Version 5 Release 1

IBM Db2 High Performance Unload for z/OS
User's Guide
Including and excluding partitions................................................................. 242
Specifying COPYDDN file parameters .......................................................... 244
Specifying UNLOAD file parameters ............................................................ 245
Specifying options for the SELECT statement ................................................ 247
Updating the list of SELECT statements.......................................................... 249
Selecting tables ............................................................................................. 252
Processing long names.................................................................................... 254
Specifying the format of the SELECT statement ............................................. 255
Selecting columns for the SELECT statement .................................................. 259
Sorting columns of a SELECT statement ........................................................ 261
Displaying and modifying the USER format of the SELECT statement ........................................... 262
Displaying the list of OUTDDN files............................................................... 266
Displaying and modifying LOADDDN file parameters .................................... 269
Generating JCL to run a Db2 HPU job.............................................................. 270
Saving JCL into a file ..................................................................................... 271
Updating the job card .................................................................................... 272
Selecting date formats ................................................................................... 273
Selecting time formats .................................................................................. 274
Selecting timestamp formats ......................................................................... 274
Selecting mask formats ................................................................................ 275

Chapter 7. Db2 HPU user exit........................................................................ 277
Register use................................................................................................... 277
Parameter block ............................................................................................ 277
Sample Assembler user exit .......................................................................... 279
Sample COBOL/2 user exit ........................................................................... 282

Chapter 8. Troubleshooting.......................................................................... 285
Decreasing the amount of necessary memory ................................................. 285
Tools Customizer troubleshooting ................................................................. 286
Gathering diagnostic information ................................................................. 286
Determining the trace data set name .............................................................. 287
Gathering diagnostic information ................................................................. 287
Messages and codes .................................................................................... 287
Tools Customizer messages ......................................................................... 287
Db2 HPU user abend codes .......................................................................... 359
Db2 HPU messages ..................................................................................... 364

Chapter 9. Reference...................................................................................... 475
Customization reference .............................................................................. 475
Recommended settings ................................................................................ 475
Discover process parameters ................................................................ ....... 479
Db2 HPU library dsnames ............................................................................ 480
Db2 HPU Db2 parameters .......................................................................... 481
Db2 HPU output data parameters ............................................................... 499
Db2 HPU Db2 Administration Tool and Db2 Launchpad parameters .......... 534
Db2 HPU conversion parameters ............................................................... 534
Db2 HPU file management parameters ...................................................... 535
Db2 HPU reporting parameters .................................................................. 538
Db2 HPU tuning parameters ....................................................................... 542
Db2 HPU sort parameters .......................................................................... 550
Customization templates .......................................................................... 552
Locating a parameter in Tools Customizer panels ...................................... 552
INZDB21X member .................................................................................... 577
Data types for output (TYPE keyword) ....................................................... 579
Numeric data ............................................................................................. 580
Nonnumeric data ........................................................................................ 580
IBM Db2® High Performance Unload for z/OS® (Db2 HPU) is a high-speed Db2 utility that unloads Db2 tables from either a table space or from an image copy. This book describes how to customize and use Db2 HPU (in batch and interactively), and includes a description of the messages that Db2 HPU issues.

The following topics are covered in this book:

- Introduction to Db2 HPU functions and states its requirements.
- Description of authorization and configuration of Db2 HPU after installation.
- Description of input to and output from Db2 HPU.
- Description of how to use Db2 HPU in batch mode.
- Description of how to use Db2 HPU interactively.
- Description of the Db2 HPU user exit. You can use this exit to customize your output.
- Description of the user abend codes and messages that are issued by Db2 HPU.
- Description of the use of the TYPE keyword in the SELECT statement.
- Examples of Db2 HPU jobs.
- Syntax of Fast Unload that is compatible with the syntax of Db2 HPU.
- Syntax of UNLOAD PLUS that is compatible with the syntax of Db2 HPU.
- Licensing and trademark information.

The technical changes for this edition are summarized in the Summary of Changes topic. Specific changes since the previous edition of this book are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Always check the Db2 Tools Product Documentation page for the most current version of this information:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html
Chapter 1. Db2 High Performance Unload overview

IBM Db2 High Performance Unload (Db2 HPU) is a high-speed Db2 utility for unloading Db2 tables from either a table space or from an image copy. Tables are unloaded to one or more files based on a format that you specify.

Topics:

• “What does Db2 HPU do?” on page 1
• “Utilities management solutions” on page 8
• “Db2 HPU benefits” on page 1
• “Db2 HPU compatibility” on page 6
• “Maximizing performance” on page 6
• “Db2 HPU process and components” on page 2
• “Db2 HPU terminology” on page 7
• “Hardware and software requirements” on page 7
• “Db2 HPU documentation and updates” on page 8
• “Accessibility features” on page 9

What does Db2 HPU do?

Db2 HPU helps you manage and control the unload activity. Db2 HPU can retrieve the data to unload without heavy SQL processing by directly accessing the VSAM or sequential files that contain the table space or from image copy data sets.

Db2 HPU also provides superior performance, especially in terms of CPU time and elapsed time. Unless you need to process a complex SQL statement, Db2 HPU typically maximizes performance by reducing processor usage and improves availability by keeping data unavailable for a shorter time.

Db2 HPU can do the following tasks:

• Rapidly unload table spaces
• Parallel processing of several unloads that access the same table space
• Unload against any image copy to eliminate interference with Db2 production databases
• Unload selected rows and columns
• Unload every $n$ rows and maximum rows
• Generate load control statements for subsequent reload
• Inspect, modify, or discard Db2 rows by using the Db2 HPU user exit
• Create output in multiple formats during a single unload

Db2 HPU benefits

By using Db2 HPU, you can alleviate some of the problems that are associated with unloading large amounts of data.

Large sequential reads of Db2 tables take a long time. Because the large amount of time that large sequential reads of Db2 tables take, the large scans that are required for unload are hard to schedule in the ever-shrinking batch window at Db2 installations.
Performance becomes critical when several unloads read the same table space concurrently. The associated Db2 buffer pool management can cause multiple programs to compete for the same data. This competition between multiple programs can result in the following challenges:

- Conflicts in the use of the Db2 buffer pool.
- Writing over buffers that might be serving several unloads.
- Multiple reads of the same Db2 pages.
- Potential channel conflicts.

Db2 HPU relieves these problems. It provides a fast way to sequentially read and share a Db2 table space among multiple unloads. Db2 HPU scans a table space and creates the number of output files that you specify, in the format that you specify. The output format can be any one of the following types:

- DSNTIAUL compatible.
- VARIABLE, which lets you quickly create variable-length records.
- DELIMITED, which lets you quickly create a delimited file that you can export to another platform.
- USER, which lets you specify virtually any type of conversion so that your output appears as you want it.
- EXTERNAL, which lets you quickly create fixed length human-readable formatted files.

Whenever possible, Db2 HPU processes requests to unload data from the same table space in parallel. You can create different output files during the same unload process at almost no additional cost. For example, you can unload a list of customers who have payments due this week and another list of customers whose birthdays are on the first day of the week. You can create these lists in a single execution of Db2 HPU at a fraction of the cost that is required by traditional dual unload executions.

You can also run Db2 HPU against image copies of the table space, avoiding interference with Db2 production databases.

### Db2 HPU process and components

Db2 HPU is a batch utility program that is used to unload Db2 data.

Db2 HPU contains an UNLOAD command and an optional SELECT statement. The SELECT statement syntax is compatible with the Db2 SELECT statement syntax.

By optimizing sequential reads of the table space, Db2 HPU reduces both the elapsed time and the CPU time that is required to process the unloads.

Db2 HPU contains the following components:

- Tools Customizer. You use this component to customize Db2 HPU.
- Db2 HPU interactive component. You can use this component to generate the Db2 HPU batch utility job. Although using this component is not required, inexperienced users might benefit by using it to generate the batch utility job.

### Related concepts

**Tools Customizer overview**

IBM Tools Customizer for z/OS (also referred to as Tools Customizer) standardizes many of the customization processes that are required to customize IBM Tools that run on z/OS.

**Db2 HPU interactive component**

The Db2 HPU interactive component can be used to generate the Db2 HPU batch utility job. Although using the interactive application is not required, it can be useful for inexperienced users in generating the batch utility job.

**Db2 HPU batch mode operation**
You use the Db2 HPU batch utility program to unload Db2 data.

**Tools Customizer overview**

IBM Tools Customizer for z/OS (also referred to as Tools Customizer) standardizes many of the customization processes that are required to customize IBM Tools that run on z/OS.

Tools Customizer provides a consistent ISPF interface to ensure that the customization process is the same for all IBM Tools products and solution pack components. It also provides the ability to "discover" parameter values from products or solution pack components that you previously customized manually or by using Tools Customizer.

**Features and benefits**

Tools Customizer provides the following features:

- A single, consistent ISPF interface ensures that the customization process is the same for all IBM Tools products and solution pack components.

- A Discover EXEC discovers values for common product, LPAR, and Db2 parameters from a product or solution pack component that you previously customized manually or by using Tools Customizer. Each IBM Tools product and solution pack component has a unique Discover EXEC. The discovered parameters are stored in the data store. If the product or solution pack component that you want to customize exists in the Tools Customizer data store, Tools Customizer issues a warning before it overwrites existing values. Use the Discover EXEC by issuing the DISCOVER command on the Customizer Workplace panel.

- The data store retains discovered and manually specified parameter values. Because the parameter information is persistently stored, you have to manually specify or discover parameter values only once. Tools Customizer uses these parameter values where they are applicable.

- A metadata repository contains the members that define the following customization attributes for products and solution pack components:
  - Parameters, tasks, and steps for the product or solution pack component to be customized. Some product or solution pack parameters, tasks, and steps are required.
  - LPAR parameters for the local LPAR. All of the LPAR parameters are required.
  - Db2 parameters for the Db2 subsystem, Db2 group attach name, or Db2 data sharing member on which you will customize the product or solution pack component. All of the Db2 parameters are required.

- Default values are provided for product parameters and solution pack component parameters, LPAR parameters, and Db2 parameters. The default values show examples of how to complete fields.

**What's new in Tools Customizer**

This topic summarizes the technical changes for this edition. New and changed information is denoted either by a character in the left margin or by blue carets (Symbol that indicates the start of new or changed content) and (Symbol that indicates the end of new or changed content) at the beginning and end of the new or changed information. Editorial changes that have no technical significance are not noted.

- Using the VERIFY/VERIFYOFF command, you can turn on and off parameter verification of product or component parameters.

- Options can be displayed or hidden on certain panels. For more information, see “Changing display options” on page 39.

- You can now specify specific volumes in which to contain the customization library data set, the discover output data set, and the data store data set.

- You can sort and filter data in panels by column. For more information, see “Sorting and filtering columns” on page 40.

- Tools Customizer in a multi-LPAR environment is now supported.

- You can now specify the customization library on the Workplace panel.
• Version and Name columns were added to the **CCQPHLQ** panel where the name field is the name of a product or pack you previously worked with and the version field is the version of the product or pack you previously worked with.

**Service updates and support information**

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:


**Product documentation and updates**

Db2 Tools information is available at multiple places on the web. You can receive updates to Db2 Tools information automatically by registering with the IBM My Notifications service.

**Information on the web**

The most current version of this information is available on IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter

A PDF version of this information is available on the Db2 Tools Product Documentation web page; however, IBM Knowledge Center is updated more frequently than PDF books. The Db2 Tools Product Documentation web page is located at:


IBM Redbooks® publications that cover Db2 Tools are available from the following web page:

http://www.redbooks.ibm.com

The IBM Information Management System website shows how IT organizations can maximize their investment in Db2 databases while staying ahead of today’s top data management challenges:


**Receiving documentation updates automatically**

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking register now.
3. When the My Notifications page is displayed, click Subscribe to select those products that you want to receive information updates about. The Db2 Tools option is located under Software > Information Management.
4. Click Continue to specify the types of updates that you want to receive.
5. Click Submit to save your profile.

**How to send your comments**

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM product documentation, use one of the following options:

• Use the IBM Knowledge Center Contact Us link.
• Use the online reader comment form, which is located at http://www.ibm.com/software/data/rcf/.
• Send your comments by email to comments@us.ibm.com. Include the name of the book, the part number of the book, the version of the product that you are using, and, if applicable, the specific location of the text you are commenting on, for example, a page number or table number.

Accessibility features
Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in this product enable users to perform the following activities:
• Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
• Customize display attributes such as color, contrast, and font size.
• Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  – z/OS ISPF User’s Guide, Volume 1
  – z/OS TSO/E Primer
  – z/OS TSO/E User’s Guide

These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Db2 HPU and Db2 Tools products scenarios
You can use Db2 HPU with some IBM Db2 Tools to quickly unload data.

The following scenarios show how Db2 HPU can enhance the functionality of other Db2 Tools:

Db2 HPU and IBM Db2 Administration Tool for z/OS
Db2 Administration Tool helps you manage Db2 environments efficiently and effectively. When you use the Db2 Admin Alter ALT and Migrate functions, you can use Db2 HPU within a work statement list. Additionally, when you use an ALTER table space redefine against a single table space, you can use Db2 HPU as the unload method.

Db2 HPU and IBM Db2 Object Comparison Tool for z/OS
Db2 Object Comparison Tool compares two sets of Db2 objects. With it, you can compare Db2 objects by creating a batch job in which you specify all aspects of the comparison. When you specify the method for unloading data, you can use Db2 HPU, if it is available.

Db2 HPU and IBM Db2 Object Restore for z/OS
Db2 Object Restore can restore dropped objects and all related dependencies automatically, even if they do not exist in the Db2 catalog. When Db2 Object Restore is configured to access the Db2 HPU product libraries, you can use Db2 HPU to help recover a single table from an image copy data set.

Db2 HPU compared to the Db2 UNLOAD utility
Db2 HPU has some advantages over the Db2 UNLOAD utility.

Db2 HPU uses the following features to simplify the unload process. The Db2 UNLOAD utility provides limited or no support for these features.

SQL support
You can use any SQL statement with Db2 HPU as long as DB2 YES is specified.
Join operations
You can perform join operations. However, performance will be similar to the performance of the DSNTIAUL program.

Multiple output formats
You can create output in multiple formats, such as DELIMITED, VARIABLE, and USER.

User exit
You can use the Db2 HPU user exit to customize the output data set that is created by a SELECT statement. By using the exit, you can examine, modify, or reject Db2 rows.

Image copy processing
You can specify that you want to unload data against either the last full image copy or a different image copy. Additionally, you can unload data from a dropped table by specifying the object ID (OBID). In this case, you must have re-created the table.

Db2 HPU compatibility
To reduce cost and simplify conversion, Db2 HPU offers limited compatibility with the syntaxes that other products use.
Db2 HPU supports the JCL that is used with Fast Unload for Db2, Version 3.1 and the JCL that is used with UNLOAD PLUS for Db2, Version 2.1.01; however, some features of the Fast Unload and UNLOAD PLUS products might be ignored or might be interpreted differently when they are issued by Db2 HPU. In most cases, the amount of work that is required to convert Fast Unload JCL and UNLOAD PLUS JCL to Db2 HPU JCL is reduced.
Db2 HPU supports the Fast Unload and UNLOAD PLUS syntaxes only to the extent that Db2 HPU can perform processing that is like the processing that is described in the Fast Unload and UNLOAD PLUS syntaxes. Many keywords are ignored, and some options are automatically converted to Db2 HPU syntax.

Maximizing performance
To maximize performance, Db2 HPU uses buffering, synchronization techniques, and parallel processing. The CPU cost of the unload can also be reduced by exploiting the zIIP processors.

Buffering
When reading data rows, Db2 HPU directly accesses the VSAM clusters that contain the table space. This direct use of VSAM takes maximum advantage of the buffering capability that is provided by VSAM, which means that an entire cylinder can be read with a single I/O.

Synchronization
Db2 HPU permits the parallel execution of several unload jobs that access the same table space; it permits this parallel execution by synchronizing the unloads.

Parallel processing
Unload requests that work on the same table space are processed in parallel whenever possible. Unload requests that work on different table spaces can be processed in parallel if they are specified in the global parallelism setting.

zIIP processors exploitation
Db2 HPU can exploit the zIIP processor (ZIIP option). The unload jobs run in native mode and the unload jobs involving complex formatting, whatever the processing mode, offload a significant part of their processing load to the zIIP processors.

Optimized partition filtering
The performance of logical unloads run in native mode against a partitioned table space and that specify a WHERE clause can be significantly reduced with the help of the OPTIMIZE_PART_RANGE feature. This feature is triggered either by specifying the OPTIMIZE_PART_RANGE YES option or by setting the Partitions range filter optimization (VUM033/OPTRGPRT) to YES.
**Db2 HPU terminology**

Db2 HPU includes several unique terms that you must understand before using Db2 HPU.

**logical unload**

Uses SELECT statements to enable you to filter the rows and columns that you want to unload and to specify output formats.

**SQL processing**

Db2 HPU passes an SQL statement to Db2 for processing.

**native processing**

Db2 HPU reads the data to unload directly from the data sets where it is stored.

**physical unload**

Starts unloading at the beginning of a table space and does not stop unloading until the entire table space is unloaded.

**Hardware and software requirements**

Db2 HPU requires the same hardware configuration that is required by Db2 for z/OS Version 10 or later.

Db2 HPU requires the following software:

- z/OS Version 2 Release 1 or later
- Db2 Version 10.1 - in New Function Mode - for z/OS or later

**Note:**

To exploit Db2 10 for z/OS functionality, Db2 HPU V4.2 with appropriate maintenance level (PM78638 for Toleration and PM91909 for exploitation) is required. Previous versions of Db2 HPU cannot exploit Db2 10 for z/OS functionality.

**Note:**

The typical installation and use of Db2 HPU assumes that the SCEERUN and SCEERUN2 Language Environment libraries are available through the linklist.

If this condition is not or cannot be fulfilled in your environment, you’ll need to add the appropriate control cards to make these libraries available to Db2 HPU for the following:

- the installation job generated for the INZCHECK template
- the Db2 HPU job generated from the Db2 HPU interface
- the JCL samples provided by the Db2 HPU User’s Guide

For the first two items, consider adding a JOBLIB control card in the user jobcards that can be specified in both Tools Customizer ("User Job Card Settings for Customization Jobs" fields accessible through the "0. User settings for Tools Customizer" option) and the Db2 HPU interface ("JCL/JobCard" option).

**Note:**

As Db2 HPU runs with the CEE POSIX(ON) mode, the user profiles of the following userid need to contain an OMVS segment:

- the userid on behalf of which the Db2 HPU job is submitted
- the userid defined by the VUM032/ACTLGUSR parameter

If this requirement is not fulfilled, Db2 HPU fails with ABENDU4080 and REASON=CEE05101.

If you want to perform conversions that imply non-SBCS CCSIDs or pairs of SBCS CCSIDs that are not supported by the SYSSTRINGS catalog table, you must install z/OS Support for Unicode.
Optional prerequisites

There are some optional prerequisites for specific use of Db2 HPU. Ensure the following conditional requirements are fulfilled if the use case applies:

- To use POSIX(ON) exits, the user must be permitted to the BPX.MAP FACILITY class profile.

Utilities management solutions

IBM solutions help IT organizations maximize their investment in Db2 databases while staying on top of some of today’s toughest IT challenges.

Today, the size and number of databases continue to grow, and increased data volumes are becoming harder to manage. With an increasing number of applications that need to maintain access to the data, database administrators can struggle to find time to move data when necessary. Extracting and migrating data have become time-consuming and complex processes, which makes it difficult to schedule unloads of large amounts of data. As a database administrator, you might face some of the following challenges:

- Large unload processes that involve sequential reading of Db2 tables.
- Slower processes because resources contend with each other for the same data during the unload process.

Db2 High Performance Unload is only one of several Db2 Tools products that can help you meet these challenges and achieve higher availability and better performance during data maintenance while enhancing the productivity of both database administrators and system programmers.

The following Db2 tools also provide solutions for managing utilities:

- Db2 Administration Tool
- Db2 Automation Tool
- Db2 Storage Management Utility
- Db2 Utilities Enhancement Tool
- Db2 Utilities Suite

More than ever, the tools that you use for extracting and migrating data can have a critical impact on the efficiency and availability of your IT environment. Reorganization tools from IBM can help with the performance of key functions such as unloading and reloading Db2 data without impacting data access. Designed for ease of use and flexibility, Db2 HPU is a fast and efficient tool to unload and extract data for movement across enterprise systems or for reorganization in-place.

Db2 HPU can be launched from either the Db2 Administration Tool or the Db2 Tools Launchpad. To help you save time and effort, Db2 HPU integrates with Db2 Utilities Suite to offer lists of objects that can be specified by using LISTDEF and TEMPLATE statements.

Db2 HPU documentation and updates

This topic explains where to find Db2 and IMS Tools information on the Web, and explains how to receive information updates automatically.

Db2 HPU information on the Web

The Db2 Product publications Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:

https://www.ibm.com/support/knowledgecenter/SSAUUV_5.1.0/topics/inz_landing.html

You can also access documentation for many Db2 for z/OS and IMS Tools from the Information Management Software for z/OS Solutions Knowledge Center:
Documentation for many Db2 Tools that run on Linux®, UNIX, and Windows systems can be found in the IBM Db2 Tools for Linux, UNIX, and Windows Knowledge Center:

http://www.ibm.com/support/knowledgecenter/

IBM Redbooks publications that cover Db2 and IMS Tools are available from the following Web page:

http://www.ibm.com/software/data/db2imstools/support.html

The Data Management Tools Solutions Web site shows how IBM solutions can help IT organizations maximize their investment in Db2 and IMS databases while staying ahead of today's top data management challenges:


**Receiving documentation updates automatically**

To automatically receive a weekly e-mail that notifies you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Support service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support service:

1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking register now.
3. When the My Support page is displayed, click add products to select those products that you want to receive information updates about. The Db2 and IMS Tools category is located under Software > Data and Information Management > Database Tools & Utilities.
4. Click Subscribe to email to specify the types of updates that you would like to receive.
5. Click Update to save your profile.

**Accessibility features**

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in Db2 HPU enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - z/OS Interactive System Productivity Facility (ISPF) User's Guide, Volume 1, SC34-4822
  - z/OS TSO/E Primer, SA22-7787
  - z/OS TSO/E User's Guide, SA22-7794

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.
Chapter 2. What’s new and changed in Db2 High Performance Unload

This section describes the functional and technical changes for this edition. New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no functional or technical significance are not noted.

Topics:
- “What’s new in Db2 HPU” on page 11
- Operational requisite changes
- What’s changed in Db2 HPU
- Deprecated functions in Db2 HPU V5.1
- New messages
- Removed messages

What’s new in Db2 HPU

This topic summarizes the operational changes for this edition.

Performance
- Thanks to an optimized new architecture and a partial 64-bit implementation, the mean CPU cost of Db2 HPU jobs has been decreased.
- Db2 HPU can exploit the zIIP processor (ZIIP option). The unload jobs run in native mode and the unload jobs involving complex formatting, whatever the processing mode, offload a significant part of their processing load to the zIIP processors.
- The performance of logical unloads run in native mode involving a partitioned table, and a WHERE clause is enhanced by not processing the partitions that do match the WHERE clause (OPTIMIZE_PART_RANGE option). This enhancement exploits the Db2 for z/OS part range table populated by the EXPLAIN PLAN statement.
- In order to help understand the performance of unloads run in SQL mode, messages are issued to list the parameters or attributes that have a potential influence on the Db2 performance. These messages (INZU478I, INZU479I and INZU480I) are issued for both the user-handled parameters and the implicit attributes that Db2 HPU adds to the user-specified SELECT statement when it defers the data selection to Db2.
- The "Number of simultaneously opened LOB LDS(VUU080/MXOPNLLD)“ allows to specify how many LDS of a multi-LDS LOB tablespace can be kept allocated and opened simultaneously for unloading LOB data in native mode. This allows to ensure better performance when LOB data is unloaded.
- When unloading from automatically selected image copies through the COPYDDN LAST_IC or COPYDDN -n option, Db2 HPU may select image copies that require to be sorted, such as inline image copies. In terms of performance, unloading from a regular image copy that do not require to be sorted is more efficient. The EXCLUDE_FIC TO_SORT option allows to request Db2 HPU not to choose image copies that require to be sorted thus avoiding the related extra processing time.
- The output formatting task for unloads that involve CCSID conversions performed by Db2 HPU itself - either in native or SQL mode - are eligible to offload to zIIP processors.

Db2 compliance
- Db2 HPU supports (exploits or tolerates) the Db2 for z/OS V12.1 specific features available at various function levels as indicated below:
<table>
<thead>
<tr>
<th>Function level</th>
<th>Feature</th>
<th>compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>506</td>
<td>support for implicitly dropping explicitly created universal and LOB table spaces</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>506</td>
<td>alternative function names support</td>
<td>supported in SQL mode only with no specific maintenance</td>
</tr>
<tr>
<td>505</td>
<td>improved concurrency for rebind of executing packages</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>505</td>
<td>automatic page sample by default for RUNSTATS</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>505</td>
<td>improved DECFLOAT data type support</td>
<td>supported (toleration) with maintenance for APAR PH21189</td>
</tr>
<tr>
<td>505</td>
<td>transparent encryption for data columns</td>
<td>supported: in both native and SQL mode with maintenance for APAR PH14986</td>
</tr>
<tr>
<td>505</td>
<td>temporal and archive transparency for WHEN clauses on triggers</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>504</td>
<td>new support for data with IBM Z hardware-based Huffman compression of Db2 data</td>
<td>supported: in both native and SQL mode with maintenance for APAR PH18085</td>
</tr>
<tr>
<td>504</td>
<td>the ability to prevent the creation of certain new deprecated objects types</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>504</td>
<td>new support for certain built-in functions by pass-through to IBM Db2 Analytics Accelerator</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>504</td>
<td>new SQL syntax alternatives for certain special registers and NULL predicates</td>
<td>supported in SQL mode with no specific maintenance -supported in native mode with maintenance for APAR PH22201</td>
</tr>
<tr>
<td>503</td>
<td>support for IBM Db2 AI for z/OS</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>503</td>
<td>replication of system-period temporal tables and generated expression columns</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>503</td>
<td>a change to temporal auditing support for temporal data</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>503</td>
<td>a new console message that indicates when catalog levels or function levels change</td>
<td>no impact on Db2 HPU</td>
</tr>
<tr>
<td>502</td>
<td>management of key labels for z/OS DFSMS data set encryption</td>
<td>supported in both native and SQL mode with PH14986</td>
</tr>
<tr>
<td>502</td>
<td>explicit casting of numeric values to GRAPHIC or VARGRAPHIC.</td>
<td>supported: in SQL mode only with no specific maintenance</td>
</tr>
<tr>
<td>Function level</td>
<td>Feature</td>
<td>compliance</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>501</td>
<td>the LISTAGG built-in function produces a list of all values in a group</td>
<td>supported in SQL mode only with no specific maintenance</td>
</tr>
<tr>
<td>500</td>
<td>function level 500 (V12R1M500) represents the first opportunity for applications to take advantage of most new capabilities in the initial Db2 12 release, including new SQL capabilities and subsystem parameter settings. In many respects, function level 500 is analogous to new-function mode in previous releases.</td>
<td>supported (exploitation or no impact depending on the considered feature) except for the LOB compression</td>
</tr>
<tr>
<td>100</td>
<td>In function level 100 (V12R1M100), Db2 runs on Db2 12 code, and virtual storage and optimization enhancements in the initial Db2 12 release become available. However, most new function, including and subsystem parameter changes and new SQL capabilities remain disabled. In many respects, function level 100 is analogous to conversion mode in previous Db2 releases.</td>
<td>supported</td>
</tr>
</tbody>
</table>

- The consistency between the Db2 HPU native and SQL mode can be ensured automatically by requesting Db2 HPU to use the Db2 BIF_COMPATIBILITY settings instead of its own value set through the BIF_COMPATIBILITY subparameter of the Db2 HPU additional features (VUU030/ULOPTNS) parmlib parameter. The setting mode can be chosen through the "BIF_COMPATIBILITY_FROM_DB2_SETTINGS" subparameter of the VUU030/ULOPTNS parameter.
- The "Db2 plan name of the Call Level Interface (VUX044/CLIPLAN)" parmlib parameter allows to specify the plan name of the Db2 for z/OS Call Level Interface (CLI) when the default Db2 name DSNACLI was not kept.
- The Pervasive Encryption feature is exploited. Unloading data from Db2 objects which have underlying data sets encrypted using ICSF key label is possible in SQL mode and in native mode regardless of the data source (underlying VSAM data sets of the online Db2 objects or image copy data sets). For a Db2 like behavior - i.e. any user that has the requested privilege to run a given SELECT statement involving encrypted table spaces or index spaces can also run a Db2 HPU unload specifying the same SELECT statement -, set the "User who reads encrypted VSAM data set (VUM037/CRYPTUSR)" parameter with a userid that has the authorization to access the key label for any of the encrypted data set involved by your Db2 HPU jobs.
- The stop of the unload can be ensured whenever Db2 HPU cannot generate requested LOAD control cards that are fully compliant - i.e. directly usable - with the Db2 LOAD utility by specifying the ULRLDRC REJECT option or by setting the parmlib parameter VUU082/ULRLDRC to REJECT.
- The new subparameter ORDER_CLUSTER_BHV of VUU030/ULOPTNS, allows Db2 HPU to remove 'ORDER CLUSTER' clauses from unsupported SELECT statements (processed by SQL). It prevents Db2 HPU unloads to stop because of SQL error -199.
- Db2 HPU can now exploit the ADMIN_INFO_SYSPARM stored procedure to retrieve the value of the DSNZPARM parameters. This avoids undesirable IFCID 376 records be triggered when Db2 HPU retrieves the value of the BIF_COMPATIBILITY parameter. This also allows Db2 HPU to determine the
value of the TEMPLATE_TIME parameter (more details in the following note). This way of retrieving the DSNZPAM parameters is enabled by specifying USE_ADMIN_INFO_SYSPARM(YES) for the VUU030/ULOPTNS parameter.

- The time and date variables used in the data set names patterns of TEMPLATE statements can now be expanded considering local time instead of gmt/utc. If USE_ADMIN_INFO_SYSPARM(YES) is specified for the VUU030/ULOPTNS parameter and the Db2 settings specify TIME LOCAL, the date/time variables are expanded with local time where they were formerly unduly expanded with utc. They suboption TIME of TEMPLATE statements is supported.

- The data from table spaces compressed with Huffmann compression - available when the dynamic zPAM TS_COMPRESSION_TYPE is set to HUFFMAN - and from their image copies can be unloaded in native mode.

**Input**

- The unload of the data from FlashCopy data sets no longer requires the image copies to be referenced by the Db2 for z/OS catalog: the COPYDDN LAST_IC or COPYDDN integer was formerly mandatory to unload from FlashCopy data sets. The FlashCopy data set can be directly allocated by the Db2 HPU step and used through the COPYDDN ddname option (FLASHCOPY suboption).

- Db2 HPU is able to unload the data from a set of image copies made of a Full Image Copy and one or several Incremental Image Copies allocated under a single DD name (MERGE_IIC suboption of the COPYDDN ddname option).

- The uncatalogued image copies can be unloaded by using the COPYDDN LAST_IC or COPYDDN -integer options provided SEARCH_UNCATALOGED_IC(YES) is specified for the “Db2 HPU additional features (VUU030/ULOPTNS)” parameter.

- The LISTDEFTBV statement allows to select a list of tables and/or views through a TABLE selection criterion that directly applies to the table or view names (as registered in the SYSIBM.SYSTABLES Db2 catalog table) the LISTED statement that allows neither to select views nor to select some tables from a give table space (if one table matches the TABLE selection criterion, all the tables of the table space it belongs to are also selected). Like the LISTDEF statement, the name of a list defined through a LISTDEFTBV statement is meant to be used as an argument of a FROM LIST clause.

- When unloading from automatically selected image copies through the COPYDDN LAST_IC or COPYDDN -n option, the EXCLUDE_FIC TO_SORT option allows to request Db2 HPU not to choose image copies that require to be sorted which is especially useful regarding performance purposes. See the "Performance" section above for more details.

- The LISTDEFTBV statement allows to specify a list of tables and/or views by specifying a selection criterion that directly applies to the table or view names (as registered in the SYSIBM.SYSTABLES Db2 catalog table) unlike the selection criterion of the LISTDEF statement that applies to the tale space name. Like the LISTDEF statement, the name of a list defined through a LISTDEFTBV statement is meant to be used as an argument of a FROM LIST clause.

**Output**

- A new flexible date, time, and timestamp format is available (column-format-option). It is described by a string that specifies the sequence of the fields (year, month, date, hour, etc.) from the date, time, and timestamp to be displayed and the separators to be used.

- The DDL of the unloaded table can be generated within the file containing the LOAD control cards. This feature can be requested with the LOADDDL TABLE option of the LOADDND block.

- The "Oldest year assumed for DECIMAL to DATE conversions when no explicit century given by the DECIMAL source data (VUU077/ULCENT)" parmlib parameter allows to specify the rule to apply when converting decimal values into dates when the century digits are missing from the string value.

- Parmlib parameters allow to set the default value for the NULLPAD option. A specific parameter is dedicated to every applicable format, as listed below:
  - "Fill NULL with padding character, format USER (VUU075/NULLPAD)" sets the NNULLPAD default to apply for FORMAT USER is requested.
- "Fill NULL with padding character, format VARIABLE (VUU075/NULLPAD)" sets the NNULLPAD default to apply for FORMAT VARIABLE is requested
- "Fill NULL with padding character, format EXTERNAL (VUU075/NULLPAD)" sets the NNULLPAD default to apply for FORMAT EXTERNAL is requested

With the DELIMITED format, the NULL values can be unloaded as strings defined through the NULLVAL option of the DELIMITED block syntax and the "NULL processing in DELIMITED format (VUU078/ULNULLFL)" parmlib parameter. The NULLSTRING feature of the UNLOAD PLUS syntax is supported.

- The "Request the display of implicit parameters (VUU083/DISPPARM)" parmlib parameter allows to request the values of the parameters not specified by the UNLOAD command to be systematically or conditionally written into a user-specified data set.
- A title line can be generated for the data unload in DELIMITED format by specifying the TITLE ON option of the FORMAT block.
- The "Expansion ratio for conversion to UTF8 CCSID (VUU068/ULUNIEXP)" parmlib parameter and its counterpart UNICODE_EXPANSION_RATIO option from the OPTIONS block allow to specify what expansion ratio (i.e. target data length divided by source data length) to apply to char/varchar data when a conversion from a non-UTF8 CCSID to a UTF8 CCSID is made.
- The LIMIT option of the TEMPLATE block allows to specify that the template switching feature applies, i.e. that an alternate TEMPLATE is to be used when the estimated amount of data to unload exceeds a given size. This feature can be enabled by specifying TEMPLATE_LIMIT(YES) in the VUU030/ULOPTNS setting.
- The behavior when unloading LOB or XML data in native mode without making sure of the consistency between the base table space and its related LOB or XML table space can be controlled through the "Processing when inconsistency LOB encountered (VUU084/ULLOBBHV)" and the "Maximum issued messages for inconsistency LOB (VUU085/ULLOBBHM)" parmlib parameters.
- Db2 HPU can handle large block for output data sets. The TAPELBI subparameter for VUU030/ULOPTNS parmlib parameter must be set to either ALL to enable this feature for any output data set or to NOSPANNED to limit the feature to the non-spanned data sets.
- The "Sign for zoned-decimal numeric values (VUU037/ULSIGZ)" parmlib parameter allows to specify not only the positive sign but also the negative sign for zoned-decimal data.
- The ALTERNATE FORMAT sub-option for the INTERNAL option from the FORMAT block allows to specify an alternate format to be used when the INTERNAL format specified for a SELECT statement with a FROM LIST clause is not applicable to some tables involved by the related LISTDEF statement.
- The negative sign character for zoned-decimal data can be specified through the ZONED_DEC_SIGN of the OPTIONS block and the "VUU037/ULSIGZ" parmlib parameter formerly used to set the positive sign of the zoned-decimal only. Due to this change, this parameter is now described as "Sign for zoned-decimal numeric values" in the Tools Customizer's interface.
- The value (V) for the subparameter of the PIC option and the VUU018/ULPIC counterpart parameter allows to request no byte sign to be used for formatting of the non-float numeric data.
- Db2 HPU can produce a report about its exploitation of the EXPLAIN statement when performing the OPTIMIZE_PART_RANGE feature (for logical unloadss processed in native mode) and the LOCK feature (for logical unloads run in SQL access mode). The production of this report is driven by the setting of the VUU0087/DISPEXPL parmlib parameter and, depending on the setting of the latter parameter, of the allocation of a user-defined DName to the Db2 HPU step.
- Creating output data sets - for OUTDDN and UNLDDN and LOADDDN - encrypted using ICSF key label is possible by specifying the KEYLABEL attribute either on the related DD or TEMPLATE statement.
- The INTO clause behavior if field-name without output-data-type (VUU089/INTORUL1)" parmlib parameter and its corresponding GLOBAL level sysin option INTO_RULES, allows to change the behavior of an INTO clause when a field-name is specified without its output-data-type. output-data-type can therefore be taken either from a REFORMAT clause that applies or from the type of the column of the resulting table.
• The DATE/TIME and NULL DATE/TIME delimiters used for the DELIMITED format that could be specified through the “DATE/TIME delimiter option (VUU031/DTDELIM)” and “NULL DATE/TIME delimiter option (VUU043/DTNULDLIM)” parmlib parameters can now also be specified at the job level respectively through the DATE_DELIM and NULL_DATE_DELIM option of the GLOBAL OPTIONS block.

Troubleshooting
• The Db2 HPU for z/OS status is displayed when running the QUIESCE utility is impossible. The text of the messages INZU176I and INZU406E (QUIESCE IMPOSSIBLE) now display the restricted state of the unloaded object that prevented the QUIESCE to be performed.
• The report produced by the DISPEXPL feature can help better understand the reason of a failure or an unexpected behavior in the following use cases:
  – an unload run in native mode involving the OPTIMIZE_PART_RANGE feature,
  – an unload run in SQL mode involving the LOCK feature.

Concurrence and consistency
• For logical unload processed in native mode, Db2 HPU can honor the LOCK YES option by starting the table space to unload in RO restricted state. The new parmlib parameter "Technique to lock the unloaded tablespace (VUU074/ULLOCKTE)" allows to specify the technique used to lock the unloaded tablespace when LOCK(YES) is requested.
• Db2 HPU can honor the LOCK YES option in SQL mode. The "Eligible processing mode(s) for locking data feature (VUU088/ULLOCKMO)" parmlib parameter allows to specify for which processing mode(s) the LOCK YES must be honored.

Easier troubleshooting for error at record level
• The HANDLE_RECORD_ID option of the GLOBAL OPTIONS block allows to get more information to determine the location of a record for which an error at record level is raised.

Syntax compatibility
• The UNLOAD PLUS CNTLCARDS DB2 is supported. This feature was formerly ignored and this situation reported through the message INZU027W CNTLCARDS DB2 NOT SUPPORTED, WILL USE CNTLCARDS DB2LOAD INSTEAD and a completion code of 8. The new behavior can be enabled through the setting of the CNTLCARDS_DB2 subparameter of the VUU030/ULOPTNS parmlib parameter.
• The ZONEDDECOVP option of the UNLOAD PLUS syntax is supported.
• Db2 HPU now supports the two new syntax alternatives for NULL predicate introduced by the Db2 V12 function level 504. ISNULL and NOTNULL can be used instead of IS NULL and IS NOT NULL.

Control and reporting
• The behavior of Db2 HPU when it cannot generate requested LOAD control cards that are fully compliant - i.e. directly usable - with the Db2 LOAD utility can be chosen through the ULRLDRC option or the VUU082/ULRLDRC parmlib parameter. This allow to request Db2 HPU either to ignore such a situation or to issue a warning messages when it occurs or to simply reject the situation by stopping the unload process.
• The DISPLAY_PARMLIB feature gives information about format and syntax-dependent parameters by describing in the header how the values of this type of parameter are coded and by marking the format dependent parameters with (1) and the syntax-dependent parameters with (2) as in the following example:

  VUU017/ULTMSTP *(1)* = U(TMSTP_B)
  VUU071/LOSSCHAR *(2)* = HPU(IGNORE),ULPLUS(IGNORE),FASTU(IGNORE)
**Operational requisite changes**

This topic summarizes the operational requisite changes for this edition.

**Db2 connection mode**
- Db2 HPU now uses the Db2 CLI. It must be enabled before the unload can be run properly.

**Db2 catalog access**
The User used when issuing SELECT using dynamic SQL (VUM032/ACTLGUSR) must have:
- a user profile that contains an OMVS segment
- and the SELECT privilege on the Db2 catalog tables listed below:
  - SYSIBM.SYSAUXRELS
  - SYSIBM.SYSCOLUMNS
  - SYSIBM.SYSCOPY
  - SYSIBM.SYSDATABASE
  - SYSIBM.SYSDATATYPES
  - SYSIBM.SYSFIELDS
  - SYSIBM.SYSFOREIGNKEYS
  - SYSIBM.SYSINDEXES
  - SYSIBM.SYSINDEXEXPART
  - SYSIBM.SYSINDEXSPACESTATS
  - SYSIBM.SYSKEYS
  - SYSIBM.SYSOBDS
  - SYSIBM.SYSRELS
  - SYSIBM.SYSTRINGS
  - SYSIBM.SYSSYNONYMS
  - SYSIBM.SYSTABLEPART
  - SYSIBM.SYSTABLES
  - SYSIBM.SYSTABLESPACE
  - SYSIBM.SYSTABLESPACESTATS
  - SYSIBM.SYSVIEWDEP
  - SYSIBM.SYSVIEWS

**OMVS threads used**
Db2 HPU uses OMVS threads for any batch processing. Consequently, the settings of some of the MAXTHREADTASKS and MAXPROCSYS and MAXPROCUSER z/OS Unix parameters have an impact on the Db2 HPU jobs.

**What’s changed in Db2 HPU**

This topic summarizes the operational changes for this edition.

**New default for "Use real-time statistics tables for size estimation (VUX036/RTSESTIM)"**

As taking the Real Time Statistics into account when estimating the amount of data to be unloaded increases the accuracy of the estimate, the shipped setting for "Use real-time statistics tables for size
estimation (VUX036/RTSESTIM)” is YES. Parmlib created with previous releases without specifying a value for VUX036/RTSESTIM specify NO for this parameters is kept as is when retrieved by the DISCOVER utility. If you want the new default to apply to your settings, either remove its value or specify YES.

**User-exit run with POSIX(ON)**

The use of user-exit run with the POSIX(ON) LE option:

- requires the permission of the user running the Db2 HPU job to the BPX.MAP FACILITY class profile.
- is not supported if the data to be unloaded involves a FIELDPROC or an EDITPROC.

**SELENUM processing when SELECT ... FROM LIST is specified**

When an unload is requested for a set of objects defined through a LISTDEF statement (SELECT ... FROM LIST(...)), the :SELENUM evaluation has changed. In such a case, the SELENUM processing used to depend on the value of the ONDEMAND_RESOURCE_ALLOCATION sub-parameter of the ULOPTNS parameter.

When ONDEMAND_RESOURCE_ALLOCATION(NO) was specified, each table to be unloaded was associated to a different value of SELENUM. With ONDEMAND_RESOURCE_ALLOCATION(YES), the SELENUM value was equal to 1 regardless of the table involved. As this sub-parameter is deprecated and the kept behavior corresponds to ONDEMAND_RESOURCE_ALLOCATION(YES), only the processing corresponding to this setting applies. In other words, the SELENUM ONDEMAND_RESOURCE_ALLOCATION(YES) now applies.

If your setting specified ONDEMAND_RESOURCE_ALLOCATION(NO), in order to ensure the uniqueness of the data set names generated by your TEMPLATE statement, consider adding a part related to the involved tablespace, either &DB and &TS (or &SN) or :UNLNUM. Using the &UNIQ (or &UQ) also ensure the uniqueness.

For example:

The following unload command involving several table spaces among which a multi-table table space (i.e. DSNDB06.SYSCONTX contains 3 tables) was valid in former releases provided the settings specified ONDEMAND_RESOURCE_ALLOCATION(NO).

With the following unload command:

```
LISTDEF L1 INCLUDE TABLESPACE DSNDB06.SYSTSDBA
   INCLUDE TABLESPACE DSNDB06.SYSUSER
   INCLUDE TABLESPACE DSNDB06.SYSCONTX
TEMPLATE TSUNLD
DSN 'hlq.&DB..SEL&SEL..DATA'
   DISP (NEW,CATLG,DELETE)
UNIT WORK
SPACE (6,6) TRK
GLOBAL OPTIONS TEMPLATESET( SEL = :SELENUM)
   REFORMAT( TYPE CLOB INTO VARCHAR(5000));
UNLOAD TABLESPACE
   SELECT * FROM LIST(L1)
OUTDDN(TSUNLD)
```

The following data set were created:

```
h1q.DSNDB06.SEL00001.DATA
h1q.DSNDB06.SEL00002.DATA
h1q.DSNDB06.SEL00003.DATA
h1q.DSNDB06.SEL00004.DATA
h1q.DSNDB06.SEL00005.DATA
```

From HPU 5.1, such an unload is invalid and Db2 HPU issues the following message: INZT034E TEMPLATE ERROR: TEMPLATE TSUNLD REDEFINES DSNAME hlq.DSNDB06.SEL00001.DATA PREVIOUSLY GENERATED FROM TEMPLATE TSUNLD

The unload can be made valid again by using adding the UNLNUM variable to the data set name defined by the TEMPLATE statement, as follows:
LISTDEF L1 INCLUDE TABLESPACE DSNDB06.SYSTSDBA
INCLUDE TABLESPACE DSNDB06.SYSUSER
INCLUDE TABLESPACE DSNDB06.SYSCONTX
TEMPLATE TSUNLD
DSN 'hlq.&DB..UNL&UNL..SEL&SEL.'
DISP (NEW,CATLG,DELETE)
UNIT WORK
SPACE (6,6) TRK
GLOBAL DB2 FORCE OPTIONS
TEMPLATESET( SEL = :SELNUM,
UNL = :UNLNUM)
REFORMAT( TYPE CLOB INTO VARCHAR(5000));
UNLOAD TABLESPACE
SELECT * FROM LIST(L1)
OUTDDN(TSUNLD)

With the unload command above, Db2 HPU creates the following datasets:

hlq.DSNDB06.U00001.S00001.
hlq.DSNDB06.U00001.S00002
hlq.DSNDB06.U00001.S00003
hlq.DSNDB06.U00001.S00004
hlq.DSNDB06.U00001.S00005

New technical parameters and options

As a result of the new internal implementation of Db2 HPU, some new technical parameters that allow to tune the internal behavior are needed. They concern a mechanism called inter-processes communication.

• Size of work areas buffers (VUX040/UBUFSIZE)
• Number of buffers of work areas (VUX041/UBUFNO)

These new parameters can be set at the unload job level by the new UNLOAD_BUFFERS. Keep the default setting for these parameters unless directed to change it by the IBM support.

Messages

INZU376I UNLOAD n SELECT n [PARTITION n] NUMBER OF RECORDS WRITTEN n
INZU407W UNLOAD n SELECT n [PARTITION n] MAX_EXPECTED_ROWS IS LESS THAN THE UNLOADED ROWS.
INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM n1 TO n2
PARTITIONS: n1, n2, [n4 - n5] ...
INZU508I TABLE creator.table HAS BEEN LOCKED IN SHARE MODE DURING UNLOAD PROCESS [LIMITED TO UNLOADED PARTITIONS]

Return codes for INZU392E and INZU401E

The return code associated with the following messages can be 0 if a later INZU538I message has been issued for the same unload task:

• INZU392E FORMAT INTERNAL IS NOT SUPPORTED FOR THE FOLLOWING REASON: reason. THE TABLE NAME IS table_name. Reason: XML OR LOB COLUMN FIELDPROC ON A COLUMN
• INZU401E UNLOADING A VIEW WITH FORMAT INTERNAL IS NOT SUPPORTED.
• INZU391E FORMAT INTERNAL IS NOT SUPPORTED FOR SELECT STATEMENTS THAT ARE PROCESSED BY DB2.

Extended scope for VUM031/QUITSUSR parmlib parameter

The scope of the VUM031/QUITSUSR parmlib parameter has been extended. This userid is used to issue SQL LOCK against the table to unload and a START RO command against the table. The parameter label in Tools Customizer has been updated as "User who execute DB2 processes on the table space (VUM031/QUITSUSR)".
Tuning and control of internal process

The SIZE option no longer requires the DDLDDN option and can be specified whenever COPYDDN option is specified. The SIZE option can therefore be used to specify a reliable value of the number of rows contained in the input image copy whenever Db2 HPU cannot estimate it from the Db2 HPU catalog information, which occurs when the DDLDDN option is specified but can also occur when the image copy data set is no longer referenced in the Db2 catalog or when it was taken in another Db2 subsystem.

The ZIIPMXPL and ZIIPMXPU parameters that could be specified through the VUX042/ZIIPMXPL and VUX043/ZIIPMXPU parmlib parameters can now also be specified by the UNLOAD command respectively through the ZIIPMXPL and ZIIPMXPU parameters of the Technical Parameters options block.

Settings recommendation

With the enhanced architecture of Db2 HPU version 5.1, the recommended value for both the “Support parallelism for unloading several partitions (VUU044/SUBTKSOF)” and “Enforce partition parallelism for unloading a table into a single file (VUU036/GBLPARAL)” is YES. As this new setting allows more parallelism when the data from a several data sets (Db2 LDS or image copy) of a partitioned table space is unloaded into a single output data set, an increase of the overall performance of such jobs is expected. Be aware of the related extra CPU and memory consumption and consider limiting the partition parallelism through the PARALLELISM (n,,) option of jobs involving a large number of partitions or through the VUX025/PARALLEL parmlib parameter if a large number of jobs are concerned.

Modified MAXERR feature

The specifications of this MAXERR feature (required either through the MAXERR option or the VUU073/ MAXERR parmlib parameters) that apply in Db2 HPU up to version 4.3 prevent to take benefit from the performance enhancements brought by the Db2 HPU internal architecture introduced by Db2 HPU 5.1. The MAXERR feature has be fitted with a new behavior that allows to exploit the current Db2 HPU version optimized performance. The MAXERR STRICT subparameter of the VUU030/ULOPTNS parmlib parameter allows to specify whether the genuine MAXERR feature behavior must be kept.

Deprecated functions in Db2 HPU V5.1

This topic summarizes the deprecated functions for this edition.

Db2 catalog access in SQL mode only

As Db2 HPU now systematically accesses the Db2 catalog in SQL mode, running the QUIESCE utility against the catalog is no longer needed to guarantee the consistency of the data read from the catalog. Therefore, the following related parameters and options are deprecated and ignored.

- VUM014/QUIESCAT : the behavior kept is the same as with QUIESCAT = NO, i.e. Db2 HPU does not issue QUIESCE against the catalog table spaces unless a request to unload them is made.
- VUM020/QUIESUSR : this parameter is no longer used since no QUIESCE of the catalog table spaces is run.
- VUM027/SQLACCES : the behavior kept is the same as with SQLACCES = YES, i.e. the data from the catalog are always retrieved via Db2.

The following related options from the Technical Parameters options block are also deprecated:

- QUIESCECAT
- SQLACCESS

Parameters for tuning internal behavior

As the internal implementation of Db2 HPU has been significantly changed, the following parameters that were only meant for optimization purpose, are no longer relevant:

- Parameters and options related to former sort operations
- VUM024/SRTNBVRE parmlib parameter
- VUM025/SRTVSMIN parmlib parameter
- VUM026/SRTVSMAX parmlib parameter
- SRTNBVRE option from the Technical Parameters block
- SRTVSMIN option from the Technical Parameters block

- Parameter related to parallelism processing
  - ULOPTNS/ONDEMAND_Resource_Allocation: the new behavior is similar to the one corresponding to ONDEMAND_RESOURCE_ALLOCATION(YES)

- Parameter and option related to internal tasks communication
  - VUX022/VBUFSIZE parmlib parameter
  - VBUFSIZE option from the Technical Parameters block

---

**New messages**

This topic lists the new messages for this edition.

- INZI524E UNABLE TO READ PARMLIB pds name/ddname (rc = return-code)
- INZI192E UNABLE TO LOAD THE DB2 LIBRARY (explanation)
- INZI193E ERROR: cli description error
- INZI214I THE SYNTAX KEYWORD keyword IS DEPRECATED AND IS IGNORED BY THE PROCESS.
- INZI215I THE VALUE value OF THE PARMLIB VARIABLE variable IS DEPRECATED AND IS IGNORED BY THE PROCESS.
- INZI195I DB2 STATUS OF THE PARTITION partition.number OF THE tablespace/indexspace
- INZI196I DB2 STATUS OF THE tablespace/indexspace dbname.tsname/dbname.isname: status
- INZU482E UNLOAD IMPOSSIBLE: AT LEAST value POSIX EXITS MUST BE RUN AT THE SAME TIME.
- INZU489W OUTEXIT exit_name CANNOT BE USED FOR SELECT STARTING AT POS(line, column)
- INZU524E functionality NOT SUPPORTED DUE TO VUU079/EXITHLL VALUE SET TO NO
- INZI210I ZIIP ENABLE - SET TO n %
- INZI211I ONLINE CPU(S)
  - n GENERAL CPU
  - p ZIIP CPU
  - q OTHER CPU
- INZI212I NO ZIIP AVAILABLE, ZIIP SET TO OFF
- INZI213W ZIIP ACTIVATION FAILED, RC=return_code, REASON=reason_code
- INZU476I PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS LIMITED TO n BY VUX042/ZIIPMXPL PARMLIB PARAMETER.
- INZU523I PARALLELISM ACTIVATED FOR TABLESPACE PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS LIMITED TO n BY VUX043/ZIIPMXPU PARMLIB PARAMETER.
- INZU525I NO PROCESS ELIGIBLE TO RUN ON ZIIP FOR UNLOAD num FROM TABLESPACE dbname.tsname.
- INZU526I NO PROCESS ELIGIBLE TO RUN ON ZIIP FOR UNLOAD num FROM UNDETERMINED TABLESPACE.
- INZCI0559 DDNAME: ddname ROWS: number
- INZI197I EXPLAIN PLAN SUCCESSFULL: QUERYNO = number, EXPLAIN_TIME = timestamp
- INZI199W EXPLAIN TABLE: creator.name IS MISSING
- INZI216E UNABLE TO DETERMINE CURRENT DB2 APPLCOMPAT

Chapter 2. What’s new and changed in Db2 High Performance Unload 21
• INZI217I DB2 APPLCOMPAT IS SET TO value
• INZI458E utility COULD NOT START A NEW THREAD OR PROCESS (CODE reason) (reason text)
• INZI527E DDL ANALYSING: option1 OPTION REQUIREs MISSING option2 OPTION
• INZI528E DDL ANALYSING: option1 OPTION IS INCOMPATIBLE WITH option2 OPTION
• INZI529I THE PARMLIB PARAMETERS HAVE BEEN READ FROM: path
• INZI800E ERROR ACCESSING TO A DATASET. ACTION: ALLOCATE RC: return-code REASON: reason
  DDN: ddname DSN: dsname
• INZU211I LOAD STATEMENT WARNING : UNABLE TO DETERMINE TABLE NAME FOR SELECT
  STARTING AT POS (line, column). LOAD STATEMENT MUST BE MODIFIED
• INZU241I DB2 HPU IS NOT RUNNING UNDER KEY 7
• INZU415I DB2 PARAMETER SET TO FORCE DUE TO SQLID KEYWORD
• INZU416E THE SQLID KEYWORD AND THE DB2 YES OR DB2 NO OPTIONS ARE INCOMPATIBLE
• INZU451E OPTION FORMAT INTERNAL IS NOT VALID WHEN USED WITH MORE THAN ONE TABLE
• INZU474I DATA TYPE FOR TARGET COLUMN column-name NOT SPECIFIED BY THE COLUMN-LEVEL
  FORMATTING REQUEST, DEFAULT TYPE type-name APPLIES
• INZU475E FLASH COPY DATASET MUST BE ALLOCATED FOR EACH UNLOADED PARTITION
• INZU478I ATTRIBUTES attributes_list HAS BEEN ADDED TO THE DB2 STATEMENT
• INZU480I MONO-ROW FETCH READ ACTIVATED WITH nb_rows ROWS PER FETCH
• INZU481I PART RANGE OPTIMIZATION ENABLE FOR dbname.tname
• INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM number TO number
• INZU486I NO OPTIMIZATION FOUND. number_of_partitions PARTITIONS KEPT
• INZU487W NO PARTITION TO UNLOAD, SELECT select_number IS CANCELED
• INZU519E datatype FORMAT format_string CONTAINS AN INVALID ELEMENT : element
• INZU520E MM ELEMENT IS INVALID, PLEASE USE MO FOR MONTHS OR MI FOR MINUTES
• INZU521E datatype FORMAT format-string MUST BEGIN WITH A VALID ELEMENT: element_list
• INZU522E element ELEMENT CAN BE USED ONLY ONCE
• INZU529W APPLCOMPAT string IS IGNORED FOR DB2 VERSION PREVIOUS TO V11 NFM
• INZU530I LOAD STATEMENT : INDDN FROM LOADOPT IGNORED BECAUSE OF LOADINDDN(YES)
• INZU531I CHECK IF THE SEMICOLON REQUIRED AT THE END OF GLOBAL STATEMENT IS CORRECTLY
  LOCATED
• INZU463E ERROR WHEN Generating THE CREATE TABLE STATEMENT OF THE TABLE : creator.table
  RETURN CODE : rc REASON CODE : reason SQLCODE : sqlcode
• INZU464W NO CREATE TABLE STATEMENT HAS BEEN GENERATED FOR SELECT number AS THE NAME
  OF THE TABLE HAS NOT BEEN IDENTIFIED
• INZU465W NO CREATE TABLE STATEMENT HAS BEEN GENERATED FOR SELECT number AS THE TYPE
  OF THE TABLE creator.table name IS NOT SUPPORTED.
• INZU542E FORMAT DELIMITED ERROR FOR SELECT select_number STARTING AT POS(line,column)
  MERGING INCONSISTENT RECORDS WITH OPTION TITLE ON INTO OUTPUT DDN=outdd IS NOT
  ALLOWED
• INZX135 VERSION 0 MISSING FOR dbname.tname. IF ALTER TABLE PROCESSED RECENTLY, TRY
  LATER OR REORGANIZE THE TABLESPACE
• INZU544W THE OUTPUT FIELD WILL BE EXPANDED WITH A RATIO OF ratio AS REQUESTED BY THE
  SETTING OF THE PARMLIB PARAMETER VUU068/ULUNIEXP. THE OUTPUT FIELD MIGHT NOT BE LONG
  ENOUGH FOR .FIELD column-name, TABLE table-name
• INZU545I THE OUTPUT FIELD WILL BE EXPANDED WITH A RATIO OF ratio AS REQUESTED BY THE SETTING OF THE PARAMETER UNICODE_EXPANSION_RATIO. THE OUTPUT FIELD MIGHT NOT BE LONG ENOUGH FOR FIELD column-name, TABLE table-name
• INZU546I THE OUTPUT FIELD WILL BE EXPANDED WITH A RATIO OF ratio AS REQUESTED BY THE SETTING OF THE PARAMETER UNICODE_EXPANSION_RATIO OR THE PARMLIB PARAMETER VUU068/ULUNIEXP. FIELD column-name, TABLE table-name
• INZ537E TEMPLATE template_base_name IS INCOMPATIBLE WITH alternate_template TEMPLATE
• INZ543W IN TEMPLATE template_name limit OPTION IS IGNORED FOR data_set_type
• INZ531E INVALID TEMPLATE DEFINITION FOR TEMPLATE alternate_template TEMPLATE
• INZ539E ALTERNATE FORMAT IS ONLY ALLOWED UNLOADING FROM LISTDEF
• INZ540E ALTERNATE FORMAT IS ONLY ALLOWED FOR INTERNAL FORMAT
• INZ552E LOCK CANNOT BE PERFORMED FOR SELECT no at POS(line,col)
• INZ553I LOCK IS NOT PERFORMED FOR SELECT no at POS(line,col) AS VUU088/ULOCKMO IS SET TO NATIVE_ONLY
• INZU555W THE PARMLIB PARAMETER VUU074/ULOCKTE IS FORCED TO LOCK DUE TO UNSUPPORTED SELECT TO LOCK
• INZX112 ddname UNIT-COUNT = cnt USED INSTEAD OF OPTIMAL UNCNT = cnt AS PER limit SPECIFICATION
• INZX113 ddname PRIMARY-QTY = PRIMARY-QTY USED INSTEAD OF OPTIMAL PRIMARY-QTY = PRIMARY-QTY AS PER limit SPECIFICATION

Modified messages

This topic lists the messages which content has changed.

• INZU523I PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS LIMITED TO n BY ZIIPMXPL PARAMETER.
• INZI192E UNABLE TO LOAD THE DB2 LIBRARY (explanation)
• INZU476I PARALLELISM ACTIVATED FOR TABLESPACE PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS LIMITED TO n BY ZIIPMXPU PARAMETER.
Removed messages

This topic lists the messages that are no longer issued from this edition on.

