IBM DB2 Analytics Accelerator for z/OS
Version 5.1.0

User's Guide

SH12-7061-05

Note

Before you use this information and the product it supports, read the information in "Notices" on page 221.

Sixth Edition, July 2017

This edition applies to version 5.1.0 of IBM DB2 Analytics Accelerator for z/OS (product number 5697-DA5), and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces SH12-7061-04. Changes to this edition are marked with a vertical bar.

© Copyright IBM Corporation 2009, 2017.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures vii

About this book ix

Who should read this book xi

What's new in version 5? xiii

Accessibility features xv

Chapter 1. Introduction 1
Connecting to a data server 1
Creating a database connection profile 1
Connecting to a database 1
Adding accelerators 1
Defining the data to load into an accelerator 2
Selecting the proper tables 2
Checking the table size 2
Choosing a distribution key 2
Choosing an organizing key 2
Working with accelerator-shadow tables 3
Loading data into selected tables 3
Enabling tables for query acceleration 4
Disabling query acceleration for a table 4
Enabling tables for incremental updates 4
Disabling incremental updates 4
Moving partition or table data to an accelerator . 4
Removing tables 4
Order in which to complete tasks 4

Chapter 2. Basic concepts 5
Accelerator 5
Virtual accelerator 6
Static SQL query support 6
Accelerator-shadow table 6
How to select tables for query acceleration 7
Distribution keys 9
Organizing keys 11
Zone maps 12
Tables states 13
Moved tables or partitions 15
Accelerator-only tables and in-database transformation 16
Workload balancing 20
IBM DB2 Analytics Accelerator for z/OS system table entries 20

Chapter 3. Conditions for query routing to an accelerator 21
Configuring DB2 for z/OS for the acceleration of dynamic SQL queries 21
Configuring DB2 for z/OS for the acceleration of static SQL queries 23
Supported queries 26
Supported data types 27
Supported SQL functions and expressions 31
Supported special registers 34
Conditions that prevent query routing to an accelerator 35
Isolation levels 38
Incompatibilities between DB2 for z/OS and IBM DB2 Analytics Accelerator for z/OS 39
Correlated subqueries 39
COUNT_BIG function 39
Data types 40
Division and multiplication of decimal values . . 40
Encoding issues 41
Exception handling 42
Maximum row length 42
Predicate evaluation 42
Scalar functions 43
SQL issues 45
Time and date values 45
Incompatibilities related to IBM InfoSphere Data Replication for z/OS 47
Restrictions of accelerator-only tables 48
New features in DB2 10 for z/OS 51
New features in DB2 11 for z/OS 55
Differences in floating-point error tolerances . . . 56

Chapter 4. IBM DB2 Analytics Accelerator Studio. 59

Chapter 5. Interface elements of IBM DB2 Analytics Accelerator Studio . . . 61
Menu bar 61
Administration Explorer 61
Object List Editor 63
Accelerator views 66
Accelerator view for regular accelerators . . . 66
Accelerator view for virtual accelerators 75

Chapter 6. Tasks 79
Creating a database connection profile 79
Connecting to a database server 81
Exploring a database server 82
Adding accelerators 82
Obtaining the pairing code for authentication . . 83
Completing the authentication using the Add Accelerator wizard 85
Adding virtual accelerators 86
Displaying accelerator networking details 87
Defining tables on an accelerator 88
Determining the status of an accelerator 90
Enabling an accelerator 90
Disabling an accelerator 91
Preparing tables for workload balancing and accelerating static SQL queries 92

© Copyright IBM Corp. 2009, 2017

Enabling or disabling an accelerator for members of a data sharing group . . . 93
Loading tables . . . 93
Enabling tables for query acceleration . . . 98
Disabling query acceleration for tables . . . 99
Specifying or changing a distribution key or organizing keys . . . 100
Synchronizing schemas . . . 103
Running an SQL script from IBM DB2 Analytics Accelerator Studio . . . 105
Running IBM Netezza Analytics stored procedures 106
EXPLAIN information . . . 107
Displaying an access plan graph . . . 112
Types of access plan graphs and nodes in these graphs . . . 113
Selecting and ordering query monitoring columns 116
Selectable columns in the **Query Monitoring** section . . . 119
Canceling tasks . . . 125
Removing tables from an accelerator . . . 126
Exporting a table specification . . . 126
Importing a table specification . . . 127
Removing accelerators . . . 128
Tracing . . . 129
Configuring the trace behavior . . . 129
Saving trace information . . . 131
Collecting CDC trace information . . . 133
Configuring the FTP server for the **Save Trace** function . . . 134
Clearing trace information and replication events . . . 135
Setting a refresh interval or a skew threshold . . . 135
Updating accelerator tables continually . . . 136
Starting or stopping incremental updates . . . 138
Including or excluding tables from incremental updates . . . 139
Monitoring incremental updates . . . 143
Using the event viewer for incremental updates 145
Warnings related to incremental update processes . . . 146
Collecting CDC trace information . . . 147
What to do after changing a table or table space in DB2 for z/OS . . . 148
Freeing up storage in DB2 for z/OS . . . 149
Archiving partition or table data with the High-Performance Storage Saver . . . 151
Impact of special register settings . . . 155
Restoring archived partitions . . . 157
How to handle the RBA and LRSN conversion in DB2 11 for z/OS . . . 159
Federated access . . . 159
Granting access to remote tables . . . 162
Creating reference tables . . . 162
Revoking table access . . . 163
Removing reference tables . . . 164
Installing updates . . . 165
Updating IBM DB2 Analytics Accelerator Studio 166
Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine . . . 168
Updating other Netezza software . . . 173
Removing obsolete software packages from an accelerator . . . 174
Switching from 24:00:00 conversion to native 24:00:00 support . . . 175
Viewing the encryption status . . . 177

Chapter 7. Status information about the system and the incremental update function . . . 179

Chapter 8. Troubleshooting . . . 183

Redundant CDC metadata . . . 183
Firewall blocks operations because crucial ports are closed . . . 183
IBM DB2 Analytics Accelerator Studio does not start . . . 184
IBM DB2 Analytics Accelerator Studio on Linux does not start . . . 185
Message AQT10200E and SQL code -206 when running stored procedures after migration . . . 185
Package not found when running a stored procedure from IBM DB2 Analytics Accelerator Studio . . . 185
Removing orphaned system-table entries and catalog-table entries . . . 187
Long-running tasks are not completed . . . 187
SQL code -430 from IBM DB2 Analytics Accelerator for z/OS stored procedures . . . 188
Database access problems after updating IBM DB2 Analytics Accelerator Studio or IBM Optim Query Tuner . . . 189
DRDA connection does not work . . . 189
Different results for floating-point aggregates . . . 189
Replication column not shown . . . 190
Replication status: *ERROR* . . . 190
Table removal very slow . . . 191
Repeated IBM InfoSphere Data Replication for z/OS errors after abnormal end of DB2 for z/OS . 191
Stored procedure calls fail with message AQT10206I while temporary files are being created . 192
Query history is not displayed - error message returned . . . 192
DB2 for z/OS returns `SQLCODE -30040` . . . 193
Query fails with SQLSTATE: HY000 SQLCODE: 46 'ERROR' . . . 193
Restoring of archived partition fails with `'Duplicate key'` error . . . 193
Table definition fails with error message `AQT1005I` 194
SQL error as you try to enable tables for incremental updates . . . 195

Appendix A. Mappings of product SQL codes to DB2 for z/OS SQL codes . . . 197

Appendix B. Impact of special register, ZPARM, or bind option settings on accelerator-only tables . . 205

Glossary . . . 211
A . . . 211
C . . . 211
D . . . 212
F . . . 212
I. . . . 212
L . . . 212
M . . . 212
P . . . 213
Q . . . 213
R . . . 213
S . . . 213
T . . . 214
V . . . 214
W . . . 214

Index . . . 215

Notices . . . 221
Trademarks . . . 223
Terms and conditions for product documentation 223

Figures

1. IBM DB2 Analytics Accelerator Studio tasks in the overall process flow 5
2. Access plan graph of the query in the example 9
3. Possible table state transitions 15
4. IBM DB2 Analytics Accelerator for z/OS setup 59
5. Administration Explorer 62
6. Object List Editor 63
7. Accelerator view for regular accelerators 67
8. Accelerator view for virtual accelerators 75
9. Database server connection in the Administration Explorer 82
10. Icons indicating a not yet synchronized schema change 105
11. Message box that comes up when mouse pointer is hovering over a schema name with a change indicator icon 105
12. **Replication status** and functional link in the header of the accelerator view 139
13. Timestamp columns and **Replication** button in the accelerator view 142
14. **Replication latency** in the accelerator view 145
15. Federated access diagram 161
16. All members in the same LPAR 181
17. Members in two different LPARs 182
18. SQL code -430 message window 188

© Copyright IBM Corp. 2009, 2017

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, 2017

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, 2017

What's new in version 5?

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 temporal DB2 tables (tables with system or business period definitions)
- Encryption of data in motion
- Option to integrate *IBM Netezza® Analytics* modeling and prediction algorithms
- Installation enhancements
- Improved backward compatibility. Stored procedures can now be at a higher level than accelerator software.

Enhancements in version 5 PTF-5

- Columns can be added to existing accelerator-shadow tables (no need to remove, redefine, and reload a table after adding columns to a DB2 table)
- Faster loading of accelerator-only tables: (Many) more than just one row are now inserted in a single transaction (multi-row inserts)
- Federated access: Queries can now reference tables of a remote DB2 subsystem (the DB2 subsystem that passes the query to the accelerator does not have to own the referenced tables).

Enhancements in version 5 PTF-4

- Archived partitions can be restored more than once from different accelerators.
- Support for multi-row inserts (beta).
- IP address and port of Capture Agent can be changed without stopping incremental updates.
- Numerous fixes (see release notes)

IBM Knowledge Center and book revision July 2016:

The suffixes of stored-procedure messages have been corrected. They now show the proper letters E, I, or W.

Enhancements in version 5 PTF-3:

- Improved load performance for small tables
- Numerous fixes (see release notes)

Enhancements in version 5 PTF-2:

- Code stabilization
- Reduction of functional limitations

Enhancements in version 5 PTF-1:

- Maintenance release with numerous internal fixes and improvements

© Copyright IBM Corp. 2009, 2017

Accessibility features

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

Accessibility features

The following list includes the major accessibility features in IBM DB2 Analytics Accelerator Studio:

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

The IBM DB2 Analytics Accelerator for z/OS Information Center, and its related publications, are accessibility-enabled. The accessibility features of the information center are described in the topic Accessibility features in Eclipse.

Keyboard navigation

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

Navigating the user interface using the keyboard).

Interface information

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

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

IBM and accessibility

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

© Copyright IBM Corp. 2009, 2017

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 79

Connecting to a database

To connect IBM DB2 Analytics Accelerator Studio to a DB2 for z/OS database, double-click one of the database connection profiles in the Administration Explorer.

Adding accelerators

When a connection to a database has been established through one of your database connection profiles, you can see all currently deployed accelerators in the Object List Editor. You access these interface elements from the Accelerator Perspective in IBM DB2 Analytics Accelerator Studio. To connect a newly installed accelerator, use the Add Accelerator wizard in IBM DB2 Analytics Accelerator Studio.

Related tasks:

"Completing the authentication using the Add Accelerator wizard" on page 85

© Copyright IBM Corp. 2009, 2017

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 determine the data to be queried. For each DB2 for z/OS table that you select, a corresponding table is created on one or more accelerators. These tables are called accelerator-shadow tables. Accelerator-shadow tables are not necessarily copies of their originals. Often, they contain fewer columns than the tables they stem from because columns with unsupported data types are excluded. Accelerator-shadow tables must be filled with data before they can be queried. This is done in a later step (see loading accelerator-shadow tables).

Related tasks:

"Defining tables on an accelerator" on page 88

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 rows in accelerator-shadow tables 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 9

"How to select tables for query acceleration" on page 7

Related tasks:

"Specifying or changing a distribution key or organizing keys" on page 100

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 rows of accelerator-shadow tables 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 accelerator-shadow 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 11

"Zone maps" on page 12

"How to select tables for query acceleration" on page 7

Related tasks:

"Specifying or changing a distribution key or organizing keys" on page 100

Working with accelerator-shadow tables

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

Loading data into selected tables

To enable users to run accelerated queries against selected database tables, you must load the empty tables that you previously added to the accelerator with data. In doing so, you create a snapshot copy of your existing DB2 for z/OS data on the accelerator.

The DB2 Optimizer calculates the expected response times for incoming queries. If an accelerator can process a query against the selected tables faster than the database management system, the query is routed to the accelerator and evaluated against the populated accelerator-shadow tables.

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

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

Related concepts:

"Updating accelerator tables continually" on page 136

Related tasks:

"Loading tables" on page 93

Enabling tables for query acceleration

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

Disabling query acceleration for a table

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

Enabling tables for incremental updates

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

Disabling incremental updates

Disabling incremental updates excludes tables from the incremental update process.

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 (HPSS) to move this data to an accelerator to free up costly storage on your DB2 for z/OS data server. A positive side-effect is that query processing and table maintenance become faster because the remaining data consists of fewer table rows and indexes require fewer entries.

Removing tables

If tables are no longer needed for queries, you can use the removal function in IBM DB2 Analytics Accelerator Studio to remove the accelerator-shadow tables from the accelerator.

Related tasks:

"Removing tables from an accelerator" on page 126

Order in which to complete tasks

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

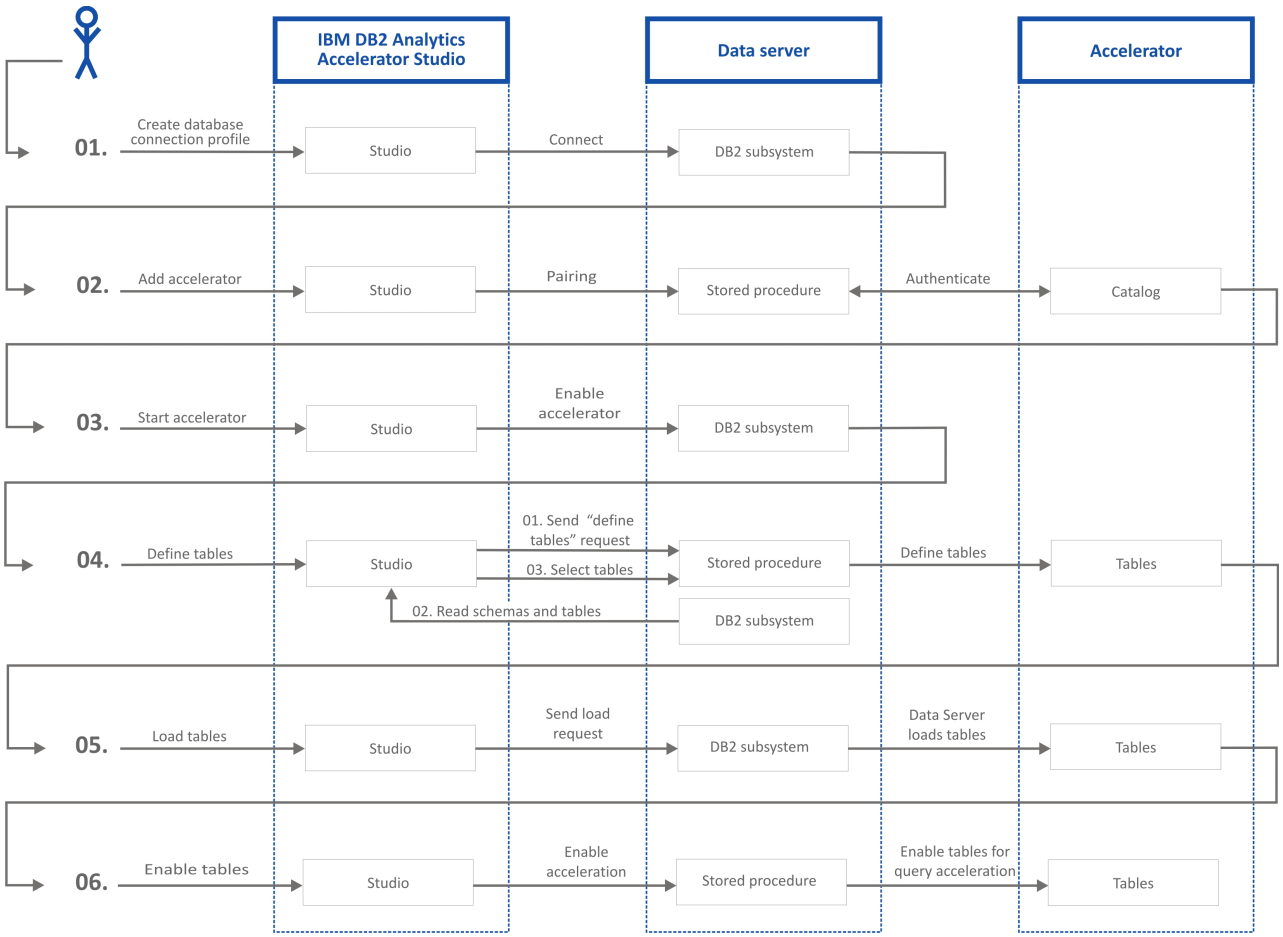


Figure 1. IBM DB2 Analytics Accelerator Studio tasks in the overall process flow

Chapter 2. Basic concepts

Get familiar with the basic concepts of IBM DB2 Analytics Accelerator for z/OS.

Accelerator

IBM DB2 Analytics Accelerator for z/OS consists of a high-performance hardware platform and an optimized database query engine. These components work together to support a variety of data analysis and business reporting tasks.

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

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

© Copyright IBM Corp. 2009, 2017

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 107

Related tasks:

"Displaying an access plan graph" on page 112

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 92

Accelerator-shadow table

An accelerator-shadow table represents a table in your database. When you define accelerator-shadow tables in IBM DB2 Analytics Accelerator Studio, you instruct IBM DB2 Analytics Accelerator for z/OS to copy the definition of that table (an empty table) to an accelerator.

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

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

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

When tables are added to an accelerator, DB2 for z/OS keeps track of these accelerator-shadow tables in its system catalog. The system table entries enable the optimizer of the database management system to complete the following tasks:

- Query matching, that is, comparing an incoming query with the information stored in the system tables to check whether the query can be handled by IBM DB2 Analytics Accelerator for z/OS.
- Query evaluation, that is, calculating the estimated query response time. If a performance gain can be predicted, the query is sent to IBM DB2 Analytics Accelerator for z/OS. If not, the query is handled by the database management system itself.

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

Related tasks:

"Defining tables on an accelerator" on page 88

How to select tables for query acceleration

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

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

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

Query:

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

The query references the following tables:

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

You would thus define these tables on the accelerator.

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

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

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

Furthermore, the query contains the condition

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

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

Access plan graph

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

In the access plan graph for this example, you can see that TPCH.LINEITEM and TPCH.ORDERS are tables because they appear in the access plan graph. Views and aliases are not represented in access plan graphs. See Figure 2 on page 9.

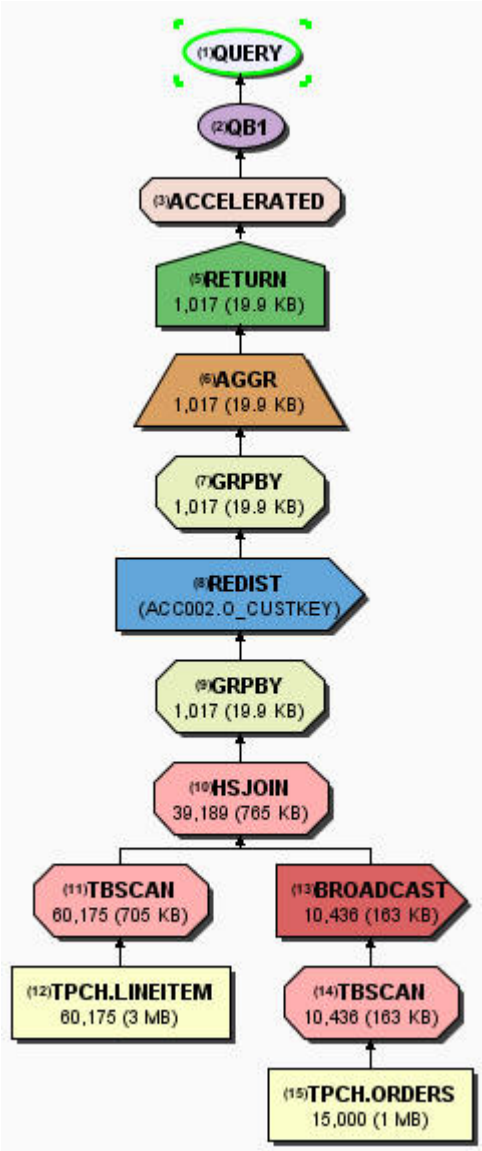


Figure 2. Access plan graph of the query in the example

Related concepts:

"Choosing a distribution key" on page 2

"Choosing an organizing key" on page 2

Related tasks:

"Displaying an access plan graph" on page 112

Related reference:

"Types of access plan graphs and nodes in these graphs" on page 113

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 the rows in an accelerator-shadow table. 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 accelerator-shadow 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 of both accelerator-shadow tables reside on the same worker node (colocated join). You can instigate colocated joins by selecting the join columns as the distribution key for both tables.

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

Best practices for selecting distribution keys

Follow these guidelines when using distribution keys:

- For fact tables and the largest dimension tables, specify a distribution key. For all other tables, start with the default (random distribution).
- Prefer single-column distribution keys to multiple-column keys.
- Select a column as the distribution key that distributes the table rows as evenly as possible across the worker nodes, that is, select primary keys or columns with a unique index. Do not specify columns that only have a small number of distinct values. For example, Boolean values like *male* and *female* would place the table rows on two of the worker nodes only.
- When joining two large tables, try to colocate the tables by specifying the join columns as the distribution key for both tables. Note that to instigate a colocated join, the data types of the join columns in both tables must match.

- In general, do not specify a date, time, or timestamp column as the distribution key. Such keys would also distribute the table rows evenly across the worker nodes, but many queries restrict the number of result rows by a range predicate on the date, time, or timestamp column. For example, a query retrieving revenue figures from last month only would be processed on a single worker node if a column MONTH was selected as the distribution key because rows with equal values in this column would all be assigned to the same data slice.
- Although adverse to the goal of faster query acceleration, a distribution key on the primary key or unique constraint of a table helps if you must update tables continually. Therefore, decide what is more important to you: query performance or the latency of incremental updates. The accelerator automatically picks a distribution key when you enable incremental updates for a table unless you have defined a distribution key by yourself.

Organizing keys

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

Organizing keys work together with zone maps, which allow the query engine to identify the disk extents (addressable storage blocks on the disk) that contain the clusters of rows relevant to a query.

Best practices for selecting organizing keys

- In general, the accelerator should be able to process your queries with adequate performance so that organizing keys are not needed. However, the use of an organizing key, particularly on large fact tables can result in table scan performance gains by multiple orders of magnitude.
- An organizing key has no effect if the table is too small. The **Organized** column in the Accelerator view reflects this by not showing a value for the degree of organization (percentage). The recommended minimum table size (compressed size on the IBM PureData® System for Analytics) 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 accelerator-shadow tables that you want to update continually by using the incremental update function. Use the primary key columns of the table for the organizing key. When the incremental update function applies DELETE or UPDATE statements, the accelerator has to search for the rows that must be deleted or updated. Defining an organizing key accelerates these operations.

- Organizing keys work best on large tables (with millions or more records), under the condition that your queries restrict on column values that are rather scattered across the table. For example, if a query restricts on a set of product IDs and customer IDs from a table of sales records in which the records are presorted by the date on which a record was added, the customer ID column and the product ID column would make good candidates for an organizing key. These keys would have the effect that records with the same customer ID and product ID are grouped together, leading to the table scan performance benefits described earlier.

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

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

  Example:

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

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

  As additional columns are chosen as organizing keys, the benefit of predicates on column subsets is reduced. Four keys are the allowed maximum. However, there is hardly a need to select more than three.
- Organizing keys are also useful the more frequently the columns that you specified as keys are used in query predicates, alone or in combination, and if the column cardinality is high (that is, if the columns have many different values).
- For organizing keys to have a positive effect on the table-scan performance, a query does not have to reference all the columns that have been defined as organizing keys. It is enough if just one of these columns is addressed in a query predicate. However, the benefit is higher if all columns are used because this means that the relevant rows are kept in a smaller number of 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 clustered table rows cause a processing overhead when you load or update these tables on the accelerator.

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.

Tables states

A table can go through various states while it is being processed by IBM DB2 Analytics Accelerator for z/OS. The meaning of these states is explained here. In addition, it is shown which operation you can run on a table in a particular state. Finally, a diagram shows the possible shifts from one state to another (state transitions).

Table states are displayed in IBM DB2 Analytics Accelerator Studio, in the Accelerator view for regular accelerators (**Tables** section, **Last Load** column). A table can be in one of the following states:

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

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

Loaded
The load process has been finished.

SchemaModificationInProgress
The schema of a table is being modified. This happens when you change distribution key or organizing key columns, or if background jobs create foreign-key relationships with columns of the table. A table is also in this state when you run a synchronization job on an accelerator-shadow table because columns have been added to an original DB2 for z/OS table.

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

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

ArchivingInProgress
Original DB2 table data (partitions) is being archived on an accelerator.

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

Possible operations

Table 2 shows which operations you can run on a table in a particular state.

Table 2. Possible operations on a table depending on its state

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

State transitions

Figure 3 on page 15 shows the state transitions that are possible.

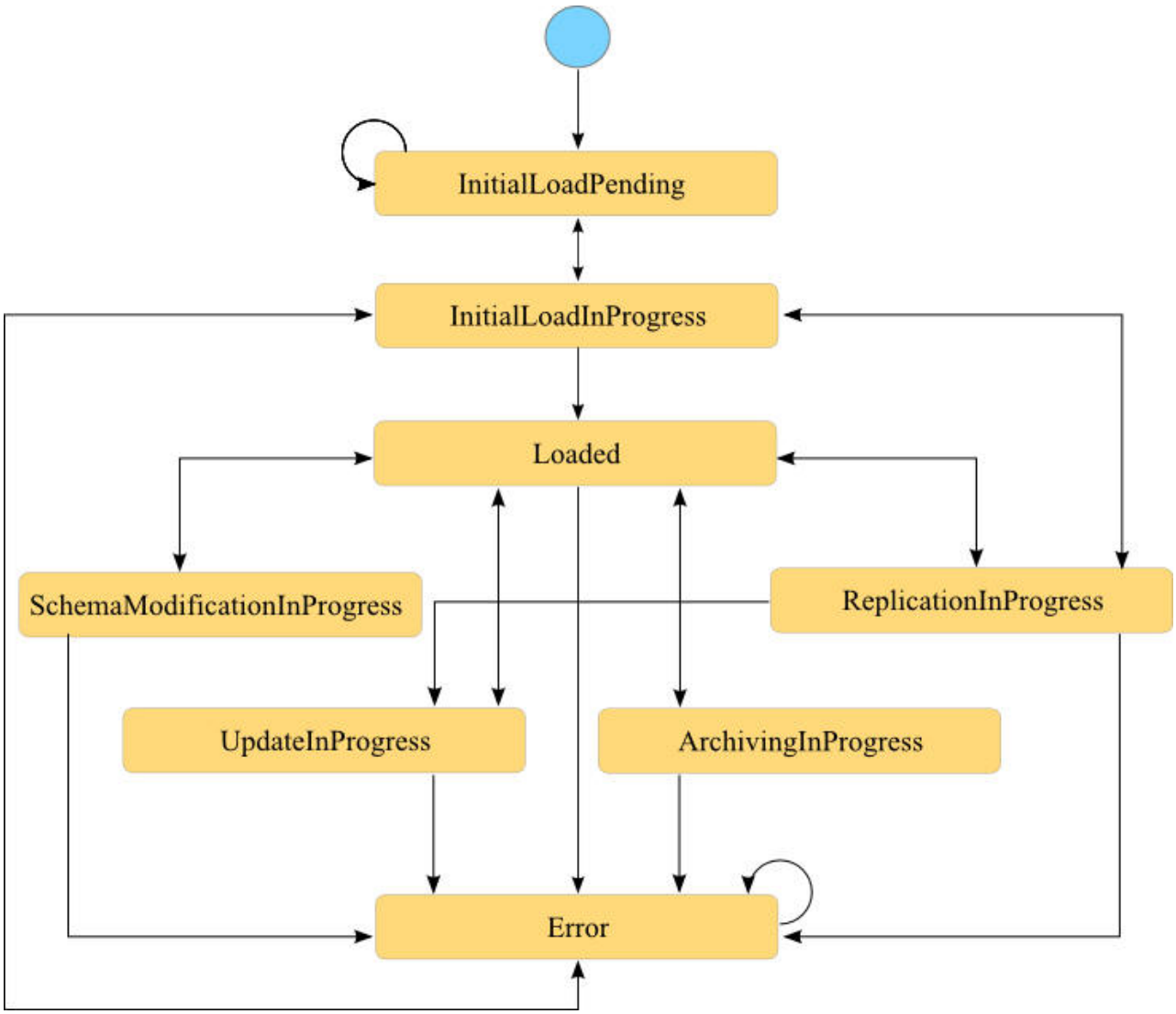


Figure 3. Possible table state transitions

Related concepts:

"Accelerator view for regular accelerators" on page 66

Moved tables or partitions

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

You can move partitions or entire tables (all partitions of a table) to an archive on an accelerator if the tables were previously defined on the accelerator (that is, if accelerator-shadow tables of these tables exist). As part of the move, the original DB2 data is deleted. The table space containing the tables or partitions in DB2 is set to a persistent (permanent) read-only state so that the original tables can no longer be updated or otherwise modified.

To be able to recover the data in an emergency case, an image copy is also created as part of the move. The recovery function of the High-Performance Storage Saver

uses the image copies to restore the data to its original location in DB2. The High Performance Storage Saver is mainly used for historical data that is no longer actively used or maintained.

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

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

Related concepts:

"Freeing up storage in DB2 for z/OS" on page 149

Accelerator-only tables and in-database transformation

Accelerator-only tables are tables that do not originate from DB2 base tables. Their data exists only on an accelerator. Data in accelerator-only tables can only be modified by data-manipulation (DML) statements (INSERT, UPDATE, and DELETE). With accelerator-only tables, you can do in-database transformation (IDT) in a more efficient way.

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

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

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

To create an accelerator-only table, you submit a `CREATE TABLE` statement in DB2 that uses the `IN ACCELERATOR <acceleratorname>` clause. Accelerator-only tables are registered in the DB2 catalog with a type of `'D'`.

Accelerator-only tables cannot be loaded by the SYSPROC.ACCEL_LOAD_TABLES stored procedure or by the corresponding function in IBM DB2 Analytics Accelerator Studio. You have to use `INSERT` statements. These can be submitted from interactive client interfaces (SPUFI, IBM Data Studio), or from client application programs using ODBC, JDBC or static SQL. It is possible to transfer many rows in a single batch or transaction (multi-row insert). However, the multi-row insert support is limited to client application programs and cannot be used interactively.

Important:

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

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

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

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

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

To accelerate the steps above, create table T1 in step 1 as an accelerator-only table. This accelerates the final (second) query and avoids the movement of data in DB2, which occupies considerable processing resources if T1 is large. If T1 is an accelerator-only table, the INSERT statement in step 2 is run on the accelerator rather than in DB2. You also omit the duplication of table T1 in DB2 because you do not need to run IBM DB2 Analytics Accelerator stored procedures to define, load, and enable table T1 on the accelerator. This is a big advantage because many extract-transform-and-load (ETL) tools that generate data transformation statements automatically do not support stored procedure calls. Using accelerator-only tables, you have to modify the statements generated by these tools only slightly to end up with accelerated in-database transformation steps that do not require stored procedure calls.

Before running the following modified steps, you must define, load and enable acceleration for table T0 on the accelerator by using the IBM DB2 Analytics Accelerator stored procedures or the corresponding functions in IBM DB2 Analytics Accelerator Studio. This is just a one-time preparation step. Then you continue with the following steps:

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

Notes:

1. Depending on your situation, you might want to keep the accelerator-only table for later use and run step 4 at a later time.
2. In contrast to temporary tables, accelerator-only tables are much more like normal DB2 tables. It is therefore recommended that you create these tables in an authorized database by using a `CREATE TABLE` statement with the `IN DATABASE` clause. Otherwise, accelerator-only tables are created in an implicit database, just like normal DB2 tables. You might also want to add an explicit schema qualifier to the table name.

The following examples consist of extended data transformations that use more than one interim table. They show how to change conventional DB2 statements to arrive at an accelerated solution to in-database-transformation. The left column shows the conventional approach. The right column shows the in-database-transformation steps that include accelerator-only tables.

Example 1

The DB2 tables T1, T2 and T3 already exist. The conventional approach uses two declared global temporary tables to store the interim results.

Table 3. Example 1: Declared global temporary tables versus accelerator-only tables

	Conventional approach (DB2)	**Accelerator-only tables**
One-time preparation	N/A	Define and load the tables T1, T2 and T3 on accelerator ACCEL1 by using stored procedures or the corresponding functions in IBM DB2 Analytics Accelerator Studio.

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

Example 2

The DB2 tables T1 and T2 already exist. The final transformation result is stored in the existing DB2 table T3. The conventional approach uses two declared global temporary tables to store the interim transformation results.

Table 4. Example 2: Declared global temporary tables versus accelerator-only tables

	Conventional approach (DB2)	**Accelerator-only tables**
One-time preparation	N/A	Define and load the tables T1 and T2 on accelerator ACCEL1 by using stored procedures or the corresponding functions in IBM DB2 Analytics Accelerator Studio.

