pointing arrows, you can change the order of the columns in the key. The order of the columns in the list has an influence on the hash value that is calculated to determine the target processing node. To place the rows of joined tables on the same processing node, the distribution keys of all tables must yield the same hash value. It is therefore important to specify the distribution key columns for all tables in the same order.

c. Repeat this step to add further columns to the key. A maximum of four columns is allowed.

Remember: It is a best practice to use as few columns as possible in a distribution key.

- To use a distribution key instead of even distribution:

  a. Clear the **Random distribution without a distribution key** check box. This enables the controls for the definition of a distribution key.

  b. In the **Alter Distribution or Organizing Keys** window, you see a list of the columns in the selected table. To specify a column as the distribution key or as part of it, select the column in the list and click the right-arrow button.

  The list on the left has a **Recommendation** column, which suggests certain columns to be used in a distribution key or as organizing keys. Suggestions are made on the basis of a previous run of Db2 RUNSTATS. So if no recommendations can be seen, use RUNSTATS and then return to the **Alter Distribution or Organizing Keys** window.

  RUNSTATS also calculates the column cardinality (**Cardinality** in the selection list). This is the number of distinct values in a column. The higher this number, the better a column is suited as a distribution key.

  Important: The RUNSTATS utility cannot collect statistics for accelerator-only tables because the data of this type of table does not reside in Db2. Hence key recommendations that rely on Db2 statistics cannot be given. You only find key recommendations for accelerator-only tables if these can be derived from the column data type.

  The **Name like** filter field makes it easier to find particular columns if the list is long. Type a column name in this field, either fully or partially, to display just the columns bearing or starting with that name. The names of columns that are currently selected as keys are grayed out.

  Selected columns appear in the upper list box on the right. You can remove a selected column by first selecting it in this box, and then clicking the left-arrow button.

  Using the buttons with the upward-pointing and downward-pointing arrows, you can change the order of the columns in the key. The order of the columns in the list has an influence on the hash value that is calculated to determine the target processing node. To place the rows of joined tables on the same processing node, the distribution keys of all tables must yield the same hash value. It is therefore important to specify the distribution key columns for all tables in the same order.

  c. Repeat this step to add further columns to the key. A maximum of four columns is allowed.

  Remember: It is a best practice to use as few columns as possible in a distribution key.

- To specify organizing keys:

  a. Select a column to be used as an organizing key in the list on the left and click the right-arrow button next to the list box at the bottom.

  The list on the left has a **Recommendation** column, which suggests certain columns to be used in a distribution key or as organizing keys. Suggestions are made on the basis of a previous run of Db2 RUNSTATS. So if no recommendations can be seen, use RUNSTATS and then return to the **Alter Distribution or Organizing Keys** window.

Important: The RUNSTATS utility cannot collect statistics for accelerator-only tables because the data of this type of table does not reside in Db2. Hence key recommendations that rely on Db2 statistics cannot be given. You only find key recommendations for accelerator-only tables if these can be derived from the column data type.

The **Name like** filter field makes it easier to find particular columns if the list is long. Type a column name in this field, either fully or partially, to display just the columns bearing or starting with that name. The names of columns that are currently selected as keys are grayed out.

Selected columns appear in the lower list box on the right. You can remove a selected column by first selecting it in this box, and then clicking the left-arrow button.

b. Repeat this step to add further keys. A maximum of four keys is allowed.

6. Click **OK**.

Related concepts

Defining the data to load into an accelerator
Defining the data to load into an accelerator basically means selecting the proper tables. However, you also need to complete a number of related sub-tasks. This section lists all required tasks in the correct order.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Running an SQL script from IBM Db2 Analytics Accelerator Studio

To modify connected databases or create objects in these databases, you can run SQL scripts directly from IBM Db2 Analytics Accelerator Studio. This is useful because you need not change the system or console to execute SQL statements.

Before you begin

Make sure that the user running IBM Db2 Analytics Accelerator Studio has sufficient privileges on the selected databases to execute the SQL code.

About this task

If you run an accelerated query from the SQL Script Editor of IBM Data Studio, the SQL Script Editor invokes the JDBC method `Statement.setNumRows()`, which causes the JDBC driver to convert the query into a read-only query (FOR READ ONLY). This happens even if the FOR READ ONLY clause is not part of the query statement. If the clause is not part of the statement, the query might end with an error message and reason code 4 (`The query is not read-only`). The error occurs with actual queries in a production environment and EXPLAIN-only queries alike. With EXPLAIN-only queries, the error does not matter. Actual queries, however, will not return results when the error occurs. Therefore, add the FOR READ ONLY clause explicitly to the SQL statement text if you submit actual queries from the SQL Script Editor.

Procedure

1. In the **Administration Explorer**, double-click a database connection profile to connect to a Db2 subsystem on your Db2 for z/OS data server.
2. On the toolbar of the **Administration Explorer**, click the downward-pointing arrow next to the **New** button.
3. Select **New SQL Script** from the menu.
4. In the blank space of the `Script<x>.sql` workspace that opens on the upper right, type or paste the SQL statement.

   Note: <x> stands for a counting number (integer). This means that your first SQL script is named `Script1.sql`, the second `Script2.sql`, and so on.
5. Select **Script** > **Run SQL** from the main menu.

What to do next
SQL scripts can be saved in your local workspace if you created an appropriate project before, for example a *Data Development* project. For information on how to create and work with Data Development projects, refer to *Creating a data development project* in the online help of IBM Data Studio.

EXPLAIN information

After creating the necessary EXPLAIN tables, you can analyze queries by invoking the Db2 EXPLAIN function. The analysis shows whether a query can be accelerated, indicates the reason for a failure, and gives a response time estimate. The outcome of the analysis can also be visualized in an access plan graph.

Preparations

To be able to use the EXPLAIN function in order to check whether a query can be accelerated, you need the following EXPLAIN tables:

- PLAN_TABLE
- DSN_QUERYINFO_TABLE

For basic EXPLAIN information, you need just the PLAN_TABLE. To create EXPLAIN tables (including DSN_QUERYINFO_TABLE) for a specific user, customize and run the DSNTESC member of the SDSNSAMP library. Modify and run the DSNTIJOS sample job to create EXPLAIN tables that are qualified by SYSIBM and Db2OSC, as these tables are used by Db2 optimization tools like IBM Query Tuner.

Description

The EXPLAIN tables provide the following information:

PLAN_TABLE
For each accelerated query, the query blocks are pruned and a single row is inserted in the PLAN_TABLE with an ACCESSTYPE of 'A'. The query is thus represented by a single row with the value 1 in the QBLOCKNO column. For queries that are not accelerated, the PLAN_TABLE can contain multiple rows representing the tables accessed by each query block, with an ACCESSTYPE other than 'A'. If there are multiple rows belonging to the same query, these are numbered consecutively in the QBLOCKNO column.

DSN_QUERYINFO_TABLE
Additional EXPLAIN information can be found in the DSN_QUERYINFO_TABLE. See Table 23 on page 96.

Table 23. Important columns in DSN_QUERYINFO_TABLE

Column	Description
QINAME1	When the ACCESSTYPE of a query in the PLAN_TABLE is A, and the REASON_CODE in DSN_QUERYINFO_TABLE is 0, this row contains the name of the accelerator that will process the query.
QINAME2	When the ACCESSTYPE of a query in the PLAN_TABLE is A, and the REASON_CODE in DSN_QUERYINFO_TABLE is 0, this row contains the location name of the accelerator that will process the query. The location name is the unique name of the accelerator in the SYSIBM.LOCATIONS table. Mostly, this is the same name as the accelerator name.
TYPE	The type of the output for this row. A type of A means that Db2 for z/OS routes the query to an accelerator.

Table 23. Important columns in DSN_QUERYINFO_TABLE (continued)

Column	Description
REASON_CODE	The reason code for a query. When the value of TYPE is A and the reason code is 0, the query can be accelerated. If the REASON_CODE column contains a value other than 0, something prevents the acceleration of the query. To look up the other reason codes, refer to Table 24 on page 97.
QI_DATA	When the value of TYPE is A and the reason code is 0, this row contains the SQL code of the converted query that was routed to the accelerator. If a query has a reason code other than 0, this column contains a description of the reason code.

Reason codes

Table 24 on page 97 lists and describes the various reason codes. A reason code greater than 0 indicates that a query cannot be accelerated.

Table 24. Possible reason codes in DSN_QUERYINFO_TABLE

REASON_CODE	Description	Suggested solution
0	The query qualifies for routing to an accelerator.	N/A
1	An active accelerator or tables enabled for query acceleration could not be found when the statement was executed. If an accelerator is available, make sure that it is started, and that enabled tables exist on that accelerator.	Start the accelerator.
2	The query could not be routed to an accelerator because the CURRENT QUERY ACCELERATION special register is set to the value NONE.	Set the CURRENT QUERY ACCELERATION special register to one of the following values: • ALL • ELIGIBLE • ENABLE • ENABLE WITH FAILBACK
3	The query is a short-running Db2 query or query acceleration is not advantageous.	If you must accelerate the query for some reason, set the CURRENT QUERY ACCELERATION special register to one of the following values: • ALL • ELIGIBLE
4	The query is not read-only.	Ensure that the query meets the criteria for a read-only query.
6	The query uses a scrollable cursor or a remotely defined rowset-positioned cursor.	Remove the SCROLL clause or WITH ROWSET POSITIONING clause from the cursor declaration.
7	The query uses multiple encoding schemes.	Ensure that all objects to which the query refers have the same encoding scheme.

Table 24. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
8	The FROM clause of the query specifies a data-change-table-reference.	Rewrite the FROM clause of the query so that it does not include a data-change-table-reference clause.
10	The query contains a recursive reference to a common table expression.	Ensure that the FROM clause of the query does not reference a common table expression.
11	The query contains an unsupported expression. The QI_DATA column contains this expression.	Remove the unsupported expression from the query.
12	The query references a table with one of the following characteristics: • The table is not defined on the accelerator. • The table is defined on a different accelerator. • The table is not enabled for query acceleration.	Ensure that all tables that are referenced by the query are defined on the same accelerator, and that all tables are enabled for query acceleration.
13	The accelerator server that contains the tables that are referenced in the query is not started.	Start the accelerator that contains the tables that are referenced in the query.
14	A column that is referenced in the query has been altered after loading the column data into the corresponding accelerator-shadow table.	Reload the accelerator-shadow table.
15	The query uses functionality that is available only in newer Db2 for z/OS versions, but not in the versions supported by IBM Db2 Analytics Accelerator.	Ensure that the query does not use any functionality that is available only in not yet supported Db2 versions.
16	The query is not in a package.	Bind the application that contains the query into a package.
17	The query is an INSERT from SELECT statement. Subsystem parameter QUERY_ACCEL_OPTIONS does not specify option 2 to enable the acceleration of INSERT from SELECT statements.	Enable the acceleration of INSERT from SELECT statements by updating subsystem parameter QUERY_ACCEL_OPTIONS to include option 2.
18	The query uses new Db2 11 for z/OS functionality.	Try to rewrite the query so that it does not use the new (unsupported) Db2 11 for z/OS functionality.

Table 24. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
19	The accelerator is not at the correct level (that is, you are using an older version) and therefore does not support all the functions in the SQL statement. The QI_DATA column in DSN_QUERYINFO_TABLE contains the function text or expression text that has caused the failure. Update the accelerator software to enable query acceleration for the SQL statement that contains the function text or expression text.	Update the accelerator software to a service level that supports all functions in the query.
20	The query is a rowset cursor that is declared WITH RETURN, executes remotely, or executes under an SQL PL routine.	If the query executes remotely, turn off multi-row-fetch (use MULTI_FETCH=1). In the other cases, set the CURRENT QUERY ACCELERATION special register or QUERY_ACCELERATION ZPARM to the value 1 to turn off query acceleration.

22	The statement references an accelerator-only table, but cannot be routed to an accelerator. For example:
	• A MERGE statement references an accelerator-only table.
	• An UPDATE or DELETE statement references an accelerator-only table, but the target table of the UPDATE or DELETE is a normal DB2 table, for example:
	`UPDATE db2_table`
	`   SET ... (SELECT ... FROM`
	`   accelerator_only_table)`
	• An INSERT, UPDATE, or DELETE refers to a row expression with a sub-select, for example:
	`UPDATE accelerator_only_table`
	`   SET (C1,C2) = (SELECT C3,C4`
	`   FROM table_2)`
		• Remove the reference to the accelerator-only table
		• If possible, rewrite the statement so that it references a regular DB2 table instead of an accelerator-only table.

23	The SELECT INTO statement is bound for acceleration but is run as a remote SELECT INTO statement, which is not supported for acceleration.	Check if you can import (copy) the table to select from into the local Db2 subsystem (the one that is paired with the accelerator), or if you can pair the remote Db2 subsystem with the accelerator.

Table 24. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	**Description**	**Suggested solution**
24	The DDL or DML statement cannot be offloaded because the connection to the accelerator server does not allow updates. This can occur when a two-phase commit requester (for example, a Db2 for z/OS requester) connects to a Db2 for z/OS server to offload a DDL or DML statement."	Issue the DDL or DML statement directly from the Db2 for z/OS server, or from a requester which does not require a two-phase commit, for example, the IBM Data Server Driver for JDBC.
25	The statement contains a reference to a column with an unsupported data type.	See if you can remove the column reference from the query. If not, consider changing the data type to a supported type. If none of this is possible, run the query in IBM Db2 Analytics Accelerator (not accelerated).
33	The query could not be routed to an accelerator because different product versions are used, and each version would have returned a different reason code.	Request EXPLAIN output and check the REASON_CODE column in the DSN_QUERYINFO_TABLE table. For situations that result in reason code 33, two rows are returned: one that explains why the query cannot be offloaded to IBM Db2 Analytics Accelerator versions earlier than V7, and another that explains why the query cannot be offloaded to IBM Db2 Analytics Accelerator or later. In the latter case, the error description might be prefixed by (Vx) to indicate the specific accelerator version. The (Vx) prefix is displayed only when reason code 33 is returned by the original query.
34	Your query was returned to Db2 for z/OS for FAILBACK processing. However, the query contains a pass-through-only expression, which can only be handled by an accelerator. Therefore, FAILBACK processing fails.	Set the CURRENT QUERY ACCELERATION special register to ENABLE , ELIGIBLE, or ALL to run queries that contain pass-through-only expressions.
35	The query uses new Db2 12 for z/OS functionality.	Try to rewrite the query so that it does not use the new (unsupported) Db2 12 for z/OS functionality.
36	USE ONLY NEW ACCELERATOR_TYPE ZPARM is ON and No active accelerator (V7 or above) was found or table is not enabled for query offloading in the accelerator (V7 or above) when the statement was executed.	Start the accelerator (version 7 or above) or enable the table for query acceleration.
37	USE ONLY NEW ACCELERATOR_TYPE ZPARM is ON and the accelerator (V7 or above) containing the tables of the query is not started.	Start the accelerator (version 7 or above) that contains the tables that are referenced in the query.

Table 24. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
38	USE ONLY NEW ACCELERATOR_TYPE ZPARM is ON and the accelerator (V7 or above) is not the right level.	Update the accelerator (to version 7 or above) to enable the acceleration of the SQL statement that contains the function text or expression text. Make sure your QUERY_ACCEL_OPTIONS values are properly specified.

For example, reason code 7 resolves to The `query uses multiple encoding schemes.`. If you receive this reason code, check the encoding schemes that are used by the tables. If different schemes are used, you might have to curtail the query so that it only refers to columns that use the same coded character set identifier (CCSID) or code page.

Sample EXPLAIN query

```
-- clean up explain tables
DELETE FROM PLAN_TABLE WHERE QUERYNO = 3825;
DELETE FROM DSN_QUERYINFO_TABLE WHERE QUERYNO = 3825;

-- enable accelerator for this session
SET CURRENT QUERY ACCELERATION=ENABLE;

-- explain query
EXPLAIN ALL SET QUERYNO = 3825 FOR
SELECT SUM(L_QUANTITY), SUM(O_TOTALPRICE), O_CUSTKEY
FROM TPCH.LINEITEM L JOIN TPCH.ORDERS O ON L.L_ORDERKEY = O.O_ORDERKEY
WHERE O.O_ORDERDATE > '01.01.1994' GROUP BY O.O_CUSTKEY;

-- check if query is eligible for acceleration
SELECT QBLOCKNO, ACCESSTYPE FROM PLAN_TABLE AS P WHERE QUERYNO = 3825;

SELECT QINAME1 AS ACCELERATOR, REASON_CODE, QI_DATA FROM DSN_QUERYINFO_TABLE
  WHERE QUERYNO = 3825;
```

The DELETE statements at the beginning remove old information from the EXPLAIN tables.

The `SET CURRENT QUERY ACCELERATION=ENABLE` special register is required. Leaving it out would mean to run an inhouse Db2 query that is not sent to any accelerator.

Then comes the actual test query. It starts with the mandatory EXPLAIN statement. A number is assigned to the query so that the EXPLAIN information pertaining to that query can be easily extracted from the EXPLAIN tables by means of a SELECT statement.

The first SELECT statement extracts the QBLOCKNO and the ACCESSTYPE column values of the query from the PLAN_TABLE. The second SELECT statement extracts the QINAME1, REASON_CODE, and QI_DATA values of the query from DSN_QUERYINFO_TABLE.

Selecting and ordering query monitoring columns

You can select the columns that are displayed in the **Query Monitoring** section of the **Accelerator** view and also change their order of appearance.

Procedure

1. In IBM Db2 Analytics Accelerator Studio, open the **Accelerators** view.
2. Scroll down until you see the heading **Query Monitoring**.
3. Click the twistie next to the heading to reveal the contents of the section.

   The contents basically consists of a table that lists the recent query history (SQL statements and other information about recent queries). You can select various information columns, some of which are displayed by default. For a list of the available columns, click the appropriate **Related reference** link at the end of this topic.

Important: The reference for all times that are displayed is the accelerator system time. The accelerator system time is determined by the first Db2 subsystem that was added to the configuration. A warning is issued if the accelerator works with data from a Db2 subsystem that was not the first to be added. It is very important to keep this in mind if the Db2 subsystems (data servers) are situated in different time zones.

4. To add columns, remove columns, or change their order, click .
5. In the **Adjust Query Monitoring Table** window, you see the following lists:

   Available columns
   The names of available columns that are currently not selected for display

   Shown columns
   The names of the columns that are currently displayed. The order from top to bottom indicates their appearance in the **Accelerator** view from left to write.

   - To add a single column for display in the **Accelerator** view, select it in the **Available columns** list and click to move the column to the **Shown columns** list.

     To move all available columns to the **Shown columns** list, click .
   - To hide a single column from the display in the **Accelerator** view, select it in the **Shown columns** list and click to move the column to the **Available columns** list.

     To remove all columns from the display, click .
   - To change the order of appearance in the **Accelerator** view, select a column in the **Shown columns** list and click or to move the column up or down.

     The order of the columns from top to bottom represents the order in which these columns appear from left to right in the **Accelerator** view.
   - To restore the default settings for the **Query Monitoring** section in the **Accelerator** view, click **Restore Defaults**.
6. Click **OK**.

   Your settings are applied to the table in the **Query Monitoring** section, that is, columns are added, removed or their order of appearance is changed.
7. To limit the number of queries that are displayed in the **Query Monitoring** table:

   a. Show all recent queries, just the queries that are currently being processed, or just the completed queries, by selecting the appropriate value from the **View** drop-down list:

      All Queries
      To show all recent queries

      Active Queries
      To show just the queries that are currently being processed

      Query History
      To show just the completed queries

   b. To limit the displayed queries on the basis of their values in one of the columns of the **Query Monitoring** table:

1) Change the default (**All**) in the first drop-down list to the right of the **Show** label to one of the restricting values (**Highest 5**, **Highest 10** and so on).
2) From the other drop-down list further to the right, select the column in the **Query Monitoring** table that the setting in the previous step refers to.
3) Click **Run**.

This limits the number of displayed queries to those with the highest value in the selected column. If you select a column containing time values, this means that the display will be restricted to the slowest queries. The designers of the product have opted for this order because the slowest queries usually have the highest potential for query optimizing, and are most probably the ones that a database administrator wants to touch again after they have been run.

8. The **Query Monitoring** section offers a few additional functions:

- To show further details about a selected query or to rerun the query, click:

  Show SQL
  To view the SQL of the query

  Show Plan
  To view the access plan graph of the query

  Note: For accelerator-only tables, the **Show SQL** and **Show Plan** functions show information about data manipulation (DML) jobs, such as INSERT, UPDATE, or DELETE, but not data definition (DDL) jobs, such as CREATE or DROP.

  A drop-down button that leads to the following functions:

  Show Plan
  Same as the button described before

  Show Explain Details
  Opens a window with a textual representation of a query's access plan and other information about the query execution. As an example, see the following extract:

```
...
FROM
  KLMDWEDB14."CUSTOMER-UID_01140003-V0" AS Q1
WHERE
  (Q1."DWA_Partition_ID (hidden)" <= 100001)

Access Plan:
-----------
    Total Cost:         105.849
    Query Degree:        8


           Rows
          RETURN
          (   1)
           Cost
            I/O
            |
          333.333
          LTQ
          (   2)
          105.849
            15
            |
          333.333
          CTQ
          (   3)
          105.778
            15
            |
          333.333
          TBSCAN
          (   4)
          105.766
            15
            |
           1000
   CO-TABLE: KLMDWEDB14
 CUSTOMER-UID_01140003-V0
            Q1


Operator Symbols :
------------------

   Symbol      Description
   ---------   ------------------------------------------
    ATQ      : Asynchrony

...
```

Cancel
: To cancel a running query or an INSERT, UPDATE, or DELETE operation on an accelerator-only table.

- To limit the number of queries that are displayed in the **Query Monitoring** table.

  a. Show all recent queries, just the queries that are currently being processed, or just the completed queries, by selecting the appropriate value from the **View** drop-down list:

     All Queries
     To show all recent queries

     Active Queries
     To show just the queries that are currently being processed

     Query History
     To show just the completed queries

  b. To limit the displayed queries on the basis of their values in one of the columns of the **Query Monitoring** table:

     1) Change the default (**All**) in the first drop-down list to the right of the **Show** label to one of the restricting values (**Top 5**, **Top 10** and so on).
     2) From the other drop-down list further to the right, select the column in the **Query Monitoring** table that the setting in the previous step refers to.
     3) Click **Run**.

     This limits the number of displayed queries to those with the highest value in the selected column. If you select a column containing time values, this means that the display will be restricted to the slowest queries. The designers of the product have opted for this order because the slowest queries usually have the highest potential for query optimizing, and are most probably the ones that a database administrator wants to touch again after they have been run.

Related reference

Selectable columns in the Query Monitoring section
The list in this section briefly describes all columns that can be selected for display in the **Query Monitoring** section of an **Accelerator** view.