- INZGE0001 ERROR: WRONG VALUE SPECIFIED FOR variable IN THE INZTVAR MEMBER OR THE VARIABLE IS MISSING
- INZGE0002 ERROR: WRONG VALUE SPECIFIED FOR variable IN THE INZTVAR MEMBER IT MUST BE NUMERIC
- INZGE0003 ERROR: WRONG VALUE SPECIFIED FOR variable IN THE INZTVAR MEMBER ITS LENGTH IS LESS THAN THE MINIMUM LENGTH FOR THIS PARAMETER
- INZGE0004 ERROR: WRONG VALUE SPECIFIED FOR variable IN THE INZTVAR MEMBER ITS LENGTH IS GREATER THAN THE MAXIMUM LENGTH FOR THIS PARAMETER
- INZGE0102 INTERNAL ERROR: FILE file, LINE line. PLEASE CONTACT YOUR TECHNICAL SUPPORT
- INZGE0200 ERROR: ddname DDCARD SHOULD BE LINE A FILE WITH REFCM=VBS
- INZI367W THE CONFIGURATION PARAMETER name MUST BE NUMERIC.
- INZI365E INVALID VALUE FOR CONFIGURATION PARAMETER name.
- INZI366E THE MANDATORY CONFIGURATION PARAMETER name IS MISSING.
- INZU383I THE FOLLOWING KEYWORD IS IGNORED IN DB2 CM8*, CM9*, V10 ENFM AND LATER RELEASES
- INZU181I WARNING: PARMLIB VARIABLE QUIESCAT IS SET TO OFF
- INZU182I WARNING: PARMLIB VARIABLE QUIESCAT IS SET TO FORCE
- INZU183I ERROR: UNABLE TO QUIESCE DB2 CATALOG. PROCESSING TERMINATED
- INZU211W COLUMN col_name IN LOAD STATEMENT IS A CONSTANT. LOAD STATEMENT MUST BE MODIFY TO BE USABLE
Chapter 3. Db2 HPU customization

The customization process consists of authorizing and enabling Db2 HPU and using Tools Customizer to set parameter values and generate the customization jobs that you will submit.

**Prerequisite:** Before you can authorize and customize Db2 HPU, you must install Db2 HPU by following the instructions that are provided in the program directory.

Customizing Db2 HPU consists of the following steps:

*Table 1. Overview of Db2 HPU customization steps*

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Authorize and enable Db2 HPU.</td>
<td>“Authorizing and enabling Db2 HPU” on page 25</td>
</tr>
<tr>
<td>2</td>
<td>Optional: Make the Db2 HPU interactive component available through TSO.</td>
<td>“Optional: Creating a TSO command for the Db2 HPU interactive component” on page 26</td>
</tr>
<tr>
<td>3</td>
<td>Review the information about planning to Tools Customizer.</td>
<td>“Planning to use Tools Customizer” on page 27</td>
</tr>
<tr>
<td>4</td>
<td>Start and prepare Tools Customizer for use.</td>
<td>“Starting and preparing Tools Customizer for use” on page 31</td>
</tr>
<tr>
<td>5</td>
<td>Use Tools Customizer to customize Db2 HPU.</td>
<td>“Customizing Db2 HPU” on page 40</td>
</tr>
<tr>
<td>6</td>
<td>Optional: Integrate Db2 HPU into Db2 Tools Launchpad.</td>
<td>“Optional: Integrating Db2 HPU into Db2 Tools Launchpad” on page 61</td>
</tr>
<tr>
<td>7</td>
<td>Optional: Integrate Db2 HPU into Db2 Administration Tool.</td>
<td>“Optional: Integrating Db2 HPU into Db2 Administration Tool” on page 61</td>
</tr>
</tbody>
</table>

**Authorizing and enabling Db2 HPU**

This section gives you the procedure to authorize and enable Db2 HPU.

**About this task**

Before you can use Db2 HPU for the first time, you must do the following tasks:

1. Give APF authorization to the SINZLINK library
2. Make DSNAO64C available to Db2 HPU
3. Install the Db2 CLI
4. Enable the Db2 HPU batch utility

Enabling the Db2 HPU batch utility lets you unload Db2 data. You must create a batch utility job before you can run the Db2 HPU batch utility.

**Procedure**

1. APF authorize the SINZLINK library: To do so, run the SETPROG command in the following format:

   ```shell
   SETPROG APF,ADD,DSNAME=SINZLINK-data-set-name,VOLUME=volume-name
   ```

   The following example shows the SETPROG command in the correct format:
SETPROG APF, ADD, DSNAME=DMT.PTF510.SINZLINK, VOLUME=V01001

You can include this command in a MVS™ procedure, in a job, or in both.

2. Make DSNA064C available to Db2 HPU:

   The Db2 DSNA064C load module is required for Db2 HPU to run properly. Make sure this module is available to Db2 HPU. Consider adding the SDSNLOD2 Db2 library - that usually contains this module - to the Db2 HPU step steplib or make sure the DSNA064C load module is available to the Db2 HPU batch jobs in any other way. If DSNA064C is made available through an additional Db2 LOAD library (eg. SDSNLOD2), add it to the "Load library" parameter via the Tools Customizer relevant panel (i.e. CCQPDB2 - Db2 Parameters. See Defining Db2 parameters).

   If this task has not been done correctly and the DSNA064C module cannot be loaded by Db2 HPU, the unload job ends with completion code 8 and the following message is issued:

   INZI192E UNABLE TO LOAD THE DB2 LIBRARY (CEE3501S The module DSNA064C was not found).

3. Install the Db2 CLI:

   Db2 HPU uses the Db2 CLI to process SQL requests in order to get information from the Db2 catalog. The CLI must be customized for the Db2 sub-system and the plan DSNACLI must be bound. Otherwise the Db2 HPU jobs issue the INZ1193E message and terminate with a completion code of 8. For a correct use of Db2 HPU, make sure the following userids have the EXECUTE privilege on the DSNACLI plan:
   • the userid that submits the Db2 HPU batch job
   • the userid specified for User who issues SELECT by using dynamic SQL (VUM032/ACTLGUSR)
   • the userid specified for User who issues EXPLAIN PLAN statements (VUM035/EXPLNUSR)

4. Select the appropriate user switching method by setting the USER_SWITCH_METHOD subparameter of the "Db2 HPU additional features (VUU030/ULOPTNS)" parmlib parameter to the desired value.

5. Enable the Db2 HPU batch utility:

   To do so, use one of the following methods to enable the Db2 HPU batch utility.
   • Put the SINZLINK library in LINKLIST.
   • Specify the SINZLINK library in the STEPLIB statement of all jobs (be careful not to lose the APF authorizations with the concatenations).

   Related information
   MVS System Commands in the z/OS Library Center

Optional: Creating a TSO command for the Db2 HPU interactive component

You can make the Db2 HPU interactive component available as a TSO command.

About this task

With the Db2 HPU interactive component, you can generate the batch utility job to be used to invoke the Db2 HPU batch utility program.

You can run the batch utility program and the Db2 HPU interactive component simultaneously.

Procedure

Use one of the following methods to create a TSO command:
   • Concatenate the SINZCLST library to the SYSPROC concatenation.
   • Copy the INZHPU procedure into a library in the SYSPROC concatenation.
What to do next
Invoke the Db2 HPU interactive component by issuing the following command: TSO INZHPU.

Related tasks
Starting the Db2 HPU interactive component in stand-alone mode
You can start Db2 HPU interactive component in stand-alone mode without using the Db2 Administration Tool or the Db2 Tools Launchpad.

Planning to use Tools Customizer
This topic provides steps to take before using Tools Customizer to customize Db2 HPU.

Procedure
1. Become familiar with Tools Customizer.
   “Tools Customizer terminology and data sets” on page 27.
2. To complete customization, you might need to use the information in the Customization reference. This reference section provides additional information about the product information required to customize Db2 HPU, details of the tasks, steps, and parameters that are displayed on the Product Parameters panel in Tools Customizer, and information about the jobs that will be generated.

Tools Customizer terminology and data sets
Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology
Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Products and components
How an IBM Tool is packaged determines whether it is referred to as a product or as a component in the Tools Customizer documentation and interface. An IBM Tool that is ordered as a stand-alone entity (that is, not as part of a solution pack) is referred to as a product. An IBM Tool that is part of a solution pack is referred to as a component. Some IBM Tools are available in both formats; therefore, the same IBM Tool can be referred to as a product or as a component depending on how it is packaged.

Db2 entry
You can customize Db2 HPU on one or more Db2 entries. A Db2 entry can be any of the following items:

Db2 subsystem
A distinct instance of a relational database management system (RDBMS) that is not part of a data sharing group. An example of a Db2 subsystem name is DB01.

Db2 group attach name
The name that is used by the TSO/batch attachment, the call attachment facility (CAF), DL/I batch, utilities, and the Resource Recovery Services attachment facility (RRSAF) as a generic attachment name. An example of a group attach name is DSG1.

Db2 data sharing member
A Db2 subsystem that is assigned by the cross-system coupling facility (XCF) to a data sharing group. An example of a Db2 data sharing member name is DB02.

Tools Customizer maintains the following lists of Db2 entries:

Associated list
The list of Db2 entries that are associated with Db2 HPU. If the product to be customized requires Db2 entries, you can customize Db2 HPU only on Db2 entries that are in the associated list. When
you customize Db2 HPU, this list is displayed in the DB2® Entries, Associations, and Parameter Status section of the Customizer Workplace panel.

You can add and copy Db2 entries to the associated list. When you add or copy Db2 entries to the associated list, the entries are associated with Db2 HPU.

Main list

The list of all Db2 entries that are defined but are not associated with Db2 HPU. Tools Customizer obtains information about these Db2 entries either from entries that were created manually or from the customizations of other products that were discovered. If you remove a Db2 entry from the associated list, the Db2 entry is added to the main list. When you create a new Db2 entry, it is added to the main list, and when you associate the new entry with Db2 HPU, it is removed from the main list and added to the associated list. The main list is displayed on the Associate a DB2 Entry for Product panel.

If the associated list does not have the Db2 entries on which you want to customize Db2 HPU, you can associate existing entries from the main list to the associated list.

You can create new Db2 entries and copy existing entries to the main list.

High-level qualifier

The high-level qualifier is considered to be all of the qualifiers except the lowest level qualifier. A high-level qualifier includes a mid-level qualifier.

Product parameters

Parameters that are specific to Db2 HPU. These parameters are defined by Db2 HPU and are stored in a data member that is defined by Db2 HPU.

LPAR parameters

Parameters on the local LPAR that are required to customize Db2 HPU. These parameters are defined by Tools Customizer and are stored in an LPAR parameter data member.

Db2 parameters

Parameters for a Db2 entry. These parameters are defined by Tools Customizer and are stored in a Db2 parameter data member.

Status type

Product, LPAR, and Db2 entry status type

After you specify the product that you want to customize, the product, the LPAR, and the Db2 entries have a status. The status is partly based on whether required parameters are defined. For some products, LPAR parameters or Db2 parameters might not be required. In these cases, the status is Not Required.

To customize Db2 HPU, all of the required parameters must be defined.

If required parameters for the product parameters, LPAR parameters, or Db2 parameters are not defined, the status of the parameters is Incomplete. Define values for parameters by manually editing them or by generating the customization jobs and specifying values for all of the required parameters that are displayed on the panels.

When values for all of the required parameters are defined, the status is Ready to Customize. Customization jobs can be generated only when all of the required parameters are defined and the status is Ready to Customize or Customized for the product parameters, LPAR parameters, and Db2 parameters for the Db2 entries on which Db2 HPU will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.
Table 2. Status types for the product, the LPAR, and the Db2 entries

<table>
<thead>
<tr>
<th>Status</th>
<th>Product</th>
<th>LPAR</th>
<th>Db2 entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>The required product parameters are not defined.</td>
<td>The required parameters are not defined.</td>
<td>The required parameters are not defined.</td>
</tr>
<tr>
<td>Discovered</td>
<td>The product parameter definitions were discovered by using the product Discover EXEC.</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ready to Customize</td>
<td>The required product, is Ready to Customize or Customized for the LPAR and at least one associated Db2 entry. You can generate the customization jobs.</td>
<td>The required LPAR parameters are defined or LPAR parameters are not required.</td>
<td>The required Db2 parameters are defined or Db2 parameters are not required.</td>
</tr>
<tr>
<td>Verify Values</td>
<td>The required product or component parameter values are defined but they either have not been verified or verification is not enabled on the Product or Component Parameters panel.</td>
<td>The required LPAR parameter values are defined, but they either have not been verified or verification is not enabled on the LPAR Parameters panel.</td>
<td>The required Db2 parameter values are defined, but they either have not been verified or verification is not enabled on the Db2 Parameters panel.</td>
</tr>
<tr>
<td>Customized</td>
<td>The jobs are customized on the local LPAR.</td>
<td>The jobs are customized for the product or for all of the associated Db2 entries on the local LPAR.</td>
<td>The jobs are customized for the Db2 entry.</td>
</tr>
<tr>
<td>Errors in Customization</td>
<td>N/A</td>
<td>N/A</td>
<td>Errors occurred while the customization jobs were being generated.</td>
</tr>
<tr>
<td>Not Required</td>
<td>N/A</td>
<td>LPAR parameters are not required.</td>
<td>Db2 parameters are not required.</td>
</tr>
</tbody>
</table>

**Related tasks**

**Creating and associating Db2 entries**
You can create new Db2 entries and associate them with Db2 HPU.

**Copying Db2 entries**
You can copy associated and not associated Db2 entries to other Db2 entries or to new Db2 entries.

**Removing Db2 entries**
You can remove Db2 entries from the associated list.

**Data sets that Tools Customizer uses during customization**
Tools Customizer uses several unique data sets during the customization process. Familiarize yourself with these data sets before you begin to use Tools Customizer.

Several different data sets are required to customize Db2 HPU with Tools Customizer. These data sets are supplied by Db2 HPU, supplied by Tools Customizer, or allocated by Tools Customizer.

Db2 HPU provides the following data sets:
Metadata library
Contains the metadata for the product to be customized. Tools Customizer uses the metadata to determine which tasks, steps, and parameters to display on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel. This data set also contains the templates that Tools Customizer uses to generate the customization jobs.

The metadata library naming convention is high_level_qualifier.SINZDENU, where high_level_qualifier is all of the segments of the data set name except the lowest-level qualifier.

You specify the metadata library on the Specify the Metadata Library panel. READ access to this data set is required.

Discover EXEC library
Contains the Db2 HPU Discover EXEC. When you customize Db2 HPU, you can use the Discover EXEC to automatically retrieve and store product information, such as parameter values from an already customized product. Tools Customizer saves the discovered information in the data store.

The default name of the data set is the high-level qualifier for the metadata library plus a lowest-level qualifier. For Db2 HPU, the lowest-level qualifier is SINZDENU. You can change the default value on the Discover Customized Product Information panel. EXECUTE access to this data set is required.

Tools Customizer provides the following data sets:

Tools Customizer metadata library
Contains the metadata for the Db2 and LPAR parameters that are required to customize Db2 HPU. Tools Customizer uses the metadata to determine which parameters to display on the DB2 Parameters panel and the LPAR Parameters panel. In addition, Tools Customizer uses information in the metadata library to determine whether additional Db2 and LPAR parameters need to be displayed on these panels. As you customize different products, different Db2 and LPAR parameters might need to be defined.

The default name of the data set is DB2TOOL.CCQ110.SCCQDENU. You can change the default value on the Tools Customizer Settings panel. READ access to this data set is required.

Tools Customizer table library
Stores information about jobs that are customized. Job information that is stored includes a description of the job, its member name and template name, the SSID, group attach name, and when the job was generated.

The default name of the data set is DB2TOOL.CCQ110.SCCQTENU. WRITE access to this data set is required.

Tools Customizer requires that the following data sets exist during the customization process. If the data sets do not exist, Tools Customizer automatically allocates them.

Discover output data set
Contains the output that is generated when you run the Db2 HPU Discover EXEC. The Db2 HPU Discover EXEC retrieves the metadata and values for the parameters from a previous customization of Db2 HPU.

The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the Tools Customizer Settings panel or the Discover Customized Product Information panel. WRITE access to this data set is required.

Data store data set
Contains product, LPAR, and Db2 parameter values, and Db2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, Db2 subsystems or data sharing groups, and LPAR when you customize products on the local LPAR.

The default name of the data set is DB2TOOL.CCQ110.DASTOR. You can change the default value on the Tools Customizer Settings panel. WRITE access to this data set is required.

Customization library
Contains the customization jobs that Tools Customizer generates for Db2 HPU.
Tools Customizer checks whether a customization library name was specified for more than one instance of the same version of the same product. If the same customization library name is specified for more than one product of the same version, the CCQD123E message is issued to prevent you from overwriting previously generated customization jobs. Ensure that you specify unique qualifier for the customization library for each instance of the product.

To customize Db2 HPU, submit the members of the data set in the order in which they are displayed on the Finish Product Customization panel.

The data set naming convention is \textit{hlq}.\$\textit{LPAR\_name}\$.\textit{xyzvrm}, where:

- \textit{hlq} is the value of the \textbf{Customization library qualifier} field on the \textbf{Tools Customizer Settings} panel (CCQPSET)
- \textit{LPAR\_name} is the four-character LPAR name
- \textit{xyzvrm} is the three-letter product identifier with the version, release, and modification level

For example, the data set name might be DB2TOOL.PRODUCT.CUST.$MVS1$.XYZ410.

WRITE access to this data set is required.

Tools Customizer allocates the data sets for the discover output, the data store, and the customization library with the attributes that are shown in the following table:

\begin{table}[h]
\centering
\begin{tabular}{|l|c|c|c|c|l|}
\hline
Data set & Organization & Record format & Record length & Block size & Data set name type \\
\hline
Discover output data set & PO & Variable block & 16383 & 32760 & LIBRARY \\
\hline
Data store data set & PO & Variable block & 16383 & 32760 & LIBRARY \\
\hline
Product customization library & PO & Fixed block & 80 & 32720 & LIBRARY \\
\hline
\end{tabular}
\caption{Data set attributes for allocating the Discover output, data store, and customization library data sets}
\end{table}

\textbf{Restrictions:}

- Multiple users cannot simultaneously share the discover output data set, data store data set, Tools Customizer metadata library, and metadata library.

**Starting and preparing Tools Customizer for use**

Use the provided REXX EXEC to start Tools Customizer. The first time that you use Tools Customizer, you must modify the settings that Tools Customizer uses to customize Db2 HPU.

**Best Practice: SMP/E and runtime libraries maintenance strategy for Tools Customizer**

Tools Customizer creates relationships between the values for the Product Parameters, LPAR Parameters, and Db2 Subsystem Parameters for each Tools Customizer enabled product. Determining the correct maintenance strategy for your Tools Customizer runtime libraries, after SMP/E processing, can reduce problems working with Tools Customizer and the enabled products through their life cycles.

Tools Customizer has very specific requirements for data set names:

- Only one DATASTOR data set exists per LPAR
- The product metadata library data set names do not change during the life of that release of the Tools Customizer enabled product.
The DATASTOR data set is the repository for all the information that Tools Customizer requires to generate customization JCL for enabled products.

When you update and save the Tools Customizer Settings panel (CCQPSET), as described in “Modifying Tools Customizer user settings” on page 36, the name of the DATASTOR data set is saved in the ISPF profile. This allows Tools Customizer to know the active DATASTOR data set when the TSO user id logs in and starts the Tools Customizer EXEC.

Maintenance scenarios

IBM expects maintenance to be applied to libraries which are then used by Tools Customizer. In practice, different customer shops distribute SMP/E APPLY maintenance in different ways.

The following scenarios explains some considerations and alternatives for determining your maintenance strategy. The one overriding objective is to preserve and maintain the same data set names for the Tools Customizer instance.

Apply SMP/E maintenance to the same data sets (using the SMP/E APPLY command):

If you apply SMP/E maintenance (using the SMP/E APPLY command) using the same data set names with each maintenance cycle, you can either use these target libraries as your Tools Customizer runtime libraries or you can copy the SMP/E target data sets to the runtime libraries that are used by Tools Customizer to customize enabled products.

Tools Customizer assumes that if the product metadata library (*DENU) has the same name, this metadata library is for the same release of the enabled product. For example, assume that you customize Db2 Log Analysis Tool v3.3 and name the metadata library SYS2.DB2TOOL.SALADENU, with no indication of the version or release. You then upgrade to Db2 Log Analysis Tool v3.4 and employ the same naming convention, SYS2.DB2TOOL.SALADENU. Tools Customizer will assume that you are continuing to work with Db2 Log Analysis Tool v3.3 and will report v3.3 on panels and continue to use the same v3.3 Customization Library data set.

A more sustainable approach to naming the data sets is to include a product version, release identifier, or other distinguishing qualifier in the name of the metadata library, so that Tools Customizer can determine the new product release when you upgrade. For example, using metadata library names that include a product version, release identifier, or other distinguishing qualifier, similar to the following, can make product maintenance and upgrades easier:

- DB2TOOL.R330.SALADENU for Db2 Log Analysis Tool v3.3
- DB2TOOL.R340.SALADENU for Db2 Log Analysis Tool v3.4

Apply SMP/E maintenance to new data sets (using the SMP/E APPLY command):

If you apply SMP/E maintenance (using the SMP/E APPLY command) to new data sets rather than to the same data sets, the next time you open the product metadata library, Tools Customizer will return a data set error that indicates that the library name is being used by another product or component.

For example, assume that you name the Db2 High Performance Unload for z/OS target metadata libraries to reflect the date of an upgrade or to reflect a specific RSU, as follows:

- To reflect a specific upgrade date (August 2014):
  - DB2TOOL.PTF420.SINZDBRM.D201408
  - DB2TOOL.PTF420.SINZDENU.D201408
  - DB2TOOL.PTF420.SINZLOAD.D201408
- To reflect a specific RSU (RSU 1406):
  - DB2TOOL.PTF420.SINZDBRM.RSU1406
  - DB2TOOL.PTF420.SINZDENU.RSU1406
  - DB2TOOL.PTF420.SINZLOAD.RSU1406

Using either of these naming conventions, the next time you start the Tools Customizer EXEC, it will return a data set error.
To handle this type of SMP/E maintenance processing, you can do either of the following:

- Define aliases (using ALIAS control statements) to reference the appropriate libraries for Tools Customizer processing.
- Copy the SMP/E libraries to a set of runtime libraries that are specifically for Tools Customizer processing.

**Define aliases (using ALIAS control statements) to reference the appropriate libraries for Tools Customizer processing:**

Defining aliases (using ALIAS control statements) for the SMP/E created new product data set names is likely the best strategy when planning for Tools Customizer.

If you are setting up Tools Customizer for the first time, consider specifying the product library data set names with an indicator that these data sets will be used for Tools Customizer processing. For example, use data set names similar to the following names:

- TCZ.PTF420.SINZDBRM
- TCZ.PTF420.SINZDENU
- TCZ.PTF420.SINZLOAD

After applying maintenance using SMP/E, which creates new product library data sets, you should define aliases (using ALIAS control statements) for the new data set names to the data set names that Tools Customizer originally processed. For example, the following maintenance data sets have aliases defined to the original data sets:

- DB2TOOL.PTF420.SINZDBRM.RSU1406 --> TCZ.PTF420.SINZDBRM
- DB2TOOL.PTF420.SINZDENU.RSU1406 --> TCZ.PTF420.SINZDENU
- DB2TOOL.PTF420.SINZLOAD.RSU1406 --> TCZ.PTF420.SINZLOAD

You will need to define an alias (using ALIAS control statements) to each of the following IBM-distributed Tools Customizer data sets:

- SCCQDENU
- SCCQEXEC
- SCCQLOAD
- SCCQMENU
- SCCQPPN
- SCCQSAMP
- SCCQTENU

If you have already set up Tools Customizer and customized the product, you will have to define aliases (using ALIAS control statements) for the newly created data set names to the data sets that were specified when the product was originally customized using Tools Customizer.

After defining the aliases, you should be able to run Tools Customizer successfully.

**Note:** Only define aliases for IBM-distributed SMP/E libraries.

**Restriction:** Do not define an alias for any Tools Customizer created data sets, like the following three data sets on the Tools Customizer Settings panel (CCQPSET):

- Customization library qualifier
- Discover output data set
- Data store data set

Do not define an alias for any data sets that you create on behalf of a product, for example the Db2 High Performance Unload for z/OS parameter library (INFPLIB).
Copy the SMP/E libraries to a set of runtime libraries that are specifically for Tools Customizer processing:

If you are setting up Tools Customizer for the first time, consider specifying the product library data set names with an indicator that these data sets will be used for Tools Customizer processing. For example, use data set names similar to the following names:

- TCZ.R420.SINZDBRM
- TCZ.R420.SINZDENU
- TCZ.R420.SINZLOAD

After applying maintenance using SMP/E, which creates new product library data sets, you should copy the new data sets to the data sets that Tools Customizer originally processed. For example, copy the following maintenance data sets to the original data sets:

- DB2T00L.PTF420.SINZDBRM.RSU1406 --> TCZ.R420.SINZDBRM
- DB2T00L.PTF420.SINZDENU.RSU1406 --> TCZ.R420.SINZDENU
- DB2T00L.PTF420.SINZLOAD.RSU1406 --> TCZ.R420.SINZLOAD

You will need to copy each of the following IBM-distributed Tools Customizer data sets:

- SCCQDENU
- SCCQEXEC
- SCCQLOAD
- SCCQMENU
- SCCQPENU
- SCCQSAMP
- SCCQTENU

If you have already set up Tools Customizer and customized the product, you will have to copy the newly created data sets to the data sets that were specified when the product was originally customized using Tools Customizer.

After copying the SMP/E data sets to the Tools Customizer instance libraries, you should be able to run Tools Customizer successfully.

**Note:** Only copy IBM-distributed SMP/E libraries.

**Restriction:** Do not copy any Tools Customizer created data sets, like the following three data sets on the Tools Customizer Settings panel (CCQPSET):

- Customization library qualifier
- Discover output data set
- Data store data set

Do not copy any data sets that you create on behalf of a product, for example the Db2 High Performance Unload for z/OS parameter library (INFPLIB).

**Related tasks**

*Modifying Tools Customizer user settings*

Before you can customize a product or a component with Tools Customizer, you must review the settings that Tools Customizer uses.

**Related information**

*The SMP/E APPLY command*

*Alias processing: SMP/E for z/OS Commands*
Starting Tools Customizer

Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.

Before you begin
Tools Customizer must be SMP/E installed. You must know the high-level qualifier of where the Tools Customizer libraries reside. The high-level qualifier is considered to be all the segments of the data set name except the lowest-level qualifier, which is SCCQEXEC.

Attention: Ensure that Tools Customizer load libraries are not APF authorized. APF authorizing Tools Customizer libraries results in an abend.

About this task
To run the REXX EXEC, you must either change the placeholder in the EXEC for the high-level qualifier of the Tools Customizer EXEC library or pass the high-level qualifier as a parameter when you run the EXEC. The REXX EXEC is in the CCQTCZ member of the EXEC library.

Procedure
1. Optional: Change the placeholder for the high-level qualifier in the REXX EXEC:
   a) Find the EXEC library data set for Tools Customizer. The name of the data set is high_level_qualifier.SCCQEXEC.
   b) Edit data set member CCQTCZ and replace the <TCZ HLQ> string with the high-level qualifier of the EXEC library data set.
      For example, if the name of the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC, replace <TCZ HLQ> with CCQTCZ.USABSAND.
      You have to change the placeholder for the high-level qualifier only once. When you run the REXX EXEC, you do not have to pass the high-level qualifier as a parameter.

2. Run the REXX EXEC (CCQTCZ):
   a) From the ISPF Primary Option Menu, select option 6.
      The ISPF Command Shell panel is displayed.
   b) Specify the EX command to run the REXX EXEC.
      For example, if the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC and you changed the placeholder for the high-level qualifier in the REXX EXEC, specify:
      ```
      EX 'CCQTCZ.USABSAND.SCCQEXEC(Ccqtcz)'
      ```
      If you did not change the placeholder for the high-level qualifier in the REXX EXEC, specify:
      ```
      EX 'CCQTCZ.USABSAND.SCCQEXEC(Ccqtcz) ', 'CCQTCZ.USABSAND'
      ```
      You can also specify a trace data set name and a user profile when you run the REXX EXEC.
      - The default trace data set name is USERID.CCQ.TRACE. To specify a different trace data set name, append the trace data set name to the command. For example, to specify a trace data set name of CCQTCZ.MYTRACE, enter:
        ```
        EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ) ', 'CCQTCZ.USABSAND, CCQTCZ.MYTRACE'
        ```
      - To specify a user profile other than your own, append the user profile name to the command. For example, to specify a user profile of SHRPROF, enter:
        ```
        CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ) ', 'CCQTCZ.USABSAND, CCQTCZ.SHRPROF'
        ```
      Tools Customizer will use the settings from the specified profile. This profile will be updated when you exit Tools Customizer, but your own profile will remain unchanged.
Results
The IBM Customizer Tools for z/OS main menu panel is displayed.

What to do next
If you are running Tools Customizer for the first time, you must modify the Tools Customizer user settings. If you have already set the Tools Customizer user settings, either customize or recustomize Db2 HPU.

Modifying Tools Customizer user settings
Before you can customize Db2 HPU with Tools Customizer, you must review the settings that Tools Customizer uses. You might have to change the default values to suit your environment. In most cases, you can change the Tools Customizer values at any time. For example, after you have customized Db2 HPU and are customizing a different product or solution pack, you might have to change the settings.

Procedure
1. On the IBM Tools Customizer for z/OS main panel (CCQPHME), specify option 0, User settings for Tools Customizer.
   The Tools Customizer Settings panel (CCQPSET) is displayed, as shown in the following figure:

   ![Figure 1. The Tools Customizer Settings panel (CCQPSET)](image)

   Note: An asterisk next to a field indicates that the field is required.

2. Review the values for the following required fields. Use the default value or specify your own value.
   You must have appropriate READ and WRITE access to the data sets that are specified.

   **Customization library qualifier**
   The high-level qualifier that is used as the prefix for the customization library. The customization library is a data set in which the generated jobs to customize Db2 HPU are stored. WRITE access to this qualifier is required.

   For each product to be customized, the first value that is specified for the qualifier is always used, even if you change it after you have generated the customization jobs. For example, if you customize a product and then specify a new qualifier for recustomization, although the new qualifier is saved and displayed, the original value is used.

   To maintain multiple instances of Tools Customizer, specify a unique customization library qualifier for each instance of Tools Customizer.

   Data set names that exceed 42 characters must be enclosed in single quotation marks (').
Volume serial
The volume name in which the customization library will reside. If you don’t specify a volume name, it will be assigned by the system.

Use DB2 group attach
Determines the value that is used in the CONNECT statements in the generated customization jobs. Specify YES for data sharing environments, which causes the group attach name to be used. Specifying NO, in most cases, causes the SSID to be used in the Db2 CONNECT statement.

Important: This field has no effect when you are customizing a product on a Db2 subsystem that is not a member of a data sharing group. In this case, the Db2 subsystem ID (SSID) is always used in the CONNECT statements in the generated customization jobs.

When you are customizing a product on a Db2 subsystem that is a member of a data sharing group, how the Db2 subsystem is defined and the value of the Use DB2 group attach field determines the value that is used in the CONNECT statements in the generated jobs. The following table shows whether the SSID or the group attach name is used:

<table>
<thead>
<tr>
<th>Db2 subsystem definition</th>
<th>Value of the Use DB2 group attach field</th>
<th>Value that is used in the CONNECT statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Db2 subsystem is defined with an SSID.</td>
<td>Yes</td>
<td>Group attach name</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>SSID&lt;sup&gt;1&lt;/sup&gt;</td>
</tr>
<tr>
<td>The Db2 subsystem is not defined with an SSID.</td>
<td>Yes or No</td>
<td>Group attach name</td>
</tr>
</tbody>
</table>

Note 1: If you generate jobs for multiple Db2 subsystems that are defined with an SSID and belong to the same data sharing group, the SSID of the first Db2 subsystem that is selected is used.

For example, assume that on the Customizer Workplace panel, you generated jobs for the following Db2 subsystems:

- DB2C, which is a stand-alone Db2 subsystem
- DB2A, which is a Db2 subsystem that is a member of data sharing group DSG1
- A Db2 subsystem that was not defined with an SSID that is a member of data sharing group DSGA

The following figure shows how these Db2 entries might be listed on the Customizer Workplace panel:

<table>
<thead>
<tr>
<th>Associated Db2 Entries and Parameter Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line commands: G - Generate jobs E - Edit B - Browse C - Copy R - Remove</td>
</tr>
<tr>
<td>Cmd SSID GpAttach Lvl Mode User ID Date Status Message</td>
</tr>
<tr>
<td>DB2C -- 121 NFM SYSADM 2017/11/09 Ready to Customize</td>
</tr>
<tr>
<td>DB2A DSG1 121 NFM SYSADM 2017/11/09 Ready to Customize</td>
</tr>
<tr>
<td>-- DSGA 121 NFM SYSADM 2017/11/09 Ready to Customize</td>
</tr>
</tbody>
</table>

The following table shows which values are used in the CONNECT statements in the generated jobs, based on the value of the Use DB2 group attach field.
Tools Customizer metadata library
The name of the data set that contains the metadata that is used to display the Db2 and LPAR parameters. The parameters that are displayed on the LPAR Parameters panel and the DB2 Parameters panel depend on the parameters that you define and the tasks and steps that you select on the Product Parameters panel for the product that you are customizing. For example, the Db2 parameters that are required, based on the selected tasks and steps, are displayed on the DB2 Parameters panel, and you can edit them. If they are not required, they are not displayed. Read access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (`').

Discover output data set
The name of the data set in which the output from the Db2 HPU Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product, LPAR, and Db2 parameters from a previously customized product. Write access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (`').

Volume serial
The volume name in which the discover output data set will reside. If you don't specify a volume name, it will be assigned by the system.

Data store data set
The name of the data set where Tools Customizer stores information about product, LPAR, and Db2 parameter values. Information about which products are associated with which Db2 entries (Db2 subsystems, Db2 group attach names, and Db2 data sharing members) is also stored in this data set. Data set names that exceed 42 characters must be enclosed in single quotation marks (`'). The specified data store data set can be used with only one invocation of Tools Customizer at a time. Data set names that exceed 42 characters must be enclosed in single quotation marks (`').

Volume serial
The volume name in which the data store data set will reside. If you don't specify a volume name, it will be assigned by the system.

User job card settings for customization jobs
The job card information to be inserted into the generated jobs for customizing a product. The default value is the job statement information from the ISPF Batch Selection panel.

The first line of the job card automatically begins with the following information:

```
//          JOB
```

where characters 3 - 10 are reserved by Tools Customizer for the job name and includes a blank space after JOB. This name cannot be edited. Information that you specify on the first line of the job card cannot exceed 57 characters. This character limit includes a continuation character. All other lines of the job card cannot exceed 72 characters.

3. Press End to save and exit.
If the Discover output data set and the data store data set that you specified do not exist, Tools Customizer creates them.

**Important:** If the ISPF sessions unexpectedly ends before you exit Tools Customizer, the fields on the **Tools Customizer Settings** panel (CCQPSET) will be repopulated with default values, and you will be required to review them or specify new values again.

**Results**
The values are saved, and the **IBM Tools Customizer for z/OS** main menu panel (CCQPHME) is displayed again.

**What to do next**
You are ready to customize or recustomize Db2 HPU or to change parameter settings.

**Related concepts**
Customizing Db2 HPU
Using Tools Customizer to customize Db2 HPU consists of identifying the product to customize; defining any required Db2 HPU, LPAR, and Db2 parameters; generating the customization jobs; and submitting the jobs.

**Changing display options**
You can choose which types of information to show on Tools Customizer panels. You can also copy your user profile to another data set so that it can be shared with other users.

**About this task**
By using the OPTIONS command, you can choose to show or hide the following information on Tools Customizer panels:

- The instructions on all panels
- The Product to Customize section on the **Customizer Workplace** panel (CCQPWRK)
- The Usage Notes section on the **Product Parameters** panel (CCQPPRD)

The OPTIONS command also allows you to copy your user profile to another data set so that it can be shared with other users. By sharing a copy of your profile, other uses can customize the same products that you initially customized or started to customize.

**Procedure**
1. On any Tools Customizer panel, issue the OPTIONS command.

   The **Miscellaneous Options** panel (CCQPOPT) is displayed, as shown in the following figure. By default, all panel display options are preselected with a slash (/), which means that they will be displayed.

   ![Panel Display Options panel (CCQPOPT)](Figure 2. The Panel Display Options panel (CCQPOPT))

   **Figure 2. The Panel Display Options panel (CCQPOPT)**
2. To hide the panel instructions, the Product to Customize section, or the Usage Notes section, remove the slash from the appropriate option or options.

3. To copy your user profile to another data set so that it can be shared with other users:
   a) Type a slash in the **Copy user profile to another data set** field and press Enter.
   b) Specify the fully qualified name of the data set into which you want to copy the current user profile.
      If the data set name exceeds 42 characters, enclose the name in quotation marks. ALTER or UPDATE authorization to this data set is required.
   c) Optionally specify a volume name in which the user profile data set will reside. If you don't specify a volume name, it will be assigned by the system.

4. Press Enter to save your changes.

### Sorting and filtering columns

You can sort data in Tools Customizer columns by up to two columns. You can also filter the data in columns to display only the data that matches the filter criteria that you specify.

**About this task**

Sorting and filtering is available only on the **Customizer Workplace** panel, the **Finish Product Customization** panel, the **Associate DB2 Entry for Product** panel, and the **Copy Associated DB2 Entry** panel.

**Procedure**

The following instructions describe how to sort and filter data in Tools Customizer columns:

- **To sort data in Tools Customizer columns**, issue the **SORT** command.

  On the **SORT** command, specify up to two column names followed by the sort order: **A** for ascending or **D** for descending. If you don't specify a sort order, the default sort order is used, which can change depending on the column type.

  For example, the following command sorts the column entries by SSID in ascending order, and then by GrpAttch in descending order within SSID.

  ```
  COMMAND ===> SORT SSID A GrpAttch D
  ```

  You cannot specify the **Cmd** column on the **SORT** command.

- **To filter data in Tools Customizer columns**, overwrite the asterisk (*) under the column names with the filtering arguments for those columns.

  For example, to filter SSIDs that start with DB, overwrite the *\ under the **SSID** column with **DB** or **DB***. When you press Enter, all the SSIDS that meet that criteria, such as DB01 and DB02, are displayed.

  A filter argument in the form **DB*** means that only the characters up to the asterisk are considered. When you specify an asterisk in the last nonblank position of the argument, asterisks embedded in the argument are treated as data.

### Customizing Db2 HPU

Using Tools Customizer to customize Db2 HPU consists of identifying the product to customize; defining any required Db2 HPU, LPAR, and Db2 parameters; generating the customization jobs; and submitting the jobs.

Customization roadmaps describe the steps that you must complete to customize Db2 HPU. Separate roadmaps are provided for the three most common types of customizations.

Use the following table to determine which roadmap corresponds to your environment.
### Table 6. Customization roadmaps

<table>
<thead>
<tr>
<th>Environment description</th>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>You do not have a customized version of Db2 HPU, and you need to customize it for the first time.</td>
<td>“Roadmap: Customizing Db2 HPU for the first time” on page 41</td>
</tr>
<tr>
<td>You have already customized a version of Db2 HPU, and you want to use the same parameter values to customize a different version.</td>
<td>“Roadmap: Customizing a new version of Db2 HPU from a previous customization” on page 42</td>
</tr>
</tbody>
</table>
| You have a customized version of Db2 HPU, but you want to change one or more parameter values. | “Roadmap: Recustomizing Db2 HPU” on page 43  

#### Roadmap: Customizing Db2 HPU for the first time

This roadmap lists and describes the steps that are required to customize Db2 HPU for the first time.

If you are customizing a previous version of Db2 HPU, see “Roadmap: Customizing a new version of Db2 HPU from a previous customization” on page 42.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- The LPAR ISPF libraries that are required to submit the jobs are known.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize Db2 HPU for the first time.

### Table 7. Steps for customizing Db2 HPU for the first time

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify the product metadata library for the product that you want to customize. The name of this library is hlq.SINZDENU.</td>
<td>“Specifying the metadata library for the product to customize” on page 44</td>
</tr>
<tr>
<td>2</td>
<td>Create new Db2 entries and associate them with Db2 HPU.</td>
<td>“Creating and associating Db2 entries” on page 47</td>
</tr>
<tr>
<td>3</td>
<td>Define the required parameters.</td>
<td>“Defining parameters” on page 48</td>
</tr>
<tr>
<td>4</td>
<td>Generate the customization jobs for the product or for the Db2 entries on which Db2 HPU is ready to be customized.</td>
<td>“Generating customization jobs” on page 54</td>
</tr>
<tr>
<td>5</td>
<td>Submit the generated customization jobs.</td>
<td>“Submitting customization jobs” on page 54</td>
</tr>
</tbody>
</table>

The following table lists some of the common administrative tasks that you might need to do during the customization process.

### Table 8. Administrative tasks

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse the different types of parameters.</td>
<td>“Browsing parameters” on page 56</td>
</tr>
<tr>
<td>Copy an existing Db2 entry to the list of Db2 entries on which Db2 HPU can be customized.</td>
<td>“Copying Db2 entries” on page 57</td>
</tr>
<tr>
<td>Remove one or more Db2 entries from the associated list.</td>
<td>“Removing Db2 entries” on page 58</td>
</tr>
</tbody>
</table>
Table 8. Administrative tasks (continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete one or more Db2 entries from the main list.</td>
<td>“Deleting Db2 entries” on page 58</td>
</tr>
<tr>
<td>Display a list of customization jobs that have been previously generated.</td>
<td>“Displaying customization jobs” on page 59</td>
</tr>
<tr>
<td>Maintain the customization jobs in the customization library.</td>
<td>“Maintaining customization jobs” on page 59</td>
</tr>
</tbody>
</table>

Roadmap: Customizing a new version of Db2 HPU from a previous customization

This roadmap lists and describes the steps for customizing a new version of Db2 HPU based on the existing customization values of a previous version of the same product.

Use this roadmap even if the previous version of Db2 HPU was not customized by using Tools Customizer.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize a new version of Db2 HPU from a previous customization.

Table 9. Steps for customizing a new version of Db2 HPU from a previous customization

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify the product metadata library for the product that you want to customize. The name of this library is hlq.SINZDENU.</td>
<td>“Specifying the metadata library for the product to customize” on page 44</td>
</tr>
<tr>
<td>2</td>
<td>Use the Db2 HPU Discover EXEC to discover information about the version of Db2 HPU that you previously customized manually.</td>
<td>“Discovering Db2 HPU information automatically” on page 45</td>
</tr>
<tr>
<td>3</td>
<td>Define the required parameters.</td>
<td>“Defining parameters” on page 48</td>
</tr>
<tr>
<td>4</td>
<td>Generate the customization jobs for the product or for the Db2 entries on which Db2 HPU is ready to be customized.</td>
<td>“Generating customization jobs” on page 54</td>
</tr>
<tr>
<td>5</td>
<td>Submit the generated customization jobs.</td>
<td>“Submitting customization jobs” on page 54</td>
</tr>
</tbody>
</table>

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 10. Administrative tasks

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse the different types of parameters.</td>
<td>“Browsing parameters” on page 56</td>
</tr>
<tr>
<td>Copy an existing Db2 entry to the list of Db2 entries on which Db2 HPU can be customized.</td>
<td>“Copying Db2 entries” on page 57</td>
</tr>
<tr>
<td>Remove one or more Db2 entries from the associated list.</td>
<td>“Removing Db2 entries” on page 58</td>
</tr>
<tr>
<td>Delete one or more Db2 entries from the main list.</td>
<td>“Deleting Db2 entries” on page 58</td>
</tr>
</tbody>
</table>
Table 10. Administrative tasks (continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display a list of customization jobs that have been previously generated.</td>
<td>“Displaying customization jobs” on page 59</td>
</tr>
<tr>
<td>Maintain the customization jobs in the customization library.</td>
<td>“Maintaining customization jobs” on page 59</td>
</tr>
</tbody>
</table>

Roadmap: Recustomizing Db2 HPU

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for Db2 HPU after you have customized it for the first time.

The new customization jobs will replace the customization jobs that were previously generated and stored in the customization library. Part of the recustomization process includes selecting or deselecting optional tasks or steps, changing the definitions of parameters that have already been defined, or both. Use the method in this roadmap instead of deleting customization jobs from the customization library.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.

Complete the steps in the following table to recustomize Db2 HPU.

Table 11. Required steps for recustomizing Db2 HPU

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specify the product metadata library for the product that you want to recustomize. The name of this library is hlq.SINZDENU.</td>
<td>“Specifying the metadata library for the product to customize” on page 44</td>
</tr>
</tbody>
</table>
| 2    | Edit the specific tasks, steps, or parameters that need to be changed. | • “Defining Db2 HPU parameters” on page 49  
• “Defining LPAR parameters” on page 50  
• “Defining Db2 parameters” on page 52 |
| 3    | Generate the customization jobs for the product or for the Db2 entries on which Db2 HPU is ready to be customized. | “Generating customization jobs” on page 54 |
| 4    | Submit the new generated customization jobs. | “Submitting customization jobs” on page 54 |

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 12. Administrative tasks

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse the different types of parameters.</td>
<td>“Browsing parameters” on page 56</td>
</tr>
<tr>
<td>Copy an existing Db2 entry to the list of Db2 entries on which Db2 HPU can be customized.</td>
<td>“Copying Db2 entries” on page 57</td>
</tr>
<tr>
<td>Remove one or more Db2 entries from the associated list.</td>
<td>“Removing Db2 entries” on page 58</td>
</tr>
</tbody>
</table>
Table 12. Administrative tasks (continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete one or more Db2 entries from the main list.</td>
<td>“Deleting Db2 entries” on page 58</td>
</tr>
<tr>
<td>Display a list of customization jobs that have been previously generated.</td>
<td>“Displaying customization jobs” on page 59</td>
</tr>
<tr>
<td>Maintain the customization jobs in the customization library.</td>
<td>“Maintaining customization jobs” on page 59</td>
</tr>
</tbody>
</table>

Specifying the metadata library for the product to customize

You must specify a metadata library for the product that you want to customize.

About this task

The product metadata library contains the information that determines which tasks, steps, and parameters are required to customize Db2 HPU. This information controls what is displayed on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel.

After Db2 HPU has been SMP/E installed, the default name of the product metadata library is high_level_qualifier.SINZDENU, where high_level_qualifier is all of the segments of the data set name except the lowest-level qualifier.

Procedure

1. Specify option 1 on the Tools Customizer for z/OS panel.

   The Specify the Product or Pack Metadata Library panel is displayed. This panel contains a list of the product metadata libraries that you specified most recently. If you are using Tools Customizer for the first time, this list is empty, as shown in the following figure:

   ![Specify the Product or Pack Metadata Library panel](image)

   CCQPHLQ Specify the Product or Pack Metadata Library 15:12:22

   Type the name of the metadata library for the product or the pack in the Metadata library field, or select the library in the list of previous libraries and press Enter to populate the field. Press Enter to continue.

   The default name of the metadata library after the product or pack has been SMP/E installed is <hlq>.SxxxDENU, where <hlq> is the high-level qualifier for the product or the pack, and xxx is the three-character prefix for the product or the pack.

   **Product or pack metadata library** . XYZ.HINZ.SINZDENU

   **Name**     **Version**     **Metadata**
   =>
   =>
   =>

   Figure 3. The Specify the Metadata Library panel

2. Use one of the following methods to specify the product metadata library:

   - Type the name of a fully qualified partitioned data set (PDS) or an extended partitioned data set (PDSE) in the Product or pack metadata library field. Double quotation marks (") cannot be used around the name. Single quotation marks (') can be used but are not required. If you are customizing Db2 HPU for the first time, you must use this method.

   - Place the cursor in any column of the Recent Metadata Libraries list, and press Enter to populate Product or pack metadata library field. Press Enter again to select product or pack for customization.
Results
If you are customizing Db2 HPU for the first time, the Run Discover EXEC panel is displayed. Otherwise, the Customizer Workplace panel is displayed.

What to do next
• Complete the steps that correspond to your environment:

  Customizing Db2 HPU for the first time
  Do not run the Db2 HPU Discover EXEC. Press End. The Customizer Workplace panel is displayed.
  If your environment requires associated Db2 entries, ensure that they are created and associated. If your environment does not require associated Db2 entries, skip this step, and edit Db2 HPU parameters.

  Customizing Db2 HPU from a previous or current customization
  Press Enter to run the Db2 HPU Discover EXEC. The Discover Customized Product Information panel is displayed. Specify the required information for running the EXEC.

Discovering Db2 HPU information automatically
You can use the Db2 HPU Discover EXEC to discover information from a previous or current customization of Db2 HPU.

About this task
Tip: Using the Db2 HPU Discover EXEC to discover information from a previous or current customization saves time and reduces errors that can occur when parameters are specified manually.

Db2 HPU provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on Db2 HPU.

Parameter values that are discovered and parameter values that are specified manually are saved in the data store. If parameter values for the product that you want to customize exist in the data store, Tools Customizer issues a warning before existing values are replaced.

Procedure
1. On the Customizer Workplace panel, issue the DISCOVER command.
   If you chose to run the Db2 HPU Discover EXEC on the pop-up panel after you specified the product to customize, skip this step.

   Tip: You can run any Tools Customizer primary command by using either of the following methods:
   • Place the cursor on the name of the primary command, and press Enter.
   • Type the primary command name in the command line, and press Enter.

   The Discover Customized Product Information panel is displayed, as shown in the following figure:
For the product you are customizing, the Discover EXEC retrieves product information from an already customized product. Specify the required information. To save your information and run the Discover EXEC, issue the RUN command. To save your information and stay on this panel, issue the SAVE command. To verify the syntax of your information without saving it, press Enter. To save and exit, press End.

Commands: RUN  SAVE

Product to Customize

- Product metadata library : XYZ.HINZ.SINZDENU > LPAR...: ZPS1
- Product name . . . . . . : IBM DB2 High Performance U > Version . :

Discover EXEC for Extracting Information from an Already Customized Product

- Discover EXEC library . . . . XYZ.HINZ.SINZCLST
- Discover EXEC name . . . . : INZTDSC
- Discover output data set .. XYZ.HINZ.DISCOUT

Verbose mode for procedure . . . . . . . OFF (OFF,ON)

Figure 4. The Discover Customized Product Information panel

2. Either accept the default values for the following input fields that Tools Customizer generates, or replace the default values with your own values:

 **Discover EXEC library**

   The fully qualified data set name that contains the Db2 HPU Discover EXEC.

 **Discover EXEC name**

   The name of the Db2 HPU Discover EXEC.

 **Discover output data set**

   The fully qualified data set where output from the Db2 HPU Discover EXEC is stored.

3. Either accept or change the default values in the Information for Discover EXEC fields.

   These fields are generated by Db2 HPU. They show the information that is required to run the Db2 HPU Discover EXEC.

4. Issue the RUN command to run the Db2 HPU Discover EXEC.

   Alternatively, save your information without running the Db2 HPU Discover EXEC by issuing the SAVE command.

   If you issue the RUN command to run the Db2 HPU Discover EXEC, the parameter information is discovered for Db2 HPU, and the Customizer Workplace panel is displayed.

Results

The discovered parameter values for Db2 HPU replace any existing values.

**What to do next**

The next step depends on your environment:

- If Db2 entries were not discovered, or if you need to customize Db2 HPU on new Db2 entries, create and associate the entries.
- If Db2 entries were discovered and you want to customize Db2 HPU on only these entries, define the parameters.

**Related tasks**

Creating and associating Db2 entries

You can create new Db2 entries and associate them with Db2 HPU.

Defining parameters
To customize Db2 HPU, you must define Db2 HPU parameters, LPAR parameters, and Db2 parameters, if your customization requires Db2 entries.

**Creating and associating Db2 entries**

You can create new Db2 entries and associate them with Db2 HPU.

**About this task**

The list of associated Db2 entries is on the **Customizer Workplace** panel.

**Procedure**

1. Issue the **ASSOCIATE** command on the **Customizer Workplace** panel. The **Associate DB2 Entry for Product** panel is displayed, as shown in the following figure:

   ![Figure 5. The Associate DB2 Entry for Product panel](image)

   **Figure 5. The Associate DB2 Entry for Product panel**

2. Create Db2 entries.

   If you need to associate Db2 entries that are already in the main list, skip this step and go to step 3.
   a) Issue the **CREATE** command to create one Db2 entry, or issue **CREATE nn** to create multiple Db2 entries, where **nn** is the number of new entries to be created.

      ![Figure 6. The Create DB2 Entries panel](image)

      **Figure 6. The Create DB2 Entries panel**

   b) In the appropriate columns, specify a Db2 subsystem ID, Db2 group attach name, or Db2 data sharing member name for the Db2 entry that you want to create, and press Enter. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

      **Tips:**

      - To insert multiple Db2 entries, specify the **Inn** line command, where **nn** is the number of Db2 entries to be inserted.
      - You will define specific parameters for these new Db2 entries on the **DB2 Parameters** panel. This panel is displayed after you select these new Db2 entries and issue the line command to generate...
the jobs, after you issue the primary command to generate the jobs for all associated Db2 entries, or when you manually edit the Db2 parameters.

The **Associate DB2 Entry for Product** panel is displayed, and the new Db2 entry is displayed in the main list, as shown in the following figure:

![Figure 7. Associate DB2 Entry for Product panel](image)

3. **Associate Db2 entries.**
   a) Specify A against one or more Db2 entries in the main list, and press Enter to associate them with Db2 HPU.

**Results**
The **Customizer Workplace** panel is displayed with the associated Db2 entries displayed in the associated list.

**What to do next**
Define the parameters.

**Related concepts**
Tools Customizer terminology
Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

**Defining parameters**
To customize Db2 HPU, you must define Db2 HPU parameters, LPAR parameters, and Db2 parameters, if your customization requires Db2 entries.

**About this task**
You must define the Db2 HPU parameters first for the following reasons:

- If you ran the Db2 HPU Discover EXEC, you must review the values that were discovered.
- If you select optional tasks and steps on the **Product Parameters** panel that affect the Db2 entry on which you will customize Db2 HPU, additional parameters might be displayed on the **DB2 Parameters** panel.
- If other steps must be completed in a specific sequence, customization notes on the **Product Parameters** panel will display the correct sequence.
Defining Db2 HPU parameters
Db2 HPU parameters are specific to Db2 HPU.

About this task
If you ran the Db2 HPU Discover EXEC, you must review the parameters that were discovered.

Procedure
1. Specify E next to the **Product parameters** field on the **Customizer Workplace** panel, and press Enter.

   The **Product Parameters** panel is displayed, as shown in the following figure. If other steps must be completed in a specific sequence before you define the Db2 HPU parameters, a note labeled **Important** will display the correct sequence on this panel.

   **Figure 8. The **Product Parameters** panel**

   You can use the following primary commands on this panel:

   **SAVE**
   Saves the specified product or component parameter values.

   **VERIFY / VERIFYOFF**
   Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of product or component parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The product or component parameter status of Verify Values on the Customize Workplace panel indicates that the values have not been verified.

   Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the **DB2 Parameters** panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

   Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the Product or Component Parameters Values panel.

2. Select any required tasks and steps, and specify values for any parameters. After you select a task or step with a slash (/), put the cursor in the selected field and press Enter.
If tasks, steps, and parameters are required, they are preselected with a slash (/). Otherwise, they are not preselected.

All of the required parameters have default values, which you can either accept or change.

**Tips:**
- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
  - **Add** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add**, and press Enter. Use the displayed panel to add or delete additional values.
  - **List** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List**, and press F1 or the key that is mapped to Help.
  - **More** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More**, and press Enter.

3. Optional: Select other tasks and steps with a slash (/) and press Enter to activate the input fields. Either accept or change the default values that are displayed.

4. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the Product Parameters panel.

**Results**
The **Customizer Workplace** panel is displayed, and the status of the product parameters is Ready to Customize.

**What to do next**
If the status of other parameters on the **Customizer Workplace** panel is Incomplete, Verify Values, or Discovered, edit these parameters.

**Related tasks**
- Defining LPAR parameters
  LPAR parameters are parameters on the local LPAR that are required to customize Db2 HPU.
- Defining Db2 parameters
  Db2 parameters are parameters for a Db2 entry.

**Defining LPAR parameters**
LPAR parameters are parameters on the local LPAR that are required to customize Db2 HPU.

**Procedure**
1. Specify E next to the **LPAR parameters** field, and press Enter.
   The **LPAR Parameters** panel is displayed, as shown in the following figure:
Ensure that values are specified for the required LPAR parameters. Press End to save and exit.

Commands: SAVE  VERIFYOFF

More:

Other Parameters - common
* Unit name for batch work data sets . . . WORK

DB2 HPU Library Dsnames
* Load modules library ........................................... /SINZLOAD  HINZ510.CODEDROP.SINZLOAD
* APF load modules library ...................................... /SINZLINK  HINZ510.CODEDROP.SINZLINK
* Samples library .................................................. /SINZSAMP  HINZ510.CODEDROP.SINZSAMP
* PARM LIB library ................................................. INZPLIB/INFPLIB
* Product's DBRM library ....................................... /SINZDBRM  HINZ510.CODEDROP.SINZDBRM
* REXX EXEC library ............................................... /SINZCLST  HINZ510.CODEDROP.SINZCLST

Figure 9. The LPAR Parameters panel

You can use the following primary commands on this panel:

SAVE
Saves the specified product or component parameter values.

VERIFY / VERIFYOFF
Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of LPAR parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The LPAR parameter status of Verify Values on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the LPAR Parameters panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the LPAR Parameters Values panel.

2. Specify values for all required parameters that are displayed.

Many parameters have default values, which you can either accept or change.

Tips:
* In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
* For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
* The following elements apply to specific fields:
  - Add is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on Add, and press Enter. Use the displayed panel to add or delete additional values.
  - List is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on List, and press F1 or the key that is mapped to Help.
  - More is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on More, and press Enter.
The following LPAR parameters can contain 1 - 64 values:

- LPAR macro library
- Message library
- Panel library
- Skeleton library
- ISPF table input library
- ISPF user profile library
- File tailoring output library
- Link list library
- Command procedures library
- Macro library
- Link-edit library
- Load library
- Started task library name

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results
The **Customizer Workplace** panel is displayed, and the status of the LPAR parameters is Ready to Customize.

What to do next
If the status of other parameters on the **Customizer Workplace** panel is Incomplete, Verify Values, or Discovered, edit these parameters.

Related tasks
- Defining Db2 HPU parameters
  - Db2 HPU parameters are specific to Db2 HPU.
- Defining Db2 parameters
  - Db2 parameters are parameters for a Db2 entry.

**Defining Db2 parameters**
Db2 parameters are parameters for a Db2 entry.

About this task
If you did not run the Db2 HPU Discover EXEC, you must create and associate one or more Db2 entries before you can define the Db2 parameters. For more information, see “Creating and associating Db2 entries” on page 47.

Procedure
1. Specify E next to one or more Db2 entries in the associated list, which is in the Associated Db2 Entries and Parameter Status section on the **Customizer Workplace** panel, and press Enter.
   - The **DB2 Parameters** panel is displayed, as shown in the following figure:
Ensure that values are specified for the required DB2 parameters. Press End to save and exit.

Commands: SAVE
VERIFYOFF

DB2 subsystem ID: DZ7M
Group attach name: 

General DB2 Information - common
* Mode: NFM (CM, CM8, CM9, NFM)
* Level Number: 101 (810, 910, 101, 111, 121)
* Mixed Mode: NO (YES, NO)

DB2 Libraries - common
* Load Library: DSNA10.SDSNLOAD
* Run Library: DSNA10.RUNLIB.LOA
* Exit Library: DSNA10.SDSNEXIT

DB2 Utilities - common
SYSAFF for DB2 utilities: DSNTIA10
* Plan name for the DSNTIAD utility: DSNTIA10

Figure 10. The DB2 Parameters panel

You can use the following primary commands on this panel:

SAVE
Saves the specified product or component parameter values.

VERIFY / VERIFYOFF
Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of Db2 parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The Db2 parameter status of Verify Values on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the DB2 Parameters panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the DB2 Parameters Values panel.

2. Specify values for all parameters that are displayed.

Tips:
• In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
• For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
• The following elements apply to specific fields:
  – Add is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on Add, and press Enter. Use the displayed panel to add or delete additional values.
  – List is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on List, and press F1 or the key that is mapped to Help.
  – More is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on More, and press Enter.
Many parameters have default values, which you can either accept or change.

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

**Results**
The status of the Db2 entries that you selected on the **Customizer Workplace** panel is Ready to Customize.

**What to do next**
If the status of other parameters on the **Customizer Workplace** panel is Incomplete, Verify Values, or Discovered, edit these parameters.

**Related tasks**
- Defining Db2 HPU parameters
- Db2 HPU parameters are specific to Db2 HPU.
- Defining LPAR parameters
- LPAR parameters are parameters on the local LPAR that are required to customize Db2 HPU.

**Generating customization jobs**
To generate customization jobs for Db2 HPU and any associated Db2 entries, issue the `GENERATEALL` command, or select one or more Db2 entries on which to customize Db2 HPU.

**Procedure**
Generate the customization jobs by using one of the following methods.

- If you want to generate customization jobs at the product level and for any associated Db2 entries, issue the `GENERATEALL` command, and press Enter.
- If you want to generate customization jobs for specific Db2 entries, select the Db2 entries by specifying the `G` line command against them, and press Enter. The available Db2 entries are in the associated list in the Associated Db2 Entries and Parameter Status section.

**Important:** Regenerating customization jobs will replace any existing jobs, including jobs that you might have manually modified after they were generated.

**Results**
If the status is Incomplete or Discovered for Db2 HPU parameters, LPAR parameters, or Db2 parameters, Tools Customizer automatically starts an editing session for the types of parameters that are required. The session continues until the panel for each type of required parameter has been displayed.

**What to do next**
If an automatic editing session is started, accept the displayed parameter values or define values for the required types of parameters, select optional parameters, tasks, or steps for your environment, and save the parameter values. Otherwise, the customization jobs are generated, and you can submit them.

**Tip:** If the customization jobs are generated, but you are not ready to submit them, you can see them later by issuing the `JOBLIST` command on the **Customizer Workplace** panel. The `JOBLIST` command displays the **Finish Product Customization** panel, which you can use to submit the jobs.

**Submitting customization jobs**
Submit the customization jobs to customize Db2 HPU.

**Before you begin**
Ensure that the correct jobs are generated.
About this task

The following figure shows part of the Finish Product Customization panel. The table on this panel shows the customization jobs that are generated by Tools Customizer. They are grouped by job sequence number.

Figure 11. The Finish Product Customization panel

The member-naming conventions depend on whether the customization jobs are for Db2 entries, and LPAR, or the product.

The New field indicates if the job member is newly created/updated. It is either YES or NO. YES indicates the job member is newly created or updated, and it needs to be submitted for customization. NO indicates the job member is not newly created/updated, it does not need to be submitted for customization.