Table 4. Example 2: Declared global temporary tables versus accelerator-only tables (continued)

	Conventional approach (DB2)	Accelerator-only tables

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

Notes:

- Bear in mind that alternative constructs, such as declared global temporary tables, which can exist in DB2, but not on an accelerator, cannot be accelerated by IBM DB2 Analytics Accelerator.
- It not possible to create image copies of accelerator-only tables because these tables do not hold any data in DB2. You can only recover the data from their original sources, which is why accelerator-only tables should be used only for temporary data.
- An accelerator-only table can reside on just a single accelerator. Copies on other accelerators cannot exist (no high availability).
- For information on how to create an accelerator-only table using the `CREATE TABLE` command or the restrictions governing the use of these tables, follow the **Related information** link at the end of this topic.
- When you submit a `CREATE TABLE` statement to create an accelerator-only table, the statement is committed in DB2 for z/OS before the corresponding action is completed on the accelerator. That is, there is a brief period during which the table is listed in the DB2 catalog, but not in the accelerator catalog. If you run the SYSPROC.ACCEL_GET_TABLES_INFO stored procedure during that time, the integrity status `ViolatedOnAccelerator` is reported for that table in the result set. (In IBM DB2 Analytics Accelerator Studio, you see the error icon next to the table name in the **Tables** section of the Accelerator view. If you move the mouse pointer over the icon, the integrity status is displayed in a fly-over help window.)

Related concepts:

"Configuring DB2 for z/OS for the acceleration of dynamic SQL queries" on page 21

"Configuring DB2 for z/OS for the acceleration of static SQL queries" on page 23

Related reference:

"Restrictions of accelerator-only tables" on page 48

Appendix B, "Impact of special register, ZPARM, or bind option settings on accelerator-only tables," on page 205

Related information:

DB2 for z/OS 12: CREATE TABLE statement

DB2 for z/OS 12: INSERT statement

Batch size limit when using DB2 Universal JDBC Driver is 32K due to limitation of DRDA architecture

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 92

IBM DB2 Analytics Accelerator for z/OS system table entries

When you select tables for query acceleration, information about these tables is added to the IBM DB2 Analytics Accelerator for z/OS system tables in the DB2 for z/OS catalog of the connected subsystem. Each entry (system table row) serves as a link between DB2 and the table on the accelerator. An entry contains, among other information, the table name.

Chapter 3. Conditions for query routing to an accelerator

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

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

1. The accelerator is in an operational mode and has been started.
2. DB2 for z/OS has been configured so that the query will be routed to a connected accelerator provided that this query meets all other conditions.
3. The accelerator supports the SQL expressions that are used in the query.
4. All data that is referenced by the query, such as the contents of tables and views, is available on one and the same accelerator. To satisfy this condition, the accelerator must have been loaded with the data in question.

Related information:

DB2 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. This also applies to queries against accelerator-only tables.

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

1 (NONE)
No routing of dynamic SQL queries to an accelerator. Queries will be processed by DB2 for z/OS only (inhouse query processing).

2 (ENABLE)
A dynamic SQL query will be routed to an accelerator if it fulfills all required conditions. An incoming query is tested against a set of heuristics, which include the table size and a response time estimate based on cost information from the SYSIBM.DSN_PROFILE_ATTRIBUTES table. Both tests ensure that a query will only be routed to an accelerator if the query can be expected to run faster than in DB2 for z/OS. However, if an error occurs while the query is being processed by the accelerator, DB2 for z/OS will return a negative SQLCODE to the application and query processing will stop.

3 (ENABLE WITH FAILBACK)
Dynamic queries are accelerated only if DB2 for z/OS determines that it is advantageous to do so. If an accelerator returns an error during the PREPARE phase or when first opening (OPEN) the query, the query is processed by DB2 for z/OS rather than sent to the accelerator. If the accelerator returns an error during a FETCH operation or a subsequent OPEN operation, DB2 for z/OS returns an error to the user and the query ends abnormally.

© Copyright IBM Corp. 2009, 2017

4 (ELIGIBLE)
Dynamic queries are accelerated if they are eligible for acceleration. DB2 for z/OS does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are executed by DB2 for z/OS. If an accelerator fails while a query is running, or if the accelerator returns an error, DB2 for z/OS returns a negative SQL code to the application.

5 (ALL)
A dynamic query will always be routed to an accelerator, no matter if it fulfills the conditions or not. If processing cannot start or continue because an incoming query fails to fulfill all the conditions for accelerated query processing, DB2 for z/OS returns a negative SQLCODE to the application and query processing ends abruptly. That is, the query will not be processed at all.

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

Exceptions: If the QUERY_ACCELERATION ZPARM is set to the value 5 (`SET CURRENT QUERY ACCELERATION = ALL`), the following dynamic queries are always processed by DB2 for z/OS, regardless of the setting of 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 and no value in the ACCESSTYPE 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 apply to cursor queries and the SELECT portion of the SQL INSERT from SELECT statement. Since product version 4.1 PTF-3, the SQL SELECT INTO statement can also 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 5. Bind options for static SQL queries

QUERYACCELERATION	GETACCELARCHIVE
Binds static SQL queries for execution on an accelerator	Includes or excludes partition data that was archived by the High Performance Storage Saver (HPSS) from accelerated static SQL queries

Table 5. Bind options for static SQL queries (continued)

QUERYACCELERATION	GETACCELARCHIVE
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 archived partition data from the accelerator. If the static query is not bound for acceleration, it will be be bound to run in DB2. If the static query is bound for acceleration because the bind option **QUERYACCELERATION** was specified, the query is routed to the accelerator when the application is run, but the query will only retrieve *active* data from the accelerator and ignore any HPSS data. **YES** If the bind option QUERYACCELERATION is also specified and the static query references a table from which data has been archived by the HPSS, one of the following courses of action is taken: • If the static query satisfies the acceleration criteria (as specified by QUERYACCELERATION), the query is bound for acceleration, and when the application is run, the query includes the archived data in the query. • If the static query cannot be bound for acceleration because it does not satisfy the criteria, DB2 fails to complete the BIND or REBIND PACKAGE operation and returns an error message for that query.

Table 5. 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

Important:
- 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.
- If you submit a CREATE, DROP, or COMMIT statement against an accelerator-only table together with an INSERT, UPDATE, DELETE, or SELECT statement (against the same table) in the same thread, the table does not come into existence during the bind phase. However, the current implementation requires that all referenced tables are defined and enabled for query acceleration during the bind phase.

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

The bind 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.

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

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

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

The 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 data types

See this overview of supported DB2 data types, corresponding Netezza types, ranges, and hints at restrictions. The section also lists the unsupported data types at the end.

Numeric data types

Table 6. Numeric data types

DB2 type	Netezza type	Range DB2	Range Netezza	Comment
SMALLINT	SMALLINT (INT2)	-32768 to +32767	-32768 to +32767	
INTEGER	INTEGER (INT, INT4)	-2147483648 to +2147483647	-2147483648 to +2147483647	
BIGINT	BIGINT (INT8)	-9223372036854775808 to +9223372036854775807	-9223372036854775808 to +9223372036854775807	
REAL	REAL	~ $-7.2 * 10^{+75}$ to $-5.4 * 10^{-79}$ or ~ $+5.4 * 10^{-79}$ to $+7.2 * 10^{+75}$	~ $-3.4 * 10^{+38}$ to $-1.18 * 10^{-38}$ or ~ $1.18 * 10^{-38}$ to $+3.4 * 10^{+38}$	For details, see "Differences in floating-point error tolerances" on page 56
DOUBLE	DOUBLE	~ $-7.2 * 10^{+75}$ to $-5.4 * 10^{-79}$ or ~ $+5.4 * 10^{-79}$ to $+7.2 * 10^{+75}$	~ $-1.79 * 10^{+308}$ to $-3.40 * 10^{-308}$ or ~ $+3.4 * 10^{-308}$ to $+1.79 * 10^{+308}$	For details, see "Differences in floating-point error tolerances" on page 56
DECIMAL(*precision*, *scale*)	NUMERIC(*precision*, *scale*)	Max. precision 31	Max. precision 38	

Date, time, and timestamp data types

Table 7. Date, time, and timestamp data types

DB2 type	Netezza type	Range DB2	Range Netezza	Comment
DATE	DATE	0001-01-01 to 9999-12-31	0001-01-01 to 9999-12-31	
TIME	TIME	00:00:00.000000 to 24:00:00.000000	00:00:00 to 23:59:59 or 00:00:00 to 24:00:00	The precision must be six fractional digits. You can set a ZPARM to truncate values with precision 12. See *Setting ZPARMs for IBM DB2 Analytics Accelerator in DB2 11 for z/OS* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

Table 7. Date, time, and timestamp data types (continued)

DB2 type	Netezza type	Range DB2	Range Netezza	Comment
TIMESTAMP	TIMESTAMP	**Date portion:** 0001-01-01 to 9999-12-31 **Time portion:** 00:00:00.000000 to 24:00:00.000000	**Date portion:** 0001-01-01 to 9999-12-31 **Time portion:** 00:00:00.000000 to 24:00:00.000000 or 00:00:00.000000 to 23:59:59.999999	

String data types

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

Table 8. Supported combinations of CCSIDs and subtypes

CCSID	Allowed character subtypes	Comment
ASCII (for example 819; there are numerous others)	SBCS	
EBCDIC (for example 37; there are numerous others)	SBCS MIXED (partially supported) DBCS (graphic, partially supported)	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*
UNICODE (367, 1208, 1200)	SBCS (367) MIXED (1208) DBCS (1200)	

Table 9. String data types

DB2 type	Netezza type	Supported CCSIDs	Comment
CHAR FOR SBCS DATA	CHAR	ASCII EBCDIC UNICODE (367)	
VARCHAR FOR SBCS DATA	VARCHAR	ASCII EBCDIC UNICODE (367)	
CHAR FOR MIXED DATA	NVARCHAR	UNICODE (1208) EBCDIC (partially supported)	
VARCHAR FOR MIXED DATA	NVARCHAR	UNICODE (1208) EBCDIC (partially supported)	

Table 9. String data types (continued)

DB2 type	Netezza type	Supported CCSIDs	Comment

CHAR FOR BIT DATA	CHAR	EBCDIC	EBCDIC tables that were added and loaded with product version 4.1 PTF-2 or later support columns with a data type of CHAR FOR BIT DATA. This means that queries can reference CHAR FOR BIT DATA columns in EBCDIC tables that were added with version 4.1 PTF-2 or a later version.
			Restriction: A query that references a FOR BIT DATA column can reference other columns that use supported data types, but only if these columns contain single-byte encoded (SBCS) data. That is, none of the other columns must contain MIXED data. Assuming that `EBCDIC_MIXED_COL` is a MIXED data column and `EBCDIC_FOR_BIT_COL` a FOR BIT DATA column in an EBCDIC table, a query of the following type cannot be accelerated:
			`SELECT .. FROM TABLE`
			`WHERE`
			`EBCDIC_MIXED_COL = 'abc' AND`
			`EBCDIC_FOR_BIT_COL = X'C1C2';`
			
			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.

VARCHAR FOR BIT DATA	VARCHAR	EBCDIC	See the comment on VARCHAR FOR BIT DATA. It also applies here.
GRAPHIC	NVARCHAR	UNICODE (1200) EBCDIC (partially supported)	Normally, this type is supported for UNICODE only. To enable this type for EBCDIC, include the value 1 when setting QUERY_ACCEL_OPTIONS.
VARGRAPHIC	NVARCHAR	UNICODE (1200) EBCDIC (partially supported)	Normally, this type is supported for UNICODE only. To enable this type for EBCDIC, include the value 1 when setting QUERY_ACCEL_OPTIONS.

Unsupported DB2 data types

- BINARY
- BLOB
- CLOB
- DBCLOB

- DECFLOAT
- Distinct (user-defined) data types
- ROWID
- VARBINARY
- XML

Related concepts:

"Conditions that prevent query routing to an accelerator" on page 35

Supported SQL functions and expressions

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 and the CAST specification.

Supported scalar functions

Table 10. Supported scalar functions and specifications

Function name	Function name	Function name	Function name
• ABS	• DECIMAL	• LTRIM (also with two arguments)	• RTRIM (also with two arguments)
• ADD_MONTHS	• DECODE (also with two arguments)	• MAX (also GREATEST; see Table 11 on page 32)	• SECOND
• BIGINT	• DEGREES	• MEDIAN (see Table 11 on page 32)	• SIGN (see Table 11 on page 32)
• BITAND	• DIGITS	• MICROSECOND	• SMALLINT
• BITANDNOT	• DOUBLE_PRECISION (also DOUBLE, FLOAT)	• MIDNIGHT_SECONDS	• SPACE
• BITNOT	• EXP	• MIN (also LEAST; see Table 11 on page 32)	• SQRT
• BITOR	• EXTRACT	• MINUTE	• STRIP
• BITXOR	• FLOAT	• MOD	• SUBSTR (see Table 11 on page 32)
• CEILING	• FLOOR	• MONTH	• SUBSTRING (see Table 11 on page 32)
• CAST specification	• HOUR	• MONTHS_BETWEEN	• TIME
• CHAR (see Table 11 on page 32)	• IFNULL	• NEXT_DAY	• TIMESTAMP (see Table 11 on page 32)
• CHARACTER_LENGTH (see Table 11 on page 32)	• INTEGER (INT)	• NULLIF	• TIMESTAMPDIFF
• COALESCE	• JULIAN_DAY	• NVL (Rewriting to COALESCE no longer required. Also supported with two arguments.)	• TIMESTAMP_FORMAT
• CONCAT	• LAST_DAY (see Table 11 on page 32)	• POSITION (see Table 11 on page 32)	• TIMESTAMP_ISO (see Table 11 on page 32)
• CORR	• LCASE	• POSSTR (see Table 11 on page 32)	• TRANSLATE (see Table 11 on page 32)
• COVAR	• LEFT (see Table 11 on page 32)	• POWER	• TRIM (also with two arguments)

Table 10. Supported scalar functions and specifications (continued)

Function name	Function name	Function name	Function name
• COVAR_SAMP	• LENGTH (see Table 11)	• QUARTER	• TRUNCATE (see Table 11)
• DATE	• LN	• RADIANS	• UCASE
• DAY (see Table 11)	• LOCATE (see Table 11)	• REAL	• UPPER (see Table 11)
• DAYOFMONTH	• LOCATE_IN_STRING (see Table 11)	• REPEAT	• VALUE
• DAYOFWEEK	• LOG10	• REPLACE (see Table 11)	• VARCHAR (see Table 11)
• DAYOFWEEK_ISO	• LOG	• RIGHT (see Table 11)	• VARCHAR_FORMAT
• DAYOFYEAR	• LOWER (see Table 11)	• ROUND	• WEEK_ISO
• DAYS	• LPAD (see Table 11)	• RPAD (see Table 11)	• YEAR

Table 11. Restrictions

Function name	Restriction
• BITAND • BITANDNOT • BITNOT • BITOR • BITXOR	Arguments must be of the type BIGINT, INT, or SMALLINT.
• CHAR	• If CHAR is used for character-to-character or graphic-to-character conversions, one of the arguments in the expression must be of type CODEUNIT16 or CODEUNIT32. • If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters.
• CAST specification • CHARACTER_LENGTH • LEFT • LENGTH • LOCATE • LOCATE_IN_STRING	Enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3 if one of these conditions applies: • One of the arguments is of the type CODEUNITS16 or OCTETS. • One of the arguments is a string encoded in UTF-8 or UTF-16. Mind that result deviations can occur if the input string contains multibyte characters. **Important:** LEFT is supported for character and graphic strings only.
• DAY	The argument must not be of the type INTEGER or FLOAT.
• LAST_DAY	The argument must not be a timestamp.
• LENGTH	The argument must be of type CHAR or VARCHAR. Numeric types, time values, date values, or timestamps cannot be used.
• LOWER (LCASE)	• If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters. • When a query is accelerated, the value of the locale argument is ignored and `en_US` is used instead.
• LPAD	If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters.
• MAX • MIN	Arguments of this function must be of the type BIGINT, INT, or SMALLINT. In addition, no more than four arguments are allowed. If these criteria are not met, the query or operation is not accelerated.

Table 11. Restrictions (continued)

Function name	Restriction
• MEDIAN	This function is fully supported by DB2 11 for z/OS, but not by DB2 10 for z/OS. In connection with DB2 10 for z/OS, you can only run the MEDIAN function on an accelerator. This means that a query including this function will inevitably fail if the accelerator becomes unavailable because failback processing is not possible.
• POSITION	If one of the arguments of this function is of the type CODEUNITS16 or OCTETS, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3.
• POSSTR • REPLACE	If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters.
• RIGHT	Enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3 if one of these conditions applies: • One of the arguments is of the type CODEUNITS16 or OCTETS. • One of the arguments is a string encoded in UTF-8 or UTF-16. Mind that result deviations can occur if the input string contains multibyte characters.
• RPAD • SUBSTR	If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters.
• SIGN	Arguments must not be of the data type DECFLOAT.
• SUBSTRING	SUBSTRING is supported only if the expression refers to character or graphic data types. To enable support for graphic data types, set the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3.
• TIMESTAMP	The following restrictions with regard to the formats of string literals that can be used as timestamps do not apply if Netezza Platform Software (NPS®) 7.2 or later is used. However, they do apply if you still use an older version of NPS. That is, string literals in the following formats are not supported as arguments of the TIMESTAMP function: • String literals with a space character ( ) or dash (-) as the separator (delimiter) between the date and the time portion and a period (.) or colon (:) to separate hours, minutes, and seconds: **Examples:** – `yyyy-mm-dd[ hh.mm.ss[.nnnnnn]]` (space character before `hh`; period to separate hours, minutes, and seconds) – `yyyy-mm-dd[-hh:mm:ss[.nnnnnn]]` (dash before `hh`; colon to separate hours, minutes, and seconds) • String literals with the capital letter T as the separator (delimiter) between the date and the time portion and a period (.) or colon (:) to separate hours, minutes, and seconds: – `yyyy-mm-dd[Thh.mm.ss[.nnnnnn]]` (`T` before `hh`; period to separate hours, minutes, and seconds) – `yyyy-mm-dd[Thh:mm:ss[.nnnnnn]]` (`T` before `hh`; colon to separate hours, minutes, and seconds)
• TIMESTAMP_ISO	The argument must not be of the type CHAR or VARCHAR.
• TRANSLATE	If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters.
• TRUNCATE	You cannot use the TRUNCATE function on accelerator-only tables.

Table 11. Restrictions (continued)

Function name	Restriction
• UPPER (UCASE)	• If one of the arguments of this function is a string encoded in UTF-8 or UTF-16, enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3. Mind that result deviations can occur if the input string contains multibyte characters. • When a query is accelerated, the value of the locale argument is ignored and `en_US` is used instead.
• VARCHAR	Enable support by setting the QUERY_ACCEL_OPTIONS parameter (a ZPARM) to the value 3 if one of these conditions applies: • If VARCHAR is used for character-to-varchar or graphic-to-varchar conversions, and one of the arguments of the function is of the type CODEUNIT16 or OCTETS. • One of the arguments is a string encoded in UTF-8 or UTF-16. Mind that result deviations can occur if the input string contains multibyte characters.

Unsupported scalar functions

Table 12. 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>	• TOTALORDER
• DECFLOAT_SORTKEY	• NORMALIZE_DECFLOAT	• VARBINARY
• DIFFERENCE	• NORMALIZE_STRING	• VARGRAPHIC
• DECRYPT <xxx>	• OVERLAY	• WEEK
• ENCRYPT_TDES	• QUANTIZE	• <xxx>XML<xxx>
• GENERATE_UNIQUE	• RAISE_ERROR	

Supported special registers

IBM DB2 Analytics Accelerator for z/OS supports most DB2 special registers. During SQL routing to an accelerator, these special registers are replaced with the equivalent Netezza functions or with a literal string containing the register's value at the time of statement execution. In some cases, this leads to different query results.

The following special registers are supported by Netezza function substitution:

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.

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

Conditions that prevent query routing to an accelerator

DB2 for z/OS does not route a query to an accelerator if any of the following conditions applies.

- You use a distributed data facility (DDF) client, such as an Open Database Connectivity (ODBC) client, that does not support *level 8* of the IBM Distributed Relational Database Architecture™ (DRDA).
- The encoding scheme of an SQL statement is `M` because the tables use different encoding schemes, or the query contains an expression that explicitly relates to a coded character set identifier (CCSID), for example, a cast specification with a CCSID option.
- The FROM clause of the query specifies a data-change-table-reference, or, in other words, the query is selected from a FINAL TABLE or from an OLD TABLE.
- The query contains a 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 Platform Software (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>)`
- The query contains a string expression (including columns) with an unsupported subtype. See Table 8 on page 29.
- The query contains an expression with an unsupported result data type or subtype. See “Supported data types” on page 27.
- The query refers to a column that uses a field procedure (FIELDPROC).
- The query uses a special register other than:
  - CURRENT DATE
  - CURRENT TIME
  - CURRENT TIMESTAMP
- A LOCAL date format is used, that is, one or more of the following conditions apply:
  1. The query contains a CHAR function in which LOCAL is specified as the second argument.

2. The query contains a date or time expresssion and the DATE FORMAT field of the DSNTIP4 installation panel specifies LOCAL.
3. Application programs that process SQL on DB2 for z/OS have been precompiled with the DATE(LOCAL) option.

Exception: You can enable LOCAL date expressions in the format `dd/mm/yyyy` or `yyyymmdd` (with two space characters at the end) by including the `QUERY_ACCEL_OPTIONS` ZPARM value `4` or `7`. For more information, see *Installing DB2 libraries with IBM DB2 Analytics Accelerator for z/OS support* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

- The query contains a sequence expression (NEXTVAL or PREVVAL).
- The query contains a user-defined function (UDF)
- The query contains a ROW CHANGE expression
- 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 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
  - CHARACTER_LENGTH
  - LEFT
  - LENGTH
  - 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 *scalar* MIN or MAX function with more than four arguments or the result of the scalar 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 Table 12 on page 34
- The instruction to process "SELECT FROM ... INSERT INTO ... " statements (ZPARM QUERY_ACCEL_OPTIONS includes the value 2) is ignored if the target table uses an encoding scheme that is different from the encoding scheme of the tables in the SELECT statement.
- A *static* SQL query that includes a `FOR SYSTEM_TIME ...` or `FOR BUSINESS_TIME ...` is not routed to an accelerator if one of the following bind options (value YES, default) is used.

Table 13. Unsupported bind options

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

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

Related concepts:

"EXPLAIN information" on page 107

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. An accelerator-shadow table is thus a copy of a projection of a DB2 table. In most cases, the projection is the DB2 table itself. Columns of the original DB2 table are excluded from the projection only if a column uses an unsupported data type.

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

The update of accelerator-shadow tables is an asynchronous process, that is, there is no guarantee that the data in the DB2 tables and the data in the accelerator-shadow tables are always in sync. 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 accelerator-shadow tables immediately before you run the query and choose an appropriate lock scope so that the affected DB2 tables cannot be altered during the update. This, however, still does not guarantee that the results of an accelerated query will show the latest changes because the lock might prevent or postpone an update of the original DB2 table data.

If you use the incremental update function, you can ensure that committed DB2 for z/OS changes are propagated to the accelerator by calling the ACCEL_CONTROL_ACCELERATOR stored procedure with the **<waitForReplication>** command.

If the most recent table changes must be reflected in the result set, set the CURRENT QUERY ACCELERATION special register to the value NONE. This means that the query is executed by DB2 and will not be accelerated.

Related concepts:

"Updating accelerator tables continually" on page 136

Related tasks:

"Loading tables" on page 93

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.

Correlated subqueries

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.

Related information:

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

COUNT_BIG function

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.

Data types

See how IBM DB2 Analytics Accelerator handles certain DB2 data types:

CHAR (n)

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.

FLOAT

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

Related concepts:

"Differences in floating-point error tolerances" on page 56

Division and multiplication of decimal values

During the division or multiplication of decimal values, DB2 for z/OS might truncate operand values if at least one of the operands is defined with a precision greater than 15. If that happens, the result of the calculation will be imprecise or wrong. IBM DB2 Analytics Accelerator 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 information:

Arithmetic with two decimal operands

Encoding issues

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

Unicode SQL statements

Generally, when a UNICODE-encoded SQL statement contains characters or strings (as a literal, host variable, or parameter marker) that refer to a single-byte-encoded EBCDIC column, then DB2 tries to convert these strings to EBCDIC. Sometimes, such a string cannot be converted properly because it contains invalid UNICODE characters or characters that do not exist in the target EBCDIC code page. In such cases, the results of the SQL operation and the SQLCODE returned by the accelerator might be different from those returned by DB2 for z/OS with regard to the following aspects:

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

Mixed data and parameter markers

If a DB2 subsystem is set up to contain mixed data (`mixed=yes`), that is, if differently encoded data is allowed to coexist, and the coded character set identifier (CCSID) is not specified for parameter markers in a query, the query might return incorrect results.

Multibyte and double-byte EBCDIC columns

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.

Exception handling

When an exception, such as a buffer overflow, a division by zero, or an out-of-range value occurs, DB2 returns the value NULL combined with a `+802` warning, provided that the faulty expression is in the outermost SELECT statement and affects only one row. This is also called *friendly arithmetic*. IBM DB2 Analytics Accelerator for z/OS will always issue an error in cases like these.

Maximum row length

The maximum row length for a table on an accelerator is 65535 bytes. If the combined length of all column fields in a DB2 table exceeds this limit, you cannot define that table on an accelerator.

Predicate evaluation

DB2 tolerates SQL errors during predicate evaluation as long as the error has no impact on the result of the evaluation. IBM DB2 Analytics Accelerator 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 14 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 14. 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)

Table 14. Relevant combinations of two-term predicates (continued)

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

The reason for choosing this evaluation method is that DB2 sometimes changes the order of term evaluation internally. The change is not announced or reported, and therefore goes unnoticed. Depending on the order of term evaluation, DB2 returns an error in one case, but not the other.

The behavior is very similar for the CASE and DECODE functions. These are basically SQL implementations of IF ... THEN ... ELSE constructs and exist in DB2 and IBM DB2 Analytics Accelerator. DB2 will ignore a wrongly defined option as long as it can select a correct one that evaluates to true. IBM DB2 Analytics Accelerator, on the other hand, will first check all available options, and return an error if one of these is not correctly defined. In the latter case, it does not matter whether selecting one of the correctly defined options would have led to a correct result.

Scalar functions

See how the behavior of certain scalar functions deviates from the behavior of these functions in DB2 for z/OS.

DATE(numeric_exp)

See DATE(numeric_exp) scalar function.

LEFT

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.

LOCATE

The DB2 scalar functions LOCATE and LOCATE_IN_STRING are converted to the INSTR function. 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.

LOCATE_IN_STRING

See LOCATE.

LOWER

- 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.

- IBM DB2 Analytics Accelerator for z/OS does not support string functions like UPPER() and LOWER() for strings with termination characters (`0x00`) in the following constructs:
  - Literals
  - CHAR columns
  - VARCHAR columns
  - GRAPHIC columns
  - VARGRAPHIC columns

  The resulting strings are truncated. The truncation starts at the first termination character.

MONTHS_BETWEEN

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.

NEXT_DAY

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.

ROUND

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.

SUBSTR

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.

TRANSLATE

- See LOWER.
- 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.

TRUNC

See ROUND.

UPPER

See LOWER.

SQL issues

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

Time and date values

Time and date constructs in the Netezza database often differ from the constructs used by DB2 for z/OS, so that values and expressions must be converted. The conversion is often accompanied by limitations.

24:00:00 conversion

This product supports the DB2 for z/OS time representation 24.00.00 for midnight (end of day) in TIME and TIMESTAMP columns or SQL expressions. There is no need to set up a conversion.

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 into an accelerator-shadow table.

Related tasks:

"Switching from 24:00:00 conversion to native 24:00:00 support" on page 175

DATE(numeric_exp) scalar function

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.

Dates in TIMESTAMP expressions

If a date is supplied as the only argument to a TIMESTAMP expression, for example `TIMESTAMP(DATE('2015-12-31'))`, DB2 for z/OS returns a timestamp with a time portion that shows no fractional seconds, whereas IBM DB2 Analytics Accelerator for z/OS returns fractional seconds with a precision of six digits:

DB2 for z/OS	`2015-12-31 00:00:00`
IBM DB2 Analytics Accelerator	`2015-12-31 00:00:00.000000`

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

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

Formats

- 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 queries 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
- 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.

Note: This restriction no longer exists with Netezza Platform Software (NPS) 7.2 or later.

- 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.

Strings and their conversion

For some scalar functions, such as TIME, DB2 accepts a time value, a timestamp, and also a string representation of times and timestamps. When DB2 evaluates the scalar function, it first checks whether a value can be interpreted as a time. If this fails, it tries to interpret the value as a timestamp. The corresponding accelerator functions do not accept string values. During query conversion, the content of a string is not evaluated. Instead, it is always cast to a time value. If the string represents a valid time format on the accelerator, IBM DB2 Analytics Accelerator 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

Incompatibilities related to IBM InfoSphere Data Replication for z/OS

You might encounter execution errors or deviating query results when you use the incremental update function. Some of these errors or result deviations are due to incompatibilities between DB2 for z/OS, IBM DB2 Analytics Accelerator for z/OS, and IBM InfoSphere® Change Data Capture for z/OS. The following section deals with these incompatibilities.

IBM InfoSphere Data Replication for z/OS is the main software component behind the incremental update function of IBM DB2 Analytics Accelerator for z/OS.

All incompatibilities that are listed in the section *Incompatibilities between DB2 for z/OS and IBM DB2 Analytics Accelerator for z/OS* also apply to IBM InfoSphere Data Replication 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 Data Replication for z/OS. However, the following exceptions exist when IBM InfoSphere Data Replication for z/OS transfers data to IBM DB2 Analytics Accelerator for z/OS:

- Values of VARCHAR FOR MIXED and VARGRAPHIC columns that exceed 16000 characters are cut off when they are received by IBM DB2 Analytics Accelerator for z/OS. The difference to DB2 for z/OS is that in the case of IBM InfoSphere Data Replication 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 Data Replication 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.

Restrictions of accelerator-only tables

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

Current limitations

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

Data-definition (DDL) statements

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

- The `IN ACCELRATOR` clause can be used only once in a statement.
- A `CREATE TABLE` or `ALTER TABLE` statement must not define a Unicode column in an EBCDIC table if that table is an accelerator-only table.
- A `CREATE TABLE` must not contain a materialized query definition if the table is an accelerator-only table.
- A `CREATE MASK` statement must not define a mask for an accelerator-only table.
- A `CREATE PERMISSION` statement must not define a column permission for an accelerator-only table.
- A `DECLARE GLOBAL TEMPORARY TABLE` statement must not reference an accelerator-only table.
- A `CREATE TRIGGER` or `ALTER TRIGGER` statement must not reference an accelerator-only table in the body of the trigger.
- A `CREATE TRIGGER` or `ALTER TRIGGER` statement with an `INSTEAD OF` clause must not reference a view that references an accelerator-only table in the body of the trigger.
- An `ALTER TABLE` statement is not allowed if the table is an accelerator-only table.
- An `ALTER TABLESPACE` statement is not allowed if the table space contains accelerator-only tables.
- A `DROP TABLESPACE` statement fails if the table space contains accelerator-only tables. You must first drop the accelerator-only tables by using a `DROP TABLE` statement.
- The `DROP DATABASE` statement does not behave as expected:
  - If you create an accelerator-only table without specifying a database, the table is created in the default database as usual, but a `DROP DATABASE` statement does not remove that database.
  - A `DROP DATABASE` statement does not remove a database and the tables therein if some of the tables are accelerator-only tables. To remove such a database, first remove all accelerator-only tables by using separate `DROP TABLE` statements, and then the database.

- A `RENAME TABLE` statement fails if the table is an accelerator-only table.
- A `LOCK TABLE` statement must not refer to an accelerator-only table.
- A `CREATE INDEX` statement fails if the table to be indexed is an accelerator-only table.
- A `CREATE SYNONYM` statement fails if the synonym is for an accelerator-only table.
- You cannot create more than one accelerator-only table in a single transaction. The following example does not work:

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

  This example, on the other hand, does work:

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

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

Table 15. Unsupported clauses and keywords

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

Other SQL statements

The following statements are not allowed:

- You cannot run an `INSERT INTO <accelerator-only table> SELECT ... FROM ...` statement if the `SELECT` part of the statement references a normal DB2 table that has no loaded counterpart on the accelerator. The statements succeeds only if the `SELECT` part references one or more tables on the same accelerator as the accelerator-only table.
- You cannot submit an `UPDATE` statement against table columns that serve as distribution keys.
- You cannot insert values directly into a decimal column of an accelerator-only table if these values are the result of a division where the dividend, the divisor, or both are decimals. However, the operation works if the division is part of the `SELECT` portion in an `INSERT INTO ... SELECT FROM ...` statement.

Restrictions on data types, column types, and code pages

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

- The column names in an accelerator-only table must not start with a tilde (~) followed by an underscore (_) character.

- You cannot have string columns of the type CHAR FOR BIT DATA or VARCHAR FOR BIT DATA in accelerator-only tables if these tables are encoded in ASCII or UNICODE. Only EBCDIC tables can have such columns.
- The CHAR FOR BIT DATA and VARCHAR FOR BIT DATA data types are not supported in accelerator-only tables if the environment has been set up for the use of mixed data (ZPARM MIXED=YES).
- You cannot define a large-object (LOB) column for an accelerator-only table.
- You cannot define a security label column for an accelerator-only table.

Stored procedures

Some of the IBM DB2 Analytics Accelerator stored procedures do not work on accelerator-only tables:

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

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

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

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

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

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

Netezza limitations

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

Different results

Results obtained by running the same query against a normal DB2 table and an accelerator-only table might differ under the following conditions:

- One of the tables that are specified in the query contains a CHAR column in Unicode or in a mixed format, and the column values are padded with blank characters.
- Decimal values with fractional digits and a total length of 31 (the maximum) are allowed in a table column that is referenced by the query. A value to be inserted has a fractional portion that is longer than the defined maximum. In this - and only in this special case - does Netezza round the fractional portion where otherwise, it would be truncated. The different treatment of such values in DB2 for z/OS and Netezza might lead to different results. For example:

```
CREATE TABLE AOT(
   N_DEC DECIMAL(31,5) NOT NULL
   ) IN ACCELERATOR A1;

INSERT INTO AOT VALUES(8999999999999999999999999.12345 5)
```

In this case, DB2 would use the value 89999999999999999999999999.12345, while the Netezza database would use 89999999999999999999999999.12346.