Related information

Query history is not displayed - error message returned
The query history is not displayed in the **Accelerator** view and a corresponding error message is returned.

Selectable columns in the Query Monitoring section

The list in this section briefly describes all columns that can be selected for display in the **Query Monitoring** section of an **Accelerator** view.

Accounting String
: Accounting information that can be provided by an application programming interface (API) on the client side, such as the `setDb2ClientAccountingInformation` API of a connecting JDBC client.

For more information, see Providing extended client information to the data source with IBM Data Server Driver for JDBC and SQLJ-only methods

Client User ID

The ID of a connecting client user. This ID can be provided by an application programming interface (API) on the client side, such as the `setDb2ClientUser` API of a connecting JDBC client.

For more information, see Providing extended client information to the data source with IBM Data Server Driver for JDBC and SQLJ-only methods

Client Workstation Name

The name or ID of a connecting client workstation. This ID can be provided by an application programming interface (API) on the client side, such as the `setDb2ClientWorkstation` API of a connecting JDBC client.

For more information, see Providing extended client information to the data source with IBM Data Server Driver for JDBC and SQLJ-only methods

Collection ID

The ID or name that identifies a group of static SQL packages. If a collection ID is displayed, this means that the static SQL package containing the query belongs to that group.

Connection Name

The connection name. Possible values are:

For batch:
BATCH

For TSO:
TSO

For QMF:
Db2CALL

For Db2 utilities:
UTILITY

For Db2 private protocol:
The Db2 subsystem ID

For IMS:
The IMS ID

For CICS®:
The CICS ID

For DRDA connections from non-Db2 requesters:
SERVER

For more information, see IFCID 147 - Thread Correlation Data.

Connection Type

The type of the Db2 connection. Valid values are:

APPLCTN
Application directed access

BATCH
Batch job

CICS
CICS attach

Db2CAF
Db2 call attach

DLIBATCH
DL/I batch

IMS
IMS thread

IMS BMP
IMS attach BMP

IMS TBMP
IMS transaction BMP

IMS CTL
IMS control region

IMS MPP
IMS attach MPP

SYSTEM
System directed access

TSO/E
TSO/E foreground and background

UNKNOWN
Unrecognizable connection type

UTILITY
Db2 utility

For more information, see Thread SQL Counts Attributes.

Correlation ID
The correlation ID value. Possible values are:

For BATCH:
Job name.

For TSO:
Logon ID.

For IMS/VS:
PST#.PSBNAME

For CICS:
CONNECTION_TYPE.THREAD_TYPE.THREAD_#.TRAN-ID

For RRSAF:
CORRELATION-ID value from SIGNON function.

For threads using the Db2 private protocol or DRDA from a Db2 requester:
This field contains the correlation-ID name of the thread at the requesting location.

For threads using DRDA from a non-Db2 requester:
This field contains the first 12 characters in the DDM external name (EXTNAM) parameter of the DDM EXCSAT command received as part of the SQL connect.

For more information, see IFCID 147 - Thread Correlation Data

CPU Time
The overall CPU time that was needed by an accelerator to process a query or data manipulation (DML) statement. This value reflects parallel processing such that the CPU value might exceed the accumulated elapsed time. This is an indicator for the amount of used resources.

For completed statements, this is the CPU time from the initial request to the last row being returned to Db2. In case of DML statements, this is measured until the final result (SQLCA) is sent to Db2.

For in-process statements, this is the time from the initial request to the current point in time. The counter includes the CPU time that was used by the product and also the CPU time spent on the accelerator (on the coordinator node and all worker nodes). Because parallel processing is the standard, the CPU time cannot be related to the elapsed time. If query processing has already finished on the accelerator, the CPU time spent inside this system is included. If the SQL statement is still running on the accelerator, only the CPU time that was used by IBM Db2 Analytics Accelerator is available.

Data Sharing Group Name

The name of the Db2 data sharing group (if the accelerator is connected to such a group rather than a Db2 subsystem)

Data Sharing Member Name

If the accelerator is connected to a Db2 data sharing group, the name of the group member that sent the query.

Db2 Subsystem ID

The Db2 subsystem ID.

Elapsed Time

The total time that was needed to process a query (from the state QUEUED to FAILED, SUCCESSFUL, or CANCELED).

The following diagram shows the beginning, duration, and the end of each time in the **Query Monitoring** section on a time scale. This way, you can better relate the values to each other. It starts with the **Start Time** (point in time, where the diagram starts and when the accelerator receives the query). The following interval, the **Wait Time** overlaps with the **Execution Time** because the **Execution Time** includes the period that a query stays in the accelerator's execution queue. The **Execution Time** overlaps with the following interval, the **Fetch Time**, because Db2 starts fetching results when the first result rows have been returned by the accelerator, that is, at a time when the entire query has not necessarily finished processing. Finally, the diagram shows the **Elapsed Time**, which includes all intervals mentioned before because it lasts from the beginning of the **Start Time** to the time when the query controller stops processing the query.

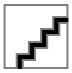


Figure 9. Wait time, execution time, fetch time, and elapsed time in comparison

Error Description

The error description or error token that is returned with the **SQL Code**. The description might be truncated.

Execution time

The time that a query spent in the queue before it was sent to the accelerator plus the time needed to process the query on the IBM Integrated Analytics System.

Fetch Time

The time interval starting with the first retrieval of a result row by Db2 and ending with the last. In most cases, query execution has already finished when the result retrieval process is still ongoing.

Location Name

Location name of the Db2 subsystem sending the query.

LU Name

Network LU (logical unit) name of the requester sending the query. This is used in the end-to-end monitoring of accelerated queries in multi-tier architectures where Db2 for z/OS handles requests passed on by an application server on behalf of a client application. This allows you to identify queries that were issued by a certain LU Name.

Network ID

Network ID of the requester sending the query. This is used in the end-to-end monitoring of accelerated queries in multi-tier architectures where Db2 for z/OS handles requests passed on by an application server on behalf of a client application. This allows you to identify queries that were issued by a certain Network ID.

Original Authorization ID

The original authorization ID. Possible values are:

For TSO:

The logon ID.

For batch:

The user ID on the job statement.

For IMS (message-driven regions):
The sign-on ID, LTERM, ASXBUSR, or PSB name.

For IMS (control regions):
The user ID on the job statement, or the RACF® started procedure entry if RACF is used.

For CICS:
The user ID, TERM ID, TRAN ID, or as specified in the resource control table.

For MVS operator commands and Db2 system internal agents:
SYSOPR

For a distributed application server (AS):

- If the application requester (AR) is a Db2 system, then this is the same value that was assigned at the AR.
- If the application requester is not a Db2 system, then this is the user ID used to make the initial connection with the application server.

For more information, see IFCID 147 - Thread Correlation Data

Package Name
The name that identifies a static SQL package or application package. If such a name is displayed, this means that the query was submitted by an application program in the form of a bound package for static SQL processing.

Plan Name
The plan name. It is blank for a Db2 command thread; otherwise:

DSNESPRR
For SPUFI with repeatable read

DSNESPCS
For SPUFI with cursor stability

DSNUTIL
For Db2 utilities

DSNTEP2
For DSNTEP2

DSNBIND
For binding

The application plan name
For IMS or CICS

A blank plan name
For IMS and CICS commands

DSQPLAN
For QMF

The first 8 bytes of the application name
For DRDA connections to the common servers

For more information, see IFCID 147 - Thread Correlation Data.

Priority
The Workload Manager (WLM) importance level that the query or job holds in Db2.

Product ID
ID of the product (application requester) that routed the query, such as a JDBC driver.

Result Size
The data size of the rows that were returned in query results (in MB or GB) from the accelerator to Db2 for z/OS.

Rows Returned
The number of rows that were returned by the accelerator as query results.

SQL Code
The SQL code that is returned to the user after the query or job execution.

SQL Text
The SQL statements of recently submitted queries.

Start Time
Times when queries were received by IBM Db2 Analytics Accelerator.

State
Processing states that queries are in. The following states are possible:

UNKNOWN
The state cannot be determined.

QUEUED
A query has been passed to the accelerator for processing.

RUNNING
The query is being processed by the accelerator.

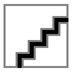
FETCHING
The accelerator has finished processing the query, but Db2 for z/OS has not yet finished collecting the results.

FAILED
The query did not run to completion because one or more errors occurred.

SUCCESSFUL
Query execution has been completed successfully.

CANCELED
The query did not run to completion because it was canceled by a user.

With the exception of the UNKNOWN state, each query passes through all the states, unless an error leads to an abnormal ending or a user cancels the query. This is shown in the following diagram:



User ID
The names of users who submitted past queries.

Wait Time
The times that queries had to “spend” in the queue before they were processed by the IBM Integrated Analytics System.

Canceling tasks

See how to cancel tasks that are in progress on an accelerator, such as load operations.

About this task

Using the **List Tasks** function, you cannot cancel queries. To cancel a running query or an INSERT, UPDATE, or DELETE operation on an accelerator-only table, use the **Cancel** button in the **Query Monitoring** section of the **Accelerator** view.

Important: When a load job fails, and you run the command to check the status of the Db2 UNLOAD utility in Db2 for z/OS, you might find numerous UNLOAD threads in the STOPPED state. The stopped threads are ended automatically when you start another load job.

Procedure

1. Open the accelerator view of the accelerator that the task is running on.
2. Click **List Tasks** on the toolbar.

3. The table in the **List Tasks** window shows all tasks that are currently running on the accelerator. To cancel a task, select it in the table and click **Cancel Selected Task**.

   The table that lists the tasks contains the following columns, which give you further details about a task:

   Name
   The type of the task, for example **Load Tables**.

   Execution Time
   The time that passed since the start of the task.

   Progress
   The degree of completeness since the start of the task.

   If individual tables were selected for processing, the number denotes the number of records (table rows) that were processed so far.

   If an entire schema was selected for processing, the number represents the percentage of completion. The maximum value is 1, denoting 100 percent. Therefore, a value of 0.25 means 25 percent complete.

   Progress Message
   Additional information about running tasks.

   User
   The IDs of the users who started the tasks.
4. Click **Close** to close the **List Tasks** window.

Removing tables from an accelerator

Follow the steps in this topic to remove accelerator-shadow tables from an accelerator.

About this task

If a part or all of a table (all partitions) have been archived by the High-Performance Storage Saver (HPSS), then this table (an accelerator-archive table) cannot be removed without first restoring the archived data. After restoring the data, the table remains on the accelerator, but becomes and ordinary accelerator-shadow table, which means that it can be removed.

The table removal task is carried out by the **`SYSPROC.ACCEL_REMOVE_TABLES`** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. Double-click the accelerator containing the tables that you want to remove.
3. Select the tables that you want to remove.
4. Click **Remove** on the toolbar.

Related tasks

Including or excluding tables from incremental updates
Incremental updates can be enabled or disabled on the table level. So you can individually select the tables that take part in the process. When a subsystem has been configured for incremental updates, the toolbar of the accelerator view for regular accelerators shows an extra button, which works like a drop-down menu.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Exporting a table specification

You can save a list of the accelerator-shadow tables on an accelerator to an XML file and in a later step import that list for use with another accelerator. This way, tables can be added more quickly than by adding schemas and tables individually.

About this task

A typical use case is an administrator who saves a valid table specification that was created in a test environment in order to use this specification in a production environment. To make the definition work in a different environment, you can edit the XML file with a text editor, and, for example, use the find-and-replace function to replace the names of schemas and tables.

Procedure

1. Click **File** > **Export**.
2. In the **Export** window, expand the **IBM Db2 Analytics Accelerator** folder to see its contents.
3. Select **Table Specifications**.
4. Click **Next**.
5. In the **Export Table Specification** window, select an appropriate database connection profile.

   The profile gives you access to the accelerators that a Db2 subsystem is connected to. Tables are exported from these accelerators. You thus have to select the proper profile, that is, one that gives you, by way of the connected Db2 subsystem, access to the accelerators that contain the right tables. If the proper profile does not exist yet, you can click **New** to create it.
6. Click **Next**.
7. From the **From accelerator** drop-down list, select the accelerator containing the tables to be exported.
8. In the table below, you can see the schemas and tables on the selected accelerator. From this table, you can select the tables or entire schemas whose specification you want to save to an XML file.

   If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.
9. Click **Browse**.
10. In the **Export to File** window, navigate to the folder in which you want to save your table specification XML file.
11. Type a file name in the **File name** field.

    By default, the file is given an extension of `idaa.xml`. Files of this type are encoded in UTF-8.
12. Click **Save**.

    You return to the **Export Table Specification** window. The path and the file name of your table specification XML file are displayed in the **To XML file** field.
13. Click **Finish**.

Importing a table specification

You can import a list of accelerator-shadow tables for query acceleration that was previously saved to an XML file.

About this task

A typical use case is an administrator who saves a valid table specification that was created in a test environment in order to use in a production environment. To make the definition work in a different environment, you can edit the XML file with a text editor, and, for example, use the find-and-replace function to replace the names of schemas and tables.

Procedure

1. Click **File** > **Import**.
2. In the **Import** window, expand the **IBM Db2 Analytics Accelerator** folder to see its contents.
3. Select **Table Specifications**.
4. Click **Next**.
5. In the In the **Import Table Specification** window, select an appropriate database connection profile.

   The profile gives you access to the accelerators that a Db2 subsystem is connected to. Tables are exported from these accelerators. You thus have to select the proper profile, that is, one that gives you, by way of the connected Db2 subsystem, access to the accelerators that contain the right tables. If the proper profile does not exist yet, you can click **New** to create it.
6. Click **Next**.
7. From the **To accelerator** drop-down list, select the accelerator on which you want to define the tables.
8. Click **Browse** to navigate to the folder that contains the XML file with the list of accelerator-shadow tables.
9. In the **Import from File** window, navigate to the proper XML file, select it, and click **Open**.

   Important: XML files in UTF-8 format are recommended. Such files start with the following header:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   ```

   You return to the **Import Table Specification** window. The path and the file name of the selected XML file is displayed in the **From XML** file field.
10. In the table below, you can see the schemas and tables that are listed in the selected XML file. From this list, select entire schemas or individual tables that you want to define on the accelerator.

    If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.
11. Click **Finish**.

Removing accelerators

Follow the steps in this topic to remove accelerators from the system configuration. Bear in mind that this action will also remove the tables from an accelerator.

About this task

The table removal task is carried out by the following stored procedures on your data server:

- **SYSPROC.ACCEL_REMOVE_TABLES**
- **SYSPROC.ACCEL_REMOVE_ACCELERATOR**

For information about the privileges that are required to run these procedures and further details, see *Appendix C. Required access rights* in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related information** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor** on the right, select the accelerator that you want to remove.
3. Click on the toolbar.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Tracing

IBM Db2 Analytics Accelerator offers a variety of predefined trace profiles in case you need diagnostic information. These profiles determine the trace detail level and the components or events that will generate trace information. After collecting the information, you can save it to a file. If you have access to the Internet, you can directly send the information to IBM from IBM Db2 Analytics Accelerator Studio using the built-in FTP function. This method automatically adds the information to an existing IBM problem management record (PMR).

Configuring the trace behavior

IBM Db2 Analytics Accelerator provides trace profiles. These profiles consist of lists of components and events that result in the collection of trace information. In addition, they determine the verbosity level of the information, that is, how detailed the trace information will be. Profiles are available for accelerators as well as for stored procedures. You can also use custom profiles that have been saved to an XML file.

About this task

When a traceable event was caused by a component that is covered by the selected profile, trace information about this event is collected and stored in a temporary file.

You can save the temporary trace information to a permanent file or even add the information to a problem management record (PMR). Where to save the collected information is configured in another dialog, the **Save Trace** window.

Recommendation: Tracing can have a considerable impact on the system performance. The more verbose the information, the bigger the (negative) impact. Therefore, use the default profiles unless you are instructed otherwise by IBM support.

If none of the available profiles covers your problem scenario, you can ask IBM support for an appropriate profile. If you open a PMR, IBM support will give you detailed instructions on how to configure the trace behavior.

Two sets of profiles are available. The first set captures accelerator events; the second events that were caused by stored procedures. The scope of these profile sets is different. Accelerator trace profiles are saved on the selected accelerator. When active, they capture any event on this accelerator, no matter if incidents were caused by an (external) application program or by a function of IBM Db2 Analytics Accelerator Studio. In contrast, stored procedure trace profiles are a part of IBM Db2 Analytics Accelerator Studio. This means that you can only capture events with the help of these profiles if a stored procedure is called by a function of IBM Db2 Analytics Accelerator Studio. Stored procedures that are called from external applications can thus not be traced. Furthermore, the temporary file, which holds the collected trace information about stored procedures, is deleted when you close IBM Db2 Analytics Accelerator Studio. Therefore, to retain access to this information, specify a permanent trace file in the **Save Trace** window.

Using the **Configure Trace** function invokes the **SYSPROC.ACCEL_CONTROL_ACCELERATOR** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. Unless you are already connected to a database, double-click the database-connection-profile icon for the Db2 subsystem that is associated with the accelerator that you want to trace.
2. In the **Administration Explorer**, select the **Accelerators** folder.
3. In the **Object List Editor**, double-click an accelerator to open the **Accelerator** view.
4. In the header section at the top of the **Accelerator** view, click **Configure**.

5. Under the heading **Accelerator**, from the **Profile** drop-down list, select a profile for accelerator tracing.

   By default, the currently active trace profile is selected. You can change this setting by selecting one of the other standard profiles that are delivered with IBM Db2 Analytics Accelerator or by selecting a custom profile. A custom profile must be imported before you can select it from the **Profile** drop-down list. See step "6" on page 114.

   About the standard profiles, you find additional information in the **Description** field. The text in this field gives you details about the events that are covered by the profile.
6. To select a custom profile that has been saved to an XML file:
   a) Click **Import**

   b) Navigate to the XML file in your file system.

   c) Select the file and click **OK**.

   d) Select the custom profile from the **Profile** drop-down list.
7. Under the heading **Stored Procedures**, from the second **Profile** drop-down list, select a profile for stored-procedure tracing.

   By default, the currently active trace profile is selected. You can change this setting by selecting one of the other standard profiles that are delivered with IBM Db2 Analytics Accelerator or a custom profile. For importing a custom profile, see step "6" on page 114.
8. To discard your individual settings and return to the default profiles for both, accelerator tracing and stored-procedure tracing (**DEFAULT** and **Off**), click **Restore Defaults**.
9. Click **OK** to save your settings.

Results

The selected profiles remain active until other profiles are selected or until they are reset to the default by the **Reset configuration profiles to default after collecting traces** option, which is selectable in the **Save Trace** window.

Related tasks

Saving trace information

The **Save Trace** function saves the collected trace information. In the **Save Trace** window, which opens before the action is completed, you can specify or change the settings for the save operation.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Saving trace information

The **Save Trace** function saves the collected trace information. In the **Save Trace** window, which opens before the action is completed, you can specify or change the settings for the save operation.

About this task

You can specify or change the location of the trace file on your local system. The **Save Trace** window also allows you to include or exclude certain information from the bulk of information that was collected as a result of using a particular profile. You can also choose to add the Eclipse error log to the trace file, since the information in the Eclipse error log is not generated by means of an IBM Db2 Analytics Accelerator trace profile. Moreover, you can send trace information to IBM using the file transfer protocol (FTP) to the effect that the trace information is added to an existing problem management record (PMR).

Using the **Save Trace** function invokes the **SYSPROC.ACCEL_CONTROL_ACCELERATOR** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the **Object List Editor**, double-click an accelerator to open the **Accelerator** view.
3. In the header section at the top of the **Accelerator** view, click **Save**.
4. Select the information that you want to store in the trace file.

   Accelerator

   Log and trace information produced by components running inside Docker container (mainly IBM Db2 Analytics Accelerator and SQL engine). This includes, for example: accelerator logs, SQL engine logs, system logs of the Docker container.

   Note: When you open the **Save Trace** window from the context menu of the Accelerators folder rather than from the header of the **Accelerator** view, the **Accelerator** check box is inactive. The reason is that selecting the entire Accelerators folder does not allow you to specify or select the particular accelerator in whose trace information you are interested. The **Save Trace** menu choice on the context menu of the Accelerators folder is a kind of "last rescue" option. Without it, there would be no way to submit trace information to IBM in case an error prevented the proper listing of the accelerators in the right pane (accelerators view). If such an error occurs, you have no access to the **Save Trace** function for individual accelerators.

   Appliance

   Trace information from the physical system in which IBM Db2 Analytics Accelerator and the SQL engine run (IBM Integrated Analytics System or the Secure Service Container LPAR for Db2 Analytics Accelerator on Z)

   Stored procedure trace information

   Trace information that is collected while running stored procedures of IBM Db2 Analytics Accelerator for z/OS, as the result of using a particular profile for stored-procedure tracing in the **Configure Trace** window. Using the check box, you can include or exclude this information from the trace file to be saved.

   Eclipse error log

   Error information that is collected by the Eclipse framework, on which the foundation for IBM Db2 Analytics Accelerator Studio, IBM Data Studio, is built. Hence this information does not pertain to errors in IBM Db2 Analytics Accelerator for z/OS, but to its software environment.
5. To revoke all individual settings and in a future run save just the information that is selected by using the default trace profiles, select **Reset configuration profiles to default after collecting traces**. This prevents that verbose trace information consumes too much space on your hard disks. If you want to retain your individual settings after saving them once, clear the **Reset configuration profiles to default after collecting traces** check box or leave it deselected (default).
6. Under the heading **Save to local file system**, in the **Folder** field, type the location of the trace file (full directory path) on your local system.

   Alternatively, you can click **Browse** to select a folder.
7. Select **Add trace information to IBM problem management record (PMR) using FTP** to add the selected information to an existing PMR.
8. In the fields next to **PMR number**, type the identification number of the PMR.
9. In the **FTP password** field, type the password for logging on to the IBM RETAIN server on which the PMR was opened.

   Note: Most IBM FTP servers for PMRs accept the registered email address as the password.

   The currently selected server is displayed in square brackets in the text above the **Password** field. To change the server or the proxy settings required to access the server, select **Window** > **Preferences** > **IBM Db2 Analytics Accelerator for z/OS** > **Save Trace Dialog** from the **IBM Data Studio** main menu.
10. If you want to reset your choices in the **Save Trace** window to make corrections or start over, click **Restore Defaults**.

Note: This button just undoes all of your recent actions in the **Save Trace** window. It does not reset profiles and affects your current settings rather than future settings. Compare this with the previous description of the **Reset configuration profiles to default after collecting traces** check box.

11. Click **OK** to save your settings.

Related tasks

Configuring the trace behavior

IBM Db2 Analytics Accelerator provides trace profiles. These profiles consist of lists of components and events that result in the collection of trace information. In addition, they determine the verbosity level of the information, that is, how detailed the trace information will be. Profiles are available for accelerators as well as for stored procedures. You can also use custom profiles that have been saved to an XML file.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Setting a refresh interval or a skew threshold

Select **Preferences** > **IBM Db2 Analytics Accelerator** > **Accelerator View** from the main menu to set the refresh interval for the **Accelerator** view or the warning threshold for table skew.