Customization jobs for Db2 entries

The members use the following naming convention:

\(<\text{job\_sequence\_number}>\text{job\_ID}<$DB2\_entry\_ID>\)

where

**job\_sequence\_number**

Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

**job\_ID**

Characters 4 - 7 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. Db2 HPU assigns the template name.
**DB2_entry_ID**

Two alphanumeric characters, AA - 99, that Tools Customizer assigns to a Db2 entry.

For example, the XYZBNDDB2_entry_ID_1 and XYZBNDDB2_entry_ID_2 jobs are generated from the XYZBNDGR template, and the XYZ4DB2_entry_ID_1 and XYZ4DB2_entry_ID_2 jobs are generated from the XYZ4 template. If the jobs are generated on two Db2 entries, the following member names are listed sequentially: A0BNDGAA, A0BNDGAB, A14AA, A14AB.

**Customization jobs for an LPAR or the product**

The members use the following naming convention:

```
<job_sequence_number><job_ID>
```

where

**job_sequence_number**

Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

**job_ID**

Characters 4 - 8 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. For example, for the XYZMAKE template, the job ID is MAKE. For the XYZM template, the job ID is M. Db2 HPU assigns the template name, and it is displayed in the Template column.

For example, the XYZBNDGR job is generated from the XYZBNDGR template, and the XYZ4 job is generated from the XYZ4 template. The following member names are listed sequentially: A0BNDGR, A14.

**Procedure**

1. Submit the generated customization jobs by following the process that you use in your environment or by using the following method:
   a) Specify B or E against a customization job or the product customization library, and press Enter.
      An ISPF browsing or editing session is started.
   b) Browse the customization job or each member in the library to ensure that the information is correct.
   c) Run the TSO SUBMIT command.

2. Press End.

**Results**

Db2 HPU is customized, and the **Customizer Workplace** panel is displayed. The status is Customized for the Db2 entries on which Db2 HPU was customized.

**What to do next**

You can generate more customization jobs for other Db2 entries, view a list of customization jobs that you previously generated, or recustomize Db2 HPU.

**Browsing parameters**

You can browse the product or component parameters, the LPAR parameters, and the Db2 parameters in read-only mode.

**Procedure**

1. On the **Customizer Workplace** panel, specify B next to the **Product parameters** field, the **LPAR parameters** field, or the Db2 entry that you want to browse, and press Enter.
   The panel that corresponds to your specification is displayed.
2. Press End to exit.

**Copying Db2 entries**

You can copy associated and not associated Db2 entries to other Db2 entries or to new Db2 entries.

**About this task**

Go to the step that applies to your environment:

- To copy an associated Db2 entry to another associated Db2 entry or to an entry that is not associated, go to step 1.
- To copy an associated Db2 entry to a new entry, go to step 2.
- To copy a Db2 entry that is not associated to a new entry, go to step 3.

**Procedure**

1. To copy an associated Db2 entry to another associated Db2 entry or to an entry that is not associated, complete the following steps:
   a) Specify **C** against a Db2 entry in the associated list of Db2 entries on the **Customizer Workplace** panel, and press Enter.
      The **Copy Associated DB2 Entry** panel is displayed.
   b) Select one or more Db2 entries to which information will be copied by specifying the / line command, and press Enter.
      The Associated column indicates whether the Db2 entry is associated.
      **Tip:** To copy information into all of the Db2 Entries in the list, issue the SELECTALL primary command, and press Enter.
      The **Copy DB2 Parameter Values** panel is displayed.
   c) Specify an option for copying common and product-specific Db2 parameter values.
      Common Db2 parameter values apply to all Db2 entries for all products that you have customized by using Tools Customizer. Product-specific Db2 parameter values apply only to the product that you are currently customizing.
      - To copy the common Db2 parameter values and the product-specific Db2 parameter values, specify option 1, and press Enter.
      - To copy only the product-specified Db2 parameter values, specify option 2, and press Enter.
      In some cases, the Db2 parameter values might contain the Db2 subsystem ID as an isolated qualifier in data set names. For example, in the DB01.DB01TEST.DB01.SANLLOAD, data set name, the DB01 subsystem ID is isolated in the first and third qualifiers but is not isolated in the second qualifier. When the Db2 subsystem ID is an isolated qualifier in data set names, the **Change DB2 Subsystem ID in DB2 Parameter Values** panel is displayed. Otherwise, the **Customizer Workplace** panel is displayed.
   d) If the **Change DB2 Subsystem ID in DB2 Parameter Values** panel is displayed, specify an option for changing the subsystem IDs. Otherwise, skip this step.
      - To change the subsystem ID in isolated qualifiers in data set names, specify option 1, and press Enter.
      - To use the same subsystem ID in all values, specify option 2, and press Enter.
      The **Customizer Workplace** panel is displayed with the copied associated entry in the list.

2. To copy an associated Db2 entry to a new entry, complete the following steps:
   a) Specify **C** against a Db2 entry in the associated list of Db2 entries on the **Customizer Workplace** panel, and press Enter.
      The **Copy Associated DB2 Entry** panel is displayed.
   b) Issue the CREATE command.
The **Create DB2 Entries** panel is displayed.

c) Specify the SSID, the group attach name, or both in the appropriate columns for each new Db2 entry, and press Enter.

**Tip:** To add rows for additional entries, specify the Inn line command, where **nn** is the number of entries to be created, and press Enter.

The **Copy Associated DB2 Entry** panel is displayed with the new entries in the list. The new entries are preselected.

d) Press Enter to complete the copy process.

The **Customizer Workplace** panel is displayed with the copied entries in the list.

3. To copy a Db2 entry that is not associated to a new entry, complete the following steps:

a) Issue the ASSOCIATE command on the **Customizer Workplace** panel.

The **Associate DB2 Entry for Product** panel is displayed.

b) Select one or more Db2 entries by specifying the / line command, and press Enter.

The **Copy a DB2 Entry** panel is displayed.

c) Specify the SSID, the group attach name, or both in the appropriate columns for the new Db2 entry, and press Enter.

The **Associate DB2 Entry for Product** panel is displayed with the copied entry in the list.

d) If you want to associate the copied entry, specify **A** against it, and press Enter.

The **Customizer Workplace** panel is displayed with the copied entries in the list.

**What to do next**

Edit any of the parameters or generate the jobs.

**Related concepts**

Tools Customizer terminology

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

### Removing Db2 entries

You can remove Db2 entries from the associated list.

**About this task**

When you remove Db2 entries from the associated list, any customization jobs for the entries are removed from the list of jobs on the **Finish Product Customization** panel, and they are deleted.

**Procedure**

On the **Customizer Workplace** panel, specify **R** next to one or more Db2 entries that you want to remove, and press Enter.

The selected Db2 entries are removed from the associated list and added to the main list on the **Associate DB2 Entry for Product** panel, and the customization jobs are deleted.

**Related concepts**

Tools Customizer terminology

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

### Deleting Db2 entries

You can delete Db2 entries from the main list.

**About this task**

When you delete Db2 entries from the main list, any associations and all customization jobs for products that are customized on the entries will be deleted.
Procedure

1. On the **Customizer Workplace** panel, issue the ASSOCIATE command. The **Associate DB2 Entry for Product** panel is displayed.

2. Specify D next to one or more Db2 entries that you want to delete, and press Enter.

   If the entry is associated with any products, the **Delete Associated DB2 Entry** panel for the first Db2 entry that you selected is displayed. Otherwise, the **Delete DB2 Entry** panel is displayed.

3. To delete the Db2 entries, press Enter.

   If the Db2 entries are associated with any products in the table on the **Delete Associated DB2 Entry** panel, any associations and all customization jobs for the products that are customized on it are deleted. Otherwise, only the Db2 entries are deleted. If you selected multiple Db2 entries to delete, the next Db2 entry that you selected is displayed on either the **Delete Associated DB2 Entry** panel or the **Delete DB2 Entry** panel. Otherwise, the **Associate DB2 Entry for Product** panel is displayed.

**What to do next**

If you selected multiple Db2 entries to delete, repeat step 3 until all selected entries are deleted. Then, continue the customization process.

**Displaying customization jobs**

You can view a list of the members that contain the customization jobs before or after you submit the jobs.

**About this task**

The customization jobs that you generate for one Db2 entry are also displayed when you customize Db2 HPU for another Db2 entry later.

**Procedure**

On the **Customizer Workplace** panel, issue the JOBLIST command.

The **Finish Product Customization** panel is displayed. This panel shows the list of jobs that you have previously generated. They are grouped by job sequence number. Use this panel to browse or edit the generated jobs before you submit them.

**Maintaining customization jobs**

Instead of deleting customization jobs outside of Tools Customizer, you can maintain the correct jobs for Db2 HPU by completing the steps for re-customization.

**About this task**

You cannot delete or rename customization jobs from the customization library by starting an ISPF browse or edit session from the **Finish Product Customization** panel. If you try to delete customization jobs by using this method, the CCQC034S message is issued. If you try to rename customization jobs, the CCQC035S message is issued.

If you delete or rename customization jobs from the customization library by using ISPF outside of Tools Customizer, Tools Customizer will not recognize that the jobs were deleted, and the **Finish Product Customization** panel will still display them. If you browse or edit jobs that were deleted from the library outside of Tools Customizer, the CCQC027S message is issued.

**Procedure**

To maintain the correct customization jobs in the customization library, complete the steps for re-customization.
Using Tools Customizer in a multiple-LPAR environment
Currently, Tools Customizer supports only the local LPAR; however, you can propagate customizations to additional LPARs by using either of two different methods.

About this task
In a multiple-LPAR environment, Tools Customizer identifies the LPAR to which you are logged on. Tools Customizer uses this LPAR name for several different parameter settings, one of which is the data store. When you use the data store during the customization of Db2 HPU that is on a different LPAR, Tools Customizer issues message CCQD586S, which indicates that the product has already been customized based on values from the data store on the first LPAR. This message is issued to prevent the data store from becoming corrupted.

This behavior occurs in the following conditions:
• Tools Customizer is installed on a DASD device that is shared by multiple LPARs.
• After a product is customized by using Tools Customizer, the data store is copied to another LPAR.

Procedure
To customize products running against a Db2 subsystem on an LPAR where Tools Customizer is not installed, consider using one of the following methods:

Install one instance of Tools Customizer on one LPAR
If you intend to reuse the customization values for all the instances of your products on all LPARs, use this method.

a. Associate all the Db2 entries in this one instance of Tools Customizer. The LPARs on which the Db2 subsystems reside do not matter.

b. Generate the customization jobs for each Db2 entry.

c. Copy the generated customization jobs to the LPAR to run against the specific Db2 entries. Some LPAR-specific edits might be required. You can make these edits in the customized jobs that you copied. Note that this situation is one of the few situations where you might need to make manual changes to the jobs that are customized by Tools Customizer.

Install one instance of Tools Customizer on each LPAR
If you do not want to reuse previous customization values and you want to start new customizations, use this method.

Important: This method will likely not be the preferred approach for most organizations because most organizations tend to use similar or identical customization values for each product instance on all LPARs.

Important: Tasks to perform when upgrading Db2 for z/OS but not Db2 HPU

About this task
After any upgrade affecting either the mode or the release of Db2, proceed as follows in order to make sure Db2 HPU will go on working correctly.

Procedure
1. Update the Db2 parameter according to the Db2 level (version or mode) of the upgraded subsystem(s).
2. Generate the customization jobs.
3. Submit the customization jobs associated to the INZBIND template.
4. Submit the customization jobs associated to the INZCHECK template.
Optional: Integrating Db2 HPU into Db2 Tools Launchpad

Optionally, you can integrate Db2 HPU into Db2 Tools Launchpad. Tools Customizer will create the necessary JCL, but you must manually complete some steps after you submit the customization job.

Before you begin
Before you complete these steps, ensure that the following prerequisites have been met:

- The Db2 Tools Launchpad is installed.
- Tools Customizer generated the job from the INZLAUNC template, and you submitted the job.

About this task
The Db2 Tools Launchpad is a centralized panel from which you can launch integrated Db2 Tools. After you integrate Db2 HPU, you can launch Db2 HPU from the Db2 Tools Launchpad.

Procedure
1. Run the CLIST INZADBI in SINZCLST.
   The CLIST uses the high-level qualifier that you specify for the Db2 Admin data sets and the name of the library that contains the ADBDMTI EXEC.
   The Db2 Tools Table – ADD An Entry panel is displayed, as shown in the following figure:

   ![Figure 12. Adding an entry to the Db2 Tools Launchpad panel](image)

   Command ===>
   Tool Name : HPU High Performance Unload
   Code      : HPU       (User-defined code, for shortcut tool identifier)
   Prog No.  : 5655-AA1  (IBM program product number or equivalent)
   Release   : 420       (Product release number)
   Group     : 2         (Tool category, as follows:
                           1 - Administration Tools
                           2 - Application Management Tools
                           3 - Performance Management Tools
                           4 - Recovery and Replication Management)
   Installed : Y         (Y - yes, N - no)
   Command   : SELECT CMD(INZHPU D LP )

2. Press Enter to confirm the new Db2 HPU command.

Results
When INZADBI completes successfully, a new line, HPU, is added to the Db2 Tools Launchpad.

Optional: Integrating Db2 HPU into Db2 Administration Tool

Optionally, you can integrate Db2 HPU into Db2 Administration Tool (Db2 Admin). Tools Customizer will create the necessary JCL, but you must manually complete some steps after you submit the customization job.

Before you begin
Before you complete these steps, ensure that the following prerequisites have been met:

- Db2 Admin is installed.
- Tools Customizer generated the job from the INZADTOO template, and you submitted the job.
About this task
Db2 Admin helps you manage Db2 environments efficiently and effectively. After you have integrated Db2 HPU, you can run Db2 HPU unload operations by using Db2 Admin.

Procedure
1. Follow the instructions that are described in INZDB21X in the SINZCLIST library.
   You can customize these instructions by changing the name of the library which contains Db2 Admin commands tables and the name of the library which contains the ADBDMTI EXEC.
   
   Requirement: Before you can use the INZDB21X member, you must generate it with Tools Customizer whenever you change the library names.
2. Run the ADB21S and ADB21T procedures that you have modified. These procedures are used to re-create the Db2 Admin Tool command tables.

Results
When ADB21S and ADB21T complete successfully, you can start Db2 HPU interactively by using the Db2 Admin.

Related reference
INZDB21X member
Use the INZDB21X member to integrate Db2 HPU with Db2 Administration Tool.
Chapter 4. Input and output

Db2 HPU uses various sources of input data, various formats for output data, and a user exit to customize the output data set that is created by a SELECT statement.

Topics:
- “Input data sources” on page 63
- “Output formats” on page 67
- “Specifying output file options” on page 73
- “LOB data processing” on page 76
- “XML data processing” on page 76
- “Output data consistency” on page 77

Input data sources

Db2 HPU can unload from various input data sources.

Topics:
- “Nonpartitioned table spaces” on page 63
- “Partitioned table spaces” on page 63
- “Full image copies or incremental image copies” on page 64
- “Last full image copies” on page 66

Nonpartitioned table spaces

Db2 HPU can use nonpartitioned table spaces as input data.

If the selected tables belong to the same table space, multiple unload requests can be processed on this object in parallel.

Example: Unloading tables in parallel

The mydb.myTS01 table space contains the me.myTable01 table and the me.myTable02 table. The UNLOAD command unloads both tables in parallel, as shown in the following example:

```
UNLOAD TABLESPACE mydb.myTS01
   SELECT * FROM me.myTable01
   OUTDDN (DDNTBL01)
   FORMAT VARIABLE END
   SELECT * FROM me.myTable02
   OUTDDN (DDNTBL02)
   FORMAT DSNTIAUL
```

Partitioned table spaces

Db2 HPU can use partitioned table spaces as input data.

Db2 HPU can use as input all partitions or a subset of partitions. As with nonpartitioned table spaces, Db2 HPU can process multiple unload requests in parallel.

The following examples show how to unload data in partitioned table spaces. In these examples, the mydb.myTS02 table space has five partitions.

Example: Unloading a partitioned table space

This example shows how to unload all partitions in the mydb.myTS02 table space. In this example, a single OUTDDN ddname, MYOUT, is allocated in the JCL. All rows are written in this ddname. If a
ddname per partition is allocated in the JCL, such as MYOUT01, MYOUT02...MYOUT05, the rows from each partition are unloaded in the corresponding ddname.

Example: Unloading specific partitions

This example shows how to unload partitions 1, 3, and 4 only.

```
UNLOAD TABLESPACE mydb.myTS02 PART (1,3,4)
SELECT * FROM me.myTable01 PART (1,3,4)
OUTDDN (MYOUT) PART (1,3,4)
FORMAT VARIABLE END
```

Example: Unloading partition subsets by using multiple SELECT statements

This example shows how to unload different subsets of partitions from a single table space by using multiple SELECT statements. Instead of specifying the list of partitions to be unloaded at the table space level, you can specify the partitions at the SELECT level.

```
UNLOAD TABLESPACE mydb.myTS02
SELECT * FROM me.myTable01 PART (1,3,4)
OUTDDN (MYOUT) PART (1,3,4)
FORMAT VARIABLE END
```

Example: Unloading a partitioned table space with one output file per partition in DB2 FORCE when an unsupported SELECT statement is used

This example shows how to unload data from partitions to get one output file per partition when an unsupported SELECT statement is used. The SELECT statement is not supported because the WHERE clause uses a subselect. A separate output file per partition is allowed because SQLPART (ALL) and a template that contains the &PART variable are specified.

```
TEMPLATE OUTFILE DSN qual1.&DB..&TS..P&PART. UNIT SYSDA SPACE(200, 200) CYL
UNLOAD TABLESPACE mydb.myTS02 DB2 FORCE
SELECT * FROM me.myTable01 SQLPART (ALL)
WHERE COL1 IN (SELECT COLREF FROM Reference.Table)
OUTDDN (OUTFILE) PART (1,3,4)
FORMAT DSNTIAUL
```

Related concepts

Full image copies or incremental image copies

Db2 HPU can use full image copies (FIC) or incremental image copies as input data.
If the input image copy is a global copy of an entire partitioned table space, Db2 HPU can generate one output file for every partition by allocating a ddname per unloaded partition in the JCL, or it can generate one output file per partition by using a TEMPLATE statement that contains the &PART variable.

When the input FIC corresponds to a table space that contains a single table, the header page contains the object ID (OBID) for this table. When this input FIC is unloaded and the ORIGINOBID keyword is specified, Db2 HPU checks to see whether both OBIDs match. If both OBIDs do not match, the unload is not processed, and an error message is issued.

The ddname that is specified in the COPYDDN keyword can correspond to a JCL-allocated file or to a TEMPLATE definition.

The following examples show how Db2 HPU uses FICs and incremental image copies as input to generate output files:

**Example: Unloading data from an image copy of an entire partitioned table space in a single output file**

The `mydb.myTS02` table space has five partitions. The CPYDDN ddname points to an image copy of this table space.

```
UNLOAD TABLESPACE mydb.myTS02
COPYDDN CPYDDN
SELECT * FROM me.myTable01
OUTDDN (MYOUT)
FORMAT VARIABLE END
```

**Example: Unloading data from an image copy of an entire partitioned table space with partition filtering and with one output file per partition**

The `mydb.myTS02` table space has five partitions. The CPYDDN ddname corresponds to a JCL-allocated file and points to a full image copy of this table space. One output file per partition is requested by using the &PART variable in the TEMPLATE definition for the output file. Partition filtering is requested by specifying the PART keyword in the UNLOAD command.

```
TEMPLATE OUTFILE DSN &USERID..&DB..&TS..U.P&PART. UNIT SYSDA
UNLOAD TABLESPACE mydb.myTS02 PART(1:3)
COPYDDN CPYDDN
SELECT * FROM me.myTable01
OUTDDN (OUTFILE)
FORMAT DSNTIAUL
```

**Example: Unloading data from an image copy with COPYDDN and DDLDDN as input files**

The CPYDDN1 and DDLDDN1 ddnames are used as input files.

```
//UNLOAD EXEC PGM=INZUTILB,PARM='DB81,DB2UNLOAD',
//REGION=0M,TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
// DD DSN=DSNA10.SDSNLOAD,DISP=SHR
// DD DSN=DSNA10.SDSNLOD2,DISP=SHR
//CPYDDN1 DD DISP=SHR,DSN=HLQ.FIC
//SYSREC DD DISP=(NEW,CATLG,DELETE),DSN=HLQ.UNLOAD,
//SPACE=(CYL,(5,5),RLSE),UNIT=SYSDA
//DDLDDN1 DD DISP=SHR,DSN=HLQ.DDL
//SYSIN DD *
//UNLOAD TABLESPACE COPYDDN CPYDDN1 DDLDDN DDLDDN1
SELECT * FROM me.myTable01
ORIGINOBID 0
OUTDDN (SYSREC)
FORMAT DSNTIAUL
```

**Unload from a set of image copies (a full image copy and three incremental image copies) allocated under the ICCONCAT ddname.**

As a concatenation of image copies is used as an input, the MERGE_IIC sub-option must be specified.

```
UNLOAD TABLESPACE
COPYDDN ICCONCAT,MERGE_IIC
```
SELECT *
FROM mytable
OUTDDN SYSREC

with ICCONCAT defined as follows:

//ICCONCAT DD DISP=SHR, DSN=full.image.copy.for.myts
// DD DISP=SHR, DSN=incremental.image.copy.for.myts.n01
// DD DISP=SHR, DSN=incremental.image.copy.for.myts.n02
// DD DISP=SHR, DSN=incremental.image.copy.for.myts.n03

Example: Unloading data from an image copy of a table space created by the COPYTOCOPY utility into a single output file

The CPYDDN ddname points to the image copy of the table space that was created by the COPYTOCOPY utility.

UNLOAD TABLESPACE mydb.myTS02
COPYDDN CPYDDN, FROM_COPYTOCOPY
SELECT * FROM me.myTable01
OUTDDN (MYOUT)
FORMAT VARIABLE END

Related tasks
Specifying one output file per partition
You can unload each partition in a table space to a separate file.

Last full image copies

Db2 HPU can use last full image copies as input data.

You can request Db2 HPU to unload data from the last full image copy that was taken for a table space by specifying COPYDDN LAST_IC, as shown in the following example:

UNLOAD TABLESPACE mydb.myTS02
COPYDDN LAST_IC
SELECT * FROM me.myTable01
OUTDDN (MYOUT)
FORMAT VARIABLE END

The last full image copy of a partitioned table space can be an image copy of the entire table space or an image copy of each partition. You can control which types of image copies are used by specifying various keywords.

- By using the GLOBAL keyword, you can specify Db2 HPU to use only global image copies.
- By using the PARTITIONED keyword, you can specify Db2 HPU to use only image copies of partitioned table spaces.
- By using the ANYTYPE keyword, you can specify Db2 HPU to use any type of image copy.
- By using the CONSISTENT keyword, you can ensure that the image copies of all unloaded partitions have the same START_RBA value in the SYSCOPY column.

The following example shows how you can unload data from a consistent image copy per partition with one output file per partition.

Example: Unloading data from a consistent image copy per partition of a partitioned table space with one output file per partition

The mydb.myTS02 table space has five partitions. One image copy per partition has been taken in a single COPY invocation. The ANYTYPE keyword authorizes Db2 HPU to use these image copies, if they are the most recent for each partitions. The CONSISTENT keyword ensures that all the image copies that are used as input have the same START_RBA value in the SYSCOPY column. One output file per partition is requested by using the &PART variable in the TEMPLATE definition for the output file.

TEMPLATE OUTFILE DSN &USERID..&DB..&TS..U.P&PART. UNIT SYSDA
UNLOAD TABLESPACE mydb.myTS02
COPYDDN LAST_IC ANYTYPE CONSISTENT
SELECT * FROM me.myTable01
Related reference

Db2 HPU last full image copies: COPYDDN option
See this topic for descriptions of the keywords in the GLOBAL OPTIONS block syntax.

Concurrent image copies

The concurrent image copies are not supported by Db2 HPU.

If a concurrent image copy is provided as an input via the COPYDDN ddname option, HPU fails.

If a concurrent image copy is referenced in the Db2 catalog and the COPYDDN LAST_IC or COPYDDN integer is specified, Db2 HPU ignores the concurrent image copies and does not take it into account when determining the rank of the image copy that satisfies the criteria specified as sub-option of the COPYDDN option.

Concatenation of image copies

When requesting to unload the data from image copies by using the "COPYDDN ddname" option, you may allocate a set of data sets under a single ddname. Before doing so, check the following requirements and considerations that apply to such a concatenation.

Items of the concatenation

The supported concatenations must:

• consist of non FlashCopy image copies of a single table space. If some of the data set do not contain the data from the expected table space, unpredictable results or failure can occur.
• satisfy one of the following requirements:
  – be a set of FIC of distinct partitions
  – be a set of a FIC and one or several IIC (requires the specification of the MERGE_IIC suboption)

Sequence of the data sets in the concatenation

There is no requirement for the sequence of the concatenated data sets except where the concatenation consists of image copies of partitions of a Universal Table Space -UTS), in which case the data sets must be concatenated according to the related partition number, in ascending order.

Consistency

As Db2 HPU cannot check the consistency of the concatenated data sets, the consistency of the unloaded data depends on the consistency of the input data.

Output formats

Db2 HPU can create output files in various formats for physical and logical unloads.

Use a physical unload to unload the entire table space, which includes all tables, columns, and rows unless sampling was requested. Use a logical unload to filter the rows, select the columns, and compute the output fields that you want to unload by using SELECT statements.

Topics:

• “Performing physical unloads” on page 68
• “Performing logical unloads” on page 68
• “Output encoding scheme” on page 73
Performing physical unloads

A physical unload starts unloading at the beginning of a table space and does not stop unloading until the entire table space is unloaded unless sampling is requested. A physical unload uses the same format as the IBM Db2 REORG utility to physically unload the table space.

About this task

The main use of data that is unloaded by a physical unload is to reload it in a table space that has the same structure or the same table space by using a LOAD with FORMAT UNLOAD.

When you use the REORG unload-only format, the table space is unloaded regardless of the tables that it contains.

You must use an exit to select the rows to unload or to change the format of these rows. You can limit the number of rows to be unloaded and take a sample of some rows. The sampling is done before the call to the exit.

Procedure

Specify either the table space name or a LISTDEF statement in the UNLOAD command, and use the UNLDDN parameter in the UNLOAD command.

Example

The following example shows how you can take a sample of some rows when you do a physical unload.

Example: Sampling rows to be unloaded

The mydb.myTS01 table space contains two tables. One out of every 10 rows is unloaded until the maximum number of rows is reached. The maximum number of rows is 150.

```
UNLOAD TABLESPACE mydb.myTS01
UNLDDN MYDDN
UNLMAXROWS 150
UNLFREQROWS 10
```

Important: When you do a physical unload, you can code one or more SELECT statements. However, you can have only one UNLDDN keyword per UNLOAD command.

The following example shows how you can unload data by using LISTDEF and TEMPLATE statements.

Example: Unloading data by using LISTDEF and TEMPLATE statements

The mydb database contains three table spaces. The LISTDEF and TEMPLATE statements unload each table space in a separate file with a DSN that includes the complete table space name and the date.

```
LISTDEF LIST1 INCLUDE TABLESPACE mydb.myTS%
TEMPLATE DDUNL DSN HLQ.&DB..&TS..&DATE..UNLOAD
UNLOAD TABLESPACE LIST(LIST1)
UNLDDN DDUNL
```

Performing logical unloads

A logical unload uses SELECT statements to filter the rows and columns that you want to unload. Use a logical unload to determine the exact content of the output data by using an SQL SELECT statement and to specify the output format.

Procedure

1. Specify one or more SELECT statements.

   The SELECT statement can specify the name of a table or view, or you can use a LISTDEFTBV statement to automatically generate a logical unload for each table from each table space in the LISTDEF.
You can use SQL comment indicators in these SELECT statements.

2. Specify the OUTDDN keyword for each SELECT statement to specify the output data set for each
SELECT statement.

   The OUTDDN keyword can refer to a JCL-allocated file or to a template to use dynamic data set
allocation.

3. Select one of the following output formats: DSNTIAUL, DELIMITED, VARIABLE, USER, EXTERNAL, or
INTERNAL.

   The following example shows how you can unload data by using SELECT statements.

**Example: Unloading data by using SELECT statements**

   The mydb.myTS01 table space contains two tables. One out of every 10 rows is unloaded until the
maximum number of rows is reached. The maximum number of rows is 150. The SELECT
statements request a logical unload of the same two tables.

```
UNLOAD TABLESPACE mydb.myTS01
UNLDDN MYDDN
UNLMAXROWS 150
UNLFREQROWS 10
   SELECT * FROM me.myTable01
   OUTDDN (MYOUTTB1)
   FORMAT VARIABLE END
   SELECT * FROM me.myTable02
   OUTDDN (MYOUTTB2)
   FORMAT VARIABLE END
```

Details about the various output formats are provided in the following sections.

**DSNTIAUL format**

   When you create output in the DSNTIAUL format, the output is identical to the output that is produced by
the DSNTIAUL program.

**Unloading a table in DSNTIAUL format**

   The following example shows how unload a table in DSNTIAUL format.

   The mydb.myTS01 table space contains two tables. This example shows how to unload the
myTable01
table in DSNTIAUL format.

```
UNLOAD TABLESPACE mydb.myTS01
   SELECT * FROM myQual.myTable01
   OUTDDN (MYDDN)
   FORMAT DSNTIAUL
```

**DELIMITED format**

   When you create output in the DELIMITED format, you can specify a separator character and a delimiter
character.

   CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC columns are enclosed by the delimiter character. Null
columns are not enclosed by the delimiter character if DELIM val and NULL DELIM are coded.

**Unloading a table by using a delimiter and a separator**

   The following example shows how to use a delimiter and a separator when you unload a table.

   The mydb.myTS01 table space contains two tables. The example shows how to unload a table by using an
asterisk (*) as the delimiter and a semicolon (;) as the separator.

```
UNLOAD TABLESPACE mydb.myTS01
   SELECT * FROM myQual.myTable01
   OUTDDN (MYDDN)
   FORMAT DELIMITED SEP ' ; ' DELIM ' * '
```

**Related reference**

   Db2 HPU output data parameters
The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

**VARIABLE format**
When you create output in the VARIABLE format, the output is compatible with the Db2 LOAD utility input data set.

**Unloading a table so that all variable-length fields are treated as variables**
The following example shows how to unload a table so that all variable-length fields are treated as variables.

Table space `mydb.myTS01` contains two tables. The example shows how to unload a table so that all the variable-length fields are treated as variables.

```sql
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT VARIABLE ALL
```

**Related reference**
Db2 HPU output data parameters
The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

**USER format - deprecated**
Please know that this format is deprecated. When you create output in the USER format, you can customize every output column in any manner that you want.

**Redundant thus deprecated feature**
The USER format feature is deprecated and is only supported for compatibility purpose. This feature that allowed to get a user-defined format and to specify the format at the column level was made redundant by the following capabilities:

- ability to customize any of the other formats via parmlib parameters,
- ability to specify the format at the column level via the INTO clause,
- ability to specify an OPTIONS block at the GLOBAL or UNLOAD level for all formats (VUU057/OPALLFMT parameter) and the more powerful REFORMAT feature that allow to defining rules for the format changes.

As no change (neither enhancements nor fixes) of this feature will ever occur, prefer using the above mentioned features.

For example, you can force the conversion between data types, change the date or time format, add or remove a length field, add or remove a null indicator, justify the content left or right, select a padding character, select a delimiter character for date or time, and so on.

**Changing columns with FORMAT USER**
In this example, the `mydb.myTS01` table space contains two tables. The example shows how to change column 1 from a VARCHAR(10) to a CHAR(15) with null indicator, pad the column with the number sign (#) and left align it. The other columns are unloaded in the default format.

```sql
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT USER (COL 1 TYPE CHAR(15)
               PADDING ' #'
               NULLID YES
               JUST LEFT)
```
In this example, the mydb.myTS01 table space contains two tables. The example shows how to change column 1 from a VARCHAR(10) to a CHAR(15) with null indicator, pad the column with the number sign (#) and left align it. The other columns are unloaded in the default format.

```sql
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT USER (COL 1 TYPE CHAR(15)
           PADDING '#'
           NULLID YES
           JUST LEFT)
```

**Concatenating columns with FORMAT USER**

In this example, the mydb.myTS01 table space contains two tables. The example shows how to concatenate columns FIRSTNAME, LASTNAME, and COMPANY to create a list of e-mail addresses. This example assumes that all these columns are VARCHAR.

```sql
UNLOAD TABLESPACE mydb.myTS01
SELECT FIRSTNAME, '.', LASTNAME, '@',
       COMPANY, '.com' FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT USER (COL FIRSTNAME  NULLID NO LENGTH REAL LENGTHBYTE NO,
              COL LASTNAME   NULLID NO LENGTH REAL LENGTHBYTE NO,
              COL COMPANY    NULLID NO LENGTH REAL LENGTHBYTE NO)
```

The following output is the result:

```
john.smith@company.com
```

The following examples show how to replace the FORMAT USER feature by other features.

In these examples, me.mytable is a table with columns defined as follows:

```sql
COL_A SMALLINT NOT NULL,
COL_B VARCHAR(10) NOT NULL,
COL_C DECIMAL(5,2),
COL_D VARCHAR(20),
COL_E SMALLINT
```

Depending on your settings of the following parameters

- NULL indicator, format USER (VUU014/ULNULL)
- DATE, format USER (VUU015/ULDNUM)
- TIME, format USER (VUU016/ULTIME)
- TIMESTAMP, format USER (VUU017/ULTMSTP)
- Decimal picture, format USER (VUU018/ULPIC)
- Display format for numeric values, format USER (VUU032/ULMASK)
- Position for NULL indicator, format USER (VUU022/NULLPOS)
- Fill NULL with padding character, format USER (VUU075/NULLPAD)
- Delimiter for external date, format USER (VUU055/DATEDEL)
- Delimiter in external time, format USER (VUU056/TIMEDEL)
• Trim the trailing blanks, format USER (VUU059/ULTRIM)

You might need to set the counterpart parameter for FORMAT VARIABLE with the same value to get the same output format as with FORMAT USER.

**Replacement example #1: same format as with FORMAT USER with no field specification**

Instead of using the USER format with no field specification such as in the following example:

```sql
UNLOAD TABLESPACE
SELECT * FROM me.mytable
OUTDDN(SYSREC)
FORMAT USER ()
```

consider using the FORMAT VARIABLE ALL in an UNLOAD statement as follows:

```sql
UNLOAD TABLESPACE
SELECT * FROM me.mytable
OUTDDN(SYSREC)
FORMAT VARIABLE ALL
```

**Replacement example #2: changing columns data type**

Instead of the following UNLOAD statement:

```sql
UNLOAD TABLESPACE
SELECT * FROM me.mytable
OUTDDN(SYSREC)
FORMAT USER (
  COL COL_A TYPE CHAR(10)
  , COL COL_B TYPE INTEGER
  , COL COL_C TYPE DECIMAL(20,5)
  , COL COL_D TYPE CHAR(30)
  , COL COL_E TYPE DECIMAL(15,5)
)
```

consider using an UNLOAD statement as follows:

```sql
UNLOAD TABLESPACE
SELECT *
INTO
  COL_A CHAR(10)
  , COL_B TYPE INTEGER
  , COL_C TYPE DECIMAL(20,5)
  , COL_D TYPE CHAR(30)
  , COL_E TYPE DECIMAL(15,5)
FROM me.mytable
OUTDDN(SYSREC)
FORMAT VARIABLE ALL
```

**Replacement example #3: applying rules for changing columns data type**

Instead of the following UNLOAD statement:

```sql
UNLOAD TABLESPACE
SELECT * FROM me.mytable
OUTDDN(SYSREC)
FORMAT USER (
  COL COL_A TYPE CHAR(30)
  , COL COL_B TYPE DECIMAL(10,0)
  , COL COL_D TYPE CHAR(30)
  , COL COL_E TYPE DECIMAL(10,0)
)
```

consider using an UNLOAD statement as follows:

```sql
UNLOAD TABLESPACE
SELECT *
FROM me.mytable
OUTDDN(SYSREC)
FORMAT VARIABLE ALL
OPTIONS REFORMAT (
  72
)
EXTERNAL format
When you create output in the EXTERNAL format, output fields are in the EXTERNAL format that corresponds to their default type, output records are fixed, and a field separator is not used.

Unloading files in FORMAT EXTERNAL and sorting the records in the table clustering index order
The following example shows how to unload each file in the EXTERNAL format and sort the records in the table clustering index order.

The mydb database contains three table spaces: myTS01, myTS02, and myTS03. Each of these table spaces contains three tables: me.myTable01A, me.myTable01B, and me.myTable01C for myTS01; me.myTable02A, me.myTable02B, and me.myTable02C for myTS02; and me.myTable03A, me.myTable03B, and me.myTable03C for myTS03.

LISTDEF LIST1 INCLUDE TABLESPACE mydb.myTS%
TEMPLATE DDUNL DSN HLQ.&DB..&TS..D&DATE..T&SEL.
GLOBAL OPTIONS TEMPLATESET( SEL = :SELNUM ) ;
UNLOAD TABLESPACE
SELECT * FROM LIST(LIST1) ORDER CLUSTER OUTDDN DDUNL FORMAT EXTERNAL

Related reference
Db2 HPU output data parameters
The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

Output encoding scheme
For the DSNTIAUL, DELIMITED, VARIABLE, USER, and EXTERNAL output formats, the translations from EBCDIC to ASCII and from ASCII to EBCDIC are supported only for single-byte character set (SBCS) character strings. Data is translated by using the translation tables in the SYSIBM.SYSSTRINGS table.

Other types of translation are done by using Unicode Conversion Services.

Related reference
OPTIONS block syntax and description
Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

Specifying output file options
You can override the automatic output file allocation, specify one output file per partition, or limit the number of unloaded rows and call a user exit.

Db2 HPU allows to unload data into the following types of dataset:
- sequential data set
- spanned data set
- PDS member
- HFS data set

Topics:
- “Overriding the automatic output file allocation” on page 74
- “Specifying one output file per partition ” on page 74
- “Limiting the number of unloaded rows and calling a user exit” on page 75
Overriding the automatic output file allocation

Db2 HPU automatically calculates the output file allocation by using the DCB parameters that you provided in the JCL for each OUTDDN and UNLDDN that is specified in the SYSIN. However, you can override the automatic allocation.

Before you begin

Ensure that the allocation that you want to specify is compatible with the unload that you request by ensuring that the record length is long enough to contain the longest row that you want to unload. If you do not specify a DCB parameter in the DD statement, Db2 HPU uses the calculated DCB.

About this task

To override the automatic output file allocation:

Procedure

1. Ensure that the syntax that is used allows you to override the output file allocation, and specify DFSIGDCCB YES.
2. Code DD statements in the JCL, or specify a TEMPLATE definition.

Db2 HPU generates the output data sets based on the JCL or the TEMPLATE definition.

Specifying one output file per partition

You can unload each partition in a table space to a separate file.

Procedure

To specify one output file per partition, do one of the following steps:

- Use a TEMPLATE with a dsname that contains the &PART variable or code a base ddname in the OUTDDN statement, and declare this ddname in your JCL.
- Use the partition number as a suffix on the base ddname.

The following example shows how to specify one output file per iteration by using the partition number as a suffix.

Example: Specifying one output file per iteration by using the partition number as a suffix on the base ddname

In this following example, the mydb.myTS02 table space is a partitioned table space with five partitions. The example shows how to unload only partitions 1, 3, and 4 into separate output files by using the partition number as a suffix on the base ddname. The base ddname is MYOUT.

```
UNLOAD TABLESPACE mydb.myTS02 PART (1,3,4)
    SELECT * FROM me.myTable01
    OUTDDN (MYOUT)
    FORMAT VARIABLE END
```

In the JCL, specify the following DD statements:

```
MYOUT001 DD ...
MYOUT003 DD ...
MYOUT004 DD ...
```

The following example shows how to specify one output file per iteration by using a TEMPLATE statement.

Example: Specifying one output file per iteration by using a TEMPLATE statement

In this example, the me.myTable02 table was created in a partitioned table space with 128 partitions. The example shows how to unload all partitions into separate output files. The unloaded data is
sorted in the order of the clustering index, and the output files are dynamically allocated by using a template named PARTDS.

```
TEMPLATE PARTDS DSN HLQ.&DB..&TS..&DATE..&PART.
UNLOAD TABLESPACE
SELECT * FROM me.myTable02
ORDER CLUSTER
OUTDDN PARTDS FORMAT DSNTIAUL
```

The following example shows how to specify one output file per partition in an unload from a global full image copy.

**Example: Specifying one output file per partition in an unload from a global full image copy**

In this example, a global FIC exists for the DBINFD.M.TSPART6 partitioned table space. This FIC is allocated in the JCL by using the COPYDD statement.

```
TEMPLATE SYSREC DSN IBMUSER.&DB..&TS..P&PART.
UNLOAD TABLESPACE DBINFD.M.TSPART6
COPYDDN COPYDD
SELECT * FROM TBPART6
OUTDDN SYSREC
FORMAT DSNTIAUL
```

**Limiting the number of unloaded rows and calling a user exit**

An OUTDDN statement can be followed by one of several keywords that you can use to limit the number of unloaded rows, sample these rows, and call a user exit.

**Procedure**

1. To limit the number of rows, do one of the following steps:
   - For a physical unload, use the UNLMAXROWS and UNLFREQROWS keywords.
   - For a logical unload, use the OUTMAXROWS and OUTFREQROWS keywords.
     - For a SELECT statement that is processed by Db2, the OUTMAXROWS and OUTFREQROWS keywords are always applied on rows that are returned by Db2 after the WHERE and ORDER BY clauses are applied.
     - By default, for a SELECT statement that is natively processed by Db2 HPU, the OUTMAXROWS and OUTFREQROWS are applied when the rows are read in the VSAM LDS files. To be consistent with a SELECT statement that is processed by Db2, use the optional keyword ON_RESULT_TABLE after OUTMAXROWS and OUTFREQROWS.
   2. To request that each output row that is produced by Db2 HPU be processed by a user exit before it is written to the output file, use the OUTEXIT option.

**Unloading partitions into separate output files, sampling rows, and calling a user exit**

The following example shows how to unload partitions into separate output files, sample rows, and call a user exit.

In this example, the `mydb.myTS02` partitioned table space has five partitions. The example shows how to unload only partitions 1, 3, and 4 into separate output files. One row out of 10 is selected and passed to a COBOL/2 user exit that applies user modifications before output. The unload process stops when 150 rows have been selected.

**Requirement:** The exit must be in STEPLIB or in LINKLIST and must reside in an authorized library.
**LOB data processing**

Db2 HPU can unload LOB data.

**Supported LOB data sources**

You can unload LOB input data on the base table space, but support is limited by the following restrictions:

- Db2 HPU cannot directly unload a LOB table space.
- Db2 HPU does not support the unloading of LOB data from non-flash image copies.
- Unloading LOB data is only supported for a logical unload. When a physical unload is done on a base table space that contains a table with an LOB column, the LOB data is not unloaded.

**Unloading LOB data along with other data**

Unloading LOB data in the same output data set that contains the remaining columns of the table is called instream processing. In-stream processing is supported only when the SELECT statement is processed by Db2, which means that DB2 FORCE or DB2 YES was specified along with an unsupported SELECT statement. When this method is used, Db2 HPU might have to truncate the LOB data to limit the record size to the maximum size that is allowed.

With the SPANNED YES option, you can unload LOB data and other data type columns (except for XML data) into a single data set without any risk of truncation and in any of the processing modes.

**Unloading LOB data in separate data sets**

Unloading LOB data by using a LOB file reference is supported in all cases (DB2 NO, DB2 YES, or DB2 FORCE), and truncation cannot occur. To use a LOB file reference, specify the BLOBF, CLOBF, or DBCLOBF option in a REFORMAT clause, an INTO clause, or a USER format definition.

**Related reference**

Example: Unloading data from a table with a LOB column by using a TEMPLATE statement
This example shows how to use a TEMPLATE statement to unload data from a table with a LOB column (COL_LOB).

**XML data processing**

Db2 HPU can unload XML data.

**Supported XML data sources**

You can unload XML input data from the base table space, but support is limited by several restrictions:

- Db2 HPU cannot directly unload an XML table space.
- Db2 HPU does not support the unloading of XML data from non-flash image copies.
- Unloading XML data is supported only for a logical unload. When a physical unload is done on a base table space that contains a table with an XML column, the XML data is not unloaded.

**XML output**

Unloading XML data in the same output file that contains the remaining columns of the table is called instream processing. It is supported only when the SELECT statement is processed by Db2, but truncation of the XML data is not supported. Processed by Db2 means that DB2 FORCE or DB2 YES was specified along with an unsupported SELECT statement. When this method is used, if an XML value is longer than the output field, an SQL CODE -433 error is issued by Db2, and processing ends.
Unloading XML data by using a LOB file reference is supported when DB2 NO, DB2 YES, or DB2 FORCE is specified. Truncation cannot occur. To use a LOB file reference, specify the BLOBF, CLOBF, or DBCLOBF option in a REFORMAT clause, an INTO clause, or a USER format definition.

Related reference
Example: Unloading data from a table with an XML column by using a TEMPLATE statement
This example shows how to use a TEMPLATE statement to unload data from a table with an XML column.

Output data consistency

Ensuring output data consistency depends on the data source and the settings of various options.

The data consistency of the output during the unload process depends on the following data sources and keywords:

Data sources
- A table accessed through Db2 by SQL access.
- The table space that contains the table by direct access.
- An image copy data set of the table space.

Keywords
- LOCK and QUIESCE.

Topics:
- “Interaction with the Db2 catalog” on page 77
- “Consistency considerations” on page 77

Interaction with the Db2 catalog

Db2 HPU uses SQL access to read the descriptions of the unloaded objects in the Db2 catalog.

SQL access
Db2 HPU uses SQL to access the Db2 catalog. This method returns the most current value of information that is found in the Db2 catalog.

The user ID that is used to access the Db2 catalog depends on the setting of the VUM032/ACTLGUSR parameter.

Related reference
Db2 HPU general parameters: VUM027/SQLACCES
The Db2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the Db2 parameters that are used by Db2 HPU.

Consistency considerations

You should understand the consistency considerations when you unload data from an image copy or from a table.

Data that is unloaded from an image copy

When you unload data from an image copy of the table space, the level of consistency is inherited from the SHRLEVEL option that was run with the COPY utility.

When you unload data from an image copy, the updates on the table at run time have no impact on the output data. Therefore, do not use the QUIESCE and LOCK keywords in this case because they are ignored.

When you unload data from image copies that are taken per partition with the COPYDDN LAST_IC PARTITIONED, COPYDDN LAST_IC ANYTYPE, or COPYDDN number PARTITIONED keywords, you can ensure the consistency between the image copies of each partition by using the CONSISTENT keyword so that all input full image copies that were used have been taken at the same START_RBA.
Data that is unloaded from a table or its underlying table space

When you unload from the table itself (the online data that is handled by Db2), the DB2, QUIESCE, and LOCK keywords can affect the output data consistency.

**DB2**

Db2 HPU retrieves the data from the table by using one of the following methods:

**Direct access**

When the SELECT statement can be evaluated by Db2 HPU, the data is read directly from the table space linear data sets (LDS) that contain the table and the related index space LDS, when they are applicable.

**SQL access**

When the SELECT statement cannot be handled by Db2 HPU, the data selection is done by Db2.

The setting of the DB2 parameter affects not only the data consistency but also the performance. Consider both data consistency and performance when you select the appropriate DB2 option.

The following options for the DB2 parameter are available:

- **FORCE**
  - SQL access is always used.

- **YES**
  - Direct access is used when the SELECT statement can be directly processed by Db2 HPU.
  - Otherwise, SQL access is used.

- **NO**
  - Direct access is used, or the UNLOAD statement is rejected because Db2 HPU cannot process it.

**QUIESCE**

When direct access is used with the unload, some data from the table might be updated, and the current data might be stored only in the Db2 buffers. To ensure that all the updates are written into the table space LDS, set the QUIESCE keyword to YES. However, new updates can occur during the unload. Specifying QUIESCE YES does not prevent new updates from happening, so this setting is not sufficient to ensure consistent data. You must specify QUIESCE YES when you use the Db2 direct access retrieval method.

**LOCK**

To prevent concurrent access for updates, set the LOCK keyword to YES. This option applies to SQL and/or direct access modes depending on the setting of the "Eligible processing mode(s) for locking data feature (VUU088/ULLOCKMO)" parmlib parameter. When you specify LOCK YES, the table can be accessed only in read-only mode while Db2 HPU is unloading data from the table. YES is required when any access mode is selected.

You can specify the DB2, QUIESCE, and LOCK keywords in the Db2 HPU SYSIN and when you customize Db2 HPU. Use the VUU011/ULSEDDB2, VUU013/ULQSCE, and VUU012/ULLOCK parameters to control the default behavior of Db2 HPU with these options.

When you specify QUIESCE, set an appropriate value for the VUU028/ULQSCEBH parameter.

**Tip:** To ensure data consistency, consider specifying QUIESCE YES and LOCK YES. The DB2 FORCE and LOCK YES combination also ensures data consistency, but reduces performance.

The following table shows the consistency level that is achieved when you specify different combinations of the QUIESCE, LOCK, and DB2 keywords:

<table>
<thead>
<tr>
<th>QUIESCE setting</th>
<th>LOCK setting</th>
<th>DB2 setting</th>
<th>Consistency level</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>Not ensured (1,2)</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>Not ensured (2,3)</td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td>FORCE</td>
<td>Not ensured (2)</td>
</tr>
<tr>
<td>QUIESCE setting</td>
<td>LOCK setting</td>
<td>DB2 setting</td>
<td>Consistency level</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>Not ensured (4)</td>
</tr>
<tr>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>Not ensured (3)</td>
</tr>
<tr>
<td>NO</td>
<td>YES</td>
<td>FORCE</td>
<td>Ensured (5)</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>Not ensured (2)</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>Not ensured (2)</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td>FORCE</td>
<td>Not ensured (2,6)</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>Ensured</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>Ensured</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>FORCE</td>
<td>Ensured (5,6)</td>
</tr>
</tbody>
</table>

**Notes:**

1. You can obtain data consistency by performing a QUIESCE or a STOP/START sequence against the table. However, other action might be needed to achieve consistency. Refer to notes 2 - 6 for more information.

2. Data consistency is obtained only if no Db2 update is made against the table while the unload process is in progress.

3. Data consistency depends on the SELECT statement because it can be processed in direct access mode or in SQL access mode. If direct access is selected by Db2 HPU, note 1 also applies.

4. Avoid this setting because it locks the table without ensuring that the output data is consistent.

5. Although this setting ensures consistency, it can lead to lower performance because any SELECT statement will be processed in the SQL access mode. Avoid using this setting.

6. Specifying QUIESCE YES in this case is useless because DB2 FORCE is also specified. Specify QUIESCE YES only when you specify DB2 NO or DB2 YES.

**Related reference**

Db2 HPU general parameters: VUM027/SQLACCE

The Db2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the Db2 parameters that are used by Db2 HPU.
Chapter 5. Db2 HPU batch mode operation

You use the Db2 HPU batch utility program to unload Db2 data.

Db2 HPU uses syntax and JCL that is compatible with the standard Db2 UNLOAD utility. However, Db2 HPU does not support the entire Db2 syntax for SELECT statements. SELECT statements that are not supported are passed to Db2 for processing.

Topics:
- “Required privileges for running Db2 HPU” on page 81
- “JCL for running Db2 HPU in batch mode” on page 82
- “Db2 HPU EXEC statement” on page 82
- “Ddname allocation” on page 87
- “Db2 HPU syntax” on page 90
- “The SYSPRINT data set” on page 232

Required privileges for running Db2 HPU

You must have specific user-authority privileges to run Db2 HPU.

The following privileges are required:
- When RACF® is used, the READ authority is not required on Db2 data sets that contain the Db2 catalog and the unloaded Db2 table spaces and indexes because Db2 HPU bypasses the RACF control. When other security products are used, such as ACF2, Db2 HPU does not bypass the control. Therefore, the READ authority is required on all the Db2 data sets that Db2 HPU accesses.
- If an image copy is unloaded, RACF READ authority is required on the image copy data set.
- Db2 SELECT privilege is required on the tables or views to be unloaded, and DISPLAYDB privilege is required on the database.
- If QUIESCE YES is specified, you must be authorized to run the Db2 QUIESCE utility for the table space.
- If LOCK YES is specified, you must have SELECT privilege on all tables of the unloaded table space.
- Db2 HPU can use real-time statistics to more accurately estimate the amount of data to be processed, which improves the allocation of work data sets. To use real-time statistics, the owner of the Db2 HPU plan, which is set by the VUM012/PLANOWN parameter, must have the Db2 SELECT privilege for the SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSSPINDEXSPACESTATS tables. If the requested authority is not available, Db2 HPU uses a less accurate method for estimating the amount of data.
- For unload run in native mode against online Db2 objects (not against image copy data sets) with underlying datasets encrypted using ICSF key label, the Db2 HPU job must have the authorization to access key label for the involved encrypted data sets. This authorization is inherited from the privileges of the userid specified through is inherited the (VUM037/CRYPTUSR parmlib parameter. For a Db2 like behavior, set this parameter with a userid that has the authorization to access the key label for any of the encrypted data set involved by your Db2 HPU jobs.
- For an unload from the image copy of a Db2 object encrypted with underlying datasets encrypted using ICSF key label, the userid specified for the USER parameter of the Db2 HPU job must have the authorization to access the keylabel of the involved encrypted image copy data set.
JCL for running Db2 HPU in batch mode

Sample JCL for running Db2 HPU in batch mode is provided in SINZSAMP member INZEXECU. You can copy and modify this JCL as needed for your environment.

For example, two SELECT statements have been added to the following JCL. These SELECT statements unload data from the USER01.TABLE01 table and the USER01.TABLE02 table, both of which reside in the DBNAME1.TSNAME1 table space.

```plaintext
//*****************************************
//*  DB2 UNLOAD JCL
//*  IN THIS SAMPLE :
//*  - THE DB2 SUBSYSTEM IS DB2P
//*  - THE DB2 UNLOAD LOAD MODULES ARE
//*    IN THE LOADLIB &VIZ004
//*  - THE EXECUTION REPORT WILL BE
//*    WRITTEN ON THE ddname SYSPRINT
//*  *****************************************
//STEP1 EXEC PGM=INZUTILB,REGION=0M,DYNAMNBR=99,
//      PARM='DB2P,DB2UNLOAD'
//STEPLIB DD  DSN=DB2UNLOAD.LOAD,DISP=SHR
//         DD  DSN=PRODDB2.DSNEXIT,DISP=SHR
//         DD  DSN=PRODDB2.DSNLOAD,DISP=SHR
//*/
//SYSIN DD  *
UNLOAD TABLESPACE DBNAME1.TSNAME1
   DB2 YES
   QUIESC YES
   OPTIONS DATE DATE_A
         SELECT COL1,COL2 FROM USER01.TABLE01
         ORDER BY 1 , COL2 DESC
         OUTDDN (UNLDDN1)
         FORMAT VARIABLE ALL
         SELECT COL3,COL4 FROM USER01.TABLE02
         OUTDDN (UNLDDN2)
         FORMAT DSNTIAUL
         LOADDDN LOADDDN1 LOADOPT (RESUME NO REPLACE)
/*
//SYSPRINT DD  SYSOUT=* 
//*********** ddnameS USED BY THE SELECT STATEMENTS ***********
//*/
//UNLDDN1 DD  DSN=.....,DISP=SHR
//UNLDDN2 DD  DSN=.....,DISP=SHR
//LOADDDN1 DD  DSN=.....,DISP=SHR
```

Db2 HPU EXEC statement

A sample EXEC statement that you use to run Db2 HPU in batch mode is provided in SINZSAMP member INZEXECU. You can copy and modify this sample as needed for your environment.

The following example shows the format of the EXEC statement:

```
EXEC PGM=module-name,PARM='subsystem-name/group-attachment-name,job-ID[,HIDDEN=YES/NO)]'
```

The EXEC statement contains the following parameters. Content in the PARM field must be enclosed in single quotation marks (').

**Module-name**

INZUTILB: Specify this name to process

- a Db2 HPU native syntax
- an Unload Plus syntax
- a Fast Unload syntax.
INZUTILC: Specify this name to process

- a Db2 UNLOAD syntax

**Note:** INZUTILB is an alias on either the INZUTIB1 or INZUTIB2 load modules defined through the customization job built from the template INZALIAS. The relevant load module is chosen at customization time according to the setting for Management of exit written in C or COBOL (VUU079/EXITHLL).

**subsystem-name/group-attachment-name**

Specifies the name of the Db2 subsystem. In a non-data-sharing environment, this value is the name of a Db2 subsystem. In a SYSPLEX data-sharing environment, this value may be either a Db2 group attachment name or the name of one of the subsystems that belongs to the data sharing group. The name must be one of the SSID listed on the Tool Customizer "Associate Db2 Entry for Product".

**job-ID**

Specifies the unique identifier for your Db2 HPU job. Special characters are not permitted. This identifier is used as a utilid when Db2 HPU needs to invoke a Db2 utility (for example the QUIESCE utility).

If not specified, Db2 HPU launches the Db2 utilities with an utilid built with userid.jobname unless the Fastunload syntax is used, in which case the utilid is built with jobname.userid.

For concurrency purpose, make sure you do not try to run Db2 HPU jobs using the same utilid - as described previously - against the same Db2 subsystem/data sharing simultaneously.

**HIDDEN**

Specifies whether hidden columns are unloaded when a SELECT * statement from a table name or from a LISTDEF statement is used. This parameter is optional. It applies only to SELECT statements that are processed natively by Db2 HPU. Use the HIDDEN keyword in the OPTIONS block to override this parameter.

**NO**

Specifies that hidden columns are not unloaded when a SELECT * statement is used.

**YES**

Specifies that hidden columns are unloaded when a SELECT * statement is used. For SELECT statements that are processed by Db2, the HIDDEN parameter is ignored.

If the HIDDEN parameter is not specified in the SYSIN or in the PARM field, the default value is the value that is specified for the VUU042/ULHIDDEN PARMLIB parameter.

**Important:** If you specify FASTUNLOAD in SYSIN to use the Fast Unload syntax, precede `subsystem_name/group_attachment_name/job_id` in the PARM field with the `EP=UTLGLCTL` positional parameter, as shown in the following example:

```
PARM='EP=UTLGLCTL/DSN5,,,,DB2UNLOAD'
```

**Note:** The input parm string can be specified in a file allocated to the ddname PARM or to the ddname specified in the EXEC PARM, using the syntax `PARM='DDNAME=ddname'` positional parameter, as shown in the following example:

```
EXEC INZUTILB,PARM='DDNAME=MYDD'
```

---

# Memory use considerations

Here is some advice on how to set the REGION parameter for a Db2 HPU step.

**Impacting parameters**

The amount of memory needed for a Db2 HPU step depends on a lot of parameters such as:
• the Db2 object composition (number of partitions or datasets, number of related indexes, number and data types of columns)
• the unload statement complexity (number of involved Db2 objects, number of SELECT, number of output datasets....)
• the processing mode (native vs SQL mode, parallelism degree, SORT program used...)

Minimum value
The minimum amount of memory can be determined by running a plain unload or, better, what you consider your typical unload and to measure the amount of memory. It should be at least:
• 30M if VUU079/EXITHLL is set to NO
• 40M if VUU079/EXITHLL is set to YES

Maximum value
Most of the memory is allocated above the 16 MB bar and increases with the unload complexity if the parallelism degree is not limited. However, as some memory must be allocated below the 16 MB bar, the maximum complexity that an unload can reach is limited by the 16 MB memory thus preventing an unload job to use a very big amount of memory above the 16 MB. However, as computing and predicting the exact amount of memory needed for a given unload job is very complicated and does not make much sense when parallelism is involved, it is recommended to set the REGION parameter to 0M so that the required memory is available to the unload job.

Processing mode considerations

Native processing versus SQL mode processing
Db2 HPU retrieves the data to be unloaded in two ways:
• either by directly accessing the data from the data sets where it is stored (either the VSAM data sets handled by Db2 or the image copy data sets),
• or by requesting Db2 to provide the data from its buffers.

The first is called the native processing (also referred to as native mode) and the latter is called the SQL processing (also referred to as SQL mode). The applicable modes depend on the kind of unload to be performed. When both modes are available for a given unload, the mode to be used can be specified via the DB2 option or its default counterpart in the Db2 HPU settings (VUU011/ULSEDB2) with the Db2 HPU syntax or with an equivalent keyword with the other syntaxes.

The table below shows which mode is applicable.

<table>
<thead>
<tr>
<th>Unload request</th>
<th>Native mode</th>
<th>SQL mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical unload (1)</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Unload from an image copy (2)</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Logical unload (3) with a supported SQL statement (4)</td>
<td>YES (5)</td>
<td>YES (5)</td>
</tr>
<tr>
<td>Logical unload (3) with an unsupported SQL statement (4)</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

(1) Db2 HPU UNLDDN option or equivalent feature with other syntaxes.
(2) Db2 HPU COPYDDN option or equivalent feature with other syntaxes.
(3) Db2 HPU OUTDDN option or equivalent feature with other syntaxes.
Db2 HPU only supports a part of the SQL SELECT statements (see “Fast select, fast listdef and fast listdeftbv select blocks syntax and description” on page 188 for more details).

Db2 HPU selects the mode according to the instructions or preferences given by the DB2 option or its default counterpart in the Db2 HPU settings (VUU011/ULSEDB2).

**SQL processing details**

When Db2 HPU passes a SELECT statement to Db2 to retrieve the data (SQL mode), the following systematical or optional actions that might have an influence on the way Db2 processes the statement are performed.

- The FOR READ ONLY clause is appended to the SELECT statement (1).
- The USE CURRENTLY COMMITTED or WAIT FOR OUT-COME is appended to the SQL statement according to the ACCPREP specification (via the ACCPREP option in technical parameters block or the VUU061/ACCPREP parameter) (2).
- The SQL statement, with potential changes described above is run after some special registers have been set as follows:
  - The CURRENT DEGREE special register is set according to the ULDEGREE option in technical parameters block or the VUU021/ULDEGREE parameter (1)
  - The CURRENT SQLID special register is set according to the SQLID specified via the GLOBAL OPTIONS block (2).
  - The CURRENT QUERY ACCELERATION special register is set according to the QUERY_ACCELERATION option in the OPTIONS block (2).
  - The CURRENT OPTIMIZATION HINT special register is set according to the OPTIMIZATION_HINT option in the OPTIONS block (2).
- According to the UNLROWSET option or the VUU035/ULROWSET parameter, the SQL statement is run in either single or multi-row fetch mode (1).

(1) systematical action

(2) optional action

**OMVS threads requirements**

Db2 HPU uses OMVS threads for any batch processing. Each Db2 HPU step (EXEC statement) launches a single OMVS process that is closed when the Db2 HPU step ends. This process is launched on behalf of the UID associated to the userid of the submitter of the job that includes the Db2 HPU step.

Consequently, the settings of some z/OS Unix parameters have an impact on the Db2 HPU jobs as described below. Take these impacts into account to choose settings that meet your requirements in terms of Db2 HPU use.

**MAXTHREADTASKS**

This parameter has an impact on the maximum parallelism degree allowed for a single OMVS process thus for a Db2 HPU step.

The number of threads needed for an unload strongly depends on the processing mode and is much bigger for the unload processes run in native mode.

For the native mode, the number of threads needed by Db2 HPU can be computed with the help of following rules:

- 20 base threads always used per unload statement and,
- for every partition unloaded in parallel -- 2 threads always used -- 1 thread if the data is unloaded from an INLINE image copy -- 2 threads if ZIIP mode is enabled -- 1 thread if a SORT process is involved -- 1 thread for every LOB column to unload -- 4 threads for every XML column to unload -- 1 thread if SPANNED YES is requested
For example, for an unload, run in native mode, from a partitioned table space with no LOB or XML to
be unloaded and partition parallelism set to 50, the number of threads needed is 20 + 50 (2 + 1) =
170 threads.

The MAXTHREADTASKS parameter should be set to a value greater than or equal to the number of
threads needed for the more demanding of your unload steps in terms of threads. A rough and very
likely oversized (unless you have objects with a large number of LOB or XML columns) estimate based
on the number of partitions of the table space to unload is given by the following formula: 20 +
partitions_number * 12.

For unload steps with several unload statements and table space parallelism enabled (via the third
argument of the PARALLELISM option VUX030/UTLPARAL PARMLIB parameter), the contributions of
the table space processed in parallel must be added to compute the number of threads of the step.

MAXPROCSYS and MAXPROCUSER

The number of Db2 HPU steps that can be run concurrently is limited by the following rules:

- the number of Db2 HPU steps, regardless of the userid of the submitter of the related job, does not
  exceed the value set for MAXPROCSYS
- the number of Db2 HPU steps in jobs submitted by userids associated with the same z/OS Unix UID
does not exceed the value set for MAXPROCUSER

The actual limit set by the second rule depends on:

- the way z/OS userids are associated with z/OS Unix UIDs
- the dispatching of jobs submitters

The limit increases whenever:

- the number of involved UID increases
- the distribution of jobs among UID is flatter

For a strictly flat distribution, i.e. each UID is associated to the same number of Db2 HPU jobs, the
maximum number of Db2 HPU steps that can be run simultaneously is given by the following formula:
min (MAXPROCSYS, number_of_UID_involved * MAXPROCUSER). The number of jobs that can
actually be run in parallel can be less than this value if other programs use z/OS Unix processes or if
the outage of another shared system resource occurs.

Enabling the use of zIIP processors

The use of zIIP processors can be requested:

- either by default (for all unload jobs) by setting the VUX039/ZIIP parmlib parameter
- or at the job level by using the ZIIP option of the Technical parameters block in the GLOBAL OPTIONS
  block

If you want to use the zIIP processors, either set VUX039/ZIIP to 100 or specify ZIIP 100 in your unload
SYSIN. Do not specify another value unless directed to by the IBM support.

In most cases (i.e. in any processing mode, unload jobs involving complex formatting and, in native mode,
unload jobs with a SELECT statement involving SQL expressions or WHERE clause), the use of zIIP
processor is efficient. Therefore, enabling the use of zIIP processor via the parmlib parameter is relevant.
However, for some specific unloads among those are not supposed to benefit from the zIIP use (unload in
INTERNAL or physical format), the use of zIIP processors might bring no gain or even lead to a slight
increase of the Global Purpose CPU use. For such unloads, consider adding ZIIP OFF to the GLOBAL
OPTIONS block to disable the zIIP whatever the corresponding parmlib setting. As the ZIIP option applies
to all unloads of the unload step, avoid mixing unloads that can benefit from the zIIP processors use and
others that do not take advantage of it in a single unload command.

If no zIIP processor is available, specify ZIIP OFF via the parmlib parameter to prevent Db2 HPU from
checking if a zIIP processor can be used.
When you request Db2 HPU to use the zIIP processors (i.e. “ZIIP percentage” applies), some messages report how the request has been taken into account.

For example, if ZIIP 100 is specified on a machine where no zIIP processor is available, messages such as the following are issued:

```
INZI211I ONLINE CPU(S)
  2 GENERAL CPU
  0 ZIIP CPU
  0 OTHER CPU

INZI212I NO ZIIP AVAILABLE, ZIIP SET TO OFF
```

and messages as follows are issued if a zIIP processor is available.