However, the INSERT statement would fail with a numeric data overflow error if the allowed maximum was 31 without fractional digits, but the value to be inserted was longer than that and had a fractional portion equal to or greater than 0.5, as in the following example:

```
CREATE TABLE AOT(
   N_DEC DECIMAL(31,0) NOT NULL
   ) IN ACCELERATOR A1;

INSERT INTO AOT VALUES(9999999999999999999999999999999.9)
```

The value to be inserted here has a total length of 32. Rounding it would require an extra, non-fractional digit, which is why an overflow error is returned.

New features in DB2 10 for z/OS

Some 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 DB2 tables with row access control as follows:

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

Cross-loading of tables

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

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

CURRENT EXPLAIN MODE special register

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

CURRENT EXPLAIN MODE = YES
: Enables the DB2 EXPLAIN facility and causes EXPLAIN information to be inserted into the EXPLAIN tables for eligible dynamic SQL statements after preparing and executing the statement. Dynamic SQL statements are hence compiled and executed normally.

CURRENT EXPLAIN MODE = EXPLAIN
: Enables the DB2 EXPLAIN facility and captures EXPLAIN information for eligible dynamic SQL statement after preparing the statement. Note that with setting, a dynamic statement or query will not be executed, except for the SET clauses that it might contain.

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

Although tables that keep track of system time intervals and business time intervals (system-temporal tables, business-temporal tables, and bi-temporal tables) were introduced in DB2 10 for z/OS, IBM DB2 Analytics Accelerator for z/OS supports system-temporal tables and bi-temporal tables only in connection with *DB2 11 for z/OS*. Business-temporal tables, on the other hand, are also supported with DB2 10 for z/OS. Queries referencing temporal tables usually contain the following expressions or clauses:

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

System-temporal and bi-temporal tables use columns with a timestamp data type that has a precision of 12 fractional seconds. On the accelerator, these timestamps are truncated, leaving a precision of only 6 fractional seconds. For that reason, you must enable the use of system-temporal and bi-temporal tables. To do so, set or include option 5 in the value of the QUERY_ACCEL_OPTIONS parameter or ZPARM.

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

Table 16. Unsupported bind options

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

Table 16. Unsupported bind options (continued)

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

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

Timestamps

The portion that denotes fractional seconds in timestamps can have up to twelve digits in DB2 for z/OS, which means that even picoseconds can be represented. IBM DB2 Analytics Accelerator for z/OS, on contrary, only supports timestamps with a precision up to six digits (microseconds), and can handle incoming timestamps with a precision of 12, which are truncated to a length of six digits. This behavior reflects a limitation of the underlying Netezza database. To deal with a precision of 12, you must allow the truncation of such values by setting the QUERY_ACCEL_OPTIONS parameter or ZPARM to the value `6`.

Be aware that the truncation of the fractional seconds portion has implications:

- The truncation leads to a loss of precision. Bear this in mind if timestamp precision is critical.
- A timestamp column might have been chosen as the primary key of a DB2 table because the high precision of the fractional seconds guarantees the uniqueness of the values. After a truncation, the values might not be unique anymore. Therefore, the very same column might be unsuitable as the primary key of an accelerator-shadow table.

Restrictions: The following constructs are not supported, no matter if truncation is enabled or not:

- Timestamps with timezone specifications, to the effect that:
  - SQL queries that require TIMESTAMP with a timezone specification as the result data type are not routed to an accelerator.
  - Timestamps with timezone constants are not accepted.
  - Functions and special registers that are related to timestamps with timezone specifications are not accepted.
- Fractional seconds with a precision greater than 6 in values of parameter markers
- Fractional seconds with a precision greater than 6 in CHAR columns that are referenced as part of an AS OF clause

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. This might cause a run-time error during query execution if the string column contains timestamp values with a precision other than 6.

For more information, see *Setting ZPARMs for IBM DB2 Analytics Accelerator for z/OS* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

Related concepts:

"EXPLAIN information" on page 107

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 DB2 table, a DB2 view, or another DB2 alias. It can be used to reference an object wherever that object can be referenced directly. Synonyms are similar to aliases, but are supported only for compatibility with previous releases. Synonyms will be deprecated in a future release. DB2 11 for z/OS extends the support for aliases so that these can be defined for sequence objects, and also provides support for *public aliases*. This product 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.

Starting with this version of IBM DB2 Analytics Accelerator for z/OS (version 5.1.0), grouping sets are supported.

Differences in floating-point error tolerances

For queries that involve floating-point aggregates, you can expect IBM DB2 Analytics Accelerator for z/OS to produce results that are different from those delivered by inhouse DB2 for z/OS processing.

Reason

You know that the same query processed twice by DB2 for z/OS can give you different results. This is due to the fact that the results depend on the order in which the rows or tuples are processed. It is not possible to determine the order that DB2 will choose beforehand.

In the case of IBM DB2 Analytics Accelerator for z/OS, another sort of result deviation occurs, which is caused by the different methods of representation for floating-point numbers. DB2 for z/OS uses hexadecimal floating-point (HFP) numbers, whereas IBM DB2 Analytics Accelerator for z/OS uses IEEE754 floating-point numbers.

When a query is routed from DB2 for z/OS to IBM DB2 Analytics Accelerator for z/OS, all processing takes place on the accelerator.

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

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

Important: In contrast to HFP numbers of the type REAL in DB2 for z/OS, which might range from -7.2E+75 to +7.2E+75, IEEE754 is only capable of processing REAL floating point numbers between -3.4E+38 and 3.4E+38. This means that it is not possible to load an accelerator-shadow table with REAL values outside the range of -3.4E+38 to 3.4E+38.

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

Table 17. Minimum and maximum REAL and DOUBLE values (HFP and IEEE754)

	HFP (DB2 for z/OS)		**IEEE754 (IBM DB2 Analytics Accelerator for z/OS)**	
	REAL	DOUBLE	REAL	DOUBLE
Min.	~-7.2 $*10^{+75}$	~-7.2 $*10^{+75}$	~-3.4 $*10^{+38}$	~-1.79 $*10^{+308}$
Max.	~+7.2$*10^{+75}$	~+7.2$*10^{+75}$	~+3.4 $*10^{+38}$	~+1.79$*10^{+308}$
Min. positive	~+5.4$*10^{-79}$	~+5.4$*10^{-79}$	~+1.18$*10^{-38}$	~+3.4 $*10^{-308}$
Max. negative	~-5.4 $*10^{-79}$	~-5.4 $*10^{-79}$	~-1.18 $*10^{-38}$	~-3.4 $*10^{-308}$

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:

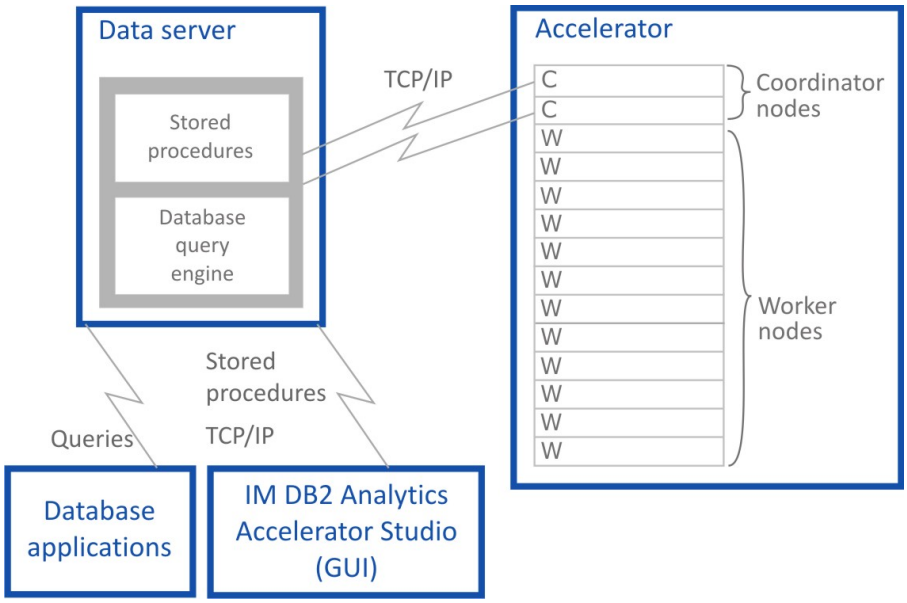


Figure 4. IBM DB2 Analytics Accelerator for z/OS setup

© Copyright IBM Corp. 2009, 2017

Chapter 5. Interface elements of IBM DB2 Analytics Accelerator Studio

Here is a brief description of the interface elements that show IBM DB2 Analytics Accelerator Studio functionality.

Note: When you start IBM DB2 Analytics Accelerator Studio for the first time, you see the Welcome screen. To see the main window of IBM DB2 Analytics Accelerator Studio with the principal functions and controls, just close the Welcome screen by clicking the cross-shaped icon on the tab of the Welcome screen (usually positioned on the upper left).

Menu bar

When IBM DB2 Analytics Accelerator Studio has been opened, specific menus or menu choices are added to the menu bar of the Accelerator perspective. In addition, some standard menu choices of the Data perspective might have a different effect when IBM DB2 Analytics Accelerator Studio is loaded.

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

Window > Open Perspective > Other > Accelerator
Opens the Accelerator perspective, which shows IBM DB2 Analytics Accelerator Studio (if it has been installed).

Window > Reset Perspective
Restores all default views of IBM DB2 Analytics Accelerator Studio to their default size and position in the IBM Data Studio main window.

Window > Show View
Opens a submenu that allows you to reopen a currently closed IBM DB2 Analytics Accelerator Studio view.

Help > Welcome > Overview > IBM DB2 Analytics Accelerator Studio
Opens the *Introduction* topic in the online help system.

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

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

Administration Explorer

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

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

© Copyright IBM Corp. 2009, 2017

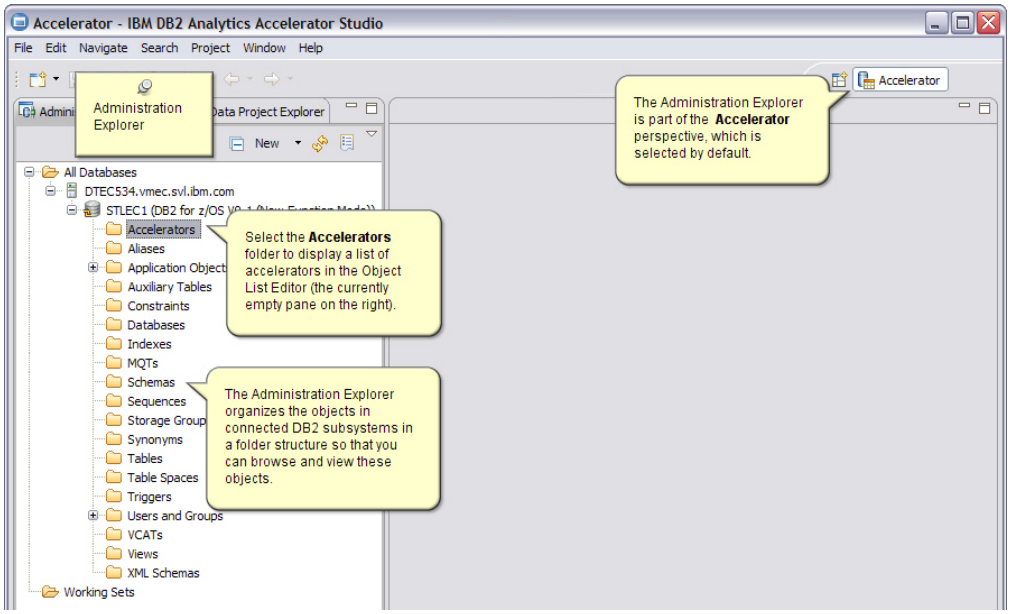


Figure 5. Administration Explorer

Menus

Depending on your selection in the Administration Explorer, the menu (right-click action) offers the following menu choices specific to IBM DB2 Analytics Accelerator for z/OS:

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

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

Disconnect
Disconnects an active database connection.

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

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

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

Folders

When a database server has been connected to, you can see the various objects on that server, which are organized in folders. Most folders pertain to database objects that are created or configured in DB2 for z/OS. An exception is the **Accelerators** folder, which pertains to IBM DB2 Analytics Accelerator for z/OS only.

Accelerators folder

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

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

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

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

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

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

Object List Editor

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

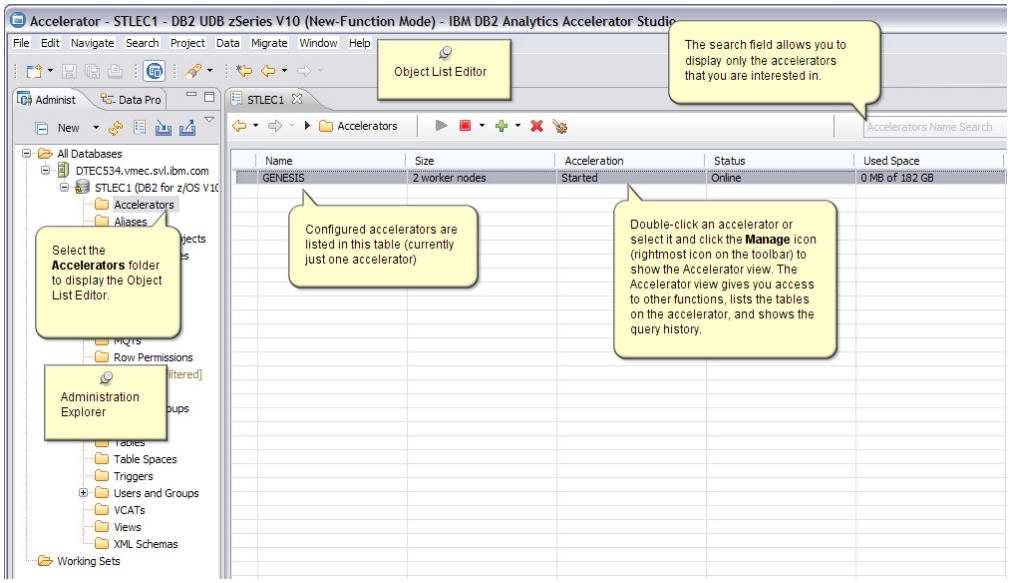


Figure 6. Object List Editor

Buttons in the Object List Editor

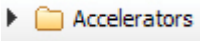
On top of the table in the Object List Editor, you find the following buttons:

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

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

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

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



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

To enable the selected accelerator in DB2 for z/OS.

Notes:

- Multiple selections are not possible here. You can enable only one accelerator at a time.
- If the accelerator is attached to a DB2 data sharing group, acceleration is enabled for all members of the group.

To disable the selected accelerator in DB2 for z/OS.

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

Stop Same as clicking the button on the toolbar

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

Notes:

- Multiple selections are not possible here. You can disable only one accelerator at a time.
- If the accelerator is attached to a DB2 data sharing group, acceleration is disabled for all members of the group.

To 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 accelerator-shadow tables and invoke various other functions.

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

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

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

Column information

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

Name Name of an accelerator

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

Acceleration
Whether acceleration has been *started* or *stopped* for the DB2 subsystem that is associated with an accelerator.

Note: The status of an accelerator cannot not been determined if acceleration has been stopped for a DB2 subsystem.

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

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

Fully Operational
Ready to be enabled in DB2 for z/OS.

Online
Ready for processing.

Running
The accelerator is processing queries.

Stopping
Interim state that is shown when you click the **Stop** button or issue the **STOP -ACCEL** command from DB2 for z/OS. During the *Stopping* phase, already started queries are completed.

Stopped
The accelerator is currently disabled in DB2 for z/OS.

Unknown
The current status of the accelerator cannot be determined.

Used Space
Overall size of the accelerator-shadow tables on an accelerator

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

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

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

Configured
The incremental update function is available for this accelerator, but accelerator-shadow 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 accelerator-shadow tables that have been included in the process.

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

Accelerator 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.

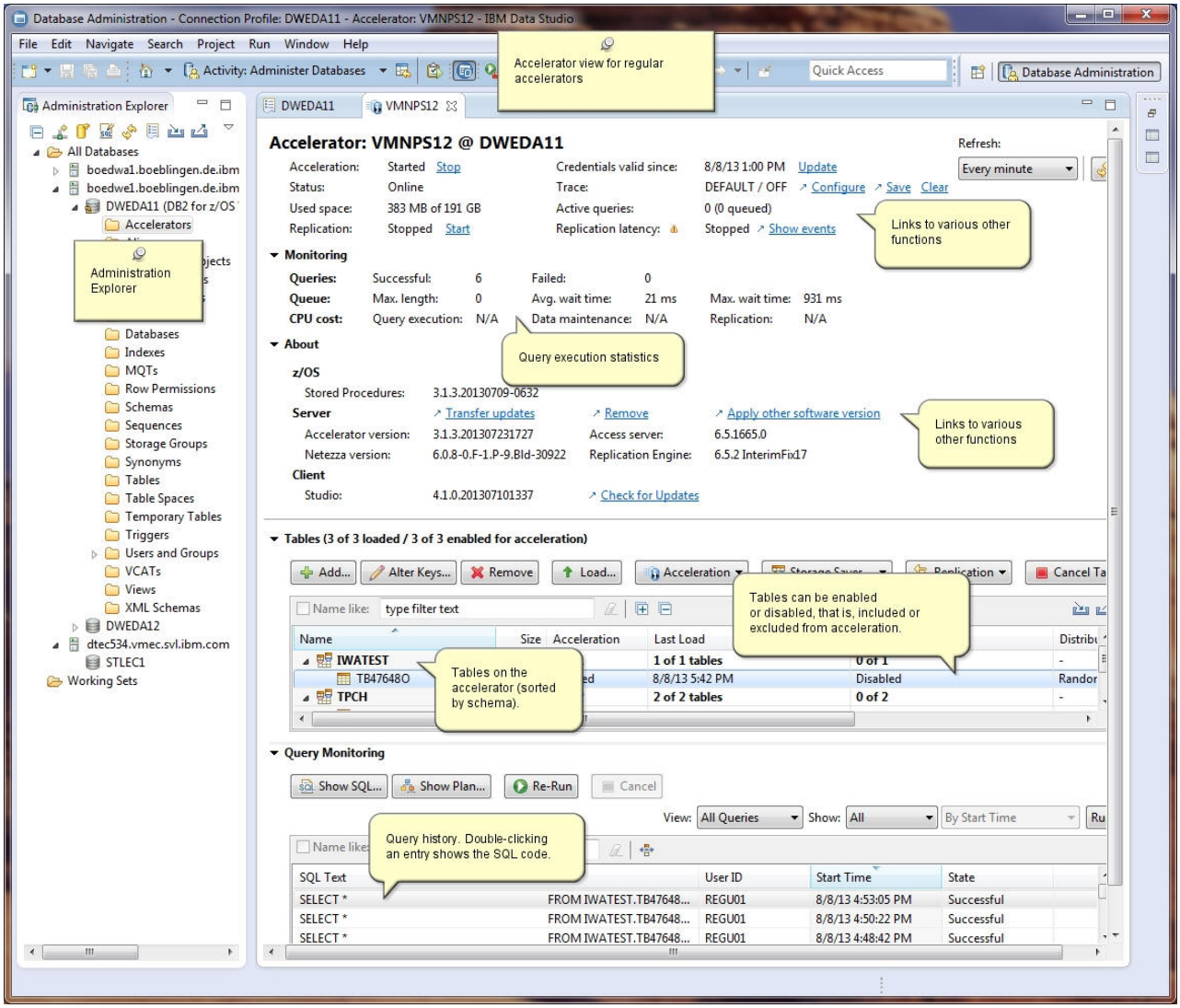


Figure 7. Accelerator view for regular accelerators

Status information and functional links

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

Table 18. Header information

Label	Description
Acceleration	Whether the accelerator is currently enabled in DB2 for z/OS. To change the current status, click **Start** or **Stop**. If the accelerator is attached to a DB2 data sharing group, the behavior is different. The status *Started* is displayed if acceleration has been started for at least one member of the data sharing group. Other statuses are shown only if the status is the same for all members of the data sharing group. If the status of individual members is different, a warning icon is displayed. If you move the mouse pointer over the displayed status or the warning icon, a pop-up window shows the acceleration status for all members of the data sharing group. An additional functional link labeled **Change** is displayed for data sharing groups. Clicking it opens a window in which you can change the acceleration status of each member in the data sharing group.
Status	The status of the accelerator hardware. If the accelerator is attached to a DB2 data sharing group, you can move the mouse pointer over the status information to display the status of each member in the data sharing group.
Used space	The amount of disk space that is taken up by the tables, in proportion to the available space on the accelerator.
Replication	Only visible if incremental updates are configured. A status of **Started** indicates that the incremental update process is running for tables included in the process. **Stopped** indicates that the incremental update process has been suspended. To change the current status, click **Start** or **Stop**.
Credentials valid since	Shows the date on which the last authentication of the accelerator took place. To renew the authentication, click **Update**.
Encryption details	If you click the **Encyption details**, the Encyption Details window opens, which shows information about the use of encryption for the selected accelerator. This includes the status of hard-disk encryption, the status of encryption of data in motion, a list of encrypted IPsec tunnels (IP connections), and a list of the certificates that are used.
Trace	Shows the trace profiles for the accelerator and the stored procedures that are currently in use. To change a profile, click **Configure**. To change the settings for the saving of trace information to a file, click **Save**. To delete the trace information that has been collected, click **Clear**. **Note:** The **Clear** option is not available for cloud accelerators.
Active queries	The number of accelerated queries that are currently being processed. In parentheses, you see the number of queued queries.

Table 18. Header information (continued)

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

Table 19. Header information (continued)

Label	Description
Replication latency (continued)	**Source Agent** Number of data changes to relevant DB2 for z/OS tables (tables for which accelerator-shadow tables have been defined and that are included in the incremental update process. **Inserts** Number of row inserts in the relevant DB2 for z/OS tables. **Updates** Number of updates to rows in the relevant DB2 for z/OS tables. **Deletes** Number of row deletions from the relevant DB2 for z/OS tables. **Latency** Repetition of the **Replication latency** value. Clicking **Show events** opens the event viewer, which protocols all transactions that are related to incremental updates.

Monitoring

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

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

Label	Description
Queries	Metrics related to queries since the start of the currently viewed accelerator, divided into the following categories: **Successful** Number of queries that ran successfully **Failed** Number of queries that failed
Queue	Metrics related to queuing since the start of the currently viewed accelerator: **Max. length** The highest number of queries in the queue, up to this time. **Important:** • This value is not displayed for cloud accelerators. • The **Max. length** and **Max. wait time** values cover all system transactions during a given period of time. This includes internal system queries, as well as INSERT, UPDATE, and DELETE operations. Therefore, these fields might display positive values even if there have been no active user queries recently. A discrepancy becomes noticeable when you compare the values with the ones reported in the query history, which provides the same information for individual queries. **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. **Note:** This value is not displayed for cloud accelerators.

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

Label	Description
CPU Cost	CPU time that was used since the start of the currently viewed accelerator, with regard to the connected DB2 subsystem. The total CPU time is divided into the following categories: **Query execution** CPU time needed for query processing **Data maintenance** CPU time needed for load and High Performance Storage Saver (HPSS) operations **Replication** CPU time needed for incremental updates

About

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

Table 21. Information in the **About** *section*

Label	Description	Information blocks	Description
z/OS	Information about components on the z/OS data server	**Stored procedures**	The currently installed version of the IBM DB2 Analytics Accelerator stored procedures
Server	Information about software components that are installed on the IBM PureData System for Analytics	**Accelerator server**	The version of the IBM DB2 Analytics Accelerator for z/OS software that is currently in use. To transfer an update package from the z/OS UNIX file system (zFS) 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**. **Note:** The **Transfer updates**, **Remove**, and **Apply other software version** options are not available for cloud accelerators.
		Netezza Performance Server (NPS) dashDB™ Level	The current version of the Netezza Platform Software (NPS) or, if a cloud accelerator is used, the current version of the dashDB database on the cloud server. Before you install a new version of the Netezza Platform Software (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 z/OS UNIX file system (zFS) 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 z/OS UNIX file system (zFS) 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.

*Table 21. Information in the **About** section (continued)*

Label	Description	Information blocks	Description
Server (continued)		**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 z/OS UNIX file system (zFS) 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.
Optional Server Packages	If external packages (remote stored procedures) are installed, the package name is shown here (for example: *IBM Netezza Analytics*). In addition, functional links are displayed to enable or disable the package for the selected accelerator.		

Tables

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

*Table 22. Buttons in the **Tables** section*

Button	Description
Add	Starts the Add Tables wizard.

*Table 22. Buttons in the **Tables** section (continued)*

Button	Description
Alter Keys	Opens a window in which you can select or change the distribution and organizing keys that are used by the accelerator. **Note:** This option is not available for cloud accelerators.
Remove	Removes the selected tables from the accelerator.
Load	Opens the Load Tables window, in which you can select accelerator-shadow tables for loading and set the lock scope for the load process.
Acceleration	Opens a pull-down menu, from which you can enable or disable the selected accelerator-shadow tables so that these are included or excluded from query acceleration.
Storage Saver	Allows you to archive partitions on 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 archive the data therein on the accelerator currently viewed. **Restore Partitions to DB2** Opens Restore Storage Saver Partitions to DB2 window, in which you can select accelerator-archive tables and partitions that have been archived on the currently selected accelerator in order to restore these. **Note:** The **Storage Saver** option is not available for cloud accelerators.
Replication	Only visible if incremental updates are configured. This is not the case in Figure 7 on page 67. Clicking the button opens a pull-down menu from which you can enable or disable incremental updates for selected accelerator-shadow tables. **Note:** This option is not available for cloud accelerators.
List Tasks	Opens the List Tasks window, which displays the running tasks on the currently viewed accelerator, such as table load processes. To cancel a running task, select it in the list and click **Cancel Selected Task**.

Table 23. Icons and columns in the table of tables

Item	Description
	Icon representing the schema of a table
	Icon representing a table schema with not yet synchronized changes. It is shown if, for example, columns have been added to a DB2 for z/OS table that are not yet part of the corresponding schema definition on the accelerator.
	Icon representing an accelerator-only table
	Icon representing an accelerator-shadow table
	Icon representing a table with not yet synchronized changes. It is shown if, for example, columns have been added to a DB2 for z/OS table, but not yet to the corresponding schema accelerator-shadow table.
Name	Column containing the table names
Size	Column containing the table sizes
Acceleration	Column that shows whether acceleration is enabled for a table
Last Load	Column that shows the timestamp of the last successful load for each table or the current table state. Because accelerator-only tables are not loaded by IBM DB2 Analytics Accelerator functions, the column always shows the table state (for example *AOT ready*) for this type of table.

Table 23. Icons and columns in the table of tables (continued)

Item	Description
Storage Saver Partition	Column that provides information about the archived partitions of a table. The column is visible only if tables with archived partitions exist. At the schema level, the column shows how many partitions out of the total number of partitions were archived. At the table level, it shows the compressed data size of the archived partitions and the number of rows in these partitions. If archiving is in progress for a partition, the columns shows *In Progress*.
Replication Since	Column that shows the last incremental-update capture-point (a timestamp) for incrementally updated accelerator-shadow tables or *Disabled* for the tables that are not updated in that way. If continuous incremental updates are enabled, the column shows *Suspended* when a table load has started and incremental updates are therefore on halt, and *Refreshing* when table changes are being applied, be this through the ongoing load process or the draining of incremental updates from the spill queue.
Distribution Key	Column that shows *Random* if random distribution is used or the names of distribution-key-columns if a distribution key is used.
Skew	Column that shows the skew value for each table on the accelerator. The skew value is a metric for the distribution imbalance of table rows across the snippet-processing units (SPUs) or worker nodes. The value is the difference between the smallest and the largest data slice for a table in megabytes (MB).
Organizing Keys	Column that shows the names of the organizing keys (column names) that were selected for a table.
Organized	Column that shows the percentage of the table data that is organized based on the specified organizing keys for that table. For organized tables, this value is typically 100%, and 0% for tables that are not organized at all. Values in between indicate that an organizing key has been specified, and that the organization process is ongoing. For small tables, the value is not displayed (it only makes sense to use organizing keys when the table has at least 5 extents per data slice). If you archive tables with the High-Performance Storage Saver (HPSS), the value decreases and ends up at 0% when you have archived the entire table. The reason is that the former accelerator-shadow table is emptied during the process. Hence the degree of organization can no longer be displayed for the archived partitions. The degree of organization of an accelerator-archive table (a table in the read-only table space) is not displayed, but remains the same as before archiving.

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 116.

Stored procedure call

To display or refresh the list of schemas and tables, IBM DB2 Analytics Accelerator Studio invokes the **SYSPROC.ACCEL_GET_TABLES_INFO** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Related concepts:

"Tables states" on page 13

"How to select tables for query acceleration" on page 7

Related tasks:

"Defining tables on an accelerator" on page 88
"Loading tables" on page 93
"Enabling tables for query acceleration" on page 98
"Specifying or changing a distribution key or organizing keys" on page 100
"Including or excluding tables from incremental updates" on page 139
"Selecting and ordering query monitoring columns" on page 116

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.

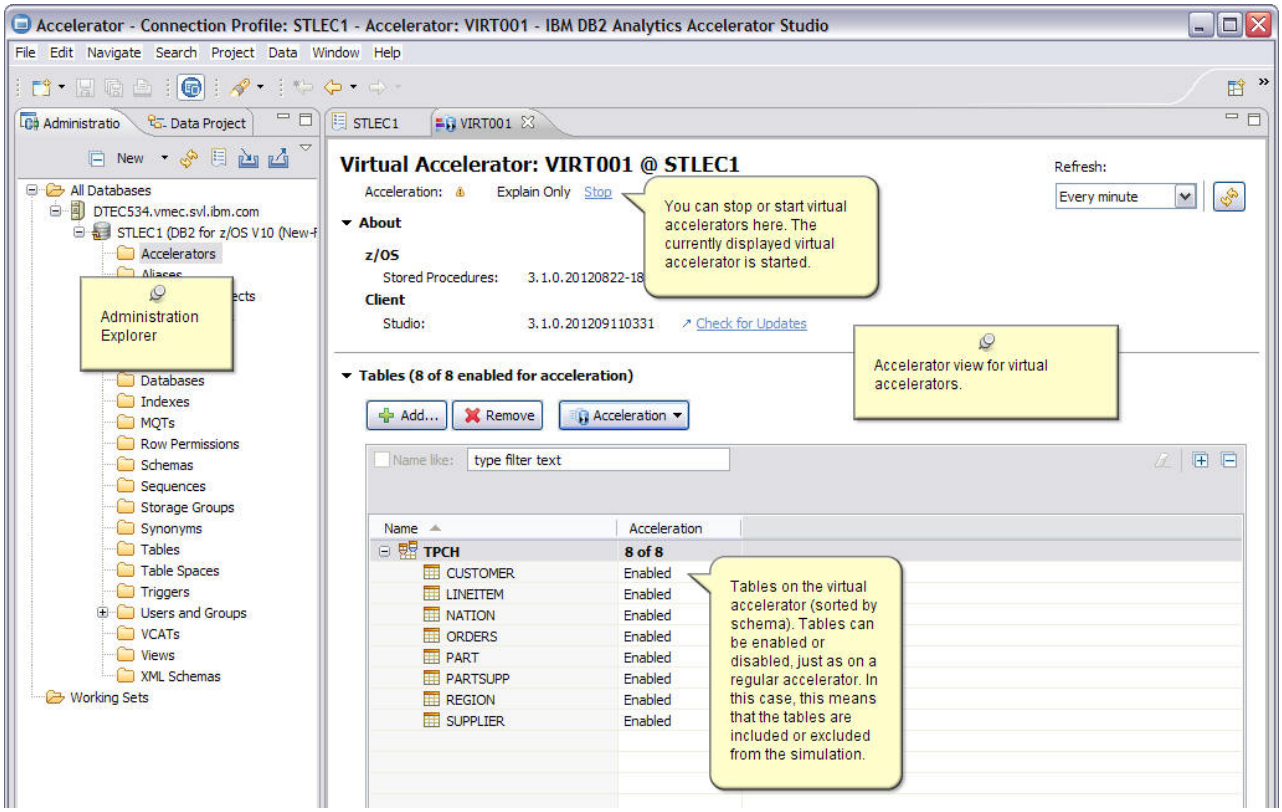


Figure 8. Accelerator view for virtual accelerators

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

Table 24. Information in the **About** *section*

Label	Description	Information blocks	Description
z/OS	Information about components on the z/OS data server	**Stored procedures**	The currently installed version of the IBM DB2 Analytics Accelerator stored procedures
Client	Information about your IBM DB2 Analytics Accelerator Studio installation	**Studio**	The currently installed version of IBM DB2 Analytics Accelerator Studio. Click **Check for Updates** to look for a newer version.

Tables

The **Tables** section in the Accelerator view for virtual accelerators contains buttons and a table listing the accelerator-shadow tables on the virtual accelerator.

Table 25. Buttons in the **Tables** *section*

Button	Description
Add	Starts the Add Tables wizard.
Remove	Removes the selected accelerator-shadow tables from the virtual accelerator.
Acceleration	Opens a pull-down menu, on which you can enable or disable the selected accelerator-shadow tables so that these are included or excluded from the simulation.

Table 26. Icons and columns in the table of accelerator-shadow tables

Item	Description
	Icon representing the schema of an accelerator-shadow table
	Icon representing an accelerator-shadow table
Name	Column containing the names of accelerator-shadow tables
Acceleration	Column that shows whether acceleration is enabled for an accelerator-shadow table

Stored procedure call

To display or refresh the list of schemas and accelerator-shadow 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 7

Related tasks:

"Defining tables on an accelerator" on page 88

"Enabling tables for query acceleration" on page 98

Related information:

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

Chapter 6. Tasks

See an overview of the tasks that you can complete by using IBM DB2 Analytics Accelerator Studio.

Prerequisites

Before you go about a task, check whether you meet the requirements for running IBM DB2 Analytics Accelerator Studio:

- Network connections must exist between the database server and the machine on which IBM DB2 Analytics Accelerator Studio is installed. In addition, network connections must exist between the database server and the accelerators.
- At least one accelerator (hardware) must be installed and connected to your database server.
- The appropriate Program Temporary Fixes (PTFs) must be installed so that IBM DB2 Analytics Accelerator for z/OS can connect to your database management system.
- IBM DB2 Analytics Accelerator for z/OS stored procedures must exist on the database server.
- IBM DB2 Analytics Accelerator for z/OS jobs have been run to configure and integrate the accelerator hardware and software into your database environment.

