IBM DB2 Analytics Accelerator for z/OS
Version 4.1.0

User's Guide

SH12-7040-01

Note

Before using this information and the product it supports, read the information in “Notices” on page 159.

Second Edition, March 2014

This edition applies to Version 4.1.0 of IBM DB2 Analytics Accelerator for z/OS, program number 5697-DAB, and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces SH12-7040-00. Changes to this edition are marked with a vertical bar.

This product includes software developed by Macromedia, Inc. (http://www.macromedia.com).

© Copyright IBM Corporation 2009, 2014.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 2009, 2014

Figures

© Copyright IBM Corp. 2009, 2014

About this book

This book provides information about the following subjects:

- Connecting to a data server
- Adding query accelerators
- Adding and maintaining tables for query acceleration
- Configuring and activating system traces

© Copyright IBM Corp. 2009, 2014

Who should read this book

This book is intended for database administrators who need to run IBM® DB2® Analytics Accelerator Studio to complete administration tasks for IBM DB2 Analytics Accelerator for z/OS®.

The most important tasks in this context are the definition and deployment of data for accelerated database queries and the adding, removal, activation, and deactivation of individual accelerators within the system configuration.

© Copyright IBM Corp. 2009, 2014

What's new in version 4?

This version of IBM DB2 Analytics Accelerator for z/OS, when used with the latest DB2 for z/OS support, includes the following new features:

- Support for acceleration of static SQL queries in DB2 for z/OS applications.
- DB2 for z/OS multi-row fetching of results from accelerated queries (a configurable number of rows can be retrieved in a single time unit).
- Workload balancing of accelerated queries. If a query can be executed on more than one accelerator, DB2 for z/OS automatically distributes the workload among qualifying accelerators. Accelerators with a low utilization are favored.
- High-performance storage saver (HPSS) improvements. A full-scale restore function has been added to the HPSS scope. This function takes advantage of the newest DB2 for z/OS Utilities support. In addition, an enhanced DB2 for z/OS Utility function now sets the affected table space to a persistent read-only state when the HPSS is invoked to move partition data. Furthermore, the same partition data can now be moved to multiple accelerators.
- Table-load performance improvements, resulting in a two-digit percentage of CPU-cost reduction.
- Incremental updates configurable for more than two DB2 subsystems.
- Automated transfer and installation (apply action) of updates for the Netezza Performance Server® (NPS®).
- Incremental updates continue while tables are being reloaded.
- Workload Manager (WLM) support for local DB2 for z/OS applications. That is, WLM priorities can be assigned to local applications to support your processing preferences.
- Support for DB2 11 for z/OS.

Enhancements in version 4 PTF-2:

PTF-2 of the product introduces the following enhancements:

- Migration stored procedure that adjusts table names in accordance with new naming scheme (required for workload balancing and the acceleration of static SQL queries)
- Support for TIME and TIMESTAMP columns containing 24:00:00 values (conversion no longer required)
- Support for FOR BIT DATA values in single-byte character EBCDIC tables
- Elimination of cast problems related to character data in EBCDIC and Unicode tables located on the same accelerator
- Support for lock-free table loading of incrementally updated tables, which allows INSERTs, UPDATEs, and DELETEs to continue
- Template support for temporary data sets created by the High Performance Storage Server (HPSS)

© Copyright IBM Corp. 2009, 2014

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information.

If you have any comments about this book or any other IBM DB2 Analytics Accelerator for z/OS documentation:

- Go to the IBM DB2 Analytics Accelerator for z/OS support home page at:

  http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Analytics_Accelerator_for_z~OS

  The link **Support feedback** > **Help us improve online support**, which is usually on the lower right, opens the feedback form, in which you can enter your comments.
- Send your comments by email to swsdid@de.ibm.com. Be sure to include the name of the book, the publication number of the book, the version of IBM DB2 Analytics Accelerator for z/OS, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
- Complete one of the forms at the back of this book and send it by fax, or give it to an IBM representative.

© Copyright IBM Corp. 2009, 2014

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 *Viewing information in the help system*, which becomes available after downloading the information center from http://www.ibm.com/support/docview.wss?uid=swg24033669, extracting its content, and starting it.

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 in the Eclipse documentation (http://help.eclipse.org/juno/index.jsp).

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 website:

http://www.eclipse.org/swt/faq.php#tableheaderswithJAWS

IBM and accessibility

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

© Copyright IBM Corp. 2009, 2014

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" on page 59
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.

© Copyright IBM Corp. 2009, 2014

Related tasks:

"Adding accelerators" on page 65
To complete the authentication, you enter the IP address, 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 and choosing favorable distribution keys and organizing keys. However, you also need to complete a number of related sub-tasks. This section lists all required tasks in the correct order.

Selecting the proper tables

In choosing the proper tables, you select the data to be queried. This data will be copied to the accelerator when you load the selected tables.

Related tasks:

"Defining tables on an accelerator" on page 68
Using the Add Tables to Accelerator wizard, you can define database 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.

Choosing a distribution key

By default, all tables rows are evenly distributed among the existing worker nodes (random distribution). When a query includes table joins, it can be that a join can only be executed if table rows are sent from one worker node to another (redistribution), or that all rows are first sent to the Netezza® host to be distributed to the worker nodes from there (broadcast). Both, redistribution and broadcast, can be very time-consuming when big tables with millions or even billions of records are involved, to the extent that the accelerator becomes ineffective.

Using distribution keys, you can instruct IBM DB2 Analytics Accelerator for z/OS to distribute the rows of tables to be joined to the same worker node so that redistributions and broadcasts of large amounts of data can be avoided.

Related concepts:

"Distribution keys" on page 8
You can choose between two methods to distribute table rows to the worker nodes of your accelerator: random distribution and distribution using a distribution key.

"How to select tables for query acceleration" on page 12
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" on page 80
Distribution and organizing keys have a considerable impact on the query response time.

Choosing an organizing key

Organizing keys can speed up accelerated queries even further as they diminish the time that is needed to scan the disks belonging to a single worker node.

When you create an organizing key, you select certain columns to the effect that table rows with equal and nearby values in the selected columns are grouped together. These clusters are then written to the extents, that is, addressable storage blocks on disk.

IBM DB2 Analytics Accelerator for z/OS creates zone maps for each column that is selected as an organizing key. Zone maps contain information about the value range (minimum and maximum) that can be found in an extent.

When an incoming query restricts on a column that was selected as an organizing key, IBM DB2 Analytics Accelerator for z/OS first uses the information in the zone maps to identify the extents that contain table rows with relevant data. When an extent must be scanned, the query engine benefits from the fact that the table rows are organized in clusters. If a table row is found with a column value that is not relevant to the query, this and all remaining rows in the cluster (rows with the same value in that column) can be skipped. Only blocks with relevant table rows must be read line-by-line, which can reduce the time needed for disk and table scans dramatically.

Related concepts:

"Organizing keys" on page 9
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 a 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 by skipping entire blocks rather than having to scan the entire table on disk. Thus the time needed for disk output operations related to the query is drastically reduced.

"Zone maps" on page 15
Zone maps are automatically generated, internal tables that summarize the range of column values (minimum and maximum) of rows that fall into the same extent, that is, an addressable storage block on disk.

"How to select tables for query acceleration" on page 12
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" on page 80
Distribution and organizing keys have a considerable impact on the query response time.

Working with accelerator tables

To populate, maintain, and control the tables on an accelerator, 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 tables.

You can also reload tables, and, under certain conditions, selected partitions of tables on an accelerator. 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.

Furthermore, you can enable incremental updates. Once this ongoing process is enabled, changes in the original DB2 data tables are detected, captured, and propagated automatically to the corresponding accelerator tables.

Related concepts:

"Updating accelerator tables continuously" on page 106
The incremental update function of IBM DB2 Analytics Accelerator for z/OS allows you to update accelerator tables continuously. Changes to the data in original DB2 for z/OS tables are thus propagated to the corresponding accelerator 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" on page 74
Successful queries against tables on an accelerator 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 table on an accelerator. 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 tables on an accelerator 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 explicitly mark the tables that you want to include in this process.

Disabling incremental updates

Disabling incremental updates excludes tables from the incremental update process.

Moving partition or table data to an accelerator

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 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 tables from the accelerator.

Related tasks:

"Removing tables from an accelerator" on page 96
Follow the steps in this topic to remove tables from an accelerator.

Order in which to complete tasks

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

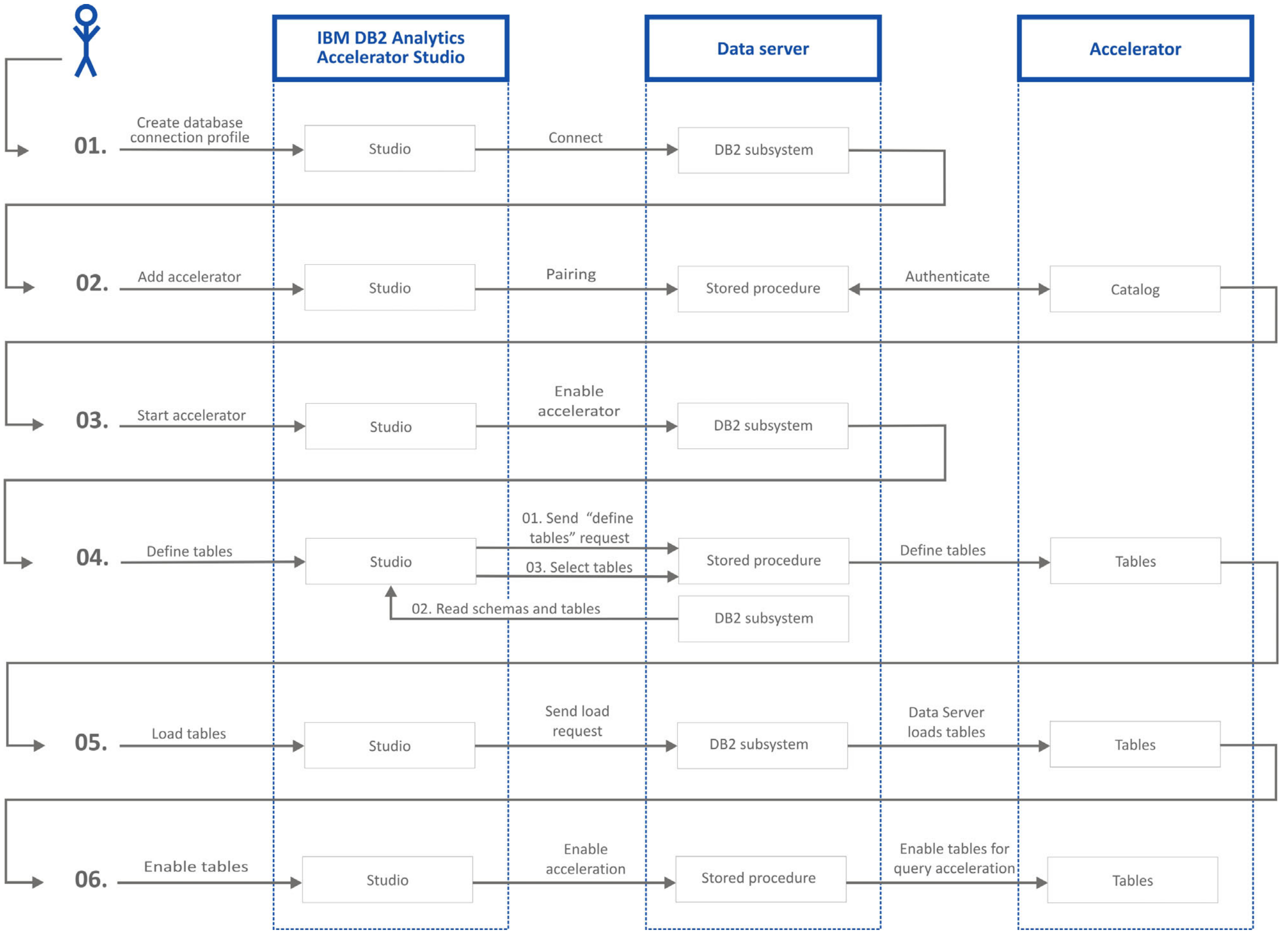


Figure 1. IBM DB2 Analytics Accelerator Studio tasks in the overall process flow

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 table. These portions are called *data slices*. The tables are thus horizontally partitioned because each worker node processes a data slice of each table.

© Copyright IBM Corp. 2009, 2014

Virtual accelerator

For testing purposes, you can employ virtual accelerators. These are simulators that use the EXPLAIN function offered by DB2 for z/OS. The output will always be DB2 EXPLAIN information because virtual accelerators cannot process regular queries and hence cannot return the corresponding results.

The EXPLAIN information can be visualized in an access plan graph (Visual Explain Graph). While the same information can also be obtained from real accelerators, virtual accelerators have the advantage that they do not require accelerator hardware. You can thus check whether queries can be accelerated, see if queries contain errors, and calculate response time estimates without making extra demands on the hardware resources.

Related concepts:

"EXPLAIN information" on page 84
After creating the necessary EXPLAIN tables, you can analyze queries by invoking the DB2 EXPLAIN function. Such queries are accepted by regular and virtual accelerators alike. 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.

Related tasks:

"Displaying an access plan graph" on page 87
An access plan graph is 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.

Distribution keys

You can choose between two methods to distribute table rows to the worker nodes of your accelerator: random distribution and distribution using a distribution key.

Random distribution
A circular or revolving distribution of table rows. Each data slice, which is the portion of data that is processed by a single worker node, is assigned an equal number of table rows starting with the first slice, then moving on to the next, and so on. After table rows have been assigned to the last unit in the chain, the next chunk of rows is distributed to the first again. This process continues until all table rows have been distributed.

Use of a distribution key (hash partitioning)
You can specify up to four columns that make up a distribution key. The accelerator uses a hash function on the key columns to determine the data slice or worker node that receives a table row. Rows with equal hash values are distributed to the same worker node.

By default, random distribution is used to distribute table rows to worker nodes for query processing. That is, all tables are evenly distributed among the existing worker nodes.

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 tables are scattered across the worker nodes even though they are evenly distributed in terms of quantity, you can easily spent 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 from both table reside on the same worker node (collocated join). You can instigate collocated joins by selecting the join columns as the distribution key for both tables.

However, if a 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, it is best to collocate 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 collocate 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.

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 a 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 by skipping entire blocks rather than having to scan the entire table on disk. Thus the time needed for disk output operations related to the query is drastically reduced.

Organizing keys work together with zone maps, which allow the query engine to identify the disk extents (addressable storage blocks on 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). The recommended minimum table size (compressed size on the IBM PureData™ System for Analytics N1001) depends on the number of worker nodes:

  Table 1. Recommended minimum table sizes for organizing keys

IBM PureData System for Analytics	Minimum size
N1001-002	0.4 GB
N1001-005	0.75 GB
N1001-010	1.5 GB

- It is recommended to define an organizing key for tables that you want to update continuously 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 parent attribute in a joined dimension, requests 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 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 extents.
- 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 clustering the table rows causes a processing overhead when you load or update the tables on the accelerator.

Static SQL query support

Static SQL query support means that you can now accelerate SQL queries that are embedded within DB2 for z/OS application programs. Older versions of IBM DB2 Analytics Accelerator for z/OS could not accelerate such queries.

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.

Related tasks:

"Preparing tables for workload balancing and accelerating static SQL queries" on page 72
You need not enable workload balancing or static SQL support. However, the development of these features made it necessary to introduce a new naming scheme for tables. So if you have tables that were defined with older versions of IBM DB2 Analytics Accelerator for z/OS and want to use workload balancing or accelerate static SQL queries, you must remove and then redefine those tables.

Table

A table on an accelerator represents a table in your database. By adding a table 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 tables on the accelerator 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 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 added (copied) to the accelerator. All other columns will copied. So the job of adding 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" on page 68
Using the Add Tables to Accelerator wizard, you can define database 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 collocated joins, which lead to shorter query response times, table columns 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 table scans, 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 14.

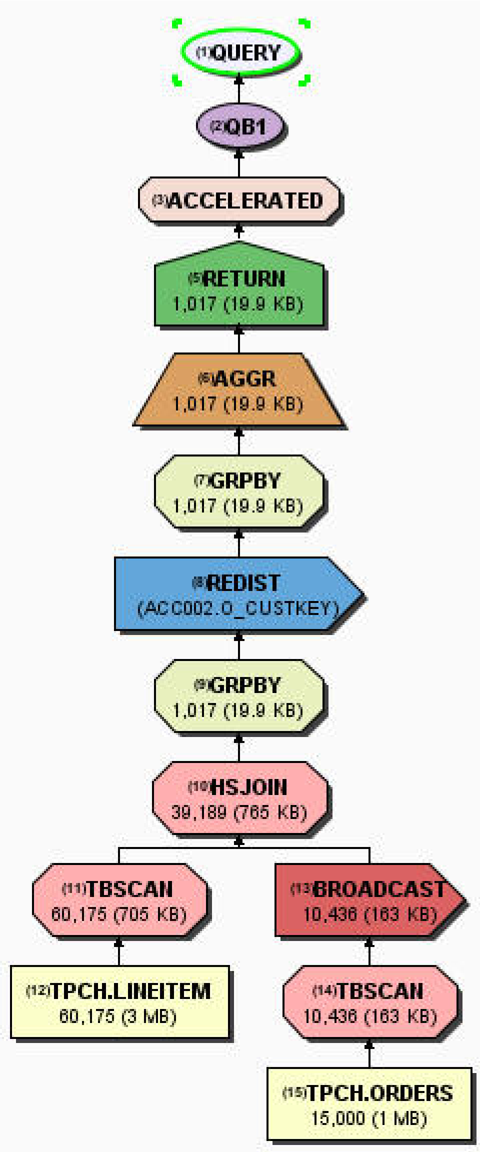


Figure 2. Access plan graph of the query in the example

Related concepts:

"Choosing a distribution key" on page 2
By default, all tables rows are evenly distributed among the existing worker nodes (random distribution). When a query includes table joins, it can be that a join can only be executed if table rows are sent from one worker node to another (redistribution), or that all rows are first sent to the Netezza host to be distributed to the worker nodes from there (broadcast). Both, redistribution and broadcast, can be very time-consuming when big tables with millions or even billions of records are involved, to the extent that the accelerator becomes ineffective.

"Choosing an organizing key" on page 3
Organizing keys can speed up accelerated queries even further as they diminish the time that is needed to scan the disks belonging to a single worker node.

Related tasks:

"Displaying an access plan graph" on page 87
An access plan graph is 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.

Related reference:

“Types of access plan graphs and nodes in these graphs” on page 89
Different graphs are displayed for queries that can be accelerated and queries that cannot be accelerated. Read a brief article on how these graphs differ, and which set of nodes you can expect in either of these.

Related information:

Graphing the access plan for a query

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 multiple accelerators, IBM DB2 Analytics Accelerator for z/OS automatically picks the accelerator with the lowest utilization for processing.

Related tasks:

“Preparing tables for workload balancing and accelerating static SQL queries” on page 72
You need not enable workload balancing or static SQL support. However, the development of these features made it necessary to introduce a new naming scheme for tables. So if you have tables that were defined with older versions of IBM DB2 Analytics Accelerator for z/OS and want to use workload balancing or accelerate static SQL queries, you must remove and then redefine those tables.

Zone maps

Zone maps are automatically generated, internal tables that summarize the range of column values (minimum and maximum) of rows that fall into the same extent, that is, an addressable storage block on disk.

Zone maps are always created for columns that were selected as organizing keys. Furthermore, you can only select a column as an organizing key if the data type of this column allows IBM DB2 Analytics Accelerator for z/OS to create a zone map.

Using the information in the zone maps, IBM DB2 Analytics Accelerator for z/OS “knows” which extents it needs to “look at” during a disk scan. Extents with relevant data (table rows containing the values demanded by a query) must be scanned; those that contain just irrelevant data can be skipped.

Zone maps thus tell the query engine whether it is worth to look at a certain extent on disk. If it is, then the clusters within an extent allow the query engine to scan the table rows in the extent much faster because rows with relevant data will be found in the same block, while blocks that do not contain relevant data can be skipped entirely.

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.

Important: Unlike the dynamic statements that are executed in DB2, the queries that are routed to an accelerator cannot be cached in the Dynamic Statement Cache. Problems arise if queries were not accelerated in the past, but would become accelerated queries due to certain changes. In these cases, the obsolete information in the Dynamic Statement Cache would prevent acceleration, although the queries fulfill the conditions (once in the cache, the routing decision will never be reevaluated).

For example, you must explicitly invalidate the cache entries after you have changed Profile Table attributes that are relevant for the accelerator because you want DB2 to re-prepare the statement and consider its applicability to accelerator routing instead of using the cached version in the Dynamic Statement Cache. The least obtrusive way to invalidate the statements in the Dynamic Statement Cache is to execute the RUNSTATS utility with the options REPORT NO UPDATE NONE for the corresponding table spaces.