```
INZI211I ONLINE CPU(S)
  2 GENERAL CPU
  1 ZIIP CPU
  0 OTHER CPU

INZI210I ZIIP ENABLE - SET TO 100 %
```

If a technical issue prevents Db2 HPU to use the zIIP processor as requested, the INZI213W message is issued.

**Ddname allocation**

Db2 HPU uses two types of ddnames: ddnames that Db2 HPU allocates and ddnames that you must allocate.

**Db2 HPU-allocated ddnames**

Db2 HPU dynamically allocates some of the ddnames that are required to run unload jobs.

The following reserved ddnames are allocated dynamically by Db2 HPU:

- **dispparm**
  This user-defined ddname - named dispparm hereafter - is dynamically allocated by Db2 HPU as a SYSOUT data set if the "Display implicit parameters" feature - driven either by the VUU083/DISPPARM parmlib parameter or the SYSIN DISPLAY_PARMLIB option from the Technical parameters options block - is enabled by choosing the YES,dispparm and the dispparm ddname is not already allocated by the user.

- **dispexpl**
  This user-defined ddname - named dispexpl hereafter - is dynamically allocated by Db2 HPU as a SYSOUT data set if the "display of a EXPLAIN report" feature - driven either by the VUU087/DISPEXPL parmlib parameter - is enabled by choosing the YES,dispexpl and the dispexpl ddname is not already allocated by the user. Db2 HPU sets RECFM=VB and LRECL=124 as DCB attributes of this dataset, regardless of explicit DCB parameters specified through the DD statement.

- **ICxx-ICxxnnnn**
  The ddname that is used to allocate the input image copy, where xx is a unique two-character alphanumeric value, and nnnnn is the partition number. This ddname is used in the following conditions:
  - When you specify COPYDDN LAST_IC or COPYDDN integer, the ddname that is used to allocate the input image copy is ICxx or ICxxnnnn, depending on whether you specified the PARTITIONED keyword or the ANYTYPE keyword.
  - When you use the COPYDDN keyword with a template name, ICxx is used to allocate a global input image copy, and ICxxnnnn is used to allocate an input image copy per partition.
**LBFnnnnn**
The ddname that is used to allocate the output file for a LOB file reference, where **nnnn** is the partition number. The same ddname is reused during the unload process to allocate all LOB file references for a specific LOB column and a specific partition.

**SORT ddnames**
Db2 HPU calls the sort utility when you specify an ORDER BY clause or an ORDER CLUSTER clause, which means that sort ddnames are dynamically allocated.

**User-allocated ddnames**
To run unload jobs, you must allocate certain ddnames in the Db2 HPU JCL.

To allocate ddnames, specify the following DD statements in your Db2 HPU JCL. Some of these DD statements are required and some are optional:

**STEPLIB or JOBLIB**
To use Db2 HPU, the job must have access to the Db2 HPU LOADLIB, the Db2 DSNLOAD and the Language Environment SCEERUN and SCEERUN2 libraries. The Db2 DSNLOD2 library is needed if the DSNAO64C module is stored in this library. Make sure that any library allocated under the STEPLIB ddname is APF authorized. You can give Db2 HPU access to these libraries by specifying a JOBLIB DD statement, a STEPLIB DD statement, or by including the library names in the LINKLIST.

If you do not specify the DSNEXIT library in the VZD007 PARMLIB parameter for the corresponding Db2 subsystem, you must specify the DSNEXIT library in the STEPLIB or JOBLIB libraries, and you must specify it before the Db2 DSNLOAD in the concatenation.

**SYSIN**
This data set contains commands for Db2 HPU.

**SYSPRINT**
This ddname specifies the data set that receives the report from Db2 HPU.

**Restriction:** if you allocate SYSPRINT to a non-SYSOUT data set and your Db2 HPU step involves an unload run in native mode with a request to QUIESCE the related table space (QUIESCE YES option or no QUIESCE option and VU013/ULQSCE set to YES), the DISP parameter for this data set must be MOD. Otherwise, the part of the SYSPRINT written before the output of the QUIESCE utility misses from the SYSPRINT data set.

**HPUPRINT**
Allocate this optional ddname to get INZCI0559 row count/completion messages printed in this separate data set. This is a deprecated feature. Consider using the more flexible SELMSG feature instead.

**SYYSTEM**
This ddname is optional. It receives the additional diagnostic information from running Db2 HPU.

**SYSABEND**
This ddname is optional. Because the INZUTILB module runs in key 7, the standard dump mechanism does not apply. If an abend occurs, a dump is produced by using the SNAP macro in the SYSABEND ddname, if this ddname is allocated in the JCL. The VZM009 parameter defines a list of system codes, such as system codes for X37 abends, for which dumps are not produced.

Other dump ddnames, such as SYSUDUMP and SYMSMDUMP, are not used by Db2 HPU.

**UTOPRINT**
This ddname is optional. It specifies the output data set for sort utility messages. If the VUX020/SORTCLAS PARMLIB parameter is set to a nonblank value, the UTPRINT ddname is not used for the sort messages, and Db2 HPU dynamically allocates one ddname for the messages of each sort that is invoked.

**INFPLIB**
This ddname is conditional. It connects Db2 HPU to the PARMLIB that contains the INZUTIL member.

If the INZPARM member was customized and submitted during installation, you can omit this ddname.
**copydd**
This DD statement is optional. Specify a DD statement with a name that matches the value of *copydd* in the COPYDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of allocating a DD statement in the JCL.

This DD statement names the image copy data set from which the unload is to be done.

For nonpartitioned table spaces or to use a global image copy of a partitioned table space, allocate the COPYDD ddname in your JCL. You can also concatenate image copies for each partition under a single ddname to avoid using partition parallelism. See “Concatenation of image copies” on page 67 section.

To enable Db2 HPU to process partitioned table spaces in parallel by using image copies as input, use a 1- to 7-digit sequential number to specify one *copyddnnn* statement for each partition that you want to unload. The following example shows how you might specify this DD statement:

```
copydd01
copydd02
...copyddnnn...
```

where *nnn* is a 1- to 7-digit sequential number. Db2 HPU searches for all possible ddnames and partition numbers, with or without leading zeros. For example, if you specify COPYDDN CPY, for partition 1, Db2 HPU searches in the following order for CPY00001, CPY0001, CPY001, CPY01, and CPY1, and uses the first ddname that is found. Ddnames that are allocated for partitions can be different lengths, such as CPY1, CPY02, CPY003, and CPY0004.

**outdd**
This DD statement is optional. Specify a DD statement with a name that matches the value of *outdd* in the OUTDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of a DD statement that is allocated in the JCL.

This DD statement names the data set that will contain the result of a SELECT statement (logical unload).

**unldd**
This DD statement is optional. Specify a DD statement with a name that matches the value of *unldd* in the UNLDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of a DD statement that is allocated in the JCL.

This DD statement names the data set that will contain the physical unload of your table space.

**loaddd**
Specify a DD statement with a name that matches the value of *loaddd* in the LOADDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of a DD statement that is allocated in the JCL.

This DD statement names the data set that will contain the SYSIN DD for a load that allows a RELOAD of a SELECT statement in DSNTIAUL or VARIABLE format into the same or a different table.

**Restriction:** Depending on the type of format, Db2 HPU cannot generate the RELOAD of a SELECT statement if records of variable size that do not contain headers are detected. When this situation occurs, the following conditions exist:

- The DELIMITED format is supported only in Db2 Version 8 and later releases because the Db2 LOAD utility accepts this format.
- Except for the variable columns without headers, the DSNTIAUL, VARIABLE, USER, and EXTERNAL formats are supported. Db2 HPU does not generate a LOADDDN statement when you specify LENGTHBYTE NO and LENGTH REAL with variable columns in the USER format, an INTO clause, or a REFORMAT clause.

**dispparm**
This DD statement is optional. Specify a DD statement with a user-defined name - named dispparm hereafter - to enable the "Display implicit parameters" feature - driven either by the VUU083/DISPPARM parmlib parameter or the SYSIN DISPLAY_PARMLIB option from the Technical parameters.
options block - when the value chosen for this feature is IF,dispparm or by the value YES,dispparm. The default DCB attributes that Db2 HPU sets for this data set are RECFM=VB and LRECL=124. However, the useful part of the report is contained in the left 80 characters and this data set can be allocated with LRECL=80 without any lack of information.

**dispexpl**

This DD statement is optional. Specify a DD statement with a user-defined name - named dispexpl hereafter - to enable the "display of EXPLAIN report" feature - driven either by the VUU087/DISPEXPL parmlib parameter - when the value chosen for this feature is IF,dispexpl or by the value YES,dispexpl. Db2 HPU sets RECFM=VB and LRECL=124 as DCB attributes of this dataset, regardless of explicit DCB parameters specified through the DD statement.

**Related reference**

**TEMPLATE block syntax and description**

Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

**Db2 HPU Db2 parameters: VZM009**

The Db2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the Db2 parameters that are used by Db2 HPU.

**Db2 HPU SELECT syntax: OUTDDN**

The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

**Db2 HPU UNLOAD syntax: UNLDDN**

See this topic for descriptions of the keywords in the GLOBAL OPTIONS block syntax.

---

**Db2 HPU syntax**

The Db2 HPU syntax is divided into five major blocks: PROCESS OPTIONS, LISTDEF, LISTDEFTBV, TEMPLATE, GLOBAL OPTIONS, and UNLOAD. Only the UNLOAD block is required. All other major blocks are optional.

The following syntax diagram describes the Db2 HPU syntax and how to code the different major syntax blocks.

You can specify the PROCESS OPTIONS block, LISTDEF block, LISTDEFTBV block and TEMPLATE block either before the GLOBAL OPTIONS block or before each UNLOAD command. LISTDEF, LISTDEFTBV and
TEMPLATE definitions are processed sequentially and can be used in all UNLOAD commands that follow their definition.

You cannot redefine a TEMPLATE with a name that is already defined in the same SYSIN. However, you can redefine a TEMPLATE in the SYSIN that is defined in a TEMPLATE library.

The Db2 HPU parsers accepts comment lines that begin with two dashes (--) LISTEDFTBV block in column one.

Comment lines that begin with an asterisk (*) in column one are also accepted when they are coded before the first keyword of the SYSIN.

Topics:
- “PROCESS OPTIONS block syntax and description” on page 91
- “LISTDEF and LISTDEFTBV block syntax and description” on page 92
- “TEMPLATE block syntax and description” on page 97
- “GLOBAL OPTIONS block syntax and description” on page 113
- “UNLOAD block syntax and description” on page 120

PROCESS OPTIONS block syntax and description

Use the PROCESS OPTIONS control statement to change and restore the default ddnames that Db2 HPU uses for LISTDEF and LISTDEFTBV definition libraries and TEMPLATE definition libraries.

The PROCESS OPTIONS control statement corresponds to a subset of functions in the OPTIONS control statement in the Db2 utilities. To avoid confusing the OPTIONS keyword that exists in the Db2 HPU syntax with the OPTIONS control statement in the Db2 utilities, the equivalent keyword in Db2 HPU is PROCESS_OPTIONS.

The LISTDEFDD and TEMPLATEDD parameters can point to an existing LISTDEF and LISTDEFTBV definition library or to a TEMPLATE definition library that is used by the standard Db2 utilities. Any unsupported keywords in the LISTDEF and LISTDEFTBV definitions or TEMPLATE definitions are ignored.

You must specify at least one keyword after you specify PROCESS_OPTIONS. The following diagram shows the syntax of the PROCESS_OPTIONS block:

**PROCESS_OPTIONS block**

- PROCESS_OPTIONS
- LISTDEFDD `ddname`
- TEMPLATEDD `ddname`
- PATH_VARIABLES
- ON
- OFF

**LISTDEFDD `ddname`**

Specifies the `ddname` of the LISTDEF and LISTDEFTBV definition library. A LISTDEF and LISTDEFTBV definition library is a data set that contains only LISTDEF and/or LISTDEFTBV control statements. This data set is processed only when a referenced LIST is not found in the SYSIN.

The default value is SYSLISTD.

**TEMPLATEDD `ddname`**

Specifies the `ddname` of the TEMPLATE definition library. A TEMPLATE library is a data set that contains only TEMPLATE control statements. This data set is processed only when a referenced name does not exist in the job step as a `ddname` and is not found in the SYSIN as a TEMPLATE name.

The default value is SYSTEMPL.

**PATH_VARIABLES**

Specifies whether template variables are substituted in path names.

- **ON**
  - Specifies that template variables are substituted.
OFF
Specifies that template variables are not substituted. Specified path names are used as they are coded.

The default value is ON.

OFF
Specifies that all default options are to be restored. You cannot specify any other PROCESS_OPTIONS keywords if you specify PROCESS_OPTIONS OFF. Specifying PROCESS_OPTIONS OFF is equivalent to specifying OPTIONS LISTDEFDD SYSLISTD TEMPLATEDD SYSTEMPL PATH_VARIABLES ON.

Related reference
LISTDEF and LISTDEFTBV block syntax and description
Use the LISTDEF control statement to group table spaces into reusable lists and to specify these lists in UNLOAD commands or in SELECT statements to indicate that all the items in the lists are to be processed. Use the LISTDEFTBV control statement to group tables and/or views into lists and to specify these lists in SELECT statements to unload data from the selected tables and/or views. It differs from LISTDEF which behavior when a TABLE is specified is to retrieve its TABLESPACE and then unload the entire TABLESPACE.

TEMPLATE block syntax and description
Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

LISTDEF and LISTDEFTBV block syntax and description
Use the LISTDEF control statement to group table spaces into reusable lists and to specify these lists in UNLOAD commands or in SELECT statements to indicate that all the items in the lists are to be processed. Use the LISTDEFTBV control statement to group tables and/or views into lists and to specify these lists in SELECT statements to unload data from the selected tables and/or views. It differs from LISTDEF which behavior when a TABLE is specified is to retrieve its TABLESPACE and then unload the entire TABLESPACE.

The following diagram shows the syntax of the LISTDEF and LISTDEFTBV block.

LISTDEF and LISTDEFTBV block

```
LISTDEF  list-name  listdef-desc
         |                     |
         v                     v
LISTDEFTBV  list-name  listdeftbv-desc
```

listdef-desc

```
INCLUDE
EXCLUDE  type-spec
         |                     |
         v                     v
LIST          referenced-list-name
         |                     |
         v                     v
CLONED     YES    RI
            NO
listdef-initial-object-spec
```

listdeftbv-desc:

```
INCLUDE  listdeftbv-initial-object-spec
         |                     |
         v                     v
EXCLUDE
```

listdef-initial-object-spec
Attention: To share LISTDEF libraries with Db2 utilities, Db2 HPU accepts, but ignores, all other keywords that are not part of the LISTDEF block syntax.

Unlike Db2 utilities, the Db2 HPU implementation of the LISTDEF statement always returns a list of base table spaces. Because you cannot unload an index, index spaces are not selected by the LISTDEF statement. LOB table spaces are not selected by the LISTDEF statement. To unload LOB data, use a SELECT statement on the base table.

**LISTDEF list-name**
Defines a list of Db2 table spaces and assigns a name to that list. The list name makes the list available for subsequent processing as the object of an UNLOAD command or as an element of another LISTDEF statement.

Specify the name of the defined list. Valid values are 1 - 18 characters. You can put LISTDEF statements either in a separate LISTDEF library data set or before an UNLOAD command that refers to the list name.

**LISTDEFTBV list-name**
Defines a list of Db2 tables or views and assigns a name to that list. The list name makes the list available for subsequent processing as the object of a SELECT statement.

Specify the name of the defined list. Valid values are 1 - 18 characters. You can put LISTDEFTBV statements either in a separate LISTDEF library data set or before an UNLOAD command that refers to the list name.

**INCLUDE**
Specifies that the list of table spaces that results from the subsequent expression will be added to the list. You must first specify an INCLUDE clause. You can then specify subsequent INCLUDE or EXCLUDE clauses in any sequence to add clauses to or delete clauses from the existing list.

**EXCLUDE**
Specifies, after the initial INCLUDE clause, that the list of table spaces that results from the subsequent expression will be excluded from the list if the objects are in the list. If the table spaces are not in the list, they are ignored, and Db2 HPU processes the next INCLUDE or EXCLUDE clause.

**TABLESPACES**
Specifies that the INCLUDE or EXCLUDE clause will create a list of related table spaces. TABLESPACES is the default type for lists that use a table space or a table for the initial search. For lists that use other object types for the initial search, you must explicitly specify TABLESPACES. Db2 HPU can process only lists of table spaces.

**INDEXSPACES COPY YES|NO**
Specifies that the INCLUDE or EXCLUDE clause will create a list of related index spaces. INDEXSPACES is accepted only for compatibility with the syntax of the LISTDEF statement in the Db2 utilities. Db2 HPU cannot process lists of index spaces.

**LIST referenced-list-name**
Specifies the name of a previously defined object list that will be expanded and used for the initial search for the object. You must explicitly specify the referenced-list-name name. You cannot specify pattern-matching characters, such as %, *, ?, and _, for lists. You can use the LIST keyword to make aggregate lists of lists, to exclude entire lists from other lists, and to develop lists of objects that are related to other lists.

**DATABASE database-name**
Specifies the database that will be used for the initial search for the object. You can specify the database name explicitly or as a pattern-matched name. DATABASE * and DATABASE % are not supported. If you specify DATABASE, you must also specify either TABLESPACES or INDEXSPACES as the list type. Depending on the list type that you specify, Db2 HPU includes all table spaces or index spaces.
spaces in the database name that satisfy the pattern-matching expression in the list. You cannot specify DSNDB01, DSNDB06, DSNDB07, or user-defined work file databases in a LISTDEF definition.

**TABLESPACE database-name.table-space-name**

Specifies the table space that will be used for the initial search for the object. If you specify TABLESPACE, the default list type is TABLESPACES. All table spaces that satisfy the pattern-matching expression are included in the list unless the list is modified by other keywords. TABLESPACE *.* and TABLESPACE %.% are not supported.

Use `database-name` to specify the name of the database to which the table space belongs. The default value is DSNDB04. Use `table-space-name` to specify the name of the table space. You can explicitly specify `database-name, table-space-name` or both, or you can use pattern-matching characters to specify these values. You cannot include any objects in DSNDB07 or any user-defined work file databases in a LISTDEF definition. Pattern matching is not supported for DSNDB01 and DSNDB06 objects.

LOB and XML table spaces are not selected when they match a LISTDEF definition that was processed by Db2 HPU.

**INDEXSPACE database-name.index-space-name**

Specifies the index space that will be used for the initial object search. If you specify INDEXSPACE, the default list type is INDEXSPACES. All index spaces that satisfy the pattern-matching expression are included in the list unless the index spaces are excluded by other LISTDEF options. INDEXSPACE *.* and INDEXSPACE %.% are not supported.

Use `database-name` to specify the name of the database to which the index space belongs. The default value is DSNDB04. Use `index-space-name` to specify the name of the index space. You can explicitly specify `database-name, index-space-name` or both, or you can use pattern-matching characters to specify these values. You cannot include any objects in DSNDB07 or any user-defined work file databases in a LISTDEF definition. Pattern matching is not supported for DSNDB01 and DSNDB06 objects.

**TABLE creator-ID.table-name**

Specifies the table that will be used for the initial search for the object. If you specify TABLE with LISTDEF, the default list type is TABLESPACES, if you specify TABLE with LISTDEFTBV, the default list type is TABLES (Db2 objects TABLEs of type VIEW will be excluded).

All tables that satisfy the pattern-matching expression are included in the list unless the list is modified by other keywords. TABLE *.* and TABLE %.% are not supported.

Use `creator-ID` to specify the qualifier creator ID for the table. The default value is the user identifier for the utility. Use `table-name` to specify the name of the table. You can explicitly specify or use pattern-matching characters to specify `creator-ID, table-name`, or both.

However, the underscore (_) pattern-matching character is ignored in an table name. Pattern matching is not supported for catalog and directory objects. When you include a catalog and directory objects in a LISTDEF or LISTDEFTBV statement, you must use their fully qualified names. If the name contains a blank, enclose it in quotation marks.

**VIEW creator-ID.view-name**

Specify a view identifier. This option is allowed for LISTDEFTBV statements and not supported for LISTDEF statements. All views that satisfy the pattern-matching expression are included in the list unless the list is modified by other keywords. VIEW *.* and VIEW %.% are not supported.

Use `creator-ID` to specify the qualifier creator ID for the view. The default value is the user identifier for the utility. Use `view-name` to specify the name of the view. You can explicitly specify or use pattern-matching characters to specify `creator-ID, view-name`, or both. However, the underscore (_) pattern-matching character is ignored in a view name. Pattern matching is not supported for catalog and directory objects. When you include a catalog and directory objects in a LISTDEFTBV statement, you must use their fully qualified names. If the name contains a blank, enclose it in quotation marks.

**INDEX creator-ID.index name**

Specifies the index that will be used for the initial search for the object. If you specify INDEX, the default list type is INDEXSPACES. All index spaces that contain indexes that satisfy the pattern-
matching expression are included in the list unless the list is modified by other keywords. INDEX *.* and INDEX %.% are not supported.

Use creator-ID to specify the qualifier creator ID for the index. The default is the user identifier for the utility. Use index-name to specify the name of the index. You can explicitly specify creator-ID, index-name, or both, or you can use pattern-matching characters to specify these values. However, the underscore pattern-matching character is ignored in an index name. Pattern matching is not supported for catalog and directory objects. When you include a catalog and directory objects in a LISTDEF statement, you must use their fully qualified names. If the name contains a blank, enclose the index name in quotation marks.

**PARTLEVEL**

Specifies the partition granularity for partitioned objects. The integer partition number n is greater than or equal to zero (n ≥ 0). If you specify PARTLEVEL 0, the resulting list contains one entry for each non-partitioned object. If you specify PARTLEVEL with a nonzero operand, the resulting list contains one entry for the specified partition for partitioned objects and one entry for each non-partitioned object. If you specify PARTLEVEL without specifying the integer part number, the resulting list contains one entry for each partitioned object and one entry for each non-partitioned object. Using PARTLEVEL without specifying the integer part number n does not mean that data is unloaded per partition. You can remove an INCLUDE clause with the PARTLEVEL keyword from the list only by using an EXCLUDE clause with the PARTLEVEL keyword.

*integer*

Integer is the physical partition number where integer n ≥ 0.

If you specify PARTLEVEL 0, the resulting list contains one entry for each non-partitioned object.

If you specify PARTLEVEL with a nonzero operand, the resulting list contains one entry for each partition in the partitioned object and one entry for each non-partitioned object.

If you specify PARTLEVEL without specifying "integer", the resulting list contains one entry for each partitioned in the partitioned object and one entry for each non-partitioned object.

*integer1:integer2*

*integer1:integer2* indicates the physical partitions or a range of partitions to be specified in a list. The partition range must follow these guidelines:

- integer 1≥1
- integer 1< integer 2

An INCLUDE with the PARTLEVEL keyword can be removed from the list only by an EXCLUDE with PARTLEVEL.

**Partition processing with LISTDEF or LISTDEFTBV and PARTLEVEL**

A LISTDEF definition that includes the PARTLEVEL keyword without a partition number is supported only for compatibility purposes with the IBM LISTDEF utility and has no impact on how Db2 HPU unloads the data.

LISTDEF or LISTDEFTBV definitions such as these do not activate partition parallelism.

When Db2 HPU writes in a separate file per partition, partition parallelism is activated automatically. When Db2 HPU writes to a single output file, partition parallelism can be forced by using the PARALLELISM keyword of the UNLOAD command or by setting the VUU036/GBLPARAL PARMLIB parameter to YES. In Db2 HPU, when the list that is generated by a LISTDEF or LISTDEFTBV control statement contains a partitioned table space, and if the DSNAME that is defined in the TEMPLATE statement contains the &PART variable, the unloaded rows are written in a separate file for each partition. Otherwise, all rows that are unloaded from the partitioned table space are written in a single file. This condition applies to output files that are generated by the UNLDDN or OUTDDN keywords. When the list that was generated by a LISTDEF or LISTDEFTBV control statement is used, the PART keyword in the UNLOAD command is ignored, and the PART keyword in the SELECT statement is not valid. You can select partitions by using the PARTLEVEL keyword in the LISTDEF or LISTDEFTBV statement. All partitions of the same object that are defined in the list that was created by the LISTDEF are processed together. The unloaded rows are written in the file that was allocated by the TEMPLATE with these limitations, depending on the usage of the &PART variable.
When you specify DB2 FORCE and you use the PARTLEVEL keyword in a LISTDEF statement to select partitions, the selected partitions can be used only to select table spaces. When you specify DB2 FORCE to select a partitioned table space, the complete table space is unloaded in a single unload file.

When you specify DB2 FORCE and you use the PARTLEVEL keyword in a LISTDEFTBV statement to select partitions, the complete table is unloaded.

CLONED
This option is supported only with a LISTDEF statement. Specifies that the INCLUDE or EXCLUDE clauses will return only the names of cloned tables, table spaces that contain cloned tables, indexes on cloned tables, or index spaces that contain indexes on cloned tables. If you specify CLONED, other keywords in the LISTDEF statement that refer to related objects refer to the clones of those objects. If you also specify a table name, the CLONED keyword is ignored.

RI
Specifies that all objects that are referentially related to the object expression (PRIMARY KEY <-> FOREIGN KEY) will be included in the list. Db2 HPU processes all referential relationships repeatedly until the entire referential set is developed. You cannot specify RI with PARTLEVEL(n).

Related reference
PROCESS OPTIONS block syntax and description
Use the PROCESS OPTIONS control statement to change and restore the default ddnames that Db2 HPU uses for LISTDEF and LISTDEFTBV definition libraries and TEMPLATE definition libraries.

Partition processing with LISTDEF and PARTLEVEL

LISTDEF specification
When you specify LISTDEF, the LISTDEF that you specify depends on whether the unload is a physical unload or a logical unload.

For physical unloads (UNLDDN), specify LIST(list-name) in the UNLOAD TABLESPACE command instead of specifying the table space name. A physical unload is generated for each table space that is contained in the generated list of table spaces. If the UNLDDN uses a template, a separate data set is allocated for each table space that was unloaded.

For logical unloads (SELECT + OUTDDN), specify LIST(list-name) in the FROM clause of the SELECT statement, and do not specify a table space name in the UNLOAD TABLESPACE command. A SELECT statement is generated for each table of each table space in the list. If a template is used for the OUTDDN and for the LOADDNN, a separate file is dynamically allocated for each table of each table space in the generated list.

LISTDEFTBV specification
LISTDEFTBV is only usable with logical unloads (SELECT+OUTDDN). Specify LISTTBV(list-name) in the FROM clause of the SELECT statement, and do not specify a table space name in the UNLOAD TABLESPACE command. A SELECT statement id generated for each table of each table space in the LIST. If a template is used for the OUTDDN and for the LOADDNN, a separate file is dynamically allocated for each table or view in the generated list.

TEMPLATE block syntax and description
Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

You can use the TEMPLATE control statement when you process a LISTDEF or LISTDEFTBV list, but the statement can also be used with standard UNLOAD commands. In its simplest form, the TEMPLATE statement defines the data set naming convention. You can also write TEMPLATE statements so that they contain allocation parameters that define data set size, location, and attributes.

The TEMPLATE statement uses the z/OS DYNALLOC macro (SVC 99) to perform data set allocation. Therefore, the facility is constrained by the limitations of this macro and by the subset of DYNALLOC that is supported by TEMPLATE.
The following diagram shows the syntax of the TEMPLATE block:

**TEMPLATE block**

```
 TEMPLATE  →  template-name  ←  DSN  →  name-expression  ←  SUBSYS  →  name  ←
                 path-expression  ↓
                         ┌───────┐
                         │      │
                         │      │
                         └──────┘

 LRECL  →  integer  ←  RECFM  →  F
                   │
                   │
                   │
                   └── common-options

 disk-options
 tape-options

 name expression

 qualifier-expression

 (parenthetical-expression)

 path expression

 PATH  →  path-name  ←

 FILEDATA RECORD  →  RECFM VB  ←  LRECL 32756

 FILEDATA

 TEXT

 BINARY

 RECORD

 V

 FB

 F

 LRECL  →  integer

 PATHOPTS

 (ORDONLY
  OCREAT
  OWRONLY
  ONONBLOCK)

 PATHMODE

 (SIRUSR
  SIWUSR
  SIXUSR
  SIRGRP
  SIWGRP
  SIXGRP
  SIRWXU
  SIWXU
  SIRWXG
  SIWXG
  SIROTH
  SIWOTH
  SIXOTH
  SIROTH)

 PATHDISP

 (KEEP
  DELETE
  KEEP
  DELETE)

 qualifier expression
Notes:

1 SUBSYS and path-expression are mutually exclusive.
2 The entire name expression represents one character string and cannot contain any blanks.
3 If you use substring notation, the entire DSN operand must be enclosed in single quotation marks, such as 'P&PA(4,2).
4 If a TAPEUNIT specification (defined either by a TAPEUNIT option or the TAPEUNIT parmlib parameter) applies to the UNIT specified by the UNIT suboption of the TEMPLATE statement, the default_devices_value is used as a default UNCNT value. For allocation of sequential data sets on disk, i.e. when no TAPEUNIT specification applies to the UNIT specified by the TEMPLATE statement, where MAXPRIME is specified and no VOLUME is specified, the value of UNCNT is calculated to allow the data
set to be written over several volumes according to the estimated amount data it must store. If none of
the previous condition is fulfilled, a value of 1 is used for UNCNT.

**disk options**

- **SPACE**
  - primary
  - secondary

- **CYL**
  - TRK
  - MB

- **PCTPRIME**
  - 100

- **MAXPRIME**
  - integer

- **NBRSECND**
  - 10

- **DIR**
  - integer

- **DSNTYPE**
  - LIBRARY
  - PDS
  - HFS
  - NULL
  - PIPE
  - BASIC
  - MARGE
  - EXREQ
  - EXTPREF

- **KEYLABEL**
  - key label name

**tape options**

- **STACK**
  - NO
  - YES

- **TRTCH**
  - NONE
  - COMP
  - NOCOMP

**Important**: To share TEMPLATE libraries with other Db2 utilities, Db2 HPU accepts, but ignores, all
keywords that are not specified in the syntax diagram.

**TEMPLATE template-name**

 Defines a data set allocation template and assigns a name to the template for subsequent reference
in an UNLOAD command. The name is followed by keywords that control the allocation of tape and
disk data sets. You cannot specify both disk options and tape options in the same TEMPLATE
statement. The UNIT keyword specifies a generic unit name that is already defined on your system.

Valid values are 1 - 8 alphanumeric characters and must begin with an alphabetic character.
**DSN name-expression**

Specifies the TEMPLATE for the z/OS data set name. You can specify the data set name by using symbolic variables, non-variable alphanumeric or national characters, or any combination of these characters. The name must adhere to the z/OS data set naming rules, including those rules about name length, valid characters, name structure, and qualifier length.

Data set names consist of a series of qualifiers and qualifier expressions that are separated by a period and an optional expression in parentheses. Embedded blanks are restricted. If the DSN name operand contains any special characters, it must be enclosed in single quotation marks. For example, in the following TEMPLATE statement, the DSN operand contains parentheses as special characters, so the entire operand is enclosed in single quotation marks:

```
TEMPLATE X DSN 'A.GDG.VERSION(+1)'
```

Parentheses around the DSN name operand are optional. The following DSN specification shows a DSN name operand in parentheses:

```
DSN(&DB..&TS..D&DATE.)
```

**PATH path-name**

Specifies a z/OS UNIX System Services (USS) file path name. This path name can be a USS pipe path name. The path name must be in single-byte EBCDIC format and must not exceed 255 bytes. If the path name contains blanks, you must enclose it in single quotation marks.

Db2 HPU can use a template with the PATH keyword only for the following types of files:

- UNLDDN for output of physical unloads
- OUTDDN for output of logical unloads
- LOADDDN for the LOAD SYSIN for logical unloads
- DDLDDN to specify the DDL for unloading from an image copy
- A template that is used to define a LOB file reference when LOB or XML data is unloaded

You can use a variable name for a complete directory name or part of a directory name. Before the files are generated, Db2 HPU ensures that the directory exists. Db2 HPU does not create the directory. If the directory does not exist, Db2 HPU issues an error.

Syntax rules regulate how Db2 HPU interprets variable names. When you use an ampersand (&), Db2 HPU interprets the ampersand as the introduction to a variable name. To generate an ampersand in the path name, you must use a backslash (\) as an escape character before the ampersand. In the following example, &TS. is coded as a variable name, and &SPECIAL is coded as regular text.

```
PATH '/u/&TS.\&SPECIAL'
```

After variables have been replaced, the following path name is generated:

```
/u/TS01&SPECIAL
```

To create a USS pipe file, you must specify DSNTYPE PIPE in the TEMPLATE statement.

**FILEDATA**

Specifies the content type of a z/OS USS file. Valid values are TEXT, BINARY, and RECORD.

The default value is RECORD, and RECFM VB and LRECL 32756 are assumed.

**RECFM**

Specifies the record format of the z/OS USS file. The RECFM option is required when FILEDATA is specified. The following values are valid:

- F (Fixed)
- FB (Fixed block)
- V (Variable)
• VB (Variable block)

If FILEDATA is not specified, RECFM VB is assumed with FILEDATA RECORD LRECL 32756.

LRECL
Specifies the record length of the z/OS USS file. The LRECL option is required when FILEDATA is specified.

If FILEDATA is not specified, LRECL 32756 is assumed with FILEDATA RECORD RECFM VB.

PATHOPTS
Specifies the access and status for the z/OS USS file that is named in the PATH parameter. The following values are valid:

ORDONLY
Specifies that the utility will open the file only for read access.

OCREAT
Specifies that files are created based on the following conditions:

• If the file does not exist, the system creates it.
• If a directory that is specified in the path name does not exist, the directory and the new file are not created.
• If the file exists, the existing file is used by the utility.

OWRONLY
Specifies that the utility will open the file only for write access.

ONONBLOCK
Specifies that an open() function will return or block (wait) based on conditions for the following file types:

For a first-in first-out (FIFO) special file, such as a USS pipe file:

• If ONONBLOCK is specified with ORDONLY access, an open() function for only reading returns a file descriptor (handler) without delay.
• If ONONBLOCK is not specified with ORDONLY access, an open() function for only reading waits until a process opens the file for writing.
• If ONONBLOCK is specified with OWRONLY access, an open() function for only writing returns an error if a process does not currently have the file open for reading.
• If ONONBLOCK is not specified with OWRONLY access, an open() function for only writing waits until a process opens the file for reading.

For a character special file that supports nonblocking open:

• If ONONBLOCK is specified, an open() function returns a file descriptor without waiting until the device is ready or available. The device response depends on the type of device.
• If ONONBLOCK is not specified, an open() function waits until the device is ready or available.

Specifying ONONBLOCK does not affect other file types.

The default values are OCREAT and OWRONLY.

PATHMODE
Specifies the file mode of the HFS file. The following values are valid:

SIRUSR
Allows the file owner to read the file.

SIWUSR
Allows the file owner to write the file.

SIXUSR
If the file is a directory, allows the file owner to search; if the file is not a directory, allows the file owner to run the program in the file.
SIRWXU
If the file is a directory, allows the file owner to read, write, and search; if the file is not a directory, allows the file owner to read, write, and run the file. Specifying SIRWXU is equivalent to specifying SIRUSR, SIWUSR, and SIXUSR together.

SIRGRP
Allows users in the file group class to read the file.

SIWGRP
Allows users in the file group class to write the file.

SIXGRP
If the file is a directory, allows users in the file group class to search; if the file is not a directory, allows users in the file group class to run the program in the file.

SIRWXG
If the file is a directory, allows users in the file group class to read, write, and search; if the file is not a directory, allows users in the file group class to read, write, and run the file. Specifying SIRWXG is equivalent to specifying SIRGRP, SIWGRP, and SIXGRP together.

SIROTH
Allows users in the file other class to read the file.

SIWOTH
Allows users in the file other class to write the file.

SIXOTH
If the file is a directory, allows users in the file other class to search; if the file is not a directory, allows users in the file other class to run the program in the file.

SIRWXO
If the file is a directory, allows users in the file other class to read, write, and search; if the file is not a directory, allows users to read, write, and run the file. Specifying SIRWXO is equivalent to specifying SIROTH, SIWOTH, and SIXOTH together.

The default value is SIRUSR.

PATHDISP
Specifies the disposition of a z/OS USS file when the job step ends. Valid values are KEEP and DELETE. The MVS system default is KEEP, KEEP.

SUBSYS name
Specifies the MVS BATCHPIPES SUBSYSTEM name. The specified name must be a valid BATCHPIPES SUBSYSTEM name and must be 1 - 8 characters. When SUBSYS is specified, LRECL and RECFM are required, and TEMPLATE keywords that are not compatible with SUBSYS, such as UNIT, are ignored.

Restriction: When you use BATCHPIPES and TEMPLATE with the SUBSYS keyword, the utility cannot be restarted, and the LOAD DISCARDDN keyword is not supported.

LRECL integer
Specifies the record length of the MVS BATCHPIPES SUBSYSTEM file. LRECL is required when SUBSYS is specified.

RECFM
Specifies the record format of the MVS BATCHPIPES SUBSYSTEM file. RECFM is required when SUBSYS is specified. Valid values are F, FB, V, or VB.

character-expression
Specifies the data set name or part of the data set name by using nonvariable alphanumeric or national characters.

&variable
Specifies the data set name or part of the data set name by using nonvariable alphanumeric or national characters.

In addition to the list of variables, you can use user-defined variables in the DSN expression. You set the values for these user-defined variables by using the TEMPLATESET keyword in the OPTIONS block. Syntax rules for such user-defined variables are identical to the rules for standard variables. To
avoid duplicate data set names when you specify a TEMPLATE statement with a LISTDEF and LISTDEFTBV list that contains a multi-table table space, specify TEMPLATESET and use a user-defined variable that corresponds with the :SELNUM predefined variable in the DSN expression.

Each symbolic variable is substituted with its related value at run time to form a specific data set name. When you use substitution variables in a DSN expression, they begin with an ampersand sign (&) and end with a period (.), as shown in the following example:

```
DSN &DB..&TS..&JDATE..&PART.
```

If you use only numeric variables, an invalid data set qualifier for all numeric-type variables, such as all date or time-type variables, &SEQ, and &PART, is generated. These variables must be preceded by character constants to form valid DSN qualifiers. The following specifications are valid:

```
&PART.
&D&DATE.
```

Some variables inherit default values when their values are not known. For example, the default value of &PART becomes 00000 for nonpartitioned objects.

You can also use substring notations for the data set name. This notation can prevent the data set name from exceeding the 44-character maximum. If you use substring notation, you must enclose the entire DSN operand in single quotation marks. To specify a substring, use the form &variable(start), or &variable(start,length).

The following table contains a list of JOB variables and their descriptions:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;JOBNAME. or &amp;JO.</td>
<td>The z/OS job name.</td>
</tr>
<tr>
<td>&amp;STEPNAME. or &amp;ST.</td>
<td>The z/OS step name. This variable might be needed if data set names from two different job steps conflict.</td>
</tr>
<tr>
<td>&amp;USERID. or &amp;US.</td>
<td>The user ID of the person who is running Db2 HPU. Valid values are 1 - 8 characters.</td>
</tr>
<tr>
<td>&amp;UTILID. or &amp;UT.</td>
<td>The utility ID truncated to eight characters and checked for invalid DSN characters.</td>
</tr>
<tr>
<td>&amp;SSID. or &amp;SS.</td>
<td>Db2 identifier against which Db2 HPU is run. This variable is resolved taking the value of the subparameter TEMPLATE_SSID_SUBSTITUTION of the VUU030/ULOPTNS parmlib parameter into account.</td>
</tr>
<tr>
<td></td>
<td>• if TEMPLATE_SSID_SUBSTITUTION is set to MEMBER_NAME or is not specified the &amp;SSID or &amp;SS variable is substituted with the current Db2 member name to which Db2 HPU has connected.</td>
</tr>
<tr>
<td></td>
<td>• if TEMPLATE_SSID_SUBSTITUTION is set to GROUP_NAME the &amp;SSID or &amp;SS variable is substituted with theDb2 member name in a non datasharing environment and with the group attachment name for datasharing environment.</td>
</tr>
</tbody>
</table>

The following table contains a list of UTILITY variables and their descriptions:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;UTILNAME. or &amp;UN.</td>
<td>The utility name. The value is set to Db2 HPU.</td>
</tr>
<tr>
<td>&amp;SEQ. or &amp;SQ.</td>
<td>The sequence number of the list item.</td>
</tr>
</tbody>
</table>
The following table contains a list of OBJECT variables and their descriptions:

**Table 16. OBJECT variables and descriptions**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;LIST. or &amp;LI.</td>
<td>The name of the list that is defined by using the LISTDEF and LISTDEFTBV control statement and that is referenced on the same control statement as this TEMPLATE.</td>
</tr>
<tr>
<td>&amp;DB.</td>
<td>The database name.</td>
</tr>
<tr>
<td>&amp;TS.</td>
<td>The table space name.</td>
</tr>
<tr>
<td>&amp;SN.</td>
<td>The table space name.</td>
</tr>
<tr>
<td>&amp;PART. or &amp;PA.</td>
<td>The five-digit partition number, padded with leading zeros.</td>
</tr>
</tbody>
</table>

**Notes:**

- To generate one output file per partition, use the &PA. variable when you process LISTDEF and LISTDEFTBV lists that include partitioned table spaces.
- When you specify the &TS. or &SN. variables in a TEMPLATE statement that is used by an UNLOAD statement with BLOBF, CLOBF, or DBCLOBF columns, Db2 HPU substitutes the name of the table space that stores the LOB column value, not the base table space name. This substitution enables Db2 HPU to generate unique data set names for each LOB column with partitioned table spaces.
- When the SELECT statement in which the TEMPLATE is used is processed by using SQL (DB2 FORCE or DB2 YES with an unsupported SELECT statement), the &PART variable is set to one of the following values based on whether SQLPART is used:
  - The partition number when the SQLPART keyword is used,
  - 00000 when the SQLPART keyword is not used.

The &DB, &TS, and &SN variables are set to DBn, TSn, LOBn or XMLn, depending on the table space type, where n is a sequential number. This sequential number ensures unity, but it does not have a specific meaning regarding the actual database or the table space that it represents.

- In Db2 HPU V3.1, when a partitioned table space was unloaded from a global full image copy (FIC) that used a template with a DSNAME that contains the &PART variable, a single output file with partition number set to 00000 was generated. In Db2 HPU V3.2, an output file per partition with the &PART variable set to the partition number is generated.

The following table contains a list of DATE and TIME variables and their descriptions:

**Table 17. DATE and TIME variables and descriptions**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE. or &amp;DT.</td>
<td>YYYYDDD</td>
</tr>
<tr>
<td>&amp;TIME. or &amp;TI.</td>
<td>HHMMSS</td>
</tr>
<tr>
<td>&amp;JDATE. or &amp;JU.</td>
<td>YYYYDDD</td>
</tr>
<tr>
<td>&amp;YEAR. or &amp;YE.</td>
<td>YYYY portion of &amp;DATE.</td>
</tr>
<tr>
<td>&amp;MONTH. or &amp;MO.</td>
<td>MM</td>
</tr>
<tr>
<td>&amp;DAY. or &amp;DA.</td>
<td>DD</td>
</tr>
<tr>
<td>&amp;JDAY. or &amp;JD</td>
<td>DDD portion of &amp;DATE.</td>
</tr>
<tr>
<td>&amp;HOUR. or &amp;HO.</td>
<td>HH portion of &amp;TIME.</td>
</tr>
<tr>
<td>&amp;MINUTE. or &amp;MI.</td>
<td>MM portion of &amp;TIME.</td>
</tr>
</tbody>
</table>
Table 17. DATE and TIME variables and descriptions (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;SECOND. or &amp;SC.</td>
<td>SS portion of &amp;TIME.</td>
</tr>
<tr>
<td>&amp;UNIQ. or &amp;UQ.</td>
<td>Unique eight characters that Db2 HPU derives from the system clock. This set of characters begins with an alphabetic character and is followed by seven alphabetic or numeric characters.</td>
</tr>
</tbody>
</table>

Attention: All date and time values are set by using the STCK instruction and reflect the date and time value in Greenwich Mean Time (GMT). DATE and TIME values are captured during the initialization phase of Db2 HPU and remain constant during the STEP.

**start**

Specifies the substring starting byte location within the current variable base value at run time.

**length**

Specifies the length of the substring. If you specify start but do not specify length, the default value for length is the number of characters from the start character to the last character of the variable value at run time. For example, given a five-digit base value, &PART(4) specifies the fourth and fifth digits of the value. The length value must be an integer that does not cause the substring to extend beyond the end of the base value.

**parenthetical-expression**

Specifies part of the data set name by using nonvariable alphanumeric or national characters that are enclosed in parentheses. For example, the following expressions are valid: Q1.Q2.Q3(member) and Q1.Q2.Q3(+1).

**UNIT**

Specifies the device number, device type (generic), or group name for the data set. The default value is SYSALLDA.

**MODELDCB dname**

Specifies the name of the data set on which the TEMPLATE statement is based. DCB information is read from this model data set.

When the template corresponds to a GDG, MODELDCB uses the VUX033/TMPLDSCB PARMLIB parameter as the default value.

**BUFNO integer**

Specifies the number of BSAM buffers. Valid values are 0 - 99. The default value is 30.

**DATACLAS name**

Specifies the SMS data class. The name value must be a valid SMS data class and must be 1 - 8 characters. If you specify DATACLAS, the data set is cataloged. If you do not specify DATACLAS, no SMS DATACLAS is specified.

**MGMTCLAS name**

Specifies the SMS management class. The name value must be a valid SMS management class and must be 1 - 8 characters. If you specify MGMTCLAS, the data set is cataloged. If you do not specify MGMTCLAS, no MGMTCLAS is specified to SMS.

**STORCLAS name**

Specifies the SMS storage class. The name value must be a valid SMS storage class and must be 1 - 8 characters. If you specify STORCLAS, the data set is cataloged. If you do not specify STORCLAS, no STORCLAS is specified to SMS.

**RETPD integer**

Specifies the retention period in days for the data set. Valid values are 0 - 9999. If you specify DATACLAS, MGMTCLAS, or STORCLAS, the class definition might control the retention. RETPD cannot be specified with EXPDL.

**EXPDl 'date'**

Specifies the expiration date for the data set, in the form YYYYDDD, where YYYY is the 4-digit year, and DDD is the 3-digit Julian day. The ‘date’ value must be enclosed in single quotation marks. If you
specify DATACLAS, MGMTCLAS, or STORCLAS, the class definition might control the retention. EXPDL cannot be specified with RETPD.

**VOLUMES (volume-1,volume-2,...)**

Specifies a list of volume serial numbers for this allocation. The specified number of volumes cannot exceed the specified value or default value of the VOLCNT keyword. The first volume must contain enough space for the primary space allocation. If an individual volume serial number contains leading zeros, it must be enclosed in single quotation marks.

**VOLCNT (integer)**

Specifies the maximum number of volumes that an output data set might require. Valid values are 0 - 255.

Unless a tape storage class is used, the default value for tape templates is 95. If a tape storage class is used, the default value is the value that is set by the tape storage class.

Db2 HPU does not set a default value for disk templates. Operating system defaults apply.

**UNCNT**

Specifies the number of devices that are to be allocated. The valid values are 0 - 59. If UNIT specifies a specific device number, the value of UNCNT must either be 1, or it must be omitted. If UNCNT is not specified the default value is determined as follows. If a TAPEUNIT specification (defined either by a TAPEUNIT option or the TAPEUNIT parmlib parameter) applies to the UNIT specified by the UNIT suboption of the TEMPLATE statement, the default_devices_value is used as a default UNCNT value. For allocation of sequential data sets on disk, i.e. when no TAPEUNIT specification applies to the UNIT specified by the TEMPLATE statement, where MAXPRIME is specified and no VOLUME is specified, the value of UNCNT is calculated to allow the data set to be written over several volumes according to the estimated amount of data it must store. If none of the previous condition is fulfilled, a value of 1 is used for UNCNT.

**GDGLIMIT (integer)**

Specifies the number of entries that are to be created in a GDG base if a GDG DSN is specified and the base does not exist. If a GDG base does not exist and you do not want to define one, specify GDGLIMIT (0). Valid values are 0 - 255.

The default value is 99.

**LIMIT**

Specifies that the template switching feature applies, i.e. that an alternate TEMPLATE is to be used when the estimated amount of data to unload exceeds a given size.

**Important:** This keyword is taken into account only if TEMPLATE_LIMIT(YES) is specified in the setting for the VUU030/ULOPTNS parmlib parameter.

The following details must be specified. The output data set size taken into account is an estimate. The estimation method, thus its result, depends on the setting for the VUX036/RTSESTIM parmlib.

- \( n \) Specifies the maximum size of the output data set that is permitted using this TEMPLATE. This number is expressed in the unit specified by one of the following keywords:
  - CYL: Cylinders
  - GB: Gigabytes
  - MB: Megabytes

- \( new\_template \) Specifies a character string that specifies the name of a TEMPLATE to use if the size limit is exceeded.

**Restriction:**

- You cannot switch to a DD card.
- The template control statement that LIMIT references must exist in SYSIN or SYSTEMPL and it cannot refer to itself. If the reference contains also a LIMIT statement, the limit size must be greater than the referee.
• You cannot use LIMIT for LOADDDN file, LOB or XML file.

• Data set names patterns specified in the templates chained by the template switching feature must be consistent regarding the &PART. variable: if one template of among the related templates specify the &PART. variable, all the other templates must specify a pattern with the &PART. variable as well.

• If the SPACE option is specified, the LIMIT option is ignored.

TIME
Specifies time used in expansion of date and time DSN variables. The default TIME value is determined by the TEMPLATE_TIME subsystem parameter if USE_ADMIN_INFO_SYSPARM(YES) is specified for the VUU030/ULOPTNS parameter. Otherwise, UTC applies.

LOCAL
Use local time at the DB2 server in the expansion of date and time in DSN variables.

UTC
Use Coordinated Universal Time (UTC) in the expansion of date and time in DSN variables.

DISP (status, normal-termination, abnormal-termination)
Specifies the data set disposition. You must specify values for all of the following variables:

status
Specify one of the following standard z/OS values: NEW, OLD, SHR, and MOD.

normal-termination
Specify one of the following standard z/OS values: DELETE, KEEP, CATLG, and UNCATLG.

abnormal-termination
Specify one of the following standard z/OS values: DELETE, KEEP, CATLG, and UNCATLG.

Default values for the disposition are NEW, CATLG, CATLG for output files for TEMPLATE statements that UNLDDN, OUTDDN, or LOADDDN statements, and SHR, KEEP, KEEP for input files for TEMPLATE statements that use COPYDDN or DDLDDN statements. When a PDS or PDSE is created by a TEMPLATE statement, its default disposition is NEW, CATLG, CATLG. When new members are created in the PDS or PDSE, the disposition is SHR, KEEP, KEEP.

SPACE (primary, secondary)
Specifies the z/OS disk space allocation parameters. If you specify (primary, secondary) values, these values are used instead of the values that are calculated Db2 HPU. When specifying primary and secondary quantities, you must either specify both values or omit both values. Use the MAXPRIME option to set an upper limit on the primary quantity.

Valid values are 1 - 1677215

CYL
Specifies that allocation quantities, if present, are to be expressed in cylinders and that allocation is to occur in cylinders. If SPACE CYL is specified, without (primary, secondary), the Db2 HPU-calculated quantities are allocated in cylinders by using 3390 quantities for byte conversion.

TRK
Specifies that, in the absence of values for (primary, secondary), the Db2 HPU-calculated quantities are to be allocated in tracks by using 3390 disk drive quantities for byte conversion. If the amount calculated is greater than one cylinder, the TRK keyword is ignored, and the data set is allocated in cylinders (CYL).

MB
Specify that if allocation quantities are present, they are to be expressed in megabytes, and that allocation is to occur in records. One MB is 1048576 bytes. If SPACE MB is specified without (primary, secondary), the Db2 HPU-calculated quantities are allocated in records by using the average record length for the data set.

The default value is CYL.

Consider the following settings and keywords for default SPACE values for allocating NEW data sets:
Default SPACE values for allocating NEW data sets

**DB2 NO or DB2 YES with supported SELECT statements**
When Db2 HPU uses a TEMPLATE statement to allocate an output file (using UNLDDN or OUTDDN), the space that is allocated is calculated by using the catalog statistics for the unloaded object. If there are no statistics, Db2 HPU uses the physical characteristics of the underlying data set to compute the space for the output files. If necessary, you can override the calculated space allocation by using the SPACE parameter of the TEMPLATE definition.

**DB2 FORCE or DB2 YES with unsupported SELECT statements**
Use the SPACE keyword with specification of primary and secondary allocation in the TEMPLATE statement, or an error message is issued.

**RLSE keyword**
The RLSE (release) keyword is always used to allocate new data sets so that unused space is released on deallocation except when the allocated data set is a PDS or a PDSE (using DSNTYPE PDS or LIBRARY parameters).

**PCTPRIME integer**
Specifies the percentage of the estimated required space that will be obtained as the primary quantity. Use the MAXPRIME keyword to set the upper limit of this value for large objects.

The default value is 100.

**MAXPRIME integer**
Specifies the maximum allowable primary space allocation. The value is expressed in cylinders (CYL). This value constrains the primary space value, the PCTPRIME calculation, and the size of each secondary allocation.

**NBRSECND integer**
Specifies the division of secondary space allocations. After the primary space is allocated, an amount of space equal to the estimated required space is divided into the specified number of secondary allocations. Valid values are 1 - 10.

The default value is 10.

**DIR integer**
Specifies the number of 256-byte records that are to be allocated for the directory of a new partitioned data set. If you are allocating a new partitioned data set, you must specify this keyword. If the TEMPLATE is being used in an UNLOAD statement with BLOBF, CLOBF, or DBCLOBF, and you specify a DSNTYPE of LIBRARY or PDS but you do not specify DIR, Db2 HPU calculates the number of 256-byte records to allocate by dividing the estimated number of records by 20.

**DSNTYPE**
Specifies the type of data set to be allocated.

- **LIBRARY**
  Specifies that a partitioned data set extended (PDSE) will be allocated.

- **PDS**
  Specifies that a partitioned data set (PDS) will be allocated.

- **HFS**
  Specifies that a hierarchical file system (HFS) file will be allocated. An HFS file is allowed only when a TEMPLATE is used to allocate a LOB file reference (CLOBF, BLOBF, or DBCLOBF columns).

- **NULL**
  Specifies a null file. Use this value for a TEMPLATE statement with UNLOAD CLOBF, BLOBF, or DBCLOBF columns to unload a null LOB value. In this case, the unload data set contains a null file name.

- **PIPE**
  Specifies a USS pipe file. Specify PIPE only when you allocate a new pipe output file by using the PATH option in the TEMPLATE statement. You cannot use DSNTYPE PIPE when you allocate input files, existing pipe files, standard z/OS files with the DSN option, or a LOB file reference.
BASIC
Specifies that a basic data set will be allocated. Basic format data sets are sequential data set that are specified as neither extended-format nor large-format. Basic format data sets have a size limit of 65,535 tracks (4,369 cylinders) per volume.

LARGE
Specifies that a large data set will be allocated. Large format data sets are sequential data sets that can grow beyond 65,535 tracks (4,369 cylinders) up to 16,777,215 tracks per volume.

EXTREQ
Specifies that an extended format data set will be allocated. The allocation should fail if that is not possible.

EXTPREF
Specifies that allocation is preferred to be in extended format, but if the necessary system resources for extended are not available, then the data set should be allocated as BASIC.

If you omit DSNTYPE, the type of data set is determined by other data set attributes, the data class for the data set, or an installation default.

KEYLABEL key-label-name
Specifies the key label for the allocation of the data set. If the security administrator has specified a key label for RACF data set profile, that key label takes precedence over the DB2 provided key label.

STACK
Specifies whether output data sets are to be stacked contiguously on the same tape volumes.

NO
Specifies that output data sets are not stacked contiguously on tape.

YES
Specifies that output data sets are stacked as successive files on one tape volume. A logical tape volume can consist of an aggregate of more than one volume.

Stacking is allowed only for the output data sets from a physical unload (UNLDDN) or a logical unload (OUTDDN). When you unload LOB or XML data, stacking output files on tape is not supported for CLOBF, DBCLOBF, or BLOBF columns.

TRTCH
Specifies the track recording method for magnetic tape drives that have improved data recording capability.

NONE
Specifies that a track recording method is not specified during dynamic allocation.

COMP
Specifies that data is written in compacted format.

NOCOMP
Specifies that data is written in standard format.

See MVS JCL Reference for more information about the PATHOPTS and PATHMODE options.

Related reference
User-allocated ddnames
To run unload jobs, you must allocate certain ddnames in the Db2 HPU JCL.