Creating a database connection profile

Create a database connection profile to gain access to a DB2 subsystem on a database server. A DB2 subsystem houses one or more databases, in which the source data for query acceleration (schemas and tables) is kept.

About this task

In IBM DB2 Analytics Accelerator Studio, the connection information is stored in profiles for reuse. Having created a profile, you can reconnect to a database by double-clicking the icon representing the profile in the Administration Explorer.

Procedure

1. On the header of the Administration Explorer on the left, click the downward-pointing arrow next to **New** and select **New Connection to a Database**.
2. In the New Connection window, decide how to name the database connection profile:
   - To use the name of the database server that you want to connect to, leave **Use default naming convention** selected.
   - To choose a different name, clear **Use default naming convention**, and type the name in the **Connection Name** field.
3. From the **Select a database manager** list, select **DB2 for z/OS**.
4. Make sure that in the **JDBC driver** drop-down list, **IBM Data Server Driver for JDBC and SQLJ (JDBC 4.0) Default** is selected.
5. In the **Location** field, type the name of the database server that you want to connect to.

© Copyright IBM Corp. 2009, 2017

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

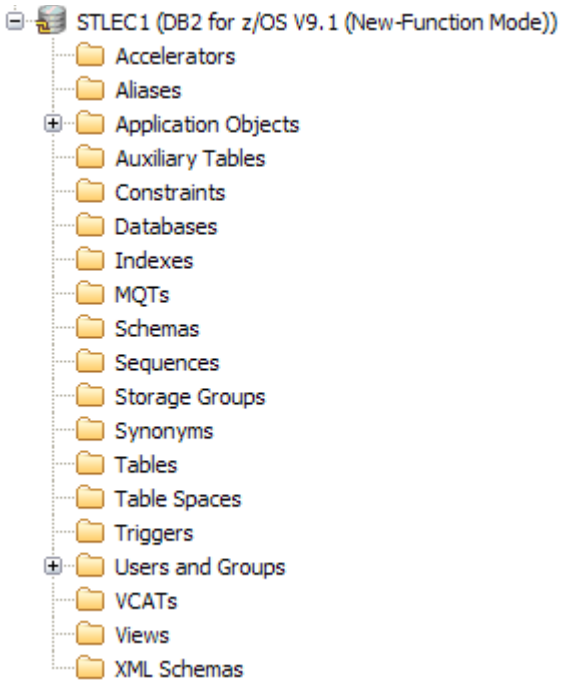


Figure 9. Database server connection in the Administration Explorer

Exploring a database server

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

Procedure

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

Results

When IBM DB2 Analytics Accelerator for z/OS has been set up correctly, the Administration Explorer shows a folder called **Accelerators**. You can open this folder to display the list of configured accelerators. Double-clicking an accelerator name opens a view that shows the details of this accelerator.

Adding accelerators

Adding an accelerator to your configuration is a two-step process. First, you must obtain a pairing code from the IBM DB2 Analytics Accelerator Console. Second, you enter this paring code along with the IP address and the name of the accelerator in the Add Accelerator wizard of IBM DB2 Analytics Accelerator Studio.

Obtaining the pairing code for authentication

Communication between an accelerator and a DB2 subsystem requires both components to share credentials. These credentials are generated after you submit a temporarily valid pairing code. This step is required each time you add a new accelerator. The following steps describe how to obtain the pairing code.

About this task

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

The steps *Obtaining the pairing code for accelerator authentication* and *Adding Accelerators* (next topic) belong together, but are seldom carried out by the same person. Since the pairing code obtained from the IBM DB2 Analytics Accelerator Console is only valid for a limited time (30 minutes by default), the persons operating the console and IBM DB2 Analytics Accelerator Studio must coordinate the steps.

To obtain the pairing code, the **SYSPROC.ACCEL_UPDATE_CREDENTIALS** stored procedure is run on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. Ask the network administrator or the person who did the TCP/IP setup for the IP address (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
 (6) - (Menu) Manage 'Call Home'
 (7) - (Menu) Manage Encryption

 ---------------------------------------------------------------------
 (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) - Enable the conversion mode for 24:00:00 h values to 23:59:59 h
(9) N/A: - Disable the conversion mode for 24:00:00 h values to 23:59:59 h
(10) - Set a system-wide resource limit
```

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:

"Completing the authentication using the Add Accelerator wizard"

Related information:

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

Completing the authentication using the Add Accelerator wizard

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

Before you begin

Make sure that the following conditions apply:

- You need privileges to run DB2 administration commands and stored procedures on z/OS. If you created a power user as suggested in the *Installation Guide*, the power user will have the required privileges.
- You have a valid pairing code. The pairing code, which is of temporary validity, can be obtained by using the IBM DB2 Analytics Accelerator Console.

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

   Note: If the network connection between the z/OS data server and the accelerator is encrypted, an MTU (maximum transmission unit or frame size) error is reported. The message says that the maximum frame sizes on the z/OS data server and on the accelerator differ. Although the difference is real (is has to do with the IPsec implementation), you can ignore this message because the difference has no impact.
9. Click **OK**. A connection test is carried out. See the note in step 8. It also applies here because it is the same test.

Related tasks:

"Obtaining the pairing code for authentication" on page 83

Related information:

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

"DRDA connection does not work" on page 189

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 in DB2 for z/OS:

- 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 accelerator-shadow 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 112

Displaying accelerator networking details

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

About this task

The information can be extracted from certain DB2 catalog tables and system tables:

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

Procedure

1. In the header of the Administration Explorer in IBM DB2 Analytics Accelerator Studio, click the downward-pointing arrow and select **New SQL script**.
2. Copy and paste the following query in the script window on the right:

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

3. Click

4. If no connection profile is selected in the Administration Explorer, the Select Connection 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 accelerator-shadow tables on an accelerator.

Before you begin

- Make sure that at least one accelerator has been configured.
- Make sure that a database connection is associated with this accelerator.
- Make sure that your resource database contains tables.

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 EDITPROC encryption. SQL encryption and decryption functions, such as DECRYPT_CHAR(), require binary table columns, which cannot be used on an accelerator. Furthermore, data that is stored in DB2 with the EDITPROC encryption tool will not be encrypted on the hard disk of the accelerator. However, the data on an accelerator is encoded with a special algorithm for efficient compression and therefore cannot be easily extracted from the media.

This task is carried out by the **SYSPROC.ACCEL_ADD_TABLES** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In IBM DB2 Analytics Accelerator Studio, in the Administration Explorer, go to the **Accelerators** folder.
2. In the Object List Editor, double-click the accelerator that you want to define tables on.

3. In the Accelerator view, click **Add** on the toolbar, which is located above the list of tables.
4. In the Add Tables to Accelerator window, you see a list of all available schemas and tables under **Available tables**. If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.
5. Specify the tables. The list of selectable items is organized hierarchically. At first, you might only see the available schemas. If you click the twistie in front of a schema name, the tables belonging to the schema come into view.

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

   The **Unsupported Columns** column lists the unsupported columns in each DB2 table (if any). If possible, do not select tables with unsupported columns because such columns are not loaded on the accelerator. Queries referencing such columns cannot be accelerated. In most cases, a column is not supported because its data type cannot be handled by IBM DB2 Analytics Accelerator for z/OS.

   Note: Detecting unsupported columns takes a long time if IBM DB2 Analytics Accelerator Studio has to connect to a database with numerous tables. To save this time, you can suppress the display of unsupported columns. To do so, go to **Window** > **Preferences** > **IBM DB2 Analytics Accelerator** > **Accelerator View** and clear the **Check for unsupported columns** check box.

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

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

   Note: If a DB2 source table was created after IBM DB2 Analytics Accelerator Studio connected to a DB2 subsystem on the data server, you might be unable to locate this table in the list of available tables. In such a case, click .

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

   a. Close the Add Tables window.
   b. Remove the filter.
   c. Reopen the Add Tables window.
   d. Click .

6. Click **Finish**.

Related tasks:

"Preparing tables for workload balancing and accelerating static SQL queries" on page 92

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"

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"

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 accelerator-shadow tables; each table can be disabled individually.

Before you begin

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

Procedure

1. In the Administration Explorer, select the **Accelerators** folder.
2. In the Object List Editor, double-click an accelerator to open the Accelerator view.
3. Complete one of the following steps:
   - Click on the accelerator to disable the selected accelerator without canceling running queries. The accelerator status first changes to *Stopping*, then to *Stopped*. During the *Stopping* phase, running queries are completed.
   - Click the button on the right of the button and select **Force** from the menu to disable the selected accelerator and cancel all running queries. The status of the accelerator changes to *Stopped* immediately.

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

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 accelerator-shadow 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 accelerator-shadow 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 accelerator-shadow 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 accelerator-shadow tables.
  5. Enable query acceleration for the redefined accelerator-shadow tables.

Related concepts:

"Static SQL query support" on page 6

"Workload balancing" on page 20

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

Before you begin

- A database connection profile must exist for the database that is associated with the accelerator.
- You (your user ID) must be connected to the database. Your user ID requires privileges to:
  - Control the accelerator
  - Read the data to be loaded into the accelerator
- The accelerator-shadow tables that you want to load must exist on the accelerator.
- The following procedure is not for accelerator-only tables. To load accelerator-only tables, you have to use `INSERT` statements. These can be

submitted from DB2 for z/OS directly, from various client programs using ODBC or JDBC connections, or from application programs with embedded (static) SQL code.

About this task

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

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 original table. The conversion of the values in such columns is CPU-intensive.
- Compression of data in DB2 for z/OS.
- Number of available processors.
- Workload Manager (WLM) configuration.
- Workload on the 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 EDITPROC encryption. SQL encryption and decryption functions, such as DECRYPT_CHAR(), require binary table columns, which cannot be used on an accelerator. Furthermore, data that is stored in DB2 with the EDITPROC encryption tool will not be encrypted on the hard disk of the accelerator. However, the data on an accelerator is encoded with a special algorithm for efficient compression and therefore cannot be easily extracted from the media.
- During a load process, you cannot close IBM DB2 Analytics Accelerator Studio or the database connection before the loading has finished.
- Incremental updates are not restarted automatically if you enable these for just one table and complete the enablement steps in this order:
  1. Define the accelerator-shadow table.
  2. Enable incremental updates for this table.
  3. Load the accelerator-shadow table.
  4. Start incremental updates (**Replication** > **Start**).
  5. Reload the accelerator-shadow table.

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

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

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

For information about the privileges that are required to run these procedures and further details, see *Appendix C. Required access rights* in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related information** at the end of this section.

Procedure

1. In the Administration Explorer, select the **Accelerators** folder.
2. In the Object List Editor, double-click the accelerator containing the tables that you want to load.
3. In the Accelerator view, you see a list of the tables on the accelerator. Select the accelerator-shadow tables that you want to load. To see the tables, you might have to expand the schema nodes first, by clicking the plus sign in front of a schema name. Selecting an entire schema will select all accelerator-shadow tables belonging to that schema for loading.
4. Click **Load**. The list in the Load Tables window contains a column labeled **Load Recommended**. The information in this column is supposed to help you with the selection. Because table loads are long-running processes, you should only load or reload an accelerator-shadow table or partition when necessary.

   Important:
   - You can only select table partitions if the tables are partitioned by ranged or partitioned by growth, and if they were already loaded in the past. In all other cases, you can only load entire tables.
   - The information in the **Load Recommended** column might be inaccurate if DB2 realtime statistics are not up-to-date (RUNSTATS) or if incremental updates have been applied to the accelerator-shadow table since the last load.

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

   No - data unchanged
   Table or partition data has not changed in DB2 for z/OS. Therefore, you need not load or reload the accelerator-shadow table or partition.

   Yes - changed data
   A load or reload of the accelerator-shadow table or partition is recommended because data changes have been applied to the original DB2 for z/OS tables.

   Unknown - no statistics
   A load or reload of the accelerator-shadow table or partition is recommended because the current status of the table or partition cannot be determined. It must therefore be assumed that the table is out of sync. The reason might be that access to DB2 statistics is not available or that the last successful invocation of RUNSTATS was some time ago.

   Unknown - replication enabled
   Data changes have been detected since the last load of the accelerator-shadow table, but the table is currently enabled for incremental updates. The data changes are propagated asynchronously

by the incremental update function, but it can not be determined whether the table is already in sync. A reload is usually not required under this condition.

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

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

All (default)
Displays all accelerator-shadow tables and partitions.

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

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

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

Unknown - no statistics
Displays only the accelerator-shadow tables and partitions for which a load or reload is recommended because the status cannot be determined.

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

5. Select the appropriate tables or partitions. Tick the check boxes in front of the table or partition names, or use the buttons on the right:

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

Deselect All
Deselects all tables and partitions.

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

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

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

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

6. Select **Lock DB2 tables while loading** if you want to protect (lock) the original tables or table partitions in the database against changes during the load operation:

   `All Tables`
   Protects the DB2 source tables of the tables to be loaded against changes during the load operation.

   `Current Table`
   Protects just the DB2 source table of the table that is currently being loaded.

   `Current Partitions`
   Protects the DB2 table-space partition containing the part of the of the source table that is currently being loaded. With this setting, an unpartitioned table is always locked completely.

   `Current Row`
   Protects at least the row or page that is being loaded against updates. Whether a row, a page, or an entire table is locked is determined by the LOCKSIZE defined for the table space of the source tables. (For more information, see the DB2 documentation for the CREATE TABLESPACE command. You find a link at the end of this topic.) When lock mode ROW is set, DB2 data is unloaded at isolation level CS (cursor stability), but in contrast to lock mode `NONE`, rows locked by an application are not skipped. This is the recommended choice for loads in connection with *continuous incremental updates*.

   Attention: This lock mode might lead to a loss of not yet transferred changes if *continuous* incremental updates are not enabled. This is because a load stops "ordinary" incremental updates, and a new capture point (point in time after which table changes are monitored) is not set before load completion. If table rows are locked for the unloading of row data and unlocked shortly after, when the process moves on to the next row, unmonitored row changes can occur. Such changes would be lost.

   Therefore, use lock mode **Row** only if you have enabled *continuous incremental updates* because this type of incremental updates saves all changes not yet applied to a spill queue, which is emptied after the load (changes that occurred before the setting of the capture point are applied at a later time).

   To avoid deadlocks during the draining of the spill queue, set the table space of the source tables to `LOCKSIZE ROW`. For more information, see *Enabling continuous incremental updates* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

   If you want to use this lock mode in a different context, contact IBM support before doing so.

   `NONE`
   No locking at all. However, only committed data is loaded into the accelerator-shadow tables because the DB2 data is unloaded with isolation level `CS` and `SKIP LOCKED DATA`.

7. The option **After the load, enable acceleration for disabled tables** ensures that you can run accelerated queries against all accelerator-shadow tables immediately after the completion of the load operation. It is selected by default. Deselect it only if there is a good reason to do so. The following scenario is thinkable: the table is big and one or more columns could not be loaded

because their data type is not supported. These columns, however, would have held the crucial data. In this case, it makes perfect sense to switch the option off to prevent a waste of processing resources.

8. Click **OK**.

   Note: If the load fails, first check whether DSNUTIL was started in DB2 for z/OS.

Related concepts:

"Accelerator-only tables and in-database transformation" on page 16

"Updating accelerator tables continually" on page 136

Related tasks:

"Preparing tables for workload balancing and accelerating static SQL queries" on page 92

Related information:

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

IDLE THREAD TIMEOUT field (IDTHTOIN subsystem parameter)

Enabling tables for query acceleration

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

Before you begin

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

About this task

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

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

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

The task of enabling tables for query acceleration is carried out by the **SYSPROC.ACCEL_SET_TABLES_ACCELERATION** stored procedure on your data server. For information about the privileges that are required to run this

procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

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

Important:

- Query acceleration can be always disabled regardless of the state an accelerator-shadow table is in. This is useful if you need to recover from an error situation in which an accelerator is down or unreachable.
- You might have to reload or even remove tables from an accelerator after the execution of an ALTER TABLE or ALTER TABLESPACE statement in DB2 for z/OS. For more information, see *What to do after changing a table or table space in DB2 for z/OS*. You find a **Related reference** link at the end of this topic.

Procedure

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

Results

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

Related tasks:

"Disabling query acceleration for tables"

"Loading tables" on page 93

"Preparing tables for workload balancing and accelerating static SQL queries" on page 92

"Disabling an accelerator" on page 91

Related reference:

"What to do after changing a table or table space in DB2 for z/OS" on page 148

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 accelerator-shadow table on disk.

Organizing keys must be columns that can be referenced in zone maps. By default, an accelerator creates zone maps for columns of the following data types:
- Integers (SMALLINT, INT, BIGINT)
- DATE
- TIMESTAMP

In addition, an accelerator creates zone maps for columns of the following data types if columns of this type are used as organizing keys:
- CHAR (all sizes, but only the first eight bytes are used in the zone map)
- VARCHAR (all sizes, but only the first eight bytes are used in the zone map)
- DECIMAL (up to 18 digits;)
- REAL
- DOUBLE
- TIME

Important: An organizing key has no effect if the table is too small. The **Organized** column in the Accelerator view reflects this by not showing a value for the degree of organization (percentage). The recommended minimum table size (compressed size on the IBM PureData System for Analytics) depends on the number of worker nodes:

Table 27. 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.

        Important: The RUNSTATS utility cannot collect statistics for accelerator-only tables because the data of this type of table does not reside in DB2. Hence key recommendations that rely on DB2 statistics cannot be given. You only find key recommendations for accelerator-only tables if these can be derived from the column data type.

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

Important: The RUNSTATS utility cannot collect statistics for accelerator-only tables because the data of this type of table does not reside in DB2. Hence key recommendations that rely on DB2 statistics cannot be given. You only find key recommendations for accelerator-only tables if these can be derived from the column data type.

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

"Choosing an organizing key" on page 2

Related information:

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

Synchronizing schemas

When columns have been added to a DB2 for z/OS table and a corresponding accelerator-shadow table exists, the schemas of both tables differ because the accelerator-shadow table does not yet include the new columns. The synchronization function adds the missing columns to the schema of the accelerator-shadow table, so that the schemas of both tables are in sync.

About this task

The only supported schema change is an add-column operation (ALTER TABLE ADD COLUMN). Although it is detected and affected accelerator-shadow tables are flagged with a *synchronization needed* icon in the **Tables** section of the accelerator view, the dropping of a column from an original DB2 for z/OS table cannot be handled by the synchronization function. If you try to synchronize such a table, you receive an error message saying that the synchronization cannot be carried out. This means that in this and all other unsupported cases, you must:

1. Remove the table from the accelerator.
2. Redefine the table.
3. Reload the table.

Already running queries against an accelerator-shadow table with a pending schema change are completed. New queries against the table are blocked.

If you run the stored procedure on a set of different tables and an error or warning occurs, the affected tables are skipped and processing continues (no rollback).

It is possible to synchronize the schema of a table or partition that was archived with the High-Performance Storage Saver (HPSS). That is, columns are added to the archived table, either with their current values, or, when empty, with the default values defined for these columns.

However, bear in mind that the image copy taken at archiving time contains a different table schema (with fewer columns). The image copy thus restores an older state of the table or partition. The added columns not included in the image copy will be present, but filled with the default values defined for these columns.

If more than one accelerator is used for high availability and workload balancing, you must run synchronize the schema on each accelerator with a copy of the affected table. Repeat this after each schema change.

If *federated access* has been set up for a table with added columns, you must remove and then re-create the reference tables because these tables no longer reflect the actual state of the referenced accelerator-shadow table.

The following restrictions exist:

- Currently, incrementally updated tables cannot be synchronized if a schema change has occurred. To synchronize such tables successfully, first disable incremental updates by selecting **Replication** > **Disable Replication** for the affected tables.
- Some restrictions exist for the column default values that are allowed. In general, these must be explicit rather than uncertain. For example, a table with a column definition that includes a TIMESTAMP WITH DEFAULT cannot be synchronized. Likewise, a table cannot be synchronized if a hexadecimal string was used to define the default.
- In addition, the maximum number of schema changes that can be handled for a single table is 255. If this number is exceeded, SQL code `-4732` is returned. To recover from a situation like this, you must:
  1. Restore the archived table or partition.
  2. Remove the image copy by running the MODIFY utility.
  3. Re-archive the table or partition.
- It is not possible to modify the schema of (add columns to) accelerator-only tables. So if you use in-database transformation or IBM Netezza Analytics, you must re-create the underlying tables.

To detect diverging table schemas, the SYSPROC.ACCEL_GET_TABLES_INFO stored procedure is run on your data server. The synchronization itself is carried out by the SYSPROC.ACCEL_SYNCHRONIZE_SCHEMA stored procedure. For information about the privileges that are required to run these procedures and further details, see the appropriate sections 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 topic.

Procedure

1. When you open the accelerator view of an accelerator with tables whose schemas differ from the schemas of the original tables in DB2, the Synchronize Schema window comes up automatically, and you can select tables with added columns for synchronization. See step 7 on page 105 and continue from there.
2. To synchronize the schemas at a later time, reopen the accelerator view in question.

3. Navigate to the **Tables** section.
4. A schema change is indicated by a modified icon for schemas and tables (a white cross in a red circle is added). The icon shows up next to the schema name and the names of tables affected by the change. See the following screen capture:

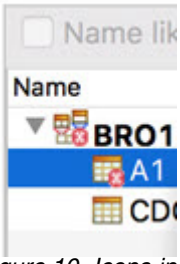


Figure 10. Icons indicating a not yet synchronized schema change

5. Move the mouse pointer over the icon. A message box with details about the schema change opens:

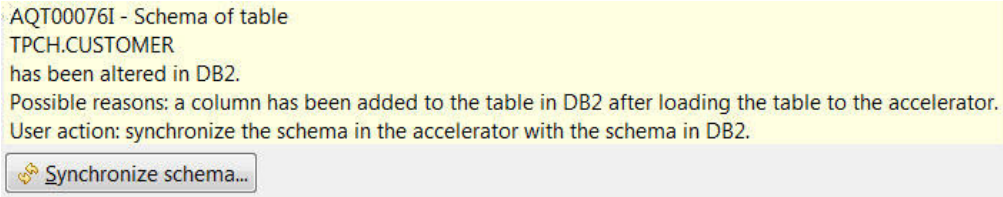


Figure 11. Message box that comes up when mouse pointer is hovering over a schema name with a change indicator icon

6. Click **Synchronize Schema** in the message box.
7. The Synchronize Schema window lists all tables on the accelerator that have not yet been synchronized. Select those that you want to process. The window provides buttons to speed up the process if you have to select many tables: Click **Select All** to select all tables in the window or **Deselect All** to clear all selected check boxes in front of table names.
8. Click **OK**.

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.

Running IBM Netezza Analytics stored procedures

You can now run a number of IBM Netezza Analytics stored procedures on an accelerator to speed up data preparation and the creation of models for data analytics. In addition, there is a support package for the R programming language (called `ibmdbR`), which allows you to start these stored procedures from R scripts.

The following IBM Netezza Analytics stored procedures are currently supported:

- arule, predict_arule
- dectree, grow_dectree, prune_dectree, predict_dectree
- divcluster, predict_divcluster
- export_model_to_table

  Important: This stored procedure is needed for the support of the R programming language.
- glm, predict_glm, print_glm

  Restriction: You can use the glm stored procedure on Unicode data only.
- impute_data
- iqr
- kmeans, predict_kmeans
- linear_regression, predict_linear_regression

Restriction: You can use the linear_regression stored procedure on Unicode data only.

- median, median_disc
- model_exists, drop_model, drop_all_models, list_models
- moments
- naivebayes, predict_naivebayes
- outliers,
- pmml_model
- quantile, quantile_disc
- quartile, quartile_disc
- random_sample
- regtree, grow_regtree, prune_regtree, predict_regtree
- split_data
- std_norm
- twostep, predict_twostep

Installation

For information on how to install and enable these stored procedures, see *Installing the IBM Netezza Analytics stored procedure package* in the *Installation Guide*. This chapter also contains information on how to obtain the R support package (`ibmdbR`).

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.

Preparations

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

- PLAN_TABLE
- DSN_QUERYINFO_TABLE

For basic EXPLAIN information, you need just the PLAN_TABLE. To create EXPLAIN tables (including DSN_QUERYINFO_TABLE) for a specific user, customize and run the DSNTESC member of the SDSNSAMP library. Modify and run the DSNTIJOS sample job to create EXPLAIN tables that are qualified by SYSIBM and DB2OSC, as these tables are used by DB2 optimization tools like IBM Query Tuner.

Description

The EXPLAIN tables provide the following information:

PLAN_TABLE

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

DSN_QUERYINFO_TABLE

Additional EXPLAIN information can be found in the DSN_QUERYINFO_TABLE. See Table 28.

Table 28. 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 29.
`QI_DATA`	When the value of TYPE is `A` and the reason code is `0`, this row contains the SQL code of the converted query that was routed to the accelerator. If a query has a reason code other than `0`, this column contains a description of the reason code.

Reason codes

Table 29 lists and describes the various reason codes. A reason code greater than `0` indicates that a query cannot be accelerated.

Table 29. 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.

Table 29. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
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.
12	The query references a table with one of the following characteristics: • The table is not defined on the accelerator. • The table is defined on a different accelerator. • The table is not enabled for query acceleration.	Ensure that all tables that are referenced by the query are defined on the same accelerator, and that all tables are enabled for query acceleration.
13	The accelerator server that contains the tables that are referenced in the query is not started.	Start the accelerator that contains the tables that are referenced in the query.
14	A column that is referenced in the query has been altered after loading the column data into the corresponding accelerator-shadow table.	Reload the accelerator-shadow table.

Table 29. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
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.

22	The statement references an accelerator-only table, but cannot be routed to an accelerator. For example:
	• A MERGE statement references an accelerator-only table.
	• An UPDATE or DELETE statement references an accelerator-only table, but the target table of the UPDATE or DELETE is a normal DB2 table, for example:
	`UPDATE db2_table SET ... (SELECT ... FROM accelerator_only_table)`
	• An INSERT, UPDATE, or DELETE refers to a row expression with a sub-select, for example:
	`UPDATE accelerator_only_table SET (C1,C2) = (SELECT C3,C4 FROM table_2)`
		• Remove the reference to the accelerator-only table
		• If possible, rewrite the statement so that it references a regular DB2 table instead of an accelerator-only table.

Table 29. Possible reason codes in DSN_QUERYINFO_TABLE (continued)

REASON_CODE	Description	Suggested solution
24	The DDL or DML statement cannot be offloaded because the connection to the accelerator server does not allow updates. This can occur when a two-phase commit requester (for example, a DB2 for z/OS requester) connects to a DB2 for z/OS server to offload a DDL or DML statement."	Issue the DDL or DML statement directly from the DB2 for z/OS server, or from a requester which does not require a two-phase commit, for example, the IBM Data Server Driver for JDBC.

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" on page 112

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 accelerator-shadow tables and then revisit the access plan graph of the query to check whether the query used these tables. In addition, an access plan graph can show which tables are referenced indirectly by the query if the query references a view. You can thus identify all accelerator-shadow tables that you need to load before you can accelerate the query.

Access plan graphs are derived from 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 runs 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 source 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 in the query that was archived by the HPSS. 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 107

Related reference:

"Types of access plan graphs and nodes in these graphs" on page 113

Related information:

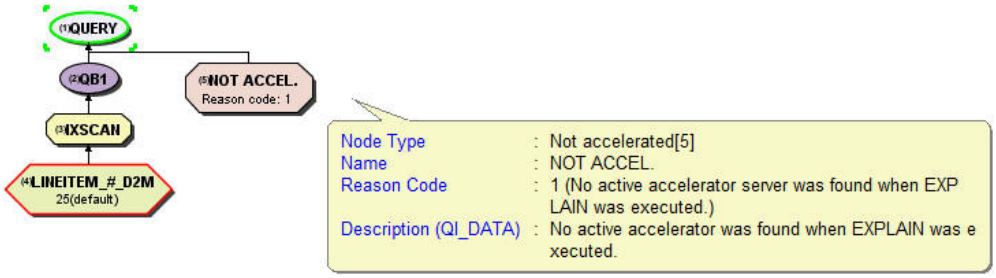
IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference

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



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 30. The nodes reflect the different stages during query processing. Some of these give you valuable information for query optimization.

Table 30. 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 30. 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 colocated 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 30. 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 accelerator-shadow 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 colocated 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 30. 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 an accelerator-shadow 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://www.ibm.com/support/knowledgecenter/SS7L89_3.1.0/com.ibm.datatools.nodes.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 Accelerator view and also change their order of appearance.

Procedure

1. In IBM DB2 Analytics Accelerator Studio, open the Accelerators view.
2. Scroll down until you see the heading **Query Monitoring**.
3. Click the twistie next to the heading to reveal the contents of the section. The contents basically consists of a table that lists the recent query history (SQL statements and other information about recent queries). You can select various information columns, some of which are displayed by default. For a list of the available columns, click the appropriate **Related reference** link at the end of this topic.

Important: The reference for all times that are displayed is the accelerator system time. The accelerator system time is determined by the first DB2 subsystem that was added to the configuration. A warning is issued if the accelerator works with data from a DB2 subsystem that was not the first to be added. It is very important to keep this in mind if the DB2 subsystems (data servers) are situated in different time zones.

4. To add columns, remove columns, or change their order, click .
5. In the Adjust Query Monitoring Table window, you see the following lists:

   Available columns
   The names of available columns that are currently not selected for display

   Shown columns
   The names of the columns that are currently displayed. The order from top to bottom indicates their appearance in the Accelerator view from left to write.

   - To add a single column for display in the Accelerator view, select it in the **Available columns** list and click to move the column to the **Shown columns** list.

     To move all available columns to the **Shown columns** list, click .
   - To hide a single column from the display in the Accelerator view, select it in the **Shown columns** list and click to move the column to the **Available columns** list.

     To remove all columns from the display, click .
   - To change the order of appearance in the Accelerator view, select a column in the **Shown columns** list and click or to move the column up or down.

     The order of the columns from top to bottom represents the order in which these columns appear from left to right in the Accelerator view.
   - To restore the default settings for the **Query Monitoring** section in the Accelerator view, click **Restore Defaults**.
6. Click **OK**. Your settings are applied to the table in the **Query Monitoring** section, that is, columns are added, removed or their order of appearance is changed.
7. The **Query Monitoring** section offers a few additional functions:
   - To show further details about a selected query or to rerun the query, click:

     Show SQL
     To view the SQL of the query

     Show Plan
     To view the access plan graph of the query

     Note: For accelerator-only tables, the **Show SQL** and **Show Plan** functions show information about data manipulation (DML) jobs, such as INSERT, UPDATE, or DELETE, but not data definition (DDL) jobs, such as CREATE or DROP.

     This drop-down button is available for cloud accelerators only. It leads to the following functions:

Show Plan
Same as the button described before

Show Explain Details
Opens a window with a textual representation of a query's access plan and other information about the query execution. As an example, see the following extract:

```
...
FROM
  KLMDWEDB14."CUSTOMER-UID_01140003-V0" AS Q1
WHERE
  (Q1."DWA_Partition_ID (hidden)" <= 100001)

Access Plan:
-----------
 Total Cost:   105.849
 Query Degree:  8


           Rows
          RETURN
          (   1)
           Cost
            I/O
            |
          333.333
          LTQ
          (   2)
          105.849
            15
            |
          333.333
          CTQ
          (   3)
          105.778
            15
            |
          333.333
          TBSCAN
          (   4)
          105.766
            15
            |
           1000
   CO-TABLE: KLMDWEDB14
 CUSTOMER-UID_01140003-V0
            Q1


Operator Symbols :
------------------

   Symbol      Description
   ---------   -----------------------------------------
    ATQ      : Asynchrony