For information about the attributes that affect the DB2 routing decision, follow the appropriate link under **Related information**. Note that these are serviceability parameters. Do not modify the values without guidance from IBM.

Related information:

DB2 Version 9.1 for z/OS: SYSIBM.DSN_PROFILE_ATTRIBUTES

DB2 10 for z/OS: SYSIBM.DSN_PROFILE_ATTRIBUTES

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.

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:

© Copyright IBM Corp. 2009, 2014

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.

Exceptions: The following dynamic queries are always processed by DB2 for z/OS, regardless of any relevant ZPARM settings (QUERY_ACCLERATION or GET_ACCEL_ARCHIVE):

- If all the tables referenced in the query have the qualifier:
  - SYSIBM

    Exception: Test queries against SYSIBM.SYSDUMMY1, SYSIBM.SYSDUMMYU and SYSDUMMYA are supported.
  - SYSACCEL
  - DB2GSE
  - SYSXSR
  - DGTT
- Dynamic queries whose top query block is pruned, and which therefore return an empty result set. To check whether a query falls into this category, follow these steps:
  1. Explain the query using the DB2 EXPLAIN function. Use the following special register setting in the SQL statement:
     ```
     SET CURRENT QUERY_ACCELERATION = NONE
     ```
  2. Check the PLAN_TABLE. If the top query block has been pruned, the entry for the query block shows `PRUNED` in the QBLOCK_TYPE column.

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 options for DB2 packages.

Important:

- The new acceleration bind options only apply to cursor queries and the SELECT portion of the SQL INSERT from SELECT statement. The SQL SELECT INTO statement cannot be bound for acceleration.
- The functions and behavior of these acceleration bind options for static queries are the same as the comparable existing acceleration special registers CURRENT QUERY ACCELERATION and CURRENT GET ACCEL ARCHIVE that are 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 options are 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 2. Bind options for static SQL queries

QUERYACCELERATION	**GETACCELARCHIVE**
Binds static SQL queries for execution on an accelerator	Includes or excludes partition data that was moved 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 (ENABLE WITH FAILBACK)** 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 moved 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 moved by the HPSS, one of the following course of actions 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 moved 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 2. 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 new bind options are 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 new bind options are not supported for:
- ALTER TRIGGER
- CREATE TRIGGER
- REBIND TRIGGER PACKAGE

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.

The bind options can also be used for the acceleration of dynamic queries. If the bind options are specified, they set the initial values in the same manner as the comparable special registers CURRENT QUERY_ACCELERATION and CURRENT GET_ACCEL_ARCHIVE when the application is run provided that the special registers have not already been set explicitly before the invocation of that application. The values of the bind options are then used as special register values for dynamic queries in that application, so that you do not have to add explicit SET statements to the application for the acceleration special registers.

The special registers do not apply for static queries, but if you do not want your static queries in that same application to be bound for acceleration and their tables are accelerated, then you must add the explicit SET statements to the application instead of using the new bind options to initialize the special registers for dynamic queries. Precedence order (low to high) for setting the initial value of the special registers is as follows:

1. DB2 installation system parameters 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 registers, the use of the new bind options requires you to set ACCEL (DB2 ZPARM) to a value other than NONE to enable the DB2 subsystem for query routing to IBM DB2 Analytics Accelerator for z/OS.

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. In connection with DB2 10 for z/OS, the principal expression can also be 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 = STTSTR = '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.

Related information:

DECLARE CURSOR

Supported SQL functions

IBM DB2 Analytics Accelerator for z/OS supports all aggregate functions, except for the XMLAGG function. In addition, IBM DB2 Analytics Accelerator for z/OS supports a variety of scalar functions.

Table 3 lists the supported scalar functions.

Table 3. Supported scalar functions

Function name	Function name	Function name	Function name
• ABS	• FLOAT	• MIDNIGHT_SECONDS	• SIGN
• ADD_MONTHS	• FLOOR	• MIN	• SMALLINT
• BIGINT	• HOUR	• MINUTE	• SPACE
• CEILING	• IFNULL	• MOD	• SQRT
• CHAR	• INTEGER	• MONTH	• STRIP
• COALESCE	• JULIAN_DAY	• MONTHS_BETWEEN	• SUBSTR
• CONCAT	• LAST_DAY	• NEXT_DAY	• TIME
• DATE	• LCASE	• NULLIF	• TIMESTAMP
• DAY	• LEFT	• POSSTR	• TIMESTAMP_FORMAT
• DAYOFMONTH	• LENGTH	• POWER	• TRANSLATE
• DAYOFWEEK	• LN	• QUARTER	• TRUNCATE
• DAYOFWEEK_ISO	• LOCATE	• RADIANS	• UCASE
• DAYOFYEAR	• LOCATE_IN_STRING	• REAL	• UPPER
• DAYS	• LOG10	• REPEAT	• VALUE
• DECIMAL	• LOG	• REPLACE	• VARCHAR
• DEGREE	• LOWER	• RIGHT	• VARCHAR_FORMAT
• DIGITS	• LPAD	• ROUND	• WEEK_ISO

Table 3. Supported scalar functions (continued)

Function name	Function name	Function name	Function name
• DOUBLE	• LTRIM	• RPAD	• YEAR
• EXP	• MAX	• RTRIM	
• EXTRACT	• MICROSECOND	• SECOND	

Supported special registers

IBM DB2 Analytics Accelerator for z/OS supports only a few special registers. During query routing to an accelerator, these special registers are replaced with the equivalent Netezza functions. In some cases, this leads to different query results.

The following special registers are supported:

CURRENT DATE
Replaced with the Netezza function CURRENT_DATE on the accelerator

CURRENT TIME
Replaced with Netezza function 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 following Netezza TIMESTAMP function:

```
TIMESTAMP(to_char(current_timestamp, 'yyyy-mmdd-hh.mm.ss.ts'))
```

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

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.

- 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 correlated table expression. A correlated table expression is a table expression that contains one or more correlated references to other tables in the same FROM clause. For example:

  ```
  SELECT ... FROM TA A, TABLE ( SELECT ... FROM TB B WHERE B.C1 = A.C1) TX (..),
  TC WHERE ...
  ```
- The query contains a recursive common table expression reference.
- Your version of the Netezza Performance Server (NPS) is lower than 7.0.4 and the query contains one of the following predicates:
  - `= ALL (<subquery>)` (equal to)
  - `<> ALL (<subquery>)` (not equal to)
  - `> ALL (<subquery>)` (greater than)
  - `< ALL (<subquery>)` (less than)
  - `>= ALL (<subquery>)` (greater than or equal to)

- <= ALL (<subquery>) (less than or equal to)
- NOT IN (<subquery>)

Important: Sub-queries of the type NOT IN must be enabled. Contact IBM support for more information.

- The query contains a string expression (including columns) with a subtype that is not in the following list. That is, only the following combinations of coded character set identifiers (CCSID) and subtypes are allowed in string expressions:

Table 4. Supported combinations of CCSIDs and subtypes

CCSID	Allowed subtypes
ASCII	SBCS
EBCDIC	SBCS MIXED DBCS (graphic)
UNICODE	SBCS MIXED DBCS (graphic)

Important:

- The combinations EBCDIC + MIXED and EBCDIC + DBCS are supported only if the system parameter QUERY_ACCEL_OPTIONS includes the value 1 and if the value in the SUPPORTLEVEL column of SYSACCEL.SYSACCELERATEDTABLES is 2 or higher for the affected tables. For more information about these options, 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 an expression with an unsupported result data type. Supported result data types:
  - BIGINT
  - CHAR
  - CHAR FOR BIT DATA (beginning with version 4.1 PTF-2, EBCDIC only)

    EBCDIC tables that were added and loaded with product version 4.1 PTF-2 or later now support columns with a data type of CHAR FOR BIT DATA. This means that queries can now reference CHAR FOR BIT DATA columns in EBCDIC tables that were added with version 4.1 PTF-2 or a later version.

    Queries referencing CHAR FOR BIT DATA columns in EBCDIC tables that were loaded with product version 4.1 PTF-1 or earlier will not be accelerated because the FOR BIT DATA columns have been excluded during the load. That is, these columns do not exist on the accelerator. Queries not referencing FOR BIT DATA columns continue to work. To enable FOR BIT DATA support for tables loaded with earlier versions, remove the relevant tables from the accelerator, then define and load these again.
  - DATE
  - DECIMAL
  - DOUBLE
  - FLOAT
  - GRAPHIC (Normally, this type is supported for UNICODE only. To enable this type for EBCDIC, include the value 1 when setting QUERY_ACCEL_OPTIONS.)
  - INT
  - REAL
  - SMALLINT

  - TIME
  - TIMESTAMP
  - VARCHAR
  - VARCHAR FOR BIT DATA (beginning with version 4.1 PTF-2, EBCDIC only)

    EBCDIC tables that were added and loaded with product version 4.1 PTF-2 or later now support columns with a data type of VARCHAR FOR BIT DATA. This means that queries can now reference VARCHAR FOR BIT DATA columns in EBCDIC tables that were added with version 4.1 PTF-2 or a later version.

    Queries referencing VARCHAR FOR BIT DATA columns in EBCDIC tables that were loaded with product version 4.1 PTF-1 or earlier will not be accelerated because the FOR BIT DATA columns have been excluded during the load. That is, these columns do not exist on the accelerator. Queries not referencing FOR BIT DATA columns continue to work. To enable FOR BIT DATA support for tables loaded with earlier versions, remove the relevant tables from the accelerator, then define and load these again.
  - VARGRAPHIC (The note under GRAPHIC applies here as well)

  Note: The listed types refer to the built-in types. User-defined types (UDTs) are not allowed.

- 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` by including the `QUERY_ACCEL_OPTIONS` ZPARM value 4. 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
- A date, time, or timestamp duration is specified in the query. Only labeled durations are supported.
- 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 contains a correlated scalar fullselect (correlated subquery) that the accelerator cannot handle. In such a case, the DB2 EXPLAIN table DSN_QUERYINFOTABLE shows the value 21 in the `REASON_CODE` column. The term *correlated* means that the scalar fullselect references a column of a table or view that is named in an outer subselect. For example, a query will be blocked if any of the following conditions applies:

  - All of the following is true:
    1. The subquery correlates directly with its parent.
    2. The subquery contains an IFNULL, COALESCE, VALUE, or CASE expression.
    3. The parameter of any of these expressions is an aggregate function.

       Example:

       ```
       SELECT * FROM T2 AS A
       WHERE C0=(SELECT COALESCE(MAX(C0),1)
       FROM T2 AS B
       WHERE A.C0=B.C0 AND 1=2);
       ```
  - The subquery is in a NOT-IN list.
  - The predicate (WHERE clause) of the subquery specifies an IS NULL condition when the column is defined as NOT NULL.

    Example: C3SINT is defined as NOT NULL:

    ```
    SELECT * FROM T1
    WHERE C1 = C2
    AND C1 in (SELECT C1 FROM T2 WHERE T2.C3SINT IS NULL)
    ```
  - The previous restrictions for correlated subqueries no longer exist in DB2 11 for z/OS and later versions. However, the following restriction still holds true for all supported DB2 for z/OS versions (including DB2 11 for z/OS):

    The query contains a correlated scalar fullselect query in an IN list.

    Example:

    ```
    SELECT * FROM TEST2 T1 WHERE T1.C0 IN
    (2,(SELECT C0 FROM TEST2 T2 WHERE T1.C1=T2.C1),3);
    ```

  For more information, follow the appropriate link under **Related information** at the end of this topic.
- All three of the following statements are true for character-based scalar functions or cast specification expressions in a query:
  - The `CODEUNIT16`, `OCTETS`, or `NO CODEUNITS` option is used.
  - The QUERY_ACCEL_OPTION parameter does not include the value 3.
  - The string arguments are not single-byte (SBCS) values.
- The query contains a cast specification expression with a result data type of GRAPHIC or VARGRAPHIC.
- The query contains one of the following scalar functions or cast specification expressions with a string argument that is encoded in UTF-8 or UTF-16. The problem occurs because Netezza evaluates the character values of these string functions, whereas DB2 evaluates the byte values.
  - CAST (arg as CHAR(*n*))
  - CAST(arg AS VARCHAR(*n*)) where *n* is less than the length of the argument
  - CHAR
  - LEFT
  - LOCATE
  - LOCATE_IN_STRING
  - LOWER(arg, *n*) where *n* is not equal to the length of the argument
  - LPAD
  - POSSTR
  - REPLACE
  - RIGHT
  - RPAD
  - SUBSTR

  - TRANSLATE (if more than one argument is specified)
  - UPPER(arg, *n*) where *n* is not equal to the length of the argument
  - VARCHAR(arg, *n*) where *n* is less than the length of the argument

  Note: To enable these types for query routing to an accelerator, include the value 3 in the setting of QUERY_ACCEL_OPTIONS. Result deviations might occur if the input string contains multibyte characters.

- The query uses a LENGTH function, but the argument of this function is not a string or is encoded in UTF-8 or UTF-16. To enable query acceleration for queries containing the LENGTH function with an argument of these types, include option 3 in the value of QUERY_ACCEL_OPTIONS.
- The query uses a DAY function where the argument of the function specifies a duration.
- DB2 10 for z/OS: The query uses the LAST_DAY function and specifies a TIMESTAMP expression as its first argument.
- The query uses the MIN or MAX function with more than four arguments or the result of the MIN or MAX expression is not of the INTEGER or FLOAT data type.
- 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 the EXTRACT function, and the function specifies that the SECOND portion of a TIME or TIMESTAMP value must be returned.
- The query uses one of the following aggregate functions with the DISTINCT option:
  - STDDEV
  - STDDEV_SAMP
  - VARIANCE
  - VAR_SAMP
- The query uses a table function, such as ADMIN_TASK_LIST or ADMIN_TASK_STATUS.
- The query uses an unsupported scalar function. See the following table:

Table 5. Unsupported scalar functions

• BINARY	• GETHINT	• RAND
• BLOB	• GETVARIABLE	• ROWID
• COLLATION_KEY	• GRAPHIC	• SCORE
• COMPARE_DECFLOAT	• HEX	• SOA <xxx>
• DBCLOB	• IDENTITY_VAL_LOCVAL	• SOUNDEX
• DECFLOAT	• MQ <xxx>	• SUBSTRING
• CHARACTER_LENGTH	• NORMALIZE_DECFLOAT	• TOTALORDER
• DECFLOAT_SORTKEY	• NORMALIZE_STRING	• VARBINARY
• DIFFERENCE	• OVERLAY	• VARGRAPHIC
• DECRYPT <xxx>	• POSITION	• WEEK
• ENCRYPT_TDES	• QUANTIZE	• <xxx>XML<xxx>

Table 5. Unsupported scalar functions (continued)

• GENERATE_UNIQUE	• RAISE_ERROR	

- The instruction to process “INSERT FROM SELECT” 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.

Related concepts:

“EXPLAIN information” on page 84
After creating the necessary EXPLAIN tables, you can analyze queries by invoking the DB2 EXPLAIN function. Such queries are accepted by regular and virtual accelerators alike. 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.

Related information:

How IBM DB2 Analytics Accelerator for z/OS handles correlated subqueries

Isolation levels

DB2 isolation levels are ignored by IBM DB2 Analytics Accelerator for z/OS.

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. The accelerator 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 of a DB2 table are propagated automatically to the associated table on the accelerator only if the incremental update feature is used. If incremental updates are not enabled, you must update the tables on an accelerator 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 tables is an asynchronous process, that is, there is no guarantee that the data in the DB2 tables and the data in the accelerator tables are always in sync. 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 for z/OS. 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 for z/OS. Result sets that are returned by IBM DB2 Analytics Accelerator for z/OS do not necessarily comply with any isolation level setting.

To ensure data consistency on a relatively safe level, update the table data on the accelerator 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 spropagated to the accelerator by calling the ACCEL_CONTROL_ACCELERATOR stored procedure with the **<waitForReplication>** command.

So 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 continuously" on page 106
The incremental update function of IBM DB2 Analytics Accelerator for z/OS allows you to update accelerator tables continuously. Changes to the data in original DB2 for z/OS tables are thus propagated to the corresponding accelerator 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" on page 74
Successful queries against tables on an accelerator 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 for z/OS

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 for z/OS might return for the same query.

- Columns encoded in EBCDIC MIXED or DBCS are only partially supported by IBM DB2 Analytics Accelerator for z/OS. This is why you must explicitly enable the routing of queries that refer to such columns to an accelerator. In DB2, the data in such columns is encoded in EBCDIC, but in IBM DB2 Analytics Accelerator for z/OS, it is encoded in Unicode (UTF-8). These encoding schemes have significant differences that can lead to different result sets depending on whether the query is executed in DB2 or on the accelerator. **Examples:** EBCDIC and Unicode implement different collating sequences. The collating sequence for Unicode is numeric, uppercase characters, and then lowercase characters (1, 2, 3, A, B, C, a, b, c). In EBCDIC, the collating sequence is lowercase, uppercase, and then numeric characters (a, b, c A, B, C, 1, 2, 3). The collating sequences for national characters are also different. This affects both, data ordering and the results from range predicates. Therefore, if the tables include character columns where the column values belong to more than one of these groups and the SQL statements include range predicates or ordering on these columns, a query executed in DB2 might return a different result set than the same query executed in IBM DB2 Analytics Accelerator for z/OS.

  Another prominent example is the different handling of blanks in some cases.

  On the contrary, there are many situations in which DB2 data in multi-byte EBCDIC tables can be processed by IBM DB2 Analytics Accelerator for z/OS without problems. However, to avoid problems, always contact IBM support before you connect an accelerator to a DB2 subsystem that contains table data in EBCDIC MIXED or DBCS encoding.
- During query routing to an accelerator, the aggregate DB2 function COUNT_BIG is converted to the COUNT function, which is the corresponding accelerator function. The DB2 COUNT_BIG function returns DECIMAL(31,0) values, while

the accelerator COUNT function returns BIGINT values. If the result returned by the accelerator function COUNT exceeds the maximum length that can be represented by the BIGINT data type, an accelerated query fails. The same query would succeed in DB2 as long as the result of COUNT_BIG can be represented in a DECIMAL(31,0) value.

- 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 for z/OS will always issue an error in cases like these.
- 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 for z/OS 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
- DB2 accepts an argument in the format `yyyynnn` as input for the DATE function. IBM DB2 Analytics Accelerator for z/OS does not accept this format and therefore issues an error. Similarly, DB2 accepts an argument in the format `yyyyxxddhhmmss` and also a 13-byte string as input for the TIMESTAMP function, whereas IBM DB2 Analytics Accelerator for z/OS does not. Queries that contain strings in these formats are routed to the accelerator none the less because DB2 bind time is unaware of this incompatibility. The string could be a valid timestamp string.
- During query routing to an accelerator, the DATE(numeric_exp) scalar function is converted to `DATE(to_date('0000-12-31','YYYY-MM-DD')+trunc(numeric_exp))`, If the `numeric_exp` result is a negative number, IBM DB2 Analytics Accelerator for z/OS adds `BC` to the date. An error is returned when the converted scalar function is evaluated in DB2.
- The DB2 scalar functions LOCATE and LOCATE_IN_STRING are converted to the INSTR. That is, `LOCATE(search,source,start)` ends up as `INSTR(source,search,start)` and `LOCATE_IN_STRING(source,search,start,instance)` ends up as `INSTR(source,search,start,instance)`. If the starting point is an expression that returns a negative value, IBM DB2 Analytics Accelerator for z/OS evaluates the expression from the right with an offset of 1. If the converted query is executed in DB2 (due to a FAILBACK, for example), LOCATE returns an error with `SQLCODE171`, while LOCATE_IN_STRING will give you a result that differs from the result returned by the original DB2 query.

- If the third parameter in the LPAD or RPAD function is an empty string, IBM DB2 Analytics Accelerator for z/OS issues an error. DB2 can process such queries successfully.
- The result data type of the MONTHS_BETWEEN scalar function in DB2 is DECIMAL(31,15). In IBM DB2 Analytics Accelerator for z/OS, it is NUMERIC(17,9). As a consequence, the query results contain fewer digits if such a query is routed to and processed by IBM DB2 Analytics Accelerator for z/OS.
- If the SUBSTR scalar function (`SUBSTR(`*`string-expression`*`, `*`start`*`, `*`length`*`)`) uses a start position that is greater than the value of *`length`*, DB2 returns `SQLCODE138`. IBM DB2 Analytics Accelerator for z/OS returns an empty string.
- If the value of *`from-str`* in the TRANSLATE scalar function (`TRANSLATE(`*`str`*`, `*`to-str`*`, `*`from-str`*`)`) is an expression that returns an empty string, DB2 returns `SQLCODE171`. IBM DB2 Analytics Accelerator for z/OS returns the first argument.
- 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 for z/OS, 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 6 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 6. 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.
- Version 4 PTF-2 of the 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.