Procedure

1. Set values:

   - To set or change the interval after which the information in the **Accelerator** view is automatically refreshed, select the appropriate value from the **Automatically refresh information in the Accelerator view** drop-down list. The default is **Every minute**, meaning that the information in the **Accelerator** view is refreshed each time after the passing of one minute.
   - To set or change a table skew value that will result in a warning message when it has been exceeded by a very big accelerator-shadow table in the range of 100 million rows, type a decimal number between 0 and 1 in the text field next to **Show a warning if the skew value of very big tables exceeds**. Example: `0.7`. The default is `0.25`.

     Notes:

     – What is considered to be *very big* varies with the size of your IBM Integrated Analytics System. On smaller systems, the tables that will cause a warning are smaller and vice versa. The threshold is somewhere in the range between 50 million and 100 million table rows.

     – The actual skew value can be greater than 1, and thus you could enter a value greater than one, but for the tables that are affected by the skew threshold (very big accelerator-shadow tables), 1 is already a bad value.

     To be able to issue a warning, IBM Db2 Analytics Accelerator Studio must compare the specified skew threshold with the actual skew values of the tables on your accelerators. The actual values are obtained by running the **SYSPROC.ACCEL_GET_TABLES_INFO** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.
   - To disable checking for unsupported columns, clear the **Check for unsupported columns** check box. By default, this feature is enabled because it tells you beforehand why a table cannot be added or why certain columns were skipped.

     Important: For the deactivation to take effect, you must disconnect, then reconnect to the Db2 subsystem. To do so:

     a. Go to the **Administration Explorer**.
     b. In the **All Databases** folder, navigate to the appropriate Db2 subsystem icon.
     c. Right-click the icon.

d. Select **Disconnect** from the menu.

e. Double-click the icon to reconnect.

2. Click **Apply** to save the changes or **Restore Defaults** to undo the changes.
3. Click **OK** to save your settings and exit the window or just exit the window.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Updating accelerator tables continually

The incremental update function of IBM Db2 Analytics Accelerator for z/OS allows you to update accelerator-shadow tables continually. Changes to the data in original Db2 for z/OS tables are thus propagated to the corresponding target tables with a high frequency and just a brief delay. This way, query results from an accelerator are always extracted from recent, close-to-realtime data.

The information about changes to be transferred comes from the Db2 for z/OS log.

In the past, data in accelerator-shadow tables could only be updated by use of the table load function. With this function, you can only update entire tables and entire table partitions, and the latter only if the tables are partitioned by range or partitioned by growth. The incremental update function, in contrast, updates individual table rows. Advantages and disadvantages are connected with either method. Table 25 on page 117 compares the advantages and disadvantages.

Table 25. Comparison of table update methods

Manual update of entire table using load function	**Manual update of entire partitions using load function**	**Incremental update**
For all types of tables	Only for range-partitioned tables and tables partitioned by growth	For all types of tables
Potentially long-running process, depending on the size of the tables	Potentially long-running process, depending on the size of partitions and the distribution of data changes across partitions (the more partitions are involved, the more time-consuming the update becomes)	Short-running process after initial load
Fast (up to one TB per hour)	Fast (up to one TB per hour)	Slower (in the range of GBs per hour)
Redundant load of table rows that have not changed, resulting in waste of processing resources (MIPS)	Redundant load of table rows that have not changed, resulting in waste of processing resources (MIPS)	No redundant load of table rows, no waste of processing resources.
Just a snapshot of the original data at some point in the past	Just a snapshot of the original data at some point in the past	Continual updates, recent data

Use cases

Incremental updates are the preferable method with regard to the following points of interest:

Realtime business intelligence

If realtime data is very important for your business intelligence operations, incremental updates are the preferable choice. Incremental updates do not guarantee that query data is the most recent available, but the latency is low, and this makes incremental updates a cost-efficient alternative.

Quick reporting on operational data

Incremental updates provide an uncomplicated, easy-to-use, and fast way to reporting on data in an operational data store (ODS) because the function is deeply integrated into your Db2 for z/OS and IBM

Db2 Analytics Accelerator for z/OS setup. No up-front extraction and transformation of data is required to obtain results.

Cost-efficiency

Incremental updates are more cost-efficient than traditional update methods because there is no redundancy during updates and hence no waste of processing resources.

In addition, internal tests have shown that running incremental update processes affect the query response time only marginally.

High availability

Incremental updates can be configured in such a way that data changes in a Db2 for z/OS table are propagated to more than one accelerator-shadow table on multiple accelerators. If one accelerator fails, another one can take over to process incoming queries. This happens with just a short delay.

Making queries wait for incremental updates

To minimize the gap in data recency between the original Db2 for z/OS tables and incrementally updated accelerator-shadow tables (replication latency), you can delay the execution of queries until the latest incremental updates have been applied. This guarantees that an accelerated query against incrementally updated tables always "sees" the latest changes that have been committed in Db2 for z/OS.

This capability fulfills the criteria for Hybrid Transactional Analytical Processing (HTAP). It allows you to run transactional processes as well as analytical queries on a unified data store with a minimized time lag between transactional completion and analytical insight. Informed business decisions can be made much faster, on the basis of much more recent data.

To activate the delay, you must add a special register to your queries. You can also use a ZPARM or bind option to the same effect. The special register is:

```
SET CURRENT QUERY ACCELERATION WAITFORDATA = n.m.
```

where *n.m.* is a decimal number denoting seconds. The minimum is `0.0` (no delay). The maximum is `3600.0` (a delay of one hour).

Important: If you want to use a delay, specify at least 20 seconds. Lower values might work, but you must take the time into account that is needed to scan and read the SYSLOG and to report the findings. Experience has shown that realistic delays are around 20 seconds long.

Example

```
SET CURRENT QUERY ACCELERATION = ENABLE;
SET CURRENT QUERY ACCELERATION WAITFORDATA = 120.0;
SELECT * FROM XYZ WHERE [...]
```

The first special register in line 1 enables query acceleration. The second special register sets the delay to 120 seconds. What follows is the SQL code for the query. The execution of the query will be postponed for 120 seconds. During that time, the replication engine tries to apply all committed changes that have accumulated before the query started. If the replication engine cannot finish the job during that period, the query will not be accelerated. The further handling depends on the option set in the first special register and on the expiration behavior configured on the IBM Db2 Analytics Accelerator Console. The first special register is set to ENABLE. That means the query fails if one the conditions for query acceleration is not met. Let's assume that the conditions can be met; so normally, the query would run on the accelerator. However, the default expiration behavior is to abort a query if the incremental updates cannot be applied in time. Supposing the default behavior is set, the query would indeed fail if not all of the pending incremental updates could be applied in 120 seconds.

How to calculate the proper delay time

You certainly ask yourself how to determine the proper delay time (WAITFORDATA value) for your environment. There are two values by which an initial delay time or starting point can be calculated: the *replication latency* and the *replication velocity*. The replication latency is the time that the incremental update process is lagging behind. The replication velocity is a measure for the speed at which data changes could be applied in the past. You get these two values by running the following Db2 command:

```
-DIS ACCEL(<name>) DETAIL
```

where <name> is the name of the accelerator attached to the Db2 subsystem. If incremental updates have been configured for this pair of Db2 subsystem and accelerator, you find the two values in the command output. For example:

```
...
CURRENT REPLICATION LATENCY FOR THIS DB2 SYSTEM = 2000 MS
NUMBER OF SUCCESSFUL QUERY REQUESTS WITH DELAY PROTOCAL FOR ALL DB2 SYSTEMS = 0
NUMBER OF EXPIRED QUERY REQUESTS WITH DELAY PROTOCAL FOR ALL DB2 SYSTEMS = 1
REPLICATION VELOCITY = 0.04 DB2 LOG SECONDS APPLIED PER SECOND
...
```

In this example, The latency is 2000 milliseconds or 2 seconds, and the velocity is 0.04. You get the proper WAITFORDATA value by dividing latency by velocity. For the example, this gives you a WAITFORDATA value of 50 seconds to start with.

See the note on replication latency.

Conditions for the success or failure of a query

Whether a query fails depends on the setting of the SET CURRENT QUERY ACCELERATION special register. It also depends on the type of the table. For accelerator-only tables, the WAITFORDATA option is always ignored because it can be assumed that the data in these tables is always the most recent data available (there is no other data in a Db2 for z/OS source table).

If the WAITFORDATA option has been set for queries that reference incrementally updated tables as well as tables not enabled for incremental updates (loaded tables), the query might fail if loaded tables are to be excluded.

For information on how to configure the treatment of loaded tables, see *Including or excluding loaded tables from WAITFORDATA queries* in the *IBM Db2 Analytics Accelerator for z/OS: Installation Guide*.

Table 26 on page 119 lists the possible combinations of settings and their effect with regard to query processing. It is assumed that an incoming query satisfies the conditions for query acceleration.

Table 26. Settings that influence WAITFORDATA query processing

CURRENT QUERY ACCELERATION	**WAITFORDATA**	**Loaded tables[2]**	**Query processing**
NONE	N/A	N/A	Db2 for z/OS
ENABLE / ELIGIBLE / ALL	`0.0`	*Include*	Accelerator
		Exclude	Query fails
	`> 0.0`	*Include*	Query fails if outstanding incremental updates cannot be a applied within WAITFORDATA period.
		Exclude	Query fails
ENABLE WITH FAILBACK	`0.0`	*Include*	Accelerator
		Exclude	Db2 for z/OS
	`> 0.0`	*Include*	Db2 for z/OS
		Exclude	Db2 for z/OS

[2] This has an effect only if loaded tables are referenced by the query

Error codes in connection with WAITFORDATA processing

If an error occurs in connection with WAITFORDATA processing, an error message with `SQLCODE = -904` is returned in the format:

```
WAITFORDATA rsn=<reason code>:<reason code text>
```

where

<reason code>
An integer (positive, negative, or 0)

<reason code text>
The message text

Example

```
SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN UNAVAILABLE RESOURCE.
REASON 00E7000E, TYPE OF RESOURCE 00001080, AND RESOURCE NAME WAITFORDATA
rsn=1:Table(s) are in invalid state.
```

The message shows `rsn=1` as the `<reason code>`, and `Table(s) are in invalid state.` as the `<reason code text>` .

Table of error codes

Table 27 on page 120 lists the possible reason codes and message texts. It also provides a description of the most common causes.

Table 27. Reason codes and texts in WAITFORDATA error messages

RSN code	Message text	Common causes
1	`Table(s) are in invalid state.`	Not all tables to be processed are in the *replicationInProgress* state. This state, however, is a precondition for WAITFORDATA processing. Check the **Last Load** column in the **Tables** section of the **Accelerator** view in IBM Db2 Analytics Accelerator Studio.
2	`Subscription state is invalid.`	Incremental updates have not been started for the accelerator on which the tables are located. In IBM Db2 Analytics Accelerator Studio, check the **Replication** status in the header of the **Accelerator** view. The status must be *Started*. If necessary, click **Start**.

Table 27. Reason codes and texts in WAITFORDATA error messages (continued)

RSN code	Message text	Common causes
3	`Set of tables is empty or invalid.`	Some tables are empty or have not been enabled for incremental updates. Make sure that none of the tables in the set are empty. If you find empty tables, reload them, or exclude them from the set. Check the **Replication Since** column in the **Tables** section of the **Accelerator** view in IBM Db2 Analytics Accelerator Studio. The column must show a date for all the tables to be processed. If necessary, click **Replication** > **Enable Replication**.
4		
5	`Timeout for WAITFORDATA reached.`	Pending incremental updates could not be applied within the WAITFORDATA period.

Dependencies on other processing states, configurations, or product features

Other processing states or the use of certain configuration settings or other product features might affect query processing if WAITFORDATA is enabled (value > 0.0):

- If one of the tables referenced by the query is currently being reloaded, or in the spilling or draining state, the query will fail or be processed by Db2 for z/OS (ENABLE WITH FAILBACK). This is because the end of these processes or table states cannot be predicted, and in most cases, they take much longer than the WAITFORDATA period.
- If partitions of a table have been archived with the High Performance Storage Saver (HPSS), WAITFORDATA processing can still be used for the active partitions of the table. The archived partitions do not receive incremental updates, so a WAITFORDATA setting will not have an effect.
- If workload balancing is used in a high-availability setup, a WAITFORDATA setting > 0.0 might lead to successful or failing queries. The decisive factor is the replication latency of the accelerator that a query is sent to. If the latency is higher than the WAITFORDATA period, the query fails. The problem is that one does not know beforehand to which accelerator a query is sent. The accelerator is picked at random.
- If the product software on the accelerator is back-level, and does not support the WAITFORDATA protocol, a query does not necessarily fail. If `CURRENT QUERY ACCELERATION` is set to `ENABLE`, `ELIGIBLE`, or `ENABLE WITH FAILBACK`, the query might be processed by Db2 for z/OS. If, on the other hand, the special register is set to `ALL`, the query will fail. The same applies to the equivalent ZPARM or bind option settings.
- WAITFORDATA processing is not possible if *federated access* has been granted to any of the tables referenced by a query. This is because the reference tables (referencing accelerator-only tables), which contain metadata, cannot be updated by the same incremental update process, as they were created from a different, disconnected Db2 subsystem. The reference tables would inevitably be out-of-sync. Therefore, queries against such tables will fail.
- You cannot change the distribution key or the organizing keys of a table while incremental updates are enabled. To enable the **Alter Keys** function, you must first disable replication, and that means that a setting of WAITFORDATA > 0.0 will have no effect.

- A failure of the database engine on the accelerator has the effect that all queries with a WAITFORDATA setting will fail. However, these queries will be restarted and processed as usual after the database engine has recovered or after a replacement instance has been started.
- Treatment of date, time, and timestamp values: An accelerated query that retrieves dates might give you the same values as Db2 for z/OS, but a query that selects time values or timestamps will probably produce different results. The reason is that the WAITFORDATA period and the difference between the z/OS system clock and the accelerator's system clock (=< 7 seconds) are added to these values while the incremental updates are being applied.

Related concepts

Accelerator view
To show the tables on an accelerator, view the query history, or access various other functions, double-click an accelerator name in the **Object List Editor**, which opens the related **Accelerator** view.

Related tasks

Including or excluding tables from incremental updates
Incremental updates can be enabled or disabled on the table level. So you can individually select the tables that take part in the process. When a subsystem has been configured for incremental updates, the toolbar of the accelerator view for regular accelerators shows an extra button, which works like a drop-down menu.

Starting or stopping incremental updates

When a Db2 subsystem has been configured for incremental updates, you can start or stop incremental updates for attached accelerators from the corresponding accelerator views in IBM Db2 Analytics Accelerator Studio. Using the functional links in the header starts or stops incremental updates for an entire accelerator.

About this task

When a subsystem has been configured for incremental updates, the header of accelerator view for regular accelerators shows an additional entry, which is labeled **Replication**. Next to this label, you find a status indicator and a functional link. Status indicator and link vary according to the current status of the accelerator. For example, if incremental updates have been stopped, the status indicator shows *Stopped* and a functional link labeled **Start**. This is reversed to *Started* and **Stop** when incremental updates have been started. The following statuses are possible:

Disabled
The incremental update function has not been configured for this accelerator and is therefore not available.

Error
The incremental update function has been configured, but an error occurred. In most cases, this means that a component cannot be reached. Therefore, it cannot be started or controlled. The event log might provide useful information about the root cause of the error.

Started
The incremental update function is running and updates are applied to accelerator-shadow tables that have been included in the process.

Stopped
The incremental update function is available and accelerator-shadow tables have been included in the process. However, the function is not running right now.

See Figure 10 on page 122.

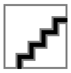


*Figure 10. **Replication status** and functional link in the header of the accelerator view*

Important: Using the functional links starts or stops the incremental update process for all tables that have been enabled for incremental updates on the selected accelerator.

Procedure

1. Start IBM Db2 Analytics Accelerator Studio and connect to a Db2 subsystem.
2. In the **Administration Explorer**, select the **Accelerators** folder.
3. In the **Object List Editor** on the right, double-click the accelerator that you want to include in the incremental update process.
4. Start or stop incremental updates by taking one of the following actions:
   - To start incremental updates, click **Start** in the header of the accelerator view. The **Replication status** in the header of the accelerator view changes to *Started*.
   - To stop incremental updates, click **Stop** in the header of the accelerator view. This opens the **Stop Replication** window, in which you can select between the following options for stopping the incremental update process:

     Controlled
     Completes all work in progress before stopping the process.

     Immediate
     Stops the process without completing work in progress. Compared with the **Controlled** option, using this option causes a restart of the process to take longer.

     Click **OK** after selecting an option. You return to the accelerator view, which now shows a **Replication status** of *Stopped*.

Including or excluding tables from incremental updates

Incremental updates can be enabled or disabled on the table level. So you can individually select the tables that take part in the process. When a subsystem has been configured for incremental updates, the toolbar of the accelerator view for regular accelerators shows an extra button, which works like a drop-down menu.

Before you begin

An accelerator-shadow table can only be included in the incremental update process if it contains data, that is, if it has been loaded.

When loading a table, make sure that the lock mode **TABLE_SET** is selected. This ensures that data changes will not go amiss during the time between unloading the table data from Db2 to the accelerator and the point where the incremental update process sets in.

You can select lock mode **ROW** for a better performance. This mode effects a faster propagation of changes for the following reason: Incremental updates are not halted when a table load starts and continued after the load, but are temporarily written to a spill queue. This queue is drained in the course of events.

However, this comes at the price of a slower load process because lock mode **ROW** prevents the loading of the temporary table, which eventually replaces the former accelerator-shadow table. That is, loads are executed as a series of row changes. Decide whether faster loads or faster incremental updates are more important to you.

Attention:

- Tables to be included must have a unique constraint (primary key or primary index). If such a key does not exist in the accelerator-shadow table or cannot be determined, you must redefine the table and specify such a key. You cannot do this from IBM Db2 Analytics Accelerator Studio; you must run the corresponding stored procedure (SYSPROC.ACCEL_ADD_TABLES) with a special option. The columns that you choose for the key must contain unique values or form such values when they are combined. The values of the selected key columns will then be written to a hidden column in the accelerator-shadow table. This column, called *informational unique constraint*, then serves as the primary key or primary index. For more information, see the entry about the *`table_specifications`* parameter of the SYSPROC.ACCEL_ADD_TABLES stored procedure in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*.

- When lock mode ROW is set, Db2 data is unloaded at isolation level CS (cursor stability). Whether a row, a page, or an entire table is locked is determined by the LOCKSIZE defined for the table space of the source tables. To avoid deadlocks during the draining of the spill queue, set this table space to `LOCKSIZE ROW` if you want to use lock mode ROW for your incrementally updated tables.
- If you update Db2 for z/OS tables by running the LOAD utility rather than an INSERT, UPDATE, or DELETE operation, you must set the following parameters (for the LOAD utility):
  - `SHRLEVEL CHANGE`
  - `LOG YES`

  Otherwise, the changes that were made by the LOAD utility are not detected by the incremental update function, and will thus not be reflected in your accelerator-shadow tables.
- Tables on an accelerator have a Db2 attribute named DATA CAPTURE. The attribute can carry the value Y or N (default), for *yes* or *no*. When incremental updates are enabled for a table, the DATA CAPTURE attribute of the table is set to the value Y. Once set, this attribute value persists, even if the table is disabled at a later time. Bear this in mind, especially if you run applications that use the DATA CAPTURE attribute.
- You might have to reload or even remove tables from an accelerator after the execution of an ALTER TABLE or ALTER TABLESPACE statement in Db2 for z/OS. For more information, see *What to do after changing a table or table space in Db2 for z/OS*. You find a **Related reference** link at the end of this topic.
- IBM InfoSphere Data Replication for z/OS uses a metadata table called DMMD_SIGNALS. This table is a kind of an interface for the entire process. It does not make sense to select it for incremental updates. Ensure that you do not select it inadvertently as doing so will lead to unpredictable and unwanted side-effects.

About this task

Two different situations must be considered before adding tables to an incremental update process:

- An accelerator-shadow table has not yet been loaded, that is, it is in *Initial load pending* state. If you enable such a table, and load it immediately thereafter, the tables in Db2 and on the accelerator are in sync. No data changes will be lost because table contents are the same at the outset and the incremental update process captures all future changes and propagates these to the accelerator. Tables of this type show the same timestamp in the **Last Load** and **Replication Since** columns of the accelerator view. See the entry for the REP1 table in Figure 11 on page 124.
- An accelerator-shadow table has already been loaded when it is added to the incremental update process. In this case, it is likely that the tables are out of sync because data changes might have been applied to the original Db2 table, and not been reflected in the corresponding accelerator-shadow table because the last manual update of the table was before these changes happened. If a safety mechanism was not in place, such tables would continue to be out of sync because the incremental update process can only capture and propagate changes that occur in the future. To avoid this, the status of such tables changes to *Initial load pending* when incremental updates are enabled. This state enforces a complete reload of the tables before they can be used, so that the synchronization gap is closed before the incremental update process sets in.

Important: If the load fails for some of the selected tables, incremental updates are automatically disabled for these tables.

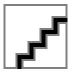


*Figure 11. Timestamp columns and **Replication** button in the accelerator view*

The **Replication Since** column shows the following states in the order indicated as soon as a load process begins:

1. *Suspended*

2. *Refreshing*

The *Refreshing* state is shown until the last change has been drained from the spill queue. When you move the mouse pointer over the *Refreshing* label, the current processing phase is displayed in a tool-tip window. The refresh process consists of the following phases:

Pending
The table load process was started, but no data has yet been transferred. Therefore, only the data that has been transferred before the refresh is available for accelerated queries.

Spilling
The table load is in progress, and data is being transferred. Incremental updates that occur during this time are written to the spill queue. Therefore, only the recently transferred load data is available for accelerated queries.

Draining
The load has finished and the changes that have been collected in the spill queue are being applied. Only the data changes that have already been transferred to the table are available for accelerated queries.

Tip: If a table remains in the draining phase for several hours, you are probably in the very rare situation in which new changes arrive in the spill queue faster than they can be drained from it, so that the draining phase never ends (infinite drain). In this case, stop incremental updates completely for a while.

Restriction: It is not possible to enable tables if the table name or the schema name contains GB18030 characters (Simplified Chinese) of Unicode plane 2 (U+20000-U+2FFFF: Supplementary Ideographic Plane). Trying to do so results in an error.

Procedure

1. Select the accelerator-shadow tables to be included or excluded.
2. Include or exclude tables from the process:

   Important: The incremental update process must be running when you enable the very first table. Make sure that this is the case. If not, click **Start** in the header of the **Accelerator** view.