PROCESS OPTIONS block syntax and description
Use the PROCESS OPTIONS control statement to change and restore the default ddnames that Db2 HPU uses for LISTDEF and LISTDEFTBV definition libraries and TEMPLATE definition libraries.

How to use TEMPLATES
As an alternative to using JCL to specify the data sets, you can use the TEMPLATE control statement to dynamically allocate Db2 HPU data sets.

**How to use TEMPLATES**

As an alternative to using JCL to specify the data sets, you can use the TEMPLATE control statement to dynamically allocate Db2 HPU data sets.

You can specify the following information with the TEMPLATE statement:

- The data set naming convention
- DFSMS parameters
- Disk allocation parameters

By using TEMPLATE statements, you can standardize data set allocation, which reduces the need to customize and alter Db2 HPU job streams.

You can specify a TEMPLATE statement in the SYSIN data set, preceding the UNLOAD command that references the TEMPLATE statement, or in a TEMPLATE library.

A TEMPLATE library is a data set that contains only TEMPLATE statements. You can specify a TEMPLATE data set DD name by using the TEMPLATEDD keyword in the PROCESS OPTIONS statement. This specification applies to all subsequent UNLOAD commands until the end of input or until Db2 HPU encounters the next PROCESS_OPTIONS TEMPLATEDD(ddname) statement.

You can use a template name instead of a JCL-allocated ddname with the following keywords:

**COPYDDN**
- Input image copy file.

**DDLDDN**
- Input DDL specification for image copy files of dropped objects.

**UNLDDN**
- Output file for physical unload.

**OUTDDN**
- Output file for logical unload.

**LOADDDN**
- Output file that contains the generated LOAD SYSIN for a logical unload.

Additionally, a template name can be used for unloading LOB columns by using LOB file reference CLOBF, DBCLOBF, or BLOBF data type attributes.

When a name is specified in the COPYDDN, DDLDDN, UNLDDN, OUTDDN, or LOADDDN keywords, Db2 HPU searches for the following items in the following sequence:

1. A DD statement that is allocated in the JCL that corresponds to the exact name specified in the SYSIN data set.
2. A template with the same name that is defined in the SYSIN data set.
3. A template with the same name that is defined in the TEMPLATE library.
4. A generic ddname (DDNxxxx) that is already allocated in the JCL that corresponds to the ddnames that are used for processing per partition. This item applies only to the COPYDDN, UNLDDN, and OUTDDN keywords.

To avoid allocation errors when an unload is processed in its initialization phase, Db2 HPU checks that all data set names that were generated from TEMPLATE statements are unique. If a duplicate statement is found, the INZT034E message is issued.

Db2 HPU always assumes that the object has at least 1000 records. Therefore, the allocated size for the output LOB files is at least 1000 times the size of the LOB. To get a better estimate of the size of the output data sets, specify YES for VUX036/RTSESTIM.

When a huge amount of data is expected for the output data sets (OUTDDN and UNLDDN), which allocation is specified through a TEMPLATE statement, it is advisable to specify the MAXPRIME option to
make sure the system limit allowed for such data sets is not exceeded. This is particularly useful for non SMS and non extended data sets

Related reference

TEMPLATE block syntax and description

Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

Example: Unloading data from a table with an XML column by using a TEMPLATE statement

This example shows how to use a TEMPLATE statement to unload data from a table with an XML column.

In this example, data is unloaded from the HISTORY XML column in the DSN8910.CUSTOMER table into CLOBF files that are written in an output PDS. The output PDS is named LABO.XMLFILES.P10459A. The other column in the SELECT statement is unloaded into a single data set that is allocated as ddname OUTPUT.

```sql
TEMPLATE XMLFILE DSN LABO.XMLFILES.P10459A(R&TMP.)
UNIT WORK DSNTYPE PDS DIR 20
UNLOAD TABLESPACE
OPTIONS TEMPLATESET (TMP=:RECNUM)
QUIESCE YES
DB2 NO
SELECT CID,
HISTORY
INTO HISTORY_CLOBF VARCHAR(44) CLOBF XMLFILE
FROM DSN8910.CUSTOMER
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

Related concepts

XML data processing

Db2 HPU can unload XML data.

Example: Unloading data from a table with a LOB column by using a TEMPLATE statement

This example shows how to use a TEMPLATE statement to unload data from a table with a LOB column (COL_LOB).

In this example, data from the base table and data from the LOB column, which is unloaded as LOB files, are unloaded into HFS files.

```sql
TEMPLATE FILEL PATH '/u/test/MYDATA/LOB.&COL.&NUM.'
TEMPLATE FILED PATH '/u/test/MYDATA/BASE.&TS.'
TEMPLATE LOAD DSN MYLOAD.RELOAD(HPULOB1) UNIT WORK
UNLOAD TABLESPACE
DB2 NO
OPTIONS
TEMPLATESET (COL =: COLNAME,
NUM =: RECNUM )
REFORMAT( TYPE CLOB INTO VARCHAR(100) CLOBF FILEL)
SELECT COL_INT, COL_LOB
FROM ME.MYTABLE
OUTDDN(FILED)
FORMAT DSNTIAUL
LOADDDN (LOAD) LOADOPT(RESUME NO REPLACE)
; /*
```

After the statement is run, data is unloaded into the following HFS files. Data from the COL_INT column in the base table is unloaded into /u/test/MYDATA/BASE.MYTS. Data from the COL_LOB column is unloaded into the following files:

- `/u/test/MYDATA/LOB.COL_LOB0000001`
- `/u/test/MYDATA/LOB.COL_LOB0000002`
- `/u/test/MYDATA/LOB.COL_LOB0000003`
Each data set contains a single LOB.

**Related concepts**

| LOB data processing |
| Db2 HPU can unload LOB data. |

**Example: Unloading into a data set allocated via a TEMPLATE with DSNTYPE EXTREQ**

In this example, you unload data in XML format with an SQL statement involving SQL expressions from an indentified table.

```sql
TEMPLATE XMLFILE DSN LABO.XMLFILES.P10459A(R&TMP.)
UNIT WORK DSNTYPE EXTREQ DIR 20
UNLOAD TABLESPACE
OPTIONS TEMPLATESET (TMP=:RECNUM)
QUIESCE YES
DB2 NO
SELECT CID,
HISTORY INTO,
HISTORY_CLOBF VARCHAR(44) CLOBF XMLFILE
FROM DSN81010.CUSTOMER
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

**GLOBAL OPTIONS block syntax and description**

A Db2 HPU command may contain several UNLOAD blocks. Use the GLOBAL OPTIONS block to specify default values that apply to all UNLOAD blocks that you specify in your SYSIN. If you specify a keyword or value in both the GLOBAL OPTIONS block and the UNLOAD block, the value in the UNLOAD block is used.

A keyword that is specified in the GLOBAL OPTIONS block applies to all UNLOAD TABLESPACE statements. Keywords that are specified in an UNLOAD TABLESPACE statement apply only to the specific UNLOAD statement.
GLOBAL OPTIONS block

Notes:
1 the rule depends on whether TAPEUNIT_DEFAULT_DEV_TO_MAX is specified in the value of the VUU030/ULOPTNS parmlib parameter.
Note: Options DB2, LOCK, QUIESCE, INDEXSCAN, EXECUTE, PARALLELISM, SORT, QSAM-BUFFERS, VSAM-BUFFERS and MAXPART can also be specified through UNLOAD block. See the UNLOAD block syntax and description section for explanations of these options.

- PROCMSG
- SELMSG
- TAPEUNIT
- TAPEREPOR
- SQLID
- APPLCOMPAT
- PROGRESS_MSG_FREQ
- HANDLE_RECORD_ID

Technical Parameters options block
Use the Technical Parameters options block to override many of the processing parameters that are specified by the Db2 HPU settings. You can override these parameters in SYSIN to change a processing option for specific jobs without impacting other Db2 HPU jobs.

These keywords can be specified only in the GLOBAL OPTIONS block after the GLOBAL keyword. They apply to all the UNLOAD commands that are coded in the SYSIN.

See “Customization reference” on page 475 for descriptions of the corresponding PARMLIB parameters with the identical name. The descriptions give a detailed explanation of each keyword. The only exceptions are the SORTCLASS SYSIN keyword, which corresponds to the VUX020/SORTCLAS parameter, and the SORTDEVT keyword, which is an alias of WRKUNIT and corresponds to the VUM013/WRKUNIT parameter.

The same keyword can be specified only once.
Notes:

1. The corresponding PARMLIB parameter is VUU0061/ACCPREP.
2. The MAXERR operand overrides any of the value specified for VUU073/MAXERR, regardless of the applicable syntax.
The arguments of the UNLOAD BUFFERS options are associated with the parameters as follows:

- **size**: is associated with the « Size of work areas buffers (VUX040/UBUFSIZE) » parameter.
- **number**: is associated with the « Size of work areas buffers (VUX040/UBUFSIZE) » parameter.

The corresponding PARMLIB parameter is VUM033/OPTRGRT.

The default value depends on the value of the VUM034/EXPLNTB PARMLIB parameter. If this parameter is not specified, its default value applies.

This option corresponds to the value DELETE of the VUM034/EXPLNTB PARMLIB parameter.

This option corresponds to the value KEEP of the VUM034/EXPLNTB PARMLIB parameter.

DISPLAY_PARMLIB corresponds to the VUU083/DISPPARM parmlib parameter.

**Related reference**

**UNLOAD block syntax and description**

See this topic for descriptions of the keywords in the GLOBAL OPTIONS block syntax.

**Examples: Using committed data that is being updated or deleted**

These examples show how you can use the CONCURRENT_ACCESS keyword when you unload data that is being updated or deleted. The data must be committed.

The following example shows how to specify the CONCURRENT_ACCESS keyword so that Db2 HPU uses committed data that is being updated or deleted.

```sql
GLOBAL CONCURRENT_ACCESS USE ;
UNLOAD TABLESPACE QUIESCE YES DB2 FORCE
SELECT * FROM DSN81010.EMP FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

The following example shows how to specify the CONCURRENT_ACCESS keyword so that Db2 HPU does not use committed data that is being updated or deleted.

```sql
GLOBAL CONCURRENT_ACCESS WAIT ;
UNLOAD TABLESPACE QUIESCE YES DB2 FORCE
SELECT * FROM DSN81010.EMP FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

**Examples: Invoking sort utilities**

These examples show how you can use the SORTUTIL keyword to invoke your on-site sort program or the Db2 Sort utility.

In the following example, the table space has 50 partitions. Several sort operations are required because of partition processing parallelism. Partition processing parallelism was requested because one output file is allocated per partition, and partition parallel processing is allowed because PARALLELISM (5, ,) is specified. The sort operations are run outside of Db2 because DB2 NO was specified. The GLOBAL statement specifies that Db2 Sort will do the sort operation.

```sql
GLOBAL SORTUTIL DB2SORT ;
UNLOAD TABLESPACE PARALLELISM (5 , ,) DB2 NO SELECT *
FROM MZLFLB.TBDZ8AFLB20101 ORDER BY 1 DESC, 2 DESC
OUTDDN(SYSREC) FORMAT DSNTIAUL
;
In the following example, several sort operations are required because several SELECT statements with ORDER BY clauses are in a single UNLOAD command. The sort operations are run outside of Db2 because DB2 NO was specified. The GLOBAL statement specifies that Db2 Sort will do the sort operations.

```
GLOBAL SORTUTIL DB2SORT
   OPTIONS TEMPLATESET(NUM=:SELNUM)
 ;
TEMPLATE SYSREC DSN MZLFLB.&DB..S&NUM. UNIT WORK
UNLOAD TABLESPACE
   DB2 NO
   SELECT *
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 1 DESC,
       2 DESC
   OUTDDN(SYSREC)
   SELECT C1, C2
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 1 DESC,
       2 DESC
   OUTDDN(SYSREC)
   SELECT C1, C2, C3
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 2 DESC,
       1 DESC
   OUTDDN(SYSREC)
   FORMAT DSNTIAUL
 ;
```

In the following example, several sort operations are required because several SELECT statements with ORDER BY clauses are in a single UNLOAD command. The sort operations are run outside of Db2 because SORT (EXTERNAL) is specified. Row selection is done in SQL mode because DB2 FORCE is specified. Db2 Sort will do the sort operations.

```
GLOBAL SORT(EXTERNAL)
   SORTUTIL DB2SORT
   OPTIONS TEMPLATESET(NUM=:SELNUM)
 ;
TEMPLATE SYSREC DSN MZLFLB.&DB..S&NUM. UNIT WORK
SPACE (100, 50) TRK
UNLOAD TABLESPACE
   DB2 FORCE
   SELECT *
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 1 DESC,
       2 DESC
   OUTDDN(SYSREC)
   SELECT C1, C2
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 1 DESC,
       2 DESC
   OUTDDN(SYSREC)
   SELECT C1, C2, C3
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 2 DESC,
       1 DESC
   OUTDDN(SYSREC)
   FORMAT DSNTIAUL
 ;
```

In the following example, the DB2SORT value that was specified in the Db2 HPU settings for the VUX037/SORTUTIL parameter is ignored. The on-site sort program is used instead of Db2 Sort.

```
GLOBAL SORTUTIL SORT
   OPTIONS TEMPLATESET(NUM=:SELNUM)
 ;
TEMPLATE SYSREC DSN MZLFLB.&DB..S&NUM. UNIT WORK
UNLOAD TABLESPACE
   DB2 NO
   SELECT *
   FROM MZLFLB.TBDZ8AFLB00101
   ORDER BY 1 DESC,
       2 DESC
   OUTDDN(SYSREC)
   SELECT C1, C2
 ;
```
Examples: Unloading partitioned tables with OPTIMIZE_PART_RANGE option

The examples below show how to request a logical unload with a WHERE clause run against a partitioned table in native mode to be optimized by only reading the partitions where the expected data is located. This is done by using the OPTIMIZE_PART_RANGE feature.

The following example requests the logical unload involving a partitioned table and specifying a WHERE clause to be optimized by using the OPTIMIZE_PART_RANGE feature so that only the relevant partitions are read by the utility.

The rows inserted into the PLAN tables by the EXPLAIN PLAN Db2 HPU requests when the OPTIMIZE_PART_RANGE feature is used are automatically deleted by Db2 HPU afterwards as the DELETE_EXPLAIN_TABLES_ROWS option is specified.

With the following unload statement:

```
GLOBAL OPTIMIZE_PART_RANGE YES, DELETE_EXPLAIN_TABLES_ROWS;
UNLOAD TABLESPACE
DB2 NO
PARALLELISM 5
SELECT * FROM MZLF.BTBUTSRG1
WHERE COL_VARCHAR >= 'AAD' AND COL_VARCHAR <= 'GL' OR COL_VARCHAR = 'Q'
OUTDDN FOUT
FORMAT EXTERNAL

SELECT * FROM MZLF.BTBUTSRG1
WHERE COL_VARCHAR >= 'AAD' AND COL_VARCHAR <= 'GL'
OUTDDN DOUT
FORMAT EXTERNAL

SELECT *
FROM DSN81010.EMP
WHERE EMPNO IS NOT NULL
OUTDDN GOUT FORMAT EXTERNAL
...```

the SYSPRINT data set contains the following messages:

```
----------------------------------------------------
INZU481I PART RANGE OPTIMIZATION ENABLE FOR DBTLS00.UTS02AA
INZU280I - UNLOAD STARTING AT POS(4, 1)
INZU280I - SELECT 00001 STARTING AT POS(6, 1)
INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM 16 TO 9 PARTITIONS [5-14],31
INZU280I - SELECT 00002 STARTING AT POS(12, 1)
INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM 16 TO 8 PARTITIONS [5-14]
----------------------------------------------------
INZU481I PART RANGE OPTIMIZATION ENABLE FOR DSN8D10A.DSN8S10E
INZU280I - UNLOAD STARTING AT POS(4, 1)
INZU280I - SELECT 00001 STARTING AT POS(6, 1)
INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM 16 TO 9 PARTITIONS [5-14]
INZU486I NO OPTIMIZATION FOUND. 5 PARTITIONS KEPT
```

The following example requests the logical unload involving a partitioned table and specifying a WHERE clause to be optimized by using the OPTIMIZE_PART_RANGE option so that only the relevant partitions are read by the utility.

The rows inserted into the PLAN tables by the EXPLAIN PLAN Db2 HPU requests when the OPTIMIZE_PART_RANGE feature is used are kept as the KEEP_EXPLAIN_TABLES_ROWS option is specified.

Chapter 5. Db2 HPU batch mode operation 119
With the following unload statement:

```sql
GLOBAL
  OPTIMIZE_PART_RANGE YES, KEEP_EXPLAIN_TABLES_ROWS
;
UNLOAD TABLESPACE PARALLELISM 5
PART(ALL)
SELECT * FROM MZLFD.TBUTSRG1
  WHERE COL_VARCHAR >=' AAD' AND COL_VARCHAR <= 'GL'
  OR COL_VARCHAR = 'Q'
OUTDDN FOUT
  FORMAT EXTERNAL

SELECT * FROM MZLFD.TBUTSRG1
  WHERE COL_VARCHAR >=' AAD' AND COL_VARCHAR <= 'GL'
OUTDDN GOUT
  FORMAT EXTERNAL
...```

The SYSPRINT data set contains the following messages:

```
INZU481I PART RANGE OPTIMIZATION ENABLE FOR DBTLS00.UTSDZAA
INZU281I - UNLOAD STARTING AT POS(4, 1)
INZU280I - SELECT 00001 STARTING AT POS(6, 1)
INZI197I EXPLAIN TABLE SUCCESSFULL:
  QUERYNO = 1794246139 , EXPLAIN_TIME = 2016-11-29 14:22:48.510000
INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM 16 TO 9
INZU280I - SELECT 00002 STARTING AT POS(12, 1)
INZI197I EXPLAIN TABLE SUCCESSFULL:
  QUERYNO = 1794331636 , EXPLAIN_TIME = 2016-11-29 14:22:48.680000
INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM 16 TO 8
```

**Example: Unloading with a specific CURRENT APPLICATION COMPATIBILITY value.**

Request the unload being done with the CURRENT APPLICATION COMPATIBILITY register set to V11R1 in order to allow the use of the CHAR9 scalar function only available.

```sql
GLOBAL
  APPLCOMPAT(V11R1)
;
UNLOAD TABLESPACE DB2 NO
SELECT
  CHAR (DECIMAL(-1.230)),
  CHAR9(DECIMAL(-1.230))
FROM SYSIBM.SYSDUMMY1
OUTDDN(SAMPLE1)
  FORMAT DELIMITED SEP '/'
```

**UNLOAD block syntax and description**

See this topic for descriptions of the keywords in the GLOBAL OPTIONS block syntax.

The following diagram shows the syntax of the UNLOAD block:
UNLOAD block

Notes:

1 When you specify LIST (list-name) in the UNLOAD TABLESPACE command, you can do only a physical unload (using UNLDDN). You cannot specify a SELECT statement in the same UNLOAD TABLESPACE command. Additionally, you cannot do a physical unload with the same UNLOAD TABLESPACE command as a SELECT statement from a LIST (list-name) statement.

2 You can specify the ANYTYPE keyword only with COPYDDN LAST_IC.

3 default value estimated from the online objects characteristics or the value of VUX003/SIZE

4 COPYDDN -n with n > 1 and COPYDDN ddname and ANYTYPE are not supported if LOB or XML data is to be unload.

5 As the COPYDDN option requires the native mode, this keyword cannot be specified if either DB2 YES (through the DB2 option or ULSEDB2 parmlib parameter) or DB2 FORCE is requested. This rule applies if COPYDDN STRICT(YES) is specified for ULOPTNS. If COPYDDN STRICT(NO) is specified, the COPYDDN option is ignored whenever either DB2 YES or DB2 FORCE is specified.
UNLOAD TABLESPACE
Identifies an UNLOAD statement. This UNLOAD statement is required.

database-bname.table-space-name
Specifies the complete name of the table space.

Restrictions:

- Db2 HPU does not support table spaces that are defined as work file.
- Do not use the following reserved words for the first word of database-name.table-spacename unless you delimit the first word with single or double quotation marks:
  - COPYDDN
  - DB2
  - LOCK
  - OPTIONS
  - PART
  - QUIESCE
  - SELECT
  - UNLDDN
  - CLONE
  - LIST
  - WITH

For example, the following command is incorrect:

UNLOAD TABLESPACE DB2.SELECT

The following forms of the command are correct:

UNLOAD TABLESPACE "DB2".SELECT
or
UNLOAD TABLESPACE 'DB2'.SELECT

LIST (list-name)
Identifies the name of a list of objects that are defined by a LISTDEF statement. The list can include table spaces, index spaces, databases, and partitions. The list cannot include LOB table spaces and directory objects. When you specify LIST (list-name), the referenced LISTDEF identifies the following objects:

- The table spaces from which the data is to be unloaded. You can use the LISTDEF pattern-matching feature.
- For partitioned table spaces, the partitions from which the data is to be unloaded. These partitions are defined by the INCLUDE, EXCLUDE, and PARTLEVEL keywords in the LISTDEF statement.

When you use the LIST keyword with a LISTDEF that represents multiple table spaces, you can define a data set TEMPLATE statement that corresponds to all the table spaces, and then specify the template name in the UNLDDN keyword. In this case, Db2 HPU associates a single table space with one output data set, except when partition parallelism is activated. To activate partition parallelism with a TEMPLATE statement, the data set name that is defined in the TEMPLATE statement must contain the &PART variable.

CLONE
Indicates that Db2 HPU is to do a physical unload of only the cloned tables from the specified table spaces when you unload a single table space by using the database-name.table-space-name variable or a list of tables spaces by using a LISTDEF. Base tables in the specified table spaces are not
unloaded. If the list contains table spaces that do not have a clone, these table spaces are not unloaded.

You can use the CLONE keyword only when you do a physical unload and specify the UNLDDN keyword.

**UNLDDN ddname**

Specifies that a physical unload of the table space is to be done, and specifies the ddname of the output data set.

The format of this data set is the same as the format when a Db2 REORG UNLOAD ONLY is done. The ddname variable is the base ddname of the output data set.

To process partitioned table spaces in parallel, specify one unlddnn statement for each partition (unldd01, unldd02,... unlddnn), where nnn is a 1- to 7-digit sequential number that identifies a partition to be unloaded. During the unload process, data from each partition is directed to the corresponding ddname. If the corresponding ddname is allocated, it is used for the given partition. Otherwise, the base ddname is used if it is allocated. The following example uses one unlddnn statement:

```sql
UNLOAD TABLESPACE DB1.TS1 PART(1,2,4,5) UNLDDN(MYDD)
```

If MYDD, MYDD01, and MYDD0004 are allocated, then MYDD contains the rows from partitions 2 and 5, MYDD01 contains the rows from partition 1, and MYDD0004 contains the rows from partition 4.

If you do not specify this keyword, specify OUTDDN on the SELECT statement.

**Attention:** When the UNLDDN keyword is used on a table space that contains a LOB column, the physical unload is performed only on the base table space, not on the LOB table space.

**Important:** When you do a physical unload, the UNLDDN keyword is valid only if a table space name or a list name is also specified in the UNLOAD TABLESPACE command. Although doing a physical unload with UNLDDN specified but without specifying the table space name was possible in previous releases of Db2 HPU, this deprecated syntax is no longer supported.

**INTERNAL_FORMAT**

Specifies that the physical unload will be made in Db2 INTERNAL format. This format is compatible with FORMAT INTERNAL in the Db2 LOAD utility.

**Important:** If the unloaded table space contains more than one table, the Db2 LOAD utility might not be able to reload the data into a table space with the same structure.

The following example shows how you can request a physical unload of data in the INTERNAL format. In this example, the entire DSN8D81A.DSN8S81E table space is unloaded.

```sql
UNLOAD TABLESPACE DSN8D81A.DSN8S81E
QUIESCE YES
DB2 NO
UNLDDN(OUTPUT)
INTERNAL_FORMAT
```

**UNLMAXROWS integer**

Specifies the maximum number of rows to unload for a physical unload. If you are unloading a partitioned table space, which is processed on a partition-by-partition basis, the limit applies to each partition.

**UNLFREQROWS integer**

Specifies sampling frequency for a physical unload. One row of every n rows is written to the UNLDDN data set.

**PART**

Specifies the partitions of the table space to be processed. Use the PART keyword only with partitioned table spaces. The PART keyword can be specified in the UNLOAD block and the SELECT block:

- If you do not specify the PART keyword in the UNLOAD block, ALL is the default value.
• If you do not specify the PART keyword in a SELECT block of this UNLOAD block, the PART specification in the UNLOAD block is used.

• If you specify the PART keyword in any SELECT statement, it overrides any values that are specified in the UNLOAD block.

• For each UNLOAD statement, the union of all the subsets of partitions that are selected in all SELECT blocks must be equal to the subset that is specified in the PART keyword unless you use UNLDDN.

• If a statement is processed by Db2, the PART keyword is ignored.

If you are unloading from partition-level full image copy, you must provide a unique name for the DD statement for each partition to be unloaded. For example, if you specify one SELECT statement to unload from partitions 1, 2, and 4 and one SELECT statement to unload from partitions 2 and 3, you must specify a DD statement with a unique name for each of the following image copy partitions: 1, 2, 3, and 4.

**integer**

Indicates which partitions are to be processed.

**ALL**

Specifies that the entire table space is to be processed.

**integer-1:integer-2**

Designates a range of partitions from `integer-1` to `integer-2`. `Integer-1` must be positive and less than the highest partition number in the table space. `Integer-2` must be greater than `integer-1` and less than or equal to the highest partition number.

**Important:** When you use a list that was generated from a LISTDEF statement, use the PARTLEVEL keyword in the LISTDEF statement to select partitions.

The default value is ALL.

**COPYDDN**

Indicates that the data from an image copy must be unloaded. When this option is specified, Db2 HPU takes the described image copy as the source of data except if COPYDDN_STRICT(NO) is specified for the ULOPTNS parmlib parameter and DB2 YES or DB2 FORCE is requested. In such a case, Db2 HPU ignores the COPYDDN option and unloads the data from the table. As this behavior is misleading, prefer setting COPYDDN_STRICT(YES) and/or clarifying the intention by specifying DB2 NO.

**ddname**

Specifies the ddname that points to the image copy.

When you specify a ddname, no control is provided by using the information from the Db2 catalog on the image copy. The ddname must allocate a full or incremental image copy data set or a set of a full image copy data set and one or several incremental image copy data sets (in such a case, the MERGE_IIC suboption must be specified) or a set of full image copy data set of partitions of a table space. See the Chapter 4, “Input and output,” on page 63 section for more details about concatenation of image copy data sets. Db2 HPU can unload from an image copy of a compressed table space if the image copy was created with the SYSTEMPAGES option. If the ddname points to an inline copy, such as a REORG share level change, you must specify the INLINE option. If you do not specify INLINE, duplicate or missing lines can occur in the output, or the job can fail during the unload process itself. If the copy is inline or is not known, you must specify the CHECK option.

When you specify a ddname, Db2 HPU prevents allocation of VSAM data sets within the table space, and Db2 HPU must access an image copy except when the SELECT statement is issued by Db2.

When you specify a ddname, Db2 HPU works on archives, avoids possible contention, and improves performance because QSAM is more efficient than VSAM.

Db2 HPU can unload an image copy that contains the data in a table space whose structure does not match the table space that is described in the Db2 catalog or the table space that is described by the DDLDDN data set. These mismatched table space structures concern only the definition of...
partitioning, not the data itself (the table structure). The following table shows whether the mismatched table space structure is supported:

**Table 18. Db2 HPU support for table space structures**

<table>
<thead>
<tr>
<th>Type of table space in the image copy</th>
<th>Nonpartitioned table space in the Db2 catalog or DDLDDN</th>
<th>Partitioned table space in the Db2 catalog or DDLDDN</th>
<th>Range-partitioned table space in the Db2 catalog or DDLDDN</th>
<th>Partition-by-growth table space in the Db2 catalog or DDLDDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonpartitioned table space (simple or segmented)</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Partitioned table space with a different number of partitions</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Range-partitioned table space with a different number of partitions</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Partition-by-growth table space</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>

**Notes:**

1. If you request a file per partition, informational message INZU394I is issued, and all rows are unloaded in the file that corresponds to partition 1.
2. Support is limited by the following restrictions:
   - Unloading from an image copy when the number of partitions in the image copy exceeds the number of partitions in the Db2 catalog.
   - Unloading into an output file per partition when the number of partitions in the image copy is less than the number of partitions in the Db2 catalog. In this case, you can do only a global unload of all data into a single output file.
   - Unloading LOB and XML can only be done from FlashCopy image copies and only the LAST_IC option is supported.

Unless you use a DDL input file by specifying DDLDDN, the image copy must correspond precisely to the online table space, particularly to the level of the object IDs (OBID) of Db2 objects. If you have issued a DROP command or a CREATE statement, you must specify the ORIGINOBID keyword in the SELECT statement.

To work with a simple table space with multiple data sets, concatenate these data sets under the same ddname.

You can specify a generic value for COPYDDN, as shown in the following example:

```
COPYDDN FIC*
```

Values are concatenated in the order that they are encountered in the JCL.

**Attention:** The ddname that results from the asterisk (*) is not checked. For example, if you specify COPYDDN  ST*, all data sets that are allocated to the STEPLIB ddname are considered, and the job stops.
INLINE
Allows you to specify that an image copy is of type INLINE, which means that this image copy was taken when Db2 HPU is run. If COPYDDN LAST_IC is specified, you do not need to specify INLINE.

As the INLINE copy processing requires the image copy to be sorted, do not specify INLINE for an image copy which is certainly not an inline image copy to avoid a useless sort operation to be performed.

CHECK
Specifies that Db2 HPU determines the relevant way to process the image copy allocated under the ddname specified by the COPYDDN option by checking if either of the following is true:

- it is an inline image copy
- it was built by the COPYTOCOPY utility.

This option is not valid for generic ddnames and ddnames referring to a concatenation of image copies.

FROM_COPYTOCOPY
Allows you to specify that an image copy was created by the COPYTOCOPY utility so that Db2 HPU processes the data set accordingly.

MERGE_IIC
Specifies that the ddname is allocated to a concatenation of a Full Image Copy dataset and one or several Incremental Image copies datasets. Db2 HPU merges data pages from all these datasets before unloading the rows. The Full Image Copy can be a global image copy or a partitioned image copy.

Requirement: the image copies must contain SYSTEM PAGES. Otherwise the result is unpredictable. See “Concatenation of image copies” on page 67 section.

FLASHCOPY
Indicates that the allocated image copies are VSAM Flash Copy data sets. For a partitioned table space, one Flash Copy data set must be allocated per partition to unload.

Restriction: Only copies of single data set non-partitioned table spaces or single data set partitions are supported.

DDLDDN $ddldd$
Db2 HPU can use an image copy as input for an unload even when the structure of the unloaded objects does not match the table definition in the Db2 catalog. This unload includes tables and table spaces that have been dropped since the creation of the image copy file and eventually re-created with a different structure, and image copies that were generated on another Db2 subsystem.

When you specify DDLDDN $ddldd$, Db2 HPU reads the definition of the Db2 objects that are in the specified file instead of the Db2 catalog. If you specify DDLDDN $ddldd$, the corresponding ddname or TEMPLATE statement must correspond to a file that contains the CREATE statements that correspond to the structure of the data in the unloaded image copy file. You can code DDLDDN only when the input copy is allocated by using a ddname or a TEMPLATE statement. DDLDDN cannot be used with COPYDDN LAST_IC or COPYDDN integer.

If the input image copy is an inline copy, you must specify the INLINE option. The CHECK option of the COPYDDN keyword is not compatible with DDLDDN.

The DDLDDN file must contain the following CREATE statements in the following order:

1. CREATE DATABASE
2. CREATE TABLESPACE
3. CREATE TABLE
4. CREATE INDEX for the clustering index for an index partitioned table space
Any ALTER statement or CREATE VIEW statement in the DDLDDN file is ignored. The DDL that is contained in the DDLDDN can specify an OBID in the CREATE TABLE statement. This OBID is the default for the ORIGINOBID option of the SELECT statement. If the DDL does not contain an OBID, the default value is 0 for the ORIGINOBID keyword on all SELECT statements for the UNLOAD command. For a multi-table table space, you must specify the OBID of each table by using ORIGINOBID either in the CREATE TABLE statements of the DDLDDN file or in the SELECT statements of the SYSIN.

**SIZE integer**

Specifies the size, expressed as a number of table rows, of the input image copy designated through the "COPYDDN ddnname" option. Db2 HPU needs to know the size of the input image copies to determine the relevant sort parameters to use when sorting the input image copy is required. Whenever possible, Db2 HPU estimates the size of the input copy by itself taking the following information into account with the following order of precedence:

1. the value of the COPYPAGESF column of the SYSIBM.SYSCOPY catalog table if the image copy is referenced and the CHECK option was specified.
2. the value specified by the SIZE option, if specified.
3. the cardinality of the object to unload, if the DDLDDN option is not specified.
4. the value specified by the VUX003/SIZE parmlib parameter.

Specify a value through the SIZE option when the information #1 above is not available and the size of the input image copy is significantly different from the information #3 and #4 above. This avoids an under or overestimate of the resources for the sort process that could lead to a useless overconsumption of some system resources or a failure of the sort processing and the subsequent failure of the unload.

As sorting the image copy requires to determine the number of pages of the image copies, this number is calculated from the value given by the SIZE option or parameter by the following formula:

\[
\text{Number of pages} = \left(\frac{(\text{SIZE} \times \text{average length of rows})}{4096}\right) \times \left(\frac{\text{PGDFIN}}{100}\right)
\]

where PGDFIN stands for the value set through the PGDFIN option or the VUX023/PGDFIN parmlib parameter. If you only know the number of pages of the image copy, use this formula to compute the value to specify for SIZE.

The value specified by the SIZE option is also systematically used to determine the sort parameters when a logical unload from the image copy specifies an ORDER clause.

Default value: the value of the VUX003/SIZE parameter.

**OBID_REPORT**

Specifies that Db2 HPU generates in SYSPRINT a list of all OBIDs for rows of tables that are found in the input full image copy (FIC) when you unload from an image copy.

**Attention**: This option might have a negative impact on the overall performance of Db2 HPU when you unload from an image copy.

**LAST_IC or integer**

Indicates that the data to be unloaded must be retrieved from one image copy or a set of image copies determined by Db2 HPU by searching the Db2 catalog for image copies that fulfill the following conditions:

- The image copies are full image copies (FIC).
- The image copies are cataloged dataset, unless SEARCH_UNCATALOGED_IC(YES) option is set in the VUU030/ULOPTNS parameter, in which case uncatalogued image copies are considered as well.
- The image copies are one of the following types:
  - Regular or FlashCopy image copies if no LOB or XML data is to be unloaded.
FlashCopy if LOB or XML data are to be unloaded. Only LAST_IC is allowed.

- The set of image copies includes all the following necessary objects:
  - The table space that includes the table to be unloaded, if this table space is partitioned, the image copies set must include all the data of the involved partitions (see the GLOBAL/PARTITIONED/ANYTYPE options for more details). The later rule also applies to partitioned-by-size table spaces.
  - If the unload request involves a LOB or XML column, the base table space (see above considerations on partitioned objects that also apply) and both the auxiliary table space and the nodeid index for the involved LOB or XML column.

- All the images copies have the same START_RBA (see SYSIBM.SYSCOPY table) if either of the following conditions applies:
  - The CONSISTENT option is specified.
  - A LOB or XML column is to be unloaded. Only LAST_IC is allowed.

When COPYDDN LAST_IC or COPYDDN -integer options is specified, Db2 HPU searches the Db2 catalog (i.e. the SYSIBM.SYSCOPY table) for images copies that satisfy the criteria mentioned above and the additional conditions given through the LAST_IC or -integer option and the GLOBAL, PARTITIONED and CONSISTENT sub-options. If such an image copy is found, Db2 HPU attempts to dynamically allocate the corresponding data set and unloads the data from this data set.

LAST_IC

Specify LAST_IC to get the data unloaded from the last eligible set of image copies. As all the above conditions must be satisfied by the set of image copies, be aware that the image copy selected for a single object might not be the latest copy made for this object.

-integer

Specify a negative integer (-n) to request Db2 HPU to retrieve the data from the nth last set of eligible image copies. Specifying n greater than 1 is forbidden if LOB or XML columns are to be unloaded.

COPYDDN LAST_IC is a synonym of COPYDDN -1.

Note:

If the last image copy of a partitioned table space consists of one image copy per partition, the LAST_IC option is not supported unless you also specify the PARTITIONED keyword or the ANYTYPE keyword.

You can specify the following keywords for COPYDDN LAST_IC or COPYDDN integer:

GLOBAL

This keyword is the default value when COPYDDN LAST_IC or COPYDDN integer is specified. If a nonpartitioned table space is processed, this value is ignored. However, for a partitioned table space, LAST_IC processing looks for a global FIC of the complete table space.

PARTITIONED

If the unloaded table space is not partitioned, the PARTITIONED keyword is ignored. However, for a partitioned table space, LAST_IC processing looks only for FICs that were created for each partition. When you specify COPYDDN n, only full image copies per partition are counted.

When you specify the PARTITIONED keyword, the most recent FIC for all unloaded partitions is selected. If some of the unloaded partitions do not have an FIC, Db2 HPU issues an error message.

ANYTYPE

Allows COPYDDN LAST_IC processing to look for the most recent global FIC or FIC per partition. The ANYTYPE keyword can be specified only with COPYDDN LAST_IC.
If the ANYTYPE keyword is coded without partitions, Db2 HPU looks for the latest FIC for each partition and for the latest global FIC. If all FICs for each partition are more recent than the global FIC, the more recent FICs are used. Otherwise, the global FIC is used. ANYTYPE cannot be used if a LOB or an XML column is to be unloaded.

**CONSISTENT**

Checks whether the selected FIC has the same START_RBA for all unloaded partitions. This option ensures consistency in the unloaded data. You can specify the CONSISTENT keyword with only the PARTITIONED keyword or the ANYTYPE keyword.

The following conditions affect the CONSISTENT keyword:

- If the selected FIC is a global FIC that was created for all the partitions in the table space, the CONSISTENT keyword is always verified.
- If the selected FIC is a copy per partition, the CONSISTENT keyword checks whether the START_RBA is identical to the selected input FIC of all unloaded partitions:
  - If the START_RBA is identical, the unload is processed.
  - If the START_RBA is different, depending on the PART keyword or the ANYTYPE keyword, Db2 HPU looks for the preceding FIC.

If you did not specify the CONSISTENT keyword, the START_RBA is not checked. You can check whether the available copies meet the specifications of the COPYDDN option by completing the following steps:

1. From the Db2 catalog, extract a list of the image copies for `database-name.table-space-name` and partition numbers that match the list that is specified by the PART keyword. If you specified the ANYTYPE keyword, you must add 0 to the list. Order this list by descending START_RBA.
2. Split the list into groups that have the same START_RBA value.
3. Depending on the option that you specified in the COPYDDN keyword, select the relevant group. If the LAST_IC keyword is specified, select the last group. If you specified an integer, select an nth older group.
4. Check whether the list of image copies covers the list of partitions to be unloaded (0 match the whole set of partitions).

**EXCLUDE_FIC**

The EXCLUDE_FIC option allows to specify an exclusion criterion to apply when the eligible full image copies (FIC) are searched as requested by the COPYDDN LAST_IC or COPYDDN -integer options.

- NONE: no kind of FIC is excluded.
- TO_SORT: the images copies which processing by Db2 HPU requires a sort process are not eligible. This kind of image copies is created by the LOAD or REORG utilities (inline image copies) or by the COPYTOCOPY utility.

The eligible image copy(ies) is(are) selected according to the selection criterion expressed by LAST_IC or -integer and the GLOBAL, PARTITIONED, ANYTPE or CONSISTENT suboptions if any. Db2 HPU determines a chronological ordered list of candidate FICs for unloading.

COPYDDN LAST_IC permits to select the most recent FIC from this list. COPYDDN integer (-n) permits to browse this list back n times and select the corresponding FIC (-1 stands for the most recent FIC).

EXCLUDE_FIC TO_SORT allows Db2 HPU to exclude from its candidate list of FICs, the ones that require to be sorted. EXCLUDE_FIC NONE do not exclude any FIC. The default value is the value of the VUU086/EXCLFIC parmlib parameter.

**Examples**

The following examples are based on the assumption that the Db2 catalog references 5 Full Image Copies of the table space DBINFDM/TSSCA. These images copies are designated FIC1,
FIC2, FIC3, FIC4 and FIC5, from the less to the most recent point in time (i.e. the STARTBA column of the SYSIBM/SYSCOPY table). Only FIC2; FIC3 and FIC5 require to be sorted.

<table>
<thead>
<tr>
<th>FIC</th>
<th>Sort Required</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIC1</td>
<td>No</td>
<td>The less recent FIC</td>
</tr>
<tr>
<td>FIC2</td>
<td>Yes</td>
<td>The less recent FIC</td>
</tr>
<tr>
<td>FIC3</td>
<td>Yes</td>
<td>The less recent FIC</td>
</tr>
<tr>
<td>FIC4</td>
<td>No</td>
<td>The most recent FIC</td>
</tr>
<tr>
<td>FIC5</td>
<td>Yes</td>
<td>The most recent FIC</td>
</tr>
</tbody>
</table>

**Example 1: Unloading data from last FIC with no FIC exclusion**

The following unload requests the data from the latest FIC be unloaded without excluding any kind of FIC.

```
UNLOAD TABLESPACE DBINFDM.TSSCA
COPYDDN LAST_IC EXCLUDE_FIC NONE
SELECT *
FROM USER1.TBSCA
```

The image copy FIC5 is chosen and unloaded.

**Example 2: Unloading data from last FIC not requiring to be sorted**

The following unload requests the data from the latest FIC which process does not require a sort phase be unloaded.

```
UNLOAD TABLESPACE DBINFDM.TSSCA
COPYDDN LAST_IC EXCLUDE_FIC TO SORT
SELECT *
FROM USER1.TBSCA
```

The image copy FIC4 is chosen and unloaded.

**Example 3: Unloading data from the before last FIC not requiring to be sorted**

The following unload requests the data from the before last FIC which process does not require a sort phase be unloaded.

```
UNLOAD TABLESPACE DBINFDM.TSSCA
COPYDDN LAST_IC EXCLUDE_FIC NONE
SELECT *
FROM USER1.TBSCA
```

The image copy FIC1 is chosen and unloaded.

**Restriction:** When an image copy is used, the LOCK and QUIESCE keywords are not supported.

**DB2**

Specifies the processing to be performed for SELECT statements that are not supported by Db2 HPU.

**YES**

Indicates that if the SELECT statement is too complex to be handled directly by Db2 HPU, Db2 is called to extract the rows. A warning message is issued to report this occurrence. The overall return code is raised to 4.

**NO**

Indicates that the SELECT statements must be processed by Db2 HPU. If a SELECT statement is not supported by Db2 HPU, an error occurs and processing stops. The control is done when the SYSIN is read and before an unload is processed.

**FORCE**

Indicates that Db2 must be used to extract the requested rows. Specifying DB2 FORCE is useful when the SELECT statement uses a filtering predicate that is efficiently processed through SQL, and the filtering factor is high. An informational message is issued in the report.

**Restriction:** When you use COPYDDN to unload data from an image copy, DB2 YES or DB2 FORCE is accepted if you specify COPYDDN_STRCT(NO) in the VUU030/ULOPTNS PARMLIB parameter. If you specify COPYDDN_STRCT(YES), unloading an image copy forces the default value of the DB2 option to NO, and specification of DB2 YES or DB2 FORCE in SYSIN is not supported.

When a logical unload is processed, Db2 HPU can retrieve the data to be unloaded by native mode or SQL mode. In native mode, Db2 HPU retrieves the data by a direct reading of VSAM linear data sets (LDS) of the table space. In SQL mode, Db2 HPU retrieves the data by passing the SELECT statement to Db2. For some complex SQL statements, which are called unsupported SELECTs, native mode is
unavailable. To determine which access method to use for other statements, consider performance, output, and data consistency.

**Performance**

**Native mode**
Use native mode when you need to unload a large amount of table data or a partition of a table space.

**SQL mode**
Use SQL mode for SELECT statements for which Db2 access paths are efficient. For example, use SQL mode when you have a highly filtered WHERE clause with a predicate that is indexed, and access is index only.

**Output**
The output does not depend on the access mode. However, for the following cases, Db2 HPU does not create the same output in both modes. Consider these differences when you choose an access mode.

<table>
<thead>
<tr>
<th>Output</th>
<th>Condition</th>
<th>Native mode</th>
<th>SQL mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD statement</td>
<td>The resulting table contains IDENTIFY columns.</td>
<td>The LOAD statement uses the IGNOREFIELDS option so that identity columns are correctly handled.</td>
<td>The LOAD statement describes the identity columns as standard, non-identity columns.</td>
</tr>
<tr>
<td>Data format</td>
<td>FORMAT DSN TIAUL is specified where DSN TIAUL STRICT is not specified in the VUU030/ULOPTNS parameter.</td>
<td>String constants are unloaded as CHAR data.</td>
<td>String constants are unloaded as VARCHAR data.</td>
</tr>
<tr>
<td>Output data</td>
<td>PART option is specified.</td>
<td>Data is unloaded normally.</td>
<td>The PART option is not supported. Use SQLPART instead.</td>
</tr>
<tr>
<td></td>
<td>• COPYDDN and DB2 YES are specified with an unsupported SELECT statement.</td>
<td>The unloaded data comes from the image copy.</td>
<td>The unloaded data comes from the table space.</td>
</tr>
<tr>
<td></td>
<td>• COPYDDN and DB2 FORCE are specified, but COPYDDN STRICT was not specified in the VUU030/ULOPTNS parameter.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Data consistency**
Depending on the QUIESCE and LOCK keywords, which affect data consistency, the values of the unloaded data might not be the same in both access modes because data is not selected from the same sources, such as the table space LDS or data that is provided by Db2.

The access mode that Db2 HPU uses depends on whether the SQL is supported and on the DB2 option that you specify.

**DB2 YES**
Specify DB2 YES to ensure that Db2 HPU determines which access mode to use. Db2 HPU uses native mode when SQL statements are supported.

Consider performance when you specify DB2 YES because native mode is not always the fastest method.
When you apply maintenance and DB2 YES is specified, data might be processed in native mode instead of SQL mode because Db2 HPU might support SQL statements that were previously unsupported. Therefore, changes in the output might occur.

**DB2 NO**
Specify DB2 NO to ensure that Db2 HPU uses native mode.

**DB2 FORCE**
Specify DB2 FORCE to ensure that Db2 HPU uses SQL mode.

To select the best DB2 option in your environment, unload the data with DB2 NO specified, then unload the data with DB2 FORCE specified, and select the option that runs fastest. However, remember that when you specify DB2 FORCE, performance depends on the access paths that are chosen by Db2, the table, and the index statistics. Ensure that the output, such as the data format and the LOAD statement, from the selected mode meets your requirements. Consider specifying EXECUTE NO to check whether the SELECT statement is supported and the content of the related LOAD statement that was generated.

The default value is the value of the VUU011/ULSEDB2 PARMLIB parameter.

**EXECUTE**
Specifies whether the LOADDN file is generated without unloading the data.

**YES**
Db2 HPU processes all LISTDEF, LISTDEFTBV, TEMPLATE, and UNLOAD commands that are specified in SYSIN, allocates the output files that correspond to the TEMPLATE definition, and unloads the data that is specified in the UNLOAD and SELECT statements. If you specify LOADDN, Db2 HPU generates the LOAD SYSIN.

**NO**
Db2 HPU does not allocate the output files that correspond to the TEMPLATE definition except for the LOADDN files, and Db2 HPU does not unload the data that is specified in the UNLOAD and SELECT statements. However, Db2 HPU does process all LISTDEF, LISTDEFTBV, TEMPLATE, and UNLOAD commands that are specified in SYSIN. If requested, Db2 HPU writes the LOAD commands in the LOADDN.

Required FIC information that is used as input is displayed, and execution reports are generated, but DISPLAY and QUIESCE commands on the unloaded table spaces are not issued. If you specified LOCK YES in SYSIN, locks are not issued.

**Important:** When you use JCL-allocated output files instead of templates, the output files must be allocated because processing depends on the DD statements that are allocated in the JCL.

The default value is YES.

**LOCK**
Indicates whether Db2 HPU must lock the table during the unload.

**YES**
The data is accessed in read-only mode while Db2 HPU runs. The "Eligible processing mode(s) for locking data feature (VUU088/ULLOCKMO)" parmlib parameter allows to choose whether this feature applies to the SQL and/or the native processing modes. For a given Db2 HPU invocation, Db2 HPU chooses one of the following technique:

- a SQL LOCK statement is issued against the tables involved in the considered SELECT statement. When Db2 HPU can determine the access to only a subset of partitions of the object is necessary, the LOCK is restricted accordingly. This is the only technique applicable when a SELECT statement is processed in SQL mode.
- Db2 HPU starts the table space containing the data to be unload in UTRO restricted state. This is not applicable to the SELECT statements process in native mode.

The rules to choose the LOCK technique are shown in the table below:
### Table 20.

<table>
<thead>
<tr>
<th>VUU088/ULLOCKMO</th>
<th>LOCK requested for a SELECT in native mode?</th>
<th>LOCK feature requested for a SELECT in SQL mode?</th>
<th>VUU074/ULLOCKTE</th>
<th>chosen technique (comment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>any</td>
<td>no</td>
<td>no</td>
<td>any</td>
<td>none (no LOCK is requested)</td>
</tr>
<tr>
<td>NATIVE_ONLY</td>
<td>no</td>
<td>yes</td>
<td>any</td>
<td>none (the LOCK request is ignored in SQL mode)</td>
</tr>
<tr>
<td>NATIVE_ONLY</td>
<td>yes</td>
<td>no</td>
<td>LOCK</td>
<td>SQL LOCK</td>
</tr>
<tr>
<td>NATIVE_ONLY</td>
<td>yes</td>
<td>no</td>
<td>START_UTRO</td>
<td>START UTRO</td>
</tr>
<tr>
<td>NATIVE_ONLY</td>
<td>yes</td>
<td>yes</td>
<td>LOCK</td>
<td>SQL LOCK (it applies only to the native processing mode - the LOCK request is ignored in SQL mode)</td>
</tr>
<tr>
<td>NATIVE_ONLY</td>
<td>yes</td>
<td>yes</td>
<td>START_UTRO</td>
<td>START UTRO (it applies only to the native processing mode - the LOCI request is ignored in SQL mode)</td>
</tr>
<tr>
<td>ALL</td>
<td>no</td>
<td>yes</td>
<td>any</td>
<td>SQL LOCK</td>
</tr>
<tr>
<td>ALL</td>
<td>yes</td>
<td>no</td>
<td>LOCK</td>
<td>SQL LOCK</td>
</tr>
<tr>
<td>ALL</td>
<td>yes</td>
<td>no</td>
<td>START_UTRO</td>
<td>START UTRO</td>
</tr>
<tr>
<td>ALL</td>
<td>yes</td>
<td>yes</td>
<td>LOCK</td>
<td>SQL LOCK</td>
</tr>
<tr>
<td>ALL</td>
<td>yes</td>
<td>yes</td>
<td>START_UTRO</td>
<td>SQL LOCK (the SQL LOCK technique is required for the SELECT statement processed in SQL mode)</td>
</tr>
</tbody>
</table>

**NO**

Db2 HPU processes the table without changing its access mode. Specifying LOCK NO does not preclude Db2 from taking locks, if SQL Access is used to access the data.

**QUIESCE**

Specifies whether to issue a QUIESCE command against the table space before unloading it. If the unload is against an image copy and DB2 FORCE is specified, or if DB2 YES is specified and no SELECT statement is processed natively, the QUIESCE keyword is ignored.

**YES**

The QUIESCE command is processed if the table space is not in COPY-pending status; otherwise, the table space is stopped and restarted.
**NO**

The table space is processed without the QUIESCE command.

**Attention:** Db2 HPU operates on the physical VSAM data set level that is outside of Db2. If you run Db2 HPU on a table in which a new row was recently inserted, the unloaded data might not contain the new row. The unloaded data might not show the row because Db2 might not have externalized the data to DASD yet. This situation can occur when you use Db2 HPU without issuing a QUIESCE (or STOP) on the object. Be careful when you specify QUIESCE NO.

**INDEXSCAN**

Specifies to unload the rows of a table space according to the order of its index cluster. When you specify the INDEXSCAN keyword, it is a substitute for the SORT utilities (DFSORT).

When you specify INDEXSCAN, all SELECT statements that are relative to the same table space are sorted according to the index cluster. If the pages of the index cluster are too unorganized, the INDEXSCAN keyword is deactivated, and DFSORT is run. See the JESMSGGLG card of the job execution output reports for more information about the using the INDEXSCAN keyword.

The following conditions are required to use the INDEXSCAN keyword:
- The table space to be unloaded must have a valid index cluster.
- Only a table space can be unloaded.
- The table space to be unloaded must be single-table (partitioned or nonpartitioned).
- At least one SELECT statement of the unload must be native and must contain either an ORDER BY clause or an ORDER CLUSTER clause.

**Restrictions:** The INDEXSCAN keyword feature is disabled whenever in the following situations:
- Partition parallelism is activated, the table space is partitioned, and there is only one output file.
- The cluster index is nonpartitioned, and the table space is partitioned.
- The cluster index must not involve a DECFLOAT type column.

In such cases, the INDEXSCAN option may be specified but is ignored.

**NO**

The INDEXSCAN keyword is not activated.

**AUTO**

The following conditions are required to use the AUTO option:
- At least one native SELECT statement without a WHERE clause must be specified. However, depending on the order of the index cluster columns (complete or partial), it must have either an ORDER CLUSTER clause or an ORDER BY clause.
- There must be no SELECT statement with an ORDER BY clause that is not in the order of the index cluster columns (complete or partial).

**YES**

The conditions for YES are the same as the conditions for AUTO except YES supports the WHERE clause.

The default value is AUTO.

**PARALLELISM lds or (lds,db2,ts)**

Specifies the parallelism degree for an unload. You can define parallelism for the following objects:
- Table spaces
- SELECT statements that are processed by Db2
- Partitions for a partitioned table space

**lds**

Specifies the parallelism degree to use when a partitioned table space is unloaded, where lds indicates the maximum number of partitions that are processed in parallel.
When unloading a partitioned table space in a single output file, you can force partition parallelism by using the *lds* subparameter of the PARALLELISM keyword in the UNLOAD TABLESPACE command or by specifying YES for the VUU036/GBLPARAL PARMLIB parameter.

When a single output file is used as output and if GBLPARAL is set to NO and PARALLELISM *lds* or PARALLELISM(*lds*,*db2*,*ts*) is not specified in the UNLOAD command, Db2 HPU does not use partition parallelism. Valid values are greater than or equal to 1.

The default value is the value that is specified in the VUX025/PARALLEL PARMLIB parameter.

### db2

Specifies the parallelism degree to use for SELECT statements that are processed by Db2 (using DB2 FORCE or DB2 YES with unsupported SELECT statements). This parallelism is defined in a single UNLOAD command. Parallelism is not possible for SELECT statements that are coded in separate UNLOAD commands. Valid values are greater than or equal to 1.

The default is the value that is specified in the VUX031/DB2PARAL PARMLIB parameter.

### ts

Specifies the maximum parallelism degree value to use when data is unloaded from several table spaces in the same UNLOAD command. This value applies when more than one supported SELECT statement that corresponds to different table spaces are coded in the same UNLOAD command. In this case, Db2 HPU uses the parallelism degree that is specified by the *ts* value to unload table spaces in parallel.

**Attention:** When at least one SELECT statement that is coded in the same UNLOAD command is processed by using Db2, the *ts* value is decreased by 1. For example, if you specify PARALLELISM(10,10,5), and if at least one unsupported SELECT statement is coded in the UNLOAD command, only four table spaces can be processed in parallel.

The default is the value that is specified in the VUX030/UTLPARAL PARMLIB parameter.

The variables *lds*, *db2*, and *ts* values are positional, and the commas are required even if you do not want to set the value for one subparameter. For example, to specify a parallelism degree of 10 for Db2-processed SELECT statements without changing other values, specify PARALLELISM(*,10,*).

For SELECT statements that are processed natively by Db2 HPU, the parallelism degrees that can be set by using the PARALLELISM keyword or by the corresponding PARMLIB parameters are limited by the value of the VUX005/MAXSORT PARMLIB parameter when the ORDER CLUSTER clause or the ORDER BY clause is used.

### SORT

Specifies how to process the ORDER BY clause for a SELECT statement that is processed by Db2.

- **INTERNAL**
  
  Specifies that the ORDER BY clause is processed by Db2.

- **EXTERNAL**
  
  Specifies that the SELECT statement is processed by Db2 without the ORDER BY clause. Db2 HPU sorts the rows that are returned by Db2 to process the ORDER BY clause. Db2 HPU determines the estimated number of rows depending on which of the following options is specified:

  - **number-of-lines**
    
    Specifies the estimated number of rows to be sorted.

  - **VSAMSIZE**
    
    Specifies that Db2 HPU calculates the estimated number of rows to be sorted based on the size of the VSAM data sets.

If you specify EXTERNAL without specifying number-of-lines or VSAMSIZE, the value of the VUX003/SIZE PARMLIB parameter is equal to the number of sorted rows.

**Restriction:** The EXTERNAL keyword is supported only when the columns of the ORDER BY clause are part of the list of selected items. The VSAMSIZE option is accepted only for supported SELECT statements that are processed with DB2 FORCE.

The default value is INTERNAL.
**QSAM-BUFFERS**
Specifies the BUFNO parameter of the DCB for sequential QSAM access.

The default value is the value that is specified in the VUM022/QSBUFNO PARMLIB parameter.

**VSAM-BUFFERS**
It specifies the number of data buffers for sequential VSAM (the BUFND parameter of the ACB for VSAM).

**Note:** Specifying a large value might increase the amount of memory that Db2 HPU requires.

Valid values are 0 - 65535.

The default value is the value that is specified in the VUM023/VSBUFND PARMLIB parameter.

**MAXPART n**
Specifies the maximum number of partitions that Db2 HPU can process in a single unload process. If the total number of partitions that need to be processed for a table space is larger than the value that is you specify in MAXPART n, Db2 HPU automatically splits the single unload into several unloads to respect the value of the MAXPART keyword. You can use this parameter to reduce the memory consumption of Db2 HPU. If the value of n is smaller than the value of the PARALLELISM keyword at the partition level, the effective partition parallelism is limited by the MAXPART keyword.

The MAXPART keyword affects only physical unloads and logical unloads that are processed natively by Db2 HPU.

The default value is the value of the VUU060/ULMAXPAR PARMLIB parameter. When you specify 0, no splitting is done. When you specify a nonzero value, ORDER BY and ORDER CLUSTER clauses can be used only when each partition is unloaded into a separate file. When you specify a nonzero value and a split is done, the OUTMAXROWS keyword or the UNLMAXROWS keyword applies to each partition.

**Impact of the MAXPART parameter on the global parallelism**
When you specify MAXPART, the parallelism between table spaces (the ts subparameter of the PARALLELISM keyword or the VUX030/UTLPARAL PARMLIB parameter) applies to each subset of partitions that are processed internally as a separate table space. Setting the MAXPART keyword impacts the global parallelism that is used when you unload a table space.

The following example shows how different values for the MAXPART keyword can affect a particular table space when the PARALLELISM keyword is specified:

**Example: A table space with 1000 partitions and PARALLELISM(20, 5, 5)**

- When you do not specify the MAXPART keyword or when you specify MAXPART 0, the complete table space with 1000 partitions is processed in a single set of partitions, and 20 partitions are unloaded in parallel. Db2 HPU allocates memory to correspond to the 1000 partitions.
- When you specify MAXPART 40, the 1000 partitions are split into 25 sets of 40 partitions. Five sets of 40 partitions are processed concurrently because parallelism between table spaces is set to 5. In each set of partitions, a parallelism degree of 20 is used between partitions.

Memory usage corresponds to 200 (5 x 40) partitions, and the maximum global parallelism degree between partitions is 100 (5 x 20) partitions. If you specify MAXPART 40 and PARALLELISM(20, 5, 1), the memory usage corresponds to 40 partitions, and the global parallelism between partitions is 20.

**Impact of the MAXPART keyword and the table space parallelism**
Each set of partitions is processed as a separate table space so Db2 HPU starts to process a new set of partitions only when a previous set is completely processed. This type of processing can reduce the global partition parallelism under certain circumstances (for example, when sets that contain empty partitions are processed). To keep the global partition parallelism degree at its highest level, try specifying MAXPART 1, and change the parallelism between table spaces.

The following examples show how different settings for the MAXPART and PARALLELISM keywords can affect table space parallelism. These examples show that you can easily manage
partition parallelism when you prioritize parallelism between table spaces rather than the MAXPART keyword.

**Example: MAXPART 50, PARALLELISM(50, 5, 1)**
These settings create sets of 50 partitions. One set of partitions is processed at a time. Db2 HPU starts to unload 50 partitions in parallel, but after one partition is unloaded, Db2 HPU does not start to unload a new partition because it must finish unloading the complete partition set before it starts to unload a new partition set.

**Example: MAXPART 25, PARALLELISM(25, 5, 2)**
These settings create sets of 25 partitions. Two sets of partitions are processed in parallel. Db2 HPU starts to unload 50 (2 x 25) partitions in parallel. When one partition is unloaded, Db2 HPU waits for the set of 25 partitions to unload before it starts to unload a new partition because it needs to finish unloading the complete partition set before it starts to unload a new partition set.

**Example: MAXPART 1, PARALLELISM(1, 5, 50)**
These settings create sets that contain only one partition. Db2 HPU processes 50 sets of partitions in parallel. When one partition is unloaded, Db2 HPU starts to unload a new set.

**TAPEUNIT**
Specifies the way allocations of data sets with a given tape unit or tape storage class are done. You can control the maximum number of devices associated with a tape unit or storage class that can be allocated to a single Db2 HPU step. You can also specify the default number of devices to use for a single data set allocation with a given tape unit or storage class.

A TEMPLATE statement allocates files on tape when the tape unit or tape storage class is defined by the TAPEUNIT parameter or when the template statement contains the tape option STACK or TRTCH. The VOLUMES and UNCNT parameters in the TEMPLATE statement are ignored, and scratch volumes are used.

When you unload to tape units, each physical unload or logical unload is serialized, and Db2 HPU reads the input files (table space or image copy files) once for each type of unload.

When OUTDDN or UNLDDN keywords specify several ddnames that allocate files on tape, each tape output file is generated by unloading the input data separately.

Using partition parallelism to unload a partitioned table space depends on the number of output files (one per partition or a single global unload file) and the number of available tape devices. When you request a global unload, a single tape device is used. Depending on the third value of the PARALLELISM keyword, parallelism can be used to read partitions. Each partition can be unloaded on a separate tape device. The parallelism degree is defined by the first value of the PARALLELISM keyword. The maximum parallelism degree is limited by the number of usable tape devices that are defined by the TAPEUNIT keyword. When you unload a partitioned table space into tape files, use either one global output file for all partitions, or use a separate file for each partition. You cannot unload some partitions in a global file and other partitions of the same table space into a file per partition.

When you use a TEMPLATE statement to allocate tape output files, the VOLUMES and UNCNT keywords in the TEMPLATE statement are ignored. In this case, scratch volumes are always used.

When you unload LOB or XML data, stacking output files on tape is not supported for CLOBF, DBCLOBF, or BLOBF.

**tape-unit-name**
Specifies the name of the tape unit.

**tape-storage-class-name**
Specifies the name of the tape storage class.

**maximum_devices_number**
Specifies the maximum number of devices associated with a tape unit or storage class that can be allocated to a single Db2 HPU step. Db2 HPU ensures that the maximum number of tape devices is not exceeded by automatically reducing the parallelism degree, if necessary.
Valid values are 1 - 4096.
The default value is 1 for tape units or storage classes that are specified in the TAPEUNIT keyword.

**default_devices_number**
Specifies the number of devices used for an allocation of an output dataset with a given tape unit or storage class, if no explicit unit-count or UNCNT is specified. default_devices_number can not be greater than maximum_devices_number. This subparameter is ignored for an allocation made according to a TEMPLATE statement that specifies a UNCNT option.

Valid values are 0-59.
The default value is 1.

The following examples show how you can use TAPEUNIT.

**Example: Using five tape devices to unload a set of table spaces that are based on the LISTDEF that is specified in SYSIN**
One output file is created for each table that is found in the unloaded table spaces, and the files are stacked on the five tapes that are allocated with the unit name TAPE.

The parallelism degree at the table space level is set to five and is equal to the number of tape drives that are used for output files.

As the output files are stacked sequentially on tapes, if a table space contains several tables, the table space is read once for each table. Partitioned table spaces are read sequentially without partition parallelism (PARALLELISM(1,x,x)), and one single output file is created for each partitioned table space.