  Restrictions:

  – The value 23:59:59.999999 must not occur in your data and must not be used in SQL expressions because it is needed internally for the automatic conversion.

  – 24:00:00 can never be the result of an arithmetic calculation. For example, 00:00:00 + 24 hours will be 00:00:00 (on the next day) rather than 24:00:00, as in DB2 for z/OS.

If you have used the LOAD_ENABLE_HOUR24_CONVERSION parameter so far, and now want to use the native 24:00:00 support, you must set this parameter to the value `false`. A menu entry has been added to the IBM DB2 Analytics Accelerator Console for doing this, so that the conversion can be switched off without the help of an IBM support person. If your accelerators contain tables with converted values, you have to reload these tables after resetting the parameter. For more information, follow the appropriate link at the end of this topic.

For versions earlier than version 4 PTF-2, the restriction regarding the 24:00:00 time value still exists. This means: IBM DB2 Analytics Accelerator for z/OS cannot process queries if the value 24:00:00 occurs in a TIME or TIMESTAMP expression, and table columns containing such values cannot be loaded on an accelerator.

- DB2 and IBM DB2 Analytics Accelerator for z/OS might return different results if the UPPER, LOWER, or TRANSLATE scalar functions are used. The differences are as follows:

  Functions that do not use a locale setting (specified explicitly or by means of a special register)

  When a query is not routed to an accelerator, only the plain Latin letters from a-z and from A-Z are converted to uppercase or lowercase. If the incoming data is encoded using a MIXED or a double-byte character set (DBCS), the entire set of Latin characters is converted.

  When a query is routed to an accelerator, the characters with diacritical marks are converted in addition to the plain Latin characters from a-z or from A-Z. In other words, more characters are converted to uppercase or lowercase if a query is accelerated.

  For functions that do use a locale setting (specified explicitly or by means of a special register)

  When a query is not routed to an accelerator, special characters (as defined in the Unicode file `SpecialCasing.txt`) are converted to uppercase or lowercase.

  When the query is routed to an accelerator, these special characters are not converted.

  The might lead to the following differences in your query results:

  – DB2 and IBM DB2 Analytics Accelerator for z/OS return different numbers of rows for the same query if the UPPER or LOWER function is used in a predicate.
  – DB2 and IBM DB2 Analytics Accelerator for z/OS return rows in different orders or group rows differently if the UPPER or LOWER function is used in an ORDER BY or GROUP BY clause.
  – DB2 and IBM DB2 Analytics Accelerator for z/OS return different data. For example, the results of any aggregate function can differ if UPPER or LOWER is used in a GROUP BY clause or in a predicate. Some characters will show a different case (uppercase in DB2 results, lowercase in accelerator results, or vice versa).
  – Other differences are possible, depending on where the UPPER or LOWER function appears in the query.

- 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.

- If the second argument of the LEFT scalar function contains a parameter marker or value that is longer than the value of the first argument, the padding of the second argument value in IBM DB2 Analytics Accelerator for z/OS might be different from the padding in DB2 for z/OS.
- During query routing to an accelerator, the DB2 UTF-8 data type `CHAR(n)` is converted to the `NVARCHAR(n)` data type. Bear this in mind if your queries contain rules regarding the result data type. These can be, for example, result expressions that use the CASE function on set operations such as UNION, INTERSECT, or EXCEPT. The result data type will be CHAR in DB2 and VARCHAR in IBM DB2 Analytics Accelerator for z/OS. This affects the output if DB2 for z/OS internally uses padding to cast the results to the expected data type. Inhouse DB2 queries and accelerated queries might return different results.
- IBM DB2 Analytics Accelerator for z/OS does not support all date and time formats that DB2 supports. This might lead to different results for accelerated and inhouse DB2 queries if the VARCHAR_FORMAT or TIMESTAMP_FORMAT scalar function is used. The following formats are not supported by IBM DB2 Analytics Accelerator for z/OS:
  - `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 1-6. 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
- IBM DB2 Analytics Accelerator for z/OS cannot handle correlated subqueries if the subquery cannot be de-correlated internally. If a query that was routed to an accelerator contains a correlated subquery, the query might fail. In this case, the query is returned to DB2 for processing (FAILBACK). To find out which correlated subqueries are supported, see the external document that is referenced under **Related information** at the end of this topic.
- Results can be different if a query contains parameter markers.
- DB2 and IBM DB2 Analytics Accelerator for z/OS might return different results in connection with the NEXT_DAY scalar function if the function handles years before the year 1582. For years before 1582, DB2 uses Julian calendar dates while IBM DB2 Analytics Accelerator for z/OS continues to use Gregorian calendar dates.
- If the second argument of the ROUND or TRUNC scalar function is less than -38 or greater than 37, DB2 for z/OS can still evaluate the expression, whereas IBM DB2 Analytics Accelerator for z/OS will return an error.
- 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 for z/OS 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 for z/OS 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 10 for z/OS information center. A link to this topic is provided under **Related information**.

Related concepts:

"Differences in floating-point error tolerances" on page 40
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.

Related tasks:

"Switching from 24:00:00 conversion to native 24:00:00 support" on page 134
If you have so far converted 24:00:00 time values to 23:59:59 by using the `LOAD_ENABLE_HOUR24_CONVERSION` configuration parameter and now want to use the native 24:00:00 support provided by newer versions of the Netezza database, you must remove this parameter or set it to the value `false`. In addition, you must reload accelerator tables that contain converted values.

Related information:

How IBM DB2 Analytics Accelerator for z/OS handles correlated subqueries

Arithmetic with two decimal operands

Incompatibilities related to IBM InfoSphere Change Data Capture 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 Change Data Capture 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 for z/OS* also apply to IBM InfoSphere Change Data Capture for z/OS.

In addition, incompatibilities that affect the regular load function of IBM DB2 Analytics Accelerator for z/OS (the SYSPROC.ACCEL_LOAD_TABLES stored procedure) also affect IBM InfoSphere Change Data Capture for z/OS. However, the following exceptions exist when IBM InfoSphere Change Data Capture for z/OS transfers data to IBM DB2 Analytics Accelerator for z/OS:

- Values of VARCHAR FOR MIXED and VARGRAPHIC columns that exceed 16352 characters are cut off when they are received by IBM DB2 Analytics Accelerator for z/OS. The difference to DB2 for z/OS is that in the case of IBM InfoSphere Change Data Capture for z/OS, *no warning or error message* is issued.
- In contrast to the handling of ideographic space characters (U+3000) coming from DB2 for z/OS, which are converted to U+0020 characters when they are received by the Netezza system, *no conversion* occurs when such characters are received from IBM InfoSphere Change Data Capture for z/OS. U+3000 characters remain unchanged when they are transferred to the Netezza database. This means that operations on the data in the Netezza database might not show the intended result. For example, an execution of the **trim** command will not remove leading or trailing spaces.

New features in DB2 10 for z/OS

Many new features that were introduced with DB2 10 for z/OS are not yet supported by IBM DB2 Analytics Accelerator for z/OS.

General

The EXPLAIN reason codes that hint at routing errors in connection with new features in DB2 10 for z/OS 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 for z/OS support* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*

Access control of columns or rows

IBM DB2 Analytics Accelerator for z/OS handles columns with access control or tables with row access control as follows:

- Columns with access control are ignored when the corresponding table is loaded on the accelerator.
- 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.

Implicit casting

In DB2 10 for z/OS, string data types and numeric data types can be made compatible by an implicit casting of the string data type to the DECFLOAT data type. Because IBM DB2 Analytics Accelerator for z/OS does not support the DECFLOAT data type, queries containing such constructs cannot be routed to an accelerator.

New OLAP functions

IBM DB2 Analytics Accelerator for z/OS supports the following moving aggregate functions:

- AVG
- CORRELATION (see the **Notes**)
- COUNT

- COUNT_BIG
- COVARIANCE (see the **Notes**)
- COVARIANCE_SAMP (see the **Notes**)
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE (VARIANCE_SAMP)

Notes:

- The moving aggregate functions CORRELATION, COVARIANCE, and COVARIANCE_SAMP are supported only if none of the arguments uses the DECFLOAT data type, which cannot be handled by IBM DB2 Analytics Accelerator for z/OS.
- The result data type that IBM DB2 Analytics Accelerator for z/OS returns for the CORRELATION, COVARIANCE, and COVARIANCE_SAMP functions is DOUBLE, and not DECFLOAT, as in DB2 10 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;
  ```

For a complete list of the new OLAP functions, see the DB2 10 for z/OS documentation.

Syntax

If a query contains syntactical constructs that are supported by DB2 10 for z/OS, but not DB2 Version 9.1 for z/OS, the query is not routed to IBM DB2 Analytics Accelerator for z/OS. For example, a query containing TIMESTAMP(6) will not be routed to an accelerator, although the construct delivers the same result as CURRENT TIMESTAMP.

Temporal tables

DB2 10 for z/OS introduces temporal tables to keep track of system time intervals and business time intervals. Queries referencing such tables usually contain the following expressions or clauses, which are not supported by IBM DB2 Analytics Accelerator for z/OS:

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

Queries containing any of these expressions are not routed to an accelerator.

Timestamps

The portion that denotes fractional seconds in timestamps can have up to 12 digits in DB2 10 for z/OS, which means that nanoseconds and even picoseconds can be represented. IBM DB2 Analytics Accelerator for z/OS does not support timestamps if the precision of fractional seconds is anything else than six digits (microseconds), to the effect that:

- Timestamp columns defined with a fractional seconds precision other than six digits (default) are not loaded into tables on the accelerator.
- SQL queries that require TIMESTAMP with a fractional seconds precision other than six as the result data type are not routed to an accelerator.
- New date/time constants with a precision other than six are not accepted.
- Scalar functions that take parameters in the high-precision format are not accepted.

Timestamps including timezone specifications are also 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.
- New functions and special registers that are related to the new timestamp features 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 for z/OS regardless of the content of the string expression because the string constant or expression is not validated.

Related concepts:

"EXPLAIN information" on page 84
After creating the necessary EXPLAIN tables, you can analyze queries by invoking the DB2 EXPLAIN function. Such queries are accepted by regular and virtual accelerators alike. 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.

New features in DB2 11 for z/OS

See which new features in DB2 11 for z/OS are supported by IBM DB2 Analytics Accelerator for z/OS and which are not.

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 that application scope.

Global variables are not supported by IBM DB2 Analytics Accelerator for z/OS. A query that contains a global variable statement is hence not routed to an accelerator.

Furthermore, the following IBM DB2 Analytics Accelerator for z/OS special registers cannot be set to the value of a global variable:

- CURRENT GET_ACCEL_ARCHIVE
- CURRENT_QUERY_ACCELERATION

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. DB2 11 for z/OS 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 supports 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 table, a view, or another 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 for z/OS Version 4.1.0 does not support aliases if these are used in a sequence object.

SQL grouping sets and supergroups

A grouping-sets specification allows multiple grouping clauses to be specified in a single statement. This can be thought of as the union-all of two or more groups of rows into a single result set. It is logically equivalent to the union-all of multiple subselects with the GROUP BY clause in each subselect corresponding to one grouping set. A grouping set can be a single element or can be a list of elements delimited by parentheses and commas, where an element is either a grouping-expression or a supergroup. Grouping sets allow the computing of groups with just a single pass over the base table.

A supergroup stands for a ROLLUP clause, a CUBE clause, or a grand-total clause. DB2 11 for z/OS also introduces a new aggregate function named GROUPING. This function is used in conjunction with grouping sets and supergroups to return a value that indicates whether a row that is returned in a GROUP BY answer set is a null value generated by a grouping set. IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 does not support queries that include grouping set statements. Therefore, such queries are not routed to an accelerator.

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 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 FLOAT numbers:

Table 7. Minimum and maximum REAL and FLOAT values (HFP and IEEE754)

	HFP (DB2 for z/OS)		**IEEE754 (IBM DB2 Analytics Accelerator for z/OS)**	
	REAL	FLOAT	REAL	FLOAT
Min.	~-7.2* 10^{75}	~-7.2* 10^{75}	~-3.4 * 10^{38}	~-7.2* 10^{75}
Max.	~7.2 * 10^{75}	~7.2 * 10^{75}	~3.4 * 10^{38}	~7.2 * 10^{75}
Min. positive	~5.4 *10^{-79}	~5.4 *10^{-79}	~1.18*10^{-38}	~5.4 *10^{-79}

Table 7. Minimum and maximum REAL and FLOAT values (HFP and IEEE754) (continued)

	HFP (DB2 for z/OS)		**IEEE754 (IBM DB2 Analytics Accelerator for z/OS)**	
Max. negative	~-5.4*10^{-79}	~-5.4*10^{-79}	~-1.18*10^{-38}	~-5.4*10^{-79}

Chapter 4. IBM DB2 Analytics Accelerator Studio

IBM DB2 Analytics Accelerator for z/OS is a combined hardware and software solution that accelerates database queries. You control this solution from IBM DB2 Analytics Accelerator Studio, which is installed on a remote computer, typically a notebook or PC.

Instead of using IBM DB2 Analytics Accelerator Studio, you can also control and operate IBM DB2 Analytics Accelerator for z/OS by using DB2 commands and the stored procedures that come with the product.

Launching an IBM DB2 Analytics Accelerator Studio function causes IBM DB2 Analytics Accelerator Studio 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 for z/OS. 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:

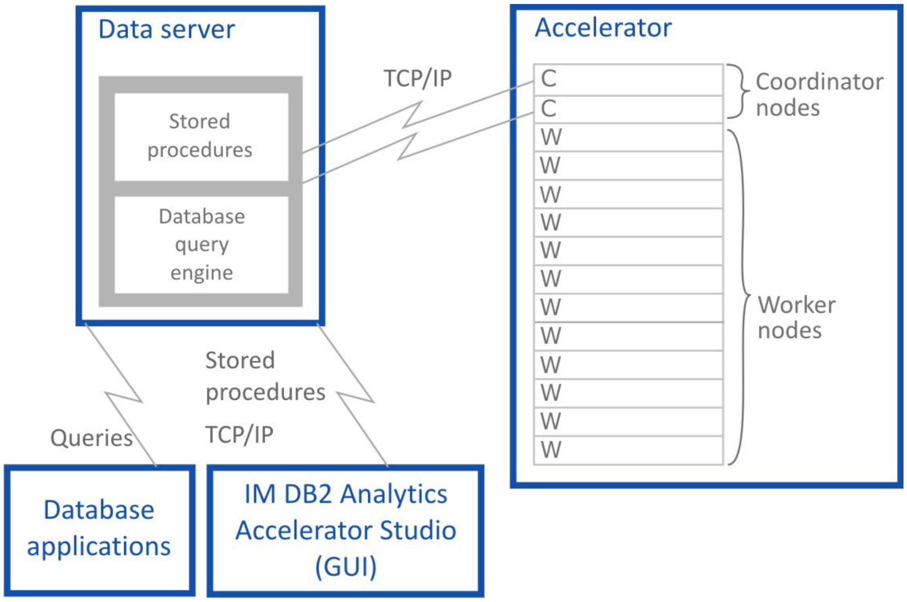


Figure 3. IBM DB2 Analytics Accelerator for z/OS setup

© Copyright IBM Corp. 2009, 2014

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.

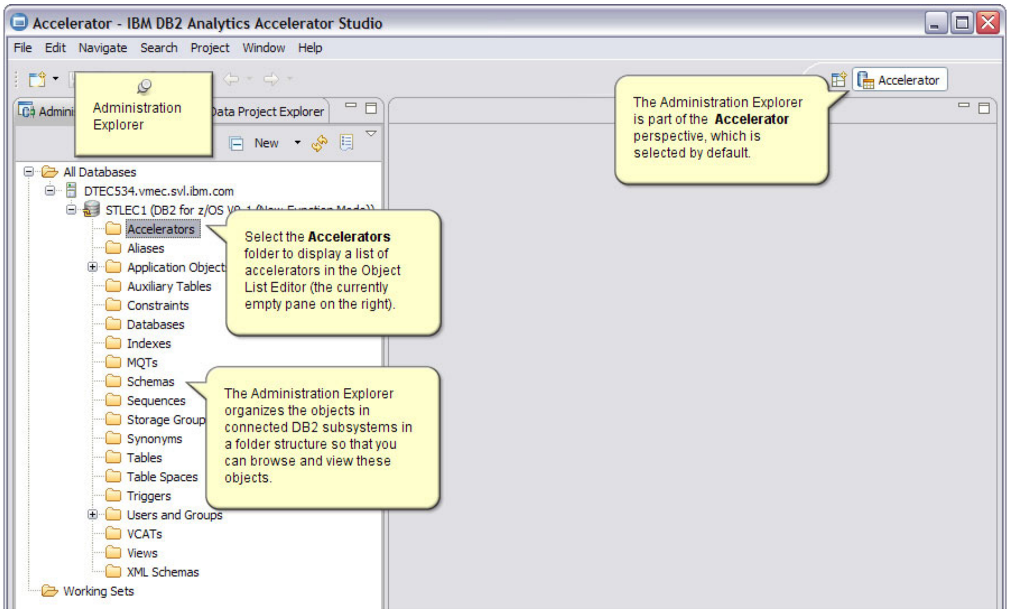
Help > Welcome > Workbench
Closes the Welcome screen and displays the main view of IBM DB2 Analytics Accelerator Studio.

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 4 on page 46. It is usually located on the left-hand side.

© Copyright IBM Corp. 2009, 2014



The Administration Explorer is part of the **Accelerator** perspective, which is selected by default.

Select the **Accelerators** folder to display a list of accelerators in the Object List Editor (the currently empty pane on the right).

The Administration Explorer organizes the objects in connected DB2 subsystems in a folder structure so that you can browse and view these objects.

Figure 4. 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.

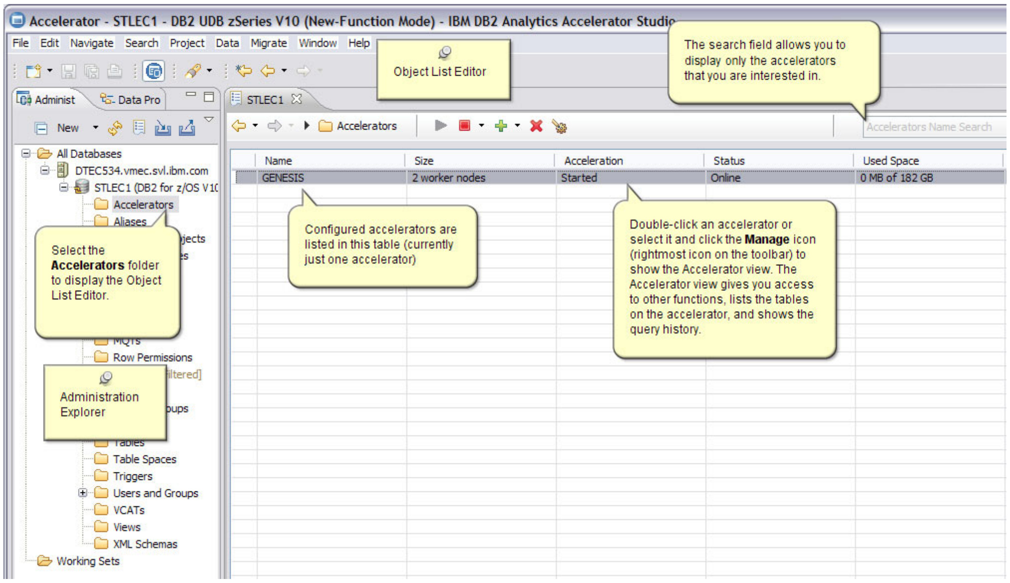
Add Virtual Accelerator
Starts the Add Virtual Accelerator wizard, which allows you to add a virtual accelerator to the system configuration.

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.



Accelerator - STLEC1 - DB2 UDB zSeries V10 (New-Function Mode) - IBM DB2 Analytics Accelerator Studio

Name	Size	Acceleration	Status	Used Space
GENESIS	2 worker nodes	Started	Online	0 MB of 182 GB

Object List Editor

The search field allows you to display only the accelerators that you are interested in.

Select the **Accelerators** folder to display the Object List Editor.

Configured accelerators are listed in this table (currently just one accelerator)

Double-click an accelerator or select it and click the **Manage** icon (rightmost icon on the toolbar) to show the Accelerator view. The Accelerator view gives you access to other functions, lists the tables on the accelerator, and shows the query history.

Administration Explorer

Figure 5. 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 error allows you to select a particular later display from a list.

▸ 📁 Accelerators

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 open the Add Accelerator wizard. See the previous description for the context menu of the Accelerators folder.

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

Add Accelerator
Same as clicking the button on the toolbar

Add Virtual Accelerator
Starts the wizard for adding a virtual accelerator

✖

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 the tables on the accelerator 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 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.

Configured
The incremental update function is available for this accelerator, but tables have not yet been added to the process.

Error The incremental update function has been configured, but cannot be started due to an error.

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

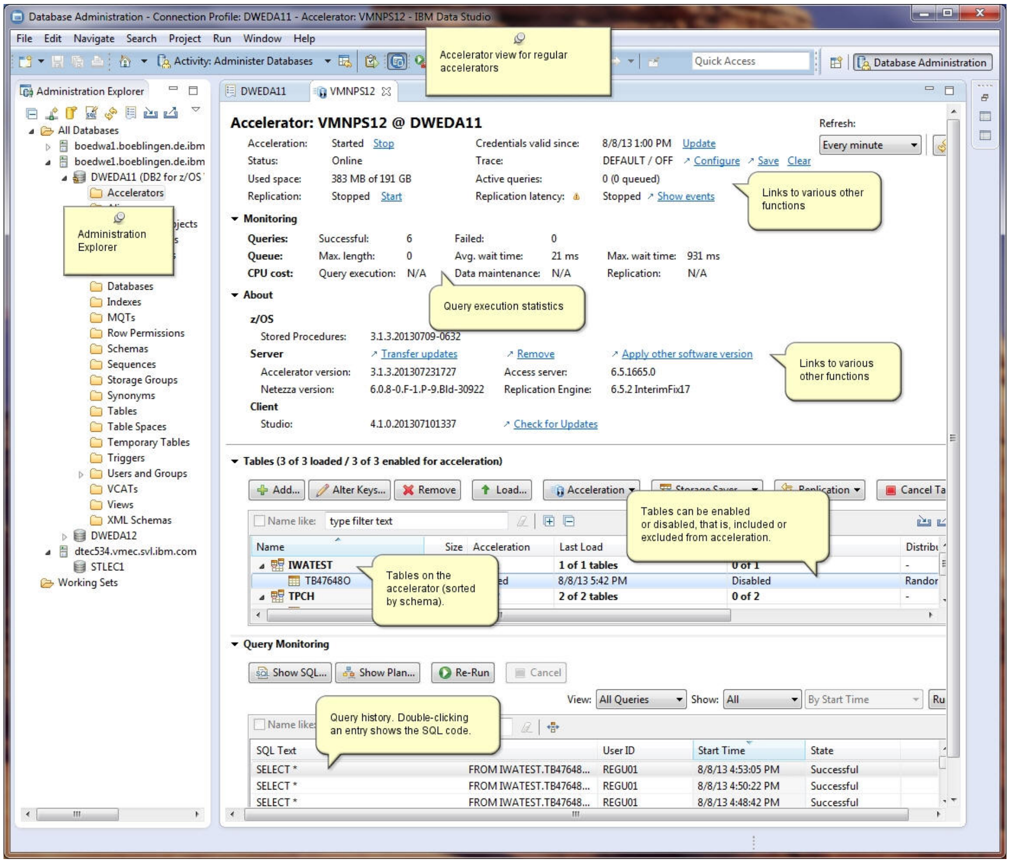
Stopped
The incremental update function is available and tables have been included in the process. However, the function is not running right now.

Accelerator views

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.

Accelerator view for regular accelerators

Here is a brief description of the status information and the function buttons on the Accelerator view for regular accelerators.


Database Administration - Connection Profile: DWEDA11 - Accelerator: VMNPS12 - IBM Data Studio
File Edit Navigate Search Project Run Window Help
Activity: Administer Databases
Quick Access
Database Administration
Accelerator view for regular accelerators
Administration Explorer
All Databases
boedwa1.boeblingen.de.ibm
boedwe1.boeblingen.de.ibm
DWEDA11 (DB2 for z/OS
Accelerators
Administration Explorer
Databases
Indexes
MQTs
Row Permissions
Schemas
Sequences
Storage Groups
Synonyms
Tables
Table Spaces
Temporary Tables
Triggers
Users and Groups
VCATs
Views
XML Schemas
DWEDA12
dtec534.vmec.svl.ibm.com
STLEC1
Working Sets
DWEDA11
VMNPS12
Links to various other functions
Query execution statistics
Links to various other functions
Tables can be enabled or disabled, that is, included or excluded from acceleration.
Tables on the accelerator (sorted by schema).
Query history. Double-clicking an entry shows the SQL code.
Add...
Alter Keys...
Remove
Load...
Acceleration
Storage Saver
Replication
Cancel Ta
Show SQL...
Show Plan...
Re-Run
Cancel
Ru

Accelerator: VMNPS12 @ DWEDA11

Refresh: Every minute

Acceleration: Started Stop
Status: Online
Used space: 383 MB of 191 GB
Replication: Stopped Start
Credentials valid since: 8/8/13 1:00 PM Update
Trace: DEFAULT / OFF ↗ Configure ↗ Save Clear
Active queries: 0 (0 queued)
Replication latency: Stopped ↗ Show events

Monitoring

Queries:	Successful: 6	Failed:	0		
Queue:	Max. length: 0	Avg. wait time:	21 ms	Max. wait time:	931 ms
CPU cost:	Query execution: N/A	Data maintenance:	N/A	Replication:	N/A

About

z/OS
Stored Procedures: 3.1.3.20130709-0632
Server ↗ Transfer updates ↗ Remove ↗ Apply other software version
Accelerator version: 3.1.3.201307231727 Access server: 6.5.1665.0
Netezza version: 6.0.8-0.F-1.P-9.Bld-30922 Replication Engine: 6.5.2 InterimFix17
Client
Studio: 4.1.0.201307101337 ↗ Check for Updates

Tables (3 of 3 loaded / 3 of 3 enabled for acceleration)

□ Name like: type filter text

Name	Size	Acceleration	Last Load		Distribu
IWATEST			**1 of 1 tables**	**0 of 1**	-
TB47648O		ed	8/8/13 5:42 PM	Disabled	Randor
TPCH			**2 of 2 tables**	**0 of 2**	-

Query Monitoring

View: All Queries Show: All By Start Time

□ Name like

SQL Text		User ID	Start Time	State
SELECT *	FROM IWATEST.TB47648...	REGU01	8/8/13 4:53:05 PM	Successful
SELECT *	FROM IWATEST.TB47648...	REGU01	8/8/13 4:50:22 PM	Successful
SELECT *	FROM IWATEST.TB47648...	REGU01	8/8/13 4:48:42 PM	Successful

Figure 6. Accelerator view for regular accelerators

Status information and functional links

The header of the Accelerator view for regular accelerators contains status information and links to related functions.

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.

Credentials valid since
Shows the date on which the last authentication of the accelerator took place. To renew the authentication, click **Update**.

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**. To delete the trace information that has been collected, click **Clear**.

Active queries
The number of accelerated queries that are currently being processed. In parentheses, you see the number of queued queries.

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**.

Replication latency
Only visible if incremental updates are configured. This is not the case in Figure 6 on page 51. 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 tables.

Medium
Recent changes to the original DB2 tables will be propagated to the corresponding accelerator tables within three and six minutes.

High
It will take six minutes or longer to apply recent changes to DB2 tables to the corresponding tables on the accelerator.

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 tables that have been included in the incremental update process.

Deletes
Number of row deletions from accelerator 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 translates to 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.

Source Agent

Number of data changes to relevant DB2 for z/OS tables (tables that have been added to the accelerator and 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 protocols all transactions that are related to incremental updates.

Monitoring

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:

Max. length

The highest number of queries in the queue, up to this time.

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.

Max. wait time

Longest time that a query stayed in the queue, that is, before it was processed

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

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 PureData™ System for Analytics:

Accelerator
The version of the IBM DB2 Analytics Accelerator for z/OS software that is currently in use. To transfer an update package from the hierarchical file system (HFS) on z/OS to the accelerator, click **Transfer updates**. To remove an obsolete version no longer used, click **Remove**. To use a different version of the IBM DB2 Analytics Accelerator for z/OS software, click **Apply other software version**.

Netezza Performance Server (NPS)
The current version of the Netezza Performance Server (NPS).

Before you install a new version of the Netezza Performance Server (NPS), make sure that the required minimum versions of the Netezza Firmware (FDT) and the Netezza Host Platform (HPF) are already installed.

To transfer an update package from the hierarchical file system (HFS) on z/OS to the IBM PureData System for Analytics, click **Transfer files**.

Netezza Firmware (FDT)
Only visible if version 4 or later of the IBM DB2 Analytics Accelerator for z/OS is installed on the accelerator. This is the current version of the Netezza Firmware and Diagnostics and Tools (FDT) running on the accelerator (IBM PureData System for Analytics). To transfer an FDT update package from the hierarchical file system (HFS) on z/OS to the IBM PureData System for Analytics, click **Transfer files**. The installation of the Netezza Firmware (FDT), however, must be carried out by a service engineer and requires a downtime of the IBM PureData System for Analytics.

Netezza Host Platform (HPF)
Only visible if version 4 or later of the IBM DB2 Analytics Accelerator for z/OS is installed on the accelerator. This is the current version of the Netezza Host Platform (HPF) running on the accelerator (IBM PureData System for Analytics). The HPF software contains kernel updates, firmware, and other special software used by the Netezza host servers.

To transfer an FDT update package from the hierarchical file system (HFS) on z/OS to the IBM PureData System for Analytics, click **Transfer files**. The installation of the Netezza Host Platform (HPF), however, must be carried out by a service engineer and requires a downtime of the IBM PureData System for Analytics.

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 PureData System for Analytics, according to the instructions received from the CDC Access Server.

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.

Buttons

The Accelerator view for regular accelerators provides the following buttons:

Add Starts the Add Tables wizard.

Alter Keys

Opens a window in which you can select or change the distribution and organizing keys that are used by the accelerator.

Remove

Removes the selected tables from the accelerator.

Load Opens the Load Tables window, in which you can select tables for loading and set the lock scope for the load process.

Acceleration

Opens a pull-down menu, on which you can enable or disable the selected tables so that these are included or excluded from query acceleration.

Storage Saver

Allows you to move partitions to an accelerator or restore these to their original locations. The button opens a menu with the following choices:

Move Partitions to Accelerator

Opens the Move Storage Saver Partitions to Accelerator window, which allows you to select DB2 for z/OS tables or partitions in order to move the data therein to the accelerator currently viewed.

Restore Partitions to DB2

Opens Restore Storage Saver Partitions to DB2 window, in which you can select tables and partitions that have been moved to the currently selected accelerator in order to restore these.

Replication

Only visible if incremental updates are configured. This is not the case in Figure 6 on page 51. Clicking the button opens a pull-down menu from which you can enable or disable incremental updates for selected tables.

Cancel Tasks

Opens the Cancel Tasks window, which allows you to select and cancel tasks running on the currently viewed accelerator, such as a load table process.

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 92.

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:

“How to select tables for query acceleration” on page 12
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” on page 68
Using the Add Tables to Accelerator wizard, you can define database tables on an accelerator.

“Loading tables” on page 74
Successful queries against tables on an accelerator 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” on page 77
You can permit or prevent the sending of queries to an accelerator by enabling or disabling the corresponding tables.

“Specifying or changing a distribution key or organizing keys” on page 80
Distribution and organizing keys have a considerable impact on the query response time.

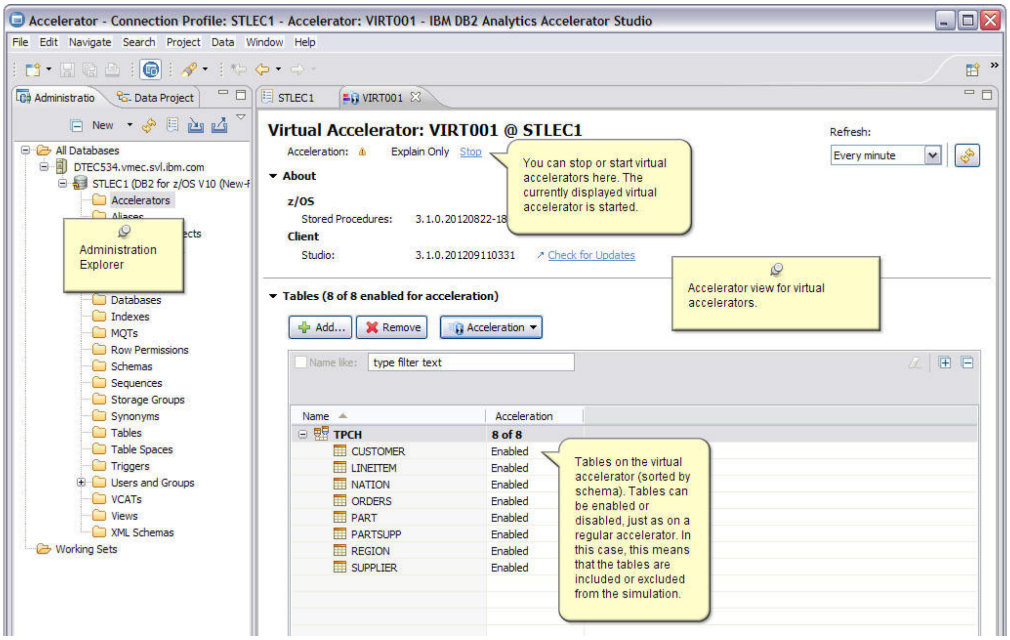
“Selecting and ordering query monitoring columns” on page 92
You can select the columns that are displayed in the **Query Monitoring** section of the Accelerators view and also change their order of appearance.

Related information:

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

Accelerator view for virtual accelerators

Here is a brief description of the status information and the function buttons on the Accelerator view for virtual accelerators.



Accelerator - Connection Profile: STLEC1 - Accelerator: VIRT001 - IBM DB2 Analytics Accelerator Studio

Administration Explorer

Virtual Accelerator: VIRT001 @ STLEC1

Acceleration: Explain Only Stop

Refresh: Every minute

You can stop or start virtual accelerators here. The currently displayed virtual accelerator is started.

About

z/OS

Stored Procedures: 3.1.0.20120822-18

Client

Studio: 3.1.0.201209110331 Check for Updates

Accelerator view for virtual accelerators.

Tables (8 of 8 enabled for acceleration)

Name	Acceleration
TPCH	8 of 8
CUSTOMER	Enabled
LINEITEM	Enabled
NATION	Enabled
ORDERS	Enabled
PART	Enabled
PARTSUPP	Enabled
REGION	Enabled
SUPPLIER	Enabled

Tables on the virtual accelerator (sorted by schema). Tables can be enabled or disabled, just as on a regular accelerator. In this case, this means that the tables are included or excluded from the simulation.

Figure 7. Accelerator view for virtual accelerators

Status information and functional links

The header of the Accelerator view for virtual accelerators contains status information and a functional link.

Acceleration
The status can be **Stopped** or **Explain Only**. These states indicate whether the virtual accelerator is currently enabled in DB2 for z/OS. The **Explain Only** state, as opposed to **Started**, is a reminder that this is a virtual accelerator rather than a regular accelerator. To change the current status, click **Start** or **Stop**.

About

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

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.

Buttons

The Accelerator view for virtual accelerator provides the following buttons:

Add Starts the Add Tables wizard.

Remove
Removes the selected tables from the virtual accelerator.

Acceleration

Opens a pull-down menu, on which you can enable or disable the selected tables so that these are included or excluded from the simulation.

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:

"How to select tables for query acceleration" on page 12
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" on page 68
Using the Add Tables to Accelerator wizard, you can define database tables on an accelerator.

"Enabling tables for query acceleration" on page 77
You can permit or prevent the sending of queries to an accelerator by enabling or disabling the corresponding tables.

Related information:

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

Chapter 6. Tasks overview

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.

© Copyright IBM Corp. 2009, 2014

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

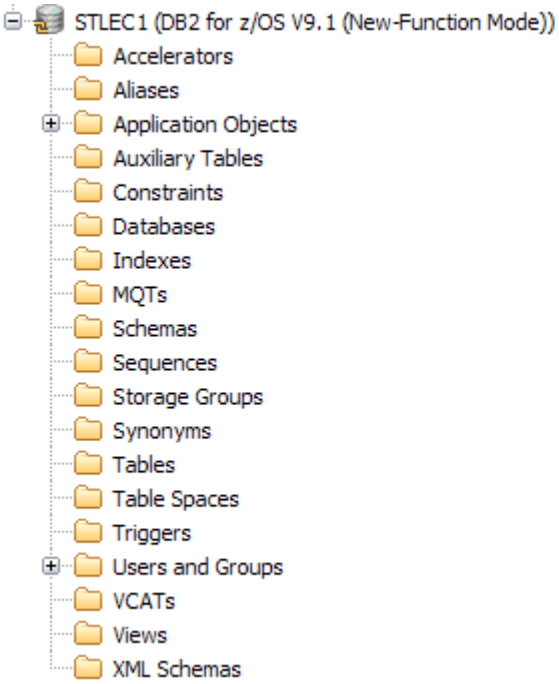
In the Administration Explorer, double-click the icon representing the DB2 subsystem. For example:

STLEC1

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



- STLEC1 (DB2 for z/OS V9.1 (New-Function Mode))
  - Accelerators
  - Aliases
  - Application Objects
  - Auxiliary Tables
  - Constraints
  - Databases
  - Indexes
  - MQTs
  - Schemas
  - Sequences
  - Storage Groups
  - Synonyms
  - Tables
  - Table Spaces
  - Triggers
  - Users and Groups
  - VCATs
  - Views
  - XML Schemas

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 shows the tables that have been added to this accelerator.

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 (virtual IP or wall IP address) of the accelerator. Make a note of this information. You need to enter it as you complete the steps that follow.
2. 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.
3. Log on to TSO/ISPF.
4. 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 10.101.8.8 1600
   ```

5. When prompted, enter the console password. The initial password is `dwa-1234`. You must change this password at the first logon.
6. Press the Pause key, then Enter to display the following screen:

```
******************************************************************************
*        Welcome to the IBM DB2 Analytics Accelerator Console
******************************************************************************

You have the following options:

 (1) - Change the Configuration Console Password
 (2) - (Menu) Run Netezza Commands
 (3) - (Menu) Run Accelerator Functions
 (4) - (Menu) Manage Hardware
 (5) - (Menu) Manage Incremental Updates

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

7. Type 3 and press Enter to display the submenu:

```
 main -> 3
 --------------------------------------------------------
 You have the following options:

 (0) - Go back one level
 (1) - Obtain pairing code, IP address, and port
 (2) - List paired DB2 subsystems
 (3) - Set resource limits for DB2 subsystems
 (4) - Clear query history
 (5) - Specify the priority of maintenance tasks
 (6) - Set the DB2 subsystem for time synchronization
 (7) - Restart accelerator process
 (8) - Disable the conversion mode for 24:00:00
```

8. Type 1 and press Enter:
9. 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    : 9.152.85.192
   Port          : 1400
   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.

10. Make a note of the following information on the console:
    - `Pairing code`
    - `IP address`
    - `Port`
11. Press Enter to return to the main menu of the console.
12. Type `x` and press Enter to exit the console and close the telnet session.

Related tasks:

"Adding accelerators"
To complete the authentication, you enter the IP address, port number, and the pairing code in the Add Accelerator wizard.

Related information:

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

Adding accelerators

To complete the authentication, you enter the IP address, 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 clear text 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**.

Related tasks:

"Obtaining the pairing code for authentication" on page 62
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" on page 142
You can ping the accelerators, but you cannot establish a distributed relational database access (DRDA) connection between your database management system and the IBM PureData System for Analytics.

Adding virtual accelerators

Virtual accelerators are simulators used for query testing and evaluation. Using such an accelerator, you can check whether a query can be accelerated, whether errors occur during processing, and whether the response time benefit is high enough to justify acceleration.

Before you begin

Make sure that the user running the test queries has the following privileges:

- SELECT, INSERT and DELETE privilege on the SYSACCEL.SYSACCELERATORS table
- SELECT, INSERT, DELETE, UPDATE privilege on the SYSACCEL.SYSACCELERATEDTABLES table
- MONITOR privilege (to display, start, and stop the virtual accelerator)
- SELECT privilege on the tables used in the test queries
- SELECT, INSERT, UPDATE and DELETE privileges on explain tables, including DSN_QUERYINFO_TABLE

About this task

Virtual accelerators do not require accelerator hardware. They produce DB2 EXPLAIN information, which can be visualized in an access plan graph.

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 Virtual Accelerator**.
4. In the Add Virtual Accelerator window, in the **Name** field, type a name for the virtual accelerator.
5. Click **OK**.

Results

The virtual accelerator is shown in the Object List Editor.

What to do next

Complete the following steps in the same way as for a regular accelerator:

1. Define tables on the virtual accelerator.
2. Enable acceleration for these tables as necessary.
3. Submit queries to the virtual accelerator.

Note: You cannot load data into a virtual accelerator.

Related tasks:

"Displaying an access plan graph" on page 87
An access plan graph is 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.

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. On 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 Profilewindow 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       9.116.85.193 1277
TF12            TF12      10.101.6.2   1400
TF3             TF3       10.101.6.6   1400
```

Defining tables on an accelerator

Using the Add Tables to Accelerator wizard, you can define database 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.