   - To include selected tables in the incremental update process, click **Replication** > **Enable Replication**. See Figure 11 on page 124 for the location of the button.
   - To exclude selected tables from the incremental update process, click **Replication** > **Disable Replication**.

   Important: A running incremental update process is not required if you want to enable additional tables later on. However, the point becomes relevant again if you have disabled all tables. The process must be running when you enable or re-enable the first table.

Related reference

What to do after changing a table or table space in Db2 for z/OS
In a number of cases, an ALTER TABLE or ALTER TABLESPACE statement in Db2 for z/OS does not involve maintenance work on your accelerator-shadow tables. There are cases, however, in which it does. See which steps you might have to take.

Related information

CREATE TABLESPACE

Monitoring incremental updates

You can monitor the changes to accelerator tables that result from incremental updates. The header of the accelerator view in IBM Db2 Analytics Accelerator Studio provides an additional line of information when incremental updates have been configured for the accelerator. This line of information is labeled **Replication latency**.

One of the following values is displayed next to this label:

Stopped

Incremental updates have been stopped for the entire accelerator.

Low

It will take less than three minutes to propagate recent changes to the corresponding accelerator-shadow tables.

Medium

Recent changes to the original DB2 tables will be propagated to the corresponding accelerator-shadow tables within three and six minutes.

High

It will take six minutes or longer to apply recent changes to DB2 tables to the corresponding accelerator-shadow tables.

When you move the mouse pointer over the latency value, a pop-up window shows the following information:

Target Agent

Operations completed by the incremental update function.

Inserts

Number of row inserts in accelerator-shadow tables that have been included in the incremental update process.

Deletes

Number of row deletions from accelerator-shadow tables that have been included in the incremental update process.

Remember:

- An UPDATE is a DELETE followed by an INSERT, so these two figures account for all incremental updates that have been applied.
- Since an UPDATE in DB2 is a DELETE followed by an INSERT on the accelerator side, the values can be different from those in Db2 for z/OS (see **Source Agent**). Furthermore, multiple operations on a single row within a short time might be reported as just one single operation.
- If a load operation is in progress and lock mode **Row** has been selected, the original DB2 tables are not locked during the load. Hence it is possible to INSERT new rows into the DB2 tables even though a load is taking place. If incremental updates are enabled for the same tables, the incremental update function stores these new rows in the spill queue until the load has finished, and applies these changes thereafter (draining the spill queue). In the course of a drain, a DELETE and an INSERT operation is carried out for each row in the affected accelerator-shadow tables. The DELETE deletes the row that was inserted by the previous load operation, and the INSERT adds this row again (applies the latest change) . This has the effect that the number **Deletes** increases in cases like this.

See Figure 12 on page 126.

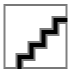


*Figure 12. **Replication latency** in the accelerator view*

Using the event viewer for incremental updates

The event viewer shows system messages that are related to incremental update processes. It is equipped with various filtering options to facilitate the search for particular messages.

About this task

The event viewer shows a log entry for each system event that occurred in connection with incremental updates. The entries are listed in a table, which contains the following columns:

Type
The type of the event or message connected with the event. Possible types are **Error**, **Warning**, and **Information**.

Message
The message code and the starting text of the event message. The full message is displayed in the **Message** window on the lower right.

Event ID
The unique identifier of the event.

Originator
The component that issued a message.

Time
The time at which an event occurred.

Procedure

- To open the event viewer:

  a) Open the **Accelerator view** of the appropriate accelerator.

  This must be an accelerator that is included in the incremental update process. That is, incremental updates must be configured for the Db2 subsystem to which the accelerator is attached. The current replication status of the accelerator (*Started*, *Stopped*, or *Error*) is irrelevant.

  b) Click the **Show events** link in the header of the **Accelerator view**.

- Selecting a message in the **Replication Events** window cause the main column values to be displayed in the **Event Details** section on the lower left, and the full message text to be displayed in the **Message** window on the lower right.

- Using the **View** drop-down list on the upper right, you can hide all messages of an event type that is different from the selected type. To apply this type of filter, click the **Run** button after selecting an event type from the drop-down list.
  For example, if you select **Error** and **Run**, only error messages and messages with a severity higher than *error* are displayed.

  The underlying mechanism is the same as for the classification of trace information. The lower the severity of the selected information type, the more verbose is the information that is displayed. While the **Error** type provides information about errors and even severer events only, the **Warning** type provides information about warnings, errors, and severer events. That is, each information type with a lower severity includes the information of the types with a higher severity.

- Using the **Show** drop-down list, you can select a time frame for the messages to be displayed. To apply this type of filter, click the **Run** button after selecting a time frame from the drop-down list.
  For example, if you select **Last 24 hours** and **Run**, the event viewer shows all events that happened during the last 24 hours.

  a) Selecting **Custom interval** from the **Show** drop-down list opens the **Define Custom Interval** window, in which you can define a time frame of your own:

  - To use a predefined time frame (same as using the **Show** drop-down list in the **Replication Events** window), leave the **Show events within** radio button as is, and select a suitable time frame from the adjacent drop-down list.
  - To define a custom time frame, select **Show events** and:

    a. In the line starting with **From**, select the start of your time frame by using the date picker and the time spin buttons on the right.

    b. In the line starting with **To**, select the end of your time frame by using the date picker and the time spin buttons on the right.

  b) Click **OK** to return to the **Replication Events** window.

- Typing a search term in the search field on the top of the table reduces the messages being displayed to those messages that contain the search term.

For example, if you know the error code belonging to a particular message, you can type in this error code to show just the messages for one particular type of error.

- Close the event viewer by clicking the cross-shaped icon on the window tab.

Warnings related to incremental update processes

The following warnings might occur in connection with incremental updates.

InsufficientLockModeWarning

This warning is issued if you did not select the option **Lock Db2 tables while loading** for the initial load of an accelerator-shadow table taking part in the incremental update process, or if you selected the lock modes **NONE** or **PARTITION**.

If locks were disabled for the initial load of an accelerator-shadow table (**Lock Db2 tables while loading** not selected), the following message is displayed:

```
One of the tables has replication enabled. This table must be locked while
loading to avoid an inconsistency of data between Db2 and the accelerator.
```

If the lock modes **NONE** or **PARTITION** were used, data consistency between tables in Db2 and accelerator-shadow tables cannot be guaranteed because table changes can be applied to the Db2 tables during the unload process. If possible, use the lock mode **TABLE_SET**.

NoUniqueConstraintsDefinedWarning

A table does not have a unique constraint, such as a unique key or unique index. This would result in a poor performance when UPDATE or DELETE changes are propagated to an accelerator-shadow table (because it takes longer to identify the changed table rows).

To avoid a performance degradation, the product inserts informational unique constraints into the accelerator-shadow table. That is, a hidden column with unique values is inserted to serve as the unique index.

OrganizingKeysAndUniqueConstraintsMismatchWarning

The unique constraints of a table in Db2 for z/OS differ from those of the corresponding accelerator-shadow table or exceed the number of matching organizing keys. This might lead to a slow propagation of data changes to the accelerator. In the case of different unique constraints, it just takes longer to identify the table rows to be updated or deleted. A mismatch between the unique constraints of the table and the organizing keys affects the change propagation negatively if there are more unique constraints than matching organizing keys. If extents cannot be located for certain table rows because there is no matching organizing key, table scans take longer.

Collecting CDC trace information

None of the options in the **Save Trace** window of IBM Db2 Analytics Accelerator Studio collects trace information about the CDC Capture Agent. This information, however, might be the key to a problem with the incremental update function.

About this task

If you are facing a problem with the incremental update function, provide as much information about the related software components as possible the support team. This includes information about the CDC Capture Agent, which is the part of IBM InfoSphere Data Replication for z/OS that runs on z/OS. The support team might even ask you for this type of information. Here is a description on how to obtain it.

Procedure

1. The CDC Capture Agent is just one of the started tasks running on your z/OS system. If you do not know the name of the started task, ask a z/OS systems programmer which task that is.
2. In the SDSF program on z/OS, type XDC in front of the relevant address space output to copy its contents to a sequential data set.
3. Transfer this data set in ASCII mode to your Linux or Windows workstation and add the resulting .txt file to your problem management record (PMR).

4. If a trace dump has been generated, include that information as well. Transfer this data in TERSED format to your workstation because this retains the mainframe-readable formatting of the data during the transfer to a non-mainframe system, and thus ensures that the data remains useful for IBM support. Use an extension of .trs for the files.

   The TERSED format also reduces the file size. See the documentation for the AMATERSE program if you need further assistance.

What to do after changing a table or table space in Db2 for z/OS

In a number of cases, an ALTER TABLE or ALTER TABLESPACE statement in Db2 for z/OS does not involve maintenance work on your accelerator-shadow tables. There are cases, however, in which it does. See which steps you might have to take.

NOT Programming Interface information

ALTER TABLE

- If a column was added, changed, or deleted by an ALTER TABLE operation on a Db2 table, queries against this table are not accelerated if they reference the new, changed, or deleted column (reason code 14). Therefore, you must complete the following steps after an ALTER TABLE operation:

  1. Remove the corresponding accelerator-shadow table from all accelerators on which it is defined.

     In most cases, a REORG operation is not required before the table can be redefined on the accelerator. However, a REORG is required in the following cases, although this seems to be just a recommended action, as the advisory *REORG-pending* (AREO) state suggests:

     – The ALTER TABLE operation added a column.

     – The ALTER TABLE operation changed column data types.

     – The ALTER TABLE operation dropped columns in *Db2 11 for z/OS*.

  2. Redefine the table on the accelerators from which you previously removed it.
  3. Load the redefined table on all accelerators.
  4. Enable query acceleration for the redefined table on all accelerators.

ALTER TABLESPACE

If a table space was changed in Db2 for z/OS (ALTER TABLESPACE), and the tables of that table space also exist on an accelerator, actions are required under the following conditions:

- If the maximum number of partitions has changed (MAXPARTITIONS), you only have to reload the new or changed partitions. It is not required to reload existing partitions that have not changed.
- If a table space was changed from *segmented* to *partitioned by growth*, you must remove the accelerator-shadow tables of that table space from all accelerators, then redefine and re-prepare these for accelerated queries. Follow steps “1” on page 129 through “4” on page 129 to do that.

End of NOT Programming Interface information

Freeing up storage in Db2 for z/OS

The high-performance storage saver (HPSS) archives data of table partitions in Db2 for z/OS to an accelerator. In moving data that is no longer actively used (historical data) to a less expensive storage device, you can free up costly storage space on your database server.

Details

The HPSS is a fully integrated function. Its use has the positive side-effect that it also makes the database system more responsive. This is because fewer objects need to be maintained in the catalog, smaller or fewer indexes need to be searched, reorganizations become quicker, report data and statistics can be gathered faster.

If a query could be accelerated while the original data resided in Db2, it can also be accelerated when the data has been archived on an accelerator.

Just like before, the query routing process is handled by the Db2 query optimizer and IBM Db2 Analytics Accelerator. If data has been archived to an accelerator, and an incoming query needs to access this data, and if furthermore the query qualifies for acceleration, then the query is automatically processed by IBM Db2 Analytics Accelerator.

You can determine whether archived data is to be included in queries. By default, this type of data is excluded, and IBM Db2 Analytics Accelerator for z/OS automatically skips archived records when it processes a query. This often leads to different results for queries that are run repeatedly because the query is run just against the remaining Db2 data. If this behavior is not wanted, you can set the `CURRENT GET_ACCEL_ARCHIVE` special register or the `GET_ACCEL_ARCHIVE` configuration parameter to YES.

You can reload accelerator tables without thinking about archived partitions. Partitions that have been archived are automatically excluded from the load process.

The HPSS is flexible. It is still possible to define one and the same query table on multiple accelerators. It is also possible to archive the data of one or more partitions on an accelerator, while the data of other partitions remains in Db2 for z/OS and on the same or other accelerators.

Before any data is moved to an accelerator and deleted from Db2, image copies are created, which allow you to restore the data if needed. The table spaces of the affected (archived) partitions are set to a persistent read-only state, which prevents future INSERT, UPDATE, and DELETE operations on these partitions. Furthermore, the HPSS ensures that the data in the image copies is consistent with the data that is going to be archived.

The table space of an archived partition continues to exist. It will be empty, but will still claim as much disk space as defined by the *minimum primary space allocation*. To meet your space-saving goal, check the minimum primary space allocation (PRIQTY) for the table spaces of the partitions that you want to archive. Decrease the value if needed by submitting an appropriate `ALTER TABLESPACE ... PRIQTY` statement.

Restrictions

The following restrictions exist in connection with the HPSS:

- The HPSS works on range-partitioned tables only. If the partitioning is controlled by an index, the index must (already) exist.
- The smallest unit that you can archive is a partition. That is, all the table rows in specified partitions are copied to the accelerator and are finally removed from Db2.
- You can only archive partitions of tables that exist as accelerator-only tables and that are in *InitialLoadPending* or *Loaded* state. This implies that partitions cannot be archived if incremental updates have been enabled for the table.
- Table data cannot be archived if a column in the table serves as the parent in a foreign-key-relationship. The reason for this is that the original data is deleted at the end of the archiving process, which removes the values of the foreign key. This cannot be permitted because it violates a fundamental rule for data integrity in a relational database.
- Normally, you cannot rotate a Db2 partition or change partition boundaries if the data of this partition data has been archived on an accelerator. Commands to rotate a partition or change partition boundaries will be rejected by Db2 for z/OS. This restriction can be circumvented by first removing the Partition Read-Only (PRO) state with the help of the Db2 REPAIR utility. If you rotate a partition after removing the PRO state, you are still able to recover the archived partitions by using the image copies, except for those partitions that have been rotated out. If you change the partition boundaries, on the other hand, the image copies become unusable and the data cannot be restored. You will also end up with orphaned data on the accelerator.

⚠ **Attention:** A rotation leads to the deletion of the data in those Db2 partitions that have been "rotated out". In addition, the corresponding image copies, which were created when the data was archived, will become unusable. The first archiving operation on the table that used to contain the "rotated -out" partitions will also delete the data of the "rotated-out" partitions from the accelerator, irrespective of the partitions that you specified for archiving.

- Starting with product version 4.1 PTF-3, you can archive partitions that contain *Db2 archive tables*. However, you cannot archive *Db2 archive-enabled tables*. Do not confuse these tables with partitions and tables that were archived on an accelerator by the HPSS (accelerator-archive tables). If necessary, check the Db2 documentation for definitions of the terms *Db2 archive table* and *Db2 archive-enabled table*. Bear in mind also that when a Db2 archive table has been processed by the HPSS, you can no longer shift rows from the corresponding Db2 archive-enabled table to the Db2 archive table or vice versa. A message with `SQLCODE=-904` is returned when you try to do that.
- No failback mechanism is in place in case an error occurs. Records that have already been archived are not rolled back. The Db2 catalog explicitly marks the partitions containing these records as “archived”.
- Columns that cannot be loaded into accelerator tables because their data types are not supported can also not be archived. These are basically the following data types:
  - DECFLOAT
  - TIMESTAMP(*n*) where *n* has a value other than 6.
  - ROWID
  - User-defined types that are based on the above

  However, an attempt results in the issuing of a warning. The rest of the partition or table data will be archived. Although the original data is deleted after archiving, the data in unsupported columns is not lost because it is saved to the image copies.

  The following data types, on the contrary, lead to a failure of the archiving operation:

  - BLOB
  - CLOB
  - DBCLOB
  - XML
- If the GET_ACCEL_ARCHIVE special register (or ZPARM) has been set to the value YES, and a query addresses tables whose data has been archived to different accelerators, the query fails.
- For an archiving operation to succeed, a shared lock is required on the involved tables or partitions. Long-running transactions, such as database queries, might prevent the lock from being obtained. Make sure that long-running transactions are finished before you start archiving.

Archiving partition or table data with the High-Performance Storage Saver

The functionality of the High-perfomance Storage Saver (HPSS) is fully integrated into IBM Db2 Analytics Accelerator Studio. See how to archive the data of partitions or entire tables from the **Accelerator view**.

Before you begin

1. Make sure that you have read the introduction to the HPSS and the list of restrictions in the superordinate topic.
2. For the creation of the image copies, the AQTENV data set must be referenced by the Workload Manager (WLM) environment that has been set up for the IBM Db2 Analytics Accelerator for z/OS stored procedures, and a user with sufficient access rights must have set at least one of the following environment variables in this data set (a z/OS systems programmer or database administrator usually has the required access rights):
   - AQT_ARCHIVE_COPY1
   - AQT_ARCHIVE_COPY2
   - AQT_ARCHIVE_RECOVERYCOPY1

- AQT_ARCHIVE_RECOVERYCOPY2

These variables determine how many local image copies and recovery image copies are created for each archived partition and how the data sets are named. If you set a variable, a corresponding data set is created when partition data is archived. At least one of the variables must be set. Default values do not exist. Each value is a template specification as used in the Db2 TEMPLATE utility, for example:

```
AQT_ARCHIVE_COPY1 = &USERID..&DB..&TS..P&PART..&UNIQ.
```

where

&USERID.
ID of the user who runs SYSPROC.ACCEL_ARCHIVE_TABLES

&DB.
Name of the database that a partition resides in

&TS.
Name of the table space that the partition resides in

&PART.
Identifier of the (physical) partition. The letter P in the example is a text or string constant used as a prefix. This is required because &PART. resolves to a numeric value like 00001, and this not valid for qualifiers in a data-set name.

&UNIQ.
Causes the creation of a unique identifier

The template specification in the example could, for instance, result in the following image copy name:

```
BCKE.V4L1.BCKERTSE.CKRANGE3.P00001.D72R4KHN
```

- All template variables that are documented for the Db2 COPY utility can be used, with the exception of &SEQ (&SQ), &LIST (&LI), and &DSNUM.
- The chosen variables must ensure the uniqueness of image-copy data-set names. It is therefore recommended that you use at least the &PART. and &UNIQ. template variables.
- Templates must resolve to valid z/OS data set names.
- The template data-set names that you use must have been mapped to suitable data classes in the DFSMS.

Important: The AQT_ARCHIVECOPY_HLQ environment variable that was used with earlier versions of the HPSS is deprecated. Remove it from the AQTENV data set.

3. The HPSS stored procedure SYSPROC.ACCEL_ARCHIVE_TABLES creates image copies whose size might exceed 65535 tracks on direct access storage devices (DASDs). Therefore, a z/OS systems programmer must define and use a data class in the storage management subsystem (SMS) that supports larger sequential data sets and must associate this data class with the range of template data-set names for image copies. Data set types that can be used to this end are shown in Table 28 on page 133.

Table 28. Suitable data-set types for HPSS image copies and corresponding SMS data-class settings

Data-set type	Settings in SMS data class	Properties of data-set type
Extended-format data set	• `Data Set Name Type = EXT` • One of the following extension sub-parameter settings: – `Ext subparameter = R` (required) – `Ext subparameter = P` (preferred) • `Volume Count > 1`	• Requires multiple volumes • Maximum volume size still 65535 tracks
Large-format data set	`Data Set Name Type = LARGE`	• Multiple volumes possible, but not required • Volume sizes of more than 65535 tracks possible
Extended-attribute data set	`EATTR = OPT`	• Data sets must be stored on an extended-address volume (EAV) with at least 65521 cylinders. • Multiple volumes possible, but not required. A single volume might be sufficient.

To associate a suitable data class with the range of template data-set names, the z/OS systems programmer must create or modify a corresponding automatic class selection routine (ACS routine). The following example shows an excerpt of such an ACS routine, in which it has been assumed that the value of AQT_ARCHIVE_COPY1 starts with `IDAAHPSS`, as in `IDAAHPSS.&USERID..&DB..&TS..P&PART..&UNIQ.`:

```
FILTLIST LARGE INCLUDE(IDAAHPSS.**)
...
WHEN (&DSN = &LARGE) DO SET &DATACLAS = 'LARGE' ...
```

Tip: You can also define the ACS routine in such a way that the image copies created by the HPSS are stored on tape devices so that space can be saved on more expensive storage devices.

4. After setting the environment variables mentioned in step “2” on page 131, the Workload Manager (WLM) environment for the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure must be refreshed. Make sure that this has been done.
5. If you intend to archive partitions from a Db2 data sharing group, make sure that the following conditions apply:
   a. The volumes or devices for the storage of image copies must be available for all members of the data sharing group.
   b. The values of the environment variables mentioned in step “2” on page 131 must be identical on all members of the data sharing group.
6. Nonpartitioning indexes might slow down the archiving process. If more than just a few small to mid-sized nonpartitioning indexes are defined for a table, but the partitions to be archived are not many and rather large, then consider dropping the indexes before archiving. Re-create the indexes thereafter.
7. If you also use the incremental update function, check the setting of the `ONUTILITYACTION` keyword in the IBM InfoSphere Data Replication for z/OS configuration. The setting `ONUTILIYACTION=IDLE`

leads to a conflict because IDLE suspends the propagation of data changes for the entire table, and not just the partitions that have been archived by the HPSS. Hence the parts that have not been archived will not be synchronized anymore. To avoid this conflict, set the keyword to the value IGNORE.

About this task

The archiving task is carried out by the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, follow the appropriate **Related reference** link at the end of this topic.

Note: It might take a long time to archive many partitions. If you use IBM Db2 Analytics Accelerator Studio to invoke the ACCEL_ARCHIVE_TABLES stored procedure, IBM Db2 Analytics Accelerator Studio must remain open until the task has been completed. Otherwise, the stored procedure aborts, in which case it might be difficult to recover from the errors that were caused by the abortion. To avoid this problem, invoke the stored procedure from a TSO batch job. You can also repeatedly archive a smaller set of partitions at a time. Before you invoke the function, make sure that you have enough disk space for the image copies.

Procedure

1. In IBM Db2 Analytics Accelerator Studio, open the **Accelerator view** of the accelerator that you want to use for archiving.

   Note: You can only archive partition or table data to a regular accelerator. Therefore, you must open the **Accelerator view** of a regular accelerator.
2. If the tables are not yet defined as accelerator-shadow tables, click **Add** to start the **Add Tables** wizard and define these tables.
3. If necessary, refresh the **Accelerator view**.
4. To archive the data of a range-partitioned table or partitions of this table, expand the appropriate schema node and select the table name in the **Name** column.
5. On the toolbar of the **Accelerator view**, select **Storage Saver** > **Move Partitions to Accelerator**.
6. In the **Move Storage Saver Partitions to Accelerator** window, you can select a partitioning key or individual partitions.
   - To select the partition data to be archived with the help of a partitioning key, leave **Move all partitions up to and including the following limit key** selected and choose the proper key from the drop-down list.