```
LISTDEF LSTTB INCLUDE TABLESPACES TABLE MZLFD.*
TEMPLATE FOUT DSN MZLFD.&LIST..&DB..&TS..&SEL.
UNIT TAPE
STACK YES
GLOBAL
TAPEUNIT ( TAPE(5) )
OPTIONS TEMPLATESET(SEL =: SELNUM)
;
UNLOAD TABLESPACE
QUIESCE NO
DB2 NO
PARALLELISM(1,1,5)
SELECT * FROM LIST(LSTTB)
OUTDDN(FOUT)
FORMAT DSNTIAUL
;
```

**Example: Using 10 tape drives to unload 500 partitions of a partitioned table space**
One output file is created for each partition, and the files are stacked on the 10 tapes that are allocated with unit name TAPE.

The MAXPART parameter is automatically forced to 1. The parallelism degree that is used to read the partitions is set by using the third subparameter of the PARALLELISM option (10). However, if a larger value were used for parallelism in this example, the parallelism degree that is used to read the partitions would be limited by the number of available tape devices.

```
TEMPLATE FOUT DSN MZLFD.EX3.&DB..&TS..P&PART.
UNIT TAPE
STACK YES
GLOBAL
TAPEUNIT ( TAPE(10) )
;
UNLOAD TABLESPACE
QUIESCE NO
```

138 Db2 High Performance Unload User's Guide
The TAPEUNIT keyword does not have default values.

**TAPEREP**
Specifies whether to generate a report about the tape usage. For each volume that was used, the report displays the data set names (DSN) and their positions on the tape (FILESEQ).

**YES**
The tape usage report is generated.

**NO**
The tape usage report is not generated.

If you do not specify TAPEREP, the tape usage report is generated according to the setting of the VZM011/TAPERPT PARMLIB parameter.

The default value is YES.

**SQLID sqlid**
Specifies the sqlid used to change the CURRENT SQLID for SELECT processed in SQL mode. This option is only available if all SELECT statements are processed in SQL mode. If the mode is not specified, it is automatically forced to SQL mode (DB2 FORCE).

**APPLCOMPAT**
APPLCOMPAT is a character string used to set the CURRENT APPLICATION COMPATIBILITY special register. The control of the validity of the value is deferred to Db2. Db2 HPU fails if the SET CURRENT APPLICATION COMPATIBILITY fails due to the specified value. If not specified, the value of the application compatibility set via Db2 applies.

**CHECK_INTEGRITY**
Specifies the level of data integrity control to be performed when reading the input data from the LDS of a Db2 object from an Image Copy. It only applies to the native mode:

**BASIC**
Basic controls of broken page indicator and PGCOMB and some other controls are performed.

**LEVEL1**
The same as basic control with additional control for LDS: check if the read page is the expected one.

Caution: higher levels (non-basic) of checking imply a performance decrease.

The default value is the value of VUX038/CHECKINT parmlib variable.

**PROGRESS_MSG_FREQ**
Specifies the frequency of the informational message that displays the current number of unloaded rows for each output file.

**Integer**
A non-null integer value means that this message is issued for each n rows. A null integer means that only one message is issued at the end of processing which indicates the total number of rows for each output file.

The default value is the value of the configuration parameter VUU050/ULFRQMSG

An additional parameter specifies interval time since the previous message was issued or since the first record was unloaded. Specify one of the following value:

**DELTA**
Messages INZX089 and INZX090 are issued to indicate the elapsed time since the previously issued message.
TOTAL
Messages INZX089 and INZX090 are issued to indicate the total elapsed time since the beginning of the unload process.

TOTAL is the default when a non-null value is specified and neither DELTA nor TOTAL is specified. TOTAL and DELTA are mutually exclusive.
The default value is TOTAL.

DATE_DELIM
This option specifies whether the DELIM literal option of FORMAT DELIMITED also applies to DATE, TIME and TIMESTAMP data types.
Valid values are YES or NO.
The default value is the value specified by the VUU031/DTDELIM parmlib parameter.

NULL_DATE_DELIM
This option specifies whether the NULL DELIM option of FORMAT DELIMITED also applies to DATE, TIME and TIMESTAMP data types. It is only applicable when YES is specified for the VUU031/DTDELIM parmlib parameter or for the DATE_DELIM option in the unload command.
Valid values are YES or NO.
The default value is the value specified by the VUU043/DTNULDLDM parmlib parameter.

options block
See “OPTIONS block syntax and description” on page 143.

select block
See “SELECT block syntax and description” on page 183.

Related concepts
Db2 HPU-allocated ddnames: Parallel processing
Db2 HPU dynamically allocates some of the ddnames that are required to run unload jobs.

Related reference
User-allocated ddnames
To run unload jobs, you must allocate certain ddnames in the Db2 HPU JCL.

GLOBAL OPTIONS block syntax and description
A Db2 HPU command may contain several UNLOAD blocks. Use the GLOBAL OPTIONS block to specify default values that apply to all UNLOAD blocks that you specify in your SYSIN. If you specify a keyword or value in both the GLOBAL OPTIONS block and the UNLOAD block, the value in the UNLOAD block is used.

Partition processing with LISTDEF and PARTLEVEL

Example: Unloading a table
In this example, you unload data from a table by using Db2 HPU.

The JCL in this example unloads data from the USER1.TBSCA table in the DBINFDM.TSSCA table space. Before the unload process begins, the table space is quiesced. All rows of the USER1.TBSCA table are unloaded in DSNTIAUL format into the USER1.UNLOAD01 file. A LOAD statement is generated in the SYSPUNCH data set.

```
//UNLOAD EXEC PGM=INZUTILB,PARM='DB81, DB2UNLOAD',
//       REGION=0M, TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
//( // DSN=DSNA10.SDSNL0LOAD,DISP=SHR
// DSN=DSNA10.SDSNL0D2,DISP=SHR
//SYSSREC00 DD DSN=USER1.UNLOAD01,
// DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA,
// SPACE=(CYL, (100, 50), RLSE)
//UTPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSPUNCH DD SYSOUT=*
//SYSIN DD *
//UNLOAD TABLESPACE DBINFDM.TSSCA 
DB2 NO  LOCK NO  QUIESCE YES
SELECT *
```
The JESMSGLG data set contains the following abbreviated information about the unloaded objects:

11.12.30 JOB01736 INZX006 TSSCA TABLESPACE UNLOAD PHASE STARTED
11.18.32 JOB01736 INZX990 SYSREC00 : 5002578 RECORDS WRITTEN
11.18.32 JOB01736 INZU222I SYSREC00 , TOTAL NUMBER OF RECORDS WRITTEN 5002578

The SYSPRINT data set contains the following information about the unloaded objects:

INZU224I IBM DB2 HIGH PERFORMANCE UNLOAD V4.1
INZI175I PROCESSING SYSIN AS EBCDIC.
  -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  000001 UNLOAD TABLESPACE DBINFDM.TSSCA
  000002 DB2 NO LOCK NO QUIESCE YES
  000003 SELECT * FROM TBS
  000004 OUTDDN (SYSREC00 )
  000005 FORMAT DSNTIAUL
  000006 LOADDDN SYSPPUNCH
  DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DB2UNLOAD
  DSNU050I DSNUGUTC - QUIESCE TABLESPACE DBINFDM.TSSCA
  DSNU474I -DZ7M DSNUQUIA - QUIESCE UTILITY COMPLETE, ELAPSED TIME= 00:00:00
INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(3, 1)

The SYSPUNCH data set contains the following SYSIN data that can later be used to reload the objects:

LOAD DATA
LOG NO NOTIFY YES ENFORCE NO
INTO TABLE USER1.TBSCA(
(COLDATE DATE EXTERNAL( 10) , POSITION( 1 )
COLTIME TIME EXTERNAL( 8) , POSITION( 11 )
COLTIMESTAMP TIMESTAMP EXTERNAL( 26) , POSITION( 19 )
COLCHAR CHAR( 8) , POSITION( 45 )
COLVARCHAR VARCHAR , POSITION( 53 )
COLLONGVARCHAR VARCHAR , POSITION( 130 )
COLINTTEGER INTEGER , POSITION( 4044 )
COLSMALLINT SMALLINT , POSITION( 4048 )
COLFLOAT FLOAT( 53) , POSITION( 4050 )
)
The other parts of the report contain information like the information that is shown in “Example: Unloading a table” on page 140.

The following example shows how to unload a partitioned table space with five partitions that have copies per partition. Partitions 2 and 3 are standard copies, and partitions 1, 4, and 5 are FlashCopies.
Examples: Requesting progress messages via the UNLOAD command

Example 1: Request progression messages be issued every 1000 rows. The messages will indicate the elapse time since the beginning of the unload.

```
GLOBAL
PROGRESS_MSG_FREQ (1000) ;
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example 2: Request progression messages be issued every 1000 rows and that indicate the elapse time spent to unload for the latest set of 1000 rows.

```
GLOBAL
PROGRESS_MSG_FREQ (1000, DELTA)
OPTION NULLPAD YES ;
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example 3: Request progression messages be issued every 1000 rows and that indicate the elapse time spent from the beginning of the unload. TOTAL is not mandatory since this is the default value.

```
GLOBAL
PROGRESS_MSG_FREQ (1000, TOTAL) ;
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example: Requesting the input data integrity to be checked with the LEVEL1 controls

```
UNLOAD TABLESPACE
DB2 NO
CHECK_INTEGRITY LEVEL1
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

OPTIONS block syntax and description

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

The effect of the options that are specified in the OPTIONS block at the GLOBAL level or at the UNLOAD level depends on the value that is specified for the VUU057/OPALLFMT PARMLIB parameter:

- If NO is specified for the VUU057/OPALLFMT parameter, all options that you specify in a GLOBAL block or an UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, UNLROWSET, LOADINDDN, AUTOTAG, and IFERROR options. The LOADOPT, UNLROWSET, LOADINDDN, AUTOTAG, and IFERROR options apply to all formats. The FLOAT option applies to DSNTIAUL, USER, and VARIABLE formats.
• If YES is specified for the VUU057/OPALLFMT parameter, all options that you specify in an OPTIONS block at the GLOBAL level or the UNLOAD level apply to all formats except for some specific options that have no meaning for some formats and for LENGTHBYTE and LENGTH options that are specific to the USER format.

The following diagram shows the syntax of the OPTIONS block:
Notes:

1. UNICODE_EXPANSION_RATIO is associated to VUU068/ULUNIEXP parmlib parameter.
2. CONVERSION_TRUNCATION_ALLOWED is associated to the VUU067/ULCNVTRC parmlib parameter.

Table 21. Default values per output format for options that are specified in the OPTIONS block

<table>
<thead>
<tr>
<th>Option name</th>
<th>Applies only to FORMAT USER if OPALLFMT=NO (GLOBAL or UNLOAD levels)</th>
<th>Applies to all formats if OPALLFMT=YES</th>
<th>PARMLIB parameter</th>
<th>Default value for FORMAT DELIMITED</th>
<th>Default value for FORMAT DSNTIAUL</th>
<th>Default value for FORMAT EXTERNAL</th>
<th>Default value for FORMAT USER</th>
<th>Default value for FORMAT VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>Yes</td>
<td>Exceet FORMAT DELIMITED</td>
<td>VUU014/ULNULL</td>
<td>OFF</td>
<td>X'6F' X'00' or VUU014/ULNULL</td>
<td>VUU014/ULNULL</td>
<td>X'6F' X'00' or VUU014/ULNULL</td>
<td></td>
</tr>
<tr>
<td>NULLID</td>
<td>Yes</td>
<td>Exceet FORMAT DELIMITED</td>
<td>None</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>NULLPOS</td>
<td>Yes</td>
<td>Exceet FORMAT DELIMITED</td>
<td>VUU022/NULLPOS</td>
<td>AFTER or VUU022/NULLPOS</td>
<td>AFTER or VUU022/NULLPOS</td>
<td>VUU022/NULLPOS</td>
<td>BEFORE or VUU022/NULLPOS</td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>Yes</td>
<td>VUU015/ULDATE</td>
<td>DATE.C or VUU015/ULDATE</td>
<td>DATE_DB2 or VUU015/ULDATE</td>
<td>DATE_DB2 or VUU015/ULDATE</td>
<td>VUU015/ULDATE</td>
<td>VUU015/ULDATE</td>
<td></td>
</tr>
<tr>
<td>DATEDELIM</td>
<td>Yes</td>
<td>VUU055/DATEDEL</td>
<td>' ' or VUU055/DATEDEL</td>
<td>Delimiter of the default Db2 format or VUU055/DATEDEL</td>
<td>Delimiter of the default Db2 format or VUU055/DATEDEL</td>
<td>VUU055/DATEDEL</td>
<td>' ' or VUU055/DATEDEL</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>Yes</td>
<td>VUU016/ULTIME</td>
<td>TIME_A or VUU016/ULTIME</td>
<td>TIME_DB2 or VUU016/ULTIME</td>
<td>TIME_DB2 or VUU016/ULTIME</td>
<td>VUU016/ULTIME</td>
<td>TIME_A or VUU016/ULTIME</td>
<td></td>
</tr>
<tr>
<td>TIMEDELIM</td>
<td>Yes</td>
<td>VUU056/TIMEDEL</td>
<td>' ' or VUU056/TIMEDEL</td>
<td>Delimiter of the default Db2 format or VUU056/TIMEDEL</td>
<td>Delimiter of the default Db2 format or VUU056/TIMEDEL</td>
<td>VUU056/TIMEDEL</td>
<td>' ' or VUU056/TIMEDEL</td>
<td></td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>Yes</td>
<td>VUU017/ULTMSTP</td>
<td>TIMESTAMP_B or VUU017/ULTMSTP</td>
<td>TIMESTAMP_B or VUU017/ULTMSTP</td>
<td>TIMESTAMP_B or VUU017/ULTMSTP</td>
<td>VUU017/ULTMSTP</td>
<td>VUU017/ULTMSTP</td>
<td></td>
</tr>
<tr>
<td>PIC</td>
<td>Yes</td>
<td>VUU038/ULPIC</td>
<td>-LEAD. or VUU038/ULPIC</td>
<td>-LEAD. or VUU038/ULPIC</td>
<td>-LEAD. or VUU038/ULPIC</td>
<td>VUU038/ULPIC</td>
<td>-LEAD. or VUU038/ULPIC</td>
<td></td>
</tr>
<tr>
<td>Mask part of the PIC option</td>
<td>Yes</td>
<td>VUU032/ULMASK</td>
<td>0.0 or VUU032/ULMASK</td>
<td>0.0 or VUU032/ULMASK</td>
<td>0.0 or VUU032/ULMASK</td>
<td>VUU032/ULMASK</td>
<td>0.0 or VUU032/ULMASK</td>
<td></td>
</tr>
<tr>
<td>LOADOPT</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>VUU019/ULOPTLDT and VUU020/ULOPTLDP</td>
<td>VUU019/ULOPTLDT and VUU020/ULOPTLDP</td>
<td>VUU019/ULOPTLDT and VUU020/ULOPTLDP</td>
<td>VUU019/ULOPTLDT and VUU020/ULOPTLDP</td>
<td></td>
</tr>
<tr>
<td>LENGTHBYTE</td>
<td>Yes</td>
<td>No</td>
<td>NO</td>
<td>REAL</td>
<td>MAX</td>
<td>MAX</td>
<td>Depends on LENGTHBYTE</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td>Yes</td>
<td>No</td>
<td>NO</td>
<td>START</td>
<td>MAX</td>
<td>MAX</td>
<td>MAX</td>
<td></td>
</tr>
<tr>
<td>LOADNODON</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>NO or contents of VUU030/ULOPTNS</td>
<td>NO or contents of VUU030/ULOPTNS</td>
<td>NO or contents of VUU030/ULOPTNS</td>
<td>NO or contents of VUU030/ULOPTNS</td>
<td></td>
</tr>
<tr>
<td>FLOAT</td>
<td>No</td>
<td>N/A</td>
<td>5390</td>
<td>5390</td>
<td>5390</td>
<td>5390</td>
<td>5390</td>
<td></td>
</tr>
<tr>
<td>UNLRSWSET</td>
<td>No</td>
<td>N/A</td>
<td>VUU035/UNLRSWSET</td>
<td>VUU035/UNLRSWSET</td>
<td>VUU035/UNLRSWSET</td>
<td>VUU035/UNLRSWSET</td>
<td>VUU035/UNLRSWSET</td>
<td></td>
</tr>
<tr>
<td>NULLPAD</td>
<td>Yes</td>
<td>No</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>PADDING</td>
<td>Yes</td>
<td>None</td>
<td>Space</td>
<td>Space for characters or graphic strings or '00' for other data types</td>
<td>Space for characters or graphic strings or '00' for other data types</td>
<td>Space for characters or graphic strings or '00' for other data types</td>
<td>Space</td>
<td>Space</td>
</tr>
</tbody>
</table>
Table 21. Default values per output format for options that are specified in the OPTIONS block (continued)

<table>
<thead>
<tr>
<th>Option name</th>
<th>Applies only to FORMAT USER if OPALLFMT=NO (GLOBAL or UNLOAD levels)</th>
<th>Applies to all formats if OPALLFMT=YES</th>
<th>PARMLIB parameter</th>
<th>Default value for FORMAT DELIMITED¹</th>
<th>Default value for FORMAT DSNTIAUL¹</th>
<th>Default value for FORMAT EXTERNAL¹</th>
<th>Default value for FORMAT USER¹</th>
<th>Default value for FORMAT VARIABLE¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTOTAG</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>VUU046/ULAUTAG</td>
<td>VUU046/ULAUTAG</td>
<td>VUU046/ULAUTAG</td>
<td>VUU046/ULAUTAG</td>
<td>VUU046/ULAUTAG</td>
</tr>
<tr>
<td>IFERROR</td>
<td>No</td>
<td>N/A</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>TRIM</td>
<td>Yes</td>
<td>Yes</td>
<td>VUU059/ULTRIM</td>
<td>NO or VUU059/ULTRIM</td>
<td>NO or VUU059/ULTRIM</td>
<td>VUU059/ULTRIM</td>
<td>VUU059/ULTRIM</td>
<td>VUU059/ULTRIM</td>
</tr>
<tr>
<td>SPANNED</td>
<td>NO</td>
<td>N/A</td>
<td>N/A</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>ENFORCE_COLN_N_ORDER</td>
<td>NO</td>
<td>N/A</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Notes:

1. When a value or a parameter name is specified in the table, you can define a default value for each format in the corresponding PARMLIB parameter. If the default value is not defined in the PARMLIB for the format, the hard-coded default value applies.

2. When a variable is specified for the USER format, the default value is the content of the corresponding PARMLIB parameter unless a hard-coded default value is specified in the table.

3. The default value applies to the DSNTIAUL and VARIABLE formats when a LIKE keyword that forces conversion to a CHAR data type is used or when an INTO clause or REFORMAT clause is used.

**NULL**

Indicates whether the null indicator is generated in the output data set. You can specify NULL in the SELECT statement in the FORMAT USER syntax also.

The following values can be specified for NULL. Although value-1 and value-2 can be specified as 1 - 4 bytes or characters, you must specify both value-1 and value-2 as the same number of bytes or characters. You can specify value-1 and value-2 in character ('c') or hexadecimal ('X'hh') formats.

- **value-1**
  - The value of the null indicator when the column value is NULL.

- **value-2**
  - The value of the null indicator when the column value is NOT NULL.

**OFF**

A null indicator is not generated.

When FORMAT DELIMITED is specified, the default value is OFF. You cannot change this value in the OPTIONS block even when it is specified at the SELECT level.

The default values for other formats are described in #unique_95/unique_95_Connect_42_defval.

**DATE DATE_x**

Specifies the default output format for the DATE columns, where x is any uppercase alphabetic character A - R or an integer 0 - 7 or DB2.

When a date column is converted into the CHAR format by specifying FORMAT USER or by using either a REFORMAT clause or an INTO clause, the DATE format that is specified in the OPTIONS block is not used. Instead, the default date format of your environment is used.

The default values for formats are shown in #unique_95/unique_95_Connect_42_defval.

You can override the DATE option at the column level with the TYPE keyword in a SELECT statement for the USER format, an INTO clause, or a REFORMAT clause.

**DATE datefmt**

datefmt is a string that describes how a date value must be formatted. It must be specified according to the rules indicated in section Syntax rules for the user-defined date, time or timestamp formats.

**DATEDELIM value**

Specifies the default delimiter that is used in external date representations, where value must be one character and must be 1 byte long, regardless of the literal CCSID.
The default delimiter that is used when DATE columns are unloaded depends on the output format that is used, as shown in #unique_95/unique_95_Connect_42_defval

**TIME**

`TIME_x` Specifies the default conversion for time representations, where x is any uppercase alphabetic character A - E or the integer 0 or DB2.

When a time column is converted into a CHAR data type by specifying FORMAT USER or by using either a REFORMAT clause or an INTO clause, the time format that is specified in the OPTIONS block is not used. Instead, the default time format of your environment is used.

The default values for formats are shown in #unique_95/unique_95_Connect_42_defval.

You can override the TIME option at the column level by specifying the TYPE keyword in a SELECT statement for the USER format, an INTO clause, or a REFORMAT clause.

**TIME timefmt**

timefmt is a string that describes how a time value must be formatted. It must be specified according to the rules indicated in section Syntax rules for the user-defined date, time or timestamp formats.

**TIMEDELM value**

Indicates the default delimiter that is used in external time representations, where value must be one character and must be 1 byte long, regardless of the literal CCSID.

The default values for other formats are described in #unique_95/unique_95_Connect_42_defval.

**TIMESTAMP**

`TIMESTAMP_x` Specifies the default conversion for the TIMESTAMP columns, where x is an uppercase alphabetic character A - G or the integer 0.

When a timestamp column is converted into a CHAR data type by specifying FORMAT USER or by using either a REFORMAT clause or an INTO clause, the timestamp format that is specified in the OPTIONS block is not used. Instead, the Db2 standard timestamp external format is used.

The default values for formats are shown in #unique_95/unique_95_Connect_42_defval.

**TIMESTAMP timestampfmt**

timestampfmt is a string that describes how a timestamp value must be formatted. It must be specified according to the rules indicated in section Syntax rules for the user-defined date, time or timestamp formats.

**Syntax rules for the user-defined date, time or timestamp formats**

A user-defined date format (datefmt), time format (timefmt) or timestamp format (timestampfmt) must be specified according to the following rules.

1. The format string must begin with one of the allowed format elements listed in the table below
2. The format string can contain only elements allowed for the related format
3. The format string can contain only a subset of the elements allowed for the related format
4. Each format element can be specified only once
5. Separators can be any character or character string but must not contain any character string used as format elements as well as the ambiguous MM string, regardless of the related format.

The data is formatted by replacing the format elements as indicated in the table below and by keeping the other characters from the string format, i.e. the separators, as specified. If necessary the whole output field, including separators, is converted into the output CCSID.

Table: Format elements for a format string
Table 22. Format elements for a format string

<table>
<thead>
<tr>
<th>format element</th>
<th>the element is replaced by</th>
<th>allowed in datefmt (1)</th>
<th>allowed in timefmt (1)</th>
<th>allowed in timestampfmt (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD</td>
<td>two digits corresponding to the day part (2)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>MO</td>
<td>two digits corresponding to the month part (2)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>YY(3)</td>
<td>the least significant two digits of the year part (2)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>YYYY(3)</td>
<td>four digits corresponding to the year part (2)</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>HH</td>
<td>two digits corresponding to the hours part (2) with a time coded over 24 hours. N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MI</td>
<td>two digits corresponding to the minutes part (2)</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SS</td>
<td>two digits corresponding to the seconds part (2)</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>NN(4)</td>
<td>two digits corresponding to the fractional seconds part (5)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>NNNN(4)</td>
<td>four digits corresponding to the fractional seconds part (5)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>NNNNNN(4)</td>
<td>six digits corresponding to the fractional seconds part (5)</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

- (1) any element indicated as not allowed (N) is forbidden for the related for the related format, even as a separator.
- (2) leading zeroes are added if necessary.
- (3) only one element among YY and YYYY can be specified.
- (4) only one element among NN and NNNN and NNNNNN can be specified.
- (5) trailing zeroes are added if necessary.
**PIC**

Defines the numeric data display format that is used when numeric values are converted for external representation. The PIC keyword has four parameters. You must specify the first three parameters. The fourth parameter is optional.

- The first parameter specifies the rules for printing the sign:
  - Indicates that the minus character (-) is present if the number is negative. Otherwise, the sign character is a blank.
  - Indicates that the sign is always present. Positive values have a plus character (+), and negative values have a minus character (-).
  - Indicates that the padding character is used for positive values and the minus character (-) is used for negative values.

- The second parameter specifies the position the sign relative to the column.
  - **LEAD** The sign is placed in front of the numeric value. LEAD is ignored for floating point numbers.
  - **TRAIL** The sign is placed after the numeric value. TRAIL is ignored for floating point numbers.

- The third parameter specifies the decimal separator.
  - Use a period as the decimal separator.
  - Use a comma as the decimal separator.

- The optional fourth parameter indicates the formatting rules that are used to display non-significant zeros for DECIMAL, SMALLINT, and INTEGER values.

  The following table lists the valid values and explains their impact:

  - The Left padding column indicates whether the number is padded on the left with non-significant zeros to fill the output field. The sign character is added to the left of these zeros.
  - The Left zero column indicates whether a zero is displayed on the left of the decimal separator when the integer part of the number is 0.
  - The Decimal separator column indicates whether the decimal separator is displayed.
  - The Right padding column indicates whether the number is padded on the right with zeros up to the number of digits of the decimal scale.

**Notes:**

- The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The ULMASK variable has no impact on the formatting of values for FLOAT.
Table 23. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>.*</td>
<td>No</td>
<td>Only when the value is 0</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>0.*</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>*.0</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>0.0</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>00.0</td>
<td>Yes</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>*.Z</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>0.Z</td>
<td>No</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>00.Z</td>
<td>Yes</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The following table contains examples of the effect of the mask parameter when the following decimal values are converted:

<table>
<thead>
<tr>
<th>Mask/value</th>
<th>001.40</th>
<th>000.40</th>
<th>123.00</th>
<th>-123.45</th>
<th>000.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>.*</td>
<td>1.4</td>
<td>.4</td>
<td>123</td>
<td>-123.45</td>
<td>0</td>
</tr>
<tr>
<td>0.*</td>
<td>1.4</td>
<td>0.4</td>
<td>123</td>
<td>-123.45</td>
<td>0</td>
</tr>
<tr>
<td>*.0</td>
<td>1.40</td>
<td>.40</td>
<td>123.00</td>
<td>-123.45</td>
<td>.00</td>
</tr>
<tr>
<td>0.0</td>
<td>1.40</td>
<td>.40</td>
<td>123.00</td>
<td>-123.45</td>
<td>0.00</td>
</tr>
<tr>
<td>00.0</td>
<td>0001.40</td>
<td>0000.40</td>
<td>0123.00</td>
<td>-0123.45</td>
<td>0000.00</td>
</tr>
<tr>
<td>*.Z</td>
<td>1.40</td>
<td>.40</td>
<td>123.00</td>
<td>-123.45</td>
<td>.00</td>
</tr>
<tr>
<td>0.Z</td>
<td>1.40</td>
<td>0.40</td>
<td>123.00</td>
<td>-123.45</td>
<td>0.00</td>
</tr>
<tr>
<td>00.Z</td>
<td>0001.40</td>
<td>0000.40</td>
<td>0123.00</td>
<td>-0123.45</td>
<td>0000.00</td>
</tr>
</tbody>
</table>

The following table contains examples of the effect of the mask parameter when decimal values are converted:

<table>
<thead>
<tr>
<th>Mask/value</th>
<th>12345.</th>
<th>00001.</th>
<th>-12345.</th>
<th>00000.</th>
</tr>
</thead>
<tbody>
<tr>
<td>.*</td>
<td>12345</td>
<td>1</td>
<td>-12345</td>
<td>0</td>
</tr>
<tr>
<td>0.*</td>
<td>12345</td>
<td>1</td>
<td>-12345</td>
<td>0</td>
</tr>
<tr>
<td>*.0</td>
<td>12345</td>
<td>1</td>
<td>-12345</td>
<td>0</td>
</tr>
</tbody>
</table>
The following table contains examples of the effect on the mask parameter when decimal values are converted:

### DEC(5,5) converted to a CHAR(9)

<table>
<thead>
<tr>
<th>Mask/value</th>
<th>12345</th>
<th>00001</th>
<th>-12345</th>
<th>00000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>12345</td>
<td>1</td>
<td>-12345</td>
<td>0</td>
</tr>
<tr>
<td>00.0</td>
<td>0012345</td>
<td>0000001</td>
<td>-0012345</td>
<td>0000000</td>
</tr>
<tr>
<td>*.Z</td>
<td>12345</td>
<td>1</td>
<td>-12345</td>
<td>0</td>
</tr>
<tr>
<td>0.Z</td>
<td>12345</td>
<td>1</td>
<td>-12345</td>
<td>0</td>
</tr>
<tr>
<td>00.Z</td>
<td>012345</td>
<td>000001</td>
<td>-012345</td>
<td>000000</td>
</tr>
</tbody>
</table>

The default value of the PIC keyword that is used depends on the output format that is used, as shown in #unique_95/unique_95_Connect_42_defval.

**LOADOPT**

Modifies the options of the Db2 LOAD command. Specify the options that you want Db2 HPU to place in the LOAD SYSIN that is created during the unload process.

The value of the LOADOPT keyword is created by merging values that are specified in the PARMLIB, the GLOBAL block, the UNLOAD block, and the SELECT block. However, if LOADOPT is also specified in the FORMAT specification, the LOADOPT keyword is used as is. It is not merged with previous levels.

- **tableoptions**
  Options to specify at the table level

- **partoptions**
  Options to specify at the partition level

- **tablespaceoptions**
  Options to specify at the table space level

The variables `tablespaceptions`, `partoptions` and `tableoptions` values are positional, and the commas are required even if you do not want to set the value for one subparameter. For example, to specify the value NUMRECS for tableoptions, specify LOADOPT (,,NUMRECS).

The following syntax diagram shows the LOADOPT keywords that are recognized by Db2 HPU:
**loadopt keywords**

- **KEEPDICTIONARY**
- **NOKEEPDICTIONARY**
- **RESUME NO**
- **RESUME NO SHRLEVEL NONE**
- **RESUME NO SHRLEVEL NONE REPLACE**
- **RESUME NO REPLACE SHRLEVEL NONE**
- **RESUME NO REPLACE**
- **RESUME YES**
- **RESUME YES SHRLEVEL NONE**
- **RESUME YES SHRLEVEL CHANGE**

- **SORTKEYS**
  - integer
  - &SORTKEYS

- **ENFORCE NO**
- **ENFORCE CONSTRAINTS**

- **CCSID**
  - (integer)

- **LOG NO**
- **LOG YES**

- **PARALLEL**
  - (num-subtask)

- **NUMRECS**
  - &NUMRECS
  - integer

**Notes:**

1. &SORTKEYS is replaced with the actual value after data is unloaded except when EXECUTE NO is specified. When EXECUTE NO is specified, &SORTKEYS is replaced with 0.
2. Repeat up to three times.
3. only recognized as part of a tableoptions block.

**Important:** Make sure to specify these keywords at the intended and applicable level (tablespaceoptions, partoptions or tableoptions). Refer to the Db2 publication for details on applicability and meaning of these keywords.

The load options in the PARMLIB and any OPTIONS blocks that you specify in the GLOBAL block or the UNLOAD block are merged. If you specify the same option in two or more places, they are merged with the following priority:

1. UNLOAD block
2. GLOBAL block
3. PARMLIB

Any keywords that you specify that are not shown in the LOADOPT syntax diagram are added to the total option sequence. They are not merged with other options.

**Important:** If a LOADOPT keyword is specified in the FORMAT block, it is used as is, and it is not merged.

**Note:** If the Db2 HPU spanned YES option is specified, do not specify the FORMAT keyword in the LOADOPT option as this can lead to a wrong LOAD statement notified at LOAD time by either a syntax error message or by an error during the load processing.
**LOADDDL TABLE**

Requests the CREATE TABLE statement of the unloaded tables be written into the data set specified by the LOADDN option. Use this option to get the CREATE TABLE statement of all the tables involved in a given load command, inserted before its related load command. The CREATE TABLE statement describes all the columns of the unloaded tables whether they are involved in the SELECT statement or not. If the LOAD command contains several tables, the content of the LOADDN data set looks as follows:

```sql
CREATE TABLE me.mytable_1_in_LOAD_COMMAND_1
  ...
CREATE TABLE me.mytable_2_in_LOAD_COMMAND_1
  ...
LOAD DATA
  ...
INTO TABLE me.mytable_1_in_LOAD_COMMAND_1
  ...
INTO TABLE me.mytable_2_in_LOAD_COMMAND_1
  ...
CREATE TABLE me.mytable_3_in_LOAD_COMMAND_2
  ...
CREATE TABLE me.mytable_4_in_LOAD_COMMAND_2
  ...
LOAD DATA
  ...
INTO TABLE me.mytable_3_in_LOAD_COMMAND_2
  ...
INTO TABLE me.mytable_4_in_LOAD_COMMAND_2
  ...
CREATE TABLE me.mytable_n_in_LOAD_COMMAND_N
  ...
CREATE TABLE me.mytable_m_in_LOAD_COMMAND_N
  ...
LOAD DATA
  ...
INTO TABLE me.mytable_n_in_LOAD_COMMAND_N
  ...
INTO TABLE me.mytable_m_in_LOAD_COMMAND_N
```

where `me.mytable_n_in_LOAD_COMMAND_N` is one of the tables involved in the Nth load command.

**Note:** In SQL mode, the CREATE TABLE statement is only generated when the table can be identified from the SELECT statement. For example, for a SELECT statement which contains a table JOIN, no CREATE TABLE statement is generated.

**Note:** The CREATE TABLE DDL is only generated when the unload refers to a table, not a view, a Materialized Query Table or any other object. The column TYPE from the CATALOG table SYSTABLES must be equal to 'T'.

**Note:** The generated CREATE TABLE contains fieldproc, editproc and partitioned clause for tablespace partitioned by the table.

**Note:** The default value for graphical column is not generated in the DDL.

**LENGTHBYTE**

Specifies whether the two length bytes for variable-length columns are written to the output data set. LENGTHBYTE applies only when FORMAT USER is specified. It is ignored when other output formats are specified.

**YES**

Writes the two length bytes.

**NO**

Does not write the two length bytes.

The default value is YES when FORMAT USER is specified. You can change the default value by specifying an OPTIONS block at the GLOBAL or SELECT level or by changing LENGTHBYTE for a specific column in the FORMAT USER definition.

The following rules apply when you specify FORMAT USER:
• When LENGTHBYTE YES is specified, the default value for the LENGTH keyword is REAL.
• When LENGTHBYTE NO is specified, the default value for the LENGTH keyword is MAX.
• When LENGTHBYTE NO and LENGTH REAL are specified, Db2 HPU cannot generate a LOADDNN and issues the INZU203E error message if a LOADDN file was requested.

When you specify FORMAT DELIMITED, the default value is NO. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

When you specify FORMAT DSNTIAL, FORMAT EXTERNAL, or FORMAT VARIABLE, the default value is YES. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

LENGTH
Specifies whether to use the real or maximum length for variable-length fields. This keyword applies only for variable-length fields and for the USER and DSNTIAL output formats. It is ignored when other output formats are specified.

REAL
The length of the field does not change (value of the two length bytes).

MAX
The output field is padded to its maximum length by adding binary zeros.

When you specify FORMAT USER, the default value for the LENGTH keyword depends on the setting of the LENGTHBYTE keyword. You can change the default value for the USER format by specifying an OPTIONS block at the GLOBAL or SELECT level or by changing the LENGTH keyword for a specific column in the FORMAT USER definition.

When you specify FORMAT DSNTIAL, the default value is MAX. You can change the default value for the DSNTIAL format by specifying an OPTIONS block at the SELECT level.

When you specify FORMAT VARIABLE or FORMAT DELIMITED, the default value is REAL. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

When you specify FORMAT EXTERNAL, the default value is MAX. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

NULLID
Specifies whether a null indicator byte is to be added at the beginning of an output field. You can also specify NULLID in the SELECT statement in the FORMAT USER syntax.

YES
The null indicator is created. If the column is null, the indicator is X'FF. Otherwise, the indicator is X'00'. The Db2 LOAD command uses the indicator to load null values into a table. You can change the values of the null indicator by specifying the NULL keyword.

NO
The null indicator is not created.

The default values for formats are shown in #unique_95/unique_95_Connect_42_defval.

NULLPOS
Specifies the position of the NULL indicator. You can also specify NULLPOS in the SELECT statement in the FORMAT USER syntax. This parameter does not apply to a LOB column unloaded in a spanned format (SPANNED YES option) in which case the NULL indicator is written before the output field (containing the data) whatever the value of NULLPOS.

BEFORE
The null indicator is placed before the field containing the related data.

AFTER
The null indicator is placed after the field containing the related data.
The null indicators are gathered at the front of each unloaded record. The sequence of the NULL indicators is the same as the sequence of the related columns.

The default value of the NULLPOS option that is used depends on the requested output format, as shown in #unique_95/unique_95_Connect_42_defval.

LOADINDDN

Specifies whether INDDN ddname cards are automatically inserted into the Db2 LOAD command according to the ddnames specified by the UNLOAD command for the data sets containing the unloaded data.

YES

INDDN ddname cards are automatically generated. If a TEMPLATE is used to allocate the output file, Db2 HPU also generates a TEMPLATE statement to allocate the ddname used by the INDDN cards generated by the LOADINDDN(YES) feature. To avoid any invalid duplicate specification of INDDN cards, any INDDN option specified via the tableoptions of a LOADOPT option from the OPTIONS block or via the “Options for global LOAD statement (VUU019/ULOPTLDT)” parmlib parameter is ignored.

NO

No INDDN card is automatically generated. This allow to disable a LOADINDDN(YES) setting made at the parmlib level and, subsequently, to ensure INDDN option specified via the tableoptions of a LOADOPT option from the OPTIONS block or via the “Options for global LOAD statement (VUU019/ULOPTLDT)” parmlib parameter are taken into account.

The default value is either NO or the value that is specified in the VUU030/ULOPTNS parameter, with parameter LOADINDDN.

FLOAT

Specifies the output format of the numeric floating point data. The FLOAT keyword applies only to the binary output format.

S390

Indicates that the binary floating point data is written to the output records in the S/390® internal format (hexadecimal floating point or HFP).

IEEE

Indicates that the binary floating point data is written to the output records in the IEEE format (binary floating point or BFP).

The default value is S390.

Restriction: The same output format is used for all FLOAT columns in one SELECT statement. You cannot specify the FLOAT keyword at the column level in the USER format.

UNLROWSET

Specifies the number of rows that a single SQL FETCH statement retrieves. Specifying the number of rows to retrieve improves the performance of SELECT statements that are processed by Db2 when you specify DB2 FORCE or DB2 YES with SELECT statements that are not supported by Db2 HPU. Valid values are 1 - 32767. When you specify 1, a single-row FETCH statement is used instead of a multiple-row FETCH statement. The multiple-row FETCH statement is available only in Db2 Version 8 NFM or later. The UNLROWSET keyword is ignored for earlier Db2 versions.

The default value is the value of the VUU035/ULROWSET parameter.

NULLPAD

Specifies whether the padding character is used to fill the output field when a null value is unloaded.

NO

When a null value is unloaded, the output field is filled with binary zeros (x’00’).

YES

When a null value is unloaded, the output field is filled with the padding character.
The default values for formats are shown in #unique_95/unique_95_Connect_42_defval.

When you specify FORMAT DELIMITED, the value of the NULLPAD keyword is always YES, and you cannot change it.

**AUTOTAG size**

Allows you to add a tag, which is an external numeric value, to each record that is unloaded. The output data for each SELECT statement has its own tag. If you specify AUTOTAG at the GLOBAL level or at the UNLOAD level, the AUTOTAG keyword applies to all SELECT statements if they are written in a common file.

If you specify AUTOTAG at the SELECT level, Db2 HPU requires all SELECT statements that write in a common output file to use the same number of digits. File records with an autotag cannot be mixed with file records without an autotag.

The generated tag starts at 1 and is automatically incremented for each SELECT statement (either explicitly coded or generated by a LISTDEF) of the same SYSIN for which AUTOTAG is specified, either explicitly coded or generated by a LISTDEF or LISDEFTBV.

The AUTOTAG keyword applies only to logical unloads, which use SELECT statements to unload Db2 data. It does not apply to physical unloads.

**size**

Specifies the number of digits in the tag. Valid values are 0 - 8. If you specify 0, an AUTOTAG is not generated for the corresponding SELECT statements.

Important: If the value of the tag that is computed for a SELECT statement exceeds the number of digits that you specified for the AUTOTAG keyword, Db2 HPU stops.

If you specify AUTOTAG and LOADDDN is requested, the LOADDDN file that is generated automatically includes a WHEN clause for each SELECT statement to reload into the corresponding tables. In this case, the unloaded tables correspond with the value of the autotag.

The default value is the value of the VUU046/ULAUTAG parameter.

**IFERROR**

Defines the value that is assigned to an output field if a conversion error occurs while the output field is formatted or if an error occurs while an expression that is natively processed by Db2 HPU is evaluated. The IFERROR keyword applies only to conversion or formatting that was made during the unload process. The IFERROR keyword does not apply to conversion errors when constants are specified in the SELECT statement and these constants are converted into a different data type by using an INTO clause, a REFORMAT clause, or by specifying FORMAT USER. These constants are converted during the initialization process. If an error occurs during this process, a syntax error is issued, and processing stops.

The value that you specify applies to all output fields for the corresponding SELECT statements.

If an expression error for a SELECT statement that is processed by Db2 occurs, Db2 issues an SQL error message, and the default value does not apply.

**DEFAULT**

The default value is assigned. This value depends on the data type.

**NULL_OR_DEFAULT**

If the output field allows a null value, the NULL value is assigned to the field. Otherwise, the default value is used.

**TRIM**

Allows you to suppress the trailing blank characters when data is unloaded into VARCHAR, VARGRAPHIC, output fields, or CLOB, and DBCLOB output fields expect if SPANNED YES is used. The length of the output field is adjusted to match the effective number of characters that are written.

If you specify FORMAT DELIMITED, the TRIM keyword applies to output fields with CHAR and GRAPHIC data types.

If you specify the TRIM keyword at the GLOBAL level, it applies only to the USER format unless YES is specified for the VUU057/OPALLFMT PARMLIB parameter. If you specified YES for the VUU057/
OPALLFMT PARMLIB parameter or if you specify the TRIM keyword at the SELECT level, the TRIM keyword applies to any output format.

The TRIM keyword does not apply when numeric data is converted to numeric external.

The default value is the value of the VUU059/ULTRIM PARMLIB parameter.

**PADDING**

Specifies the padding character that Db2 HPU uses at the column level or the record level.

**Column**

The padding character is used to pad the null values when you specify NULLPAD YES to pad the output field when the specified value is shorter than the field or to pad numeric data that is converted to character data.

**Record**

The padding character is used to pad the end of the output record up to the length of the physical record when the format of the DCB (RECFM) is fixed and when the output record is shorter than the LRECL.

The following syntaxes are supported:

**PADDING (SBCS padding [, DBCS padding [, record padding ]])**

Specifies the padding characters for SBCS and DBCS characters when a character string is converted to a longer character string and when the generated record is padded to its specified length.

**SBCS padding**

The default value is ' ' (an EBCDIC SBCS blank character). You can specify the value in the following ways:

- 'c'
  - Specifies an EBCDIC SBCS character. This character can be converted to match the output CCSID.
- X'hh'
  - Specifies an SBCS character in hexadecimal format. This character is used as-is, regardless of the output CCSID.

**DBCS padding**

The default value is the DBCS space character in the output CCSID. To obtain this DBCS space character, the UTF16 character UX'3000' is converted into the output CCSID. You can specify the value in the following ways:

- X'hhhh'
  - Specifies a DBCS character in a 2-byte hexadecimal format. This character is used as-is, regardless of the output CCSID.
- G'SI_graphic_char_SO'
  - Specifies a graphic character as a graphic constant that is expressed in EBCDIC. Shift-out/shift-in bytes are included. This character can be converted to match the output CCSID.
- GX'hhhh'
  - Specifies a graphic character as a hexadecimal EBCDIC DBCS constant. This character can be converted to match the output CCSID.
- UX'hhhh'
  - Specifies a graphic character as a hexadecimal UTF16 Unicode constant. This character can be converted to match the output CCSID.

**Record padding**

Depending on the output format that is used to unload the data, the default value can be either of the following values:

- X'00' for the DSNTIAUL format and the EXTERNAL format,
- The SBCS blank character that was converted in the output CCSID for all other formats.

You can specify the value in the following ways:
'c'
  Specifies an EBCDIC SBCS character. This character can be converted to match the output CCSID.

X'hh'
  Specifies an SBCS character in hexadecimal format. This character is used as-is, regardless of the output CCSID.

If you do not specify the PADDING keyword, the following values are default values for padding at the character level and the record level:

**SBCS characters**
- X'00' for variable-length columns in the DSNTIAUL format and the EXTERNAL format,
- The SBCS blank character that was converted to match the output CCSID for all other formats and, if padding at the column level is not specified, for columns that use an INTO clause or a REFORMAT keyword.

**DBCS characters**
  UTF16 value UX'3000' that was converted to match the output CCSID

**Record padding character**
- X'00' for the DSNTIAUL format and the EXTERNAL format,
- The SBCS blank character that was converted to match the output CCSID for all other formats.

**Important:** When you specify FORMAT DSNTIAUL or FORMAT EXTERNAL, PADDING () does not give the same result as omitting the PADDING keyword because the SBCS padding is x'00' when the PADDING keyword is omitted, and it is the SBCS blank character when the PADDING keyword is specified.

**REFORMAT**
Defines the default reformatting for all columns that correspond to a specific Db2 definition.

**internal-data-type**
  Specifies a Db2 data type as it is specified in the CREATE TABLE statement. For columns that allow a length or length and scale specification, such as CHAR or DECIMAL columns, you can specify the data type with or without the length. Specifying the data type without the length defines a default conversion for all columns with this data type. If you specify a data type plus a length, such as CHAR(8), the default conversion applies only to columns that are defined with this precise data type. The following diagram shows the syntax of the internal data type:

**internal-data-type**
output-data-type
The following syntax diagram shows the output data type:

output-data-type
The DATE, TIME, and TIMESTAMP data types without the EXTERNAL keyword correspond to the internal DATE/TIME/TIMESTAMP format, and DATE_x, TIME_x, and TIMESTAMP_x correspond to the formats that are defined in the OPTIONS description.

*column-format-option*

Use the column format option block in the REFORMAT clause of the OPTIONS block, in the FORMAT USER statement, or in the INTO clause of the SELECT statement. The following diagram shows the syntax of the column format option:

*column-format-option*
For example, you can use the REFORMAT keyword to specify that you want all CLOB columns that are processed are changed into CLOBF columns by using a specific TEMPLATE statement without having to define it in an INTO clause or in a FORMAT USER statement. You can also use the REFORMAT keyword to convert all DECIMAL(10,0) columns in INTEGER format.
The **REFORMAT** keyword is especially useful to make format conversions when you use **LISTDEF** or **LISTDEFTBV** because the INTO clause is not supported in such a case and using the USER format is either not possible.

The formatting options that are specified in the **REFORMAT** keyword are overridden by the options that are defined in the INTO clause of the fast select block by the LIKE keyword that is used in the DSNTIAUL format or the VARIABLE format and by the conversion option that is defined in the USER format.

**PADDING padding-character**

Specifies the padding character, where **padding-character** is the padding character to be used when padding is required for a column.

Padding is used to convert characters to a longer string. Padding is typically applied to the end of character strings. If you specify JUST RIGHT, padding is added at the beginning of the string.

For SBCS data, you can specify the padding character in the following ways:

- **'c'**
  - Specifies an EBCDIC SBCS character. This character can be converted to match the output CCSID.
- **X'hh'**
  - Specifies an SBCS character in hexadecimal format. This character is used as-is, regardless of the output CCSID.

For DBCS data, you can specify the padding character in the following ways:

- **X'hhhh'**
  - Specifies a DBCS character in a 2-byte hexadecimal format. This character is used as-is, regardless of the output CCSID.
- **G'SI_graphic_char_SO'**
  - Specifies a graphic character as a graphic constant that is expressed in EBCDIC. Shift-out/shift-in bytes are included. This character can be converted to match the output CCSID.
- **GX'hhhh'**
  - Specifies a graphic character as a hexadecimal EBCDIC DBCS constant. This character can be converted to match the output CCSID.
- **UX'hhhh'**
  - Specifies a graphic character as a hexadecimal UTF16 Unicode constant. This character can be converted to match the output CCSID.

You must specify a single-byte character (‘c' or X'hh') for SBCS data and a double-byte character (G'SI_graphic_char_SO' or GX'hhhh' or UX'hhhh') for DBCS data. Any other combination is rejected.

The following examples show how you can specify padding:

- **PADDING '***
- **PADDING X'00'** (binary zero)

For SBCS data, the default padding character is an SBCS blank character that is converted in the output CCSID, if needed.

For DBCS data, the default padding character is the UTF16 character UX’3000' that is converted in the output CCSID, if needed.

**DELIM literal**

When this keyword specified in a USER block, it indicates the delimiter to be used in external DATE or TIME fields. The literal must be a single character and must be one byte long, regardless of the literal CCSID.

For DATE fields, a hyphen (-) is the default value.

For TIME fields, a period (.) is the default value.
For the TIMESTAMP column, both delimiters are used.

**LENGTHBYTE**
Specifies whether to write the two length bytes for variable-length columns to the output data set.

**YES**
The two length bytes are written.

**NO**
The two length bytes are not written.

The default value is YES.

**LENGTH**
Specifies whether the real or maximum length is to be used for variable-length fields.

**REAL**
The length of the field does not change (value of the two length bytes).

**MAX**
The output field is padded to its maximum length with binary zeros.

The LENGTH keyword is useful only for variable-length fields.

The default value is REAL.

**NULLID**
Specifies whether a null indicator byte is added to the beginning of an output field. You can specify NULLID in the OPTIONS block also.

**YES**
The null indicator is created. If the column is null, this indicator is set to the X'FF' value. Otherwise, the indicator is set to X'00'. The indicator can be used by the Db2 LOAD command to load null values into a table. You can change values of the null indicator by using the NULL keyword.

**NO**
The null indicator is not created.

The default value is YES.

**JUST**
Specifies whether to align the output character string. The JUST keyword specifies right or left justification for extended numeric values or for character strings when strings are converted to a greater length.

**RIGHT**
Justify the output character string to the right.

**LEFT**
Justify the output character string to the left.

The default value depends on the type of string. For conversion between character strings, the default value is LEFT. For numeric conversions in strings, the default value is RIGHT.

**PIC block**
For a description of the PIC block, see the PIC keyword.

**NULL**
Indicates whether the null indicator is generated in the output data set. You can specify NULL in the OPTIONS block also.

**value-1**
Indicates the value of the null indicator when the column value is NULL. It can be specified in character ('C') or hexadecimal (X'hh') format.

**value-2**
Indicates the value of the null indicator when the column value is NOT NULL. It can be specified in character ('c') or hexadecimal (X'hh') format.
OFF

No null indicator is generated.

The default is the value that is specified for the VUU014/ULOCAL PARMLIB parameter.

CCSID

Specifies that a CCSID conversion will be done on the output field. The CCSID conversion between the CCSID of the unloaded data and the CCSID that was specified in this parameter must be defined. When a LOB file reference is specified, the specific CCSID conversion is applied on the LOB data. The LOBFILE DSNAME that is generated in the output record is kept in the default CCSID of the output file.

ccsid

Specify a valid CCSID value for the CCSID of the output field.

UTF8

Unicode Transformation Format, 8-bit encoding form is used. UTF8 is equivalent to 1208.

UTF16

Unicode Transformation Format, 16-bit encoding form is used. UTF16 is equivalent to 1200.

BLOBF

Specifies that the output field will contain the name of the file in which the BLOB is to be unloaded without CCSID conversion.

CLOBF

Specifies that the output field will contain the name of the file in which the CLOB will be unloaded with any required CCSID conversion.

DBCLOBF

Specifies that the output field will contain the name of the file in which the DBLOB will be unloaded with any required CCSID conversion.

template-statement-name

Specifies the name of a TEMPLATE statement that is used to allocate the file that will contain the LOB data. The DSN of the TEMPLATE statement that is used must contain either the &UNIQ variable that is explicitly coded or automatically generated for a PDS or PDS/E, or a user-defined variable that is associated with a TEMPLATESET to the :RECNUM predefined variable.

BLOBF, CLOBF, and DBCLOBF output types can be used only when converting BLOB, CLOB and DBCLOB columns. The output type that is defined with the TYPE keyword must be either CHAR or VARCHAR, and the corresponding column must be large enough to contain the generated data set name.

To use LOB file reference with a table that comes from a LISTDEF or a LISTDEFTBV, use BLOBF, CLOBF, or DBCLOBF in the REFORMAT keyword of the OPTIONS block. This option specifies that all LOB data will be unloaded by using LOB file reference. The associated TEMPLATE statement must distinguish the generated files to prevent duplicates. When you unload data from a LISTDEF or a LISTDEFTBV with tables that contain potentially more than one LOB column, the TEMPLATE statement that is used in the REFORMAT keyword can contain a user-defined variable that is associated with the :COLNUM or :COLNAME predefined variables. When you specify the &TS or &SN variables in a TEMPLATE statement that is used with BLOBF, CLOBF, or DBCLOBF, Db2 HPU substitutes the name of the table space that stores the LOB column value, not the base table space name. By substituting the name of the table space, Db2 HPU can generate unique data set names for each LOB column.

IFERROR

Defines the value that must be assigned to the output field if a conversion error occurs when the output field is formatted or if an error occurs when an expression is evaluated by Db2 HPU. IFERROR applies only to conversion or formatting that was made during the unload process. IFERROR does not apply to conversion errors when constants are specified in the SELECT statement and these constants are converted into a different data type by using an INTO clause, a REFORMAT keyword, or a FORMAT USER specification. These constants are converted during the initialization process. If an error occurs during this process, a syntax error is issued, and processing stops.
If an expression error for a SELECT statement that was processed by Db2 occurs, Db2 issues an SQL error message, and a default value does not apply.

**value**

Specifies a value to assign to the output field in a conversion error or an error that occurs when an expression is evaluated. The following constant values are valid:

**Character or hexadecimal constants**

Character constants or hexadecimal constants are accepted for any type of output field. In CCSID conversion, character constants are converted from EBCDIC into the output CCSID, but hexadecimal constants are not converted into the output CCSID.

When you use the following output data types, Db2 HPU uses CCSID conversion for character constants:

- CHAR
- VARCHAR
- CLOB
- GRAPHIC
- GRAPHIC EXTERNAL
- VARGRAPHIC
- DBCLOB
- SMALLINT EXTERNAL
- INTEGER EXTERNAL
- BIGINT EXTERNAL
- FLOAT EXTERNAL
- DECIMAL EXTERNAL
- DECFLOAT EXTERNAL
- DATE_EXTERNAL
- DATE_A to DATE_P
- DATE_DB2
- TIME EXTERNAL
- TIME_A to TIME_E
- TIME_DB2
- TIMESTAMP EXTERNAL
- TIMESTAMP_A to TIMESTAMP_G

When you use the following output data types, Db2 HPU does not use CCSID conversion for character constants:

- BINARY
- VARBINARY
- BLOB
- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- DECIMAL PACKED
- DECIMAL ZONED
- DECFLOAT
- DATE
Numeric constants
Use numeric constants only if the output field corresponds to numeric data in internal format. Otherwise, use a character constant.

The following table shows the conversions that are accepted:

<table>
<thead>
<tr>
<th>Output data type</th>
<th>Type of numeric constant accepted</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALLINT</td>
<td>Integer constant</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Integer constant</td>
</tr>
<tr>
<td>BIGINT</td>
<td>Integer or BIGINT constant</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>Integer constant that fits in the DECIMAL output field or a decimal constant with scale and precision that is smaller or equal to the scale and precision of the output field</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Integer, BIGINT, decimal, or float constants</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>Any numeric type</td>
</tr>
</tbody>
</table>

**DEFAULT**
The default value is assigned to the field. The default value depends on the data type.

**NULL_OR_DEFAULT**
If the output field allows a null value, the NULL value is assigned to the field. Otherwise, the default value is assigned.

**NULL**
The NULL value is assigned to the field. The output field must include a null indicator.

**CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP**
The assigned date, time, or timestamp is identical for each row, and its value is obtained at the beginning of the process. In internal or external format, CURRENT_DATE is allowed only for the DATE field; CURRENT_TIME is allowed only for the TIME field; and CURRENT_TIMESTAMP is allowed only for the TIMESTAMP field.

**TRIM**
Allows you to suppress the trailing blank characters when you unload data into VARCHAR, VARGRAPHIC, CLOB, and DBCLOB output fields. If you specify FORMAT DELIMITED, the TRIM keyword applies to output fields with CHAR and GRAPHIC data types. The length of the output field is adjusted to match the effective number of characters that are written.

The TRIM option does not apply when numeric data is converted to numeric external.

**NO**
Trailing blank characters are not suppressed.

**YES**
Trailing blank characters are suppressed.

The default value is the value of the VUU059/ULTRIM PARMLIB parameter.
Use the TEMPLATESET block to assign values to any user-defined variable that is used in a TEMPLATE definition. Assign values by using string constants or predefined variables that are set by Db2 HPU at run time. A colon (:) must precede predefined variable names. The following diagram shows the syntax of the TEMPLATESET block:

```
TEMPLATESET (variable-name = "string constant"
variable-name = :predefined variable)
```

A predefined variable can be one of the following variables:

:COLNUM
- Specifies the position of the column in the SELECT statement. This variable is a 5-digit numeric value and is generally used when LOB columns are unloaded by using a LOB file reference (CLOBF, BLOBF, or DBLOBF data types).

:SELNUM
- Specifies the sequential number of the SELECT statement in the UNLOAD command. This variable is a 5-digit numeric value.

:UNLNUM
- Specifies the sequential number of the UNLOAD command. This variable is a 5-digit numeric value.

:RECNUM
- The record number is a 15-digit numeric value and is internally processed. By default, the variable that is associated with :RECNUM is replaced by the last seven digits of the record number. A substring can be applied to this variable to obtain another part of the record number, or to retrieve fewer digits. This variable is generally used when LOB columns are unloaded by using a LOB file reference (CLOBF, BLOBF, or DBLOBF data types).

:COLNAME
- This predefined variable contains the first characters of the column name. When the associated variable is replaced by its value, it is truncated to make it a valid qualifier in the generated DSNAME. Valid values are 1 - 8 characters. This variable is generally used when LOB columns are unloaded by using a LOB file reference (CLOBF, BLOBF, or DBLOBF data types).

When these variables cannot be evaluated, :COLNUM and :SELNUM are set to 00000; :RECNUM is set to 0000000; and :COLNAME is set to COLn, where n is a sequential number. This sequential number ensures unicity, but it does not have a specific meaning regarding the actual database or the table space that it represents.

:SELNUM, does not apply when the TEMPLATE statement is used with UNLDDN.

:COLNUM, :COLNAME, and :RECNUM, do not apply when the TEMPLATE statement is used outside of a LOB file reference.

You can code the TEMPLATESET block in the OPTIONS or GLOBAL OPTIONS block. When the TEMPLATESET block is used with LOB columns that use a LOB file reference, the TEMPLATESET block can be used in the column format option block. The COLUMN FORMAT OPTIONS block can be used in the REFORMAT keyword, in the INTO clause, or in the FORMAT USER definition.

All TEMPLATESET definitions that are coded in high-level syntax are available at a lower level. For example, if TEMPLATESET is coded to define a user variable in the GLOBAL OPTIONS block, this variable will be available to build data set names in the templates that are used in all UNLOAD commands and for all SELECT statements in these UNLOAD commands. However, if TEMPLATESET is defined only in an OPTIONS statement that is coded in a SELECT statement, the variable definition will be available only for the TEMPLATE statements that are used in this specific SELECT statement. The definition of a user variable with a TEMPLATESET that is made at a lower level overrides the definition of the same user-defined variable that was made at a higher level but only at this lower level. For example, if a user variable is set to :SELNUM in the GLOBAL OPTIONS block and set to ABC in the OPTIONS statement that is coded in one of the SELECT statements, the replaced value will
be :SELNUM for all other SELECT statements, and ABC for the SELECT statement that overrides the variable definition.

**HIDDEN**
Specifies whether hidden columns are unloaded when a SELECT * statement is used. You can also specify this keyword as the third parameter after the UTILID in the PARM field in the EXEC statement. When the HIDDEN parameter is specified and this keyword is specified in SYSIN, the value that is specified in SYSIN is used.

**NO**
Specifies that hidden columns are not unloaded when a SELECT * statement is used.

**YES**
Specifies that hidden columns are unloaded when a SELECT * statement is used. For SELECT statements that are processed by Db2, the HIDDEN keyword is ignored.

If the HIDDEN keyword is not specified in the SYSIN, the value for HIDDEN that is specified on the EXEC card is the default value. If the HIDDEN keyword is not specified on the EXEC card, the default value is the value of the VUU042/ULHIDDEN PARMLIB parameter.

**SPANNED**
For logical unload statement (using a SELECT clause) which result table has LOB columns and one of the CLOB, BLOB or DBCLOB format is applied to these columns. This option specifies whether the output data must be unloaded into a VBS data set in spanned record format compliant with the Db2 LOAD utility used with the FORMAT SPANNED YES option.

**NO**
The records of the output data set do not have the spanned format.

**YES**
The records of the output data set have the spanned format. The content of the LOB data is unloaded along with the data from the other columns.

**Note:**
- SPANNED YES only applies to LOB data being unloaded as LOB but not if a conversion to any other type is requested.
- SPANNED YES can only be used when the FORMAT option specifies one of the following:
  - VARIABLE
  - USER provided the related LOAD statement can be generated
- SPANNED YES cannot be used with DELIMITED, INTERNAL and EXTERNAL format.
- when SPANNED YES is specified, the parameter TRIM YES does not apply for LOB columns.
- SPANNED YES cannot be used if XML data is to be unloaded.

**ENFORCE_COLUMN_ORDER**
If SPANNED YES option is requested, the LOB data will be placed at the end of the result table (defined by the column list of the SELECT clause and by the optional INTO clause). This condition might not be fulfilled because of the sequence of the explicit expressions that define the result table or because of the underlying table definition when a SELECT * is specified. Db2 HPU can rearrange the columns of the result table to ensure the LOB columns are gathered at the end of the record. The ENFORCE_COLUMN_ORDER specifies if the reordering feature is disabled, i.e. if the native column sequence of the result table is kept.

**YES**
The order of the column in the select statement is kept. If the LOB columns are not the last (rightmost) columns of the result table, the processing stops and an error message is issued.

**NO**
If necessary, Db2 HPU rearranges the columns of the specified result table to ensure the LOB columns are gathered at the end of the record.
**XMLSET**

XMLSET specifies the tags to be used for the XML output. Both columns and records can be tagged. You can specify the way columns (by specifying the COLUMN keyword) and/or records (by specifying the RECORD keyword) are tagged.

Specify either an xmltag if you want a tag to be generated or OFF if you want no tag to be generated.

- COLUMN: specifies that the next keywords apply to columns
- RECORD: specifies that the next keywords apply to records

If XMLSET is not specified, the default value XMLSET (COLUMN = &COLNAME., RECORD = OFF) applies.

The XMLSET options specified at different level are not merged. Only the more local XMLSET specification is taken into account.

For example, if XMLSET (RECORD = <&TSNAME.>) is specified at the UNLOAD level and XMLSET (COLNAME=<&COLNAME.>) is specified at the SELECT level, the output data set is generated as if the XMLSET specified at the unload level did not exist. Therefore, the default for records (RECORD=OFF) applies (i.e. no tag for records are generated).

The following diagram shows the syntax of the XMLSET block:

```
XMLSET non-delimited string &
  DBNAME
  TSNAME
  TBCREATOR
  TBNAME
  COLNAME
```

A predefined variable can be one of the following variables:

**DBNAME**

Name of the data base the unloaded table belongs to. If the data base name cannot be determined (unsupported select statements), the data base name is substituted with "DBnnn" where nnn is a unique number.

**TSNAME**

Name of the table space the unloaded table belongs to. If the data base name cannot be determined (unsupported select statements), the data base name is substituted with "TSnnn" where nnn is a unique number.

**TBCREATOR**

This variable is substituted with the following value according to the precedence order of the list below:

- the table creator specified via the INTO clause ;
- the name of the creator of the table to be unloaded ;
- "CRnnn" where nnn is a unique number of none of the above can be determined.

**TBNAME**

This variable is substituted with the following value according to the precedence order of the list below:

- the table name specified via the INTO clause ;
- the name of the table to be unloaded ;
- "TBnnn" where nnn is a unique number of none of the above can be determined.

**COLNAME**

This variable is substituted with the following value according to the precedence order of the list below:

- "UNDEFINED" if not applicable (i.e. used to specify the RECORD xmltag) ;
- the column name specified via the INTO clause ;
- the name of the column to be unloaded;
- "COLnnn" where nnn is a unique number of none of the above can be determined.

When these variables cannot be evaluated, :COLNUM and :SELNUM are set to 00000; :RECNUM is set to 0000000; and :COLNAME is set to COLn, where n is a sequential number. This sequential number ensures unicity, but it does not have a specific meaning regarding the actual database or the table space that it represents.

:SELNUM, does not apply when the TEMPLATE statement is used with UNLDDN.

:COLNUM, :COLNAME, and :RECNUM, do not apply when the TEMPLATE statement is used outside of a LOB file reference.

You can code the TEMPLATESET block in the OPTIONS or GLOBAL OPTIONS block. When the TEMPLATESET block is used with LOB columns that use a LOB file reference, the TEMPLATESET block can be used in the column format option block. The COLUMN FORMAT OPTIONS block can be used in the REFORMAT keyword, in the INTO clause, or in the FORMAT USER definition.

All TEMPLATESET definitions that are coded in high-level syntax are available at a lower level. For example, if TEMPLATESET is coded to define a user variable in the GLOBAL OPTIONS block, this variable will be available to build data set names in the templates that are used in all UNLOAD commands and for all SELECT statements in these UNLOAD commands. However, if TEMPLATESET is defined only in an OPTIONS statement that is coded in a SELECT statement, the variable definition will be available only for the TEMPLATE statements that are used in this specific SELECT statement. The definition of a user variable with a TEMPLATESET that is made at a lower level overrides the definition of the same user-defined variable that was made at a higher level but only at this lower level. For example, if a user variable is set to :SELNUM in the GLOBAL OPTIONS block and set to ABC in the OPTIONS statement that is coded in one of the SELECT statements, the replaced value will be :SELNUM for all other SELECT statements, and ABC for the SELECT statement that overrides the variable definition.

<table>
<thead>
<tr>
<th>Table 28. Variables usable in xml-templates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td>UNLNUM</td>
</tr>
<tr>
<td>DBNAME</td>
</tr>
<tr>
<td>TSNAME</td>
</tr>
<tr>
<td>SELNUM</td>
</tr>
<tr>
<td>TBCREATOR</td>
</tr>
<tr>
<td>TBNAME</td>
</tr>
<tr>
<td>COLNAME</td>
</tr>
<tr>
<td>COLNUM</td>
</tr>
<tr>
<td>COLTYPE</td>
</tr>
<tr>
<td>COLLENGTH</td>
</tr>
</tbody>
</table>
Table 28. Variables usable in xml-templates (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Alias</th>
<th>Description</th>
<th>Scope</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLPREC</td>
<td>CPRE</td>
<td>Precision of the column. Applies only to a decimal column</td>
<td>COLUMN, type decimal</td>
<td>-1</td>
</tr>
<tr>
<td>COLSCALE</td>
<td>CSCA</td>
<td>Scale of the column. Applies only to a decimal column</td>
<td>COLUMN, type decimal</td>
<td>-1</td>
</tr>
<tr>
<td>COLNULL</td>
<td>CNUL</td>
<td>Nullability attribute of the column (the value is either NULL, NOT NULL or UNDEF if Db2 HPU cannot determine the attribute.)</td>
<td>COLUMN</td>
<td>UNDEF</td>
</tr>
</tbody>
</table>

**OPTIMIZATION_HINT**

Allows to process an SQL statement: SET CURRENT OPTIMIZATION HINT 'string constant' when processing the unload in SQL mode. This option is only available if DB2 parameter is set to FORCE. For example, in Db2 HPU syntax, OPTIMIZATION_HINT 'NOHYB' generates the SQL statement: SET CURRENT OPTIMIZATION HINT = 'NOHYB'

You can clear the CURRENT OPTIMIZATION HINT special register by specifying an empty string. For example, in Db2 HPU syntax: OPTIMIZATION_HINT generates the SQL statement: SET CURRENT OPTIMIZATION HINT = ''

**QUERY_ACCELERATION value**

Allows to have the following SQL statement run when a logical unload is processed in SQL mode:

SET CURRENT SQL ACCELERATION = 'value'

The value can be one of the following values: NONE, ENABLE, ENABLE WITH FAILBACK, ELIGIBLE and ALL. It refers to the relevant Db2 publication for a description of the effect of the QUERY_ACCELERATION special register.

Important: This option is only available if DB2 parameter is set to FORCE.

See the Db2 Universal Database for z/OS SQL Reference for syntax and definitions for Db2 full select.

**Note:** If QUERY_ACCELERATION parameter is specified (except for NONE value) and if the parameter UNLROWSET or its default VUU0035/ULROWSET is set to 1, the multi row fetch mode is forced to 100 rows.

**CHARACTER_LOSS**

It specifies the behavior of the product when significant data from character column is lost during unload formatting because of an INTO or REFORMAT clause and an ssss syntax is specified.

Db2 HPU considers a data loss has occurred for a character column if the data truncation removes any character that does not belong to the padded part of the character string, i.e. a sequence of padding characters that ends at the right end of the character string.

The data loss is checked for the following data types:

- CHAR
- VARCHAR
- GRAPHIC
- VARGRAPHIC
- BINARY
- VARBINARY

Specify one of the following values:
**IGNORE**
No control is done.
The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss.
The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss.
The return code of the job is 8 unless another issue occurs.

The default value is the value of the **VUU071/LOSSCHAR** PARMLIB parameter.

**NUMERIC LOSS**
It specifies the behavior of the product when significant data from numeric column is lost during unload formatting (via INTO or REFORMAT clause) and an ssss syntax is specified.

Db2 HPU considers a data loss has occurred for a numeric column if the numeric value has changed.

The data loss is checked for the following data types:
- SMALLINT
- INTEGER
- BIGINT
- DECIMAL

Specify one of the following values:

**IGNORE**
No control is done.
The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss.
The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss.
The return code of the job is 8 unless another issue occurs.

The default value is the value of the **VUU072/LOSSNUM** PARMLIB parameter.

**ZONED_DEC_SIGN**
Specifies the value of the sign bytes used for zoned-decimal numeric values. The first character represents the positive sign, the second one the negative sign.

The default value is **ZONED_DEC_SIGN (C,D)**.

**UNICODE_EXPANSION_RATIO**
Specifies what expansion ratio (i.e. target data length divided by source data length) to apply to char/varchar data when a conversion from a non-UTF8 CCSID to a UTF8 CCSID is done.

The allowed values are integer from 1 to 4.
The default value is the value of the VUU068/ULUNIEXP parameter.

**CONVERSION_TRUNCATION_ALLOWED**
Specifies what policy to apply when data truncation is needed because of a CCSID conversion.

**YES**
Truncation of the output field after a CCSID conversion is allowed.

**NO**
Truncation of the output field after a CCSID conversion is forbidden. Records requiring truncation are discarded and the return code is set to 4.
The default value is the value of the VUU067/ULCNVTRC parameter.

**HANDLE_RECORD_ID**

Allows to get more detailed information when formatting errors occur so that identifying accurately the record involved in the error is possible.

When such an error occurs, if HANDLE_RECORD_ID is not specified, the error is reported by a message formatted as follows:

```
* error_code * error_type * output_ddname * output_area * content
```

where:

- error_code is the code for the error type
- error_type is the type of error that was encountered
- output_ddname is the output ddname into which the record causing the error is to be written
- output_area designates the area of the output record causing the error
- content is the hexadecimal representation of the output area to be formatted

The information displayed by this standard/default error message is sometimes insufficient for trouble shooting as it does not allows to accurately determine which record is involved.

If HANDLE_RECORD_ID is specified and if the unload is processed in native mode, the messages are formatted as follows:

```
* error_code * error_type * output_ddname * output_area * PART partition_number PGNUM
page_number ID id_map *  DB :  database  TS : tablespace
```

where the additional following information is displayed:

- partition_number: the partition number if the table space is partitioned
- page_number: the page number - in hexadecimal representation - in the LDS
- id_map: the ID map - in hexadecimal representation - in the page
- database,tablespace: the identifier of the table space that contains the record that caused the error

The same additional information is also displayed if a $HRECID dd statement is specified for the Db2 HPU exec statement which makes this feature available even if the Db2 HPU native syntax is not used.

**Note:** requesting the more detailed message to be issued decreases the performance of the unload job run in native mode as Db2 HPU handles longer records when it is specified. As the amount of extra information in the record does not depend on the data to unload, the shorter the output record the bigger the impact on performance.

Examples of formatting error messages:

The message below gives an example of detailed error message issue if either HANDLE_RECORD_ID or $HRECID dd statement is specified:

```
* 234 * DATA LOST * OUT      * C1                             * C4E2D5C4C2F0F440
* 234 * DATA LOST * OUT      * PART 0001 PGNUM 00000002 ID 01 *  DB : DSNDB06  TS :
SYSTSDBA
```

and the corresponding message issued otherwise:

```
* 234 * DATA LOST * OUT      * C1                             * C4E2D5C4C2F0F440
```

**INTO_RULES**

This option allows to change the behavior of the INTO clause that applies when the field-name is explicitly specified but its output-data-type is not specified.

**COLTYPE**

The output-data-type is taken from the column type of the resulting table and the column-format-options specified through the INTO clause apply. Any REFORMAT clause is ignored.
**REFORMAT**

If a REFORMAT clause is applicable to the column type of the resulting table, the output-data-type specified through the REFORMAT clause applies as well as the associated column-format-options except for the items that are also specified in the INTO clause that take precedence. If no REFORMAT clause is applicable to the resulting column, the output-data-type is that of the column and the related column-format-options applies.

The default value is the value of the VUU089/INTORUL1 parameter.

**Related concepts**

Output encoding scheme

For the DSNTIAUL, DELIMITED, VARIABLE, USER, and EXTERNAL output formats, the translations from EBCDIC to ASCII and from ASCII to EBCDIC are supported only for single-byte character set (SBCS) character strings. Data is translated by using the translation tables in the SYSIBM.SYSSTRINGS table.

**Related reference**

FORMAT block syntax and description

Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

**TIMESTAMP format types**

Use the TIMESTAMP format type to specify the output data format.

**TIME format types**

Use the TIME format type to specify the output data format.

**DATE format types**

Use the DATE format type to specify the output data format.

**Related information**

TIME Data conversion table

**Example: Unloading in XML format from table columns only with the additional AUTOTAG field**

In this example, you can unload in XML format from table column with the additional AUTOTAG field.

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options. As no column name is available for this extra field, the COL000 tag is used for this field. The NULLL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS AUTOTAG 2
SELECT DEPTNO, MGRNO, ADMRDEPT FROM DSN8910.DEPT
FORMAT XML DELIM '"' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < HEADER > )
```

Output:

```
<HEADER><COL000>01</COL000><DEPTNO>"D01"</DEPTNO>
<MGRNO>NULL</MGRNO><ADMRDEPT>"A00"</ADMRDEPT></HEADER>
<HEADER><COL000>01</COL000><DEPTNO>"D11"</DEPTNO>
<MGRNO>"000060"</MGRNO><ADMRDEPT>"D01"</ADMRDEPT></HEADER>
```

**Example: Unloading in XML format with an SQL statement involving SQL expressions from an identified table**

In this example, you unload data in XML format with an SQL statement involving SQL expressions from an indentified table.

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options and an SQL expression is involved by the SELECT. As no column
name is available for both the AUTOTAG field and the SQL expression, the COL000 and COL001 tags are used for these fields. The NULL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS AUTOTAG 2
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) FROM DSN8910.DEPT
FORMAT XML DELIM '"' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < &TBNAME > )
```

Output:

```
<DEPT><COL000>01</COL000><DEPTNO>"A00"</DEPTNO><COL002>"000010000010"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"B01"</DEPTNO><COL002>"000020000020"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"C01"</DEPTNO><COL002>"000030000030"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><COL002>NULL</COL002></DEPT>
```

Example: Unloading in XML with an SQL statement involving SQL expressions and an INTO clause

In this example, you unload data in XML format with an SQL statement involving SQL expressions and an INTO clause

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options and an SQL expression is involved by the SELECT. An INTO clause requests the field name to be changed. As no column name is available for both the AUTOTAG, the COL000 is used for this field. The NULLL values are not materialized.

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS AUTOTAG 2
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) INTO TABLE USER.NEWDEPT C1, C2 FROM DSN8910.DEPT
FORMAT XML DELIM '"'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < &TBNAME > )
```

Output:

```
<NEWDEPT><COL000>01</COL000><C1>"A00"</C1><C2>"000010000010"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"B01"</C1><C2>"000020000020"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"C01"</C1><C2>"000030000030"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"D01"</C1><C2>NULL</C2></NEWDEPT>
```

Example: Unloading in XML format using variables

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS
XMLSET ( RECORD = < &TS. CREATOR = &CREATOR., TABLE = &TBNAME. > ,
COLUMN = <COLUMN NAME = &COLNAME. , TYPE = &COLTYPE. > )
SELECT ACTNO, ACTKWD FROM DSN8910.ACT
FORMAT XML
```

The output record is as follows:

```
<DSN8S71P CREATOR = "DSN8910" TABLE = "ACT">COLUMN NAME = "ACTNO" TYPE = "SMALLINT">10</C COLUMN><COLUMN NAME = "ACTKWD" TYPE = "CHAR">'TEST'</COLUMN></DSN8S71P>
```
Example: Unloading with records tagged with 'HEADER' and columns tagged with their name

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options. As no column name is available for both the AUTOTAG, the COL000 is used for this field. The NULL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS AUTOTAG 2
SELECT DEPTNO, MGRNO, ADMRDEPT FROM DSN8910.DEPT
FORMAT XML DELIM '"' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN =< &COLNAME. >, RECORD =< HEADER > )
```

Output:

```
<HEADER><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><MGRNO>NULL</MGRNO><ADMRDEPT>"A00"</ADMRDEPT></HEADER>
<HEADER><COL000>01</COL000><DEPTNO>"D11"</DEPTNO><MGRNO>"000060"</MGRNO><ADMRDEPT>"D01"</ADMRDEPT></HEADER>
```

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options. As no column name is available for both the AUTOTAG field and the SQL expression, the COL000 and COL001 tags are used for these fields. The NULL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS AUTOTAG 2
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) FROM DSN8910.DEPT
FORMAT XML DELIM '"' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN =< &COLNAME. >, RECORD =< &TBNAME. > )
```

Output:

```
<DEPT><COL000>01</COL000><DEPTNO>"A00"</DEPTNO><COL002>"000010000010"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"B01"</DEPTNO><COL002>"000020000020"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"C01"</DEPTNO><COL002>"000030000030"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><COL002>NULL</COL002></DEPT>
```

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options and an SQL expression is involved by the SELECT. An INTO clause request the field name be changed. As no column name is available for the AUTOTAG, the COL000 is used for this field. The other fields are tagged according to the name given via the INTO clause. The NULL values are not materialized.

UNLOAD statement:

```
UNLOAD TABLESPACE
OPTIONS AUTOTAG 2
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) INTO TABLE USER.NEWDEPT C1, C2 FROM DSN8910.DEPT
FORMAT XML DELIM '"' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN =< &COLNAME. >, RECORD =< &TBNAME. > )
```

Output:

```
<NEWDEPT><COL000>01</COL000><C1>"A00"</C1><C2>"000010000010"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"B01"</C1><C2>"000020000020"</C2></NEWDEPT>
```
Example: Unloading LOB data into spanned format

```
UNLOAD TABLESPACE
SELECT *
FROM me.mytable
OUTDDN (SYSREC01)
FORMAT VARIABLE ALL
OPTIONS SPANNED YES
LOADDN LOAD
```

Examples: Setting and requesting CURRENT_QUERY_ACCELERATION

Example 1: Request the CURRENT QUERY ACCELERATION special register to be set to ENABLE for every unload performed in SQL mode by specifying the QUERY_ACCELERATION option in an option block at the TABLESPACE level.

The DB2 FORCE option ensures every unload will be performed in SQL mode.

```
UNLOAD TABLESPACE
DB2 FORCE
OPTIONS QUERY_ACCELERATION ENABLE
SELECT *
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
```

Example 2: Request the QUERY_ACCELERATION special register to be set to ENABLE only for a specific logical unload by specifying the QUERY_ACCELERATION option in an option block at the SELECT level.

The DB2 FORCE option ensures every unload will be performed in SQL mode.

```
UNLOAD TABLESPACE
DB2 FORCE
SELECT *
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION ENABLE
```

Example 3: Request the QUERY_ACCELERATION special register to be set to different values according to the related logical unload by specifying the QUERY_ACCELERATION option in an option block at the SELECT level.

The DB2 FORCE option ensures every unload will be performed in SQL mode.

```
UNLOAD TABLESPACE
DB2 FORCE
-- SELECT #1
SELECT *
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION ENABLE WITH FAILBACK
-- SELECT #2
SELECT *
FROM SYSIBM.SYSTABLESPACE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION NONE
```
Example 4: Specify a general QUERY_ACCELERATION value applicable to all logical unloads except for those having a QUERY_ACCELERATION option specified at the SELECT level.

ALL applies to the first and the third SELECT statements, ELIGIBLE for the second one and NONE for the fourth one.

```
UNLOAD TABLESPACE
  OPTIONS QUERY_ACCELERATION ALL
  DB2  FORCE
-- SELECT #1
  SELECT *
  FROM SYSIBM.SYSDATABASE
  OUTDDN (SYSREC)
  FORMAT DSNTIAUL
-- SELECT #2
  SELECT *
  FROM SYSIBM.SYSTABLESPACE
  OUTDDN (SYSREC)
  FORMAT DSNTIAUL
  OPTIONS QUERY_ACCELERATION ELIGIBLE
-- SELECT #3
  SELECT *
  FROM SYSIBM.SYSUSERAUTH
  OUTDDN (SYSREC)
  FORMAT DSNTIAUL
-- SELECT #4
  SELECT *
  FROM SYSIBM.SYSDBAUTH
  OUTDDN (SYSREC)
  FORMAT DSNTIAUL
  OPTIONS QUERY_ACCELERATION NONE
```

Example: Unloading a table space containing 3 tables and requesting the NUMRECS option to be specified in the generated LOAD statement using the actual number of unloaded rows

```
UNLOAD statement:

GLOBAL OPTIONS AUTOTAG 3
LOADOPT (,,NUMRECS &NUMRECS);
UNLOAD TABLESPACE
QUIESCE YES
--SELECT FIRST TABLE WITH 2365 ROWS
SELECT *
FROM ID6631.ID6631S1T1
OUTDDN SYSREC
LOADDDN SYSPPUNCH
--SELECT SECOND TABLE WITH 15423 ROWS
SELECT *
FROM ID6631.ID6631S1T2
OUTDDN SYSREC
LOADDDN SYSPPUNCH
--SELECT THIRD TABLE WITH 15 ROWS
SELECT *
FROM ID6631.ID6631S1T3
OUTDDN SYSREC
LOADDDN SYSPPUNCH
```

It generates a LOAD statement where the NUMRECS option specifies a different value for each table, according to the number of rows unloaded for the related table, as follows:

```
LOAD DATA LOG NO REPLACE
EBCDIC CCSID (0037)
INTO TABLE ID6631.ID6631S1T1 NUMRECS 2365
WHEN (1:3) = '001'
  (COLINT
   POSITION (4) INTEGER
   NULLIF(8) = '?')
INTO TABLE ID6631.ID6631S1T2 NUMRECS 15423
WHEN (1:3) = '002'
  (COLCHAR
```

180 Db2 High Performance Unload User's Guide
Example: Generating the access path during the data extraction using CURRENT OPTIMIZATION HINT

The unload below, run in SQL mode (ensured via the DB2 FORCE option), requests the CURRENT_OPTIMIZATION_HINT special register to be set to the value NOHYB, an optimization hint defined in the owner’s PLAN_TABLE, so that Db2 uses this optimization hint to generate the access path during the data extraction needed for this unload.

Example: Unloading time, date and timestamp with usual and simple format and delimiters

The following unload formats the current date, time and timestamp Db2 special registers with a set of basic delimiters (sometimes no delimiter is specified) by specifying the output format via the REFORMAT clause.

With the unload below:

Example 2: controlling the number of fractional seconds digits written for timestamps

The following unload formats the current timestamp Db2 special register with a various number of fractional seconds digits by specifying the output format via the INTO clause.
Example 3: unloading time, date and timestamp with complex format and delimiters

The following unload formats the current date, time and timestamp Db2 special registers with a various complex delimiters by specifying the output format via the REFORMAT clause. The format for TMSTP2 specifies a delimiter that is also the string delimiter (i.e. the quotation mark) by specifying it twice in sequence (in MI"SS", the first " is a set of two single quotation marks while the second " is a double quotation mark).

The following unload:

```
UNLOAD TABLESPACE DB2 FORCE
SELECT CURRENT_TIMESTAMP,
CURRENT_TIMESTAMP,
CURRENT_TIMESTAMP
INTO TMSTP1 TIMESTAMP 'YYYY/MO/DD HH:MI:SS.NN',
TMSTP2 TIMESTAMP 'HH h MI"SS" & NN th of second',
TMSTP3 TIMESTAMP 'YYYY MO DD HH MI SS'
FROM SYSIBM.SYSDUMMY1 ;
FORMAT DELIMITED SEP '"',
OUTDDN(SAMPLE3)
EBCDIC
```

produces an output record as follows:

```
```

Example 4: coding a timestamp format over two lines

The following unload formats the current timestamp Db2 special register with a format string written over two lines. The new lines starts in column #1 so that no undesired extra blank is added.

The following unload:

```
UNLOAD TABLESPACE DB2 FORCE
SELECT 'Today is ',
CURRENT_TIMESTAMP
INTO HEADER CHAR ,
TMSTP TIMESTAMP 'YYYY/MO/DD DD and the exact time is HH hours MI minutes SS seconds and NNNNNN microseconds'
FROM SYSIBM.SYSDUMMY1 ;
FORMAT DELIMITED SEP '"',
OUTDDN(SAMPLE4)
```

produces an output record as follows:

```
Today is  2017 07, 31 and the exact time is 18 hours 38 minutes 45 seconds and 257235 microseconds
```

Example 5: specifying a user-defined date format within a REFORMAT clause

The following unload formats expression based on the current date Db2 special register by using a REFORMAT clause that apply to all SELECT belonging to the UNLOAD statement and INTO clauses that apply at the column level.

The following unload:
UNLOAD TABLESPACE
OPTIONS REFORMAT(
  TYPE DATE INTO DATE 'YYYY MO DD DEFAULT FOR ALL SELECTS'
)
DB2 FORCE
SELECT CURRENT_DATE,
    CURRENT_DATE + 1 MONTH,
    CURRENT_DATE + 1 YEAR
INTO DATE1 DATE 'DD/MO/YY FOR COL_1 IN SEL_1',
    DATE2 DATE
FROM SYSIBM.SYSDUMMY1;
FORMAT DELIMITED SEP '/'
OUTDDN(SAMPLE7)
SELECT CURRENT_DATE,
    CURRENT_DATE + 1 MONTH,
    CURRENT_DATE + 1 YEAR
INTO DATE1 DATE 'DD/MO/YY FOR COL_1 IN SEL_2'
FROM SYSIBM.SYSDUMMY1;
FORMAT DELIMITED SEP '/'
OUTDDN(SAMPLE5)

produces output records as follows:

<table>
<thead>
<tr>
<th>Date Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/08/17 FOR COL_1 IN SEL_1</td>
<td>2017-09-01/2018 08 01 DEFAULT FOR ALL SELECTS</td>
</tr>
<tr>
<td>01/08/17 FOR COL_2 IN SEL_2</td>
<td>2017 09 01 DEFAULT FOR ALL SELECTS/2018 08 01 DEFAULT FOR ALL SELECTS</td>
</tr>
</tbody>
</table>

Note that the DATE2 specification in the first INTO clause sets the output data type to DATE and its related output format overrides the formatting specification given by the REFORMAT CLAUSE.

SELECT block syntax and description

The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

Db2 HPU can process a SQL SELECT statement in native mode if it fulfills all the following conditions:

- it involves only one table,
- it specifies only supported SQL expressions,
- it fulfills other more specific conditions that cannot be comprehensively listed in a simple manner and that depend on the maintenance level of Db2 HPU.

More complex SQL SELECT statements can be processed in SQL mode that can be allowed by setting VUU011/ULSEDB2 to YES or by specifying the DB2 YES option or requested by specifying the DB2 FORCE option. If DB2 YES applies, Db2 HPU processes the SELECT statement in native mode, if supported, and attempts to process it in SQL mode otherwise, in which case it reports the situation by issuing the INZU063I UNSUPPORTED SELECT message.

If you specify more than one SELECT statement, Db2 HPU attempts to run the unload jobs in parallel.

The following diagram shows the syntax for the SELECT block:
SELECT block

SELECT

<table>
<thead>
<tr>
<th>fast select block</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast listdef select block</td>
</tr>
<tr>
<td>fast listdeftbv select block</td>
</tr>
</tbody>
</table>

DB2 full select

OUTDDN

(output-data-set-base-ddname)

MAX_EXPECTED_ROWS

OUTMAXROWS

ON_RESULT_TABLE

OUTFREQROWS

ON_RESULT_TABLE

OUTEXIT

exit-name

ASIS

ASCII

EBCDIC

UNICODE

COBOL2

CCSID

(integer)

format block

options block

Notes:
1 You can specify a maximum of 255 ddnames for copies of the output data set
2 When the format block is not specified, the default format is the value of the VUU045/ULFORMAT parameter.

SELECT

Specifies the SELECT statement as a standard SQL statement.

fast select block

See “Fast select, fast listdef and fast listdeftbv select blocks syntax and description” on page 188 for keyword definitions.

fast listdef select block

See “Fast select, fast listdef and fast listdeftbv select blocks syntax and description” on page 188 for keyword definitions.

fast listdeftbv select block

See “Fast select, fast listdef and fast listdeftbv select blocks syntax and description” on page 188 for keyword definitions.
DB2 full select
All valid Db2 SELECT statements that do not conform to Db2 HPU fast select block requirements are considered a DB2 full select and are processed by using Db2 unless DB2 NO was specified in the UNLOAD command.

OUTDDN output-data-set-base-ddname
Specifies the ddname of the sequential output data set that contains the unloaded data or the name of a TEMPLATE statement that is defined in the same SYSIN or in a TEMPLATE library. You can specify up to 255 ddnames for copies of the output data set. When you use JCL-allocated ddnames, in the JCL, include a DD statement that corresponds to each ddname that you specify.

The variable outddn is the base ddname of the output data set.

To process partitioned table spaces in parallel, either use a TEMPLATE with a DSNAME that contains the &PART variable, or code in your JCL one outddnnn statement for each partition (outdd01, outdd02,... outddnnn), where nnn is a 1- to 7-digit sequential number that identifies a partition to be unloaded. During the unload process, data from each partition is directed to the corresponding ddname. If the corresponding ddname is allocated, it is used for the given partition. Otherwise, the base ddname is used if it is allocated.

In the following example, if MYDD, MYDD01, and MYDD0004 are allocated, MYDD contains the rows from partition 2 and 5, MYDD01 contains the rows from partition 1, and MYDD0004 contains the rows from partition 4.

UNLOAD TABLESPACE PART(1,2,4,5) SELECT * FROM Q.T OUTDDN(MYDD) FORMAT DSNTIAUL

If a single ddname is used as output for multiple UNLOAD or SELECT statements, data from one or more of the UNLOAD or SELECT statements might be included in the single output ddname file. To ensure that data collected from each source remains together in the generated output, specify unique base ddnames for each SELECT or UNLOAD statement. A base ddname that is used as an operand of either the UNLDDN keyword or the OUTDDN keyword is not used as a suffixed ddname, as shown in the following example:

```
UNLOAD TABLESPACE PART(1,2,4,5)
SELECT * FROM Q.TA OUTDDN(MYDD) FORMAT DSNTIAUL
SELECT * FROM Q.TB OUTDDN(MYDD1) FORMAT DSNTIAUL
```

To avoid including the data from tables Q.TA and Q.TB in the same data set, ddname MYDD1, if allocated, is used as the base ddname for only the second SELECT, and does not contain data from partition 1 of the Q.TA table. The data from partition 1 of the Q.TA table is written in ddname MYDD, which is allocated.

Tip: When you use a TEMPLATE in the OUTDDN keyword with a SELECT FROM LIST (fast listdef select block) or a SELECT FROM LISTTBV (fast LISTDEFTBV select block), and when some of the table spaces that are in the list are multi-table table spaces, include a user-defined variable that is set to the :SELNUM predefined variable, using a TEMPLATESET option in the DSN expression of the TEMPLATE.

The OUTDDN keyword is required for SELECT statements.

MAX_EXPECTED_ROWS n
Specifies an estimate of the number of unloaded rows. Use this keyword when a WHERE clause is specified because the number of unloaded rows is less than the number of rows that are contained in the table. You can use MAX_EXPECTED_ROWS to do the following functions:

- Limit the number of rows that are specified to the sort utility.
- Limit the size of the allocated output dataset when you use a template without allocation parameters. When you specify MAX_EXPECTED_ROWS, data is unloaded by Db2 into a template file without space allocation being specified.

Consider specifying MAX_EXPECTED_ROWS if Db2 HPU fails because of a lack of resources. However, if the Db2 HPU estimate is accurate, specifying MAX_EXPECTED_ROWS will not solve resource problems.
Important: Specifying MAXEXPECTED_ROWS disables the index scan feature and might decrease the Db2 HPU performance.

**OUTMAXROWS n**
Specifies if a limitation of the number of rows extracted for this SELECT statement must be applied.

- 0: no limitation applies. All the rows corresponding to the SELECT statement are unloaded.
- integer from 1 to 2147483647: the specified value is used to limit the number of rows unloaded for the related SELECT statement. The number of unloaded rows never exceeds the specified value except for an unload run in native mode against a partitioned table space with partition parallelism, i.e. the value of the first argument of the PARALLELISM option or the PARALLEL parmlib parameter is greater than 1. In such a case, the specified value is used as a limit for the unload of each involved partition. Therefore, the overall number of rows unloaded for the SELECT statement is p times the specified value where p is the number of involved partitions. If the UNLOAD statement requests the data of each partition be unloaded, Db2 HPU does not write more than the specified value into each partition-related output data set.

Important: If you specify OUTMAXROWS, do not specify the SQLPART keyword because these keywords are incompatible with each other.

**ON_RESULT_TABLE**
Specifies that the maximum number of rows is applied when a SELECT statement is processed natively by Db2 HPU and when a SELECT statement, with or without SORT(EXTERNAL) specified, is processed by Db2. If an ORDER BY clause is specified, the ON_RESULT_TABLE keyword is applied after the WHERE clause is applied and after the data is sorted.

For SELECT statements with SORT(INTERNAL) specified and that are processed by Db2, specifying ON_RESULT_TABLE has no effect because the setting corresponds with the default processing.

For SELECT statements with SORT(EXTERNAL) specified and that are processed by Db2, the OUTMAXROWS keyword is applied by default after data from the SQL cursor is fetched before the sort. If you specify ON_RESULT_TABLE, all data that is returned by the Db2 cursor is processed by the sort, and the OUTMAXROWS keyword allows the first n rows to be retrieved after sorting.

For SELECT statements that are processed natively by Db2 HPU, the OUTMAXROWS keyword is applied by default when data from the pages of the unloaded table space is read, before the WHERE clause is applied, and before the data is sorted, if an ORDER BY clause is specified. However, if you specify ON_RESULT_TABLE, the OUTMAXROWS keyword is applied after the WHERE clause is applied and after the sort by the ORDER BY clause.

Attention: Specifying ON_RESULT_TABLE might adversely affect Db2 HPU performance because a large amount of data might be processed before the number of records is limited.

**OUTFREQROWS n**
Specifies the unload sampling frequency.

If the SELECT statement is processed natively by Db2 HPU, one row in every n rows is kept when reading the rows from the linear data set (LDS) that contains the table space. In this case, the WHERE clause and the ORDER BY clause, if they exist, are applied after the sampling is done.

If the SELECT statement is processed by Db2, one row in every n rows is kept when fetching the rows from the cursor that is dynamically prepared by Db2 HPU for the complete SELECT statement, including the WHERE clause and the ORDER BY clause. In this case, if you specify SORT(INTERNAL), the WHERE clause and the ORDER BY clause, if they exist, are applied before the sampling is done. If the SORT(EXTERNAL) option is active, only the WHERE clause is applied before the sampling is done.

Important: If you specify OUTFREQROWS, do not specify the SQLPART keyword because these keywords are incompatible with each other.

**ON_RESULT_TABLE**
Specifies that the sampling for the SELECT statement is applied when a SELECT statement is processed natively by Db2 HPU or when a SELECT statement, with or without SORT(EXTERNAL) specified, is processed by Db2.
specified, is processed by Db2. The OUTFREQROWS keyword is applied after the WHERE clause is applied and after the data is sorted if an ORDER BY clause is specified.

For SELECT statements with SORT(INTERNAL) specified and that are processed by Db2, specifying ON_RESULT_TABLE has no effect because the setting corresponds with the default processing.

For SELECT statements with SORT(EXTERNAL) specified and that are processed by Db2, the OUTFREQROWS keyword is applied by default after data is fetched from the SQL cursor and before the sort. If you specify ON_RESULT_TABLE, all data that is returned by the Db2 cursor is processed by the sort, and the OUTFREQROWS keyword causes sampling after the sort.

For SELECT statements that are processed natively Db2 HPU, the OUTFREQROWS keyword is applied by default when data from the pages of the unloaded table space is read, before the WHERE clause is applied, and before the data is sorted by the ORDER BY clause. However, if you specify ON_RESULT_TABLE, the OUTFREQROWS keyword is applied after the WHERE clause is applied and after the sort by the ORDER BY clause.

**Attention:** Specifying ON_RESULT_TABLE might adversely affect Db2 HPU performance because a large amount of data might be processed before the records are sampled.

**OUTEXIT exit-name**

Specifies the name and the language of the exit that handles the rows during unload processing. The exit that you specify is loaded dynamically during unload processing. The exit must reside in an authorized library that must be in either the LINKLIST or an authorized JOBLIB or STEPLIB. For COBOL/2 and C, the STEPLIB, JOBLIB, or LINKLIST must also point to the LE/370 runtime libraries.

**Attention:** This option is not supported when the SQL mode is used.

**ASM**

Assembly language

**C**

C language

**COBOL2**

COBOL/2 language

The default value is ASM.

**EBCDIC/ASCII/UNICODE/ASIS**

Specifies whether the data is unloaded in EBCDIC, ASCII, or UNICODE format by using the coded character set identifier (CCSID) of the installation or the specified CCSID.

**ASCII**

Indicates that the unloaded data must be in ASCII format. Db2 HPU uses the ASCII CCSID of the subsystem, unless you override it by specifying the CCSID option.

**ASIS**

Indicates that the data is unloaded in its original format. If the specification for the underlying table space cannot be determined, such as when the data is processed by Db2, the CCSID that is returned by a standard PREPARE statement that is used in an SQL descriptor area (SQLDA) is used. You can override ASIS by specifying the CCSID keyword.

Specifying ASIS does not mean that conversion is not required. Conversion might still be required in some situations, such as between input from SYSIN and the CCSID of the system, or between the CCSID of the system and printed output.

**EBCDIC**

Indicates that the data is unloaded in EBCDIC format. Db2 HPU uses the EBCDIC CCSID of the subsystem, unless you override it by specifying the CCSID keyword.

**UNICODE**

Indicates that the data is unloaded in UNICODE format. Db2 HPU uses the UNICODE CCSID of the subsystem, unless you override it by specifying the CCSID option.
The default value is EBCDIC.

**Attention:** If the unload format that was specified in either the SYSIN or in the PARMLIB by using the UNLSCHEM parameter is not identical to the EBCDIC format of the system, all constants that are specified in SYSIN are translated to the unload format.

**Note:** No translation is allowed for LOB/XML data when they are unloaded in VBS file (SPANNED YES option).

**CCSID integer**

Specifies up to three coded character set identifiers (CCSIDs) for the unloaded data. The first identifier specifies the CCSID for SBCS data, the second identifier specifies the CCSID for MIXED DBCS data, and the third identifier specifies the CCSID for DBCS data. If you omit any of these CCSIDs or specify 0 for any of them, the CCSID of the corresponding data type is assumed to be the same as the installation default CCSID.

You can also specify the CCSID at the column level in the USER block syntax.

The default value is 0.

**Note:** No translation is allowed for LOB/XML data when they are unloaded in VBS file (SPANNED YES option)

**format block**

See “FORMAT block syntax and description” on page 206.

**options block**

See “OPTIONS block syntax and description” on page 143.

See the *Db2 Universal Database for z/OS SQL Reference* for syntax and definitions for DB2 full select.

**Related concepts**

Db2 HPU user exit

The Db2 HPU user exit is used to customize the output data set that is created by a SELECT statement.

**Related reference**

User-allocated ddnames

To run unload jobs, you must allocate certain ddnames in the Db2 HPU JCL.

**SELECT statement examples**

These SELECT statement examples show how to create sequential data sets in different formats, how to use the LISTDEF and TEMPLATE keywords, how to use the INTO clause and the REFORMAT clause, and how to use other Db2 HPU options.

**Fast select, fast listdef and fast listdeftbv select blocks syntax and description**

The fast select, fast listdef and fast listdeftbv select blocks describe the syntax for SELECT statements that can be supported natively by Db2 HPU when DB2 NO or DB2 YES is specified in the UNLOAD command. The fast select, fast listdef and fast listdeftbv select blocks are part of the SELECT block.

The INTO clause of the fast select block can also be used for any fullselect component that is coded in the Db2 HPU syntax even if the fullselect does not follow the syntax rules of the fast select block. To use the INTO clause for an unsupported SELECT statement, include the INTO clause in the SELECT statement between the list of selected items and the FROM clause. If the fullselect contains a UNION, EXCEPT, or INTERSECT clause, an INTO clause can be coded only on the first SELECT statement of the fullselect.

The following diagram shows the syntax of the fast select block:

**fast select block**
Notes:

1 Db2 HPU determines the creator and name of the table, if applicable - i.e. if a single table is involved by the SELECT statement - and sets creator.name to "????????????????" otherwise.

2 If you do not specify the PART keyword or SQLPART keyword, the default value is PART(ALL).

The fast select block contains the following keywords. For the subset of keywords that the fast listdef select block accepts, see the syntax diagram.

* Indicates that this SELECT statement applies to all columns of the table.

**column-name**
Indicates the name of a column in the table.

**constant**
Specifies a literal or a numeric value. If you specify a literal value, enclose it in single quotation marks (’) or double quotation marks (“”), according to your Db2 installation.

**expression**
The following syntax diagram shows an expression:
output-field-block:

field-name type of the resulting column \[\text{output data type}\] \[\text{default option for the output-data-type}\] \[\text{column-format-option}\]

Notes:

1 If no output-data-type is specified, the following rules apply, either the output-data-type is that of the column of the resulting table, or no REFORMAT clause is taken into account.

Note: On the output-field-block: If the output-field-block is empty, the following rules apply:

- the field-name is that of the column of the resulting table
- the output-data-type is that of the column of the resulting table modified by the applicable REFORMAT clause, if any.
- the column-format-options are inherited from the format options specified at upper levels with the following order of precedence
  - the SELECT level
  - the UNLOAD level
  - the GLOBAL that applies to the requested output format
  - the parmlib level that applies to the requested output format
- the column-format-options specified by the applicable REFORMAT clause, if any, override the values specified at any level, regardless of the level where the REFORMAT clause is specified.

Note: On the default option for the output-data-type on the output-field-block: If a column-format-option is not specified, the following rules apply:

- the output format option is inherited from the format options specified at upper levels with the following order of precedence
  - the SELECT level
  - the UNLOAD level
  - the GLOBAL that applies to the requested output format
– the parmlib level that applies to the requested output format
• no REFORMAT clause is taken into account.

**INTO**

Use the INTO clause to do either of the following actions:

• specify the name of the target table
• for some or all the columns of the resulting table, do either of the following regarding the output fields:
  – specify the field name
  – specify the data type
  – specify format options

This respectively allows to rename the target table and, like what can be done with FORMAT USER, to rename, cast and reformat the output field.

**Restriction:** If the INTO clause is specified along with a LIKE clause and FORMAT DSNTIAUL or FORMAT VARIABLE is requested, changing the output field characteristics is not allowed, and only the TABLE keyword suboption can be used.

**TABLE creator.name or TABLE name**

Specifies the name of the target table in the LOAD control cards generated if the LOADDDN feature is requested. The creator name of the table name may be omitted. In such a case, the applicable creator name is determined when the LOAD controls cards are used to run the LOAD utility.

The table name defined by the INTO clause is only used to name the target table. This table does not need to exist in the source Db2 against which the unload is run. The field description generated in the LOAD control cards is consistent with the structure of the resulting table defined either through the SELECT statement or the LIKE table-name clause of the DSNTIAUL block. It belongs to the user to make sure the target table structure is compatible with this fields description.

**output-field-block**

Specifies the output field characteristics. When left blank, If the output-field-block is empty, the following rules apply:

• the field-name is that of the column of the resulting table
• the output-data-type is that of the column of the resulting table modified by the applicable REFORMAT clause, if any.
• the column-format-options are inherited from the format options specified at upper levels with the following order of precedence
  – the SELECT level
  – the UNLOAD level
  – the GLOBAL that applies to the requested output format
  – the parmlib level that applies to the requested output format
• the column-format-options specified by the applicable REFORMAT clause, if any, override the values specified at any level, regardless of the level where the REFORMAT clause is specified.

Otherwise, specifies the field name (mandatory) and (optionally) the data type and format options for the output field.

An output-field-block applies to a specific column of the final result table in a positional manner, i.e. the output-field-block specifies in a given rank in the list of output-field-block applies to the column of the final result table having the same rank. The output-field-blocks must be separated by commas. An output-field-block must be specified for every field up to the field with a non-blank output-field-block of highest rank. The remaining blank output-field-block may be omitted as well as the related commas.
**field-name**
Specifies the name out the output-field. This allows to rename the output field or to remind the genuine field name.

**output-data-type**
Specifies the data type of the output field. This allows to cast the data. When left blank, if no output-data-type is specified, the following rules apply:

- the output-data-type is that of the column of the resulting table
- no REFORMAT clause is taken into account, since the output-data-type implicitly or explicitly specified through a non-blank output-field-block defines a casting rule such as a REFORMAT clause.

**column-format-option**
Specifies a formatting option of the output data. For an omitted formatting option, the following rules apply:

- the output format option is inherited from the format options specified at upper levels with the following order of precedence
  - the SELECT level
  - the UNLOAD level
  - the GLOBAL that applies to the requested output format
  - the parmlib level that applies to the requested output format
- no REFORMAT clause is taken into account.

**FROM**
Specifies the table or view from which columns are selected.

creator.table
Columns are selected from the table.

creator.view
Columns are selected from the view.

When you define SELECT statements that select data from views, the SELECT statements that you code must meet the following conditions if you want Db2 HPU to process the SELECT statement directly:

- The view can be defined only on a unique table. You cannot include joins, views of views, or subselects in the view definition.
- Each column of the view must precisely correspond to a column of the table. You cannot use scalar functions, column functions, expressions, or literals.
- The view cannot contain a WHERE clause.

Db2 HPU can process view definitions that do not conform to these conditions if DB2 YES or DB2 FORCE is specified. In these cases, the retrieval of the data is performed by Db2, but all other processing of the data is performed by Db2 HPU.

location.creator.table
location.creator.viewname
If the location that is used is identical to the current server (local Db2), Db2 HPU checks whether the SELECT statement can be processed. Otherwise, depending on whether you specified YES, NO, or FORCE for the DB2 parameter, the SELECT statement is not processed natively by Db2 HPU, and Db2 processing is attempted.

**correlation-name**
You can use a correlation name in a fast select block. If the AS keyword is used or if the correlation name is delimited with quotation marks, you can use any correlation name that is accepted by Db2. If any of the following keywords are used as a correlation name, they must be enclosed in quotation marks or preceded by the AS keyword:

- SELECT
LIST(list-name)
Identifies the name of a list of objects that is defined by a LISTDEF control statement. The list can include table spaces, index spaces, databases, and partitions. The list cannot include LOB table spaces and directory objects. The list that is generated by the LISTDEFTBV identifies the following objects:

- The tables from which the data is to be unloaded. You can use the pattern-matching feature of LISTDEF.
- For partitioned table spaces, the partitions from which the data will be unloaded. These partitions are defined by the INCLUDE, EXCLUDE, and PARTLEVEL keywords in the LISTDEF statement.

When you specify the LIST keyword in the fast listdef select block, Db2 HPU generates a SELECT statement for each table of each table space that is contained in the list. Using a TEMPLATE statement in the OUTDDN keyword is recommended. To activate partition parallelism with a TEMPLATE statement for partitioned tables, the data set name that is defined in the TEMPLATE statement must contain the &PART variable.

LISTTBV(list-name)
Identifies the name of a list of objects that is defined by a LISTDEFTBV control statement. The list can include tables and views. The list cannot include LOB tables and directory objects. The list that is generated by the LISTDEFTBV identifies the following objects:

- The tables and views from which the data is to be unloaded. You can use the pattern-matching feature of LISTDEFTBV.
- For partitioned tables, the partitions from which the data will be unloaded. These partitions are defined by the INCLUDE, EXCLUDE, and PARTLEVEL keywords in the LISTDEFTBV statement.

When you specify the LISTTBV keyword in the fast listdeftbv select block, Db2 HPU generates a SELECT statement for each table contained in the list. Using a TEMPLATE statement in the OUTDDN keyword is recommended. To activate partition parallelism with a TEMPLATE statement for
partitioned tables, the data set name that is defined in the TEMPLATE statement must contain the &PART variable.

**CLONE**
Indicates that Db2 HPU is to unload data only from clone tables that are in the list of table spaces that are defined by the LISTDEF. If the list contains only table spaces without clones, no SELECT statement is generated.

When the CLONE keyword is not specified, Db2 HPU generates only SELECT statements on the base tables. The CLONE keyword can be specified only when you specify SELECT FROM LISTDEF. To unload data from a clone table by specifying SELECT FROM table_name, specify the name of the clone table in the FROM clause.

**ORIGINOBID**
Indicates when the OBID table in the image copy is not the same as the OBID that is read in the catalog. This situation can occur when an image copy of a table that is dropped and then re-created with a new OBID.

If the source data is an image copy, use this keyword to specify the OBID of the rows to be processed in this image copy.

*integer*
If the image copy file contains a unique table that you can use, you can specify 0 instead of the OBID of the table. If you specify the 0, Db2 HPU processes the first OBID that is found in the image copy.

*X’hhhh’*
X’hhhh’ is the hexadecimal value of the OBID of the table in the image copy.

This keyword must be specified with the COPYDDN statement.

**PART**
Specifies the physical partitions of the table space to be processed. You can specify the PART keyword in the UNLOAD block and the SELECT block.

- If PART is not specified in the UNLOAD block, then PART(ALL) is the default.
- If PART is not specified in a SELECT block of this UNLOAD block, the PART specification in the UNLOAD block is used.
- If you specify PART in any SELECT statement, it overrides any values that are specified in the UNLOAD block.
- For each UNLOAD statement, the union of all the subsets of partitions that are selected in all SELECT blocks must be equal to the subset that is specified in the UNLOAD PART, unless UNLDDN is used.
- If a statement is processed by Db2, PART is ignored.

**Requirement:** If you are unloading from partition-level full image copies, you must provide a uniquely named DD statement for each partition to be unloaded. For example, if one SELECT statement unloads from partitions 1, 2, and 4, and another SELECT statement unloads from partitions 2 and 3, you must specify a DD statement with a unique name for image copy partitions 1, 2, 3, and 4.

*integer*
Indicates which physical partitions are to be processed.

**ALL**
Specifies that the entire table space is to be processed.

*integer-1:integer-2*
Designates a range of physical partitions integer-1 -integer-2. Integer-1 must be positive and less than the highest partition number in the table space. Integer-2 must be greater than Integer-1 and less than or equal to the highest partition number.

The default value is ALL.

**Attention:** Do not use this option when the table space is simple or segmented.
SQLPART
Specifies the partitions of the table space to be processed. The SQLPART parameter applies to a physical partition number, which can be different from logical partitions. Unlike the PART keyword, you can specify SQLPART only in the SELECT block. SQLPART is used only when partitioned table spaces are unloaded.

SQLPART applies to SELECT statements that are processed natively by Db2 HPU and to SELECT statements that are processed by Db2. SQLPART is not accepted when either the LISTDEF option, the LUSTDEFTBV option or the DDLDDN option is used.

SQLPART is used only on table spaces that are partitioned by range or partitioned by index. Table spaces that are partitioned by growth or that are not partitioned are not supported.

SQLPART acts according to the following specifications in the UNLOAD command:

• If DB2 NO is specified, SQLPART is equivalent to PART.
• If DB2 FORCE is specified, SQLPART allows unloading per partition. In this case, a separate UNLOAD file can be used for each partition, and parallelism is activated between the unloaded partitions.
• If DB2 YES is specified, SQLPART is processed as PART for supported SELECT statements. For unsupported SELECT statements, SQLPART allows unloading per partition.

If SQLPART is used and you are unloading partitioned table spaces, the rules for correspondence between partitions and output files apply.

Specifying SQLPART(ALL) is different from not specifying SQLPART in the following ways:

• When SQLPART is not specified, data is entirely unloaded into a single file. If a TEMPLATE is used to create the UNLOAD file, the &PART variable in the TEMPLATE is replaced with '00000'.
• When SQLPART(ALL) is specified, the processing depends on whether one output file per partition is used. If a TEMPLATE that contains the &PART variable is used, Db2 HPU allocates one output file for each partition, and data is unloaded on a per-partition basis. If JCL-allocated files are used, the type of process depends on whether files per partition are used.

SQLPART is incompatible with the OUTFREQROWS keyword and the OUTMAXROWS keyword.

When SQLPART is used, Db2 HPU determines the name of the unloaded table. Joins and unions are not allowed on the processed SELECT statement.

If the limit key for FLOAT and DECFLOAT columns is not defined explicitly when the partitioning key is created, Db2 HPU does not support using SQLPART with key limits for FLOAT and DECFLOAT columns of the partitioning key.

The following examples show how the SQLPART option is used:

**Example: Obtaining separate files per partition with SQLPART in DB2 FORCE**
The following example shows an unloaded table space that contains six partitions. The execution JCL contains two DD statements, SYSREC, and SYSREC2, which are allocated to two output files.

```
UNLOAD TABLESPACE DBDM0A.TSTEST16
  QUIESCE NO
  LOCK YES
  DB2 FORCE
  SELECT * FROM TSTEST16 SQLPART(ALL)
  OUTDDN(SYSREC)
  FORMAT DSN7IAUL LOADDDN SYSPUNCH
```

Data from partition 1 and partitions 3-6 is unloaded in the file SYSREC, and data from partition 2 is unloaded in file SYSREC2. At least two SELECT statements are used internally for the following partitions:

• One SELECT statement is used for partition 2, which writes in file SYSREC2 (second message INZR3000).
• One to five SELECT statements are used for partition 1 and partitions 3-6, which write in file SYSREC (third message INZR3000). The effective number of SELECT statements depends on correspondence between the physical partitions and the logical partitions. If physical partitions...
3-6 correspond to four consecutive logical partitions, a single SELECT statement is used. Otherwise, Db2 HPU generates a SELECT statement for each range of adjacent logical partitions.

The following SYSPRINT corresponds to partitioned table space DBDM8A.TSTEST16:

```
UDBU281I - UNLOAD STARTING AT POS(2, 1)
UDBU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBDM8A.TSTEST16
UDBU360I - SELECT STATEMENT PROCESSING THROUGH DB2
UDBU280I - SELECT 00001 STARTING AT POS(6, 2)
UDBU282I OUTPUT DDNAME=SYSREC
INZR3000 NUMBER OF ROWS EXTRACTEDCopied ON SYSREC : 8333
INZR3000 NUMBER OF ROWS EXTRACTED Copied ON SYSREC2 : 8333
INZR3000 NUMBER OF ROWS EXTRACTED Copied ON SYSREC : 33334
UDBU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(6, 2)
UDBU222I SYSREC , TOTAL NUMBER OF RECORDS WRITTEN 41667
UDBU376I UNLOAD 1 SELECT 1 NUMBER OF RECORDS WRITTEN 41667
UDBU222I SYSREC2 , TOTAL NUMBER OF RECORDS WRITTEN 8333
UDBU376I UNLOAD 1 SELECT 1 PARTITION 2 NUMBER OF RECORDS WRITTEN 8333
```

**Example: Implicit usage of SORT EXTERNAL with SQLPART**

When an ORDER BY clause is coded on the SELECT statement, Db2 HPU can switch automatically to SORT(EXTERNAL) processing to be able to sort data that comes from different ranges of partitions that are unloaded in a single output file. The following example shows an implicit usage of SORT(EXTERNAL). A single SYSREC file is allocated in the JCL.