About this task

Important:

- Table columns of the type CHAR and VARCHAR can be encoded in UNICODE, EBCDIC, or ASCII. By default, tables cannot co-exist on the same accelerator if columns of these types are encoded differently.

  However, setting the AQT_ENABLE_MULTIPLE_ENCODING environment variable, you can define tables with EBCIDC and UNICODE columns on the same accelerator. For details and restrictions, see the documentation of this environment variable in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*.
- IBM DB2 Analytics Accelerator for z/OS does not support 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.

   The **Unsupported Columns** column lists the unsupported columns in each 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 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 tasks:

"Preparing tables for workload balancing and accelerating static SQL queries" on page 72
You need not enable workload balancing or static SQL support. However, the development of these features made it necessary to introduce a new naming scheme for tables. So if you have tables that were defined with older versions of IBM DB2 Analytics Accelerator for z/OS and want to use workload balancing or accelerate static SQL queries, you must remove and then redefine those tables.

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*.

About this task

To compare response times, you can temporarily disable an accelerator so that all queries are processed by DB2 itself. You might also want to disable an accelerator for maintenance tasks. If you only want to disable query processing for specific tables rather than the whole accelerator, you can instead disable query acceleration for the tables.

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. It is not necessary to disable an entire accelerator for maintaining query 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. It is not necessary to disable an entire accelerator for maintaining query 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.

Preparing tables for workload balancing and accelerating static SQL queries

You need not enable workload balancing or static SQL support. However, the development of these features made it necessary to introduce a new naming scheme for tables. So if you have tables that were defined with older versions of IBM DB2 Analytics Accelerator for z/OS and want to use workload balancing or accelerate static SQL queries, you must remove and then redefine those tables.

Before you begin

Version 4 PTF-2 of the product introduces the SYSPROC.ACCEL_MIGRATE stored procedure, which converts existing table names in accordance with the new naming scheme. Running the stored procedure is less effort than removing, redefining, reloading, and re-enabling the tables, so consider upgrading the product to version 4 PTF-2 or later.

About this task

To support workload balancing and static SQL, it became necessary to change the naming scheme for tables in the SYSACCEL.SYSACCELERATEDTABLES catalog table. Older entries in this catalog table do not provide the additional information that is required for a successful use of these features. Migrate the table names by using the appropriate method for your version of the product:

Procedure

- If version 4 PTF-2 or later has been installed, run the SYSPROC.ACCEL_MIGRATE stored procedure against each accelerator that is supposed to participate in your workload balancing or static SQL scenario. The stored procedure always migrates the names of all tables on an accelerator.

  Important:

  - If you want to accelerate static SQL queries in a workload balancing environment, run the stored procedure before you bind the static packages. Otherwise, you will have to rebind the packages after the table name migration.
  - The query history (**Show Plan** function) in IBM DB2 Analytics Accelerator Studio might show unexpected table names after the migration. This is normal.

  For instructions on the use of the stored procedure, follow the **Related reference** link at the end of this topic.
- If you still use version 4 PTF-1, complete the following steps:
  1. To find the tables that do not yet use the new naming scheme, submit the following query:

```
SELECT ACCELERATORNAME, CREATOR, NAME, SUPPORTLEVEL
FROM SYSACCEL.SYSACCELERATEDTABLES
WHERE SUPPORTLEVEL < 3;
```

2. Remove the tables that are listed in the query output from all accelerators.
3. Redefine these tables on as many accelerators as required.
4. Load the redefined tables.
5. Enable query acceleration for the redefined tables.

Related concepts:

"Static SQL query support" on page 11
Static SQL query support means that you can now accelerate SQL queries that are embedded within DB2 for z/OS application programs. Older versions of IBM DB2 Analytics Accelerator for z/OS could not accelerate such queries.

"Workload balancing" on page 15
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 multiple accelerators, IBM DB2 Analytics Accelerator for z/OS automatically picks the accelerator with the lowest utilization for processing.

Related information:

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

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 tables on an accelerator 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 tables that you want to load must have been added to the accelerator.

About this task

Loading tables can take a long time (for example minutes or hours) depending on the amount of data in the original DB2 tables.

The disk throughput can be quite 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 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 zEnterprise® server.
- Workload on the accelerators.

Restrictions:

- A load fails if names of table columns start with a special character, that is, column names must start with a character from 0-9, a-z, or A-Z.
- IBM DB2 Analytics Accelerator for z/OS does not support 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.
- Incremental updates are not restarted automatically if you enable these for just one table and complete the enablement steps in this order:
  1. Add the table.
  2. Enable incremental updates for the table.
  3. Load the table.
  4. Start incremental updates (**Replication** > **Start**).
  5. Reload the 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.

You can also integrate the loading and updating of accelerator 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 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 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 a table or partition when necessary.

   Important:

   - You can only select table partitions if the tables are range-partitioned tables or tables partitioned by growth, and if the tables 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 table since the last load.

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

   Yes - changed data
   A load or reload of the 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 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 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 table is selected for loading. A possible reason is that the partition is *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 tables and partitions on the accelerator.

All recommended
Displays only the tables and partitions for which a load or reload is recommended.

No - data unchanged
Displays only the tables and partitions that need not be loaded or reloaded.

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

Yes - enforced
Displays only the 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 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 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 all tables to be loaded against changes during the load operation.

   `Current Table`

   Protects just the table that is currently being loaded.

   `Current Partitions`

   Protects the table-space partition containing that part of the table that is currently being loaded. An unpartitioned table is always locked completely.
7. The option **After the load, enable acceleration for disabled tables** ensures that you can run accelerated queries against all tables on the accelerator 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 continuously" on page 106
The incremental update function of IBM DB2 Analytics Accelerator for z/OS allows you to update accelerator tables continuously. Changes to the data in original DB2 for z/OS tables are thus propagated to the corresponding accelerator 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:

"Preparing tables for workload balancing and accelerating static SQL queries" on page 72
You need not enable workload balancing or static SQL support. However, the development of these features made it necessary to introduce a new naming scheme for tables. So if you have tables that were defined with older versions of IBM DB2 Analytics Accelerator for z/OS and want to use workload balancing or accelerate static SQL queries, you must remove and then redefine those tables.

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 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 added to the accelerator.

About this task

After its initial deployment, a table is in *load pending* state and disabled. The *load pending* state is also reached when an error has led to an operation failure. 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 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.

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 a table has been enabled, the corresponding entry in the ENABLED column of the SYSACCEL.SYSACCELERATEDTABLES is set to ON. Although a 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 a table is in. This is useful if you need to recover from an error situation in which an accelerator is down or unreachable.
- An ALTER TABLE operation on a DB2 table prevents query routing to an accelerator if the altered table is referenced in the query (reason code 14). Therefore, you must complete the following steps after an ALTER TABLE operation:
  1. Remove the table from the accelerator.
  2. Re-add the table to the accelerator.
  3. Load the table.
  4. Re-enable query acceleration for the table.

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" on page 74
Successful queries against tables on an accelerator 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.

"Preparing tables for workload balancing and accelerating static SQL queries" on page 72
You need not enable workload balancing or static SQL support. However, the development of these features made it necessary to introduce a new naming scheme for tables. So if you have tables that were defined with older versions of IBM DB2 Analytics Accelerator for z/OS and want to use workload balancing or accelerate static SQL queries, you must remove and then redefine those tables.

"Disabling an accelerator" on page 71
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. It is not necessary to disable an entire accelerator for maintaining query tables; each table can be disabled individually.

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

- 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 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 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). The recommended minimum table size (compressed size on the IBM PureData™ System for Analytics N1001) depends on the number of worker nodes:

Table 8. Recommended minimum table sizes for organizing keys

IBM PureData System for Analytics	Minimum size
N1001-002	0.4 GB
N1001-005	0.75 GB
N1001-010	1.5 GB

This task of altering or changing distribution keys or organizing 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 or organizing 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 or as organizing keys.
4. Click **Alter Keys** on the toolbar.
5. Specify a distribution key or organizing keys:
   - 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.

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 greyed 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.

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 greyed 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:

“Choosing a distribution key” on page 2
By default, all tables rows are evenly distributed among the existing worker nodes (random distribution). When a query includes table joins, it can be that a join can only be executed if table rows are sent from one worker node to another (redistribution), or that all rows are first sent to the Netezza host to be distributed to the worker nodes from there (broadcast). Both, redistribution and broadcast, can be very time-consuming when big tables with millions or even billions of records are involved, to the extent that the accelerator becomes ineffective.

“Choosing an organizing key” on page 3
Organizing keys can speed up accelerated queries even further as they diminish

the time that is needed to scan the disks belonging to a single worker node.

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. Such queries are accepted by regular and virtual accelerators alike. 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.

Note: Unlike regular accelerators, virtual accelerators can process EXPLAIN queries only. That is, the SQL queries sent to a virtual accelerator must contain an `EXPLAIN` statement. Otherwise the query is rejected.

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
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 9.

Table 9. 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.
`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 10 on page 85.
`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.

For basic EXPLAIN information, you need just the PLAN_TABLE. To create it in DB2 for z/OS, modify the sample CREATE TABLE statements in the DSNTESC member of the SDSNSAMP library accordingly. To create

DSN_QUERYINFO_TABLE, use the SQL sample code provided with this information unit. To access this code, follow the appropriate link at the end of this topic.

Table 10 lists and describes the various reason codes. A reason code greater than 0 indicates that a query cannot be accelerated.

Table 10. Possible reason codes in DSN_QUERYINFO_TABLE

REASON_CODE	Description	Suggested solution
0	The query qualifies for routing to an accelerator.	N/A
1	No active accelerator was found when the statement was executed. If an accelerator is available, make sure it is started.	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.
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.
9	The query contains a nested table expression.	Ensure that the FROM clause of the query does not contain a subquery.
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.

Table 10. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
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 from another table in the query. • The table is not enabled for query acceleration.	Ensure that all tables that are referenced by the query are 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 table.	Reload the altered table on the accelerator.
15	The query uses functionality that is available only in DB2 10 for z/OS new-function mode or later.	Ensure that the query does not use any functionality that is available in DB2 10 for z/OS new-function mode or later.
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.
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.
21	The query contains an unsupported correlated subquery.	If possible, rewrite the query without using the unsupported subquery.

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, even for virtual accelerators. 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.

Related tasks:

"Displaying an access plan graph"
An access plan graph is 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.

Related reference:

"Sample SQL for EXPLAIN tables," on page 149
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. If you do not want to run the z/OS jobs, you can also create the tables by submitting the SQL sample code provided here.

Displaying an access plan graph

An access plan graph is 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.

About this task

An access plan graph can give you valuable information. Looking at an access plan graph of a query, you can verify whether your query really uses a specific table on the accelerator. For example, you can define tables on an accelerator and then revisit the access plan graph of the query to check if the tables have been used by the query. In addition, an access plan graph can tell you which tables are referenced indirectly by the query if the query references a view. This helps you identify all tables that you need to load before you can accelerate the query.

Access plan graphs are created on the basis of information in DB2 EXPLAIN tables. To obtain the EXPLAIN data (query result) and convert it to the XML format for output, IBM DB2 Analytics Accelerator Studio invokes the **SYSPROC.ACCEL_GET_QUERY_EXPLAIN** 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, right-click the icon of the DB2 subsystem that contains the tables for the query.
2. Select **New SQL Script** from the menu.
3. The Script.sql window that opens on the upper right contains a blank pane. In this pane, type or paste the query that you want to display an access plan graph for.
4. Click the icon.
5. On the General Settings page of the Collect Explain Data wizard, click **Next**.
6. From the **QUERY ACCELERATION** drop-down list, select **ENABLE**, **ALL**, or **ELIGIBLE**.
7. If you use the High-Performance Storage Saver (HPSS), also select a suitable value from the **GET_ACCEL_ARCHIVE** drop-down list. In selecting the value **Yes**, you include data that has been moved by the HPSS in the query. In selecting **No**, you exclude this type of data.
8. Click **Finish**.

Results

You see the access plan graph at the bottom of the window.

Related concepts:

"EXPLAIN information" on page 84
After creating the necessary EXPLAIN tables, you can analyze queries by invoking the DB2 EXPLAIN function. Such queries are accepted by regular and virtual accelerators alike. 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.

Related reference:

"Types of access plan graphs and nodes in these graphs" on page 89
Different graphs are displayed for queries that can be accelerated and queries that cannot be accelerated. Read a brief article on how these graphs differ, and which set of nodes you can expect in either of these.

Related information:

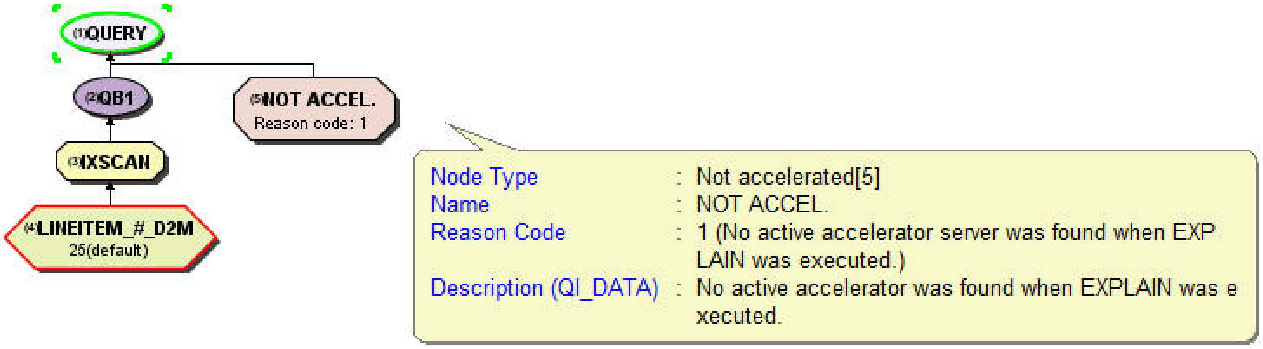
IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

Types of access plan graphs and nodes in these graphs

Different graphs are displayed for queries that can be accelerated and queries that cannot be accelerated. Read a brief article on how these graphs differ, and which set of nodes you can expect in either of these.

If a query cannot be accelerated and is processed by DB2 for z/OS, you see the regular DB2 for z/OS access plan. Unless the query is an EXPLAIN-ONLY query (queries sent to virtual accelerators invariably belong to this category), you see an additional node labeled **NOT ACCEL.**. This node contains the reason code, which explains why the query could not be accelerated.

Example



Node Type : Not accelerated[5]
Name : NOT ACCEL.
Reason Code : 1 (No active accelerator server was found when EXPLAIN was executed.)
Description (QI_DATA) : No active accelerator was found when EXPLAIN was executed.

For an overview of the nodes that might occur in a DB2 for z/OS access plan graph, follow the **Related information** link at the end of this topic.

If a query can be accelerated, you see the access plan as it will be executed on the accelerator. The nodes that can occur in this type of graph are briefly described in Table 11. The nodes reflect the different stages during query processing. Some of these give you valuable information for query optimization.

Table 11. Nodes in access plan graphs

Node	Additional information	Description
ACCELERATED		Indicates that a query could be accelerated. This node is not shown in graphs for EXPLAIN only queries, but only in graphs for real queries. That is, the query must have been routed to an accelerator by setting the SET CURRENT QUERY ACCELERATION special register to one of the following values: • ENABLE • ENABLE WITH FAILBACK • ELIGIBLE • ALL
AGGR	**First number** Number of affected rows **Second number** Total size of the affected rows	Aggregation. Indicates that data will be aggregated to calculate the results of statements, such as SUM, AVG, MAX, or COUNT. The numbers indicate the number of aggregated rows and the size of these rows.

Table 11. Nodes in access plan graphs (continued)

Node	Additional information	Description
BROADCAST	**First number** Number of broadcast table rows **Second number** Total size of the broadcast rows	Broadcast of table rows. This means that all rows relevant for a join must be sent over the network to the active coordinator node (Netezza host) to be redestributed from there. When large tables with more than 100 million rows are involved, broadcasts have a highly negative impact on the performance. Avoid broadcasts by using distribution keys.
GRPBY	**First number** Number of grouped rows **Second number** Total size of the grouped rows	Group by. Indicates that resulting rows are grouped according to a GROUP BY clause in the query.
HSJOIN	**First number** Number of surviving rows after the join **Second number** Total size of the surviving rows	Hash join. A way of executing a table join that employs hashing. It is the most efficient way of executing equality joins. When a large amount of data needs to be processed, hashing achieves its optimum performance when it is combined with a collocated join (in most cases, this means use of a distribution key). The row and size numbers refer to the surviving rows after the join.
NOT ACCEL.	The node contains the reason code for the failure.	Indicates that a query cannot be accelerated. This node is not shown in graphs for EXPLAIN only queries, but only in graphs for real queries. That is, the query must have been routed to an accelerator by setting the SET CURRENT QUERY ACCELERATION special register to one of the following values: • ENABLE • ENABLE WITH FAILBACK • ELIGIBLE • ALL
NLJOIN	**First number** Number of surviving rows after the join **Second number** Total size of the surviving rows	Nested loop join (called expression-join in Netezza terminology). A way of executing joins that is chosen for non-equality joins, that is, joins such as WHERE (T1.COL1 - T2.COL*x* > 0). It is also used for queries that use EXISTS clauses. In a nested loop join, each row in the first table is matched against each row in the second, and the join predicate is evaluated for each of the pairs. It is thus a slow way of executing a join. If the performance is poor, see if you can change the non-equality join into an equality join.

Table 11. Nodes in access plan graphs (continued)

Node	Additional information	Description
QUERY	**First number** Number of rows that are returned to DB2 **Second number** Total size of the rows that are returned to DB2	Indicates the point at which the active coordinator node (Netezza host) returns the query results to DB2 for z/OS. The numbers indicate the number of rows in the result set and the size of these rows.
REDIST	The column name indicates the join column that causes the redistribution.	Redistribution of table rows. That is, to execute a join, table rows must be sent over the network from one worker node to another. When large tables with more than 100 million rows are involved, redistributions have a negative impact on the performance and should be avoided. Consider the use of a distribution key to achieve a collocated join.
RETURN	**First number** Number of returned rows **Second number** Total size of the returned rows	Return of results from the worker nodes to the active coordinator node (Netezza host).
SORT	**First number** Number of sorted rows **Second number** Total size of the sorted rows	Sorting of rows. The numbers indicate the time needed to do the sorting, the number of rows sorted, and the size of these rows.
SUBSELECT		Subselect. That is, the query contains at least one outer query and one inner query, in which the inner query refers to a subset of the table rows captured by the outer query. The SUBSELECT node reflects the evaluation of the nested SELECT statements.
<TABLENAME>	**First number** Number of table rows **Second number** Total size of the table rows **Important:** A *<TABLENAME>* node always shows the current number of table rows and the current size of the rows. These are not necessarily the figures that were valid at the time the query was run.	The name of a base table used in the query.

Table 11. Nodes in access plan graphs (continued)

Node	Additional information	Description
TBSCAN	**First number** Number of "surviving" rows, that is, rows that have not been filtered out due to unused columns, restrictive clauses, or organizing keys. **Second number** Total size of the surviving rows	Table scan. This node represents the scan of a table during query processing.
UNIONA	**First number** Number of rows in the unified result set **Second number** Total size of the rows in the unified result set	Union. Indicates a union of two result sets. This node occurs if the query contains a UNION ALL or UNION DISTINCT statement.
UNIQUE	**First number** Number of surviving rows after the elimination **Second number** Total size of the surviving rows	Elimination of duplicate table rows.

Related information:

Reason codes

http://pic.dhe.ibm.com/infocenter/dstudio/v4r1/topic/com.ibm.datatools.visualexplain.data.doc/topics/znodes.html

Selecting and ordering query monitoring columns

You can select the columns that are displayed in the **Query Monitoring** section of the Accelerators 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). The following information columns are displayed by default:

   SQL Text

   The SQL statements of recently submitted queries.

User ID
The names of users who submitted past queries.

Start Time
Times when queries were received by IBM DB2 Analytics Accelerator for z/OS

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 Netezza system for processing.

RUNNING
The query is being processed by the Netezza system.

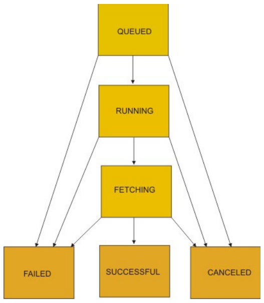
FETCHING
The Netezza system has finished processing the query, but DB2 for z/OS has not yet finished collecting the results.

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:



Wait Time
The times that queries had to “spend” in the queue before they were processed by the IBM PureData System for Analytics.