     The drop-down list is filled with the values of the LIMITKEY column in the SYSIBM.SYSTABLEPART catalog table. The values are sorted by the LOGICAL_PART column in SYSIBM.SYSTABLEPART and appear in descending order. Mostly, LIMITKEY is a column of the type DATE or DATE/TIME, which allows you to select partitions whose data covers a certain timeframe. The key that you use might be different, and so might be the choice in the drop-down list.
   - To manually select the partitions whose data you want to archive, click **Manually select partitions** and select the appropriate check boxes in front of the (logical) partition names. The names of partitions whose data has already been archived are grayed out. To select the entire table, click **Select All**. Clicking **Deselect All** deselects all table partitions, which allows you to correct your choice.
7. Click **OK**.

What to do next

Check with your database operators (the people running queries) whether the archived data needs to be accessed by their queries. Whether to include or exclude the archived data can be controlled by using the **GET_ACCEL_ARCHIVE** parameter (ZPARM) or the SET CURRENT GET_ACCEL_ARCHIVE special register. The parameter or special register can be set to the value YES or NO. If you set it to YES, the archived data will be available for queries.

The **GET_ACCEL_ARCHIVE** parameter is set on the DSN6SPRN panel of a Db2 subsystem. The systems programmer or database administrator needs to enable option 2.

The SET CURRENT GET_ACCEL_ARCHIVE special register can be prepended to the SQL script for a query.

Example: Suppose you have a table called ORDERS, which contains a history of purchase transactions. This table is partitioned by time and contains a partition for each quarter. You keep ten years of history, but only a small subset of the queries accesses older data. Most queries use a range predicate, which restricts the query to data of the current year. You decide to archive all data that was collected in the previous years. For the queries that access the current year's data only, you need not set the SET CURRENT GET_ACCEL_ARCHIVE special register because these queries need not “see” the archived data. They can run on the accelerator or in Db2 for z/OS, depending on the SET CURRENT QUERY ACCELERATION special register. For those queries that need to access older data, you specify SET CURRENT GET_ACCEL_ARCHIVE=YES. For the latter type of queries, you must also set the SET CURRENT QUERY ACCELATION special register to a value other than NONE. The query cannot run in Db2 for z/OS because Db2 does no longer contain the data.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Impact of special register settings

It has been said that a query cannot be accelerated without setting the SET CURRENT QUERY ACCELERATION special register to a value other than NONE. It has also been said that the GET_ACCEL_ARCHIVE or the SET CURRENT GET_ACCEL_ARCHIVE special register determines whether data that was moved with the help of the high-performance storage saver (HPSS) is included in a query. The table in this topic explores all combinations of these settings for qualifying and nonqualifying queries and briefly describes the consequences.

If a query can neither be executed in Db2 for z/OS, nor on an accelerator, the following message is displayed, which starts with SQL code -4742:

```
-4742 The statement cannot be executed by Db2 or in the accelerator (REASON <code>)
```

where <code> stands for the numeric reason code. For a list of the reason codes, follow the link at the end of this topic.

Table 29. Impact of special register settings on the query execution, taking the query qualification into account

Setting of SET CURRENT QUERY ACCELERATION	Setting of GET_ACCEL_ARCHIVE or SET CURRENT GET_ACCEL_ARCHIVE	Query qualifies for acceleration?	Consequence
NONE	NO	No	The query is executed in Db2.
NONE	NO	Yes	The query is executed in Db2.
NONE	YES	No	The query fails and SQL code -4742 is returned.
NONE	YES	Yes	The query fails and SQL code -4742 is returned.
ENABLE	NO	No	The query is executed in Db2.
ENABLE	NO	Yes	The query is accelerated. Data of moved tables and partitions is excluded. If the query fails on the accelerator, SQL code -901 is returned to the calling application.
ENABLE	YES	No	The query fails and SQL code -4742 is returned.

Table 29. Impact of special register settings on the query execution, taking the query qualification into account (continued)

Setting of SET CURRENT QUERY ACCELERATION	**Setting of GET_ACCEL_ARCHIVE or SET CURRENT GET_ACCEL_ARCHIVE**	**Query qualifies for acceleration?**	**Consequence**
ENABLE	YES	Yes	The query is accelerated. Data of moved tables and partitions is included, provided that this data resides on the accelerator that the query is directed at. If the query fails on the accelerator, SQL code -901 is returned to the calling application.
ENABLE WITH FAILBACK	NO	No	The query is executed in Db2.
ENABLE WITH FAILBACK	NO	Yes	The query is accelerated. Data of moved tables and partitions is excluded. If the query fails on the accelerator, an attempt is made to run the query in Db2.
ENABLE WITH FAILBACK	YES	No	The query fails and SQL code -4742 is returned.
ENABLE WITH FAILBACK	YES	Yes	The query is accelerated. Data of moved tables and partitions is included, provided that this data resides on the accelerator that the query is directed at. If the query fails, SQL code -4742 is returned.
ALL	NO	No	The query fails and SQL code -4742 is returned.
ALL	NO	Yes	The query is accelerated. Data of moved tables and partitions is excluded. If the query fails on the accelerator, SQL code -901 is returned to the calling application.
ALL	YES	No	The query fails and SQL code -4742 is returned.
ALL	YES	Yes	The query is accelerated. Data of moved tables and partitions is included, provided that this data resides on the accelerator that the query is directed at. If the query fails on the accelerator, SQL code -901 is returned to the calling application.
ELIGIBLE	NO	No	The query is executed in Db2.
ELIGIBLE	NO	Yes	The query is accelerated. Data of moved tables and partitions is excluded. If the query fails on the accelerator, SQL code -901 is returned to the calling application.
ELIGIBLE	YES	No	The query fails and SQL code -4742 is returned.

Table 29. Impact of special register settings on the query execution, taking the query qualification into account (continued)

Setting of SET CURRENT QUERY ACCELERATION	Setting of GET_ACCEL_ARCHIVE or SET CURRENT GET_ACCEL_ARCHIVE	Query qualifies for acceleration?	Consequence
ELIGIBLE	YES	Yes	The query is accelerated. Data of moved tables and partitions is included, provided that this data resides on the accelerator that the query is directed at. If the query fails on the accelerator, SQL code -901 is returned to the calling application.

Important: A query that would be executed in Db2 for z/OS according to the special register or parameter settings in Table 29 on page 135 fails with SQL code -4742 if the query references data that can only be found on the accelerator because it has been moved.

Related concepts

EXPLAIN information
After creating the necessary EXPLAIN tables, you can analyze queries by invoking the Db2 EXPLAIN function. The analysis shows whether a query can be accelerated, indicates the reason for a failure, and gives a response time estimate. The outcome of the analysis can also be visualized in an access plan graph.

Restoring archived partitions

Restoring archived partition data might become necessary if you must add or update data in the original Db2 partition. It might also become necessary if an error in IBM Db2 Analytics Accelerator for z/OS has led to the loss of archived data. Restoring archived data is required also after changing the schema of a Db2 table (for example, adding a column).

About this task

In the context of archived partition data, the restore operation is typically just a single operation in a series of actions that must be taken. For example, adding a column to a Db2 table involves the following steps:

1. Restoring the archived partitions to Db2
2. Removing the corresponding accelerator-shadow table
3. Making changes to the schema, that is, submitting the `ALTER TABLE ADD COLUMN` statement
4. Redefining the accelerator-shadow table
5. Archiving the partition data once again

What you need to know about the restore function:

- The restore function uses the last Db2 image copy to recover the data. If that image copy is damaged, the function automatically uses the next image copy in backward chronological order.
- All indexes are rebuilt.
- A CHECK DATA operation is carried out if needed.
- An archived partition is not deleted from the accelerator, but kept as a regular partition for query acceleration.
- All image copies and entries in the Db2 catalog table SYSIBM.SYSCOPY are kept.
- The status information of tables in the ARCHIVE column of the SYSIBM.SYSACCELERATEDTABLES catalog table is restored if no further copies exist on any accelerator. That is, the values `'A'` or `'C'` are reset to `' '`.

- If a restore operation fails for some reason, you can resume it if you run the function again on the same table or partitions.
- Depending on the amount of data to be restored, the operation might need up to several hours to complete.

  ⚠ **Attention:** Do not sever the connection to the Db2 subsystem during that time.
- If you have archived the same partition more than once on different accelerators, you can run a restore operation for this partition from each of the accelerators. This is not prohibited. However, the Db2 data is not updated anymore after the first restore because the system treats it like an already restored partition. So if the data is different on the accelerators, carefully decide from which accelerator you want to restore first.
- Bear in mind that queries might run longer and might return more copious results after a restore because the formerly archived partitions are no longer excluded from the queries.

The task of restoring partition or table data to Db2 for z/OS is carried out by the SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the corresponding article in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. You find a link at the end of this topic.

Procedure

1. Check whether the tables that the partitions belong to have been enabled for incremental updates. If so, disable incremental updates.

   You might come across this situation if just a subset of all the partitions of a table was archived. If unarchived partitions remain on the accelerator, the relevant table can be enabled for incremental updates. However, a restore operation fails if this has been done.
2. In IBM Db2 Analytics Accelerator Studio, open the **Accelerator view** of the accelerator that you want to restore data from.
3. If necessary, refresh the **Accelerator view**.
4. On the toolbar of the **Accelerator view**, select **Storage Saver** > **Restore Partitions to Db2**.
5. In the **Restore Storage Saver Partitions to Db2** window, you can select tables and partitions to be restored.

   Select the check box in front of a table name to restore the data of an entire table (all partitions). Select the appropriate check boxes in front of (logical) partition names to restore just the data of the selected partitions. To select all tables and partitions that can be restored, click **Select All**. Clicking **Deselect All** resets a previous selection, so that you can start over.

   When you highlight a partition name, the **Image copies** section displays the names of the image copies (data set names) that will be used to restore the partition.
6. Click **OK**.

What to do next

The recovery of partitions invalidates the statistics in the SYSIBM.SYSTABLESPACESTATS catalog table. This leads to a situation in which table metrics, such as the partition size, are not reported anymore. To avoid this, run the following Db2 command to refresh the statistics in the catalog table:

```
RUNSTATS TABLESPACE <database-name>.<table-space name> SHRLEVEL REFERENCE
```

where *<database-name>* is the name of the database and *<table-space name>* is the name of the table space.

Note: This step might take up to several hours to complete. If you plan to archive the partition data again shortly, you can omit it.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

How to handle the RBA and LRSN conversion in Db2 11 for z/OS

In Db2 11 for z/OS, the relative byte address (RBA) and the log record sequence number (LRSN) were changed from a 6-byte format to a 10-byte format. This has implications if you have archived partition data with the high-performance storage saver (HPSS). In Db2, the migration from the old format to the new requires a REORG operation. However, the empty partitions that remain after HPSS archiving cannot be reorganized because they were set to persistent read-only (PRO) state.

To maintain the compatibility of archived tables and partitions, follow the series of steps in the following table that best suits your needs:

Table 30. Steps to maintain compatibility

1. Remove the PRO state with the help of the Db2 REPAIR utility. 2. Run the REORG operation. 3. Reset the PRO state. This is the recommended solution. It saves you a lot of time because you avoid to restore data. Note that you can restore the data at a later time into already converted tables and partitions even if the image copies were taken before the RBA and LRSN conversion. A further REORG operation will be needed after that.	1. Restore the data of the archived tables and partitions from the existing image copies. 2. Run the REORG operation. 3. Archive the data again with the HPSS. If you do not want a coexistence of data in different RBA and LRSN formats, then this solution is to be preferred. Bear in mind that this will occupy system resources for a much longer time.

Federated access

The term *federated access* or *federation* describes the ability to run accelerated queries against tables that do not belong to or originate from the Db2 subsystem that issues the query.

Normally, you can only accelerate queries if a Db2 subsystem is paired with an accelerator and if the tables to be queried exist in that Db2 subsystem. This can be regarded as a security feature; a second Db2 subsystem paired with the same accelerator cannot access tables in the first Db2 subsystem and vice versa.

However, there are situations in which one would like to lift this restriction: Db2 subsystems are often organized alongside geographical boundaries, temporal boundaries, or areas of business.

Geographical boundaries translate into a segmentation of the table data by region. For example, a company does business in the U.S. and in Asia; the tables in subsystem A contain the sales figures for the U.S.; the tables in subsystem B the sales figures for Asia.

If data is temporally segmented, the tables in each subsystem cover different time spans. For example, the tables in subsystem A contain the data related to recent business activities, reaching three months from the present time into the past. Db2 subsystem B, on the other hand, contains business data that is older than three months.

If data is segmented by areas of business, the organization of the database follows a company's key business areas or its product portfolio. Assume that an organization runs a chain of gas stations and also produces cosmetics (both business areas are based on oil). Subsystem A contains the data of the gas station business; subsystem B the data of the cosmetics business.

Frequently, the aim of analytical queries is to capture trends across regions, time lines, or business areas. In accordance with the previous examples, one would like to find out how the business in the U.S. develops against the business in Asia. How do recent sales figures compare with those of the past? What are the growth rates in the gas station business as opposed to those in the cosmetics sector?

Queries that answer these question require access to different data sources. If the data is segmented by the organization of the database, access must be extended to tables in other Db2 subsystems, and this is what the *federated access* feature is about.

An access extension to tables in a Db2 subsystem is achieved through the following steps:

1. A user with sufficient rights on the Db2 subsystem that owns the tables (remote Db2 subsystem) grants the right to access a set of tables to a Db2 subsystem that normally cannot access these tables (called the referencing Db2 subsystem).
2. A user with sufficient rights on the referencing subsystem runs a command to create reference tables. These reference tables contain metadata and pointers to accelerator-shadow tables on the accelerator. They are implemented as accelerator-only tables on the shared accelerator, with corresponding entries in the catalog of the referencing Db2 subsystem.
3. Queries from the referencing Db2 subsystem can now access these referencing accelerator-only tables.

See the following figure:

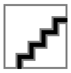


Figure 13. Federated access diagram

It is possible for the administrator of the owning Db2 subsystem to revoke the access privileges at any time.

It is also possible (for the administrator of the referencing Db2 subsystem) to remove the reference tables. This is mostly done when the data in these tables has become obsolete. To obtain results on more recent data, the administrator might want to delete the existing reference tables and create new ones.

Granting access to remote tables

As an administrator or user with sufficient access rights on a Db2 subsystem, you can grant the right to access tables in the subsystem to another (referencing Db2 subsystem). The granted permission includes the right to create reference tables from the referencing Db2 subsystem.

Before you begin

- The owning (remote) Db2 subsystem and the referencing Db2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both Db2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_GRANT_TABLES_REFERENCE stored procedure on the owning (remote) Db2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with the owning (remote) Db2 subsystem.

   Mind that although the referencing Db2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables in the owning Db2 subsystem.
2. On the toolbar of the **Tables** section, click **Federation** > **Grant Access**.
3. In the **Db2 Tables** window, select the tables that you want to give access to, that is, accelerator-shadow tables or ordinary accelerator-only tables in the owning (remote) Db2 subsystem.
4. Click **Next**.
5. In the **Db2 Subsystems** window, you see a list of the Db2 subsystems that are connected to the same accelerator. Select one or more referencing Db2 subsystems, that is, subsystems that should have access to the selected tables.
6. Click **Finish**.

Results

The administrator of the referencing Db2 subsystem can now create reference tables for the tables that she or he has been given access to.

Creating reference tables

Running the **Create Reference Tables** function in IBM Db2 Analytics Accelerator Studio, you create accelerator-only tables that contain metadata and pointers to accelerator-shadow tables on the shared accelerator, provided that access to these tables has been granted before.

Before you begin

- The owning (remote) Db2 subsystem and the referencing Db2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both Db2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_CREATE_REFERENCE_TABLES stored procedure on the referencing Db2 subsystems.

About this task

The reference tables are used when a query from a referencing Db2 subsystem includes the names of tables on the remote Db2 subsystem. The query thus goes against the reference tables. The metadata and pointers in the reference tables ensure that the right data in the accelerator-shadow tables or the (regular) accelerator-only tables is queried. The accelerator-shadow tables or accelerator-only tables that actually contain the data are the result of **Add Tables** and **Load Tables** operations from the owning or remote Db2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with a referencing Db2 subsystem.

   Mind that although the owning or remote Db2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables that access has been granted to.
2. On the toolbar of the **Tables** section, click **Federation** > **Create Reference Tables**.
3. In the **Db2 Subsystems** window, you see a list of Db2 subsystems that qualify as referencing Db2 subsystems, that is, subsystems that will query the remote tables. Select one or more of these subsystems.
4. Click **Next**.
5. In the **Db2 Tables** window, you see a list of tables on the accelerator. Select the tables for which you want to create reference tables by selecting the appropriate check boxes. To select all tables that belong to a schema, select the check box in front of the schema name.

   If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.

   To the right of the **Name like** field, you also find controls (icons) to do the following:

   a. Select or deselect all tables

   b. Expand or collapse all schemas
6. Selecting the optional check box **Grant access before creating reference tables** allows you to complete the required grant step and the creation of the reference tables in one go.

   However, you can only use this check box if your user ID has the authorization to run the SYSPROC.ACCEL_GRANT_TABLES_REFERENCE stored procedure on the owning (remote) Db2 subsystem.
7. Click **Finish**.

Results

Reference tables for the selected tables are created on the shared accelerator and corresponding entries are written to the catalog of the referencing Db2 subsystem. The new tables can be queried from the referencing Db2 subsystem.

Revoking table access

Run the **Revoke access** function to withdraw the right to access remote tables from a referencing Db2 subsystem.

Before you begin

- The owning (remote) Db2 subsystem and the referencing Db2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both Db2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_REVOKE_TABLES_REFERENCE stored procedures on the owning (remote) Db2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with the owning (remote) Db2 subsystem.

   Mind that although the referencing Db2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables in the owning Db2 subsystem.
2. On the toolbar of the **Tables** section, click **Federation** > **Revoke Access**.
3. In the **Db2 Subsystems** window, you see a list of the Db2 subsystems that are paired with the same accelerator. Select one or more referencing Db2 subsystems that should no longer be allowed to access remote tables in the owning Db2 subsystem.
4. Click **Next**.
5. In the **Db2 Tables** window, select the tables that you want to withdraw access from. Select by clicking the appropriate check boxes. To select all tables that belong to a schema, select the check box in front of the schema name.

   If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.

   To the right of the **Name like** field, you also find controls (icons) to do the following:

   a. Select or deselect all tables
   b. Expand or collapse all schemas
6. Click **Finish** when your selection is complete.

Results

Access rights to the selected tables are revoked. Even though the corresponding reference tables are not deleted, the referencing Db2 subsystem can no longer query the tables that the reference tables point to. Attempts will be answered with an authorization error.

Removing reference tables

Run the **Remove reference tables** function to delete reference tables (referencing accelerator-only tables) from an accelerator.

Before you begin

- The owning (remote) Db2 subsystem and the referencing Db2 subsystem must share (be paired with) a common accelerator.

- Database connection profiles must exist for both Db2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_REMOVE_REFERENCE_TABLES stored procedure on the referencing Db2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with the referencing Db2 subsystem.

   Mind that although the owning (remote) Db2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables in the referencing Db2 subsystem.
2. On the toolbar of the **Tables** section, click **Federation** > **Remove Reference Tables**.
3. In the **Db2 Subsystems** window, you see a list of the Db2 subsystems that are paired with the same accelerator. Select one or more Db2 subsystems containing reference tables.
4. Click **Next**.
5. In the **Db2 Tables** window, you see a list of the tables on the accelerator. Select the reference tables that you want to remove by clicking the appropriate check boxes. To select all tables that belong to a schema, select the check box in front of the schema name.

   If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.

   To the right of the **Name like** field, you also find controls (icons) to do the following:

   a. Select or deselect all tables

   b. Expand or collapse all schemas
6. Click **Finish**.

Results

The selected reference tables (referencing accelerator-only tables) are removed from the accelerator. Corresponding entries are removed from the catalog of the referencing Db2 subsystem.

Updating IBM Db2 Analytics Accelerator Studio

Follow the instructions here to download and install an upgrade of IBM Db2 Analytics Accelerator Studio.

About this task

To download a refresh pack for IBM Db2 Analytics Accelerator Studio, a workstation with an internet connection is required.

If IBM Installation Manager is installed on a workstation without an internet connection, the program cannot automatically locate the repositories for updates of already installed products or additional plugins. Therefore, an administrator has to download the update repository to a central location that users without an internet connection have access to.

Workstation users without an internet connections can then let their local IBM Installation Manager installations point to the repository in the central location or copy the entire installation package from the central location to their workstations and go on from there.

You can use the newer versions of IBM Db2 Analytics Accelerator Studio with older versions of accelerators and stored procedures. However, the user interface might not show new functions until you update the stored procedures to the next higher or newest version.

Procedure

- To download a refresh pack for yourself or for others from a computer with an internet connection:

  a) Click the following link:

Select fixes: IBM Db2 Analytics Accelerator Studio (All releases, All platforms)

b) Select the proper refresh pack and click the **Continue** button above the list.

c) Sign in with your IBM ID or click **Create IBM id** and follow the instructions to create an IBM ID first.

d) Select your preferred download method:

- **Download using Download Director** (requires Java enabled browser)
- **Download using bulk FTP**
- **Download using your browser (HTTPS)**

e) Leave the check box **Include prerequisites and co-requisite fixes (you can select the ones you need later)** selected.

f) In the **View and accept terms** window, click **I agree** to agree to the terms and conditions.

g) Click **Download now** and follow the instructions.

- To update an IBM Db2 Analytics Accelerator Studio installation:

a) Start IBM Installation Manager.

b) Update IBM Db2 Analytics Accelerator Studio by using one of the following procedures:

- The following approach works if you can still connect to a previously accessed IBM Installation Manager repository, and if that repository has been updated (contains the update you want):

  a. Click **Update**.

  b. Follow the instructions in the wizard. Deselect components that you do not want to update.

- If the first method did not work, but you can access the downloaded refresh pack (see step “7” on page 144) on your company's network or on your local workstation:

  a. From the main menu of IBM Installation Manager, select **File** > **Preferences** > **Repositories**.

  b. Click **Add Repository**.

  c. Click **Browse** and navigate to the directory that contains the refresh pack, either in the central location, or on your local workstation.

  d. Select the refresh pack (compressed archive) and click **Open**.

  e. Click **OK** twice to return the main window of IBM Installation Manager.

  f. Click **Install**.

  g. Follow the instructions in the installation wizard.

Installing updates

Refer to the appropriate sections to update individual components or migrate from one version to another.

It is rarely necessary to update all components. However, if you must update more than one component, follow the suggested order.

Order in which to update components

1. IBM Db2 Analytics Accelerator software. When in doubt: It is better if the accelerator software is at a newer level than the stored procedures. See *Updating IBM Db2 Analytics Accelerator software* in the *Installation Guide*.
2. Prerequisite PTFs for Db2 11 for z/OS or Db2 12 for z/OS. See *Installing prerequisite PTFs for Db2 for z/OS* in the *Installation Guide*.
3. Stored procedures. See *Updating the IBM Db2 Analytics Accelerator stored procedures* in the *Installation Guide*.
4. Db2 Analytics Accelerator on Z. See *Updating Db2 Analytics Accelerator on Z* in the *Installation Guide*..