...
```

Re-Run Query
To rerun the query

Cancel
To cancel a running query or an INSERT, UPDATE, or DELETE operation on an accelerator-only table.

Note: The **Re-Run Query**, and **Cancel** options are not available for cloud accelerators.

- 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 reference:

"Selectable columns in the **Query Monitoring** section"

Related information:

"Query history is not displayed - error message returned" on page 192

Selectable columns in the Query Monitoring section

The list in this section briefly describes all columns that can be selected for display in the **Query Monitoring** section of an Accelerator view.

Accounting String
Accounting information that can be provided by an application programming interface (API) on the client side, such as the `setDB2ClientAccountingInformation` API of a connecting JDBC client.

For more information, see Providing extended client information to the data source with IBM Data Server Driver for JDBC and SQLJ-only methods

Client User ID
The ID of a connecting client user. This ID can be provided by an application programming interface (API) on the client side, such as the `setDB2ClientUser` API of a connecting JDBC client.

For more information, see Providing extended client information to the data source with IBM Data Server Driver for JDBC and SQLJ-only methods

Client Workstation Name
The name or ID of a connecting client workstation. This ID can be provided by an application programming interface (API) on the client side, such as the `setDB2ClientWorkstation` API of a connecting JDBC client.

For more information, see Providing extended client information to the data source with IBM Data Server Driver for JDBC and SQLJ-only methods

Collection ID
The ID or name that identifies a group of static SQL packages. If a collection ID is displayed, this means that the static SQL package containing the query belongs to that group.

Connection Name
The connection name. Possible values are:

For batch:
BATCH

For TSO:
TSO

For QMF™:
DB2CALL

For DB2 utilities:
UTILITY

For DB2 private protocol:
The DB2 subsystem ID

For IMS™:
The IMS ID

For CICS®:
The CICS ID

For DRDA connections from non-DB2 requesters:
SERVER

For more information, see IFCID 147 - Thread Correlation Data.

Connection Type
The type of the DB2 connection. Valid values are:

APPLCTN
Application directed access

BATCH
Batch job

CICS CICS attach

DB2CAF
DB2 call attach

DLIBATCH
DL/I batch

IMS IMS thread

IMS BMP
IMS attach BMP

IMS TBMP
IMS transaction BMP

IMS CTL
IMS control region

IMS MPP
IMS attach MPP

SYSTEM
System directed access

TSO/E TSO/E foreground and background

UNKNOWN
Unrecognizable connection type

UTILITY
DB2 utility

For more information, see Thread SQL Counts Attributes.

Correlation ID
The correlation ID value. Possible values are:

For BATCH:
Job name.

For TSO:
Logon ID.

For IMS/VS:
PST#.PSBNAME

For CICS:
CONNECTION_TYPE.THREAD_TYPE.THREAD_#.TRAN-ID

For RRSAF:
CORRELATION-ID value from SIGNON function.

For threads using the DB2 private protocol or DRDA from a DB2 requester:
This field contains the correlation-ID name of the thread at the requesting location.

For threads using DRDA from a non-DB2 requester:
This field contains the first 12 characters in the DDM external name (EXTNAM) parameter of the DDM EXCSAT command received as part of the SQL connect.

For more information, see IFCID 147 - Thread Correlation Data

CPU Time
The overall CPU time that was needed by an accelerator to process a query or data manipulation (DML) statement. This value reflects parallel processing such that the CPU value might exceed the accumulated elapsed time. This is an indicator for the amount of used resources.

For completed statements, this is the CPU time from the initial request to the last row being returned to DB2. In case of DML statements, this is measured until the final result (SQLCA) is sent to DB2.

For in-process statements, this is the time from the initial request to the current point in time. The counter includes the CPU time that was used by the product and also the CPU time spent in the Netezza system (on the

coordinator node and all worker nodes). Because parallel processing is the standard, the CPU time cannot be related to the elapsed time. If query processing has already finished in the Netezza system, the CPU time spent inside this system is included. If the SQL statement is still running in the Netezza system, only the CPU time that was used by IBM DB2 Analytics Accelerator is available.

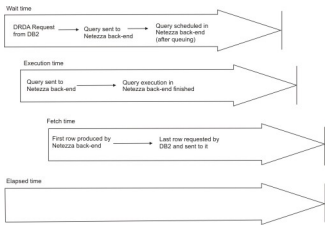
Data Sharing Group Name
The name of the DB2 data sharing group (if the accelerator is connected to such a group rather than a DB2 subsystem)

Data Sharing Member Name
If the accelerator is connected to a DB2 data sharing group, the name of the group member that sent the query.

DB2 Subsystem ID
The DB2 subsystem ID.

Elapsed Time
The total time that was needed to process a query (from the state QUEUED to FAILED, SUCCESSFUL, or CANCELED).

The following diagram shows the beginning, duration, and the end of each time in the **Query Monitoring** section on a time scale. This way, you can better relate the values to each other. It starts with the **Start Time** (point in time, where the diagram starts and when the accelerator receives the query). The following interval, the **Wait Time** overlaps with the **Execution Time** because the **Execution Time** includes the period that a query stays in the 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.



Error Description
The error description or error token that is returned with the **SQL Code**. The description might be truncated.

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

Location Name
Location name of the DB2 subsystem sending the query.

LU Name
Network LU (logical unit) name of the requester sending the query. This is used in the end-to-end monitoring of accelerated queries in multi-tier architectures where DB2 for z/OS handles requests passed on by an application server on behalf of a client application. This allows you to identify queries that were issued by a certain LU Name.

Network ID
Network ID of the requester sending the query. This is used in the end-to-end monitoring of accelerated queries in multi-tier architectures where DB2 for z/OS handles requests passed on by an application server on behalf of a client application. This allows you to identify queries that were issued by a certain Network ID.

Original Authorization ID
The original authorization ID. Possible values are:

For TSO:
The logon ID.

For batch:
The user ID on the job statement.

For IMS (message-driven regions):
The sign-on ID, LTERM, ASXBUSR, or PSB name.

For IMS (control regions):
The user ID on the job statement, or the RACF® started procedure entry if RACF is used.

For CICS:
The user ID, TERM ID, TRAN ID, or as specified in the resource control table.

For MVS™ operator commands and DB2 system internal agents:
SYSOPR

For a distributed application server (AS):
- If the application requester (AR) is a DB2 system, then this is the same value that was assigned at the AR.
- If the application requester is not a DB2 system, then this is the user ID used to make the initial connection with the application server.

For more information, see IFCID 147 - Thread Correlation Data

Package Name
The name that identifies a static SQL package or application package. If such a name is displayed, this means that the query was submitted by an application program in the form of a bound package for static SQL processing.

Plan Name
The plan name. It is blank for a DB2 command thread; otherwise:

DSNESPRR
For SPUFI with repeatable read

DSNESPCS
For SPUFI with cursor stability

DSNUTIL
For DB2 utilities

DSNTEP2
For DSNTEP2

DSNBIND
For binding

The application plan name
For IMS or CICS

A blank plan name
For IMS and CICS commands

DSQPLAN
For QMF

The first 8 bytes of the application name
For DRDA connections to the common servers

For more information, see IFCID 147 - Thread Correlation Data.

Priority
The Workload Manager (WLM) importance level that the query or job holds in DB2.

For more information, see *Mapping of WLM importance levels to Netezza priorities* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

Product ID
ID of the product (application requester) that routed the query, such as a JDBC driver.

Result Size
The data size of the rows that were returned in query results (in MB or GB) from the accelerator to DB2 for z/OS.

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

SQL Code
The SQL code that is returned to the user after the query or job execution.

SQL Text
The SQL statements of recently submitted queries.

Start Time
Times when queries were received by IBM DB2 Analytics Accelerator 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.

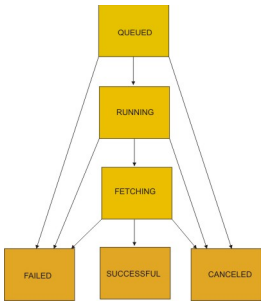
FETCHING
The Netezza system has finished processing the query, but DB2 for z/OS has not yet finished collecting the results.

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:



User ID
The names of users who submitted past queries.

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

Canceling tasks

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

About this task

Using the **List Tasks** function, you cannot cancel queries. To cancel a running query or an INSERT, UPDATE, or DELETE operation on an accelerator-only table, use the **Cancel** button in the **Query Monitoring** section of the Accelerator view.

Procedure

1. Open the accelerator view of the accelerator that the task is running on.
2. Click **List Tasks** on the toolbar.
3. The table in the List Tasks window shows all tasks that are currently running on the accelerator. To cancel a task, select it in the table and click **Cancel Selected Task**. The table that lists the tasks contains the following columns, which give you further details about a task:

   Name The type of the task, for example **Load Tables**.

   Execution Time
   The time that passed since the start of the task.

   Progress
   The degree of completeness since the start of the task.

If individual tables were selected for processing, the number denotes the number of records (table rows) that were processed so far.

If an entire schema was selected for processing, the number represents the percentage of completion. The maximum value is 1, denoting 100 percent. Therefore, a value of 0.25 means 25 percent complete.

Progress Message
Additional information about running tasks.

User The IDs of the users who started the tasks.

4. Click **Close** to close the List Tasks window.

Removing tables from an accelerator

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

About this task

If a part or all of a table (all partitions) have been archived by the high-performance storage saver (HPSS), then this table (an accelerator-archive table) cannot be removed without first restoring the archived data. After restoring the data, the table remains on the accelerator, but becomes and ordinary accelerator-shadow table, which means that it can be removed.

The table removal task is carried out by the **SYSPROC.ACCEL_REMOVE_TABLES** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Procedure

1. In the Administration Explorer, select the **Accelerators** folder.
2. Double-click the accelerator containing the tables that you want to remove.
3. 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 incremental updates" on page 139

Related information:

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

Exporting a table specification

You can save a list of the accelerator-shadow tables on an accelerator to an XML file and in a later step import that list for use with another accelerator. This way, tables can be added more quickly than by adding schemas and tables individually.

About this task

A typical use case is an administrator who saves a valid table specification that was created in a test environment in order to use this specification in a production environment. To make the definition work in a different environment, you can edit the XML file with a text editor, and, for example, use the find-and-replace function to replace the names of schemas and tables.

Procedure

1. Click **File** > **Export**.
2. In the Export window, expand the **IBM DB2 Analytics Accelerator** folder to see its contents.
3. Select **Table Specifications**.
4. Click **Next**.
5. In the Export Table Specification window, select an appropriate database connection profile. The profile gives you access to the accelerators that a DB2 subsystem is connected to. Tables are exported from these accelerators. You thus have to select the proper profile, that is, one that gives you, by way of the connected DB2 subsystem, access to the accelerators that contain the right tables. If the proper profile does not exist yet, you can click **New** to create it.
6. Click **Next**.
7. From the **From accelerator** drop-down list, select the accelerator containing the tables to be exported.
8. In the table below, you can see the schemas and tables on the selected accelerator. From this table, you can select the tables or entire schemas whose specification you want to save to an XML file. If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.
9. Click **Browse**.
10. In the Export to File window, navigate to the folder in which you want to save your table specification XML file.
11. Type a file name in the **File name** field. By default, the file is given an extension of `idaa.xml`. Files of this type are encoded in UTF-8.
12. Click **Save**. You return to the Export Table Specification window. The path and the file name of your table specification XML file are displayed in the **To XML file** field.
13. Click **Finish**.

Importing a table specification

You can import a list of accelerator-shadow tables for query acceleration that was previously saved to an XML file.

About this task

A typical use case is an administrator who saves a valid table specification that was created in a test environment in order to use in a production environment. To make the definition work in a different environment, you can edit the XML file with a text editor, and, for example, use the find-and-replace function to replace the names of schemas and tables.

Procedure

1. Click **File** > **Import**.
2. In the Import window, expand the **IBM DB2 Analytics Accelerator** folder to see its contents.
3. Select **Table Specifications**.
4. Click **Next**.
5. In the In the Import Table Specification window, select an appropriate database connection profile. The profile gives you access to the accelerators that a DB2 subsystem is connected to. Tables are exported from these accelerators. You thus have to select the proper profile, that is, one that gives you, by way of the connected DB2 subsystem, access to the accelerators that contain the right tables. If the proper profile does not exist yet, you can click **New** to create it.
6. Click **Next**.
7. From the **To accelerator** drop-down list, select the accelerator on which you want to define the tables.
8. Click **Browse** to navigate to the folder that contains the XML file with the list of accelerator-shadow tables.
9. In the Import from File window, navigate to the proper XML file, select it, and click **Open**.

   Important: XML files in UTF-8 format are recommended. Such files start with the following header:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   ```

   You return to the Import Table Specification window. The path and the file name of the selected XML file is displayed in the **From XML** file field.
10. In the table below, you can see the schemas and tables that are listed in the selected XML file. From this list, select entire schemas or individual tables that you want to define on the accelerator. If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.
11. Click **Finish**.

Removing accelerators

Follow the steps in this topic to remove accelerators from the system configuration. Bear in mind that this action will also remove the tables from an accelerator.

About this task

The table removal task is carried out by the following stored procedures on your data server:

- **SYSPROC.ACCEL_REMOVE_TABLES**
- **SYSPROC.ACCEL_REMOVE_ACCELERATOR**

For information about the privileges that are required to run these procedures and further details, see *Appendix C. Required access rights* in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related information** at the end of this section.

Procedure

1. In the Administration Explorer, select the **Accelerators** folder.
2. In the Object List Editor on the right, select the accelerator that you want to remove.
3. Click on the toolbar.

Related information:

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

Tracing

IBM DB2 Analytics Accelerator 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.

   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.

   Note: For DB2 Analytics Accelerator for z/OS Cloud Support, the set of options in this window is limited. That is, you can only select one or more of the following check-boxes:

   - **Accelerator trace information** (without sub-options for inclusion or exclusion)
   - **Stored procedure trace information**
   - **Eclipse error log**

   Accelerator trace information
   All the information that is captured as a result of your settings in the Configure Trace window. This might include the following blocks of information (if applicable). As a user of IBM DB2 Analytics Accelerator for z/OS on premises, you can include or exclude individual blocks:

   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.

nzlogcollector (includes user data)
Produces the same output as an **nzlogcollector** command that is issued from a Netezza machine (**/nz/support/bin/adm/nzlogcollector/nzlogcollector)**. The output presents trace information in a way that is familiar to Netezza experts. The option is available only if you use Netezza Platform Software (NPS) (NPS) 7.2.0.5 or later. See the previous Important note.

Network encryption (IPsec)
Only available if *encryption of data in motion* is enabled. The option collects encryption-related trace information.

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.

   Note: Most IBM FTP servers for PMRs accept the registered email address as the password.
   The currently selected server is displayed in square brackets in the text above the **Password** field. To change the server or the proxy settings required to access the server, select **Window** > **Preferences** > **IBM DB2 Analytics Accelerator for z/OS** > **Save Trace Dialog** from the IBM Data Studio main menu.
10. If you want to reset your choices in the Save Trace window to make corrections or start over, click **Restore Defaults**.

    Note: This button just undoes all of your recent actions in the Save Trace window. It does not reset profiles and affects your current settings rather than future settings. Compare this with the previous description of the **Reset configuration profiles to default after collecting traces** check box.
11. Click **OK** to save your settings.

Related information:

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

Collecting CDC trace information

None of the options in the Save Trace window of IBM DB2 Analytics Accelerator Studio collects trace information about the CDC Capture Agent. This information, however, might be the key to a problem with the incremental update function.

About this task

If you are facing a problem with the incremental update function, provide as much information about the related software components as possible the support team.

This includes information about the CDC Capture Agent, which is the part of IBM InfoSphere Data Replication for z/OS that runs on z/OS. The support team might even ask you for this type of information. Here is a description on how to obtain it.

Procedure

1. The CDC Capture Agent is just one of the started tasks running on your z/OS system. If you do not know the name of the started task, ask a z/OS systems programmer which task that is.
2. In the SDSF program on z/OS, type `XDC` in front of the relevant address space output to copy its contents to a sequential data set.
3. Transfer this data set in ASCII mode to your Linux or Windows workstation and add the resulting .txt file to your problem management record (PMR).
4. If a trace dump has been generated, include that information as well. Transfer this data in TERSED format to your workstation because this retains the mainframe-readable formatting of the data during the transfer to a non-mainframe system, and thus ensures that the data remains useful for IBM support. Use an extension of .trs for the files. The TERSED format also reduces the file size. See the documentation for the AMATERSE program if you need further assistance.

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 was 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 accelerator-shadow table in the range of 100 million rows, type a decimal number between 0 and 1 in the

text field next to **Show a warning if the skew value of very big tables exceeds**. Example: `0.7`. The default is `0.25`.

Notes:

– What is considered to be *very big* varies with the size of your IBM 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 accelerator-shadow tables), 1 is already a bad value.

To be able to issue a warning, IBM DB2 Analytics Accelerator Studio must compare the specified skew threshold with the actual skew values of the tables on your accelerators. The actual values are obtained by running the **SYSPROC.ACCEL_GET_TABLES_INFO** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

- To disable checking for unsupported columns, clear the **Check for unsupported columns** check box. By default, this feature is enabled because it tells you beforehand why a table cannot be added or why certain columns were skipped.

  Important: For the deactivation to take effect, you must disconnect, then reconnect to the DB2 subsystem. To do so:

  a. Go to the Administration Explorer.
  b. In the **All Databases** folder, navigate to the appropriate DB2 subsystem icon.
  c. Right-click the icon.
  d. Select **Disconnect** from the menu.
  e. Double-click the icon to reconnect.

2. Click **Apply** to save the changes or **Restore Defaults** to undo the changes.
3. Click **OK** to save your settings and exit the window or just exit the window.

Related information:

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

Updating accelerator tables continually

The incremental update function of IBM DB2 Analytics Accelerator for z/OS allows you to update accelerator-shadow tables continually. Changes to the data in original DB2 for z/OS tables are thus propagated to the corresponding target tables with a high frequency and just a brief delay. This way, query results from an accelerator are always extracted from recent, close-to-realtime data.

The information about changes to be transferred comes from the DB2 for z/OS log.

In the past, data in accelerator-shadow tables could only be updated by use of the table load function. With this function, you can only update entire tables and entire table partitions, and the latter only if the tables are partitioned by range or partitioned by growth. The incremental update function, in contrast, updates

individual table rows. Advantages and disadvantages are connected with either method. Table 31 compares the advantages and disadvantages.

Table 31. Comparison of table update methods

Manual update of entire table using load function	Manual update of entire partitions using load function	Incremental update
For all types of tables	Only for range-partitioned tables and tables partitioned by growth	For all types of tables
Potentially long-running process, depending on the size of the tables	Potentially long-running process, depending on the size of partitions and the distribution of data changes across partitions (the more partitions are involved, the more time-consuming the update becomes)	Short-running process after initial load
Fast (up to one TB per hour)	Fast (up to one TB per hour)	Slower (in the range of GBs per hour)
Redundant load of table rows that have not changed, resulting in waste of processing resources (MIPS)	Redundant load of table rows that have not changed, resulting in waste of processing resources (MIPS)	No redundant load of table rows, no waste of processing resources.
Just a snapshot of the original data at some point in the past	Just a snapshot of the original data at some point in the past	Continual updates, recent data

Use cases

Incremental updates are the preferable method with regard to the following points of interest:

Realtime business intelligence
If realtime data is very important for your business intelligence operations, incremental updates are the preferable choice. Incremental updates do not guarantee that query data is the most recent available, but the latency is low, and this makes incremental updates a cost-efficient alternative.

Quick reporting on operational data
Incremental updates provide an uncomplicated, easy-to-use, and fast way to reporting on data in an operational data store (ODS) because the function is deeply integrated into your DB2 for z/OS and IBM DB2 Analytics Accelerator for z/OS setup. No up-front extraction and transformation of data is required to obtain results.

Cost-efficiency
Incremental updates are more cost-efficient than traditional update methods because there is no redundancy during updates and hence no waste of processing resources.

In addition, internal tests have shown that running incremental update processes affect the query response time only marginally.

High availability
Incremental updates can be configured in such a way that data changes in a DB2 for z/OS table are propagated to more than one accelerator-shadow

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

Related tasks:

"Loading tables" on page 93

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 accelerator-shadow 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 accelerator-shadow tables that have been included in the process.

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

See Figure 12 on page 139.

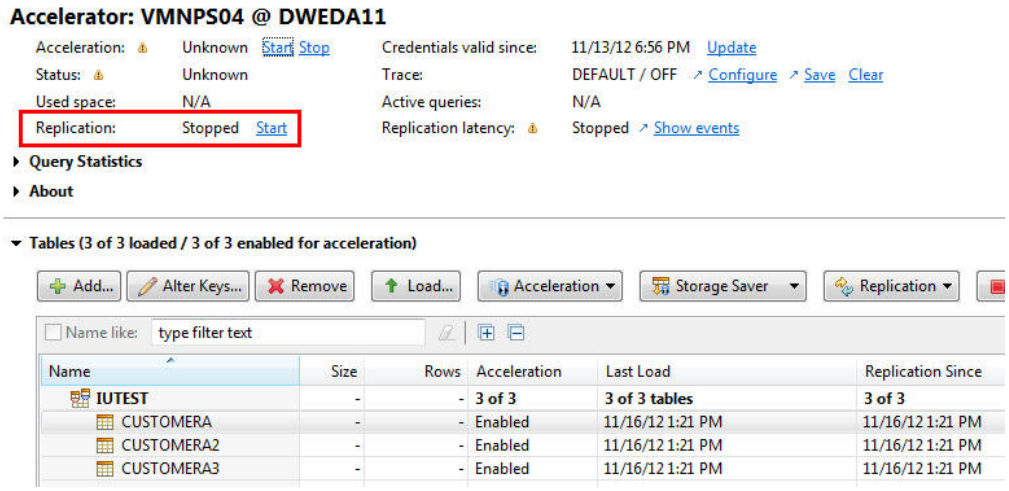


Figure 12. **Replication status** *and functional link in the header of the accelerator view*

Important: Using the functional links starts or stops the incremental update process for all tables that have been enabled for incremental updates on the selected accelerator.

Procedure

1. Start IBM DB2 Analytics Accelerator Studio and connect to a DB2 subsystem.
2. In the Administration Explorer, select the **Accelerators** folder.
3. In the Object List Editor on the right, double-click the accelerator that you want to include in the incremental update process.
4. Start or stop incremental updates by taking one of the following actions:
   - To start incremental updates, click **Start** in the header of the accelerator view. The **Replication status** in the header of the accelerator view changes to *Started*.
   - To stop incremental updates, click **Stop** in the header of the accelerator view. This opens the Stop Replication window, in which you can select between the following options for stopping the incremental update process:

     Controlled
     Completes all work in progress before stopping the process.

     Immediate
     Stops the process without completing work in progress. Compared with the **Controlled** option, using this option causes a restart of the process to take longer.

     Click **OK** after selecting an option. You return to the accelerator view, which now shows a **Replication status** of *Stopped*.

Including or excluding tables from incremental updates

Incremental updates can be enabled or disabled on the table level. 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

An accelerator-shadow table can only be included in the incremental update process if it contains data, that is, if it has been loaded.

When loading a table, make sure that the lock mode **TABLE_SET** is selected because this guarantees that data changes will not go amiss during the time between unloading the table data from DB2 to the accelerator and the point where the incremental update process sets in.

If you have enabled *continuous incremental updates* (a mode of operation in which incremental updates are not stopped when a table load begins), you can select lock mode **ROW** for a better performance. This lock mode effects a faster propagation of changes because the incremental updates are not put on halt when a table load starts and continue after the load, but are written to a spill queue, which is drained in the course of events. However, this comes at the price of a slower load process because lock mode **ROW** prevents the loading of the temporary table, which eventually replaces the former accelerator-shadow table. That is, loads are executed as a series of row changes. Decide whether faster loads or faster incremental updates are more important to you.

Attention:

- 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-shadow table. This, in turn, leads to incorrect results for accelerated queries.
- When lock mode ROW is set, DB2 data is unloaded at isolation level CS (cursor stability). Whether a row, a page, or an entire table is locked is determined by the LOCKSIZE defined for the table space of the source tables. To avoid deadlocks during the draining of the spill queue, set this table space to `LOCKSIZE ROW` if you want to use lock mode ROW for your incrementally updated tables.
- If you update DB2 for z/OS tables by running the LOAD utility rather than an INSERT, UPDATE, or DELETE operation, you must set the following parameters (for the LOAD utility):
  - `SHRLEVEL CHANGE`
  - `LOG YES`

  Otherwise, the changes that were made by the LOAD utility are not detected by the incremental update function, and will thus not be reflected in your accelerator-shadow tables.
- 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.
- You might have to reload or even remove tables from an accelerator after the execution of an ALTER TABLE or ALTER TABLESPACE statement in DB2 for z/OS. For more information, see *What to do after changing a table or table space in DB2 for z/OS*. You find a **Related reference** link at the end of this topic.

About this task

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

- An accelerator-shadow table has not yet been loaded, that is, it is in *Initial load pending* state. If you enable such a table, and load it immediately thereafter, the tables in DB2 and on the accelerator are in sync. No data changes will be lost because table contents are the same at the outset and the incremental update process captures all future changes and propagates these to the accelerator. Tables of this type show the same timestamp in the **Last Load** and **Replication Since** columns of the accelerator view. See the entry for the REP1 table in Figure 13 on page 142.
- An accelerator-shadow table has already been loaded when it is added to the incremental update process. In this case, it is likely that the tables are out of sync because data changes might have been applied to the original DB2 table, and not been reflected in the corresponding accelerator-shadow table because the last manual update of the table was before these changes happened. If a safety mechanism was not in place, such tables would continue to be out of sync because the incremental update process can only capture and propagate changes

that occur in the future. To avoid this, the status of such tables changes to *Initial load pending* when incremental updates are enabled. This state enforces a complete reload of the tables before they can be used, so that the synchronization gap is closed before the incremental update process sets in.

Important: If the load fails for some of the selected tables, incremental updates are automatically disabled for these tables.

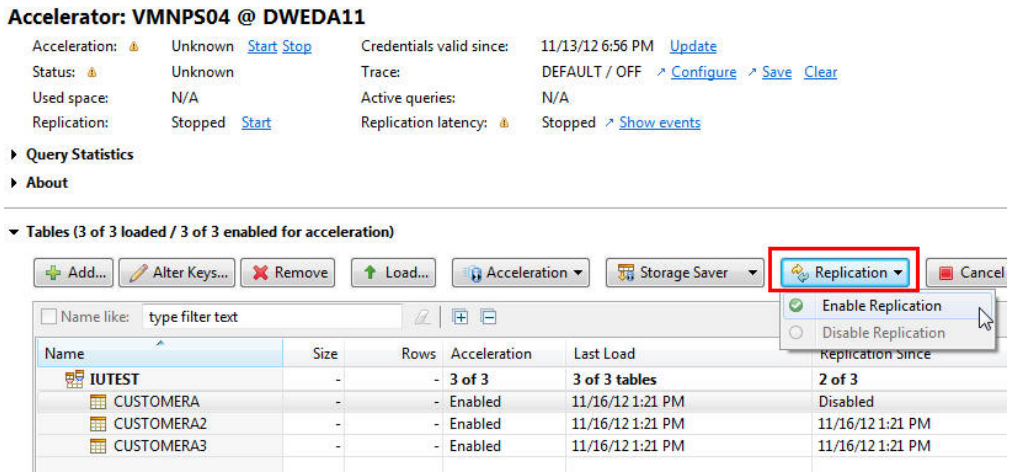


Figure 13. Timestamp columns and **Replication** *button in the accelerator view*

If continuous incremental updates are enabled (see *Enabling continuous incremental updates* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*, the **Replication Since** column shows the following states in the order indicated as soon as a load process begins:

1. *Suspended*
2. *Refreshing*

The *Refreshing* state is shown until the last change has been drained from the spill queue. When you move the mouse pointer over the *Refreshing* label, the current processing phase is displayed in a tool-tip window. The refresh process consists of the following phases:

Pending
: The table load process was started, but no data has yet been transferred. Therefore, only the data that has been transferred before the refresh is available for accelerated queries.

Spilling
: The table load is in progress, and data is being transferred. Incremental updates that occur during this time are written to the spill queue. Therefore, only the recently transferred load data is available for accelerated queries.

Draining
: The load has finished and the changes that have been collected in the spill queue are being applied. Only the data changes that have already been transferred to the table are available for accelerated queries.

  Tip: If a table remains in the draining phase for several hours, you are probably in the very rare situation in which new changes arrive in the spill

queue faster than they can be drained from it, so that the draining phase never ends (inifinite drain). In this case, stop incremental updates completely for a while.

Restriction: It is not possible to enable tables if the table name or the schema name contains GB18030 characters (Simplified Chinese) of Unicode plane 2 (U+20000-U+2FFFF: Supplementary Ideographic Plane). Trying to do so results in an error.

Procedure

1. Select the accelerator-shadow tables to be included or excluded.
2. Include or exclude tables from the process:
   - To include selected tables in the incremental update process, click **Replication** > **Enable Replication**. See Figure 13 on page 142 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. Define the accelerator-shadow table.
   b. Enable incremental updates for this table.
   c. Load the accelerator-shadow table.
   d. Start incremental updates (**Replication** > **Start**).
   e. Reload the accelerator-shadow table.

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

Related reference:

"What to do after changing a table or table space in DB2 for z/OS" on page 148

Related information:

CREATE TABLESPACE

Monitoring incremental updates

You can monitor the changes to accelerator tables that result from incremental updates. The header of the accelerator view in IBM DB2 Analytics Accelerator Studio provides an additional line of information when incremental updates have been configured for the accelerator. This line of information is labeled **Replication latency**.

One of the following values is displayed next to this label:

Stopped
: Incremental updates have been stopped for the entire accelerator.

Low
: It will take less than three minutes to propagate recent changes to the corresponding accelerator-shadow tables.

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

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

When you move the mouse pointer over the latency value, a pop-up window shows the following information:

Target Agent
Operations completed by the incremental update function.

Inserts
Number of row inserts in accelerator-shadow tables that have been included in the incremental update process.

Deletes
Number of row deletions from accelerator-shadow tables that have been included in the incremental update process.

Remember:

- An UPDATE is a DELETE followed by an INSERT, so these two figures account for all incremental updates that have been applied.
- Since an UPDATE in DB2 is a DELETE followed by an INSERT on the accelerator side, the values can be different from those in DB2 for z/OS (see **Source Agent**). Furthermore, multiple operations on a single row within a short time might be reported as just one single operation.
- If a load operation is in progress and lock mode **Row** has been selected, the original DB2 tables are not locked during the load. Hence it is possible to INSERT new rows into the DB2 tables even though a load is taking place. If incremental updates are enabled for the same tables, the incremental update function stores these new rows in the spill queue until the load has finished, and applies these changes thereafter (draining the spill queue). In the course of a drain, a DELETE and an INSERT operation is carried out for each row in the affected accelerator-shadow tables. The DELETE deletes the row that was inserted by the previous load operation, and the INSERT adds this row again (applies the latest change) . This has the effect that the number **Deletes** increases in cases like this.

See Figure 14 on page 145.

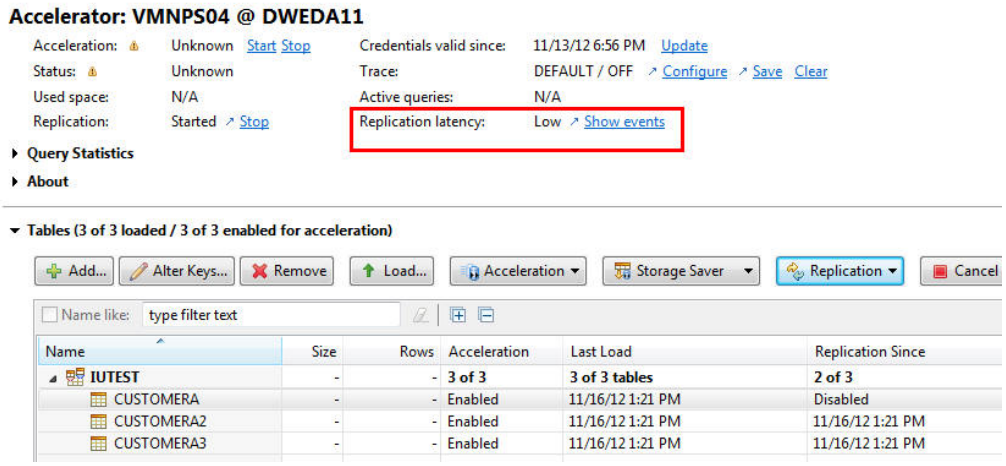


Figure 14. **Replication latency** *in the accelerator view*

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 135

Warnings related to incremental update processes

The following warnings might occur in connection with incremental updates.

InsufficientLockModeWarning

This warning is issued if you did not select the option **Lock DB2 tables while loading** for the initial load of an accelerator-shadow table taking part in the incremental update process, or if you selected the lock modes **NONE** or **PARTITION**.

If locks were disabled for the initial load of an accelerator-shadow table (**Lock DB2 tables while loading** not selected), the following message is displayed:

```
One of the tables has replication enabled. This table must be locked while
loading to avoid an inconsistency of data between DB2 and the accelerator.
```

If the lock modes **NONE** or **PARTITION** were used, data consistency between tables in DB2 and accelerator-shadow tables cannot be guaranteed

because table changes can be applied to the DB2 tables during the unload process. If possible, use the lock mode **TABLE_SET**.

NoUniqueConstraintsDefinedWarning
A table does not have a unique constraint, such as a unique key or unique index. This might result in a poor performance when UPDATE or DELETE changes are propagated to an accelerator-shadow table (because it takes longer to identify the changed table rows).

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-shadow table if just a single row is deleted from a DB2 table. That is, all rows with identical values will be deleted from the accelerator-shadow 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-shadow table or exceed the number of matching organizing keys. This might lead to a slow propagation of data changes to the accelerator. In the case of different unique constraints, it just takes longer to identify the table rows to be updated or deleted. A mismatch between the unique constraints of the table and the organizing keys affects the change propagation negatively if there are more unique constraints than matching organizing keys. If extents cannot be located for certain table rows because there is no matching organizing key, table scans take longer.

Collecting CDC trace information

None of the options in the Save Trace window of IBM DB2 Analytics Accelerator Studio collects trace information about the CDC Capture Agent. This information, however, might be the key to a problem with the incremental update function.

About this task

If you are facing a problem with the incremental update function, provide as much information about the related software components as possible the support team. This includes information about the CDC Capture Agent, which is the part of IBM InfoSphere Data Replication for z/OS that runs on z/OS. The support team might even ask you for this type of information. Here is a description on how to obtain it.

Procedure

1. The CDC Capture Agent is just one of the started tasks running on your z/OS system. If you do not know the name of the started task, ask a z/OS systems programmer which task that is.
2. In the SDSF program on z/OS, type `XDC` in front of the relevant address space output to copy its contents to a sequential data set.
3. Transfer this data set in ASCII mode to your Linux or Windows workstation and add the resulting .txt file to your problem management record (PMR).
4. If a trace dump has been generated, include that information as well. Transfer this data in TERSED format to your workstation because this retains the mainframe-readable formatting of the data during the transfer to a

non-mainframe system, and thus ensures that the data remains useful for IBM support. Use an extension of .trs for the files. The TERSED format also reduces the file size. See the documentation for the AMATERSE program if you need further assistance.

What to do after changing a table or table space in DB2 for z/OS

In a number of cases, an ALTER TABLE or ALTER TABLESPACE statement in DB2 for z/OS does not involve maintenance work on your accelerator-shadow tables. There are cases, however, in which it does. See which steps you might have to take.

NOT Programming Interface information

ALTER TABLE

- **For users of the incremental update function only:** You need not take action 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 removal, redefinition, re-enablement, or reload of the related accelerator-shadow table.
- If a column was added, changed, or deleted by an ALTER TABLE operation on a DB2 table, queries against this table are not accelerated if they reference the new, changed, or deleted column (reason code 14). 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 corresponding accelerator-shadow table from all accelerators on which it is defined.

     In most cases, a REORG operation is not required before the table can be redefined on the accelerator. However, a REORG is required in the following cases, although this seems to be just a recommended action, as the advisory *REORG-pending* (AREO) state suggests:
     - The ALTER TABLE operation changed column data types.
     - The ALTER TABLE operation dropped columns in *DB2 11 for z/OS*.
  2. Redefine the table on the accelerators from which you previously removed it.
  3. If the incremental update function is used, 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.

ALTER TABLESPACE

If a table space was changed in DB2 for z/OS (ALTER TABLESPACE), and the tables of that table space also exist on an accelerator, actions are required under the following conditions:

- If the maximum number of partitions has changed (MAXPARTITIONS), you only have to reload the new or changed partitions. It is not required to reload existing partitions that have not changed.
- If a table space was changed from *segmented* to *partitioned by growth*, you must remove the accelerator-shadow tables of that table space from all accelerators,

then redefine and re-prepare these for accelerated queries or incremental updates. Follow steps 1 on page 148 through 5 on page 148 to do that.

End of NOT Programming Interface information

Freeing up storage in DB2 for z/OS

The high-performance storage saver (HPSS) archives data of table partitions in DB2 for z/OS to an accelerator. In moving data that is no longer actively used (historical data) to a less expensive storage device, you can free up costly storage space on your database server.

Details

The HPSS is a fully integrated function. Its use has the positive side-effect that it also makes the database system more responsive. This is because fewer objects need to be maintained in the catalog, smaller or fewer indexes need to be searched, reorganizations become quicker, report data and statistics can be gathered faster.

If a query could be accelerated while the original data resided in DB2, it can also be accelerated when the data has been archived on an accelerator.

Just like before, the query routing process is handled by the DB2 query optimizer and IBM DB2 Analytics Accelerator. If data has been archived to an accelerator, and an incoming query needs to access this data, and if furthermore the query qualifies for acceleration, then the query is automatically processed by IBM DB2 Analytics Accelerator.

You can determine whether archived data is to be included in queries. By default, this type of data is excluded, and IBM DB2 Analytics Accelerator for z/OS automatically skips archived records when it processes a query. This often leads to different results for queries that are run repeatedly because the query is run just against the remaining DB2 data. If this behavior is not wanted, you can set the `CURRENT GET_ACCEL_ARCHIVE` special register or the `GET_ACCEL_ARCHIVE` configuration parameter to `YES`.

You can reload accelerator tables without thinking about archived partitions. Partitions that have been archived are automatically excluded from the load process.

The HPSS is flexible. It is still possible to define one and the same query table on multiple accelerators. It is also possible to archive the data of one or more partitions on an accelerator, while the data of other partitions remains in DB2 for z/OS and on the same or other accelerators.

Before any data is moved to an accelerator and deleted from DB2, image copies are created, which allow you to restore the data if needed. The table spaces of the affected (archived) partitions are set to a persistent read-only state, which prevents future INSERT, UPDATE, and DELETE operations on these partitions. Furthermore, the HPSS ensures that the data in the image copies is consistent with the data that is going to be archived.

The table space of an archived partition continues to exist. It will be empty, but will still claim as much disk space as defined by the *minimum primary space allocation*. To meet your space-saving goal, check the minimum primary space allocation (PRIQTY) for the table spaces of the partitions that you want to archive.

Decrease the value if needed by submitting an appropriate `ALTER TABLESPACE ... PRIQTY` statement.

Restrictions

The following restrictions exist in connection with the HPSS:

- The HPSS works on range-partitioned tables only. If the partitioning is controlled by an index, the index must (already) exist.
- The smallest unit that you can archive is a partition. That is, all the table rows in specified partitions are copied to the accelerator and are finally removed from DB2.
- You can only archive partitions of tables that exist as accelerator-only tables and that are in *InitialLoadPending* or *Loaded* state. This implies that partitions cannot be archived if incremental updates have been enabled for the table.
- Table data cannot be archived if a column in the table serves as the parent in a foreign-key-relationship. The reason for this is that the original data is deleted at the end of the archiving process, which removes the values of the foreign key. This cannot be permitted because it violates a fundamental rule for data integrity in a relational database.
- Normally, you cannot rotate a DB2 partition or change partition boundaries if the data of this partition data has been archived on an accelerator. Commands to rotate a partition or change partition boundaries will be rejected by DB2 for z/OS. This restriction can be circumvented by first removing the Partition Read-Only (PRO) state with the help of the DB2 REPAIR utility. If you rotate a partition after removing the PRO state, you are still able to recover the archived partitions by using the image copies, except for those partitions that have been ⌂rotated out⌂. If you change the partition boundaries, on the other hand, the image copies become unusable and the data cannot be restored. You will also end up with orphaned data on the accelerator.

  Attention: A rotation leads to the deletion of the data in those DB2 partitions that have been "rotated out". In addition, the corresponding image copies, which were created when the data was archived, will become unusable. The first archiving operation on the table that used to contain the "rotated -out" partitions will also delete the data of the "rotated-out" partitions from the accelerator, irrespective of the partitions that you specified for archiving.
- Starting with product version 4.1 PTF-3, you can archive partitions that contain *DB2 archive tables*. However, you cannot archive *DB2 archive-enabled tables*. Do not confuse these tables with partitions and tables that were archived on an accelerator by the HPSS (accelerator-archive tables). If necessary, check the DB2 documentation for definitions of the terms *DB2 archive table* and *DB2 archive-enabled table*. Bear in mind also that when a DB2 archive table has been processed by the HPSS, you can no longer shift rows from the corresponding DB2 archive-enabled table to the DB2 archive table or vice versa. A message with `SQLCODE=-904` is returned when you try to do that.
- No failback mechanism is in place in case an error occurs. Records that have already been archived are not rolled back. The DB2 catalog explicitly marks the partitions containing these records as “archived”.
- Columns that cannot be loaded into accelerator tables because their data types are not supported can also not be archived. These are basically the following data types:
  - DECFLOAT
  - TIMESTAMP(*n*) where *n* has a value other than 6.
  - ROWID
  - User-defined types that are based on the above

However, an attempt results in the issuing of a warning. The rest of the partition or table data will be archived. Although the original data is deleted after archiving, the data in unsupported columns is not lost because it is saved to the image copies.

The following data types, on the contrary, lead to a failure of the archiving operation:
– BLOB
– CLOB
– DBCLOB
– XML

- If the GET_ACCEL_ARCHIVE special register (or ZPARM) has been set to the value YES, and a query addresses tables whose data has been archived to different accelerators, the query fails.
- For an archiving operation to succeed, a shared lock is required on the involved tables or partitions. Long-running transactions, such as database queries, might prevent the lock from being obtained. Make sure that long-running transactions are finished before you start archiving.

Archiving partition or table data with the High-Performance Storage Saver

The functionality of the High-perfomance Storage Saver (HPSS) is fully integrated into IBM DB2 Analytics Accelerator Studio. See how to archive the data of partitions or entire tables from the Accelerator view.

Before you begin

1. Make sure that you have read the introduction to the HPSS and the list of restrictions in the superordinate topic.
2. For the creation of the image copies, the AQTENV data set must be referenced by the Workload Manager (WLM) environment that has been set up for the IBM DB2 Analytics Accelerator for z/OS stored procedures, and a user with sufficient access rights must have set at least one of the following environment variables in this data set (a z/OS systems programmer or database administrator usually has the required access rights):
   - AQT_ARCHIVE_COPY1
   - AQT_ARCHIVE_COPY2
   - AQT_ARCHIVE_RECOVERYCOPY1
   - AQT_ARCHIVE_RECOVERYCOPY2

   These variables determine how many local image copies and recovery image copies are created for each archived partition and how the data sets are named. If you set a variable, a corresponding data set is created when partition data is archived. At least one of the variables must be set. Default values do not exist. Each value is a template specification as used in the DB2 TEMPLATE utility, for example:

   ```
   AQT_ARCHIVE_COPY1 = &USERID..&DB..&TS..P&PART..&UNIQ.
   ```

   where

   &USERID.
   ID of the user who runs SYSPROC.ACCEL_ARCHIVE_TABLES

   &DB.
   Name of the database that a partition resides in

   &TS.
   Name of the table space that the partition resides in

&PART.
Identifier of the (physical) partition. The letter `P` in the example is a text or string constant used as a prefix. This is required because &PART. resolves to a numeric value like 00001, and this not valid for qualifiers in a data-set name.

&UNIQ.
Causes the creation of a unique identifier

The template specification in the example could, for instance, result in the following image copy name:

```
BCKE.V4L1.BCKERTSE.CKRANGE3.P00001.D72R4KHN
```

- All template variables that are documented for the DB2 COPY utility can be used, with the exception of &SEQ (&SQ), &LIST (&LI), and &DSNUM.
- The chosen variables must ensure the uniqueness of image-copy data-set names. It is therefore recommended that you use at least the &PART. and &UNIQ. template variables.
- Templates must resolve to valid z/OS data set names.
- The template data-set names that you use must have been mapped to suitable data classes in the DFSMS.

Important: The AQT_ARCHIVECOPY_HLQ environment variable that was used with earlier versions of the HPSS is deprecated. Remove it from the AQTENV data set.

3. The HPSS stored procedure SYSPROC.ACCEL_ARCHIVE_TABLES creates image copies whose size might exceed 65535 tracks on direct access storage devices (DASDs). Therefore, a z/OS systems programmer must define and use a data class in the storage management subsystem (SMS) that supports larger sequential data sets and must associate this data class with the range of template data-set names for image copies. Data set types that can be used to this end are shown in Table 32.

Table 32. 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

Table 32. Suitable data-set types for HPSS image copies and corresponding SMS data-class settings (continued)

Data-set type	Settings in SMS data class	Properties of data-set type
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 151, the Workload Manager (WLM) environment for the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure must be refreshed. Make sure that this has been done.
5. If you intend to archive partitions from a DB2 data sharing group, make sure that the following conditions apply:
   a. The volumes or devices for the storage of image copies must be available for all members of the data sharing group.
   b. The values of the environment variables mentioned in step 2 on page 151 must be identical on all members of the data sharing group.
6. Nonpartitioning indexes might slow down the archiving process. If more than just a few small to mid-sized nonpartitioning indexes are defined for a table, but the partitions to be archived are not many and rather large, then consider dropping the indexes before archiving. Re-create the indexes thereafter.
7. If you also use the incremental update function, check the setting of the `ONUTILITYACTION` keyword in the IBM InfoSphere 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 archived by the HPSS. Hence the parts that have not been archived will not be synchronized anymore. To avoid this conflict, set the keyword to the value `IGNORE`.

About this task

The archiving task is carried out by the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, follow the appropriate **Related reference** link at the end of this topic.

Note: It might take a long time to archive many partitions. If you use IBM DB2 Analytics Accelerator Studio to invoke the ACCEL_ARCHIVE_TABLES stored procedure, IBM DB2 Analytics Accelerator Studio must remain open until the task has been completed. Otherwise, the stored procedure aborts, in which case it might be difficult to recover from the errors that were caused by the abortion. To avoid this problem, invoke the stored procedure from a TSO batch job. You can also repeatedly archive a smaller set of partitions at a time. Before you invoke the function, make sure that you have enough disk space for the image copies.

Procedure

1. In IBM DB2 Analytics Accelerator Studio, open the Accelerator view of the accelerator that you want to use for archiving.

   Note: You can only archive partition or table data to a regular accelerator. Therefore, you must open the Accelerator view of a regular accelerator.
2. If the tables are not yet defined as accelerator-shadow tables, click **Add** to start the Add Tables wizard and define these tables.
3. If necessary, refresh the Accelerator view.
4. To archive the data of a range-partitioned table or partitions of this table, expand the appropriate schema node and select the table name in the **Name** column.
5. On the toolbar of the Accelerator view, select **Storage Saver** > **Move Partitions to Accelerator**.
6. In the Move Storage Saver Partitions to Accelerator window, you can select a partitioning key or individual partitions.
   - To select the partition data to be archived with the help of a partitioning key, leave **Move all partitions up to and including the following limit key** selected and choose the proper key from the drop-down list.

     The drop-down list is filled with the values of the LIMITKEY column in the SYSIBM.SYSTABLEPART catalog table. The values are sorted by the LOGICAL_PART column in SYSIBM.SYSTABLEPART and appear in descending order. Mostly, LIMITKEY is a column of the type DATE or DATE/TIME, which allows you to select partitions whose data covers a certain timeframe. The key that you use might be different, and so might be the choice in the drop-down list.
   - To manually select the partitions whose data you want to archive, click **Manually select partitions** and select the appropriate check boxes in front of the (logical) partition names. The names of partitions whose data has already been archived are grayed out. To select the entire table, click **Select All**. Clicking **Deselect All** deselects all table partitions, which allows you to correct your choice.
7. Click **OK**.

What to do next

Check with your database operators (the people running queries) whether the archived data needs to be accessed by their queries. Whether to include or exclude the archived data can be controlled by using the **GET_ACCEL_ARCHIVE** parameter (ZPARM) or the SET CURRENT GET_ACCEL_ARCHIVE special register. The parameter or special register can be set to the value YES or NO. If you set it to YES, the archived data will be available for queries.

The **GET_ACCEL_ARCHIVE** parameter is set on the DSN6SPRN panel of a DB2 subsystem. The systems programmer or database administrator needs to enable option 2.

The SET CURRENT GET_ACCEL_ARCHIVE special register can be prepended to the SQL script for a query.

Example: Suppose you have a table called ORDERS, which contains a history of purchase transactions. This table is partitioned by time and contains a partition for each quarter. You keep ten years of history, but only a small subset of the queries accesses older data. Most queries use a range predicate, which restricts the query to data of the current year. You decide to archive all data that was collected in the previous years. For the queries that access the current year's data only, you need not set the SET CURRENT GET_ACCEL_ARCHIVE special register because these queries need not "see" the archived data. They can run on the accelerator or in DB2 for z/OS, depending on the SET CURRENT QUERY ACCELERATION special register. For those queries that need to access older data, you specify `SET CURRENT GET_ACCEL_ARCHIVE=YES`. For the latter type of queries, you must also set the SET CURRENT QUERY ACCELATION special register to a value other than NONE. The query cannot run in DB2 for z/OS because DB2 does no longer contain the data.

Related information:

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

Impact of special register settings

It has been said that a query cannot be accelerated without setting the `SET CURRENT QUERY ACCELERATION` special register to a value other than `NONE`. It has also been said that the `GET_ACCEL_ARCHIVE` or the `SET CURRENT GET_ACCEL_ARCHIVE` special register determines whether data that was moved with the help of the high-performance storage saver (HPSS) is included in a query. The table in this topic explores all combinations of these settings for qualifying and nonqualifying queries and briefly describes the consequences.

If a query can neither be executed in DB2 for z/OS, nor on an accelerator, the following message is displayed, which starts with SQL code `-4742`:

```
-4742 The statement cannot be executed by DB2 or in the accelerator (REASON <code>)
```

where `<code>` stands for the numeric reason code. For a list of the reason codes, follow the link at the end of this topic.

Table 33. 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.

Table 33. Impact of special register settings on the query execution, taking the query qualification into account (continued)

Setting of `SET CURRENT QUERY ACCELERATION`	Setting of `GET_ACCEL_ARCHIVE` or `SET CURRENT GET_ACCEL_ARCHIVE`	Query qualifies for acceleration?	Consequence
ENABLE	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.
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.

Table 33. Impact of special register settings on the query execution, taking the query qualification into account (continued)

Setting of `SET CURRENT QUERY ACCELERATION`	Setting of `GET_ACCEL_ARCHIVE` or `SET CURRENT GET_ACCEL_ARCHIVE`	Query qualifies for acceleration?	Consequence
ELIGIBLE	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 33 on page 155 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 107

Restoring archived partitions

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

About this task

In the context of archived partition data, the restore operation is typically just a single operation in a series of actions that must be taken. For example, adding a column to a DB2 table involves the following steps:

1. Restoring the archived partitions to DB2
2. Removing the corresponding accelerator-shadow table
3. Making changes to the schema, that is, submitting the `ALTER TABLE ADD COLUMN` statement
4. Redefining the accelerator-shadow table
5. Archiving the partition data once again

What you need to know about the restore function:

- The restore function uses the last DB2 image copy to recover the data. If that image copy is damaged, the function automatically uses the next image copy in backward chronological order.

- All indexes are rebuilt.
- A CHECK DATA operation is carried out if needed.
- An archived partition is not deleted from the accelerator, but kept as a regular partition for query acceleration.
- All image copies and entries in the DB2 catalog table SYSIBM.SYSCOPY are kept.
- The status information of tables in the ARCHIVE column of the SYSIBM.SYSACCELERATEDTABLES catalog table is restored if no further copies exist on any accelerator. That is, the values `'A'` or `'C'` are reset to `''`.
- If a restore operation fails for some reason, you can resume it if you run the function again on the same table or partitions.
- Depending on the amount of data to be restored, the operation might need up to several hours to complete.

  Attention: Do not sever the connection to the DB2 subsystem during that time.
- If you have archived the same partition more than once on different accelerators, you can run a restore operation for this partition from each of the accelerators. This is not prohibited. However, the DB2 data is not updated anymore after the first restore because the system treats it like an already restored partition. So if the data is different on the accelerators, carefully decide from which accelerator you want to restore first.
- Bear in mind that queries might run longer and might return more copious results after a restore because the formerly archived partitions are no longer excluded from the queries.

The task of restoring partition or table data to DB2 for z/OS is carried out by the SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the corresponding article in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. You find a link at the end of this topic.

Procedure

1. Check whether the tables that the partitions belong to have been enabled for incremental updates. If so, disable incremental updates. You might come across this situation if just a subset of all the partitions of a table was archived. If unarchived partitions remain on the accelerator, the relevant table can be enabled for incremental updates. However, a restore operation fails if this has been done.
2. In IBM DB2 Analytics Accelerator Studio, open the Accelerator view of the accelerator that you want to restore data from.
3. If necessary, refresh the Accelerator view.
4. On the toolbar of the Accelerator view, select **Storage Saver** > **Restore Partitions to DB2**.
5. In the Restore Storage Saver Partitions to DB2 window, you can select tables and partitions to be restored. Select the check box in front of a table name to restore the data of an entire table (all partitions). Select the appropriate check boxes in front of (logical) partition names to restore just the data of the selected partitions. To select all tables and partitions that can be restored, click **Select All**. Clicking **Deselect All** resets a previous selection, so that you can start over.

   When you highlight a partition name, the **Image copies** section displays the names of the image copies (data set names) that will be used to restore the partition.

6. Click **OK**.

What to do next

The recovery of partitions invalidates the statistics in the SYSIBM.SYSTABLESPACESTATS catalog table. This leads to a situation in which table metrics, such as the partition size, are not reported anymore. To avoid this, run the following DB2 command to refresh the statistics in the catalog table:

```
RUNSTATS TABLESPACE <database-name>.<table-space name> SHRLEVEL REFERENCE
```

where *<database-name>* is the name of the database and *<table-space name>* is the name of the table space.

Note: This step might take up to several hours to complete. If you plan to archive the partition data again shortly, you can omit it.

Related information:

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

How to handle the RBA and LRSN conversion in DB2 11 for z/OS

In DB2 11 for z/OS, the relative byte address (RBA) and the log record sequence number (LRSN) were changed from a 6-byte format to a 10-byte format. This has implications if you have archived partition data with the high-performance storage saver (HPSS). In DB2, the migration from the old format to the new requires a REORG operation. However, the empty partitions that remain after HPSS archiving cannot be reorganized because they were set to persistent read-only (PRO) state.

To maintain the compatibility of archived tables and partitions, follow the series of steps in the following table that best suits your needs:

Table 34. Steps to maintain compatibility

1. Remove the PRO state with the help of the DB2 REPAIR utility. 2. Run the REORG operation. 3. Reset the PRO state. This is the recommended solution. It saves you a lot of time because you avoid to restore data. Note that you can restore the data at a later time into already converted tables and partitions even if the image copies were taken before the RBA and LRSN conversion. A further REORG operation will be needed after that.	1. Restore the data of the archived tables and partitions from the existing image copies. 2. Run the REORG operation. 3. Archive the data again with the HPSS. If you do not want a coexistence of data in different RBA and LRSN formats, then this solution is to be preferred. Bear in mind that this will occupy system resources for a much longer time.

Federated access

The term *federated access* or *federation* describes the ability to run accelerated queries against tables that do not belong to or originate from the DB2 subsystem that issues the query.

Normally, you can only accelerate queries if a DB2 subsystem is paired with an accelerator and if the tables to be queried exist in that DB2 subsystem. This can be

regarded as a security feature; a second DB2 subsystem paired with the same accelerator cannot access tables in the first DB2 subsystem and vice versa.

However, there are situations in which one would like to lift this restriction: DB2 subsystems are often organized alongside geographical boundaries, temporal boundaries, or areas of business.

Geographical boundaries translate into a segmentation of the table data by region. For example, a company does business in the U.S. and in Asia; the tables in subsystem A contain the sales figures for the U.S.; the tables in subsystem B the sales figures for Asia.

If data is temporally segmented, the tables in each subsystem cover different time spans. For example, the tables in subsystem A contain the data related to recent business activities, reaching three months from the present time into the past. DB2 subsystem B, on the other hand, contains business data that is older than three months.

If data is segmented by areas of business, the organization of the database follows a company's key business areas or its product portfolio. Assume that an organization runs a chain of gas stations and also produces cosmetics (both business areas are based on oil). Subsystem A contains the data of the gas station business; subsystem B the data of the cosmetics business.

Frequently, the aim of analytical queries is to capture trends across regions, time lines, or business areas. In accordance with the previous examples, one would like to find out how the business in the U.S. develops against the business in Asia. How do recent sales figures compare with those of the past? What are the growth rates in the gas station business as opposed to those in the cosmetics sector?

Queries that answer these question require access to different data sources. If the data is segmented by the organization of the database, access must be extended to tables in other DB2 subsystems, and this is what the *federated access* feature is about.

An access extension to tables in a DB2 subsystem is achieved through the following steps:

1. A user with sufficient rights on the DB2 subsystem that owns the tables (remote DB2 subsystem) grants the right to access a set of tables to a DB2 subsystem that normally cannot access these tables (called the referencing DB2 subsystem).
2. A user with sufficient rights on the referencing subsystem runs a command to create reference tables. These reference tables are copies or snapshots of accelerator-shadow tables. They are created as accelerator-only tables on the shared accelerator, with corresponding entries in the catalog of the referencing DB2 subsystem.
3. Queries from the referencing DB2 subsystem can now access these referencing accelerator-only tables.

See the following figure:

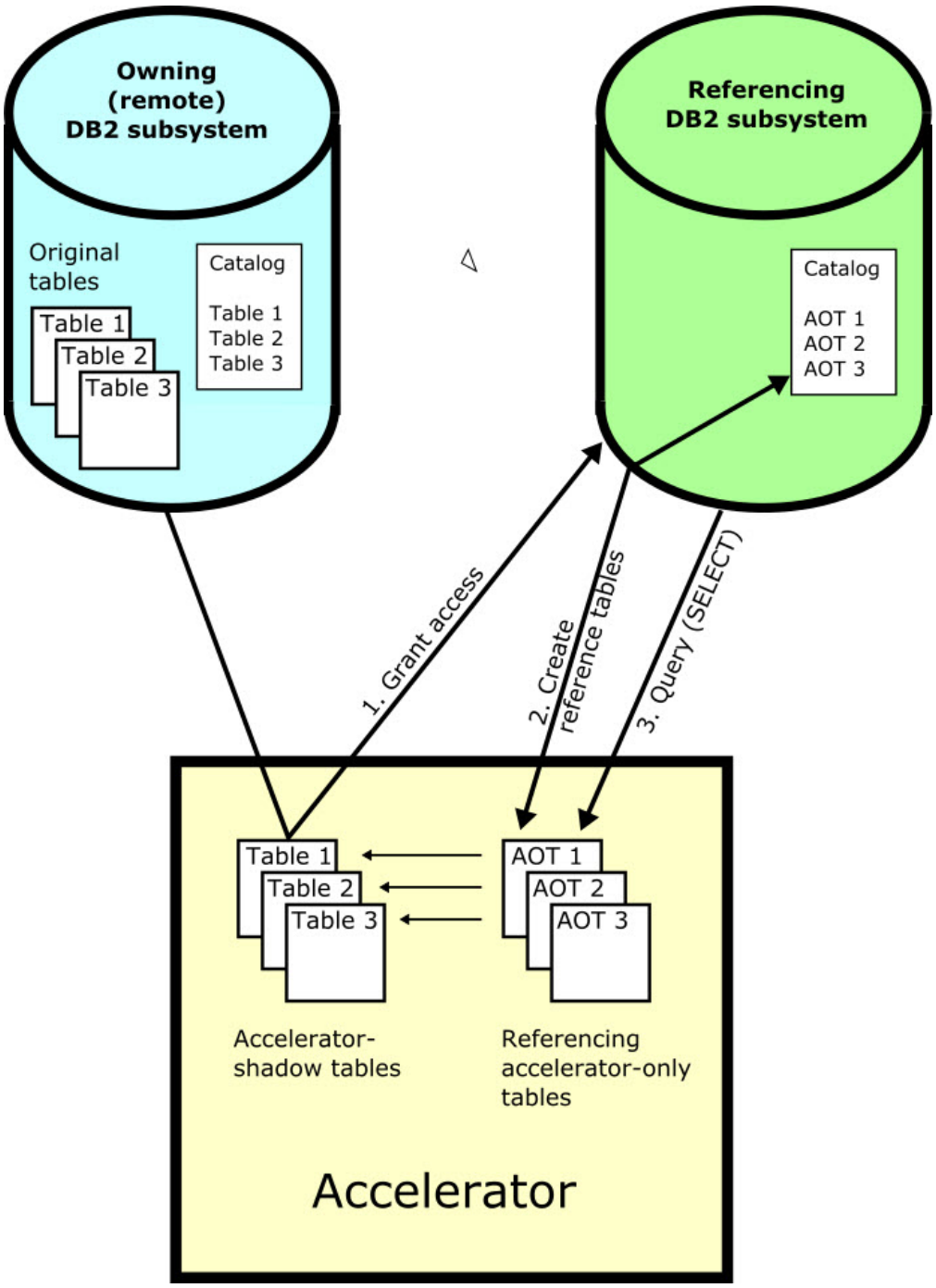


Figure 15. Federated access diagram

It is possible for the administrator of the owning DB2 subsystem to revoke the access privileges at any time.

It is also possible (for the administrator of the referencing DB2 subsystem) to remove the reference tables. This is mostly done when the data in the accelerator-only tables has become obsolete. To obtain results on more recent data, the administrator might want to delete the existing reference tables and create new ones.

Granting access to remote tables

As an administrator or user with sufficient access rights on a DB2 subsystem, you can grant the right to access tables in the subsystem to another (referencing DB2 subsystem). The granted permission includes the right to create reference tables from the referencing DB2 subsystem.

Before you begin

- The owning (remote) DB2 subsystem and the referencing DB2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both DB2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_GRANT_TABLES_REFERENCE stored procedure on the owning (remote) DB2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with the owning (remote) DB2 subsystem. Mind that although the referencing DB2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables in the owning DB2 subsystem.
2. On the toolbar of the **Tables** section, click **Federation** > **Grant Access**.
3. In the DB2 Tables window, select the tables that you want to give access to, that is, accelerator-shadow tables or ordinary accelerator-only tables in the owning (remote) DB2 subsystem.
4. Click **Next**.
5. In the DB2 Subsystems window, you see a list of the DB2 subsystems that are connected to the same accelerator. Select one or more referencing DB2 subsystems, that is, subsystems that should have access to the selected tables.
6. Click **Finish**.

Results

The administrator of the referencing DB2 subsystem can now create reference tables for the tables that she or he has been given access to.

Creating reference tables

Running the **Create Reference Tables** function in IBM DB2 Analytics Accelerator Studio, you can create copies (referencing accelerator-only tables) of tables in a remote DB2 subsystem, provided that access to these tables has been granted before.

Before you begin

- The owning (remote) DB2 subsystem and the referencing DB2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both DB2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_CREATE_REFERENCE_TABLES stored procedure on the referencing DB2 subsystems.

About this task

The copies (also called reference tables) are used when a query from a referencing DB2 subsystem includes the remote tables. The query thus goes against the copies or reference tables rather than the accelerator-shadow tables or the (regular) accelerator-only tables that belong to the remote (owning) DB2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with a referencing DB2 subsystem. Mind that although the owning or remote DB2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables that access has been granted to.
2. On the toolbar of the **Tables** section, click **Federation** > **Create Reference Tables**.
3. In the DB2 Subsystems window, you see a list of DB2 subsystems that qualify as referencing DB2 subsystems, that is, subsystems that will query the remote tables. Select one or more of these subsystems.
4. Click **Next**.
5. In the DB2 Tables window, you see a list of tables on the accelerator. Select the tables for which you want to create reference tables by selecting the appropriate check boxes. To select all tables that belong to a schema, select the check box in front of the schema name. If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.

   To the right of the **Name like** field, you also find controls (icons) to do the following:

   a. Select or deselect all tables
   b. Expand or collapse all schemas
6. Selecting the optional check box **Grant access before creating reference tables** allows you to complete the required grant step and the creation of the reference tables in one go. However, you can only use this check box if your user ID has the authorization to run the SYSPROC.ACCEL_GRANT_TABLES_REFERENCE stored procedure on the owning (remote) DB2 subsystem.
7. Click **Finish**.

Results

Reference tables (referencing accelerator-only tables) for the selected tables are created on the shared accelerator and corresponding entries are written to the catalog of the referencing DB2 subsystem. The new tables can be queried from the referencing DB2 subsystem.

Revoking table access

Run the **Revoke access** function to withdraw the right to access remote tables from a referencing DB2 subsystem.

Before you begin

- The owning (remote) DB2 subsystem and the referencing DB2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both DB2 subsystems.

- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_REVOKE_TABLES_REFERENCE stored procedures on the owning (remote) DB2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with the owning (remote) DB2 subsystem. Mind that although the referencing DB2 subsystem is paired with the same accelerator, it will use a different accelerator view, which will show a different list of tables. Make sure that you select the view which will display the tables in the owning DB2 subsystem.
2. On the toolbar of the **Tables** section, click **Federation** > **Revoke Access**.
3. In the DB2 Subsystems window, you see a list of the DB2 subsystems that are paired with the same accelerator. Select one or more referencing DB2 subsystems that should no longer be allowed to access remote tables in the owning DB2 subsystem.
4. Click **Next**.
5. In the DB2 Tables window, select the tables that you want to withdraw access from. Select by clicking the appropriate check boxes. To select all tables that belong to a schema, select the check box in front of the schema name. If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.

   To the right of the **Name like** field, you also find controls (icons) to do the following:

   a. Select or deselect all tables
   b. Expand or collapse all schemas
6. Click **Finish** when your selection is complete.

Results

Access rights to the selected tables are revoked. Even though the corresponding reference tables are not deleted, the referencing DB2 subsystem can no longer query these tables. Attempts will be answered with an authorization error.

Removing reference tables

Run the **Remove reference tables** function to delete reference tables (referencing accelerator-only tables) from an accelerator

Before you begin

- The owning (remote) DB2 subsystem and the referencing DB2 subsystem must share (be paired with) a common accelerator.
- Database connection profiles must exist for both DB2 subsystems.
- Your user ID needs the required authorizations to run the SYSPROC.ACCEL_REMOVE_REFERENCE_TABLES stored procedure on the referencing DB2 subsystem.

Procedure

1. Open the accelerator view of the accelerator that is paired with the referencing DB2 subsystem. Mind that although the owning (remote) DB2 subsystem is paired with the same accelerator, it will use a different accelerator view, which

will show a different list of tables. Make sure that you select the view which will display the tables in the referencing DB2 subsystem.

2. On the toolbar of the **Tables** section, click **Federation** > **Remove Reference Tables**.
3. In the DB2 Subsystems window, you see a list of the DB2 subsystems that are paired with the same accelerator. Select one or more DB2 subsystems containing reference tables.
4. Click **Next**.
5. In the DB2 Tables window, you see a list of the tables on the accelerator. Select the reference tables that you want to remove by clicking the appropriate check boxes. To select all tables that belong to a schema, select the check box in front of the schema name. If the list is long, first narrow the choice by using the **Name like** filter field. Type the names of schemas or tables in this field, either fully or partially, to display just schemas and tables bearing or starting with that name. The names of tables that have already been selected are greyed out.

   To the right of the **Name like** field, you also find controls (icons) to do the following:

   a. Select or deselect all tables
   b. Expand or collapse all schemas
6. Click **Finish**.

Results

The selected reference tables (referencing accelerator-only tables) are removed from the accelerator. Corresponding entries are removed from the catalog of the referencing DB2 subsystem.

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 program temporary fix (PTF) level.

Order in which to update components

Table 35. Order in which to update components

Migration from 3.1.0 PTF-*x* to 5.1.0 PTF-*x*	**Migration from 4.1.0 PTF-*x* to 5.1.0 PTF-*x* or Migration from 5.1.0 PTF-*x* to 5.1.0 PTF-*y***
1. IBM DB2 Analytics Accelerator for z/OS Version 5.1.0 FMIDs and program temporary fixes (PTFs) via SMP/E. See *Updating the z/OS base package* in the *Installation Guide*. 2. Prerequisite PTFs for DB2 10 for z/OS or DB2 11 for z/OS. See *Installing prerequisite PTFs for DB2 10 for z/OS or DB2 11 for z/OS* in the *Installation Guide*. 3. IBM DB2 Analytics Accelerator Studio. See "Updating IBM DB2 Analytics Accelerator Studio" 4. Access Server. See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168 5. Replication engine. See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168 6. IBM DB2 Analytics Accelerator software. See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168 7. Stored procedures. See *Updating the IBM DB2 Analytics Accelerator stored procedures* in the *Installation Guide*. 8. Netezza Firmware (FDT) (requires assistance from IBM support). See "Updating other Netezza software" on page 173. 9. Netezza Host Platform (HPF) (requires assistance from IBM support). See "Updating other Netezza software" on page 173. 10. Netezza Platform Software (NPS). See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168. 11. CDC Capture Agent. See See *Updating the CDC Capture Agent* in the *Installation Guide*.	1. IBM DB2 Analytics Accelerator for z/OS Version 5.1.0 FMIDs and program temporary fixes (PTFs) via SMP/E. See *Updating the z/OS base package* in the *Installation Guide*. 2. Prerequisite PTFs for DB2 10 for z/OS or DB2 11 for z/OS. See *Installing prerequisite PTFs for DB2 10 for z/OS or DB2 11 for z/OS* in the *Installation Guide*. 3. IBM DB2 Analytics Accelerator Studio. See "Updating IBM DB2 Analytics Accelerator Studio" 4. Access Server. See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168 5. Replication engine. See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168 6. IBM DB2 Analytics Accelerator software. See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168 7. Netezza Platform Software (NPS). See "Updating the IBM DB2 Analytics Accelerator software, the Netezza Platform Software (NPS), the Access Server, or the replication engine" on page 168. 8. Stored procedures. See *Updating the IBM DB2 Analytics Accelerator stored procedures* in the *Installation Guide*. 9. CDC Capture Agent. See See *Updating the CDC Capture Agent* in the *Installation Guide*.

Updating IBM DB2 Analytics Accelerator Studio

Follow the instructions here to download and install an upgrade of IBM DB2 Analytics Accelerator Studio.

About this task

You can no longer use IBM DB2 Analytics Accelerator Studio 3.2 for the administration of this product. You must upgrade to version 4.1.0 or a later version.

To download a refresh pack for IBM DB2 Analytics Accelerator Studio, a workstation with an internet connection is required.

If IBM Installation Manager is installed on a workstation without an internet connection, the program cannot automatically locate the repositories for updates of already installed products or additional plugins. Therefore, an administrator has to download the update repository to a central location that users without an internet connection have access to.

Workstation users without an internet connections can then let their local IBM Installation Manager installations point to the repository in the central location or copy the entire installation package from the central location to their workstations and go on from there.

You can use the newer versions of IBM DB2 Analytics Accelerator Studio with older versions of accelerators and stored procedures. 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.x or higher. However, the user interface might not show new functions until you update the stored procedures to the next higher or newest version (4.1.0 for version 4 functions or 5.1.0 for version 5 functions).

Procedure

- To download a refresh pack for yourself or for others from a computer with an internet connection:
  1. Click the following link:

     Select fixes: IBM DB2 Analytics Accelerator Studio (All releases, All platforms)
  2. Select the proper refresh pack and click the **Continue** button above the list.
  3. Sign in with your IBM ID or click **Create IBM id** and follow the instructions to create an IBM ID first.
  4. Select your preferred download method:
     - **Download using Download Director** (requires Java enabled browser)
     - **Download using bulk FTP**
     - **Download using your browser (HTTPS)**
  5. Leave the check box **Include prerequisites and co-requisite fixes (you can select the ones you need later)** selected.
  6. In the View and accept terms window, click **I agree** to agree to the terms and conditions.
  7. Click **Download now** and follow the instructions.
  8. Extract the content of the downloaded archive to a directory on your local workstation or to the central location (for access or download by other users).
- To update an IBM DB2 Analytics Accelerator Studio installation:
  1. Start IBM Installation Manager.
  2. Update IBM DB2 Analytics Accelerator Studio by using one of the following procedures:
     - The following approach works if you can still connect to a previously accessed IBM Installation Manager repository, and if that repository has been updated (contains the update):
       a. Click **Update**.
       b. Follow the instructions in the wizard. Deselect components that you do not want to update.
     - If the first method did not work, but you can access the repository file on your company's network or on your local workstation:
       a. From the main menu of IBM Installation Manager, select **File** > **Preferences** > **Repositories**.
       b. Click **Add Repository**.
       c. Click **Browse** and navigate to the directory that contains the extracted update package, either in the central location, or on your local workstation.
       d. Select the `repository.config` file and click **Open**.

e. Click **OK** twice to return the main window of IBM Installation Manager.
f. Click **Install**.
g. Follow the instructions in the installation wizard.

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 "Updating IBM DB2 Analytics Accelerator Studio" on page 166, 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 Platform Software (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 libraries and other code to be installed on the IBM PureData System for Analytics. Updates for the Access Server, the replication engine, and the Netezza Platform Software (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 z/OS UNIX file system (zFS).
- 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. Set the AQT_HOST_PACKAGE_DIRECTORY environment variable to the path in which the installation packages reside. By default, this is `<your_prefix>/usr/lpp/IBM/aqt/<version>/packages`, where

   <your_prefix>
   Is your customized path for SMP/E installations, starting from the root directory.

   <version>
   Is a directory name that identifies the product version, for example `/v5r1m0/`.

   Important: The directory that AQT_HOST_PACKAGE_DIRECTORY points to requires multiple gigabytes (GB) of disk space. For example, if you work with multiple accelerators in a rolling upgrade scenario, you must have multiple instances of the installation packages in the file system. So make sure that the directory can accommodate all the files. The *IBM DB2 Analytics Accelerator for z/OS: Program Directory* lists the exact disk space requirements.
3. If the stored procedures are still at a previous level (version 4 or earlier), make sure that you keep your setting of AQT_INSTALL_PREFIX in the AQTENV data set. It is still needed by the back-level stored procedures.

Results

Your local update directory now contains the accelerator update package *version*`.tar.z`.

Related information:

"Updating other Netezza software" on page 173

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 Platform Software (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/IBM/aqt/<version>/packages` directory on your z/OS data server, where `<version>` is a placeholder for a directory whose name indicates the product version. 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 Platform Software (NPS) are named differently.
- The value of the AQT_HOST_PACKAGE_DIRECTORY environment variable determines where IBM DB2 Analytics Accelerator Studio looks for software updates. If it is not set correctly, the packages cannot be located and thus cannot be transferred or installed.
- The user ID has read and write access to the directory that the AQT_HOST_PACKAGE_DIRECTORY environment variable points to and to all its subdirectories. You specify this variable in the AQTENV data set. The AQTENV data set must be referenced by the Workload Manager (WLM) environment that was set up for the IBM DB2 Analytics Accelerator for z/OS stored procedures. To avoid a setup with obsolete or wrong entries, use the sample AQTENV data set that comes with IBM DB2 Analytics Accelerator for z/OS Version 5.1.0.

  However, if the stored procedures are still at a previous level (version 4 or earlier), make sure that you copy the previous setting of AQT_INSTALL_PREFIX to your new AQTENV data set. It is still needed by the back-level stored procedures.
- 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 z/OS UNIX file system (zFS) 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 Updates window, you can see all software packages that are available in the zFS of your z/OS data server. Select the appropriate check boxes in the first column of the table to mark the packages that you want to transfer.

   Attention: Make sure that you select the proper packages, that is, packages belonging to the release level that you want to upgrade to. The list in the Transfer Updates window might be confusing, especially if it also contains older packages. To find the correct package numbers, see the closing information or the release notes for the latest program temporary fix (PTF). You find the closing information or the release notes for a PTF on the support home page, in the category *Plan and install documentation* (see link under **Related information** at the end of this topic). If 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.

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 **List Tasks** in the relevant Accelerator view of IBM DB2 Analytics Accelerator Studio and cancel all active console tasks from the List Tasks window.
- Consider the following points if you want to activate new versions of the Access Server, the replication engine, or both:
  - You cannot activate earlier versions of the Access Server or the replication engine. Just upgrades are possible here.
  - Stop incremental updates for each DB2 subsystem connected to the accelerator. To do this from IBM DB2 Analytics Accelerator Studio:
    1. In the Administration Explorer window, open the **Accelerators** folder.
    2. Select a relevant connection and open its Accelerator view.
    3. In the header section of the Accelerator view, look for **Replication:** and click **Stop**.
    4. Repeat the preceding steps for each DB2 subsystem connection to the same accelerator.

    Note: If you have connections to DB2 data sharing groups (DSGs) and prefer to run the SYSPROC.ACCEL_SET_REPLICATION stored procedure from a client command line: It is not necessary to stop incremental updates for each individual member. A DSG is treated like a single connection and stopping incremental updates for the primary member will also stop these for the other members.

About this task

- 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 Platform Software (NPS). You must activate updates of the Access Server, the replication engine, and the Netezza Platform Software (NPS) separately.
- The activation of a different accelerator software version causes a restart of the accelerator.

Activating a software update invokes the **SYSPROC.ACCEL_UPDATE_SOFTWARE** stored procedure on your data server. For information about the privileges that are required to run this procedure and further details, see the appropriate section in the *IBM DB2 Analytics Accelerator for z/OS: Stored Procedures Reference*. A link to this document is provided under **Related reference** at the end of this section.

Attention:

- The activation of a new version of IBM DB2 Analytics Accelerator for z/OS software components might delete the query history from the accelerator if the new version introduces changes to the query history function.
- The sequence in which you activate the different software components is important. The sequence depends on the source and on the target version. To find the proper sequence for your particular update, see Table 35 on page 166.
- When the accelerator is restarted with a new accelerator software version, accelerated queries cannot be processed on that accelerator before it reaches the *online* state. The only way to avoid a delay is to use a second accelerator in a workload balancing setup. That is, a second accelerator which is connected to the same DB2 subsystem, and which contains the same set of tables. Note that the tables must be loaded and enabled.

Procedure

1. In the Administration Explorer, select the **Accelerators** folder.
2. In the Object List Editor on the right, double-click the accelerator.
3. In the Accelerator view, from the **Refresh** drop-down list in the upper right, select **Automatic off**. Otherwise, you might see warnings during the activation of the new software saying that the accelerator cannot be contacted.
4. If necessary, expand the **About** section.
5. In the **About** section, click the **Apply other software version** link.
6. 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 Platform Software (NPS)** (software for the Netezza hosts)
   - **Access Server** (Access Server component for incremental updates)
   - **Replication engine** (replication engine or Apply Agent for incremental updates)

   Important: The order in which the wizard presents the components is not necessarily the order in which you must activate these. Use the same sequence for activating software components as for updating these. To find the proper sequence, see "Installing updates" on page 165.
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 after 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

- **Important:** The function that applies incremental updates in parallel was improved in product version 5.1.0. However, if this function was enabled in version 4.1.0, the old function code is still executed. To use the new code after an upgrade to product version 5.1.0 or an even newer version, stop and then restart incremental updates on the updated accelerators. For instructions, follow the appropriate link at the end of this topic.
- To avoid a cluttered Apply Software Version window, you can remove packages from the accelerator that you no longer need.

For instructions on how to complete either task, follow the appropriate link under **Related tasks**.

Related tasks:

"Removing obsolete software packages from an accelerator" on page 174

"Starting or stopping incremental updates" on page 138

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.

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 z/OS UNIX file system (zFS). 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. It specifies an absolute path (starting from the root directory). The AQT_HOST_PACKAGE_DIRECTORY environment is set in the `<HLQSP>.SAQTSAMP(AQTENV)` data set,

where <HLQSP> is the chosen high-level qualifier for stored-procedure libraries. The value of this environment variable determines where IBM DB2 Analytics Accelerator Studio looks for 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 5.1.0.

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 Updates window, you can see all Netezza Firmware (FDT) and Netezza Host Platform (HPF) files that are available in the zFS 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 Platform Software (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 z/OS UNIX file system (zFS) of your data server, manually delete no longer needed IBM DB2 Analytics Accelerator for z/OS or Netezza update packages from the zFS. 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 Platform Software (NPS), you must remove this parameter or set it to the value `false`. In addition, you must reload accelerator-shadow 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 Platform Software (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 Data Replication 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
 (6) - (Menu) Manage 'Call Home'
 (7) - (Menu) Manage Encryption

 ---------------------------------------------------------------------
 (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) - Enable the conversion mode for 24:00:00 h values to 23:59:59 h
 (9) N/A: - Disable the conversion mode for 24:00:00 h values to 23:59:59 h
 (10) - Set a system-wide resource limit
```