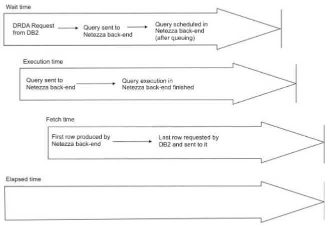
Execution time
The time that a query spent in the queue before it was sent to the accelerator plus the time needed to process the query on the IBM PureData System for Analytics.

Fetch Time
The time interval starting with the first retrieval of a result row by DB2 and ending with the last. In most cases, Netezza query execution has already finished when the result retrieval process is still ongoing.

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 Netezza 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 Netezza, 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.


Wait time
DRDA Request from DB2
Query sent to Netezza back-end
Query scheduled in Netezza back-end (after queuing)
Execution time
Query sent to Netezza back-end
Query execution in Netezza back-end finished
Fetch time
First row produced by Netezza back-end
Last row requested by DB2 and sent to it
Elapsed time

Result Size

The data size of the rows that were returned in query results (in MB or GB).

Rows Returned

The number of rows that were returned by the accelerator as query results.

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. 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

     Re-Run Query
     To rerun the query

     Cancel
     To cancel a query rerun.
   - 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.

Related information:

"Query history is not displayed - error message returned" on page 146
The query history is not displayed in the Accelerator view and a corresponding error message is returned.

Canceling tasks

See how to cancel tasks that are in progress on an accelerator, such as queries or load operations.

Procedure

1. Open the accelerator view of the accelerator that the task is running on.
2. Click **Cancel Tasks** on the toolbar.
3. The table in the Cancel 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 has 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 have been 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. Consequently, a value of 0.25 means 25 percent complete.

   Progress Message
   Additional information about running tasks.

   User The IDs of the users who have started the tasks.
4. Click **Close** to close the Cancel Tasks window.

Removing tables from an accelerator

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

About this task

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. If incremental updates are enabled for the tables that you want to remove, first disable incremental updates:
   a. In the list of tables in the Accelerator view, select the tables that you want to disable.
   b. Click **Replication** > **Disable Replication**.
4. Select the tables that you want to remove.
5. Click **Remove** on the toolbar.

Related tasks:

"Including or excluding tables from an incremental update process" on page 109
Incremental updates can be enabled or disabled on the table level. This allows you to 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 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 where an administrator 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. An appropriate profile connects you to the right DB2 subsystem, that is, the subsystem that works with the accelerator containing the tables to be exported. 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 tables for query acceleration that was previously saved to an XML file.

About this task

A typical use case is where an administrator 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. An appropriate profile connects you to the right DB2 subsystem, that is, the subsystem that works with the accelerator on which you want to define the 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 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 for z/OS 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 for z/OS 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 for z/OS 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 101.

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 for z/OS or a custom profile. For importing a custom profile, see step 6.
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 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 for z/OS 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 trace information
   All the information that is captured as a result of your settings in the Configure Trace window. This comprises the following chunks of information, which you can include or exclude here:

   Traces (includes queries)
   An incident protocol that lists the various events and actions that were taken.

   Core dumps (may include user data)
   An image of the system memory portion that is occupied or shared by IBM DB2 Analytics Accelerator for z/OS processes at a given time.

   Important: A core dump contains recent query data among which there might be sensitive information about customers, money transactions and so on. Be careful not to divulge this information to unauthorized parties or individuals.

   Netezza core dumps (may include user data)
   Netezza core files (files in the `/nz/kit/log/spucores` directory and files detected by the **find /nz/kit/log -type f -name '.[0-9]*'** command.

   Handle this information with the same care as the IBM DB2 Analytics Accelerator core dumps. See the previous Important note.

   Catalog dumps (includes database schema)
   Snapshot of the IBM DB2 Analytics Accelerator for z/OS catalog.

   Important: A catalog dump always contains your database schema. Since database schemas are often protected assets, be careful not to divulge this information to unauthorized parties or individuals.

   Plans (includes queries)
   Netezza execution plans.

   Log history
   Log information created by the operating system of an accelerator.

   Detailed Netezza diagnostics (includes user data)
   See the previous Important note.

   Note: When you open the Save Trace window from the context menu of the Accelerators folder, the **Accelerator trace information** check box and the indented check boxes underneath are 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 prevents 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.

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. 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 information:

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

Configuring the FTP server for the Save Trace function

To add trace information to an existing IBM problem management record (PMR), your IBM DB2 Analytics Accelerator Studio client must connect to the IBM RETAIN server on which the PMR was created.

About this task

The RETAIN server uses the file transfer protocol (FTP) for data transmissions. The IBM DB2 Analytics Accelerator section in the Preferences window of IBM DB2 Analytics Accelerator Studio allows you to specify the correct server and connection settings for this operation.

Procedure

1. From the main menu in IBM DB2 Analytics Accelerator Studio, select **Window** > **Preferences** > **IBM DB2 Analytics Accelerator** > **Save Trace Dialog**.
2. In the **Server Name** field, type the name of the IBM RETAIN server. European IBM customers, for example, use a server called `ftp.ecurep.ibm.com`.
3. In most cases, port 21 is used for FTP transmissions so that you need not change the default entry in the **Port** field.
4. In the **User ID** field, type `anonymous`.
5. In the **Password** field, type your e-mail address. For European customers, the logon process is described on the web. See the links under **Related information** at the end.
6. In the **Directory** field, type the path to the directory that you use to dispatch RETAIN data.
7. If your FTP transmission needs to pass a proxy server that is part of your organization's firewall, select **Use SOCKS 5 proxy server** and specify the details of this proxy server:

   Note: As the label of the check box suggests, the proxy server needs to comply with the *SOCKS 5* standard.

   a. In the **Address** field, type the host name or IP address of the proxy server.
   b. In the **Port** field, type the number of the port on which the proxy server listens to outbound requests.
   c. In the **User ID** field, type a user ID that allows you to send data via the proxy server.
   d. In **Password** field, type the password belonging to that ID.
8. Click **Apply** to save your settings or **OK** to save your settings and exit the Preferences window.

Related information:

⇨ IBM Enhanced Customer Data Repository Service

⇨ Service requests and PMRs

Clearing trace information and replication events

The Clear function deletes the collected trace information and the replication events (information about incremental updates) from the replication event log.

About this task

The deletion does not include trace information that was saved to a file. Delete trace files manually.

It is not possible to clear the trace information without clearing the replication event log at the same time or vice versa. Postpone the deletion if you still need either information.

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 **Clear**.
4. In the Confirm Clearing Trace window, click **Yes**. A window with the message `Existing trace were successfully removed` is displayed.
5. Click **OK**.

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 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 PureData System for Analytics. 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 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 continuously

The incremental update function of IBM DB2 Analytics Accelerator for z/OS allows you to update accelerator tables continuously. Changes to the data in original DB2 for z/OS tables are thus propagated to the corresponding accelerator 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 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, 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 12 compares the advantages and disadvantages.

Table 12. 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	Continuous 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 projection of this table on multiple accelerators. If one accelerator fails, another one can take over to process incoming queries. This happens with just a short delay.

Related tasks:

"Loading tables" on page 74
Successful queries against tables on an accelerator 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.

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.

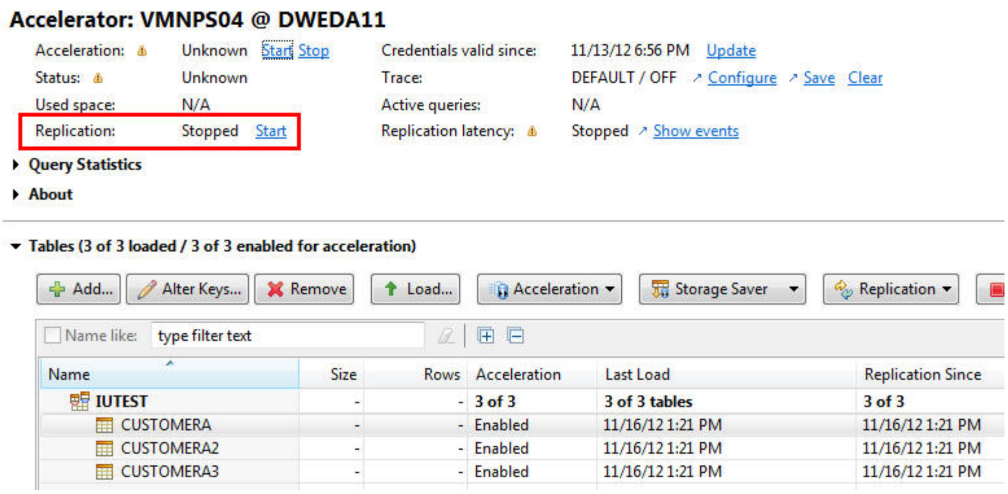
Configured
: The incremental update function is available for this accelerator, but tables have not yet been added to the process.

Error
: The incremental update function has been configured, but cannot be started due to an error.

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

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

See Figure 9.



Accelerator: VMNPS04 @ DWEDA11

Acceleration: Unknown Start Stop
Status: Unknown
Used space: N/A
Replication: Stopped Start
Credentials valid since: 11/13/12 6:56 PM Update
Trace: DEFAULT / OFF Configure Save Clear
Active queries: N/A
Replication latency: Stopped Show events

Tables (3 of 3 loaded / 3 of 3 enabled for acceleration)

□ Name like: type filter text

Name	Size	Rows	Acceleration	Last Load	Replication Since
IUTEST	-	-	3 of 3	3 of 3 tables	3 of 3
CUSTOMERA	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM
CUSTOMERA2	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM
CUSTOMERA3	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM

Figure 9. **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 an incremental update process

Incremental updates can be enabled or disabled on the table level. This allows you to 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

A 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 because this guarantees that data changes will not go amiss during the time that the table data is unloaded from DB2 to the accelerator and the point where the incremental update process sets in.

Attention:

- A risk of data loss exists if you include tables without a unique constraint, such as a primary key or a unique index. To identify rows in such a table, the values of all table columns are used. It is possible that rows with exactly the same values exist, in which case it is impossible to identify a row unambiguously. Hence a deletion of a single row from a table in DB2 for z/OS results in the deletion of this row and all its duplicates from the corresponding accelerator table. This, in turn, leads to incorrect results for accelerated queries.
- 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 tables.
- With the introduction of IBM DB2 Analytics Accelerator for z/OS Version 3.1, the DB2 attribute DATA CAPTURE is added to the tables on an accelerator. 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.
- In general, an ALTER TABLE operation on a DB2 table prevents query routing to an accelerator if the altered table is referenced in the query (reason code 14). This is independent of any incremental update configuration that might exist. Therefore, you must complete the following steps after an ALTER TABLE operation:
  1. Remove the table from all accelerators on which it is defined.
  2. Redefine the table on the accelerators from which you previously removed it.
  3. Enable incremental updates for the redefined table on all accelerators.
  4. Load the redefined table on all accelerators.
  5. Enable query acceleration for the redefined table on all accelerators.

  Special attention must be paid to tables that have been altered at any time before enabling incremental updates for these tables. You must ensure that such tables have been reorganized after the last alteration, and before the incremental update on them is enabled. If the reorganization did not happen, you must reorganize such tables before enabling incremental updates for them.

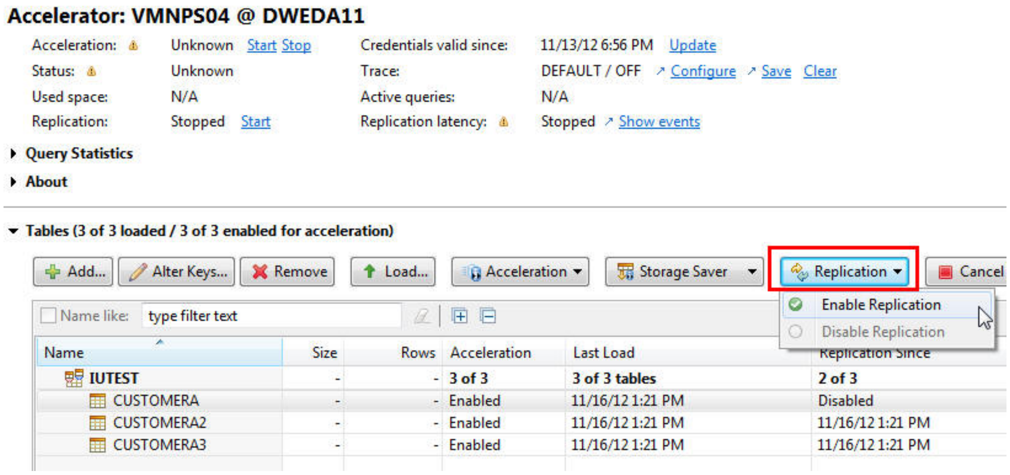
  Exception: You need not follow the steps previously described after an ALTER TABLE ROTATE PARTITION operation. The CDC capture agent automatically detects a rotation in the DB2 transaction log. If a partition has been rotated, the agent generates a DELETE DML statement for the records in the to-be-deleted partition (rolled-off partition). It will also propagate new records that were inserted into the rolled-on partition. So there is no need for a table removal, redefinition, re-enablement, or reload.

About this task

Two different situations must be considered before adding tables to an incremental update process:

- The 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 10.
- The 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 table on the accelerator 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.



Accelerator: VMNPS04 @ DWEDA11

Acceleration: Unknown Start Stop
Status: Unknown
Used space: N/A
Replication: Stopped Start

Credentials valid since: 11/13/12 6:56 PM Update
Trace: DEFAULT / OFF Configure Save Clear
Active queries: N/A
Replication latency: Stopped Show events

Query Statistics
About

Tables (3 of 3 loaded / 3 of 3 enabled for acceleration)

Name	Size	Rows	Acceleration	Last Load	Replication Since
IUTEST	-	-	3 of 3	3 of 3 tables	2 of 3
CUSTOMERA	-	-	Enabled	11/16/12 1:21 PM	Disabled
CUSTOMERA2	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM
CUSTOMERA3	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM

Figure 10. Timestamp columns and **Replication** *button in the accelerator view*

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 tables to include or exclude.
2. Include or exclude tables from the process:
   - To include selected tables in the incremental update process, click **Replication** > **Enable Replication**. See Figure 10 for the location of the button.
   - To exclude selected tables from the incremental update process, click **Replication** > **Disable Replication**.

3. If you have disabled the selected tables, click **Start** in the header of the accelerator view to restart the incremental update process for the remaining enabled tables. Also note that incremental updates are not restarted automatically if you enable just one table and the steps involved are completed in this order:
   a. Add the table.
   b. Enable incremental updates for the table.
   c. Load the table.
   d. Start incremental updates (**Replication** > **Start**).
   e. Reload the 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.

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 tables.

Medium
Recent changes to the original DB2 tables will be propagated to the corresponding accelerator tables within three and six minutes.

High It will take six minutes or longer to apply recent changes to DB2 tables to the corresponding tables on the accelerator.

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 tables that have been included in the incremental update process.

Deletes
Number of row deletions from accelerator 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 translates to 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.

Source Agent
Number of data changes to relevant DB2 for z/OS tables (tables that have been added to the accelerator and included in the incremental update process.

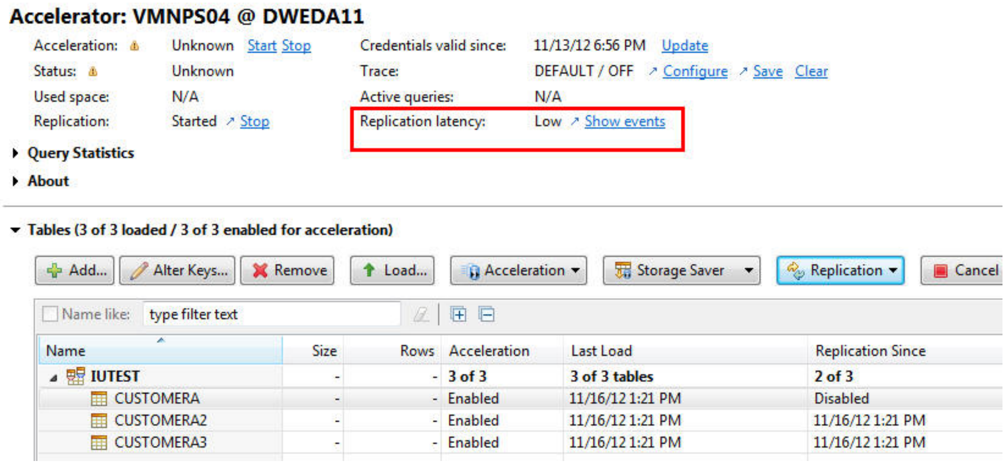
Inserts
Number of row inserts in the relevant DB2 for z/OS tables.

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.

See Figure 11.



Accelerator: VMNPS04 @ DWEDA11

Acceleration:	Unknown Start Stop	Credentials valid since:	11/13/12 6:56 PM Update
Status:	Unknown	Trace:	DEFAULT / OFF Configure Save Clear
Used space:	N/A	Active queries:	N/A
Replication:	Started Stop	Replication latency:	Low Show events

Name	Size	Rows	Acceleration	Last Load	Replication Since
IUTEST	-	-	3 of 3	3 of 3 tables	2 of 3
CUSTOMERA	-	-	Enabled	11/16/12 1:21 PM	Disabled
CUSTOMERA2	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM
CUSTOMERA3	-	-	Enabled	11/16/12 1:21 PM	11/16/12 1:21 PM

Figure 11. **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:
  1. 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.
  2. 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.
  1. 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.
  2. 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.

Related tasks:

"Clearing trace information and replication events" on page 104
The Clear function deletes the collected trace information and the replication events (information about incremental updates) from the replication event log.

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 a 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 a 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 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 might result in a poor performance when UPDATE or DELETE changes are propagated to an accelerator table (because it takes longer to identify the changed table rows).

Attention: A risk of data loss also exists because the absence of a unique constraint might cause the deletion of multiple rows from an accelerator tables if just a single row is deleted from a DB2 table. That is, all rows with identical values will be deleted from the accelerator table because the various instances cannot be uniquely identified. Incorrect query results will be the consequence. Therefore, do not include such tables in incremental update processes unless you know for sure that the tables do not contain duplicate rows.

OrganizingKeysAndUniqueConstraintsMismatchWarning

The unique constraints of a table in DB2 for z/OS differ from those of the corresponding accelerator 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.

Freeing up storage in DB2 for z/OS

The high-performance storage saver (HPSS) moves 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 moved to 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 moved 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 moved 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 moved 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 moved partitions. Partitions that have been moved 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 move the data of one or more partitions to 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 (moved) 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 moved.

The table space of a moved partition continues to exist after the move. 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 move. 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 move 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 move partitions of tables that are in *InitialLoadPending* or *Loaded* state. This implies that partitions of tables cannot be moved if incremental updates have been enabled for the table.
- Table data cannot be moved 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 move 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.
- No failback mechanism is in place in case an error occurs. Records that have already been moved 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 moved. 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 moved. Although the original data is deleted after the move, the data in unsupported columns is not lost because it is saved to the image copies.

  However, the following other data types lead to a failure of the move 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 moved to different accelerators, the query fails.
- For a move 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 a data move.

Moving 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 move 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 variable determine how many local image copies and recovery image copies are created for each moved partition and how the data sets are named. If you set a variable, a corresponding data set is created when partition data is moved. 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 13 on page 119.

Table 13. 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 117, 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 move 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 117 must be identical on all members of the data sharing group.
6. Nonpartitioning indexes might slow down the move process. If more than just a few small to mid-sized nonpartitioning indexes are defined for a table, but the partitions to be moved are not many and rather large, then consider dropping the indexes before the move and recreating these thereafter.

7. If you also use the incremental update function, check the setting of the ONUTILITYACTION keyword in the IBM InfoSphere Change Data Capture 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 moved by the HPSS. Hence the parts that have not been moved will not be synchronized anymore. To avoid this conflict, set the keyword to the value IGNORE.

About this task

The task of moving partition or table data to an accelerator 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 move 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 move 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 move partition or table data to.

   Note: You can only move 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 on the accelerator, click **Add** to start the Add Tables wizard and define these tables.
3. If necessary, refresh the Accelerator view.
4. To move the data of a range-partitioned table or partitions of this table, expand the appropriate schema node and select the table name of that table in **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 moved 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 move, 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 moved are greyed 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 moved data needs to be accessed by their queries. Whether to include or exclude the moved 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 moved 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 move all partitions with data that was not collected in the current year. 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 moved 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 14. 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.
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.

Table 14. 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
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.
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 14 on page 122 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" on page 84
After creating the necessary EXPLAIN tables, you can analyze queries by invoking the DB2 EXPLAIN function. Such queries are accepted by regular and virtual accelerators alike. 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 moved partitions

Restoring moved partition data might become necessary if you must add or update data in the original DB2 partition or if an error in IBM DB2 Analytics Accelerator for z/OS has led to the loss of moved data. Restoring moved data is required also after changing the schema of a DB2 table (for example, adding a column).