5. IBM Db2 Analytics Accelerator Studio. Actually, the order does not matter for IBM Db2 Analytics Accelerator Studio. You can update it any time. See "Updating IBM Db2 Analytics Accelerator Studio" on page 143

Updating IBM Db2 Analytics Accelerator software

These are software components that run on the accelerator hardware, that is, the IBM Integrated Analytics System. This chapter does not apply to updates for Db2 Analytics Accelerator on Z. Components on the IBM Integrated Analytics System are the IBM Db2 Analytics Accelerator, the SQL engine, and the operating system of the accelerator.

Installing update packages for the accelerator

IBM Db2 Analytics Accelerator for z/OS software includes specific libraries and other code to be installed on the IBM Integrated Analytics System.

Before you begin

Make sure that the following conditions apply:

- Make sure that IBM Db2 Analytics Accelerator for z/OS has been successfully connected to Db2 for z/OS.
- IBM UNIX System Services is installed on your z/OS data server.
- You have a user ID and a password to log on to your z/OS data server. The user ID has read access to the z/OS UNIX file system (z/FS).
- The user ID is authorized to run the SYSPROC.ACCEL_UPDATE_SOFTWARE2 stored procedure.

About this task

The entire installation or update procedure consists of the following steps:

1. Downloading the fix pack or update package.
2. Transferring the update packages to the accelerator.
3. Activating the newly installed software.

This topic covers step "1" on page 145 only.

Procedure

Downloading the fix pack or update package. You have two options here:

- Download the fix pack or update package from IBM Fix Central to your workstation and then upload the package from your workstation using IBM Db2 Analytics Accelerator Studio. Place the checksum file for installation verification in the same directory on your workstation.

  There is a checksum file for every supported algorithm. Supported algorithms are SHA256, SHA1 and MD5. The corresponding files on IBM Fix Central are:

  - `sha1sums.txt`
  - `md5sums.txt`

  Any of these files contains one line for each package. Every line consists of two values. The first value is the hash used for verification; the second value is the corresponding package name.

  Important: During the process, a large global temporary table is created in Db2 for z/OS. This table requires up to 25 GB of disk space. Make sure that this amount of disk space is available.
- Download the fix pack or update package from IBM Fix Central and save it directly in the directory specified by the AQT_HOST_PACKAGE_DIRECTORY environment variable. To use this second option, you must:

  a. Log on to your z/OS data server using a remote client, such as IBM Personal Communications.

b. Set the AQT_HOST_PACKAGE_DIRECTORY environment variable to a directory path in the z/FS. Then save the update package in this directory. Place the checksum file for installation verification in the same directory (for details, see the previous description of the first download option).

Important:

- In the z/FS, the checksum files must be available in UTF-8 format. However, if you transfer the files from an FTP, client, this format is identified as a text format and the transfer mode will probably be set to *ascii*. You must change the transfer mode to *binary* because this guarantees that the file stays as it is. If *ascii* is used as the transfer mode, chances are that the files are converted to EBCDIC format. This leads to an error when the checksum files are used for verification purposes.
- An update packages requires up to 25 GB of disk space. Make sure that enough space is available in the z/FS directory that AQT_HOST_PACKAGE_DIRECTORY points to.

Important: The directory that AQT_HOST_PACKAGE_DIRECTORY points to requires multiple gigabytes (GB) of disk space. For example, if you work with multiple accelerators in a rolling upgrade scenario, you must have multiple instances of the installation packages in the file system. So make sure that the directory can accommodate all the files.

Results

Your local workstation or z/FS directory (as set by the AQT_HOST_PACKAGE_DIRECTORY environment variable) on your z/OS data server now contains the accelerator update package `*version*.tar.z`.

Transferring update packages for the accelerator

Transfer update packages for IBM Db2 Analytics Accelerator for z/OS on an IBM Integrated Analytics System by completing the steps in this section.

Before you begin

Make sure that the following conditions apply:

- You have downloaded suitable accelerator installation packages and their respective checksum files to a workstation that is connected to your z/OS data server, or such packages exist in a z/FS directory on that server. For the latter option, the AQT_HOST_PACKAGE_DIRECTORY environment variable must point to the z/FS directory, and the checksum files must have been transferred in *binary* mode.
- The value of the AQT_HOST_PACKAGE_DIRECTORY environment variable determines where IBM Db2 Analytics Accelerator Studio looks for software updates. If it is not set correctly, the packages cannot be located and thus cannot be transferred or installed.
- The user ID has read and write access to the directory that the AQT_HOST_PACKAGE_DIRECTORY environment variable points to and to all its subdirectories. You specify this variable in the AQTENV data set. The AQTENV data set must be referenced by the Workload Manager (WLM) environment that was set up for the IBM Db2 Analytics Accelerator for z/OS stored procedures. To avoid a setup with obsolete or wrong entries, use the sample AQTENV data set that comes with IBM Db2 Analytics Accelerator for z/OS Version 7.1.0.

About this task

You can transfer update packages from the z/OS UNIX file system (z/FS) to the accelerator, or from a workstation on which IBM Db2 Analytics Accelerator Studio is installed, and which is connected to the z/OS data server. The first method requires that the packages have been placed in the proper z/FS directory. To get them there, you can, for example, download packages from IBM Fix Central directly to the z/FS of the mainframe. To this end, you need a client application on the mainframe that is capable of handling FTPS (FTP over TLS) traffic. The download site at IBM Fix Central offers an option called *Download using bulk FTPS*. The second method requires that the update packages have been downloaded to the client workstation before. To actually use a transferred package, you must activate it. How to do this is described in a later topic. See the hint at the end of this description.

Installing a software update invokes the **`SYSPROC.ACCEL_UPDATE_SOFTWARE2`** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. Start IBM Db2 Analytics Accelerator Studio.
2. In the **Administration Explorer**, select the **Accelerators** folder.
3. In the Object List Editor on the right, double-click the accelerator.
4. If necessary, expand the **About** section.
5. In the **About** section, click the **Transfer updates** link.
6. In the **Transfer Updates** window, you can see all software packages that are available in the z/FS of your z/OS data server. Select the appropriate check boxes in the first column of the table to mark the packages that you want to transfer.

   Attention:

   - Make sure that you select the proper packages, that is, packages belonging to the release level that you want to upgrade to. The list in the **Transfer Updates** window might be confusing, especially if it also contains older packages. To find the correct package numbers, see the closing information or the release notes for the latest program temporary fix (PTF). You find the closing information or the release notes for a PTF on the support home page, in the category *Plan and install documentation* (see link under **Related information** at the end of this topic).
   - If the list contains too many packages so that selecting the proper ones becomes awkward, select unwanted package names and click **Remove**. This will just remove the packages from the list or view, but won't delete the package files from the z/FS or your client workstation.
7. Choose the appropriate method to transfer the selected packages:

   Transfer file from client

   To transfer one or more downloaded packages from the connected client workstation.

   Transfer file from IBM Z

   To transfer one or more packages from the z/FS of your z/OS data server.

   Note: If the verification message says `... package was deployed without integrity check ...`, the reason might be that the checksum files were not transferred in *binary* mode. See the second **Important** note in step “1” on page 145.

What to do next

To save space in the SMP/E target directory of the z/FS, consider deleting sub-directories that contain already transferred packages. Be careful, however, if you have more than one accelerator. Do not delete packages if these are still needed for other accelerators.

If the AQT_HOST_PACKAGE_DIRECTORY environment variable does not point directly to the SMP/E target directories but to a copy, you might also want to delete the copy.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Activating an updated accelerator software version

To use a recently transferred software package, you must first activate that package by selecting the version that the package contains.

Before you begin

- You cannot activate a software version as long as a IBM Db2 Analytics Accelerator Console window is open. Therefore, close all console windows. If you are not sure whether there are any open console windows, click **List Tasks** in the relevant **Accelerator** view of IBM Db2 Analytics Accelerator Studio and cancel all active console tasks from the **List Tasks** window.

- If incremental updates are enabled, stop these for each Db2 subsystem connected to the accelerator. To do this from IBM Db2 Analytics Accelerator Studio:
    1. In the **Administration Explorer** window, open the **Accelerators** folder.
    2. Select a relevant connection and open its **Accelerator** view.
    3. In the header section of the **Accelerator** view, look for **Replication:** and click **Stop**.
    4. Repeat the preceding steps for each Db2 subsystem connection to the same accelerator.

  Note: If you have connections to Db2 data sharing groups (DSGs) and prefer to run the SYSPROC.ACCEL_SET_REPLICATION stored procedure from a client command line: It is not necessary to stop incremental updates for each individual member. A DSG is treated like a single connection and stopping incremental updates for the primary member will also stop these for the other members.

About this task

- Earlier product versions allow you to switch between older and newer versions of the IBM Db2 Analytics Accelerator software. This is not possible with IBM Db2 Analytics Accelerator for z/OS Version 7.1.0. Once you have activated a particular version, you cannot go back to an older one.
- The application of an update affects all Db2 subsystems that are connected to an accelerator.
- The activation of a different accelerator software version causes a restart of the accelerator.

Activating a software update invokes the **SYSPROC.ACCEL_UPDATE_SOFTWARE** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

⚠ **Attention:**

- The activation of a new version of IBM Db2 Analytics Accelerator for z/OS software components might delete the query history from the accelerator if the new version introduces changes to the query history function.
- The sequence in which you activate the different software components is important. The sequence depends on the source and on the target version. To find the proper sequence for your particular update, see "Order in which to update components" on page 144.
- When the accelerator is restarted with a new accelerator software version, accelerated queries cannot be processed on that accelerator before it reaches the *online* state. The only way to avoid a delay is to use a second accelerator in a workload balancing setup. That is, a second accelerator which is connected to the same Db2 subsystem, and which contains the same set of tables. Note that the tables must be loaded and enabled.

Procedure

1. In the **Administration Explorer**, select the **Accelerators** folder.
2. In the Object List Editor on the right, double-click the accelerator.
3. In the **Accelerator** view, from the **Refresh** drop-down list in the upper right, select **Automatic off**.

   Otherwise, you might see warnings during the activation of the new software saying that the accelerator cannot be contacted.
4. If necessary, expand the **About** section.
5. In the **About** section, click the **Apply other software version** link.
6. In the **Apply Software Version** wizard, you can see all software packages on the accelerator that are currently available for the selected component. To read information about a particular version before you activate it, select the appropriate entry in the list. The information is provided in the **Details of selected version** text box at the bottom.
7. Activate a version by selecting the appropriate radio button in the **Switch To** column.
8. Click **Finish**.

Results

In some cases, migration processes require some time to finish, especially after a restart of the database container. It is therefore quite normal that, after you click the **Finish** button, the accelerator is not instantly ready to process queries. You can check the status of an accelerator at the top of the corresponding **Accelerator** view. The status must be *online* for an accelerator to process queries.

What to do next

To avoid a cluttered **Apply Software Version** window, you can remove packages from the accelerator that you no longer need.

For instructions on how to complete this task, follow the appropriate link under **Related tasks**.

Related information

IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Removing obsolete software packages from an accelerator

To free up space on an accelerator and reduce the number of versions that are displayed in the **Apply Software Version** window, you can remove packages that you no longer need.

About this task

The removal wizard can remove all packages that have been deployed with the help of the transfer wizard, that is:

- Accelerator software packages
- Database software packages

Note: You can only remove software packages from regular accelerators.

Procedure

1. Open the appropriate **Accelerator view**.
2. If necessary, expand the **About** section.
3. Click the **Remove** link in the server subsection.
4. In the **Remove Software Versions** window, select the software packages that you want to remove. To do so, select the appropriate check boxes in the first column.

   As a decision aid, you might want to read the information under **Details of selected version**, which is displayed for each package.
5. Click **Remove**.

What to do next

The removal wizard that is described in this topic can only remove packages from an accelerator. To free up space in the z/OS UNIX file system (zFS) of your data server, manually delete no longer needed IBM Db2 Analytics Accelerator for z/OS packages from the zFS. To do so, open a UNIX System Services shell and delete the files by using the `rm` command.

Opening a service request

You might run into a situation in which you have to contact IBM support because you cannot solve an issue by yourself. A reason might be that specific updates or patches are required to successfully complete maintenance or repair jobs on the accelerator. Such jobs are usually carried out by an IBM service engineer. Follow the steps here to open a service request (formerly called problem management record or PMR) and provide IBM support with the necessary information.

1. In IBM Db2 Analytics Accelerator Studio, enable tracing for the accelerator that you want to update. A trace level of DEFAULT is sufficient.

   For more information, see *Tracing* in the *IBM Db2 Analytics Accelerator Studio: User's Guide*.
2. Save the trace information to a file.

3. Open a service request at https://www.ibm.com/support/servicerequest/Home.action. Consider that IBM support needs some time to analyze your system, coordinate maintenance actions, and update components as required.
4. Attach the trace file to the service request.

Transferring maintenance updates

A task that is closely related to service requests is the transfer of maintenance updates. When you have submitted the service request, IBM support will tell you the proper download location and the names of the packages that you have to download. After finishing the download, transfer the update packages as a preparation for the IBM service personnel, who will help you install the updates on your IBM Integrated Analytics System.

Before you begin

Update packages must exist in the download directory for packages of this type in the z/OS UNIX file system (zFS). The target system must be able to access this directory.

Important: The AQT_HOST_PACKAGE_DIRECTORY environment variable points to this download directory. It specifies an absolute path (starting from the root directory). The AQT_HOST_PACKAGE_DIRECTORY environment is set in the <HLQSP>.SAQTSAMP(AQTENV) data set, where <HLQSP> is the chosen high-level qualifier for stored-procedure libraries. The value of this environment variable determines where IBM Db2 Analytics Accelerator Studio looks for updates. If it is not set correctly, the packages cannot be located and thus cannot be transferred to the IBM Integrated Analytics System. To avoid a setup with obsolete or wrong entries, use the sample AQTENV data set that comes with IBM Db2 Analytics Accelerator for z/OS Version 7.1.0.

About this task

In general, maintenance update packages are not installed automatically. They are just transferred to a directory on the IBM Integrated Analytics System, from where you must install them manually with the help of IBM support.

Transferring an update invokes the **SYSPROC.ACCEL_UPDATE_SOFTWARE2** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM Db2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

To start an update transfer process, follow the steps in this section.

Procedure

1. Start IBM Db2 Analytics Accelerator Studio.
2. In the **Administration Explorer**, select the **Accelerators** folder.
3. In the Object List Editor on the right, double-click the accelerator.
4. If necessary, expand the **About** section.
5. In the **About** section, click the **Transfer updates** link.
6. In the **Transfer Updates** window, you can see all software packages that are available in the z/FS of your z/OS data server. Select the appropriate check boxes in the first column of the table to mark the packages that you want to transfer.

   ⚠ **Attention:**

   - Make sure that you select the proper packages, that is, packages belonging to the release level that you want to upgrade to. The list in the **Transfer Updates** window might be confusing, especially if it also contains older packages. To find the correct package numbers, see the closing information or the release notes for the latest program temporary fix (PTF). You find the closing information or the release notes for a PTF on the support home page, in the category *Plan and install documentation* (see link under **Related information** at the end of this topic).

    - If the list contains too many packages so that selecting the proper ones becomes awkward, select unwanted package names and click **Remove**. This will just remove the packages from the list or view, but won't delete the package files from the z/FS or your client workstation.

7. Choose the appropriate method to transfer the selected packages:

   Transfer file from client

   To transfer one or more downloaded packages from the connected client workstation.

   Transfer file from IBM Z

   To transfer one or more packages from the z/FS of your z/OS data server.

   Note: If the verification message says `... package was deployed without integrity check ...`, the reason might be that the checksum files were not transferred in *binary* mode. See the second **Important** note in step "1" on page 145.

Installing maintenance updates

While updates for IBM Db2 Analytics Accelerator for z/OS and the databaseare installed automatically at the end of the transfer process or by using the **Apply** function, other updates have to be installed manually with the help of IBM support.

Before you begin

Make sure that the following conditions apply:

- You have opened a service request as described in "Opening a service request" on page 149.
- The IBM Integrated Analytics System must be fully operational. Hardware problems must not exist. Replace defective hard disks before installing the update.
- Remote access from z/OS OMVS to the accelerator via secure shell (ssh) must be enabled.
- IBM support will ask you to open or allow a screen-sharing session on your workstation. Make sure that this complies with the security guidelines of your organization or ask the responsible staff for a special permit.

About this task

An update installation requires you to stop the database container for approximately 30 minutes. The accelerator cannot be used during this time.

Procedure

1. Contact IBM support.

   IBM support will ask you for remote screen-sharing access to the client machine.

2. With IBM assistance, open a login shell to the accelerator.

   There are several ways to do that:

   - If an ssh server and an ssh client are installed in the z/FS of your IBM Z, and if server and client have been activated, you can connect with a network client application, such as PuTTY, from your client workstation to the accelerator by opening the following connections:

     a. From your workstation to the IBM Z server by ssh. Use your user ID and password to log in. See an example of an ssh login from a PuTTY window:

     ```
     login as: willy
     willy@boedca1's password:
     ```

     Important: If your z/OS version is V2R2 or higher, make sure that the address space for the Cryptographic Service Facility (CSF) has been started. Otherwise, the ssh connection to z/OS will fail.

     b. From the IBM Z to the accelerator by ssh. See the following example:

     ```
     ssh -p 2222 root@10.104.9.3
     ```

where:

2222
is the listening port on the accelerator

10.104.9.3
is the IP address of the accelerator

- If an ssh server is installed and active on your IBM Z, and if port-forwarding is enabled for that server, you can connect to the accelerator by an ssh tunnel, which looks like a single connection from your client workstation to the accelerator. To enable port-forwarding, set `AllowTcpForwarding yes` in the `/etc/ssh/sshd_config` file of your zFS and restart the ssh server for the change to take effect.

  Example: The commands for a tunnel connection would look similar to these:

  a. `ssh -L 10022:10.0.0.1:2222 user@systemz`

     where:

     -L 10022
     is the local port on the client workstation

     10.0.0.1
     is the IP address of the accelerator

     2222
     is the remote port or listening port on the accelerator

     user@systemz
     `user` is ID with which you log on to the UNIX System Services ssh server and `systemz` is the host name of that server.

  b. `ssh -p 10022 root@localhost`

     where:

     -p 10022
     is the local port on the client workstation

     root@localhost
     `root` is the user ID on the accelerator and `localhost` is the host name of the client workstation

Tip: IBM used to provide a tool package called IBM Ported Tools for z/OS. See:

https://www-03.ibm.com/systems/z/os/zos/features/unix/port_tools.html

The package does not exist anymore, but more specific packages with similar functionality are now offered by the Rocket software company (http://www.rocketsoftware.com/zos-open-source). These packages usually include an OpenSSH component, which contains an ssh client that you can use to connect from the UNIX System Services file system to the accelerator. Note that you only need the client component of OpenSSH for maintenance updates (called **ssh**).

The resulting login screen looks similar to this:

```
Using keyboard-interactive authentication.
Secure token for service password
(this token must be sent to IBM together with the serial# the first time you request
a service password):

MHu2cK9FLC2pIkyAEbOoyDxK1yU7UQO1VPyXVviVxwGRtAC3gBiXiio5YLXzQTCQRfL/TRyuG8zFBC2CkSe3
GXPrFjM9KgB41gEFYOoxSEhjXY0S89fKGa7E/M+/VCdnsfHX2k2GpBq0XpyVPps6o0Q==

Enter Service Password (Date: '20170506' Serial#: '7130001' Rev: '4'):
```

3. As you enter the command to log on to the machine, the secure token, which contains the encrypted machine password, is displayed on the screen. In the previous example, this is the cryptic string that

starts with MHu2cK. For the first service session, submit this token and the machine serial number to IBM support. IBM support will then generate a service password.

You have to submit the secure token only for the first service session. The token is stored in an IBM service database for subsequent requests. This database cannot be accessed from the internet. If a further service session is required, you just have to give IBM support the serial number of the machine.

4. IBM support will enter the service password for you. The password will be valid exclusively for your accelerator machine and for the current date only.
5. Navigate to the directory on the accelerator in which the transferred update files are stored.
6. To proceed, follow the instructions of IBM support.
7. Having installed the updates, delete the files that are not needed anymore.
8. Terminate the ssh session by entering `exit` at the command prompt.

Chapter 7. Status information about the system and the incremental update function

When the system has been properly configured, a heartbeat connection is established between the accelerator and the connected Db2 subsystems. Every 30 seconds, the heartbeat connection provides the Db2 subsystems with status information about the accelerator. You can view most of this information by using the `-DIS ACCEL` Db2 commands. Detailed information about specific events cannot be viewed in this way, but is written to the z/OS system log (SYSLOG), in the form of DSNX881I messages. This includes messages that are related to the incremental update function.

To establish a heartbeat connection, an accelerator must have been paired successfully with a Db2 subsystem. In addition, the accelerator must have been started for that particular subsystem by the `-START ACCEL` command or by use of the corresponding function in IBM Db2 Analytics Accelerator Studio.

To receive incremental-update-related event messages in a SYSLOG, IBM InfoSphere Data Replication for z/OS must have been installed and properly configured for use with the incremental update function of IBM Db2 Analytics Accelerator.

Support model

IBM Db2 Analytics Accelerator is a solution that consists of various hardware and software components. Each of these components might issue a DSNX881I message. If the message indicates a hardware or software problem, open a problem management record (PMR) with component ID (CompID) 5697AQT00.

Never open a service request or PMR for the IBM Integrated Analytics System. Always use the IBM Db2 Analytics Accelerator CompID, even if the problem is related to Db2 for z/OS or to the attached accelerator hardware.

Your service request or PMR will be routed internally to the proper support request queue. Make sure that the request contains an IBM Db2 Analytics Accelerator trace file that was obtained by using the **Save Trace** function in IBM Db2 Analytics Accelerator Studio. Such a trace file does not only contain software trace messages, but also a complete set of diagnostic hardware information.

The **Save Trace** window in IBM Db2 Analytics Accelerator Studio allows you to select an **Accelerator** and an **Appliance** option (apart from other options). It is important to notice the difference between the two:

Accelerator
Collects trace information from components running inside the Docker container

Appliance
Collects trace information from the physical system on which the Docker container is running

For details, follow the link to the description of the **Save Trace** function at the end of this topic.

Consider using automatic DSNX881I notifications. This function can inform administrators of critical events and create PMRs automatically.

DSNX881I message structure

Each DSNX881I message is made up of the following parts, which occur in the order indicated:

```
DSNX881I -<SSID> <MESSAGE-ID> <SEVERITY>
<ACCELERATOR_MESSAGE_COUNTER> (<ACCELERATOR-TIMESTAMP>)
ACCELERATOR-NAME(ACCELERATOR-IP) <MESSAGE-TEXT>
```

The placeholders have the following meaning:

SSID
Is the Db2 subsystem ID (SSID)

MESSAGE-ID

A numeric ID for the specific error message. This ID can be used for system monitoring.