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-shadow 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 accelerator-shadow 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 DB2 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 accelerator-shadow tables that contain these columns.

Viewing the encryption status

You can check the encryption status of an accelerator from the corresponding accelerator view.

Procedure

1. Open the accelerator view of the accelerator that uses encryption.
2. In the header, click the **Encryption details** link. A window with the title Encryption Details for Accelerator <name> opens. It is devided into the following sections:

   Disk Encryption
   Shows whether disk encryption is used by your accelerator hardware.

   Network Encryption
   Shows two tables and a text box:

   Encrypted connections
   This table contains details about the encrypted connections (IPsec tunnels) to the selected accelerator. This includes the names of the connecting client LPARs, their IP addresses, the IP addresses of the active and the passive accelerator host, the names of the certificates that are used, and the date and time when the connections were established.

   Certificates
   This table shows details about the certificates that are used, such as the validity, the type (user or root certificate), and the distinguished name.

   Selected certificate in X.509 encoding
   Shows the content of a certificate that is selected in the table above, in (encrypted) X.509 Base64 encoding. Naturally, the displayed content does not include the private keys because these must remain secret.
3. Close the Encryption Details window to return to the accelerator view.

Chapter 7. Status information about the system and the incremental update function

When the system has been properly configured, a heartbeat connection is established between the accelerator and the connected DB2 subsystems. Every 30 seconds, the heartbeat connection provides the DB2 subsystems with status information about the accelerator. You can view most of this information by using the `-DIS ACCEL` DB2 commands. Detailed information about specific events cannot be viewed in this way, but is written to the z/OS system log (SYSLOG), in the form of DSNX881I messages. This includes messages that are related to the incremental update function.

To establish a heartbeat connection, an accelerator must have been paired successfully with a DB2 subsystem. In addition, the accelerator must have been started for that particular subsystem by the `-START ACCEL` command or by use of the corresponding function in IBM DB2 Analytics Accelerator Studio.

To receive incremental-update-related event messages in a SYSLOG, IBM InfoSphere Data Replication for z/OS must have been installed and properly configured for use with the incremental update function of IBM DB2 Analytics Accelerator.

Support model

IBM DB2 Analytics Accelerator for z/OS is a solution that consists of various hardware and software components. Each of these components might issue a DSNX881I message. If the message indicates a hardware or software problem, open a problem management record (PMR) with component ID (CompID) 5697AQT00.

Never open a PMR for Netezza or the IBM PureData System for Analytics. Always use the IBM DB2 Analytics Accelerator CompID, even if the problem is related to DB2 for z/OS or to the attached accelerator hardware.

Your PMR will be routed internally to the proper support request queue. Make sure that the PMR contains an IBM DB2 Analytics Accelerator trace file that was obtained by using the **Save Trace** function in IBM DB2 Analytics Accelerator Studio. Such a trace file does not only contain software trace messages, but also a complete set of diagnostic hardware information.

Consider using automatic DSNX881I notifications. This function, which became available with product version 4 PTF-4 can inform administrators of critical events and create PMRs automatically.

DSNX881I message structure

Each DSNX881I message is made up of the following parts, which occur in the order indicated:

```
DSNX881I -<SSID> <MESSAGE-ID> <SEVERITY>
<ACCELERATOR_MESSAGE_COUNTER> (<ACCELERATOR-TIMESTAMP>)
ACCELERATOR-NAME(ACCELERATOR-IP) <MESSAGE-TEXT>
```

The placeholders have the following meaning:

© Copyright IBM Corp. 2009, 2017

SSID
Is the DB2 subsystem ID (SSID)

MESSAGE-ID
A numeric ID for the specific error message. This ID can be used for system monitoring.

Messages with IDs from 1 to 19 and from 21to 2999
Accelerator heartbeat (system state) messages.

Messsages with ID 20
These are incremental-update-related messages generated by IBM InfoSphere Data Replication for z/OS (CDC). They can be informational messages, warnings, or errors. Do not configure automatic notifications and PMRs based on these messages, as the messages do not take the overall system state into account. In most cases, these messages are issued alongside messages in the range from ID3000 to ID3999. Refer to the latter type of messages because these contain proper instructions and do take the overall system state into account.

Message with IDs from 3000 to 3999
These are messages related to the incremental update function. In contrast to the ID20 messages, these messages are more specific or might ask for a concrete intervention (manual or automated) on your part. They can be used to trigger automatic notifications and PMRs.

SERVERITY

I Information message

W Warning message

E E Error message

ACCELERATOR_MESSAGE_COUNTER
An internal counter that increases with every additional error on the accelerator. If the text after the DSNX881I qualifier is longer than 255 characters, another DSNX881I message is issued. All messages belonging together will have the same <ACCELERATOR_MESSAGE_COUNTER> value. The <MESSAGE-TEXT> block of the each subsequent message contains a sequel to the information in the previous message.

ACCELERATOR-TIMESTAMP
The time when the error occurred on the accelerator. The internal clock of the accelerator is synchronized with the first DB2 subsystem that was connected to the accelerator.

ACCELERATOR-NAME
The name of the accelerator where the error occurred. This information is only sent for accelerators that are running a version of the accelerator server that is higher than V2-PTF6, V3-PTF3, or V4-PTF1. See the prerequisite website for information on how to determine the PTF number of your version.

ACCELERATOR-IP
The IP address of the accelerator on which the error occurred. This information is only sent for accelerators that are running a version of the accelerator server that is higher than V2-PTF6, V3-PTF3, or V4-PTF1. See the prerequisite website for information on how to determine the PTF number of your version. The field can be empty if no IP address can be determined. However, the parentheses will appear.

MESSAGE
A textual description of the error.

The length of a DSNX881I message does not exceed 255 characters. If more characters are needed, additional DSNX881I messages are written to the SYSLOG. If an LPAR contains multiple DB2 subsystems that are connected to the same physical accelerator, error messages are issued for every subsystem. That is, you see the same messages multiple times in the log, each time with a different subsystem ID (SSID).

If an accelerator is paired with a data sharing group (DSG), all members of the group can write messages to group's system logs (SYSLOGs), provided that the `-START ACCEL` command has been issued for all members. In this case, make sure that applications are in place that monitor the SYSLOGs. If all members of the DSG are located in the same logical partition (LPAR), there is only one SYSLOG to monitor (see Figure 16). However, if the members are located in different LPARs, you need to monitor the SYSLOGs of all LPARs involved (see Figure 17 on page 182.

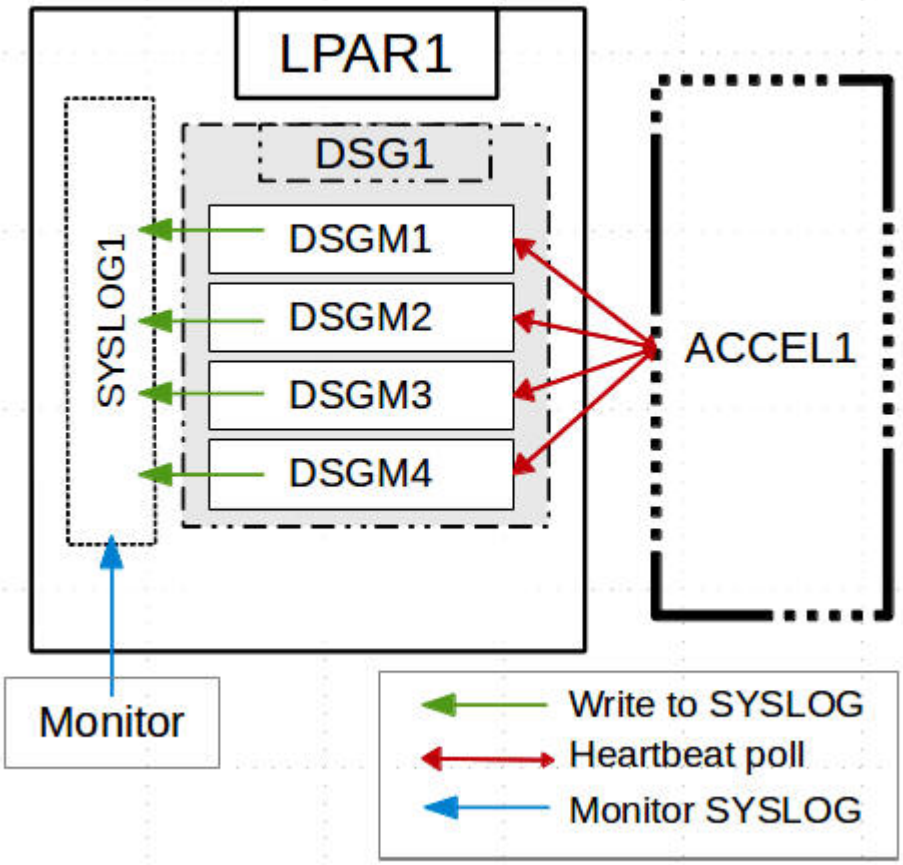


Figure 16. All members in the same LPAR

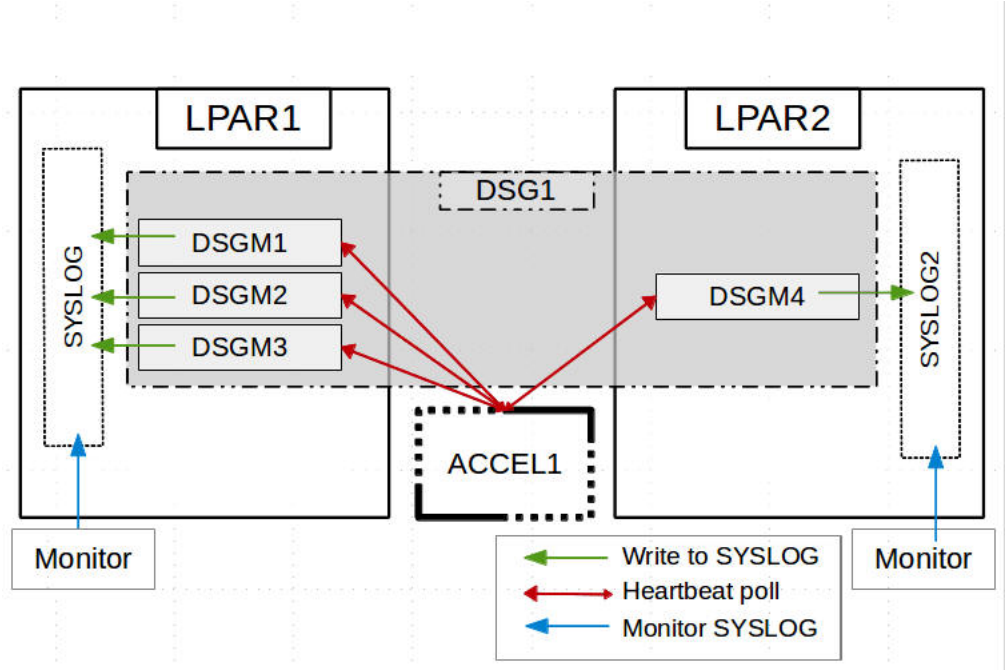


Figure 17. Members in two different LPARs

Note: It might look as if only one member writes messages to the SYSLOG, but this is actually a synchronization issue. If one member is always the first to issue a heartbeat request, then this member will receive all the messages and write these to the SYSLOGs. After that, the messages are deleted from the accelerator queue. The other members that send their heartbeat requests later will not receive these messages because the queue is empty. You might also see that only a few members write messages to the SYSLOG. This just means that the first member to send a heartbeat request is (always) found among this subset of members. The underlying mechanism is the same.

If an error occurs while an accelerator is in the *Stopped* state,that is, the `-STOP ACCEL` command has been issued before, an error message is stored on the accelerator. As soon as the accelerator becomes available again in DB2, the stored error messages are sent to the DB2 subsystem, provided that `-START ACCEL` has been issued for the subsystem, or, in case of a data sharing group, for at least one member of the group. It might happen that a DSNX881I message reports a past problem that has already been fixed.

For a list of message IDs, severity indicators, and message texts, see *Structure of DSNX881I Messages* (link under **Related information**).

Related information:

Prerequisites and Maintenance for IBM DB2 Analytics Accelerator for z/OS Version 5.1

Structure of DSNX881I Messages

Chapter 8. Troubleshooting

In the following sections, you find descriptions of known IBM DB2 Analytics Accelerator for z/OS problems. The author and the development team have tried to provide a solution wherever possible. However, a solution might be unavailable because the cause of a problem cannot be clearly identified. This is mostly the case if multiple causes can lead to the same symptom. It can also be that a solution has not yet been found. In such cases, contact IBM support.

Important: During a customer session with IBM support, system parameters might be changed. For the parameter changes to take effect, a restart of an accelerator is often required. In such situations, the accelerator will be unavailable for operational tasks until it has come back online.

Redundant CDC metadata

When you start the CDC Capture Agent on z/OS, a warning is written to the CHCRPRT report data set.

Symptoms

The following message is added to the report:

```
The following tables are defined for replication, but are not used in any
subscriptions. These can cause the product to perform inefficiently and should be
removed from the list of replication tables if not required.
```

Causes

Incremental updates were disabled for one or more accelerator-shadow tables. However, the CDC metadata describing these tables still exists.

Resolving the problem

You can ignore this message because the impact on the performance is negligible. If you want, you can remove the redundant metadata by using a function on the IBM DB2 Analytics Accelerator Console.

Mind that the removal function removes only redundant CDC metadata, but not the metadata of tables that are still actively replicated. For more information, see *Removing unused table metadata* in the *IBM DB2 Analytics Accelerator for z/OS: Installation Guide*.

Firewall blocks operations because crucial ports are closed

If product functions are not executed, check your firewall settings. IBM DB2 Analytics Accelerator requires certain ports to be open. See the table in this topic.

Symptoms

- You cannot accelerate queries.
- You cannot run functions from the IBM DB2 Analytics Accelerator Console.
- Incremental updates do not work or IBM InfoSphere Data Replication for z/OS cannot be configured.
- You cannot open a secure-shell (ssh) connection to an accelerator.
- Services are unavailable or switches (routers) cannot be reached because ICPM (Internet Control Message Protocol) traffic is blocked.

© Copyright IBM Corp. 2009, 2017

Causes

Required networking ports are closed.

Resolving the problem

Table 36. Required networking ports for IBM DB2 Analytics Accelerator

Function or protocol	Required port	Direction
Query execution (DRDA protocol)	1400	Both ways
IBM DB2 Analytics Accelerator Console	1600	Both ways
CDC data transfer	11301-11400	Outbound (from z/OS to accelerator)
CDC configuration	5999	Inbound (generally from accelerator to z/OS; depends on the configuration of the CDC Capture Agent)
DB2 DDF port (for continuous incremental updates)	446 (default)	Inbound (from accelerator to z/OS)
Secure shell connection (ssh)	22	Both ways
ICMP traffic	N/A	Both ways

IBM DB2 Analytics Accelerator Studio does not start

If IBM DB2 Analytics Accelerator Studio does not start, the reason might be that the name of the starting user contains a number sign (#).

Symptoms

IBM DB2 Analytics Accelerator Studio does not start.

Causes

At installation time, IBM DB2 Analytics Accelerator Studio creates a workspace directory in

```
@user.home\IBM\DB2 Analytics Accelerator Studio 2.1\workspace
```

where `@user.home` is the user's home directory, which is usually the same as the user name.

This means, if the user name contains a number sign (#), the workspace directory path contains a number sign, too. IBM DB2 Analytics Accelerator Studio cannot handle workspace directory paths that include number signs.

Resolving the problem

1. If the user name contains a number sign (#), open the file `datastudio.ini` in a text editor. This file is located in the installation directory of IBM DB2 Analytics Accelerator Studio.
2. Remove the following entries:

   ```
   -data
   @user.home\IBM\DB2 Analytics Accelerator Studio 2.1\workspace
   ```
3. Save and close the file.

4. The next time you start IBM DB2 Analytics Accelerator Studio, you are asked for the workspace directory path. This allows you to enter an alternative path that does not contain a number sign.

IBM DB2 Analytics Accelerator Studio on Linux does not start

IBM DB2 Analytics Accelerator Studio does not start from certain versions of Linux, particularly SUSE Linux Enterprise Server 10 with service pack 4.

Symptoms

An error message similar to the following is displayed:

```
/opt/tools/isaostudio//./jre/bin/javaw: symbol lookup error
/usr/lib/xulrunner-1.9.2.17/libxul.so: undefined symbol
gdk_screen_get_resolution
```

Causes

An updated version of a required library (`XULRunner`) is not compatible with IBM DB2 Analytics Accelerator Studio.

Resolving the problem

Use an earlier version of the `XULRunner` library. The problem also occurs in connection with IBM Rational® Performance Tester. You can apply the suggested solution analogously. See:

http://www.ibm.com/support/docview.wss?uid=swg21393436

Message AQT10200E and SQL code -206 when running stored procedures after migration

Stored procedure calls end with message `AQT10200E` 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:

```
AQT10200E - 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 `-STOP 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 an accelerator-shadow table.

Symptoms

The progress indicator never reaches the end or runs in an endless loop. You never receive a message that reports the successful completion or failure of the task.

Causes

A DB2 timeout setting (IDLE THREAD TIMEOUT parameter) prematurely ends the process (DDF thread) before the task is completed. The default value of this parameter is 2 minutes.

Resolving the problem

Increase the DB2 timeout value. For instructions, follow the link at the end of this topic.

Related information

SQL code -430 from IBM DB2 Analytics Accelerator for z/OS stored procedures

A stored procedure of IBM DB2 Analytics Accelerator for z/OS ends abnormally, and you receive an error message with SQL code `-430`.

Symptoms

IBM DB2 Analytics Accelerator for z/OS stored procedures end abnormally. In IBM DB2 Analytics Accelerator Studio, the Administration Explorer returns a message window like this one:

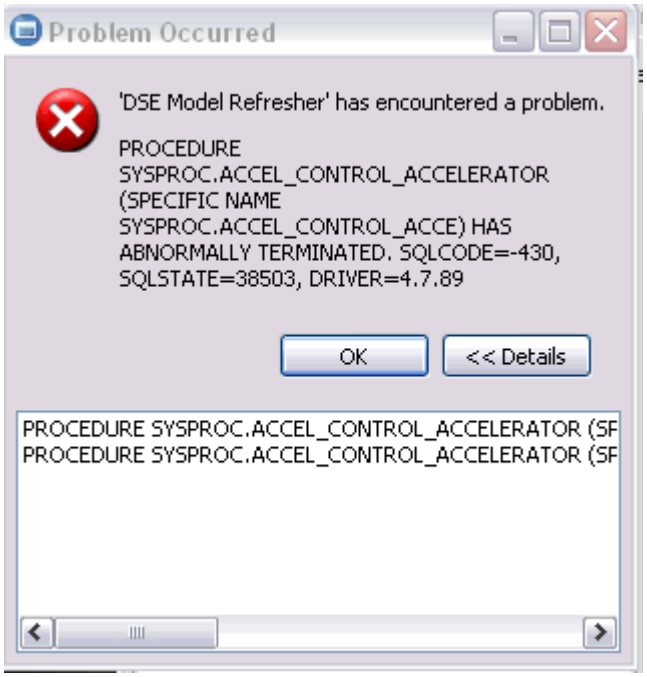


Figure 18. SQL code -430 message window

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:
   https://www.ibm.com/support/knowledgecenter/SS62YD_3.1.1/com.ibm.datatools.qrytune.configothers.doc/topics/enabledb2zfromclient_ds.html
2. To grant the EXECUTE privilege to other users, proceed as follows:
   https://www.ibm.com/support/knowledgecenter/SS7L9Q_4.1.0/com.ibm.datatools.qrytune.workloadtune.doc/topics/verify.html

DRDA connection does not work

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

Symptoms

DB2 commands or IBM DB2 Analytics Accelerator for z/OS stored procedures cannot establish a TCP/IP connection with the accelerator. Running the SYSPROC.ACCEL_TEST_CONNECTION stored procedure reveals this issue.

Resolving the problem

Make sure that the distributed data facility (DDF) of DB2 for z/OS uses the same TCP/IP stack as the ping program.

Different results for floating-point aggregates

Queries that involve floating-point aggregates produce different results when processed by IBM DB2 Analytics Accelerator for z/OS rather than DB2 for z/OS.

Symptoms

The very same query is processed twice, once by IBM DB2 Analytics Accelerator for z/OS and once by DB2 for z/OS. The results are different.

Causes

The result differences are differences in precision or error tolerances, which are caused by different methods of representation for floating-point numbers.

Resolving the problem

The problem cannot be solved. The differences must be accepted.

Related concepts

"Differences in floating-point error tolerances" on page 56

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 accelerator-shadow tables that have been enabled for incremental updates takes excessively long.

Symptoms

The removal process seems to run endlessly.

Resolving the problem

Stop the incremental update process for the accelerator before removing the 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 accelerator-shadow tables.
4. Restart incremental updates for the accelerator.

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

IBM InfoSphere Data Replication 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 Data Replication for z/OS is not notified of the shutdown. IBM InfoSphere Data Replication for z/OS still “expects” to find the instance of DB2 that was previously running. When DB2 for z/OS is restarted, IBM InfoSphere Data Replication 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 Data Replication 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 z/OS UNIX file system (zFS) for z/OS. This file system, which contains the directory for the storage of temporary files (`/tmp`), is full. No more temporary files can be created for the execution of stored procedures.

Resolving the problem

Delete no longer needed files from the zFS. For example, you can safely delete stored-procedure trace-files. Note that you need superuser privileges in IBM UNIX System Services to do that. Follow these steps to delete the stored-procedure trace-files:

1. Log in to the IBM UNIX System Services shell of the logical partition (LPAR) from where you started the IBM DB2 Analytics Accelerator for z/OS stored procedure that returned the error. To log in, you can open a secure shell (ssh) window and use the **su** command to log in with superuser privileges. You can also use the IBM UNIX System Services shell to log in directly with an ID that has the required superuser privileges.
2. Change to the `/tmp` directory:

   ```
   cd /tmp
   ```

3. Delete the stored-procedure trace files:

   ```
   rm aqt-stored-procedure-trace*
   ```

Query history is not displayed - error message returned

The query history is not displayed in the Accelerator view and a corresponding error message is returned.

Symptoms

You do not see valid values in the columns of the **Query Monitoring** section of the Accelerator view and a message similar to the following is displayed:

```
[2013-05-25 05:14:09,790] [ERROR] [HistoryCol] Query History collection
    impossible com.ibm.db2.jcc.am.SqlException: UNSUCCESSFUL EXECUTION CAUSED
    BY AN UNAVAILABLE RESOURCE. REASON 00C900D1, TYPE OF RESOURCE 00000907,
    AND RESOURCE NAME. SQLCODE=-904, SQLSTATE=57011, DRIVER=4.15.79
```

Causes

The value of the DB2 for z/OS parameter LOBVALA might be too low.

Resolving the problem

Increase the value of the LOBVALA parameter. The default value is 10240. Set it to 20480. Increase the value further if this does not solve the problem.

DB2 for z/OS returns `SQLCODE -30040`

A query cannot be executed and `SQLCODE -30040` is returned by DB2 for z/OS.

Symptoms

DB2 for z/OSreports `SQLCODE -30040`, after IBM DB2 Analytics Accelerator has reported a `DRDAResourceLimitError` (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 Platform Software (NPS) 7.0.4.

Resolving the problem

Rerun the query.

Restoring of archived partition fails with `'Duplicate key'` error

You cannot restore a partition that was archived by the High Performance Storage Saver (HPSS). The **Restore** function in IBM DB2 Analytics Accelerator Studio, or the SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLE stored procedure fails with a message that hints at a duplicate key.

Symptoms

While restoring an archived partition, you receive error message AQT0301I from a stored procedure with the following description of a utility error in the message details:

```
DSNU340I, ERROR LOADING INDEX, DUPLICATE KEY
```

Causes

The problem is known to occur after the following series of events:

1. You have created a range-partitioned table with a unique index or primary key and the unique constraint consists of a set of columns that is different from the columns used in the partitioning key.
2. You have loaded the table and refreshed the table data on the accelerator.
3. You have archived one or more partitions of the table.
4. One or more rows have been inserted in an active partition of the table. Some of the values in the unique constraint columns of these rows match the unique constraint values of rows in the archived partition. The new rows could be inserted successfully because only the active data is checked for duplicate entries.
5. You restore the archived partition and thus end up with duplicate values for the unique constraint.

Resolving the problem

You can only solve this problem by eliminating its cause. That is, make sure that unique constraint values are not used in rows to be inserted in your active tables if these values were already used in a archived partition.

Table definition fails with error message `AQT1005I`

You cannot define a table on an accelerator. Message `AQT1005I` is displayed.

Symptoms

The full error message is similar to the following:

```
AQT1005I - An internal error occurred on the <acclerator-name> accelerator:
An assertion 'rowLength <= internal::CatalogConstants::MAXIMUM_NETEZZA_ROW_LENGTH'
failed. Additional info: Maximum row length 122464 of table
<schema-name>.<table-name> exceeds limit 65535
```

Causes

The table contains column definitions that would exceed the maximum row length for accelerator-shadow tables (total length of all column fields in a row). The maximum row length is 65535 bytes.

Example:

```
CREATE TABLE <NAME> (
    A VARCHAR(32704) FOR MIXED DATA NOT NULL
);
```

On the accelerator, the corresponding accelerator-shadow table is created as follows:

```
CREATE TABLE <NAME> (
    A NVARCHAR(16000) NOT NULL
);
```

The definition of the CHAR FOR MIXED DATA column A in the original DB2 table already exceeds the maximum row length of an equivalent NVARCHAR column on the accelerator. However, this can still be fixed by reducing the column length to the allowed maximum of 16000 characters.

Each character in an NVARCHAR column has a length of four bytes. After the reduction, the total row length is 16000 * 4 plus a few additional bytes, which would equal some 64000 bytes. Hence the length of column A lies within the limit. Since the table has only this one column, the total row length of the entire table

does not exceed the limit either. However, during a load of the accelerator-shadow table, original DB2 string values exceeding the limit will be truncated after the first 16000 characters.

In contrast, the following DB2 table cannot be added to the accelerator because the maximum row length is exhausted by the definition of column A. With the definition of column B, the maximum row length would be exceeded by far. A reduction of the column length is not possible because the maximum row capacity has already been used up by column A.