About this task

In the context of moved 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 moved partitions to DB2
2. Removing the corresponding table from the accelerator
3. Making changes to the schema, that is, submitting the `ALTER TABLE ADD COLUMN` statement
4. Redefining the table on the accelerator
5. Moving 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.
- A moved 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 invoke 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.

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, follow the appropriate **Related reference** link at the end of this topic.

Procedure

1. In IBM DB2 Analytics Accelerator Studio, open the Accelerator view of the accelerator that you want to restore data from.
2. If necessary, refresh the Accelerator view.
3. On the toolbar of the Accelerator view, select **Storage Saver** > **Restore Partitions to DB2**.
4. 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.
5. Click **OK**.

Related information:

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

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 listed components. However, if you must update more than one component, follow the suggested order. The order for the individual components depends on the product version or the PTF level.

Important: You cannot upgrade components of IBM DB2 Analytics Accelerator for z/OS directly from version 2.1x to version 4.1.0 or to 3.1.0 PTF levels higher than 3.1.0 PTF-1. In all these cases, you must first migrate to 3.1.0 PTF-1.

Table 15. Order in which to update components

Migration from 2.1.*x* to 3.1.0 PTF-1	**Update from 3.1.0 PTF-*x* to 3.1.0 PTF-*y***	**Migration from 3.1.0 PTF-*x* to 4.1.0 PTF-*x***	**Update from 4.1.0 PTF-*x* to 4.1.0 PTF-*y***
1. IBM DB2 Analytics Accelerator for z/OS Version 3.1.0 FMIDs and program temporary fixes (PTFs) via SMP/E 2. Prerequisite PTFs for DB2 10 for z/OS 3. IBM DB2 Analytics Accelerator Studio 4. Netezza firmware (FDT) 5. Netezza Host Platform (HPF) 6. Netezza Performance Server (NPS) 7. IBM DB2 Analytics Accelerator software 8. IBM DB2 Analytics Accelerator stored procedures 9. IBM DB2 Analytics Accelerator Access Server 10. IBM DB2 Analytics Accelerator replication engine 11. IBM InfoSphere Change Data Capture for z/OS Capture Agent	1. Prerequisite program temporary fixes (PTFs) for DB2 10 for z/OS 2. IBM DB2 Analytics Accelerator Studio 3. Netezza Firmware (FDT) 4. Netezza Host Platform (HPF) 5. Netezza Performance Server (NPS) 6. IBM DB2 Analytics Accelerator Access Server 7. IBM DB2 Analytics Accelerator replication engine 8. IBM DB2 Analytics Accelerator software 9. IBM DB2 Analytics Accelerator stored procedures 10. IBM InfoSphere Change Data Capture for z/OS Capture Agent	1. IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 FMIDs and program temporary fixes (PTFs) via SMP/E 2. Prerequisite PTFs for DB2 10 for z/OS or DB2 11 for z/OS 3. IBM DB2 Analytics Accelerator Studio 4. IBM DB2 Analytics Accelerator Access Server 5. IBM DB2 Analytics Accelerator replication engine 6. IBM DB2 Analytics Accelerator software 7. IBM DB2 Analytics Accelerator stored procedures 8. Netezza Firmware (FDT) (requires assistance from IBM support) 9. Netezza Host Platform (HPF) (requires assistance from IBM support) 10. Netezza Performance Server (NPS) 11. IBM InfoSphere Change Data Capture for z/OS Capture Agent	1. IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 FMIDs and program temporary fixes (PTFs) via SMP/E 2. Prerequisite PTFs for DB2 10 for z/OS or DB2 11 for z/OS 3. IBM DB2 Analytics Accelerator Studio 4. IBM DB2 Analytics Accelerator Access Server 5. IBM DB2 Analytics Accelerator replication engine 6. IBM DB2 Analytics Accelerator software 7. IBM DB2 Analytics Accelerator stored procedures 8. Netezza Performance Server (NPS) 9. IBM InfoSphere Change Data Capture for z/OS Capture Agent

Important: The IBM DB2 Analytics Accelerator for z/OS software, the Access Server and the replication engine are updated in the same manner, which is why the update of these components is described in the same topics. The order of topic presentation in this manual therefore deviates from the recommended update installation order. For the update of individual components, strictly follow the order in this topic. Jump between the instruction topics where necessary.

Updating IBM DB2 Analytics Accelerator Studio

The workstation on which IBM DB2 Analytics Accelerator Studio is installed might have an Internet connection or no Internet connection. Refer to the appropriate section for adding or updating IBM DB2 Analytics Accelerator Studio.

About this task

You can use the newest version (4.1.0) of IBM DB2 Analytics Accelerator Studio with older versions of accelerators and stored procedures. Version 2.1.x and 3.1.x of IBM DB2 Analytics Accelerator for z/OS continue to work even if you have installed IBM DB2 Analytics Accelerator Studio Version 4.1.0. However, the user interface does not show the new version 3 or version 4 functions until you update the stored procedures to version 3.1.0 (for version 3 functions) or version 4.1.0 (for version 4 functions).

Using IBM Installation Manager to add or update IBM DB2 Analytics Accelerator Studio from a workstation without an internet connection

Follow the steps in this section if you want to add or update IBM DB2 Analytics Accelerator Studio from a workstation without an internet connection.

Before you begin

Because you cannot update installed products directly by connecting to a repository location on the internet, an administrator must provide relevant update repositories on a computer that you can access (shared network drive or intranet server for example). Administrators can download update repositories for IBM DB2 Analytics Accelerator Studio from the following website:

IBM Support: Fix Central - Select fixes

The administrator requires an IBM ID and password to log on to the download site. It is possible to register for such an ID during the logon. After the necessary credentials have been provided, IBM Installation Manager will connect to the download site.

After the download, the administrator must extract the compressed archive to a location that you can access.

About this task

If IBM Installation Manager is installed on a workstation without an internet connection, it cannot locate the repositories for updates or additional plugins of already installed products automatically. To update products successfully, you must therefore point IBM Installation Manager to the proper repository location.

Procedure

1. Start IBM Installation Manager.
2. From the main menu, select **File** > **Preferences** > **Repositories**.
3. Click **Add Repository**.
4. Click **Browse** and navigate to the directory that contains the extracted update package.
5. Select the `repository.config` file and click **Open**.
6. Click **OK** twice to return the main window of IBM Installation Manager.
7. Click **Install**.
8. Follow the instructions in the installation wizard.

Using IBM Installation Manager to add or update IBM DB2 Analytics Accelerator Studio from a workstation with an internet connection

Follow the steps in this section to add or update an existing version of the product from a workstation with an internet connection.

Procedure

1. Start IBM Installation Manager.
2. Click **Update**.
3. Follow the instructions in the wizard. Deselect components that you do not want to update.

Differences to earlier versions and customization

IBM DB2 Analytics Accelerator Studio 4.1 differs in some respects from its predecessors. You might want to customize your installation so that it uses the workspace directory, default perspective, and language settings of earlier versions.

If you follow the steps in “Using IBM Installation Manager to add or update IBM DB2 Analytics Accelerator Studio from a workstation with an internet connection” or “Using IBM Installation Manager to add or update IBM DB2 Analytics Accelerator Studio from a workstation without an internet connection” on page 126, IBM Data Studio Client 4.1 and IBM DB2 Analytics Accelerator Studio 4.1 are installed.

Earlier versions of IBM DB2 Analytics Accelerator Studio, for example version 3.1, used the InstallAnywhere program for the installation. In contrast, version 4.1 installs IBM Data Studio Client 4.1 by using IBM Installation Manager and adds the IBM DB2 Analytics Accelerator functionality as a third-party extension.

When comparing IBM DB2 Analytics Accelerator Studio 4.1 with version 3.1, you will notice the following differences:

- On Windows, the default folder in the start menu is named `\IBM Data Studio\` instead of `\IBM DB2 Analytics Accelerator Studio\`.
- The name of the default workspace is different. If you want to reuse your existing workspace for IBM DB2 Analytics Accelerator Studio, select the following directory path in the workspace launcher that is displayed when you start IBM Data Studio:

  ```
  C:\Documents and Settings\<username>\IBM\
  DB2 Analytics Accelerator Studio 3.1\workspace
  ```

  where `<username>` is the name of the log-on user. This is the default workspace directory for IBM DB2 Analytics Accelerator Studio Version 3.1. You might have used a custom directory with earlier versions of the program, in which case you must change your selection accordingly.
- When you start the IBM Data Studio Client, the Accelerator perspective is not the default perspective. To activate the Accelerator perspective, select **Window** > **Open Perspective** > **Other** > **Accelerator** from the menu bar.
- IBM DB2 Analytics Accelerator Studio Version 4.1 is available in English only. However, your version of IBM Data Studio might display some interface elements in the language that you specified in the regional settings of your operating system. To avoid a mixture of elements in English and another language, configure the IBM Data Studio version that you use for IBM DB2 Analytics Accelerator for z/OS to display all interface elements in English only. To do that, follow these steps:
  1. Locate the `eclipse.exe` file in the file path of your IBM Data Studio installation.
  2. Start IBM Data Studio by running the following command:

     ```
     eclipse.exe  "-nl en_US"
     ```

  On Windows, you can also complete these steps to the same end:
  1. Right-click the **Data Studio 4.1 Client** shortcut and select **Properties** from the menu.

2. In the Properties Wizard, on the **Shortcut** tab, edit the **Target** field: At the end of the target entry, add a space character followed by `"-nl en_US"`.
3. Click **OK**.

Updating the IBM DB2 Analytics Accelerator software, the Netezza Performance Server (NPS), the Access Server, or the replication engine

These are software components that run on the accelerator hardware, that is, the IBM PureData System for Analytics.

Installing update packages for the accelerator

IBM DB2 Analytics Accelerator for z/OS software includes specific libaries and other code to be installed on the IBM PureData System for Analytics. Updates for the Access Server, the replication engine, and the Netezza Performance Server (NPS) are delivered in separate packages.

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 hierarchical file system (HFS) that is used by IBM UNIX System Services.
- The user ID is authorized to run the SYSPROC.ACCEL_UPDATE_SOFTWARE stored procedure.

About this task

The entire installation or update procedure consists of several steps:

1. Installing the update packages on your z/OS data server.
2. Transferring the update packages to the accelerator.
3. Activating the newly installed software.

This topic covers step 1 only.

Procedure

1. Log on to your z/OS data server using a remote client, such as IBM Personal Communications.
2. Use SMP/E to install the PTF or APAR fix in a directory path called `usr/lpp/aqt/packages` in your hierarchical file system on z/OS.

   Important:

   - The directory path `usr/lpp/aqt/packages` must be present, but it does not have to start in the root directory. You can create this directory path in any other directory, but then you must change the value of the environment variable AQT_INSTALL_PREFIX in the `<HLQSP>.SAQTSAMP(AQTENV)` data set before you transfer the packages to the accelerator in the following step. By default, AQT_INSTALL_PREFIX is set to NULL, which means that it points to a `usr/lpp/aqt/packages` directory path starting from the root directory. So if, for example, you want to install the packages in a directory called `/mnt/software/usr/lpp/aqt/packages`, you must set the AQT_INSTALL_PREFIX environment variable as follows:

     ```
     AQT_INSTALL_PREFIX=/mnt/software
     ```

   - The `<AQT_INSTALL_PREFIX>/usr/lpp/aqt/packages` directory often 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 these packages in the file system. So make sure that the directory can accommodate all the files. The *IBM DB2 Analytics Accelerator for z/OS: Program Directory* lists the exact disk space requirements.

Results

Your local `usr/lpp/aqt/packages` directory now contains the accelerator update package `*version*.tar.z`.

Related information:

"Updating other Netezza software" on page 132
Updates of the Netezza Firmware (FDT) and the Netezza Host Platform (HPF) cannot be installed with SMP/E and are therefore 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.

IBM DB2 Analytics Accelerator for z/OS: Program Directory

Transferring update packages for the accelerator

Transfer update packages for IBM DB2 Analytics Accelerator for z/OS, the Access Server, the replication engine, or the Netezza Performance Server (NPS) by completing the steps in this section.

Before you begin

Make sure that the following conditions apply:

- Suitable accelerator installation packages exist in the `usr/lpp/aqt/packages` directory on your z/OS data server. Update packages for the IBM DB2 Analytics Accelerator for z/OS software are named `*version*.tar.z`, where *version* is the first part of the package name, which indicates the version of the software. Update packages for the Access Server, the replication engine, or the Netezza Performance Server (NPS) are named differently.
- The value of the AQT_INSTALL_PREFIX environment variable determines where IBM DB2 Analytics Accelerator Studio looks for software update packages. If it is not set correctly, the packages cannot be located and thus cannot be transferred or installed.

  Important: During the SMP/E installation or manual copying, the `usr/lpp/aqt/packages` was created on a volume specified by the installer. The target system must be able to access this directory. The path to the `usr/lpp/aqt/packages` directory does not have to start in the root directory. It might have been created in any other directory. In this case, the environment variable AQT_INSTALL_PREFIX must be set accordingly in the `<HLQSP>.SAQTSAMP(AQTENV)` data set, where `<HLQSP>` is the chosen high-level qualifier for stored-procedure libraries. By default, the AQT_INSTALL_PREFIX environment variable is set to NULL, which means that it points to a `usr/lpp/aqt/packages` directory path starting from the root directory. So if, for example, the packages have been copied to a directory `/mnt/software/usr/lpp/aqt/packages`, the AQT_INSTALL_PREFIX environment variable must have been set as follows:

  ```
  AQT_INSTALL_PREFIX=/mnt/software
  ```
- The user ID has read and write access to the `<AQT_INSTALL_PREFIX>/usr/lpp/aqt/packages` directory, where `<AQT_INSTALL_PREFIX>` stands for the value of this environment variable. 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. Furthermore, read access is required for all subdirectories of the `<AQT_INSTALL_PREFIX>/usr/lpp/aqt/packages` directory. 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 4.1.0.
- When you transfer an NPS update package, it is checked whether the NPS version is compatible with the installed Netezza Firmware (FDT) and Netezza Host Platform (HPF) versions. If the NPS is incompatible, you must first update the FDT, the HPF, or both. These tasks, however, can only be completed with the help of IBM support and require a downtime of about one day. So, if the NPS

update is really necessary, make sure that you plan it thoroughly in advance. For more information, follow the **Related information** link at the end of this topic.

Another caveat exists in connection with an NPS transfer: If the NPS package fails the compatibility check, the process ends prematurely without transferring any packages, no matter how many other packages were selected. It is therefore recommended that you first transfer packages for IBM DB2 Analytics Accelerator, the Access Server, or the replication engine, then activate these, and transfer and activate the NPS update in a later step.

About this task

The update packages are copied from the hierarchical file system (HFS) in z/OS to the accelerator, but are not yet activated. However, to actually use a new version, you must activate it. How to do this is described in a later topic. See the hint at the end of this one.

Installing 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.

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 Software Versions window, you can see all software packages that are available in the HFS 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 Software Versions window might be confusing, especially if it also contains older packages. To find the correct package numbers, see the closing information for the latest program temporary fix (PTF). You find the closing information 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 a PTF was shipped with a major product release, then, in general, you must transfer the packages included in the PTF rather than the packages in the base version.
7. Click **Transfer** to complete the installation.

What to do next

To save space in the SMP/E target directory of the HFS, 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_INSTALL_PREFIX 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

Plan and install documentation for DB2 Analytics Accelerator for z/OS

Switching between available versions of accelerator software components

To use a recently transferred software package, you must first activate that package by selecting the version that the package contains. In the case of IBM DB2 Analytics Accelerator for z/OS software, you can also switch to an older version if one of your accelerators has problems with the newest version.

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 **Cancel Tasks** in the relevant Accelerator view of IBM DB2 Analytics Accelerator Studio and cancel all active console tasks from the Cancel Tasks window.
- If you want to activate new versions of the Access Server and the replication engine and two DB2 subsystems are enabled for incremental updates, disable one of these subsystems from the IBM DB2 Analytics Accelerator Console before the activation. Otherwise, you will not be able to enable more than those two subsystems in the future (version 4.1 of the product allows you to enable up to ten DB2 subsystems).

About this task

- The application of an update affects all DB2 subsystems that are connected to an accelerator.
- An update of the accelerator software does not automatically include the components for the incremental update function or the Netezza Performance Server (NPS). You must activate updates of the Access Server, the replication engine, and the Netezza Performance Server (NPS) separately.
- The activation of a different accelerator software version causes a restart of the accelerator.
- You cannot activate earlier versions of the Access Server or the replication engine. Just upgrades are possible here.

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 Table 15 on page 125.

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. On the first page of the Apply Software Version wizard, select the component whose active version you want to change:
   - **Accelerator** (IBM DB2 Analytics Accelerator for z/OS software)
   - **Netezza Performance Server (NPS)** (software for the Netezza hosts)
   - **Access Server** (Access Server component for incremental updates)

   - **Replication engine** (replication engine or Apply Agent for incremental updates)
7. Click **Next**.
8. On the second page of 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.
9. Activate a version by selecting the appropriate radio button in the **Switch To** column.
10. Click **Finish**.

Results

When the process has been completed successfully, a message similar to the following is displayed:

```
New software version was successfully activated.
```

Note: In some cases, migration processes are still ongoing even though the success message was displayed, for example a restart of the Netezza database server. In a situation like this, it might take additional time until the accelerator is 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, remove packages from the accelerator that you no longer need. For instructions, follow the link under **Related tasks**.

Related tasks:

"Removing obsolete software packages from an accelerator" on page 134
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.

Related information:

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

Updating other Netezza software

Updates of the Netezza Firmware (FDT) and the Netezza Host Platform (HPF) cannot be installed with SMP/E and are therefore 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, preferably two weeks before the planned migration date. This gives IBM support enough time to analyze your system, coordinate maintenance actions, and update components as required.
4. If you plan to update the NPS, and the HPF or FDT updates are required for the NPS update, state this in the body of the PMR. Also state the target NPS version.
5. Attach the trace file to the service request.

What to do next

To save space in the SMP/E target directory of the HFS, 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.

Transferring Netezza Firmware (FDT) updates or Netezza Host Platform (HPF) updates

When you have submitted the service request for the update, 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 PureData System for Analytics.

Before you begin

Netezza update packages must exist in the download directory for packages of this type in the hierarchical file system (HFS) of your z/OS data server. Update packages for the Netezza host consist of single files only. The target system must be able to access this directory.

Important: The AQT_HOST_PACKAGE_DIRECTORY environment variable points to this download directory. Unlike AQT_INSTALL_PREFIX, it specifies an absolute path (starting from the root directory). The AQT_HOST_PACKAGE_DIRECTORY environment is also 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 Netezza updates. If it is not set correctly, the packages cannot be located and thus cannot be transferred to the IBM PureData System for Analytics. 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 4.1.

About this task

Netezza update packages are not installed automatically. They are just transferred to a directory on the IBM PureData System for Analytics, from where you must install them manually with the help of IBM support.

Transferring a Netezza 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.

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 Software Versions window, you can see all Netezza Firmware (FDT) and Netezza Host Platform (HPF) files that are available in the HFS of your z/OS data server. Select the appropriate check boxes in the first column of the table to mark the files that you want to transfer.
7. Click **Transfer** to complete the process.

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
- Access Server packages
- Replication engine packages
- Netezza Performance Server (NPS)
- Netezza Firmware (FDT)
- Netezza Host Platform (HPF)

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 hierarchical file system (HFS) of your z/OS data server, manually delete no longer needed IBM DB2 Analytics Accelerator for z/OS or Netezza update packages from the HFS. To do so, open a UNIX System Services shell and delete the files by using the **rm** command.

Switching from 24:00:00 conversion to native 24:00:00 support

If you have so far converted 24:00:00 time values to 23:59:59 by using the `LOAD_ENABLE_HOUR24_CONVERSION` configuration parameter and now want to use the native 24:00:00 support provided by newer versions of the Netezza database, you must remove this parameter or set it to the value `false`. In addition, you must reload accelerator tables that contain converted values.

Before you begin

The native 24:00:00 support was introduced with version 4 PTF-2 of the product. Therefore, before completing the steps listed hereafter, make sure that the following conditions apply:

- IBM DB2 Analytics Accelerator for z/OS Version 4 PTF-2 has been installed.
- Your version of the Netezza Performance Server (NPS) is 7.0.2 P11 or a higher 7.0.2 version. Alternativlely, you can use version 7.0.4.3 or higher.
- In case you use the incremental update function: your version of IBM InfoSphere Change Data Capture for z/OS (CDC) must be 10.2.1 or higher.