Messages with IDs from 1 to 19 and from 21to 2999

Accelerator heartbeat (system state) messages.

Messsages with ID 20

These are incremental-update-related messages generated by IBM InfoSphere Data Replication for z/OS (CDC). They can be informational messages, warnings, or errors. Do not configure automatic notifications and service requests (PMRs) based on these messages, as the messages do not take the overall system state into account. In most cases, these messages are issued alongside messages in the range from ID3000 to ID3999. Refer to the latter type of messages because these contain proper instructions and do take the overall system state into account.

Message with IDs from 3000 to 3999

These are messages related to the incremental update function. In contrast to the ID20 messages, these messages are more specific or might ask for a concrete intervention (manual or automated) on your part. They can be used to trigger automatic notifications and PMRs.

SERVERITY

I

Information message

W

Warning message

E

Error message

ACCELERATOR_MESSAGE_COUNTER

An internal counter that increases with every additional error on the accelerator. If the text after the DSNX881I qualifier is longer than 255 characters, another DSNX881I message is issued. All messages belonging together will have the same <ACCELERATOR_MESSAGE_COUNTER> value. The <MESSAGE-TEXT> block of the each subsequent message contains a sequel to the information in the previous message.

ACCELERATOR-TIMESTAMP

The time when the error occurred on the accelerator. The internal clock of the accelerator is synchronized with the first Db2 subsystem that was connected to the accelerator.

ACCELERATOR-NAME

The name of the accelerator on which the error occurred.

ACCELERATOR-IP

The IP address of the accelerator on which the error occurred.

MESSAGE

A textual description of the error.

The length of a DSNX881I message does not exceed 255 characters. If more characters are needed, additional DSNX881I messages are written to the SYSLOG. If an LPAR contains multiple Db2 subsystems that are connected to the same physical accelerator, error messages are issued for every subsystem. That is, you see the same messages multiple times in the log, each time with a different subsystem ID (SSID).

If an accelerator is paired with a data sharing group (DSG), all members of the group can write messages to group's system logs (SYSLOGs), provided that the -START ACCEL command has been issued for all members. In this case, make sure that applications are in place that monitor the SYSLOGs. If all members of the DSG are located in the same logical partition (LPAR), there is only one SYSLOG to monitor (see Figure 14 on page 156). However, if the members are located in different LPARs, you need to monitor the SYSLOGs of all LPARs involved (see Figure 15 on page 157.

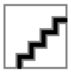


Figure 14. All members in the same LPAR

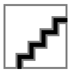


Figure 15. Members in two different LPARs

Note: It might look as if only one member writes messages to the SYSLOG, but this is actually a synchronization issue. If one member is always the first to issue a heartbeat request, then this member will receive all the messages and write these to the SYSLOGs. After that, the messages are deleted from the accelerator queue. The other members that send their heartbeat requests later will not receive these messages because the queue is empty. You might also see that only a few members write messages to the SYSLOG. This just means that the first member to send a heartbeat request is (always) found among this subset of members. The underlying mechanism is the same.

If an error occurs while an accelerator is in the *Stopped* state,that is, the `-STOP ACCEL` command has been issued before, an error message is stored on the accelerator. As soon as the accelerator becomes available again in Db2, the stored error messages are sent to the Db2 subsystem, provided that `-START ACCEL` has been issued for the subsystem, or, in case of a data sharing group, for at least one member of the group. It might happen that a DSNX881I message reports a past problem that has already been fixed.

For a list of message IDs, severity indicators, and message texts, see *Structure of DSNX881I Messages* (link under **Related information**).

Related tasks

Saving trace information
The **Save Trace** function saves the collected trace information. In the **Save Trace** window, which opens before the action is completed, you can specify or change the settings for the save operation.

Related information

Prerequisites and Maintenance for IBM DB2 Analytics Accelerator for z/OS Version 6.1

Structure of DSNX881I Messages

Chapter 8. Troubleshooting

In the following sections, you find descriptions of known IBM Db2 Analytics Accelerator for z/OS problems. The author and the development team have tried to provide a solution wherever possible. However, a solution might be unavailable because the cause of a problem cannot be clearly identified. This is mostly the case if multiple causes can lead to the same symptom. It can also be that a solution has not yet been found. In such cases, contact IBM support.

Important: During a customer session with IBM support, system parameters might be changed. For the parameter changes to take effect, a restart of an accelerator is often required. In such situations, the accelerator will be unavailable for operational tasks until it has come back online.

Firewall blocks operations because crucial ports are closed

If product functions are not executed, check your firewall settings. IBM Db2 Analytics Accelerator requires certain ports to be open. See the table in this topic.

Symptoms

- You cannot accelerate queries.
- You cannot run functions from the IBM Db2 Analytics Accelerator Console.
- Incremental updates do not work or IBM InfoSphere Data Replication for z/OS cannot be configured.
- You cannot open a secure-shell (ssh) connection to an accelerator.
- Services are unavailable or switches (routers) cannot be reached because ICPM (Internet Control Message Protocol) traffic is blocked.

Causes

Required networking ports are closed.

Resolving the problem

Table 31. Required networking ports for IBM Db2 Analytics Accelerator

Function or protocol	Required port	Direction
Query execution (DRDA protocol)	1400	Both ways
Communication between accelerator and stored procedures	1401	Both ways
IBM Db2 Analytics Accelerator Console	1600	Both ways
Secure DRDA (stunnel)	11400	Both ways
Secure communication between accelerator and stored procedures (stunnel)	11401	Both ways
Incremental update communication	11301-11350	Outbound (from z/OS to accelerator)
CDC configuration	5999	Inbound (generally from accelerator to z/OS; depends on the configuration of the CDC Capture Agent)

Table 31. Required networking ports for IBM Db2 Analytics Accelerator (continued)

Function or protocol	Required port	Direction
DB2 DDF port (for continuous incremental updates)	446 (default)	Inbound (from accelerator to z/OS)
Secure shell connection (ssh)	2222	Both ways
ICMP traffic	N/A	Both ways

IBM Db2 Analytics Accelerator Studio does not start

If IBM Db2 Analytics Accelerator Studio does not start, the reason might be that the name of the starting user contains a number sign (#).

Symptoms

IBM Db2 Analytics Accelerator Studio does not start.

Causes

At installation time, IBM Db2 Analytics Accelerator Studio creates a workspace directory in

`@user.home\IBM\Db2 Analytics Accelerator Studio 2.1\workspace`

where `@user.home` is the user's home directory, which is usually the same as the user name.

This means, if the user name contains a number sign (#), the workspace directory path contains a number sign, too. IBM Db2 Analytics Accelerator Studio cannot handle workspace directory paths that include number signs.

Resolving the problem

1. If the user name contains a number sign (#), open the file `datastudio.ini` in a text editor. This file is located in the installation directory of IBM Db2 Analytics Accelerator Studio.
2. Remove the following entries:

   ```
   -data
   @user.home\IBM\Db2 Analytics Accelerator Studio 2.1\workspace
   ```

3. Save and close the file.
4. The next time you start IBM Db2 Analytics Accelerator Studio, you are asked for the workspace directory path. This allows you to enter an alternative path that does not contain a number sign.

IBM Db2 Analytics Accelerator Studio on Linux does not start

IBM Db2 Analytics Accelerator Studio does not start from certain versions of Linux, particularly SUSE Linux Enterprise Server 10 with service pack 4.

Symptoms

An error message similar to the following is displayed:

```
/opt/tools/isaostudio//./jre/bin/javaw: symbol lookup error
/usr/lib/xulrunner-1.9.2.17/libxul.so: undefined symbol
gdk_screen_get_resolution
```

Causes

An updated version of a required library (`XULRunner`) is not compatible with IBM Db2 Analytics Accelerator Studio.

Resolving the problem

Use an earlier version of the XULRunner library. The problem also occurs in connection with IBM Rational® Performance Tester. You can apply the suggested solution analogously. See:

http://www.ibm.com/support/docview.wss?uid=swg21393436

Package not found when running a stored procedure from IBM Db2 Analytics Accelerator Studio

You receive a message saying that a package was not found when you try to run an IBM Db2 Analytics Accelerator for z/OS stored procedure from IBM Db2 Analytics Accelerator Studio.

Symptoms

You receive a message similar to this one:

```
SQL0805N  Package "<location>.NULLID.SYSSTAT.5359534C564C3031"
was not found. SQLSTATE=51002
```

Causes

The package has not been bound due to an IBM Db2 Analytics Accelerator Studio installation error.

Resolving the problem

Bind the package manually. The following methods can be used:

- From the Db2 command-line client:
  1. `cd <Db2-client-install-folder>\bnd`

     where `<Db2-client-install-folder>` is the fully qualified path to the installation folder of the Db2 command-line client.
  2. `db2 connect to <database-name>`

     where `<database-name>` is the database to which the stored procedure belongs.
  3. `db2 bind @db2cli.lst grant public`
- Using the Db2Binder utility from a Windows command-prompt:
  1. Adjust the following command as needed. Then press the Enter key.

     ```
     cd /d <idaa-studio-install-directory>
     \plugins\com.ibm.datatools.db2_2.1.403.v20120228_2105\driver
     ```

     where `<idaa-studio-install-directory>` is the drive and installation directory of IBM Db2 Analytics Accelerator Studio on your local workstation, for example `C:\Program Files\IBM\IBM Db2 Analytics Accelerator Studio 2.1`. The full name of the `com.ibm.datatools.db2_` directory changes with each new driver. So make sure that you choose the correct directory.
  2. Enter:

     ```
     ..\..\..\jre\bin\java -cp db2jcc4.jar;
     db2jcc4_license_cisuz.jar;
     db2jcc4_license_cu.jar com.ibm.db2.jcc.Db2Binder
     -url jdbc:db2://<server>:<port>/<location>
     -user <user-id> -password <password>
     ```

     where

     `<server>`
     Is the host name of the Db2 data server

     `<port>`
     Is the port on which the Db2 data server listens to JDBC requests

<location>
Is the unique name of the database server. An application uses the location name to access a Db2 database server. A database alias can be used to override the location name when accessing a remote server.

<user-id>
Is a user ID with the privilege of running the Db2 Binder utility

<password>
Is the password belonging to `<user-id>`

Long-running tasks are not completed

Tasks started from IBM Db2 Analytics Accelerator Studio are not completed if they need more than two minutes of time, as, for example, the loading of an accelerator-shadow table.

Symptoms

The progress indicator never reaches the end or runs in an endless loop. You never receive a message that reports the successful completion or failure of the task.

Causes

A Db2 timeout setting (IDLE THREAD TIMEOUT parameter) prematurely ends the process (DDF thread) before the task is completed. The default value of this parameter is 2 minutes.

Resolving the problem

Increase the Db2 timeout value. For instructions, follow the link at the end of this topic.

Related information

IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter)

SQL code -430 from IBM Db2 Analytics Accelerator for z/OS stored procedures

A stored procedure of IBM Db2 Analytics Accelerator for z/OS ends abnormally, and you receive an error message with SQL code `-430`.

Symptoms

IBM Db2 Analytics Accelerator for z/OS stored procedures end abnormally. In IBM Db2 Analytics Accelerator Studio, the **Administration Explorer** returns a message window like this one:

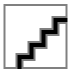


Figure 16. SQL code -430 message window

Causes

This might be a configuration problem.

Diagnosing the problem

1. Verify that the IBM Db2 Analytics Accelerator for z/OS stored procedures run in a separate Workload Manager (WLM) environment. Each application environment must be set up according to the instructions in *Setting up a WLM application environment for IBM Db2 Analytics Accelerator for z/OS stored procedures* in the *IBM Db2 Analytics Accelerator for z/OS: Installation Guide*. In particular, the ADMIN_INFO_SYSPARM and DSNUTILU stored procedures must run in different WLM environments and NUMTCB must be set to the correct value.
2. If the problem persists, collect the following diagnostic information for IBM support:

- CEEDUMP in the WLM job
- Stored procedure trace files in the z/FS directory `/tmp`. The names of these files follow this pattern:

  ```
  aqt-trace-<YYYYMMDD-HHMM>-<procedure name>-*
  ```

  Example:

  ```
  aqt-trace-20171113-1028-ACCEL_TEST_CONNECTION-ghbHxQ
  ```

Transfer the trace files in binary mode.

Database access problems after updating IBM Db2 Analytics Accelerator Studio or IBM Optim Query Tuner

IBM Db2 Analytics Accelerator Studio or IBM Optim™ Query Tuner no longer run successfully after an upgrade to a new version. A message says that privileges to access the database are missing.

Symptoms

A text similar to the following is displayed as part of the error message:

```
STEFANA DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION EXECUTE PACKAGE
ON OBJECT DB    2OSC.DSN5OADM. SQLCODE=-551, SQLSTATE=42501, DRIVER=3.58.81
```

Causes

A bind step has not been executed or users lack an execution privilege.

Diagnosing the problem

A bind step must be executed before a new version of IBM Db2 Analytics Accelerator Studio or IBM Optim Query Tuner can be employed. However, after carrying out the bind step, only the user who ran the command has the privilege to access the database from IBM Db2 Analytics Accelerator Studio or IBM Optim Query Tuner. If other users want to use IBM Db2 Analytics Accelerator Studio or IBM Optim Query Tuner, you must grant them this privilege.

Resolving the problem

1. To create and bind the Db2 packages, follow the instructions on this website:

   https://www.ibm.com/support/knowledgecenter/SS62YD_3.1.1/com.ibm.datatools.qrytune.configothers.doc/topics/enabledb2zfromclient_ds.html

2. To grant the EXECUTE privilege to other users, proceed as follows:

   https://www.ibm.com/support/knowledgecenter/SS7L9Q_4.1.0/com.ibm.datatools.qrytune.workloadtune.doc/topics/verify.html

DRDA connection does not work

You can ping the accelerators, but you cannot establish a distributed relational database access (DRDA) connection between your database management system and the accelerator.

Symptoms

Db2 commands or IBM Db2 Analytics Accelerator for z/OS stored procedures cannot establish a TCP/IP connection with the accelerator. Running the SYSPROC.ACCEL_TEST_CONNECTION stored procedure reveals this issue.

Resolving the problem

Make sure that the distributed data facility (DDF) of Db2 for z/OS uses the same TCP/IP stack as the ping program.

Different results for floating-point aggregates

Queries that involve floating-point aggregates produce different results when processed by IBM Db2 Analytics Accelerator for z/OS rather than Db2 for z/OS.

Symptoms

The very same query is processed twice, once by IBM Db2 Analytics Accelerator for z/OS and once by Db2 for z/OS. The results are different.

Causes

The result differences are differences in precision or error tolerances, which are caused by different methods of representation for floating-point numbers.

Resolving the problem

The problem cannot be solved. The differences must be accepted.

Related concepts

Differences in floating-point error tolerances

For queries that involve floating-point aggregates, you can expect IBM Db2 Analytics Accelerator for z/OS to produce results that are different from those delivered by inhouse Db2 for z/OS processing.

Stored procedure calls fail with message AQT10206I while temporary files are being created

Some or all running instances of the IBM Db2 Analytics Accelerator for z/OS stored procedures fail during the creation of temporary files.

Symptoms

Message AQT10206I is displayed for each failed stored procedure call:

```
AQT10206I - The OPEN NEW operation on the "/tmp/aqt-stored-procedure-trace-XXX"
file, data set, or pipe failed. Diagnostic information: mkstemp() failed with
rc = -1 and errno = 133
```

Causes

IBM UNIX System Services provides a z/OS UNIX file system (zFS) for z/OS. This file system, which contains the directory for the storage of temporary files (/tmp), is full. No more temporary files can be created for the execution of stored procedures.

Resolving the problem

Delete no longer needed files from the zFS. For example, you can safely delete stored-procedure trace-files. Note that you need superuser privileges in IBM UNIX System Services to do that. Follow these steps to delete the stored-procedure trace-files:

1. Log in to the IBM UNIX System Services shell of the logical partition (LPAR) from where you started the IBM Db2 Analytics Accelerator for z/OS stored procedure that returned the error. To log in, you can open a secure shell (ssh) window and use the **su** command to log in with superuser privileges. You can also use the IBM UNIX System Services shell to log in directly with an ID that has the required superuser privileges.
2. Change to the /tmp directory:

   ```
   cd /tmp
   ```

3. Delete the stored-procedure trace files:

   ```
   rm aqt-stored-procedure-trace*
   ```

Query history is not displayed - error message returned

The query history is not displayed in the **Accelerator** view and a corresponding error message is returned.

Symptoms

You do not see valid values in the columns of the **Query Monitoring** section of the **Accelerator** view and a message similar to the following is displayed:

```
[2013-05-25 05:14:09,790] [ERROR] [HistoryCol] Query History collection
    impossible com.ibm.db2.jcc.am.SqlException: UNSUCCESSFUL EXECUTION CAUSED
    BY AN UNAVAILABLE RESOURCE. REASON 00C900D1, TYPE OF RESOURCE 00000907,
    AND RESOURCE NAME. SQLCODE=-904, SQLSTATE=57011, DRIVER=4.15.79
```

Causes

The value of the Db2 for z/OS parameter LOBVALA might be too low.

Resolving the problem

Increase the value of the LOBVALA parameter. The default value is 10240. Set it to 20480. Increase the value further if this does not solve the problem.

Db2 for z/OS returns SQLCODE -30040

A query cannot be executed and `SQLCODE -30040` is returned by Db2 for z/OS.

Symptoms

Db2 for z/OSreports `SQLCODE -30040`, after IBM Db2 Analytics Accelerator has reported a `DRDARessourceLimitError` (DRDA code point `0x1233`) and one of the following reason codes:

`Reason 0x00001304 Type 0x00001409`

The allowed maximum of concurrently running queries has been reached or exceeded. Your actual maximum depends on your hardware resources. The minimum is 100, the (theoretical) maximum is 240 queries running in parallel.

`Reason 0x00000301 Type 0x00001409`

The accelerator state forbids the execution of the query. For example, if the state is *Initializing* or *Maintenance*, queries cannot be processed.

Resolving the problem

1. Depending on the reason code, wait for earlier queries to finish or wait for the accelerator to change to a state that permits query execution.
2. Resubmit the failed query.

Appendix A. Impact of special register, ZPARM, or bind option settings on accelerator-only tables

SQL statements that reference accelerator-only tables can only be executed on an accelerator. However, certain combinations of special register and ZPARM settings might prohibit the execution of a statement on an accelerator. The same is true for the QUERYACCELERATION bind option for static SQL queries. The table in this section shows how these settings affect the statement execution.

Legend

The statement examples in Table 32 on page 167 use abbreviations that indicate the type of a table:

AOT
: Accelerator-only table

ACC
: Regular accelerator table

Db2
: Db2 table (not on accelerator)

Remember that the ZPARM setting `QUERY_ACCEL_OPTIONS=2` enables `INSERT ... SELECT FROM ...` statements. With accelerator-only tables, the effect is often different from the effect it has in connection with regular accelerator tables. In many cases, the setting leads to an error. When applied successfully, it mostly leads to an acceleration of the INSERT portion of an `INSERT ... SELECT FROM ...` statement.

Table 32. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2

Statement	Special register SET CURRENT QUERY ACCELERATION= ZPARM QUERY_ACCELERATION= (ZPARM) Bind option QUERYACCELERATION=	QUERY_ACCEL_OPTIONS=2	Statement execution
SELECT * FROM T1_AOT	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator
SELECT * FROM T1_AOT, T5_ACC	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator

Table 32. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	**Special register SET CURRENT QUERY ACCELERATION=** **ZPARM QUERY_ACCELERATION= (ZPARM)** **Bind option QUERYACCELERATION=**	**QUERY_ACCEL_OPTIONS=2**	**Statement execution**
`SELECT * FROM T1_AOT, T3_Db2`	`NONE` (1)	No	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	Error, reason code 12
`SELECT * FROM T5_ACC, T3_Db2`	`NONE` (1)	No	Db2
	`ALL` (5)	No	Error, reason code 12
	`ELIGIBLE` (4)	No	Db2
	`ENABLE` (2)	No	Accelerator
`INSERT INTO T1_AOT VALUES (...)`	`NONE` (1)	No	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	Accelerator
`INSERT INTO T1_AOT SELECT FROM T2_AOT`	`NONE` (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	INSERT portion accelerated
		Yes	INSERT portion accelerated
`INSERT INTO T1_AOT SELECT FROM T5_ACC`	`NONE` (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	INSERT portion accelerated
		Yes	INSERT portion accelerated

Table 32. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	**Special register SET CURRENT QUERY ACCELERATION=** **ZPARM QUERY_ACCELERATION= (ZPARM)** **Bind option QUERYACCELERATION=**	**QUERY_ACCEL_OPTIONS=2**	**Statement execution**
`INSERT INTO T1_AOT SELECT FROM T3_Db2`	`NONE` (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	Error, reason code 12
		Yes	Error, reason code 12
`INSERT INTO T1_AOT SELECT FROM T2_AOT, T3_Db2, T5_ACC`	`NONE`	No	Error, reason code 2
		Yes	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	Error, reason code 2
		Yes	Error, reason code 12
• `INSERT INTO T5_ACC SELECT FROM T1_AOT` • `INSERT INTO T3_Db2 SELECT FROM T1_AOT`	`NONE` (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	`ENABLE` (2) `ELIGIBLE` (4) `ALL` (5)	No	SELECT portion accelerated
		Yes	SELECT portion accelerated

Table 32. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	**Special register SET CURRENT QUERY ACCELERATION=** **ZPARM QUERY_ACCELERATION= (ZPARM)** **Bind option QUERYACCELERATION=**	**QUERY_ACCEL_OPTIONS=2**	**Statement execution**
• INSERT INTO T5_ACC SELECT FROM T5_ACC • INSERT INTO T3_Db2 SELECT FROM T5_ACC	NONE (1)	No	Db2
		Yes	Db2
	ALL (5)	No	Error, reason code 17
		Yes	SELECT portion accelerated
	ELIGIBLE (4)	No	Db2
		Yes	SELECT portion accelerated
	ENABLE (2)	No	Db2
		Yes	Entirely in Db2 or SELECT portion accelerated (based on estimated response-time reduction)
• INSERT INTO T5_ACC SELECT FROM T3_Db2 • INSERT INTO T3_Db2 SELECT FROM T3_Db2	ALL (5)	No	Error, reason code 17
		Yes	Error, reason code 12
	NONE (1) ENABLE (2) ELIGIBLE (4)	No	Db2
		Yes	Db2
• UPDATE T1_AOT (SELECT FROM T2_AOT) • UPDATE T1_AOT (SELECT FROM T5_ACC) • DELETE T1_AOT (SELECT FROM T2_AOT) • DELETE T1_AOT (SELECT FROM T5_ACC)	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator

Table 32. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	**Special register SET CURRENT QUERY ACCELERATION=** **ZPARM QUERY_ACCELERATION= (ZPARM)** **Bind option QUERYACCELERATION=**	**QUERY_ACCEL_OPTIONS=2**	**Statement execution**
• UPDATE T1_AOT (SELECT FROM T3_Db2) • DELETE T1_AOT (SELECT FROM T3_Db2)	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 12
• UPDATE T5_ACC (SELECT FROM T1_AOT) • UPDATE T3_Db2 (SELECT FROM T1_AOT) • DELETE T5_ACC (SELECT FROM T1_AOT) • DELETE T3_Db2 (SELECT FROM T1_AOT)	NONE (1)	No	Error, reason code 22
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 22

Appendix B. Using telnet to log on to the IBM Db2 Analytics Accelerator Console

A telnet logon to the IBM Db2 Analytics Accelerator Console is still possible for compatibility reasons, but not recommended. Consider using the more secure ssh method instead.