```
CREATE TABLE <NAME> (
    A VARCHAR(16000) FOR MIXED DATA NOT NULL,
    B VARCHAR(16000) FOR MIXED DATA NOT NULL
);
```

Resolving the problem

To define an accelerator-shadow table that exceeds the maximum row length, you must change its source table in DB2 for z/OS, that is, define fewer columns or columns with a shorter field length.

SQL error as you try to enable tables for incremental updates

The incremental update function is enabled, but you cannot enable tables to be included in the process.

Symptoms

You receive an error message that is similar to this one:

```
DB2 SQL Error: SQLCODE=-551, SQLSTATE=42501,
SQLERRMC=CDCIDAA;UPDATE;CDCUSER.DMMD_SIGNALS.SIGNAL_STATE
```

Causes

The user (ID) who starts the CDC started task for the CDC Capture Agent does not have sufficient authorizations on the CDC metadata tables.

Resolving the problem

Make sure that the user who starts the CDC started task has the following authorizations on the *<CHCMetaID>*.`DMMD_SIGNALS` table:

- SELECT
- INSERT
- UPDATE
- DELETE

where *<CHCMetaID>* stands for the ID of the owner or the schema name of the CDC metadata tables (security identifier). You provided this ID when you ran the CDCMDMUT customization job.

Appendix A. Mappings of product SQL codes to DB2 for z/OS SQL codes

When an error occurs on one of your accelerators or in the underlying Netezza database, the product generates an SQL code, which is then mapped to the closest DB2 SQL code. The codes and a message token are then returned to DB2. The message token usually contains a part of the information of the original message, but not all of it. The following table lists all mappings.

© Copyright IBM Corp. 2009, 2017

Table 37. Mappings of product SQL codes to DB2 for z/OS SQL codes

Returned to DB2			Error source	Remark	Introduced in product version or program temporary fix (PTF)
SQLSTATE	**SQLCODE**	**Message token**			
"02000"	100	Empty	• Queries: N/A (end of file reached while reading from pipe) • Data manipulation language (DML): SQLCODE 0 and no rows affected	Created by NZRDBManager • fetch() until end of result set • No rows affected by INSERT, UPDATE, or DELETE in executeStatement() expression	V2
"00000"	0	Empty		Various places in NZRDBManager and SQLApplicationManager	V2
NZSTATE	NZCODE	A maximum of 70 bytes of the Netezza message	Netezza SQLSTATE "00xxx" or "01xxx"; Netezza SQL code >= 0		V2
"58004"	-901	A maximum of 63 bytes of the information in the Netezza SQLSTATE:	Netezza SQLSTATE "00xxx" or "01xxx"; Netezza SQL code < 0		V2
"57011"	-904	A maximum of 70 bytes of the Netezza message	Netezza SQLSTATE "08xxx"	**Note:** Also mapped to SQLCODE -904 and SQLSTATE "57011" including error tokens like "HYxxx".	
"57014"	-952	Empty	TaskCanceledException detected by accelerator	This error message is not returned in case the SQL statement was cancelled in DB2 (by a CAN -THREAD command or some other means). Such a DB2-side cancellation results in the closure of the DRDA connection, which is interpreted as a success by IBM DB2 Analytics Accelerator. (The statement execution on the accelerator was successful after all.)	V2
"26501"	-514	SQL_CURSOR	*Cursor not prepared* condition detected by accelerator	OPNQRY is chained, so if PRPSQLSTT fails to open, it will get this error. Note that PRPSQLSTT will return a SQLCA with the actual failure already.	V2

Table 37. Mappings of product SQL codes to DB2 for z/OS SQL codes (continued)

Returned to DB2					
SQLSTATE	**SQLCODE**	**Message token**	**Error source**	**Remark**	**Introduced in product version or program temporary fix (PTF)**
"07001"	-313	Empty	A condition where the number of host variables is not equal to the number of parameter markers was detected by the accelerator	can only occur on PRPSQLSTT	V2
"56084"	-352	String value of the *position* argument	*Parameter marker SQL type mismatch* condition detected by accelerator	V2	
"57012"	-30040	Empty	Too many SQL statements already running	Too many SQL statements are already running. The threshold is defined by the configuration parameter MAXIMUM_NUMBER_ CONCURRENT_ RUNNING_QUERIES and the QueryGovernorDaemon may adjust the value (only increasing it) based on the number of data slices of the Netezza backend. For such events we send DRDA message RSCLMTRM. It has no SQLCA associated, but DB2 still displays an error. The SQLCODE and SQLSTATE described here are the ones displayed by DB2 and we do store those in the SQL statement history. The SQL statement execution is rejected before any DRDA flow processing happens (after authentication).	V2

Table 37. Mappings of product SQL codes to DB2 for z/OS SQL codes (continued)

Returned to DB2					
SQLSTATE	**SQLCODE**	**Message token**	**Error source**	**Remark**	**Introduced in product version or program temporary fix (PTF)**
"429CB"	-4747	One of the following: • Unsupported data type *type-name* • Invalid CCSID *CCSID* • Unsupported FOR BIT DATA column in *table-name* table (of type Unicode or ASCII)	One of the following: • `UnsupportedColumnDataTypeException` • `InconsistentCCSIDInAcceleratorException` • `UnsupportedForBitDataColumnInNonEBCDICTableException`	Only applicable to accelerator-only tables	V4 PTF5; not applicable before
"56045"	-919	Empty	`TransactionMustRollbackException`	Only applicable to transactions with at least 1 DML/DDL statement	V4 PTF5; not applicable before

Table 37. Mappings of product SQL codes to DB2 for z/OS SQL codes (continued)

Returned to DB2			Error source	Remark	Introduced in product version or program temporary fix (PTF)
SQLSTATE	**SQLCODE**	**Message token**			
"42601"	-104	Found token in error; list of expected tokens	*Syntax error* condition detected by accelerator	Netezza error message text retrieved from SQLException and parsed to extract values for message tokens	V4 PTF5; was -904 or -901 before
"22007"	-181	Empty	Invalid *date, time, or datetime value*		
"42702"	-203	Column name	*Column is ambiguous* condition detected by accelerator		
"42704"	-204	Table name	*Table not found* condition detected by accelerator. In connection with *federated access* support, this error might mean that a reference table or the original table that a reference table is pointing to cannot be found on the accelerator.		
"42703"	-206	Column name	*Column not found* condition detected by accelerator		
"22018"	-420	Internal Netezza function name, for example **atoi**	*Conversion of string expression failed* condition detected by accelerator		
"42808"	-151	*Distribution key column* (string constant)	*Distribution key update* detected by accelerator		
"42721"	-725	1. Special register name 2. Special register value	*Invalid special register setting* condition detected by accelerator		

Table 37. Mappings of product SQL codes to DB2 for z/OS SQL codes (continued)

Returned to DB2					
SQLSTATE	**SQLCODE**	**Message token**	**Error source**	**Remark**	**Introduced in product version or program temporary fix (PTF)**
"22003"	-802	1. One of the following: • "OVERFLOW" • "OUT OF RANGE" • "EXPONENT OVERFLOW" 2. "ACCELERATOR" 3. "NUMERIC DATA" (all string constants)	*Numeric overflow* condition detected by accelerator	Netezza error message text retrieved from SQLException and parsed to extract values for message tokens	V4 PTF5; was -904 or -901 before
"22012"	-802	1. "ZERO DIVIDE" 2. "ACCELERATOR" 3. "NUMERIC DATA" (all string constants)	*Divide by zero* condition detected by accelerator		
"40001"	-911	1. 0x00C9008E 2. 00001080	*Concurrent update of same row* condition detected by accelerator		
"54010"	-670	Column name	MaximumRowLengthLimitExceededException or *row length limit exceeded* condition detected by accelerator	Applicable when creating accelerator-only tables, or when a query or DML statement requires temporary tables in Netezza database.	V4 PTF5; was generic -904 or -901 before
"57011"	-904	1. 0x00001080 2. 0x00E7000E 3. Resource name (60 bytes): *Max 60 Bytes of IDAA Message*	Internal product errors	Netezza error message text retrieved from SQL exception and parsed to extract values for message tokens.	V2

Table 37. Mappings of product SQL codes to DB2 for z/OS SQL codes (continued)

Returned to DB2					
SQLSTATE	**SQLCODE**	**Message token**	**Error source**	**Remark**	**Introduced in product version or program temporary fix (PTF)**
"57011"	-904	1. 0x00001080 2. 0x00E7000E 3. Resource name (60 bytes): *Netezza SQLSTATE: Max 53 Bytes of Netezza Message*	Netezza SQLSTATE HYxxx	Catch-all for all Netezza errors that are not mapped to more specific SQL codes or SQL states **Note:** All such errors, if originating from Netezza, should be treated as defects so that a better error mapping can be found and implemented in IDAA. The following exceptions apply: • ERROR: SPU swap partition : Disk temporary work space is full – This error is raised by Netezza if an (intermediate) query result becomes too big and not enough temp disk storage is available. DB2 raises -904 in that situation as well.	V2
"58004"	-901	*Netezza SQLSTATE:Max 63 Bytes of Netezza Message*	All other Netezza errors	Catch-all for all Netezza errors that are not mapped to more specific SQL codes or SQSQLCODE/SQLSTATE **Note:** All such errors, if originating from Netezza, should be treated as defect so that a better error mapping can be found and implemented in IDAA.	V2

Appendix B. Impact of special register, ZPARM, or bind option settings on accelerator-only tables

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

Legend

The statement examples in Table 38 on page 206 use abbreviations that indicate the type of a table:

AOT Accelerator-only table

ACC Regular accelerator table

DB2 DB2 table (not on accelerator)

Remember that the ZPARM setting `QUERY_ACCEL_OPTIONS=2` enables `INSERT ... SELECT FROM ...` statements. With accelerator-only tables, the effect is often different from the effect it has in connection with regular accelerator tables. In many cases, the setting leads to an error. When applied successfully, it mostly leads to an acceleration of the `INSERT` portion of an `INSERT ... SELECT FROM ...` statement.

© Copyright IBM Corp. 2009, 2017

Table 38. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2

Statement	Special register SET CURRENT QUERY ACCELERATION= ZPARM QUERY_ACCELERATION= (ZPARM) Bind option QUERYACCELERATION=	QUERY_ACCEL_OPTIONS=2	Statement execution
SELECT * FROM T1_AOT	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator
SELECT * FROM T1_AOT, T5_ACC	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator
SELECT * FROM T1_AOT, T3_DB2	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 12
SELECT * FROM T5_ACC, T3_DB2	NONE (1)	No	DB2
	ALL (5)	No	Error, reason code 12
	ELIGIBLE (4)	No	DB2
	ENABLE (2)	No	Accelerator
INSERT INTO T1_AOT VALUES (...)	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator
INSERT INTO T1_AOT SELECT FROM T2_AOT	NONE (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	INSERT portion accelerated
		Yes	INSERT portion accelerated

Table 38. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	Special register SET CURRENT QUERY ACCELERATION= ZPARM QUERY_ACCELERATION= (ZPARM) Bind option QUERYACCELERATION=	QUERY_ACCEL_OPTIONS=2	Statement execution
INSERT INTO T1_AOT SELECT FROM T5_ACC	NONE (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	INSERT portion accelerated
		Yes	INSERT portion accelerated
INSERT INTO T1_AOT SELECT FROM T3_DB2	NONE (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 12
		Yes	Error, reason code 12
INSERT INTO T1_AOT SELECT FROM T2_AOT, T3_DB2, T5_ACC	NONE	No	Error, reason code 2
		Yes	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 2
		Yes	Error, reason code 12
• INSERT INTO T5_ACC SELECT FROM T1_AOT • INSERT INTO T3_DB2 SELECT FROM T1_AOT	NONE (1)	No	Error, reason code 2
		Yes	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	SELECT portion accelerated
		Yes	SELECT portion accelerated

Table 38. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	Special register SET CURRENT QUERY ACCELERATION= ZPARM QUERY_ACCELERATION= (ZPARM) Bind option QUERYACCELERATION=	QUERY_ACCEL_OPTIONS=2	Statement execution
• INSERT INTO T5_ACC SELECT FROM T5_ACC • INSERT INTO T3_DB2 SELECT FROM T5_ACC	NONE (1)	No	DB2
		Yes	DB2
	ALL (5)	No	Error, reason code 17
		Yes	SELECT portion accelerated
	ELIGIBLE (4)	No	DB2
		Yes	SELECT portion accelerated
	ENABLE (2)	No	DB2
		Yes	Entirely in DB2 or SELECT portion accelerated (based on estimated response-time reduction)
• INSERT INTO T5_ACC SELECT FROM T3_DB2 • INSERT INTO T3_DB2 SELECT FROM T3_DB2	ALL (5)	No	Error, reason code 17
		Yes	Error, reason code 12
	NONE (1) ENABLE (2) ELIGIBLE (4)	No	DB2
		Yes	DB2
• UPDATE T1_AOT (SELECT FROM T2_AOT) • UPDATE T1_AOT (SELECT FROM T5_ACC) • DELETE T1_AOT (SELECT FROM T2_AOT) • DELETE T1_AOT (SELECT FROM T5_ACC)	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Accelerator
• UPDATE T1_AOT (SELECT FROM T3_DB2) • DELETE T1_AOT (SELECT FROM T3_DB2)	NONE (1)	No	Error, reason code 2
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 12

Table 38. Statement execution depending on the settings of CURRENT QUERY ACCELERATION, QUERY_ACCELERATION, or QUERYACCELERATION and QUERY_ACCEL_OPTIONS=2 (continued)

Statement	Special register SET CURRENT QUERY ACCELERATION= ZPARM QUERY_ACCELERATION= (ZPARM) Bind option QUERYACCELERATION=	QUERY_ACCEL_OPTIONS=2	Statement execution
• UPDATE T5_ACC (SELECT FROM T1_AOT) • UPDATE T3_DB2 (SELECT FROM T1_AOT) • DELETE T5_ACC (SELECT FROM T1_AOT) • DELETE T3_DB2 (SELECT FROM T1_AOT)	NONE (1)	No	Error, reason code 22
	ENABLE (2) ELIGIBLE (4) ALL (5)	No	Error, reason code 22

Glossary

This glossary includes terms and definitions for IBM DB2 Analytics Accelerator for z/OS.

The following cross-references are used in this glossary:

- *See* refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- *See also* refers you to a related or contrasting term.

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

A

accelerator-archive table
A table on an accelerator that contains data archived by the High-Performance Storage Saver (HPSS). The data in an accelerator-archive table comes from a DB2 table. The original DB2 data is deleted during the archiving process, so that storage space is freed in DB2 for z/OS. The HPSS is typically used for historical DB2 data that is no longer actively queried and maintained. It is none the less possible to include the data in accelerator-archive tables in accelerated queries. To archive a DB2 table or partitions of this table with the HPSS, the table must exist as an accelerator-shadow table. However, the tables does not have to be loaded.

accelerator-shadow table
A table on an accelerator whose definition is derived from a DB2 for z/OS table and that can be used for accelerated queries after loading (copying data from the original DB2 table to the accelerator-shadow table).

access plan graph
A visual representation of a query that shows the database objects that are accessed by the query and the order in which this is done.

APF See authorized program facility.

authorized program facility (APF)
In a z/OS environment, a facility that permits the identification of programs that are authorized to use restricted functions.

C

cluster
- Number of configured accelerators.
- Table rows that are grouped together because they have the same value in a particular column.

colocated join
A join method where columns of an accelerator-shadow table must be on the same processing node. Leads to shorter query response times.

coordinator node
The active Netezza host.

© Copyright IBM Corp. 2009, 2017

D

data slice
A portion of an accelerator-shadow table that is processed by a worker node.

DDF See distributed data facility (DDF).

distributed data facility (DDF)
A set of DB2 for z/OS components by which DB2 for z/OS communicates with another relational database management system.

Dynamic Statement Cache
A pool in which DB2 saves prepared SQL statements that can be shared among different threads, plans, and packages. By sharing these statements, applications can avoid unnecessary preparation processes and thus improve performance.

F

fact table
A database table, commonly located at the center of a star schema surrounded by dimension tables, that contains measures of a business process, and foreign keys from the dimension tables.

failback
An automatic operation that returns the processing of a query to DB2 for z/OS if an error occurs while the query is processed by an accelerator.

friendly arithmetic
When an exception, such as a buffer overflow, a division by zero, or an out-of-range value occurs, DB2 returns the value NULL combined with a +802 warning, provided that the faulty expression is in the outermost SELECT statement and affects only one row.

I

inhouse processing
A query is processed by DB2 for z/OS and not routed to an accelerator.

L

lock A means of preventing uncommitted changes made by one application process from being perceived by another application process and for preventing one application process from updating data that is being accessed by another process. A lock ensures the integrity of data by preventing concurrent users from accessing inconsistent data.

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

Special characters

<CHCMetaID>.DMMD_SIGNALS table 195

Numerics

24:00:00 39, 45, 175

A

About section, Accelerator view 66
accelerating static SQL queries
  bind option 6
  bind options 23
  preparing tables 92
acceleration conditions 21
accelerator
  adding to configuration 83
  canceling 125
  canceling tasks 125
  defining 83
  status 90
  task 125
  view encryption status 177
Accelerator view 66
  About section 66
  header 66
  Monitoring section 66
  Query Monitoring section 66, 119
  Query monitoring section, columns 116
  refresh interval 135
  regular accelerators 66
  Tables section 66
  virtual accelerators 75
accelerator-only tables 16, 66, 88, 100, 116, 205
  bind option for static SQL 23
  dynamic SQL queries against 21
  loading 93
  restrictions 48
  special register settings 21
  static SQL queries against 23
accelerator-shadow tables 3
  removing 4
accelerators
  adding 1
  basic concepts 5
  networking details 87
  removing 128
  updating software 169
access plan graph
  displaying 112
  example 7
  nodes 113
Access Server
  installing updates 169
  update packages 170
access, DB2 subsystem 79
accessibility features for this product xv
activation of software updates
  switching between software versions 171
add
  columns to DB2 for z/OS table 103
Add Tables wizard 88
adding
  accelerators 1
  virtual accelerators 86
Administration Explorer
  folders 61
  menus 61
aggregate SQL functions 31
aliases 55
ALTER
  TABLE, required actions 148
  TABLESPACE, required actions 148
ALTER TABLE operation, incremental update function 140
analyze queries 107
archive
  partition data 151
  table data 151
archived partition (data), restoring 157
array data type 55
ASCII 28
authentication 83
automatic notifications and creation of PMRs 179

B

balancing the workload 92
basic concepts
  accelerators 5
  access plan graph 7
  distribution keys 9
  organizing keys 11
  selecting tables for query acceleration 7
  system table entries 21
  tables 6
  unsupported data types 6
  virtual accelerators 6
  zone maps 12
best practices
  distribution keys 9
  organizing keys 11
bind options for accelerating static SQL queries
  GETACCELARCHIVE 23
  QUERYACCELERATION 23, 205
broadcast 2
buffer overflow 39

C

call-home function 179
canceling tasks 125
CAST specification 31
CCSID
  1200 28
  1208 28
  367 28
  37 28
  819 28
  ASCII 28

© Copyright IBM Corp. 2009, 2017

CCSID *(continued)*
EBCDIC 28
UNICODE 28
CDC
event messages 179
signals table, authorizations 195
troubleshooting 195
changing
adding columns to table 103
distribution key 100
organization keys 100
table schema 103
CHAR (n) 40
clear trace information 135
coded character set identifier
See CCSID
CODEUNITS16 31
CODEUNITS32 31
collating sequences (sort orders) 39
columns added to DB2 for z/OS 103
conditions
query routing to an accelerator 26
conditions for query acceleration 21
configuration, trace behavior 129
configure tracing 66, 129
connection
data server 1
database 1
database server 81
profile 79
connection profile 1
correlated subqueries 39
COUNT_BIG 39
create objects in connected databases 105
create reference tables 162
create referencing accelerator-only tables 162
creating PMRs
PMRs 179
credentials 83
CURRENT GET_ACCEL_ARCHIVE 155
CURRENT QUERY ACCELERATION 155, 205
CURRENT QUERY ACCELERATION special register 21

D

data consistency 38
data loss, incremental update function 140
data sharing group 63, 66, 93
data slices 5
data types
DB2 for z/OS 28
Netezza 28
ranges 28
supported 28
unsupported 28
database connection profile 1, 79
database request modules 26
database server
connection 81
exploring 82
date
format 46
DATE(numeric_exp) 46
DB2 10 for z/OS, incompatibilities 51
DB2 11 for z/OS
aliases 55
array data type 55
DB2 11 for z/OS *(continued)*
global variables 55
grouping sets 55
named memory variables 55
new features 55
supergroups 55
synonyms 55
DB2 11 for z/OS, RBA and LRSN conversion and HPSS 159
DB2 Dynamic Statement Cache 21
DBCS 41
DECLARE CURSOR 26
defining
accelerator-shadow tables 88
data 2
delete trace information 66, 135
diagnostic information 129
distributing the workload to multiple accelerators 20
distribution keys
best practices 9
changing 100
impact
join performance 9
query acceleration 9
overview 2
specifying 100
division by zero 39
DNS_QUERYINFO_TABLE 107
DOUBLE 56
double-byte data 41
DRDA 26
DSNX881I messages
structure of 179
SYSLOG 179

E

EBCDIC 28, 41
encryption
view status 177
enhancements in this version xiii
environment variables
AQT_HOST_PACKAGE_DIRECTORY 169, 170, 173
evaluation, predicate 42
event viewer 145
event viewer for incremental updates 145
exceptions 42
EXPLAIN
function 6
information 6, 107
tables 107
tables, creating 107
exploring a database server 82
export of table specifications 127

F

FAILBACK processing
overview 21
restrictions 21
federated access 159
create reference tables 162
create referencing accelerator-only tables 162
grant access to DB2 subsystem 162
grant access to remote tables 162
remove reference tables 163, 164
remove referencing accelerator-only tables 163, 164

federated access *(continued)*
revoke access 163
federation 159, 162, 163, 164
firewall 183
FLOAT 40, 56
floating-point error tolerance 56
format
date 46
time 46
FTP server configuration for Save Trace function 134

G

GB18030 characters, incremental update function 140
GET_ACCEL_ARCHIVE 155
GETACCELARCHIVE 23
global variables 55
glossary 211
grant access
to remote DB2 subsystem 162
to remote tables 162
grouping sets 55
GUI elements 61

H

hash partitioning 9
header, Accelerator view 66
health, system 179
heartbeat 179
HFP 56
high-performance storage saver
See also storage saver
restoring partition data 157
High-Performance Storage Saver
archiving partition data 151
high-performance storage saver (HPSS) 15
HPSS
See also high-performance storage saver
See also storage saver
archiving partition data 151
LRSN conversion from 6-byte to 10-byte, DB2 11 for z/OS 159
RBA conversion from 6-byte to 10-byte, DB2 11 for z/OS 159
restoring partition data 157

I

IBM DB2 Analytics Accelerator for z/OS
Console 83
installing updates 169
migrating 165
setup 59
update packages 170
updates, installing 165
IBM DB2 Analytics Accelerator Studio
language settings 168
perspective in IBM Data Studio 168
workspace directory 168
IBM InfoSphere Data Replication for z/OS
See CDC
IBM Netezza Analytics 106
ibmdbR 106
IDT
See in-database transformation
IEEE754 56
importing table specifications 127
improvements in this version xiii
in-database transformation 16
incompatibilities 39, 47
incremental updates
ALTER TABLE operation 140
data loss 140
description 136
event log 135
event messages 179
excluding tables (disabling) 140
GB18030 characters 140
including tables (enabling tables for) 140, 195
incompatibilities 47
monitoring 143
REORG 140
ROTATE PARTITION 140
start 138
stop 138
use cases 136
using the event viewer 145
warnings 146
inhouse query processing 21
installation, updates 165
interface elements
Administration Explorer 61
menu bar 61
introduction 1
isolation levels 38

L

language settings, IBM DB2 Analytics Accelerator Studio 168
LEFT 43
length, rows 42
list tasks 125
load data 3
loadaccelerator-only tables 16
tables 93
LOBVALA 192
LOCATE 43
LOCATE_IN_STRING 43
location name 85
log record sequence number
See LRSN
loss of data, incremental update function 140
LOWER 43
LRSN conversion from 6-byte to 10-byte in DB2 11 for z/OS 159

M

mapping
Netezza states to DB2 SQL code 197
maximum row length of tables 39
MBCS 41
MEDIAN function 31
menu bar 61
migration 165
mixed data 41
modification of connected databases 105
monitoring 143
monitoring incremental updates 143
Monitoring section, Accelerator view 66
MONTHS_BETWEEN 44

moved
  partitions 15
  tables 15
multibyte data 41

N

named memory variables 55
Netezza error codes 197
Netezza Firmware (FDT)
  transferring updates 173
  updating 173
Netezza Host Platform (HPF)
  transferring updates 173
  updating 173
Netezza host software, transferring update 66
Netezza Platform Software (NPS)
  installing updates 169
  update packages 170
networking details 87
new features xiii
new features in DB2 10 for z/OS 51
NEXT_DAY 44
NPS
  See Netezza server software

O

Object List Editor 63
OCTETS 31
organizing keys
  best practices 11
  changing 100
  overview 2
  specifying 100
  table size 11
out-of-range values 39

P

pairing code 83
parameter markers 41
partition (data), restoring 157
perspective, IBM DB2 Analytics Accelerator Studio 168
PMR 129, 134
  See problem management record
PMRs, creating 179
ports 183
precision of timestamp 51
predicate evaluation 39, 42
preferences 135
problem management record 129, 134
process flow 4

Q

queries, running 105
query
  evaluation 6
  matching 6
  redirection 6
  visual representation 112
query acceleration
  disabling 3, 4, 99
  enabling 3, 4
query acceleration *(continued)*
  enabling tables 98
  reason codes 107
  selecting tables 7
query history 192
query monitoring 119
  columns 116
Query Monitoring section in Accelerator view 192
Query Monitoring section, Accelerator view 66
query routing to an accelerator
  conditions 21, 26
  configuration 21
  floating-point error tolerance 56
  incompatibilities 39
  isolation levels 38
  restrictions
    result data types 35
    special registers 35
    subtypes 35
  result data types 35
  supported queries 26
  supported special registers 34
  supported SQL functions 31
query routing to an accelerator, incompatibilities with DB2 10 for z/OS 51
QUERY_ACCEL_OPTION 205
QUERY_ACCELERATION 205
QUERY_ACCELERATION ZPARM 21
QUERYACCELERATION 23, 205

R

R programming language, support for 106
random distribution 2, 9
RBA conversion from 6-byte to 10-byte in DB2 11 for z/OS 159
REAL 56
reason code 205
reason codes 107
redistribution 2
referencing accelerator-only tables 159
referencing DB2 subsystem 159
refresh interval of Accelerator view 135
regular accelerators, Accelerator view for 66
relative byte address
  See RBA
remove
  accelerators 128
  tables 126
remove reference tables 163, 164
remove referencing accelerator-only tables 163, 164
removeaccelerator-shadow tables 3
REORG in connection with incremental updates 140
replication engine
  installing updates 169
  update packages 170
replication events 135, 145
restoring
  partition data 157
  table data 157
restrictions
  accelerator-only tables 48
RETAIN server 134
revoke access to remote DB2 subsystem 163
ROTATE PARTITION, incremental update function 140
ROUND 45
row length 42

running queries 105

S

Save Trace function, FTP server configuration 134
save trace information 66, 131, 133, 147
saving storage in z/OS
  See storage saver
scalar SQL functions 31
schema change 103
schema filters 79
selecting tables
  access plan graph 7
  basic concepts 7
  overview 2
SET CURRENT QUERY ACCELERATION 205
sharing credentials 83
signals table 195
simulation 6
simulator for query testing and evaluation 86
SMP/E 170
software update transfer 66
software version
  activation 171
  switching 66, 171
sort order (collating sequence) 39
space saving 151
  See storage saver
special register 155
special registers
  equivalent Netezza functions 34
  replacing 34
SQL code 197
SQL scripts 105
SQLSTATE 197
starting incremental updates 138
states of tables 13
static SQL queries 6, 92
static SQL queries, bind options 23, 205
status of an accelerator 90
stopping incremental updates 138
storage saver
  archiving partition data 151
  description 149
  restoring partition data 157
  restrictions 149
strings in TIME function 47
SUBSTR 45
supergroups 55
supported SQL functions 31
switching between software versions 171
synchronization of table data 38
synchronize table schemas 103
synonyms 55
system
  health 179
  monitoring 179
system table entries 21

T

table skew threshold 135
table specifications
  exporting 127
  importing 127
tables
  defining 88
  disabling query acceleration 99
  enabling for query acceleration 98
  loading 93
  possible operations depending on state 13
  removing from accelerator 126
  selecting 7
  states 13
Tables section, Accelerator view 66
task completion order 4
tasks
  overview 79
  prerequisites 79
temporal tables 51
time
  format 46
  representations 39
  values 39
TIME function
  strings 47
TIMESTAMP 46
timestamp precision 51
tracing 129
  CDC Capture Agent 133, 147
  clear information 135
  configuration 129
  delete information 135
  IBM InfoSphere Data Replication for z/OS 133, 147
  incremental updates 133, 147
  performance impact 129
  profiles 129
  save information 131
  trace file location 131
transferring update
  Access Server 170
  IBM DB2 Analytics Accelerator for z/OS software 170
  Netezza Firmware (FDT) 173
  Netezza Host Platform (HPF) 173
  replication engine 170
TRANSLATE 45
troubleshooting 183
TRUNC 45
truncation 40
truncation of values 39

U

UNICODE 28
Unicode SQL statements 41
update
  authentication 1
update, credentials 66
updating
  Access Server 169
  credentials 83
  IBM DB2 Analytics Accelerator software 169
  IBM DB2 Analytics Accelerator Studio from IBM Installation Manager 166
  Netezza Firmware (FDT) 173
  Netezza Host Platform (HPF) 173
  Netezza Platform Software (NPS) 169
  replication engine 169
UPPER 45
user IDs 79

V

view encryption statuss 177
virtual accelerators
  adding 86
  basic concepts 6
  contain EXPLAIN statement 107
  start 75
  stop 75
Visual Explain Graph 6
visual representation 112

W

warnings 146
workload balancing 20, 92
workspace directory, IBM DB2 Analytics Accelerator Studio 168

X

XMLAGG function 31

Z

zone maps 2, 12
ZPARMs
  QUERY_ACCEL_OPTION 205
  QUERY_ACCELERATION 205

Notices

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

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

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

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

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

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

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

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

© Copyright IBM Corp. 2009, 2017

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

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

Trademarks

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

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

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

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

Netezza and NPS are trademarks or registered trademarks of IBM International Group B.V., an IBM Company.

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

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

Product Number: 5697-DA5

SH12-7061-05