Procedure

1. Ask the network administrator or the person who did the TCP/IP setup for the IP address (virtual IP or wall IP address) of the accelerator. Make a note of this information. You need to enter it as you complete the steps that follow.
2. 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.
3. Log on to TSO/ISPF.
4. 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 10.101.8.8 1600
   ```

5. When prompted, enter the console password. The initial password is `dwa-1234`. You must change this password at the first logon.
6. Press the Pause key, then Enter to display the following screen:

   ```
   ******************************************************************************
   *         Welcome to the IBM DB2 Analytics Accelerator Console
   ******************************************************************************

   You have the following options:

    (1) - Change the Configuration Console Password
    (2) - (Menu) Run Netezza Commands
    (3) - (Menu) Run Accelerator Functions
    (4) - (Menu) Manage Hardware
    (5) - (Menu) Manage Incremental Updates

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

7. Type 3 and press Enter to display the submenu:

   ```
   main -> 3
   --------------------------------------------------------
   You have the following options:

   (0) - Go back one level
   (1) - Obtain pairing code, IP address, and port
   (2) - List paired DB2 subsystems
   (3) - Set resource limits for DB2 subsystems
   (4) - Clear query history
   (5) - Specify the priority of maintenance tasks
   (6) - Set the DB2 subsystem for time synchronization
   (7) - Restart accelerator process
   (8) - Disable the conversion mode for 24:00:00
   ```

8. Type 8 and press Enter. This disables the conversion of 24:00:00 time values.
9. Type `0` and press Enter to go up one level.

10. Type x and press Enter to exit the console.
11. Reload accelerator tables that contain converted values. To identify such tables, run the following queries in DB2 for z/OS (do not accelerate these queries!):
    a. To identify accelerated tables with time columns:

       ```
       SELECT RTRIM(C.NAME) AS COLNAME, RTRIM(C.TBCREATOR) AS TBCREATOR,
        RTRIM(C.TBNAME) AS TBNAME
        FROM SYSIBM.SYSCOLUMNS AS C, SYSACCEL.SYSACCELERATEDTABLES AS T
        WHERE C.TBCREATOR = T.CREATOR AND C.TBNAME = T.NAME AND C.COLTYPE = 'TIME';
       ```

    b. To identify tables that contain 24:00:00 values, run the following query against each column that was listed as a result of the previous query:

       ```
       SELECT COUNT(*) FROM MYCREATOR.MYTABLE WHERE TIMECOL = '24:00:00';
       ```

       where

       MYCREATOR
       Stands for the name of any table creator listed in the result of query 1.

       MYTABLE
       Stands for the name of any table listed in the result of query 1.

       TIMECOL
       Stands for the name of any table column listed in the result of query 1.

    c. The counterparts (projections) of these columns on the accelerator contain the converted values (23:59:59). Reload the tables that contain these columns.

Chapter 7. 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.

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
```

© Copyright IBM Corp. 2009, 2014

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

Message AQT10200I and SQL code -206 when running stored procedures after migration

Stored procedure calls end with message `AQT10200I` and `SQLCODE= -206` after a migration of IBM DB2 Analytics Accelerator for z/OS from version 2.1.x to a later version.

Symptoms

The following message is displayed:

```
AQT10200I - The OPEN CURSOR operation failed. Error information: "DSNT408I
SQLCODE = -206, ERROR:  ARCHIVE IS NOT VALID IN THE CONTEXT WHERE IT IS USED"
```

Causes

Required program temporary fixes (PTFs) for DB2 for z/OS have not been installed or the respective ++HOLD migration actions have not been applied yet.

Resolving the problem

Follow the steps in section *Installing prerequisite PTFs for DB2 10 for z/OS* of the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

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>`

Removing orphaned system-table entries and catalog-table entries

If an IBM PureData System for Analytics was physically removed, but the accelerator was not removed before by running the *Remove accelerator* function from IBM DB2 Analytics Accelerator Studio or by running the SYSPROC.ACCEL_REMOVE_ACCELERATOR stored procedure, you find invalid entries in a number of DB2 for z/OS catalog tables and system tables.

Symptoms

You find invalid entries in the following tables:

- SYSIBM.IPNAMES
- SYSIBM.USERNAMES
- SYSIBM.LOCATIONS
- SYSACCEL.SYSACCELERATORS
- SYSACCEL.SYSACCELERATEDTABLES

Resolving the problem

1. Disable the accelerator in the DB2 subsystem, by using the `-DIS ACCEL` command or the appropriate function in IBM DB2 Analytics Accelerator Studio.

2. To remove the invalid entries from the system tables, run the following SQL commands in the order indicated:
   a.
   ```
   DELETE FROM SYSIBM.IPNAMES
   WHERE LINKNAME=(SELECT LINKNAME FROM SYSIBM.LOCATIONS
   WHERE LOCATION=(SELECT LOCATION FROM SYSACCEL.SYSACCELERATORS
   WHERE ACCELERATORNAME=<acceleratorName>));
   ```
   b.
   ```
   DELETE FROM SYSIBM.USERNAMES
   WHERE LINKNAME=(SELECT LINKNAME FROM SYSIBM.LOCATIONS
   WHERE LOCATION=(SELECT LOCATION FROM SYSACCEL.SYSACCELERATORS
   WHERE ACCELERATORNAME=<acceleratorName>));
   ```
   c.
   ```
   DELETE FROM SYSIBM.LOCATIONS
   WHERE LOCATION=(SELECT LOCATION FROM SYSACCEL.SYSACCELERATORS
   WHERE ACCELERATORNAME=<acceleratorName>);
   ```
   d.
   ```
   DELETE FROM SYSACCEL.SYSACCELERATORS
   WHERE ACCELERATORNAME=<acceleratorName>;
   ```
   e.
   ```
   DELETE FROM SYSACCEL.SYSACCELERATEDTABLES
   WHERE ACCELERATORNAME=<acceleratorName>;
   ```

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 a 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

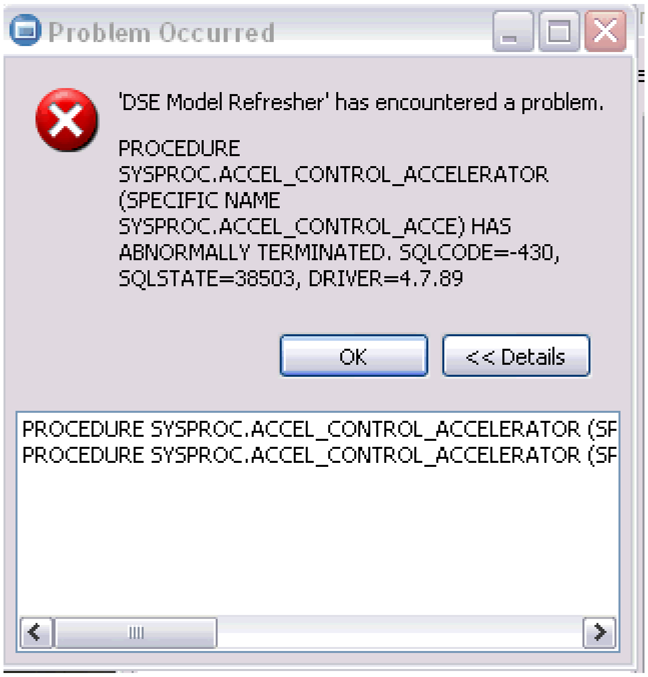
IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter)

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:



'DSE Model Refresher' has encountered a problem.

PROCEDURE SYSPROC.ACCEL_CONTROL_ACCELERATOR (SPECIFIC NAME SYSPROC.ACCEL_CONTROL_ACCE) HAS ABNORMALLY TERMINATED. SQLCODE=-430, SQLSTATE=38503, DRIVER=4.7.89

```
PROCEDURE SYSPROC.ACCEL_CONTROL_ACCELERATOR (SF
PROCEDURE SYSPROC.ACCEL_CONTROL_ACCELERATOR (SF
```

Figure 12. SQL code -430 message window

Causes

This is probably 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. Rerun the AQTSJI00 and AQTSJI01 jobs to verify the correct installation of the IBM DB2 Analytics Accelerator for z/OS stored procedures.

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:

   http://publib.boulder.ibm.com/infocenter/dstudio/v4r1/topic/com.ibm.datatools.qrytune.configothers.doc/topics/enabledb2zfromclient_ds.html
2. To grant the EXECUTE privilege to other users, proceed as follows:

   http://publib.boulder.ibm.com/infocenter/dstudio/v4r1/topic/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 IBM PureData System for Analytics.

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 might expose this symptom.

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" on page 40
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.

Replication column not shown

The **Replication** column is not displayed in the Object List Editor, although the incremental update function has been enabled for one or more accelerators.

Symptoms

You do not see the **Replication** column in the Object List Editor, and there is no way to select it for display.

Causes

To be able to remember the last column selection of a user, IBM DB2 Analytics Accelerator Studio saves information to a couple of files. These files are not overwritten when IBM DB2 Analytics Accelerator Studio is updated. So if you updated IBM DB2 Analytics Accelerator Studio, it still uses the column information that was saved by the older version of the program. The column information was probably created when the incremental update function did not exist. Hence there can be no information about the **Replication** column.

Resolving the problem

You must delete two files so that they are created anew when you start IBM DB2 Analytics Accelerator Studio the next time:

1. In a Command Prompt window or file manager, change to the following directory:

   `<Data-Studio-workspace>\.metadata\.plugins\com.ibm.datatools.uom.ui`

   where `<Data-Studio-workspace>` is the workspace or working directory of IBM Data Studio. If you do not know your workspace path, select **File** > **Switch Workspace** > **Other** from the main menu to find out.
2. In this directory, delete the following files:
   - `custom_properties.dat`
   - `property_arrangement.dat`
3. Start IBM DB2 Analytics Accelerator Studio. If you do not see the **Replication** column in the Object List Editor, pick it from the column selection dialog.

Replication status: *ERROR*

To analyze a problem indicated by a replication status of *ERROR*, you can check the event list on the IBM DB2 Analytics Accelerator console for details.

Symptoms

The header of an accelerator view shows **Replication status:** *ERROR*.

Resolving the problem

1. Open the IBM DB2 Analytics Accelerator console as described in *Enabling incremental updates for a DB2 subsystem* in the *IBM DB2 Analytics Accelerator: Installation Guide*.
2. Enter `11` to select the option `(11) - Show replication events on a subsystem`.
3. Locate and read the error information.
4. If possible, take corrective action.

Table removal very slow

Removing tables that have been enabled for incremental updates takes an excessively long time.

Symptoms

The table removal process seems to run endlessly.

Resolving the problem

Stop the incremental update process for the accelerator before removing tables. Proceed as follows in IBM DB2 Analytics Accelerator Studio:

1. Cancel the running removal process.
2. Stop incremental updates for the accelerator.
3. Remove the tables.
4. Restart incremental updates for the accelerator.

Incorrect load recommendation `Unknown - no statistics`

The load recommendation for a table remains `Unknown - no statistics` even if the table has been loaded and the table data has not changed.

Symptoms

The load recommendation does not change, but remains `Unknown - no statistics`.

Causes

Load recommendations depend on DB2 for z/OS run-time statistics from the SYSIBM.SYSTABLESPACESTATS catalog table. If DB2 did not collect run-time statistics for a table, load recommendations cannot be computed.

A common cause for this situation is a table space that was restored by the RECOVER utility. The RECOVER utility suspends the collection of the relevant DB2 run-time statistics for the affected table space or partition until the REORG utility is run the next time.

Diagnosing the problem

You can diagnose the problem by running the following catalog query in DB2 for z/OS (replace `<schema>` and `<name>` with the schema and the name of the table that shows the incorrect load recommendation):

```
SELECT
TB.TSNAME, TSS.PARTITION, TSS.UPDATESTATSTIME, TSS.REORGLASTTIME,
BIGINT(TSS.REORGINSERTS) + TSS.REORGUPDATES + TSS.REORGDELETES +
TSS.REORGMASSDELETE
FROM SYSIBM.SYSTABLES TB
JOIN SYSIBM.SYSTABLESPACESTATS TSS
ON TB.DBNAME = TSS.DBNAME AND TB.TSNAME = TSS.NAME
WHERE TB.CREATOR = '<schema>' AND TB.NAME LIKE '<name>';
```

If this query does not return rows for a partition or if the value of the returned row is NULL in the last two columns, run-time statistics are not available for the table.

Resolving the problem

Run the REORG utility on the affected table space or partition to re-enable the collection of DB2 for z/OS run-time statistics.

Note: The next time you reload the affected table or partition after running REORG, you will still see the message `Unknown - no statistics` because the meta-data on the accelerator side has not been refreshed yet. Valid load recommendations will be displayed for the next but one and subsequent reloads.

Repeated IBM InfoSphere Change Data Capture for z/OS errors after abnormal end of DB2 for z/OS

IBM InfoSphere Change Data Capture for z/OS repeatedly reports errors after an abnormal end of DB2 for z/OS.

Symptoms

An error message similar to the following is displayed:

```
CHC9210E (CDCSZA1) DB2 CAF IFI-READS request has failed. RetturnCode=X'00000100',
ReasonCode=X'00F30018'
```

Causes

If incremental updates are configured, and DB2 for z/OS ends unexpectedly, IBM InfoSphere Change Data Capture for z/OS is not notified of the shutdown. IBM InfoSphere Change Data Capture for z/OS still "expects" to find the instance of DB2 that was previously running. When DB2 for z/OS is restarted, IBM InfoSphere Change Data Capture for z/OS tries to renew the connection to the known DB2 instance. That is, it uses the session configuration or parameters of the previously stalled connection. These will never be correct, so the error message is displayed after each retry.

Resolving the problem

Restart the IBM InfoSphere Change Data Capture for z/OS address space.

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 hierarchical file system (HFS) 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 HFS. 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.

Query fails with SQLSTATE: HY000 SQLCODE: 46 'ERROR'

A running query fails when a distribution key or organizing key is changed at the same time. The message `SQLSTATE: HY000 SQLCODE: 46 'ERROR'`is returned.

Symptoms

The full error message is similar to the following:

```
SQLSTATE: HY000 SQLCODE: 46 'ERROR:  RelationClearRelation: relation
    'PARTSUPP-UID_01200030' (2881502) in database 2842888 altered in a concurrent
    session'.
```

The problem occurs in connection with Netezza Performance Server 7.0.4.

Resolving the problem

Rerun the query.

Appendix. Sample SQL for EXPLAIN tables

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. If you do not want to run the z/OS jobs, you can also create the tables by submitting the SQL sample code provided here.

SQL

To create the EXPLAIN tables, replace <yourname> in the following SQL sample with an appropriate user ID and submit it.

```
SET CURRENT SQLID='<yourname>';
CREATE DATABASE DSNQBINC
  BUFFERPOOL BP2
  INDEXBP    BP1
  CCSID      UNICODE
  STOGROUP   SYSDEFLT;
COMMIT;

SET CURRENT SQLID='<yourname>';
CREATE TABLESPACE DSNQBITS
  IN DSNQBINC
  USING STOGROUP SYSDEFLT
  PRIQTY -1 SECQTY -1
  ERASE  NO
  FREEPAGE 0 PCTFREE 5
  GBPCACHE CHANGED
  TRACKMOD YES
  SEGSIZE 16
  BUFFERPOOL BP0
  LOCKSIZE ANY
  LOCKMAX SYSTEM
  CLOSE NO
  COMPRESS NO
  CCSID      UNICODE
  DEFINE YES
  MAXROWS 255;
COMMIT;


-- DROP TABLE DSN_QUERYINFO_TABLE;

CREATE TABLE DSN_QUERYINFO_TABLE(
QUERYNO INTEGER NOT NULL WITH DEFAULT,
QBLOCKNO SMALLINT NOT NULL WITH DEFAULT,
QINAME1 VARCHAR(128) NOT NULL WITH DEFAULT,
QINAME2 VARCHAR(128) NOT NULL WITH DEFAULT,
APPLNAME VARCHAR(24) NOT NULL WITH DEFAULT,
PROGNAME VARCHAR(128) NOT NULL WITH DEFAULT,
VERSION VARCHAR(122) NOT NULL WITH DEFAULT,
COLLID VARCHAR(128) NOT NULL WITH DEFAULT,
GROUP_MEMBER VARCHAR(24) NOT NULL WITH DEFAULT,
SECTNOI INTEGER NOT NULL WITH DEFAULT,
SEQNO INTEGER NOT NULL WITH DEFAULT,
EXPLAIN_TIME TIMESTAMP NOT NULL WITH DEFAULT,
TYPE CHAR(8) NOT NULL WITH DEFAULT,
REASON_CODE SMALLINT NOT NULL WITH DEFAULT,
QI_DATA CLOB(2M) NOT NULL WITH DEFAULT,
```

© Copyright IBM Corp. 2009, 2014

```
SERVICE_INFO BLOB(2M) NOT NULL WITH DEFAULT,
QB_INFO_ROWID ROWID NOT NULL GENERATED ALWAYS
) IN DSNQBINC.DSNQBITS CCSID UNICODE;

SET CURRENT SQLID='<yourname>';
CREATE LOB TABLESPACE DSNLOBT4
  IN DSNQBINC
  USING STOGROUP SYSDEFLT
  PRIQTY -1 SECQTY -1
  ERASE  NO
  GBPCACHE CHANGED
  LOG YES
  DSSIZE 4 G
  BUFFERPOOL BP8K0
  LOCKSIZE ANY
  LOCKMAX SYSTEM
  CLOSE YES
  DEFINE YES;
COMMIT;


CREATE AUX TABLE DSN_QUERYINFO_AUX
IN DSNQBINC.DSNLOBT4
STORES DSN_QUERYINFO_TABLE COLUMN QI_DATA;

CREATE TYPE 2 INDEX DSN_QUERYINFO_AUXINX
ON DSN_QUERYINFO_AUX;

SET CURRENT SQLID='<yourname>';
CREATE LOB TABLESPACE DSNLOBT5
  IN DSNQBINC
  USING STOGROUP SYSDEFLT
  PRIQTY -1 SECQTY -1
  ERASE  NO
  GBPCACHE CHANGED
  LOG YES
  DSSIZE 4 G
  BUFFERPOOL BP8K0
  LOCKSIZE ANY
  LOCKMAX SYSTEM
  CLOSE YES
  DEFINE YES;
COMMIT;



CREATE AUX TABLE DSN_QUERYINFO_AUX2
IN DSNQBINC.DSNLOBT5
STORES DSN_QUERYINFO_TABLE COLUMN SERVICE_INFO;

CREATE TYPE 2 INDEX DSN_QUERYINFO_AUXINX2
ON DSN_QUERYINFO_AUX2;
```

Glossary

This glossary includes terms and definitions for IBM DB2 Analytics Accelerator Studio.

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.

To view glossaries for other IBM products, go to www.ibm.com/software/globalization/terminology (opens in new window).

A

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 the table columns must be on the same processing node. Leads to shorter query response times.

coordinator node
The active Netezza host.

D

data slice
A portion of a table that is processed by a worker node.

DDF See distributed data facility.

distributed data facility (DDF)
A set of DB2 for z/OS components through 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.

© Copyright IBM Corp. 2009, 2014

E

ensemble
A collection of one or more System z® nodes (including any attached zBX) that are managed as a single logical virtualized system by the zManager, through the use of a Hardware Management Console.

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.

M

manifest
An XML file that contains metadata and procedural instructions for an IBM DB2 Analytics Accelerator for z/OS update package, such as the version of the package, update information (which versions are updated to which newer version), and parameters for internal package processing.

P

plan (also execution plan)
A Netezza file with an extension of pln that shows in which order the tables in a query are scanned.

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 for z/OS handle certain SQL expressions differently. In some cases, IBM DB2 Analytics Accelerator for z/OS 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 Netezza processing nodes.

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.
- In Netezza terminology, a logical collection of extents that are assigned to a table. A table space contains all the disk space that is allocated to a

given table or table fragment and includes pages allocated to data and to indexes, pages that store TEXT or BYTE data in the dbspace, and bitmap pages that track page use within the extents.

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

wall IP address
A Netezza term for the virtual IP address that is used to contact the Netezza system over the network. A floating IP address that is automatically assigned to the active Netezza host. Using the wall IP address from the z/OS data server, the active Netezza host can always be reached. This is transparent behavior in case of a failover.

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

© Copyright IBM Corp. 2009, 2014

Notices

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

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

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

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

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

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

Any references in this information to non-IBM 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.

© Copyright IBM Corp. 2009, 2014

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

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

IBM Deutschland GmbH
Dept. M358
IBM-Allee 1
71139 Ehningen
Germany

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

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

COPYRIGHT LICENSE:

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

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_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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.

Netezza is a registered trademark of Netezza Corporation, an IBM Company.

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

Readers' Comments — We'd Like to Hear from You

IBM DB2 Analytics Accelerator for z/OS
User's Guide
Version 4.1.0

Publication No. SH12-7040-01

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via email to: swsdid@de.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Company or Organization

Phone No.

Address

Email address

Readers' Comments — We'd Like to Hear from You

SH12-7040-01

Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Research & Development GmbH
Dept. 0446
Schoenaicher Str. 220
71032 Boeblingen
GERMANY

Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

Product Number: 5697-DAB

SH12-7040-01