Procedure

1. Start a client or emulator session (using, for example, IBM Personal Communications) to communicate with the z/OS system on which your Db2 subsystem is located.
2. Log on to TSO/ISPF.
3. Enter the following command:

   ```
   tso telnet <hostname> 1600
   ```

   where

   `<hostname>`
   Is the IP address of the accelerator that is connected to the Db2 for z/OS data server.

   `1600`
   Is the number of the port configured for accessing the IBM Db2 Analytics Accelerator Console using a telnet connection between the Db2 for z/OS data server and the accelerator.

   For example:

   ```
   tso telnet 203.0.113.8 1600
   ```

4. When prompted, enter `acceladm` as the user ID, followed by the console password.

   The initial password is dwa-1234. You must change this password at the first logon.

   The console also offers an option to create additional console users with different passwords.
5. Press the Pause key, then Enter to display the following screen:

```
******************************************************************************
*         Welcome to the IBM Db2 Analytics Accelerator Console
******************************************************************************

You have the following options:
 (0) - (Menu) Manage Configuration Console
Users
 (1) - (Menu) Run Accelerator Functions
 (2) - (Menu) Manage Incremental Updates
 (3) - (Menu) Manage Encryption of Data in Motion
 (4) - (Menu) Manage Call Home

 -----------------------------------------------------------------------
 (x) - Exit the Configuration Console
```

What to do next

Type x and press Enter to exit the **console** and close the telnet session.

Appendix C. Using ssh to log on to the IBM Db2 Analytics Accelerator Console

A secure shell (ssh) connection ensures that the information exchanged between your IBM Z system and the accelerator console is encrypted.

Before you begin

- An ssh client must be installed on the mainframe computer (IBM Z). Starting with z/OS V2R2, the OpenSSH package, which contains an ssh client, is part of the operating system.

  If you still use z/OS V2R1, you can obtain an ssh client by installing the respective component from BM Ported Tools for z/OS. See:

  https://www-03.ibm.com/systems/z/os/zos/features/unix/port_tools.html

  Mind that IBM does not offer this package anymore. However, the functionality lives on in more specific packages provided by the Rocket software company (http://www.rocketsoftware.com/zos-open-source). These packages usually include the OpenSSH component that contains the ssh client, so that you can connect from the z/FS file system to the accelerator. Note that you only need the client component of OpenSSH for maintenance updates (called ssh).
- An ssh client must also be installed on the workstation used to access the IBM Db2 Analytics Accelerator for z/OS Console. You can use a tool like PuTTY, which comes free of charge.

Procedure

1. On your client workstation, open an ssh shell and connect to the mainframe computer by using the following command:

   ```
   ssh <mainframe-ip>
   ```

   where <mainframe-ip> is the IP address or host name of the IBM Z system that is connected to the accelerator.
2. Enter the logon user ID and password for the IBM Z system.
3. When connected, enter the following commands from the shell to connect to the IBM Db2 Analytics Accelerator Console:

   ```
   ssh -p 2222 configuration-console@<accelerator-ip>
   ```

   where

   -p 2222

   Specifies a non-default connection port.

   <accelerator-ip>

   Is the IP address of the accelerator.

   For IBM Db2 Analytics Accelerator on an IBM Integrated Analytics System, this is the virtual IP or wall IP address.

   For Db2 Analytics Accelerator on Z, this is the IP address of the network that you labeled DB2 in the **Appliance Installer**.
4. When prompted, enter acceladm as the user ID, followed by the console password.

   The initial password is dwa-1234. You must change this password at the first logon.

   The console also offers an option to create additional console users with different passwords.
5. Press the Pause key, then Enter to display the following screen:

```
*****************************************************************************
*         Welcome to the IBM Db2 Analytics Accelerator Console
*****************************************************************************

You have the following options:
 (0) - (Menu) Manage Configuration Console
Users
 (1) - (Menu) Run Accelerator Functions
 (2) - (Menu) Manage Incremental Updates
 (3) - (Menu) Manage Encryption of Data in Motion
 (4) - (Menu) Manage Call Home

 ----------------------------------------------------------------------
 (x) - Exit the Configuration Console
```

What to do next

Type x and press Enter to exit the **console** and close the **ssh** session.

Notices

This information was developed for products and services offered in the US. This material might 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 grant 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
USA

For license inquiries regarding double-byte character set (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 provide 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
USA

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (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 "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

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

Microsoft and Windows 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.

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.

GDPR considerations

For PID(s): 5697-DA7, 5697-DA5, 5697-DAB

Notice: This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Db2 Analytics Accelerator for z/OS that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union (“EU”) and applies from May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR

- EU GDPR Information Portal (https://www.eugdpr.org/)
- ibm.com®/GDPR website (http://ibm.com/GDPR)

The EU General Data Protection Regulation (GDPR) regulates data privacy in the European Union (EU). For details, see: https://www.eugdpr.org

The regulations laid out in the GDPR apply if the data controller (an organization that collects data from EU residents), or the processor (an organization that processes data on behalf of a data controller, such as a cloud service provider), or the data subject (an individual) is based in the EU (source: Wikipedia. See: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation). A violation of GDPR regulations might incur severe financial penalties (fines) and reputation damages. This documentation is intended for data controllers or processors who use Db2 for z/OS together with IBM Db2 Analytics Accelerator for z/OS with the aim of storing personal data of EU residents (individuals). As regards IBM Db2 Analytics Accelerator for z/OS, the IBM company is neither the data controller, nor the data processor. Nevertheless, the IBM company feels obliged to give such entities guidance on how to follow GDPR regulations.

Your responsibilities as a data controller or processor
If you act on behalf of a data controller or data processor, see the frequently asked questions (FAQ) on the official GDPR website to find out what your responsibilities are:

https://www.eugdpr.org/gdpr-faqs.html

Furthermore, refer to chapter 4 in the GDPR regulations:

https://www.eugdpr.org/article-summaries.html

Product Configuration for GDPR

The following sections provide considerations for configuring IBM Db2 Analytics Accelerator for z/OS to help your organization with GDPR readiness.

How to configure the offering so that it can be used in a GDPR environment

The following sections describe how personal data is stored and processed by IBM Db2 Analytics Accelerator for z/OS (called 'accelerator' in the text that follows).

If personal data is to be stored on an accelerator, the following configuration steps might help you meet your compliance objectives with regard to GDPR:

- Activate the encryption of data in motion (data that is transferred over a data network between Db2 for z/OS and the accelerator).
- Activate the encryption of data at rest (encryption of disks and other storage devices).
- Do not share passwords for the operation of the IBM Db2 Analytics Accelerator Console between multiple persons. Instead, create separate user IDs with different passwords for each console user.
- Give read access to personal data (the privilege to run queries) only to authorized Db2 for z/OS users. To this end, define appropriate Db2 authorizations (SELECT, and so on) in the Db2 for z/OS catalog.
- Restrict the right to unload data from and to create image copies in Db2 for z/OS. Give this right to authorized administrators only.
- Restrict access to the 'Save Trace' function in Db2 Analytics Accelerator for z/OS because the collected trace data might contain personal data. Access should be given to authorized administrators only.
- Restrict access to the table SYSIBM.USERNAMES in the catalog of the Db2 for z/OS communication database because this table contains authentication credentials. If these credentials are read or copied by unauthorized users, these users might obtain access to sensitive personal data.

Data Life Cycle

You can create copies of any Db2 for z/OS table on an accelerator (accelerator-shadow tables). As soon as you create a copy of a Db2 for z/OS table and load it with personal data, the GDPR regulations apply in the same way as they apply to personal data in the original Db2 for z/OS tables.

An accelerator also allows you to load and store external data outside of Db2 for z/OS, such as the data of VSAM files, in accelerator-shadow tables. Products like the Db2 Analytics Accelerator Loader can be used to this end. If the external data contains personal information, the GDPR regulations apply to the data that is trasferred to the accelerator.

What types of data?

The administrative environment of an accelerator (client and configuration software) does not store personal data, except for the user IDs and authentication tokens of its administrators. However, the GDPR regulations apply to all personal data that you store in database tables on the accelerator.

Where in the process?

An accelerator is loaded with data in the course of individual load operations or by the incremental update function (automated load of table updates). Adding a table to an accelerator does not yet copy any data. It merely defines the table's structure (metadata) on the accelerator. However, if you load the table or enable it for incremental updates, the data of the source table - which might contain personal data - is copied to the accelerator.

For what purpose?

An accelerator, like Db2 for z/OS, is a general database management system for SQL queries. Such queries are run with the aim of retrieving stored information and gaining further insights (analytical queries that uncover the dependencies or connections between sets of values). An accelerator can return the results of SQL queries to authorized and authenticated users of the system. The results usually consist of subsets or aggregations of numerous data records. All these records might include personal data.

Personal data used for online contact with IBM

IBM Db2 Analytics Accelerator for z/OS clients can submit online comments/feedback/requests to contact IBM about accelerator subjects in a variety of ways, primarily:

- Public comments area on pages in the IBM Integration community on IBM developerWorks®
- Public comments area on pages of the IBM Db2 Analytics Accelerator for z/OS product documentation in the IBM Knowledge Center
- Feedback forms in the IBM Integration community

Typically, only the client name and email address are used, to enable personal replies for the subject of the contact, and the use of personal data conforms to the *IBM Online Privacy Statement* (http://www.ibm.com/privacy/cc/).

Data Collection

IBM Db2 Analytics Accelerator for z/OS can be used to collect personal data. When assessing your use of IBM Db2 Analytics Accelerator for z/OS and the requirements of GDPR, you should consider the types of personal data which in your circumstances are stored on an accelerator.

The use of an accelerator does not impose new requirements for the collection of personal data. It does not introduce new users or new privileges, but makes use of existing users and their privileges as defined in Db2 for z/OS. The use of an accelerator does not introduce additional obligations for the safekeeping or adminstration of data if you already observe the GDPR regulations for the storage of data in Db2 for z/OS.

Make sure that the process of collecting data and storing it in Db2 for z/OS is GDPR-compliant. The use of an accelerator will then be equally compliant.

Data Storage

The use of IBM Db2 Analytics Accelerator for z/OS involves the storage of data in a persistent manner.

An accelerator is a general-purpose data processing and retrieval engine. It is not aware of the meaning of data or any meaning that can be inferred from it. It is thus the responsibility of the data controller or processor to classify the data as personal data - for example account data - and make sure that the collection, access and processing of this data is in accordance with the GDPR regulations.

Storage in backups:

An accelerator has no backup function. However, image copies are created in Db2 for z/OS when the accelerator's archiving function is used (high performance storage saver). If these image copies contain personal data, the GDPR rules apply. This means, for example, that if individuals request their data to be deleted, it must also be deleted from all the image copies that may exist.

Storage in archives:

The high performance storage saver deletes table data from Db2 for z/OS and stores it exclusively on an accelerator. If this data contains personal information, the GDPR regulations might require that you delete the personal information from a number of records. The current version of the accelerator software does not allow DELETE operations on archived partitions. So in case you must delete records, first restore the archived partition to Db2 for z/OS, then delete then delete the personal data from Db2 for z/OS. Finally, archive the partition again (which now does not include the personal data anymore).

Data Access

Access to data on an accelerator requires a Db2 for z/OS user ID and a password. The accelerator recognizes the privileges defined in Db2 for z/OS and does not allow a user to run functions that she or he is not entitled to use.

As long as the Db2 for z/OS authorizations, such as SELECT, INSERT, UPDATE, or DELETE privileges, are defined in accordance with the GDPR regulations, access to data on the accelerator will likewise be in accordance with these regulations.

So it is crucial that the definition of roles and access rights in Db2 for z/OS effectuates an adequate separation of duties and responsibilities.

One accelerator-specific function might provide access to personal data on the accelerator: This is the "Save Trace" function, which collects diagnostic information for troubleshooting. The resulting trace files might contain personal data because they sometimes include the code of SQL queries to which personal information has been passed in the form of query literals. Trace files can also include memory dumps that contain personal data.

It is therefore important that access privileges to data, such as SELECT privileges or superuser privileges like SYSADM and DBADM, are granted with great care. The permission to use the "Save Trace" function should be given to selected administrators only.

The privilege to create image copies in Db2 for z/OS must also be handled with care. It is required, for example, if a user wants to archive table partitions on an accelerator.

Data Processing

Privileges defined in the Db2 for z/OS catalog control data processing on an accelerator.

In addition to that, the encryption features (encryption of data in motion and encryption of data at rest) should be activated. This prevents an unauthorized person without proper privileges from gaining direct access to personal data by interferring with the network communication or by obtaining physical access to the storage devices (hard disks or solid state disks).

Encryption features and safeguarding precautions:

Encryption of data in motion:
This feature encrypts the network traffic between an IBM Z system and an accelerator.

Encryption of data at rest:
This feature encrypts the data on an accelerator's physical storage devices (hard disks and solid state disks). Depending on the model of the hardware, the feature might already be turned on by default.

Encryption key ownership:
Restrict access to the encryption keys because the keys enable a person to read or decrypt personal data on a network or storage device. Put the safekeeping and maintenance (rotation) of the encryption keys in the hands of a trusted administrator.

Data Deletion

When data is deleted from Db2 for z/OS, it might still exist in table copies on the accelerator. To delete the data on an accelerator, you have the following options:

The data in accelerator-shadow tables is deleted:

- If the data in the original Db2 for z/OS tables has been deleted, and if the affected accelerator-shadow tables have been reloaded (with the contents of the now empty Db2 tables).
- If the accelerator-shadow tables are removed from the accelerator.
- If the incremental update function is used and the function replicates empty table content to the accelerator because the original Db2 for z/OS table data has been deleted.

The data in an accelerator-only table (AOT) is deleted:

- If the AOT is removed from the accelerator.
- If an explicit DELETE statement is submitted on the Db2 for z/OS side, which references the AOT and contains a predicate that specifies the row to be deleted.

The data in partitions archived by the high-performance storage saver (HPSS) is deleted:

- If the table partitions are removed from the accelerator.
- If an archived partition is restored to Db2 for z/OS.
- If the data is deleted from Db2 for z/OS because this means that the archived partition is reloaded or that the partition is archived again after the deletion from Db2 for z/OS (empty table content will be reloaded or archived).

Data Monitoring

Db2 for z/OS provides powerful auditing and monitoring functions that allow you to track the processing of personal data. Because IBM Db2 Analytics Accelerator for z/OS is a logical component of Db2 for z/OS, and keeps only copies of Db2 for z/OS data, the monitoring functions for data processing in Db2 for z/OS are sufficient.

Responding to Data Subject Rights

To identify an individual's data that is stored on an accelerator, you can submit SQL queries from Db2 for z/OS. These queries ought to search for the name, the account number, or for other identifying information about the subject using SQL syntax. Since an accelerator works on copies of Db2 for z/OS data, the same queries can be run to retrieve the data either from Db2 for z/OS or from the accelerator.

Specific advice or examples of such queries can not be provided here because the structure and the content of the database tables, as defined by the data controller or data processor, is not known.

Does the offering facilitate being able to meet data subject rights?

Right to Access

Can the client provide individuals access to their data?

Db2 for z/OS and accelerators can process SQL queries in which an individual's ID can be specified as a search criterion. This way, all data stored about that individual can be retrieved. In Db2 for z/OS, a client's data access can be restricted, so that the client can only search and retrieve a subset of the available data records, that is, those records that belong to a specific group of individuals. To this end, a feature called label based access control (LBAC) can be employed. IBM Db2 Analytics Accelerator provides a similar functionality, which restricts access to a selected set of rows based on the CURRENT SQLID special register.

Can the client provide individuals information about what data the client has about the individual?

Yes. To identify an individual's data stored on an accelerator, the client can submit SQL queries from Db2 for z/OS. These queries ought to search for the name, the account number, or for other identifying information about the subject using SQL syntax.

Right to Modify

Can the client allow an individual to modify or correct their data?

Db2 for z/OS allows a client to update the data that is stored about an individual. Depending on the architecture of the application, it might even be possible to authorize the individuals, so that these can update their data by themselves. After an update in Db2 for z/OS, the modified data is copied to the accelerators attached to the database during the next reload operation, which can be started manually or automatically (by the incremental update function). During that process, incorrect data will be overwritten.

Can the client correct an individual's data for them?

Yes (see the previous question *Can the client allow an individual to modify or correct their data?*).

Right to Restrict Processing

Can the client stop processing an individual's data?

Yes. In Db2 for z/OS, a client can submit DELETE statements to remove data records of individuals. The removal will be reflected in the data that is stored on an attached accelerator when the next reload operation (manual or automatic) is taking place. During that process, records previously deleted from Db2 for z/OS will also be deleted from the accelerator. Furthermore, an individual's data can be excluded from SQL queries through the use of filters in query predicates. For example, a query that reads `SELECT CUSTOMER_DATA FROM CUSTOMER_TABLE where custid NOT IN (a,b,...c)` returns only data records that are not related to (do not belong to) the users in the specified group `(a,b,...c)`.

Right to Object

Same as "Right to Restrict Processing" on page 186.

Right to Be Forgotten

Can the client delete an individual's data?

Yes.

Right to Data Portability

Can the customer provide an individual with the information that they have about the individual in a user-friendly/machine readable format?

Yes.

Glossary

This glossary includes terms and definitions for IBM Db2 Analytics Accelerator for z/OS.

The following cross-references are used in this glossary:

- *See* refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- *See also* refers you to a related or contrasting term.

A

accelerator-archive table
A table on an accelerator that contains data archived by the High-Performance Storage Saver (HPSS). The data in an accelerator-archive table comes from a DB2 table. The original DB2 data is deleted during the archiving process, so that storage space is freed in Db2 for z/OS. The HPSS is typically used for historical DB2 data that is no longer actively queried and maintained. It is none the less possible to include the data in accelerator-archive tables in accelerated queries. To archive a DB2 table or partitions of this table with the HPSS, the table must exist as an accelerator-shadow table. However, the tables does not have to be loaded.

accelerator-shadow table
A table on an accelerator whose definition is derived from a Db2 for z/OS table and that can be used for accelerated queries after loading (copying data from the original Db2 table to the accelerator-shadow table).

access plan graph
A visual representation of a query that shows the database objects that are accessed by the query and the order in which this is done.

APF
See authorized program facility.

authorized program facility (APF)
In a z/OS environment, a facility that permits the identification of programs that are authorized to use restricted functions.

C

cluster

- Number of configured accelerators.
- Table rows that are grouped together because they have the same value in a particular column.

collocated join
A join method where columns of an accelerator-shadow table must be on the same processing node. Leads to shorter query response times.

coordinator node
The primary node of the accelerator. It distributes the workload among the other nodes.

D

data slice
A portion of an accelerator-shadow table that is processed by a worker node.

DDF

See distributed data facility (DDF).

distributed data facility (DDF)

A set of Db2 for z/OS components by which Db2 for z/OS communicates with another relational database management system.

Dynamic Statement Cache

A pool in which Db2 saves prepared SQL statements that can be shared among different threads, plans, and packages. By sharing these statements, applications can avoid unnecessary preparation processes and thus improve performance.

F

fact table

A database table, commonly located at the center of a star schema surrounded by dimension tables, that contains measures of a business process, and foreign keys from the dimension tables.

failback

An automatic operation that returns the processing of a query to Db2 for z/OS if an error occurs while the query is processed by an accelerator.

friendly arithmetic

When an exception, such as a buffer overflow, a division by zero, or an out-of-range value occurs, Db2 returns the value NULL combined with a +802 warning, provided that the faulty expression is in the outermost SELECT statement and affects only one row.

I

inhouse processing

A query is processed by Db2 for z/OS and not routed to an accelerator.

L

lock

A means of preventing uncommitted changes made by one application process from being perceived by another application process and for preventing one application process from updating data that is being accessed by another process. A lock ensures the integrity of data by preventing concurrent users from accessing inconsistent data.

P

port-forwarding

A networking mechanism that allows Secure Shell access to a host in a private network from the outside.

Q

query conversion

Db2 for z/OS and the query engine of IBM Db2 Analytics Accelerator handle certain SQL expressions differently. In some cases, IBM Db2 Analytics Accelerator does not support an SQL expression that is supported by Db2 for z/OS. Therefore, changes to the mode of processing are sometimes inevitable when a query is routed to an accelerator.

R

row
See table row.

rowset
A set of rows for which a cursor position is established.

S

Secure Shell (SSH)
A UNIX-based command interface and protocol for securely getting access to a remote computer.

S-FTP
See SSH File Transfer Protocol.

skew
An imbalanced distribution of table rows among the processing nodes of the accelerator.

Sockets Secure (SOCKS)
A client/server architecture that transports TCP/IP traffic through a secure gateway. A SOCKS server performs many of the same services that a proxy server does.

SOCKS
See Sockets Secure.

SSH File Transfer Protocol
A network protocol that provides the ability to transfer files securely over any reliable data stream.

SSH tunnel
A secure and encrypted path through a network.

SSH
See Secure Shell.

systems programmer
A programmer who plans, maintains, and controls the use of an operating system with the aim of improving the overall productivity of an installation.

T

table row (row)
The horizontal component of a table, consisting of a sequence of values, one for each column of the table.

table space
A logical unit of storage in a database. In Db2 for z/OS, a table space is a page set and can contain one or more tables.

temporary table
A table that holds temporary data. Temporary tables are useful for holding or sorting intermediate results from queries that contain many rows. The two kinds of temporary tables, which are created by different SQL statements, are the created temporary table and the declared temporary table.

throughput
A measure of the amount of information transmitted over a network in a given period of time. Throughput is generally measured in bits per second (bps), kilobits per second (Kbps), or megabits per second (Mbps).

V

virtual IP address
An IP address that is shared among multiple domain names or multiple servers. Virtual IP addressing enables one IP address to be used either when insufficient IP addresses are available or as a means to balance traffic to multiple servers.

W

WLM application environment
A z/OS Workload Manager attribute that is associated with one or more procedures. The WLM application environment determines the address space in which a given procedure runs.

Index

Product Number: 5697-DA7

SH12-7082-09