```
//UNLOAD EXEC PGM=INZUTILB,PARM='DZ8A,INZUTIL'
// REGION=0M,TIME=1440
//STEP1B DD DSN=INZ510.SINZLINK,DISP=SHR
// DD DSN=DSNA10.SDSNLOAD,DISP=SHR
// DD DSN=DSNA10.SDSNL02,DISP=SHR
//SYSSREC DD DSN=...
//UTPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSIN DD *
UNLOAD TABLESPACE DBDM8A.TSTEST16
QUIESCE NO LOCK YES
DB2 FORCE
SELECT * FROM TSTEST16 SQLPART(1:3,5:6)
ORDER BY COLINTEGER
OUTDDN(SYSREC)
FORMAT DSNTIAUL LOADDDN SYSPUNCH
//
```

In this example, two SELECT statements are generated to process the two ranges of partitions. Data from partitions 1 - 3 might be mixed with data from partitions 5 - 6 because both SELECT statements are processed in parallel. For these two ranges of partitions, Db2 HPU processes the sort that corresponds to the ORDER BY specification based on the result of merging the two SELECT statements to obtain data that is written in file SYSREC and sorted on column COLINTEGER.

When the SORT(EXTERNAL) option is specified in Db2 HPU, and the SORT specification is not supported natively, the sort is not done, warning message UDBU511W is issued, and the output data is not sorted.

When an ORDER BY clause is coded on the SELECT statement, Db2 HPU can automatically switch to SORT(EXTERNAL) processing to be able to sort data that comes from different ranges of partitions that are unloaded in a single output file.

**WHERE**

The following syntax diagram shows the format of the WHERE clause:

```
WHERE clause

WHERE search condition
```

**search condition**
search condition

NOT

predicate

The following predicates specify a comparison between two expressions.

• basic predicate
• between predicate
• in predicate
• like predicate
• null predicate

The following syntax diagram shows the predicate syntax:

**basic predicate**

```
expression  =  expression

<

>

<>

^=

-=

=<

>=
```

**between predicate**

```
expression  NOT  BETWEEN  expression  AND  expression  
```

**in predicate**

```
expression  NOT  (constant)  
```

**like predicate**

```
expression  NOT  LIKE  expression  ESCAPE  expression 
```

**null predicate**

```
expression  IS  NOT  NULL

NOT

ISNULL

NOTNULL
```
Notes:

1. The IN predicate compares an expression with a set of values. The result of this predicate is true if the value of the expression matches one in the list; otherwise, the result is false.
2. The LIKE predicate works the same as the standard SQL LIKE predicate.

**case expression**

```
CASE
  searched WHEN clause
  simple WHEN clause
  ELSE NULL
  ELSE expression
END
```

**searched WHEN clause**

```
WHEN search condition THEN result-expression
```

**simple WHEN clause**

```
expression WHEN expression THEN result-expression
```

**labeled duration**

```
constant
- YEAR
- YEARS
- MONTH
- MONTHS
- DAY
- DAYS
- HOUR
- HOURS
- MINUTE
- MINUTES
- SECOND
- SECONDS
- MICROSECOND
- MICROSECONDS
```

**expression**

An SQL expression is a combination of columns, functions and special registers and operators.

The following tables give the list of supported and unsupported SQL items (scalar functions and operators). Be aware that SQL statements containing only supported items may not be supported as some combinations may prevent Db2 HPU to process the statement in native mode.

The following table contains a list of supported and unsupported SQL:

<table>
<thead>
<tr>
<th>Items</th>
<th>Supported? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>Items</td>
<td>Supported? (1)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>ADD_MONTHS</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>AND</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>ANY</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>ASCII</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ASCII_CHR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ASCII_STR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ASIN</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ATAN</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>BIGINT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>BINARY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>BITAND</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>BITANDNOT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>BITNOT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>BITOR</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>BITXOR</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>BLOB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>CASE expression</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>CCSID_ENCODING</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>CEIL</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>CEILING</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>CHAR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>CHARACTER_LENGTH</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>CLOB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>COALESCE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>COLLATION_KEY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>COMPARE_DECFLOAT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>CONCAT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>COS</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>COSH</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>CURRENT</td>
<td>SUPPORTED TIME REGISTER</td>
</tr>
<tr>
<td>CURRENT_TIME</td>
<td>SUPPORTED TIME REGISTER</td>
</tr>
</tbody>
</table>

---

Table 29. Supported and unsupported scalar functions and operators (continued)
<table>
<thead>
<tr>
<th>Items</th>
<th>Supported? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT_TIMESTAMP</td>
<td>SUPPORTED TIME REGISTER</td>
</tr>
<tr>
<td>DATE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DAY</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DAYOFMONTH</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DAYOFWEEK</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DAYOFWEEK_ISO</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DAYOFYEAR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DAYS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DBCLOB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DEC</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DECFLOAT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DECFLOAT_FORMAT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DECFLOAT_SORTKEYS</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DECRYPT_BINARY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DECRYPT_BIT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DECRYPT_CHAR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DECRYPT_DB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DEGREES</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>DIGITS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DOUBLE_PRECISION</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>DSN_XMLVALIDATE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>EBCDIC_CHAR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>EBCDIC_STR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ENCRYPT_TDES</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>EXP</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>FLOOR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>GENERATE_UNIQUE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>GETHINT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>GETTABLE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>Items</td>
<td>Supported? (1)</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>GRAPHIC</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>GREATEST</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>HEX</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>HOUR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>HOURS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>IDENTIFY_VAL_LOCAL</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>IF statement</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>IFNULL</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>IN</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>INSERT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>INT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>INTEGER</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>IS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>JULIAN_DAY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>LAST_DAY</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LCASE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>LEAST</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LEFT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LENGTH</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>LIKE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>LN</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LOCAL</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>LOCATE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>LOG</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LOG10</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LOWER</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>LPAD</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>LTRIM</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MAX</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>MICROSECOND</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MICROSECONDS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MIDNIGHT_SECONDS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MIN</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MINUTE</td>
<td>SUPPORTED</td>
</tr>
</tbody>
</table>
Table 29. Supported and unsupported scalar functions and operators (continued)

<table>
<thead>
<tr>
<th>Items</th>
<th>Supported? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINUTES</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MOD</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MONTH</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MONTH_BETWEEN</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MONTHS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>MONTHS_BETWEEN</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>MQPUBLISH</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQPUBLISHXML</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQREAD</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQREADCLOB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQREADXML</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQRECEIVE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQRECEIVECLOB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQRECEIVEXML</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQSEND</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQSENDXML</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQSENDXMLFILE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQSENDXMLFILECLOB</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQSUBSCRIBE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MQUNSUBSCRIBE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>MULTIPLY_ALT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>NEXT_DAY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>NORMALIZE_DECFLOAT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>NORMALIZE_STRING</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>NOT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>NULLIF</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>NVL</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>OR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>OVERPLAY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>POSITION</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>POSSTR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>POWER</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>QUANTIZE</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>QUARTER</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>Items</td>
<td>Supported? (1)</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>RADIANS</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>RAISE_ERROR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>RAND</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>REAL</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>REPEAT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>RID</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>RIGHT</td>
<td>UNSUPPORTED (2)</td>
</tr>
<tr>
<td>ROUND</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ROUND_TIMESTAMP</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>ROWID</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>RPAD</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>RTRIM</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>SCORE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SECOND</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>SECONDS</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>SIGN</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>SIN</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SINH</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>SOAPHTTPC</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SOAPHTTPNC</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SOAPHTTPNV</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SOAPHTTPV</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SOUNDEX</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SPACE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>SQRT</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>STRIP</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>SUBSTR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>TAN</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TANH</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TIME</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>TIMESTAMP_FORMAT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TIMESTAMP_iso</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>Items</td>
<td>Supported? (1)</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>TIMESTAMPADD</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TIMESTAMPDIFF</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>TO_NUMBER</td>
<td>UNSUPPORTED (3)</td>
</tr>
<tr>
<td>TOTALORDER</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TRANSLATE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TRIM</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>TRUNC</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>TRUNC_TIMESTAMP</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>TRUNCATE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>UCASE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>UNICODE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>UNICODE_STR</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>UPPER</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>VALUE</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>VARBINARY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>VARCHAR_FORMAT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>VARGRAPHIC</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>VERIFY_GROUP_FOR_USER</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>VERIFY_ROLE_FOR_USER</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>VERIFY_TRUSTED_CONTECT_ROLE_FOR_USER</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>WEEK</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>WEEK_ISO</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLATTRIBUTES</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLCOMMENT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLCONCAT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLDOCUMENT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLELEMENT</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLFOREST</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLNAMESPACEŞES</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLPARSE</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLPI</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLQUERY</td>
<td>UNSUPPORTED (4)</td>
</tr>
<tr>
<td>XMLSERIALIZE</td>
<td>UNSUPPORTED (4)</td>
</tr>
</tbody>
</table>
Table 29. Supported and unsupported scalar functions and operators (continued)

<table>
<thead>
<tr>
<th>Items</th>
<th>Supported? (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>SUPPORTED</td>
</tr>
<tr>
<td>YEARS</td>
<td>SUPPORTED</td>
</tr>
</tbody>
</table>

Notes:
- SUPPORTED? (1): The column represents the list of scalars and words known or unknown by Db2 HPU.
- SUPPORTED: Scalar function known by Db2 HPU.
- UNSUPPORTED (2): Db2 HPU only qualifies the whole SQL statements as unsupported but not specific error message is issued for them.
- UNSUPPORTED (3): Db2 HPU issues a specific error message mentioning this item. The location is not mentioned within the SQL statement.
- UNSUPPORTED (4): Db2 HPU issues the INZU228E message and indicates the name and the location of this unsupported item.

ORDER BY
Indicates that the output data set must be sorted based on the chosen criteria.

ASC
Specifies that the rows are sorted in ascending order and the column name or an integer that identifies the nth column of the result table.

DESC
Specifies that the rows are sorted in descending order and the column name or an integer that identifies the nth column of the result table.

If the ORDER BY clause corresponds to the order of the clustering index, Db2 HPU tries to use this index to unload the rows in the required order. In all other cases, a SORT is issued.

The default value is ASC.

Attention: When an ORDER clause is specified along with the FORMAT INTERNAL, Db2 HPU automatically generates the PRESORTED YES option at the table level in the LOAD control cards if requested through the LOADDDN option. As a consequence, the target table to be loaded with the data produced for the unload should have a clustering index with a key compatible with the ORDER clause specified by the unload statement. If this is not the case, it is advisable to remove the ORDER clause that would cause extra processing time at UNLOAD time without any subsequent benefit at LOAD time.

ORDER CLUSTER
Indicates that the output data set must be sorted according to the index cluster. To display the sort messages, code the UPRINT ddname. If an index is not defined on the table, a warning message is issued and processing continues.

Restriction: ORDER CLUSTER is only supported with native mode processing. If a logical unload is processed in SQL mode, specifying ORDER CLUSTER leads to an SQL error=-199.

Attention: When an ORDER clause is specified along with the FORMAT INTERNAL option, Db2 HPU automatically generates the PRESORTED YES option at the table level in the LOAD control cards if requested through the LOADDDN option. As a consequence, the target table to be loaded with the data produced for the unload should have a clustering index with a key compatible with the ORDER clause specified by the unload statement. If this is not the case, it is advisable to remove the ORDER clause that would cause extra processing time at UNLOAD time without any subsequent benefit at LOAD time.

WITH UR
This clause is accepted for SELECT statements that are processed natively only for compatibility with SELECT statements that are processed by using Db2. Internal lock mechanisms do not apply because
the SELECT statements that are processed natively are processed outside of Db2. Specifying the WITH UR clause does not affect the results of SELECT statements that are processed natively and does not affect the integrity of data that is unloaded by Db2 HPU.

The LOCK and QUIESCE options apply to the SELECT statements that are processed natively or processed by using Db2 even if the WITH UR clause is specified.

**Restriction:** The WITH UR clause is not supported in the fast listdef select blocks.

The following example shows how you can unload data with the WITH UR clause in native mode:

```sql
UNLOAD TABLESPACE QUIESCE YES DB2 NO
SELECT * FROM DSN8810.EMP WITH UR
FORMAT DSNTIAUL OUTDDN(OUTPUT)
```

See the *Db2 Universal Database for z/OS SQL Reference* for more information about function parameters.

**Related concepts**

Db2 HPU ddname allocation: UTPRINT ddname.

Db2 HPU uses two types of ddnames: ddnames that Db2 HPU allocates and ddnames that you must allocate.

**Related reference**

FORMAT block syntax and description

Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

SELECT statement examples

These SELECT statement examples show how to create sequential data sets in different formats, how to use the LISTDEF and TEMPLATE keywords, how to use the INTO clause and the REFORMAT clause, and how to use other Db2 HPU options.

OPTIONS block syntax and description

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

**Example: Unloading data from a table with informational header**

This example shows how to unload data from a table with a line header saying from which Db2 subsystem the data has been unloaded.

```sql
UNLOAD TABLESPACE DB2 NO
SELECT CHAR(CHAR('SOURCE_SSID='),!! CHAR(GETVARIABLE('SYSIBM.SYSTEM_NAME')),16),
C1,
C2,
C3
FROM MZLFLB.TBDZ9AFLB00101 OUTDDN(SYSRECC01)
FORMAT DELIMITED
```

**FORMAT block syntax and description**

Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

The following diagram shows the syntax of the FORMAT block:
Notes:

1 Important: This feature is deprecated and is only supported for compatibility purpose. See “Input and output/Output formats/Performing logical unloads/Deprecated feature: replacement features” for more details.

You can specify the following keywords and options in the FORMAT block:

**FORMAT**
- Indicates that this block is a FORMAT block. Use the FORMAT keyword to specify the format of the data that is unloaded.

**DSNTIAUL dsntiaul block**
- Unloads data in the same format that is produced by the DSNTIAUL program.
  - **Important:** Depending on the content of the VUU057/OPALLFMT parameter, keywords and options that you specify in a GLOBAL block or UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, and UNLROWSET keywords. The LOADOPT and UNLROWSET keywords apply to all formats. The FLOAT keyword applies to DSNTIAUL, USER, and VARIABLE formats.

**DELIMITED delimited block**
- Unloads data in DELIMITED format.
  - **Important:** Depending on the content of the VUU057/OPALLFMT parameter, options that you specify in a GLOBAL block or UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, and UNLROWSET keywords. The LOADOPT and UNLROWSET keywords apply to all formats. The FLOAT keyword applies to DSNTIAUL, USER, and VARIABLE formats.

**VARIABLE variable block**
- Unloads data in a format that is compatible with the Db2 LOAD data set.
  - **Important:** Depending on the content of the VUU057/OPALLFMT parameter, options that you specify in a GLOBAL block or UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, and UNLROWSET keywords. The LOADOPT and UNLROWSET keywords apply to all formats. The FLOAT keyword applies to DSNTIAUL, USER, and VARIABLE formats.

**USER user block**
- Unloads data in the defined format. You can specify the format of a specific column using a USER block.
EXTERNAL
Unloads data in EXTERNAL format. The EXTERNAL keyword contains the following characteristics:
• Output records are fixed by default.
• Variable-length columns that are at the maximum length are preceded with the two length bytes and are padded on the right with binary zeros.
• Nullable fields are followed by a byte that contains x'00' if the data is not null and contains '?' if the data is null.
• A field separator is not used.
All fields are in the EXTERNAL format that corresponds to their default type.

INTERNAL
Unloads data in the Db2 INTERNAL format. This format is compatible with the INTERNAL format in the Db2 LOAD utility. The INTERNAL keyword is limited by the following restrictions:
• The INTERNAL keyword is accepted only on natively processed SELECT statements.
• The INTERNAL keyword is allowed only on a SELECT * FROM table-name statement or a SELECT FROM LIST (listdef-name) statement or a SELECT FROM LISTTBV (listdef-name) statement.
• The INTERNAL keyword is not accepted on views.
• The INTERNAL keyword cannot be combined with other format types in SELECT statements that are processed on the same table space with the same UNLOAD command.
• The INTERNAL keyword is not supported with tables that contain a LOB column or an XML column.
• When you use the INTERNAL keyword, an OPTIONS block that is used to change the formatting of the output record or a REFORMAT clause is ignored.
• If you specify the INTERNAL keyword, the OUTEXIT and CCSID keywords or encoding scheme specifications are ignored.
• You cannot specify the INTERNAL keyword with the OUTMAXROWS or OUTFREQROWS keywords.
• The INTO clause is not allowed for SELECT statements that use the INTERNAL keyword.
• The ORDER CLUSTER and ORDER BY clauses are accepted with the INTERNAL keyword in the following conditions:
  – The table is in reordered row format (RRF).
  – The table is in basic row format (BRF), and the ORDER CLUSTER clause or the ORDER BY clause does not include columns that are located after the first variable-length column in the table.

When an ORDER clause is specified along with the FORMAT INTERNAL option, Db2 HPU automatically generates the PRESORTED YES option at the table level in the LOAD control cards if requested through the LOADDDN option. As a consequence, the target table to be loaded with the data produced for the unload should have a clustering index with a key compatible with the ORDER clause specified by the unload statement. If this is not the case, it is advisable to remove the ORDER clause that would cause extra processing time at UNLOAD time without any subsequent benefit at LOAD time.
• When the INTERNAL keyword is specified, the setting of the VUU050/ULFRQMSG parameter is ignored, and information messages that display the number of currently written records are not displayed.

The following example shows how you can request a logical unload of data by using the INTERNAL keyword. Data from the DSN8810.EMP table is unloaded in the INTERNAL format.

```
UNLOAD TABLESPACE QUIESCE YES DB2 NO
SELECT * FROM DSN8810.EMP FORMAT INTERNAL OUTDDN(OUTPUT)
```
LOADDDN
Specifies the name of the DD statement that describes the command data set. Use this keyword if you want Db2 HPU to create a command data set for the Db2 LOAD utility. The corresponding DD statement must be present in the execution JCL. This data set contains the required commands for loading a sequential data set by using the Db2 LOAD utility.

If the LIKE table-name clause is not used, the model table is the table that is referred to in the SELECT statement.

Attention: If both an ORDER clause and the FORMAT INTERNAL option are specified, Db2 HPU automatically generates the PRESORTED YES option at the table level in the LOAD control cards. As a consequence, the target table to be loaded with the data produced for the unload should have a clustering index with a key compatible with the ORDER clause specified by the unload statement.

LOADOPT
For a description of this keyword, see “OPTIONS block syntax and description” on page 143.

Attention: If both an ORDER clause and the FORMAT INTERNAL option are specified, as Db2 HPU automatically generates the PRESORTED YES option at the table level in the LOAD control cards, the LOADOPT option should not specify the PRESORTED keyword.

The description of the XML keyword is the following one:

DELM delimeter character
Specifies the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC and VARGRAPHIC fields in the output data set.

The value of the delimiter-character can be 'c' or X'hh'
DELM does not have a default value. The value are not delimited if no DELIM option is specified.

NULLVAL character expression
Specifies the character string to be written into the output data set when a NULL value is to be unloaded. This character string is not delimited even if the DELIM option is specified.

If NULLVAL is not specified, the NULL value are not substituted and an empty string with no delimiter is unloaded.

Related reference
Fast select, fast listdef and fast listdeftbv select blocks syntax and description
The fast select, fast listdef and fast listdeftbv select blocks describe the syntax for SELECT statements that can be supported natively by Db2 HPU when DB2 NO or DB2 YES is specified in the UNLOAD command. The fast select, fast listdef and fast listdeftbv select blocks are part of the SELECT block.

SELECT statement examples
These SELECT statement examples show how to create sequential data sets in different formats, how to use the LISTDEF and TEMPLATE keywords, how to use the INTO clause and the REFORMAT clause, and how to use other Db2 HPU options.

OPTIONS block syntax and description
Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

DSNTIAUL block syntax and description
Use the DSNTIAUL block to put the output data set in the same format that is produced by the DSNTIAUL program.

DELIMITED block syntax and description
Use the DELIMITED block to indicate that the format of the output data set is a comma-separated-value file, which corresponds to a .csv file that can be processed by your personal computer.

VARIABLE block syntax and description
Use the VARIABLE block to indicate that the output data set must be compatible with the Db2 LOAD data set.

USER block syntax and description
Use the USER block to indicate that the unloaded data is formatted according to the keywords that are specified in the USER block.

Db2 HPU output data parameters
The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

**DSNTIAUL block syntax and description**

Use the DSNTIAUL block to put the output data set in the same format that is produced by the DSNTIAUL program.

The DATE and TIME columns are unloaded based on Db2 installation parameters.

The default data set record format is fixed blocked (FB), but it can be specified as fixed (F), variable (V), or variable blocked (VB) in the JCL.

If the LRECL is specified, it is kept by Db2 HPU. In this case, data can be truncated.

If the LRECL is not specified, Db2 HPU determines it at run time based on the following rules:

- If you specified RECFM=F, the LRECL of the data set is equal to the sum of the maximum length of fields, regardless of the LRECL value that you specified in the JCL. The output data set is in FB format.
- If you specified RECFM=V or RECFM=VB, the LRECL of the data set is equal to the sum of the maximum length of fields plus 4 bytes, regardless of the LRECL value that you specified in the JCL. The output data set is in VB format.

The length of variable-length fields is the sum of their maximum length plus 2 bytes.

DSNTIAUL STRICT affects the formatting of constant character strings that are specified in SELECT statements. For example:

```sql
SELECT 'character-string'
```

If you specify DSNTIAUL, `character-string` is written to the output data set as type CHAR.

If you specify DSNTIAUL STRICT, `character-string` is written to the output data set as type VARCHAR.

The following syntax diagram shows the DSNTIAUL block:

```
DSNTIAUL block
LIKE table-name
```

If the LIKE `table-name` clause is not specified, the characteristics and sequence of fields in the generated data set are the same as the selected columns. The format of data that is written in the data set is in the same format that is generated by the DSNTIAUL program:

- Numeric data has a standard format.
- Columns of type DATE, TIME, or TIMESTAMP have the default format for the site.
- Variable-length fields (VARCHAR, LONG VARCHAR, VARGRAPHIC, and LONG VARGRAPHIC) contain length bytes at the beginning of the field. The field is extended to the maximum size of the column.
- If a column accepts nulls, a null-byte indicator is generated into the encoding scheme (EBCDIC, ASCII, or UNICODE) and into the CCSID of the output data. This indicator contains the value `?` when the field is null, or a binary zero (x'00') if the value is not null. The hexadecimal value of `?` depends on the encoding scheme.

**LIKE `table-name`**

If the LIKE `table-name` clause is specified, Db2 HPU uses the characteristics of the table model as parameters and formats the data set to allow this table to be loaded.

The sequence of columns in the SELECT statement must match the columns in the model table. If format conversions are required, they follow Db2 HPU format rules.
The **table-name** must be in the format **user_ID.TABLE**. If the user ID is not specified, the name of the user (CURRENT SQLID) is used.

Field types are provided by the table model. A null-byte indicator is generated behind the field if the column of the table model can be null.

If a column in the SELECT statement is null and if the corresponding column in the table model does not allow nulls, the field in the output sequential data set has one of the following standard default values:

- 0 for numeric fields (INTEGER, SMALLINT, DECIMAL, FLOAT)
- A string of blank characters (CHAR, GRAPHIC)
- Two bytes of zero length (VARCHAR, LONG VARCHAR, VARGRAPHIC, LONG VARGRAPHIC)
- 0001-01-01 (DATE)
- 00.00.00 (TIME)
- 0001-01-01-00.00.00.000000 (TIMESTAMP)

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

**Related reference**

**VARIABLE** block syntax and description

Use the VARIABLE block to indicate that the output data set must be compatible with the Db2 LOAD data set.

**DELIMITED** block syntax and description

Use the DELIMITED block to indicate that the format of the output data set is a comma-separated-value file, which corresponds to a .csv file that can be processed by your personal computer.

When numeric and date, time, and timestamp data is unloaded in DELIMITED format, the data is always unloaded in an external, displayable format, even if an INTO clause or a REFORMAT clause specifies a conversion to INTERNAL format. For example, specifying REFORMAT (TYPE SMALLINT INTO DECIMAL(5,0)) generates a field that contains decimal external data. When data is converted into date, time, or timestamp data by using an INTO clause or a REFORMAT clause, the default output format is the ISO representation of this data. For example, specifying COL_DATE INTO C1 DATE returns a field in the ISO representation of a date column.

The following syntax diagram shows the format of the DELIMITED block:

```
DELIMITED block

SEP separator-character

DELIM delimiter-character

NULLVALL 'string'

NULL EMPTY WITH NO DELIM

NULL DELIM 1

TITLE OFF

TITLE ON
```

**Notes:**

1. deprecated keywords, use NULL EMPTY WITH NO DELIM for better clarity

**SEP separator-character**

Specifies the separator character to be used to separate fields in the output data set.

The value of the separator-character can be 'c' or X'hh'.
The default value is (X'40').

**DELIM delimiter-character**

Specifies the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set.

The value of the *delimiter-character* can be 'c' or X'hh'.

DELIM does not have a default value.

**NULLVAL 'string'**

Specifies the string value that is written in the output field whenever the corresponding column table has a NULL value. The specified string must be surrounded between apostrophes but it is not enclosed in any delimiter in the output data set. The string is written as is regardless of any format. The string is converted into the output file CCSID. The value " " (quote quote) specifies that the NULL value is an empty string.

**NULL EMPTY WITH NO DELIM**

Specifies that NULL values are unloaded as empty strings regardless of the data type of the related column. A NULL value is identified in the output record by two consecutive separators.

*Note:* Replaces former NULL DELIM option for better clarity.

**NULL DELIM**

Same as NULL EMPTY WITH NO DELIM. This deprecated syntax item is kept for backward compatibility purpose.

**NULL EMPTY WITH DELIM**

Specifies that NULL values are unloaded as empty strings and, for string data types, the empty string is enclosed by the delimiter character that is specified by DELIM delimiter-character.

**TITLE**

Specify whether a title is requested for the columns of the output data unloaded in DELIMITED format.

**OFF**

the output data set does not contain any title line.

**ON**

the first line of the output data set is a title line built from the columns names of the target table, enclosed by the DELIM character and separated by the SEP character. The name of the target column is determined by the first rules of the following ordered list that applies.

- The name defined by an INTO clause, if any.
- The name of the column, if the target column exactly matches one of the columns from a source table.
- "COLnnn" - where nnn is the rank of the considered target column in the SELECT clause - if none of the previous situation applies.

The LRECL of the output file that includes a title line, if no forced by any user specification, is set to the greatest value between the maximum data record length and the title line length. TITLE ON is supported only if the output ddname (specified by the OUTDDN option) is dedicated to a single SELECT.

The title line is encoded with the Db2 subsystem encoding scheme (ENSCHEME of the Db2 settings) and with the SBCS CCSID (SCCSID of the Db2 settings) for a non-mixed subsystem and the mixed CSSID (MCCSID of the Db2 settings) for a mixed subsystem.

These options are mutually exclusive. If none is specified, the content of the output fields for NULL values is determined according to the value of the VUU078/ULNULLFL parameter.

To ensure that you can distinguish between empty values and null values, specify all three keywords.

The following table shows the results of different settings for a row that contains five columns that have the following values:
Table 30. Results of selected DELIMITED block keyword settings

<table>
<thead>
<tr>
<th>DELIMITED block setting</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEP ';'</td>
<td>BEGINNING;;;;END</td>
</tr>
<tr>
<td>SEP ';' DELIM '/'</td>
<td>/BEGINNING;/;/;END/</td>
</tr>
<tr>
<td>SEP ';' DELIM '/' NULL DELIM</td>
<td>/BEGINNING;/;/;END/</td>
</tr>
</tbody>
</table>

The DATE, TIME, and TIMESTAMP columns are in ISO format and correspond to the following formats:

**DATE**

```
YYYY-MM-DD
```

**TIME**

```
HH.MM.SS
```

**TIMESTAMP**

```
YYYY-MM-DD-HH.MM.SS.NNNNNN
```

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

**VARIABLE block syntax and description**

Use the VARIABLE block to indicate that the output data set must be compatible with the Db2 LOAD data set.

The default format of the output data set is variable block (VB), but you can specify fixed (F), fixed block (FB), or variable (V) in the JCL. Db2 HPU determines the LRECL at run time based on the following rules:

- If you specified RECFM=F or RECFM=FB, the LRECL must be larger than or equal to the sum of the lengths of the fields.
- If you specified RECFM=V, the LRECL must be larger than or equal to the sum of the lengths of the fields plus 4 bytes.

The length of variable-length fields is the sum of their maximum length plus 2 bytes.

The following syntax diagram shows the VARIABLE block:

```
VARIABLE block
    END
    ALL LIKE table-name
```

**END**

The characteristics and the sequence of fields in the output data set correspond to the characteristics and the sequence of fields in the SELECT statement.

The fields in the data set are also like the fields in the data set in DSNTIAUL format except for the following differences:
The DATE, TIME, and TIMESTAMP columns are in ISO format and correspond to the following formats:

**DATE**

`YYYY-MM-DD`

**TIME**

`HH.MM.SS`

**TIMESTAMP**

`YYYY-MM-DD-HH.MM.SS.NNNNNN`

- If a column accepts nulls, the null indicator is generated at the beginning of the field. This indicator contains the value X'FF' if the field is null and X'00' if the value is usable.
- If the last-selected column is variable, the output data set is VB, and this column is written only on its effective length. The two length bytes are placed before the column.

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

**ALL**

All the variable columns are written using their actual length.

**LIKE table-name**

If the LIKE *table-name* clause is specified, Db2 HPU uses the characteristics of the table model as parameters and formats the data set to allow this table to be loaded.

The sequence of columns in the SELECT statement must match the columns in the model table. If format conversions are required, they follow Db2 HPU format rules.

The *table-name* must be in the format `user_ID.TABLE`. If the user ID is not specified, the name of the user (CURRENT SQLID) is used.

Field types are provided by the table model. A null-byte indicator is generated at the end of the field if the column of the table model can be null.

If a column in the SELECT statement is null and if the corresponding column in the table model does not allow nulls, the field in the output sequential data set has one of the following standard default values:

- 0 for numeric fields (INTEGER, SMALLINT, DECIMAL, FLOAT)
- A string of blank characters (CHAR, GRAPHIC)
- Two bytes of zero length (VARCHAR, LONG VARCHAR, VARGRAPHIC, LONG VARGRAPHIC)
- 0001-01-01 (DATE)
- 00.00.00 (TIME)
- 0001-01-01-00.00.00.000000 (TIMESTAMP)

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

**Related reference**

- **DSNTIAUL block syntax and description**
  Use the DSNTIAUL block to put the output data set in the same format that is produced by the DSNTIAUL program.

- **USER block syntax and description**
  Use the USER block to indicate that the unloaded data is formatted according to the keywords that are specified in the USER block.

  You can change field attributes for all selected columns, which means that you can specify several keywords for each column according to the type of data that the column contains.

  The default values are determined by the values that are specified in the OPTIONS block.
If all the unloaded fields are fixed, the default value of RECFM is FB. If at least one output field is variable, the default value of RECFM is VB.

If the LRECL is not specified, Db2 HPU determines it at run time based on the following rules:

- If you specified RECFM=F, the LRECL of the data set is equal to the sum of the maximum length of fields, regardless of the LRECL value in the JCL. The output data set is in FB format.
- If you specified RECFM=V or RECFM=VB, the LRECL of the data set is equal to the sum of the maximum length of fields plus 4 bytes, regardless of the LRECL value in the JCL. The output data set is in VB format.

The following syntax diagram shows the USER block:

**USER block**

```
USER block
```

```
COL column-name|column-number
```

A description of the keywords that specify these attributes follows:

**TYPE val**

Specifies the type of output field. The TYPE keyword specifies the conversion to be performed. The following example shows how you might specify the type of output field: TYPE CHAR(10).

The default field format for output records is the format that was specified for columns in the SELECT statement.

**column-format-options**

Specifies the formatting options of the output data.

**Important:** This feature is deprecated and is only supported for compatibility purpose. For more details, see Redundant thus deprecated feature section.

**Related reference**

Data types for output (TYPE keyword)
The TYPE keyword of the SELECT statement (OPTION block for FORMAT USER) is used to create several types of data in the output. These types are declared in the keyword TYPE. The use of this keyword implies that data is to be converted from the original column type to the type that is declared in the TYPE keyword.

Supported conversions
Db2 HPU supports conversions to specific output data types.

OPTIONS block syntax and description
Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

**INTERNAL block syntax and description**

Use the INTERNAL block to indicate that the unloaded data is formatted according to the keywords that are specified in the INTERNAL block.

The following syntax diagram shows the INTERNAL block:
**ALTERNATE FORMAT format_description / DEFAULT**

Specifies the FORMAT to use when INTERNAL format is not suitable, e.g., when a not supported SELECT statement is specified or if LOB/XML data is unloaded. ALTERNATE FORMAT can be specified only if the SELECT statement specifies a FROM LIST or FROM LISTTBV clause. The chosen format, either the INTERNAL or the alternate one applies at the table space level. If only a table of a multi-table is not eligible for the INTERNAL format, the data from all the tables of the table space are unloaded with the alternate format. If ALTERNATE FORMAT is not specified, the job terminates in error whenever the INTERNAL format is not suitable or supported for one of the tables involved in the SELECT * FROM LIST(listdef_name) statement. Specify the alternate format to be used as follows:

- **format_description**
  - is the description of an output format to be used as an alternate format. It may be chosen from the list of formats allowed as a FORMAT option, excluding INTERNAL. The format indicated is described the same way as when specified as an argument of the FORMAT option.

- **DEFAULT**
  - specifies that the alternate format is the Db2 HPU default format set by the VUU045/ULFORMAT parmlib parameter.

As this feature allows you to implicitly specify the output format, in principle, it can only be determined at run time and you may need to determine which format was chosen for post-processing purposes. You can determine which output format was used with the help of the INZU538I message that is displayed whenever the alternate format is chosen.

If you request the LOAD control cards be generated via the LOADDYN option, you can easily implement massive load/unload operations by using the couples of unloaded data datasets and their related LOAD control cards that Db2 HPU generates consistently. In such a case, you don’t need to determine the format chosen by Db2 HPU for a given SELECT statement.

**Note:** The use of ALTERNATE_FORMAT implies that the format for a given SELECT statement can change from one run of the unload job to another. It can change from the INTERNAL format to the alternate one and inversely. This occurs if a change of the definition of the table involved in the SELECT statement affects its being eligible for the INTERNAL format.

**XML block syntax and description**

Unloads data in XML format.

The following syntax diagram shows the format of the XML block:

**DELIM delimiter-character**

Specifies the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set.

The value of the **delimiter-character** can be 'c' or X'hh'.

---

216  Db2 High Performance Unload User's Guide
DELIM does not have a default value. The value is not delimited if no DELIM option is specified.

**NULL DELIM character expression**

Specifies the delimiter character string to be unloaded when a NULL value is unloaded. This character string is not delimited even if if the DELIM option is specified.

If NULLVAL not specified, the NULL value is not substituted and an empty string with no delimiter is unloaded.

**SELECT statement examples**

These SELECT statement examples show how to create sequential data sets in different formats, how to use the LISTDEF and TEMPLATE keywords, how to use the INTO clause and the REFORMAT clause, and how to use other Db2 HPU options.

The SELECT statement examples are based on the PERSONAL and PERSOBIS tables.

The following SQL statement shows how to create the PERSONAL table.

```
CREATE TABLE PERSONAL
  ( NAME     CHARACTER(30) NOT NULL                ,
    FNAME    CHARACTER(15)                         ,
    AGE      SMALLINT                              ,
    ADDRESS  VARCHAR(100) NOT NULL WITH DEFAULT   ,
    DATE_B   DATE          NOT NULL WITH DEFAULT   ,
    SALARY   DECIMAL(7,0)  NOT NULL                ,
    BONUS    DECIMAL(7,0)  NOT NULL WITH DEFAULT   )     IN TABLESPACE DBNAME.TSNAME
```

The following SQL statement shows how to create the PERSOBIS table.

```
CREATE TABLE PERSOBIS
  ( WHOLE_NAME   CHARACTER(40) NOT NULL            ,
    AGE          SMALLINT      NOT NULL            ,
    DATE_B       DATE                              ,
    COMPENSATION INTEGER       NOT NULL            )     IN TABLESPACE DBNAME.TSNAME
```

**Related reference**

Fast select, fast listdef and fast listdeftbv select blocks syntax and description

The fast select, fast listdef and fast listdeftbv select blocks describe the syntax for SELECT statements that can be supported natively by Db2 HPU when DB2 NO or DB2 YES is specified in the UNLOAD command. The fast select, fast listdef and fast listdeftbv select blocks are part of the SELECT block.

**FORMAT block syntax and description**

Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

**SELECT block syntax and description**

The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

**Example: Creating a sequential data set in USER format**

In this example, a sequential data set is created in USER format from the PERSONAL table.

```
UNLOAD TABLESPACE DBNAME.TSNAME DB2 NO
SELECT FNAME, NAME, ADDRESS, DATE_B, SALARY
FROM PERSONAL
OUTDDN (SYSUT1)
FORMAT USER
  ( COL FNAME NULLID YES            ,
    COL 3 TYPE CHARACTER(100)      ,
    COL SALARY TYPE CHARACTER(7)   ,
    COL 004 TYPE DATE_A            )
```

In this example, data in the following three columns is changed:

- The ADDRESS column is converted to fixed format and its length is increased to 100 characters.
- The SALARY column is converted from decimal to display characters and it is aligned on the units position with the sign first.
• The DATE column is formatted to display in MM/DD/YY format.

A NULL byte is reserved before the NAME field.

The following output record format is produced:

<table>
<thead>
<tr>
<th>NULL</th>
<th>FNAME</th>
<th>NAME</th>
<th>ADDRESS</th>
<th>DATE_B</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>15 bytes</td>
<td>30 bytes</td>
<td>100 bytes</td>
<td>8 bytes</td>
<td>7 bytes</td>
</tr>
</tbody>
</table>

161 bytes

Figure 13. Output record format: sequential data set in USER format

By default, the output data set is in VARIABLE format with a maximum LRECL of 165, including the 4-byte field that describes the record of variable length. If the LRECL of the DD statement is greater than or equal to 161, you can produce a data set in fixed format.

Related reference

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.
FROM PERSONAL
OUTDDN SYSUT1
FORMAT DSNTIAUL
LIKE user.PERSOBIS
LOADDDN SYSUT2

In this example, data in the following columns is changed:

- The NAME column is changed from 30 to 40 characters. Blanks are added to the right.
- The DATE column is written in Db2 format.
- The SALARY column is converted to binary format.
- The AGE column in the SELECT statement can be null, but the AGE column of the PERSOBIS table cannot be null. If the column is null, the field is set to binary 0.

The data set with the SYSUT1 ddname contains the rows that were extracted. The data set with the SYSUT2 ddname contains the command for loading data by using the Db2 LOAD utility.

The following output record format is produced:

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>DATE</th>
<th>NULL</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 bytes</td>
<td>2 bytes</td>
<td>8 bytes</td>
<td>1 byte</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>

**Figure 14. Output record format: sequential data set in DSNTIAUL format**

The following data set loads SYSUT2 output, which was generated by Db2 HPU:

```
LOAD DATA   INDDN SYSUT1
INTO TABLE  PERSOBIS
(  WHOLE_NAME           POSITION(1:40)  CHAR,
   AGE                  POSITION(41:42)  SMALLINT,
   DATE_B               POSITION(43:50)  DATE EXTERNAL NULLIF((51)='?'),
   COMPENSATION         POSITION(52:55)  INTEGER )
```

**Related reference**

Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

**Example: Creating a sequential data set that is compatible with the Db2 LOAD utility**
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

```
UNLOAD TABLESPACE DBNAME.TSNAME DB2 NO
  SELECT NAME, AGE, DATE_B, SALARY
  FROM PERSONAL
  OUTDDN SYSUT1
  FORMAT VARIABLE END
  LIKE user.PERSOBIS
  LOADDN SYSUT2
```

The columns are extracted from the PERSONAL table by using the format of the PERSOBIS table.

In this example, data in the following columns is changed:
- The NAME column changes from 30 to 40 characters. Blanks are used to pad the column.
- The DATE column is written in ISO format.
- The SALARY column is converted to binary format.
- The AGE column in the SELECT statement can be null, but the AGE column of the PERSOBIS table cannot be null. If the column is null, the field is set to binary 0.

The SYSUT1 ddname contains rows that were extracted. The SYSUT2 ddname contains the command for loading data by using the Db2 LOAD utility.

The following output record format is produced:

<table>
<thead>
<tr>
<th>NAME</th>
<th>AGE</th>
<th>NULL</th>
<th>DATE ISO</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 bytes</td>
<td>2 bytes</td>
<td>1 byte</td>
<td>8 bytes</td>
<td>4 bytes</td>
</tr>
</tbody>
</table>

*Figure 15. Output record format: sequential data set that is compatible with the Db2 LOAD utility*

The following data set loads SYSUT2, which was generated by Db2 HPU:
Related reference

Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

The TEMPLATE OUT1 contains two user-defined variables, &CONST and &NUMBER. The &NUMBER variable is set in the GLOBAL OPTIONS block to the select number with a TEMPLATESET (NUMBER = :SELNUM). This setting is available for all UNLOAD commands and SELECT statements. The &CONST variable is set to a different value in each UNLOAD command by using a different TEMPLATESET option.

```sql
TEMPLATE OUT1 DSN HLQ.&CONST.&NUMBER. UNIT SYSDA
GLOBAL OPTIONS TEMPLATESET( NUMBER = :SELNUM )
UNLOAD TABLESPACE
OPTIONS TEMPLATESET( CONST = "C" )
SELECT * FROM TABLE1
OUTDDN(OUT1)
FORMAT DSNTIAUL
SELECT * FROM TABLE2
OUTDDN(OUT1)
FORMAT DSNTIAUL
```
Four files are dynamically allocated by using the OUT1 template. These files contain the following output data:

- The HLQ.C00001 file contains rows that were unloaded from TABLE1.
- The HLQ.C00002 file contains rows that were unloaded from TABLE2.
- The HLQ.D00001 file contains rows that were unloaded from TABLE3.
- The HLQ.D00002 file contains rows that were unloaded from TABLE4.

**Related reference**

*Example: Creating a sequential data set in USER format*

In this example, a sequential data set is created in USER format from the PERSONAL table.

*Example: Creating a sequential data set in DSNTIAUL format*

In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

*Example: Creating a sequential data set that is compatible with the Db2 LOAD utility*

In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

*Example: Using LISTDEF and TEMPLATE statements*

In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

*Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference*

In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

*Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format*

In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

*Example: Using an HFS file when unloading LOB data*

When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

*Example: Using temporal tables*

These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

*Example: Using timestamp precision*

These examples show how you can use timestamp precision.

*Example: Maximizing resources during sort operations*

In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

*Example: Using LISTDEF and TEMPLATE statements*

In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

An output file that uses the OUTTEMP template and a LOAD SYSIN that uses the LOADTEMP template is generated for each table. A TEMPLATESET is used to set the value of the &SEL user-defined variable that
is used in the template definitions to :SELNUM. This value is increased by one for each unloaded table starting with 00001.

```sql
LISTDEF LIST1 INCLUDE TABLESPACE DBTEST.*
TEMPLATE OUTTEMP DSN HLQ.&DB..&TS..&SEL..UNLOAD
TEMPLATE LOADTEMP DSN HLQ.&DB..&TS..&SEL..LOADD
GLOBAL OPTIONS TEMPLATESET( SEL = :SELNUM )
UNLOAD TABLESPACE
SELECT * FROM LIST(LIST1) OUTDDN OUTTEMP FORMAT DSNTIAUL LOADDDN LOADTEMP
```

The following example uses LISTDEF and TEMPLATE statements on the DBTEST.TS1, DBTEST.TS2, DBTEST.TS3, and DBTEST.TS4 partitioned table spaces. TS1 and TS2 are unloaded using a separate file per partition. TS3 and TS4 are unloaded in a single output file. All partitions of TS1 and TS3 are unloaded, and only partitions 1 and 3 are unloaded for TS2 and TS4.

```sql
LISTDEF LIST1 INCLUDE TABLESPACE DBTEST.TS1
INCLUDE TABLESPACE DBTEST.TS2 PARTLEVEL 1
INCLUDE TABLESPACE DBTEST.TS2 PARTLEVEL 3
LISTDEF LIST2 INCLUDE TABLESPACE DBTEST.TS3
INCLUDE TABLESPACE DBTEST.TS4 PARTLEVEL 1
INCLUDE TABLESPACE DBTEST.TS4 PARTLEVEL 3
TEMPLATE OUTTEMP1 DSN HLQ.&DB..&TS..UNLOAD.P&PART.
TEMPLATE OUTTEMP2 DSN HLQ.&DB..&TS..UNLOAD
TEMPLATE LOADTEMP DSN HLQ.&DB..&TS..LOADD
UNLOAD TABLESPACE
SELECT * FROM LIST(LIST1) OUTDDN OUTTEMP1 FORMAT DSNTIAUL LOADDDN LOADTEMP
SELECT * FROM LIST(LIST2) OUTDDN OUTTEMP2 FORMAT DSNTIAUL LOADDDN LOADTEMP
```

The following files are generated:

**Output files**
- HLQ.DBTEST.TS1.UNLOAD.Pnnnnn (one file per partition)
- HLQ.DBTEST.TS2.UNLOAD.P00001
- HLQ.DBTEST.TS2.UNLOAD.P00003
- HLQ.DBTEST.TS3.UNLOAD
- HLQ.DBTEST.TS4.UNLOAD

**LOADDDN files that contain the LOAD commands**
- HLQ.DBTEST.TS1.LOAD
- HLQ.DBTEST.TS2.LOAD
- HLQ.DBTEST.TS3.LOAD
- HLQ.DBTEST.TS4.LOAD

**Related reference**

**Example: Creating a sequential data set in USER format**

In this example, a sequential data set is created in USER format from the PERSONAL table.

**Example: Creating a sequential data set in DSNTIAUL format**

In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

**Example: Creating a sequential data set that is compatible with the Db2 LOAD utility**

In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

**Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements**
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

The tables are selected by using a LISTDEF statement. All tables from all table spaces that are contained in the DBTEST1 database are unloaded in FORMAT VARIABLE ALL.

The unloaded data is stored in a sequential file for standard columns and in a PDS by using a template and a LOB file reference for LOB columns. All output files, including the LOAD SYSIN, are allocated by using templates.

If some of the table spaces contain more than a single table, the &SEL user-defined variable is used in the DSN expression of the templates that are used in the OUTDDN and LOADDN files to distinguish the result of SELECT statements for tables that are located in the same table space.

For the OUTLOB template, the &TS variable is substituted with the name of the table space that stores the LOB column value, not the base table space name. This table space name is different for each LOB column of each table. Therefore, the DSN expression of template OUTLOB does not need to contain the &SEL variable.

```
LISTDEF LISTLOB INCLUDE TABLESPACE DBTEST1.*

TEMPLATE OUTDSN DSN HLQ.&DB..&TS..&SEL..UNLOAD
TEMPLATE OUTLOB DSN HLQ.&DB..&TS..LOBDATA(&UNIQ.)
TEMPLATE LOADTEMP DSN HLQ.&DB..&TS..&SEL..LOADD

GLOBAL OPTIONS REFORMAT ( TYPE CLOB INTO VARCHAR(54) CLOBF OUTLOB ,
                          TYPE BLOB INTO VARCHAR(54) BLOBF OUTLOB ,
                          TYPE DBCLOB INTO VARCHAR(54) DBCLOBF OUTLOB )

TEMPLATESET( SEL = :SELNUM )

UNLOAD TABLESPACE
SELECT * FROM LIST(LISTLOB) OUTDDN OUTDSN FORMAT VARIABLE ALL LOADDN LOADTEMP
```

Related reference
Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

**Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format**

In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

The following column definitions are for the DSN8810.EMP table:

<table>
<thead>
<tr>
<th>Column</th>
<th>Definition</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPNO</td>
<td>CHAR(6) NOT NULL</td>
<td></td>
</tr>
<tr>
<td>FIRSTNAME</td>
<td>VARCHAR(12) NOT NULL</td>
<td></td>
</tr>
<tr>
<td>MIDINIT</td>
<td>CHAR(1) NOT NULL</td>
<td></td>
</tr>
<tr>
<td>LASTNAME</td>
<td>VARCHAR(15) NOT NULL</td>
<td></td>
</tr>
<tr>
<td>WORKDEPT</td>
<td>CHAR(3)</td>
<td></td>
</tr>
<tr>
<td>PHONENO</td>
<td>CHAR(4)</td>
<td></td>
</tr>
<tr>
<td>HIREDATE</td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td>JOB</td>
<td>CHAR(8)</td>
<td></td>
</tr>
<tr>
<td>EDLEVEL</td>
<td>SMALLINT</td>
<td></td>
</tr>
<tr>
<td>SEX</td>
<td>CHAR(1)</td>
<td></td>
</tr>
<tr>
<td>BIRTHDATE</td>
<td>DATE</td>
<td></td>
</tr>
<tr>
<td>SALARY</td>
<td>DECIMAL(9, 2)</td>
<td></td>
</tr>
<tr>
<td>BONUS</td>
<td>DECIMAL(9, 2)</td>
<td></td>
</tr>
<tr>
<td>COMM</td>
<td>DECIMAL(9, 2)</td>
<td></td>
</tr>
</tbody>
</table>

When you specify the INTO clause, the SYSIN performs the following conversions:

- The PHONENO column is converted from CHAR(4) to SMALLINT.
- The HIREDATE column is converted from DATE to DATE EXTERNAL on 15 characters.
- The EDLEVEL column is converted from SMALLINT to SMALLINT EXTERNAL on the default length of six characters.
- The SALARY, BONUS, and COMM columns are converted from DEC(9,2) to DECIMAL ZONED on nine, seven, and seven characters, respectively.
The LOAD SYSIN that is generated in the SYSPUNCH file contains a reload statement into the IBMUSER.EMP_BIS table with modified column names for the EMPNO and PHONENO columns.

```
UNLOAD TABLESPACE DB2 NO QUIESCE YES
   SELECT *
   INTO TABLE IBMUSER.EMP_BIS
       EMPLOYEE_NUMBER
       ,
       ,
       ,
       ,PHONE_NUMBER SMALLINT
       ,HIREDATE_DATE EXTERNAL(15)
       ,EDLEVEL SMALLINT EXTERNAL
       ,
       ,SALARY DEC ZONED(9,2)
       ,BONUS DEC ZONED(7,2)
       ,COMM DEC ZONED(7,2)
       FROM DSN8810.EMP
OUTDDN OUT
FORMAT DSNTIAL LOADDDN SYSPUNCH
```

The following example shows the SYSPUNCH file:

```
LOAD DATA
LOG NO  REPLACE
EBCDIC CCSID(0037)
INTO TABLE IBMUSER.EMP_BIS
 (EMPLOYEE_NUMBER (1) CHAR (6)
  ,FIRSTNME (7) VARCHAR
  ,MIDINIT (21) CHAR (1)
  ,LASTNAME (22) VARCHAR
  ,WORKDEPT (39) CHAR (3)
  NULLIF(42) = '?'
  ,PHONE_NUMBER (43) SMALLINT
  NULLIF(45) = '?'
  ,HIREDATE DEC (46) DATE EXTERNAL(15)
  NULLIF(61) = '?'
  ,JOB (62) CHAR (8)
  NULLIF(70) = '?'
  ,EDLEVEL (71) INTEGER EXTERNAL (6)
  NULLIF(77) = '?'
  ,SEX (78) CHAR (1)
  NULLIF(79) = '?'
  ,BIRTHDATE (80) DATE EXTERNAL (10)
  NULLIF(90) = '?'
  ,SALARY (91 : 99) DECIMAL ZONED
  NULLIF(100) = '?'
  ,BONUS (101 : 107 ) DECIMAL ZONED
```
Related reference

Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

**Example: Using an HFS file when unloading LOB data**
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

In this example, an HFS file for the LOB column is used when you unload the TBNAME01 table. The LOB file references are created in the existing /u/test/ directory with a file name that contains the record number, which is created because the :RECNUM keyword is specified.

```sql
TEMPLATE OUTTMPLT
DSN '/u/test/NUM.'
DSNTYPE HFS
UNLOAD TABLESPACE
DB2 NO
QUIESCE YES
OPTIONS
TEMPLATESET (NUM=:RECNUM)
SELECT COL_LOB
INTO COLOB VARCHAR(44) CLOBF OUTTMPLT
```
Related reference

Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

These examples unload reports about business activity that is based on business time columns with the following definitions. The definitions are excerpts from the CREATE TABLE statement.

- C_BUS_START DATE NOT NULL
- C_BUS_END DATE NOT NULL
- PERIOD BUSINESS_TIME(C_BUS_START, C_BUS_END)

A SELECT statement with a period-specification clause

```sql
UNLOAD TABLESPACE DB2 FORCE
SELECT C_INT_NN, 'BUSINESS STARTED ON', C_BUS_START, 'AND ENDED ON', C_BUS_END
FROM HPUDOC.TBDZAAPTVT201
FOR BUSINESS_TIME AS OF '2010-04-30'
```
OUTDDN(DB2FRC)
FORMAT DELIMITED SEP ' '  

A SELECT statement without a period-specification clause

UNLOAD TABLESPACE
DB2 NO
SELECT C_INT_NN,
    'BUSINESS STARTED ON',
    C_BUS_START,
    'AND ENDED ON',
    C_BUS_END
FROM HPUDOC.TBDZAAPTVT201
OUTDDN(DB2NO)
FORMAT DELIMITED SEP ' '  None

This report produces the following DB2NO file output data:

1 BUSINESS STARTED ON 2000-01-01 AND ENDED ON 2010-01-01
2 BUSINESS STARTED ON 2000-01-02 AND ENDED ON 2010-01-02

Related reference

Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformate some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

In the following example, three columns with various timestamp precision are unloaded. C_TMSTP_n is defined as TIMESTAMP(n). The output format is TIMESTAMP_B.
UNLOAD TABLESPACE
DB2 NO
SELECT   
   C_TMSTP_0, 
   C_TMSTP_6, 
   C_TMSTP_12
FROM HPUDOC.TBDZAAPT98101
OUTDDN(TMSTMPB)
FORMAT DSNTIAUL
OPTIONS REFORMAT (TYPE TIMESTAMP INTO TIMESTAMP_B)

This example produces the following output data:

2005-06-01-07.31.06 2005-06-01-07.31.06.123456 2005-06-01-07.31.06.123456789012

In the following example, three columns with various timestamp precision are unloaded. C_TMSTP_n defined as TIMESTAMP(n). Only the C_TMSTP_12 column is unloaded with the TIMESTAMP_B output format.

UNLOAD TABLESPACE
DB2 NO
SELECT   
   C_TMSTP_0, 
   C_TMSTP_6, 
   C_TMSTP_12
FROM HPUDOC.TBDZAAPT98101
OUTDDN(OUTPUT)
FORMAT DSNTIAUL
OPTIONS REFORMAT (TYPE TIMESTAMP(12) INTO TIMESTAMP_B)

This example produces the following output data:

2005-06-01-07.31.06 2005-06-01-07.31.06.123456 2005-06-01-07.31.06.123456789012

Related reference
Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

Example: Maximizing resources during sort operations
In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

```
UNLOAD TABLESPACE
SELECT ICDATE
FROM SYSIBM.SYSCOPY
WHERE ICDATE > '110420'
ORDER BY DBNAME, TSNAME
OUTDDN(OUTPUT)
MAX_EXPECTED_ROWS 2000
FORMAT DSNTIAUL
LOADDDN SYSPUNCH
```

Related reference
Example: Creating a sequential data set in USER format
In this example, a sequential data set is created in USER format from the PERSONAL table.

Example: Creating a sequential data set in DSNTIAUL format
In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

Example: Creating a sequential data set that is compatible with the Db2 LOAD utility
In this example, a sequential data set that is compatible with the Db2 LOAD utility is created.

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements
In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

Example: Using LISTDEF and TEMPLATE statements
In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference
In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format
In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

Example: Using an HFS file when unloading LOB data
When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

Example: Using temporal tables
These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

Example: Using timestamp precision
These examples show how you can use timestamp precision.

The SYSPRINT data set

The SYSPRINT data set contains detailed information about the unloaded objects.

Important: in order to get the whole content of the SYSPRINT data set, ensure your allocation parameters for SYSPRINT comply with the requirements (see “User-allocated ddnames” on page 88 for details).

The following examples show a SYSPRINT data set that corresponds to a table space with one table, a SYSPRINT data set that corresponds to a partitioned table space with one table, and a report that is generated when output files are written on tapes.

A SYSPRINT data set that corresponds to a table space with one table

The following example shows a SYSPRINT data set that corresponds to the unloaded DBINFDM.TSSCA table space, which contains a single table that is named user.TBSCA. This table contains 5002578 rows. If the return code is 0, no action is required. If the return code is not 0, look in the SYSPRINT for warning and error messages.

```
INZU224I IBM DB2 HIGH PERFORMANCE UNLOAD V4.1
INZI175I PROCESSING SYSIN AS EBCDIC.
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
000001 UNLOAD TABLESPACE DBINFDM.TSSCA
000002 DB2 NO LOCK NO QUIESC YES
000003 SELECT * FROM TBSCA
000004 OUTDDN (SYSREC00 )
000005 FORMAT DSNTIAUL
000006 LOADDDN SYSPUNCH
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DB2UNLOAD
DSNU050I    DSNUGUTC - QUIESCE TABLESPACE DBINFDM.TSSCA
DSNU477I  -DZ7M DSNUQUIA - QUIESCE SUCCESSFUL FOR TABLESPACE DBINFDM.TSSCA
DSNU474I  -DZ7M DSNUQUIB - QUIESCE UTILITY COMPLETE, ELAPSED TIME= 00:00:00
DSNU180I UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE
PARMLIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN.

INZU281I - UNLOAD STARTING AT POS(1, 1)
INZU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBINFDM.TSSCA
INZU279I - SELECT STATEMENTS USING SINGLE TABLE SPECIFICATION
INZU280I - SELECT 00001 STARTING AT POS(3, 1)
INZU282I   OUTPUT DDNAME=SYSREC00
INZX090 SYSREC00 : 5002578 RECORDS WRITTEN
- TABLESPACE     DBINFDM.TSSCA - DB2 HIGH PERFORMANCE UNLOAD - STATISTICS - 10/20/06
  * CREATOR.TABLE   * OBID   * ROWS READ   * ROWS KEPT   * TS Pg READ
  *------------------*-----------*---------------*---------------*------------*
  * IBMUSER.TBSCA   *       345 *       5002578 *       5002578 *            *
  *------------------*-----------*---------------*---------------*------------*
  * TOTAL UNLOAD STATISTICS ....* 5002578  5002578  201103 *
  * INVALID ROWS. 0 *
  * NUMBER OF PAGES IN ERROR....* 0
INZU222I SYSREC00, TOTAL NUMBER OF RECORDS WRITTEN 5002578
INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(3, 1)
```

A SYSPRINT data set that corresponds to a partitioned table space with one table

The following example shows a SYSPRINT data set that corresponds to the DBDM7M.TSTEST16 partitioned table space, which has six partitions and contains a single table that is named IBMUSER.TSTEST16. This table contains 1500000 rows. The ORDER CLUSTER clause unloads the data in the order of the clustering index. A template allocates a separate file per partition.

```
INZU224I IBM DB2 HIGH PERFORMANCE UNLOAD V4.1
INZI175I PROCESSING SYSIN AS EBCDIC.
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
000001 UNLOAD TABLESPACE DBDM7M.TSTEST16
000002 DB2 NO LOCK NO QUIESC YES
000003 SELECT * FROM TSSTEST16
000004 OUTDDN (SYSREC00 )
000005 FORMAT DSNTIAUL
000006 LOADDDN SYSPUNCH
DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DB2UNLOAD
DSNU050I    DSNUGUTC - QUIESCE TABLESPACE DBDM7M.TSTEST16
DSNU477I  -DZ7M DSNUQUIA - QUIESCE SUCCESSFUL FOR TABLESPACE DBDM7M.TSTEST16
DSNU474I  -DZ7M DSNUQUIB - QUIESCE UTILITY COMPLETE, ELAPSED TIME= 00:00:00
DSNU180I UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE
PARMLIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN.

INZU281I - UNLOAD STARTING AT POS(1, 1)
INZU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBDM7M.TSTEST16
INZU279I - SELECT STATEMENTS USING SINGLE TABLE SPECIFICATION
INZU280I - SELECT 00001 STARTING AT POS(3, 1)
INZU282I   OUTPUT DDNAME=SYSREC00
INZX999 SYSREC00 : 5002578 RECORDS WRITTEN
- TABLESPACE     DBDM7M.TSTEST16 - DB2 HIGH PERFORMANCE UNLOAD - ORDER CLUSTER - ORDER INDEX
  * CREATOR.TABLE   * OBID   * ROWS READ   * ROWS KEPT   * TS Pg READ
  *------------------*-----------*---------------*---------------*------------*
  * IBMUSER.TSTEST16 345 5002578 5002578 5002578 *
  *------------------*-----------*---------------*---------------*------------*
  * TOTAL UNLOAD STATISTICS ....* 5002578 5002578 201103 *
  * INVALID ROWS. 0 *
  * NUMBER OF PAGES IN ERROR....* 0
INZU222I SYSREC00, TOTAL NUMBER OF RECORDS WRITTEN 5002578
INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(3, 1)
```
The following information is shown in the data set:

**CREATOR.TABLE**
Indicates the name of the table that has been unloaded. For partitioned table spaces, the OBID is displayed under the table name.

**OBID/PART NO./IXSC**

**OBID**
Indicates the OBID value for each table in a non-partitioned table space.

**PART NO./PART**
Indicates the partition number for a partitioned table space. The OBID of the unloaded table is indicated under the table name in the CREATOR.NAME column. The value on the **TOTAL UNLOAD STATISTICS** line contains the total number of partitions in the table space. The number of partitions in the table space can differ from the number of partitions that are unloaded when PART is used in the UNLOAD command or PARTLEVEL is used in a LISDEF.
IXSC
Indicates the percentage of rows that were located in the correct order based on the clustering index when an index scan is used. This column is displayed only if an ORDER BY corresponds to the order of the clustering index or if an ORDER CLUSTER clause has been coded.

ROWS READ
Indicates the total number of rows that were read. The number on the TOTAL UNLOAD STATISTICS line can be greater than the number of rows that were read for a single table. For example, for a multi-table table space, the total number of rows that were read also includes the rows of the tables that were dropped. For a partitioned table space, each line corresponds to the number of rows that were read for the partition, and the total corresponds to the total number of rows that were read for the table space.

ROW KEPT
Indicates the total number of rows that could be successfully preprocessed before the subsequent phases of the process, i.e. the logical selection phase when a WHERE clause is specified, the user exit (OUTEXIT option) phase when requested, the formatting phase and the writing into the output dataset phase. If this number is less than the corresponding ROWS READ value, refer to the messages reporting errors at the record level and the summary message INZU447E.

TS PG READ
Indicates the number of pages that were read for each partition and the total for all the selected partitions on the TOTAL UNLOAD STATISTICS line for partitioned table spaces. For nonpartitioned table spaces, only the value on the TOTAL UNLOAD STATISTICS line is displayed for the complete table space.

IX ROWS READ
Indicates the number of records that were read from the clustering index for each partition and the total number of records on the TOTAL UNLOAD STATISTICS line. This column is displayed only when the index scan is used.

TOTAL UNLOAD STATISTICS
Shows total number of rows that were processed.

Report when output files are written on tapes
The following example shows a generated report when output files are written on tapes. Information about files that span across several volumes is documented only in the section that is related to the last volume that the files have been written on. This type of report is displayed only if TAPEREPORT YES is specified, or if the value of the VZM11/TAPERPT parameter is YES.

<table>
<thead>
<tr>
<th>- TAPE SUMMARY - STATISTICS - 08/20/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>* VOLUME = TAP040</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RP1X998B</td>
</tr>
<tr>
<td>* FILESEQ = 00000003</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X991A</td>
</tr>
<tr>
<td>* FILESEQ = 00000004</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X995G</td>
</tr>
<tr>
<td>* FILESEQ = 00000005</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X998B</td>
</tr>
<tr>
<td>* FILESEQ = 00000007</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X99LM</td>
</tr>
<tr>
<td>* FILESEQ = 00000009</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>* VOLUME = TAP030</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X998B</td>
</tr>
<tr>
<td>* FILESEQ = 00000017</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X995D</td>
</tr>
<tr>
<td>* FILESEQ = 00000018</td>
</tr>
<tr>
<td>* SIN = MZLFRLD.AEST.DBTLS00.RPV1X998B</td>
</tr>
<tr>
<td>* FILESEQ = 00000019</td>
</tr>
</tbody>
</table>
Chapter 6. Db2 HPU interactive component

The Db2 HPU interactive component can be used to generate the Db2 HPU batch utility job. Although using the interactive application is not required, it can be useful for inexperienced users in generating the batch utility job.

You can use either the Db2 Administration Tool or the Db2 Tools Launchpad to start the Db2 HPU interactive component.

Db2 HPU interactive component primary commands

Primary commands are used with the Db2 HPU interactive component.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COLUMNS</strong></td>
<td>Displays the SELECT Columns panel. For more information, see “Selecting columns for the SELECT statement” on page 259.</td>
</tr>
<tr>
<td><strong>COPYDDN</strong></td>
<td>Displays the COPYDDN File Parameters panel. This panel is used to specify parameters for the image copy file. This command is available only if a table space was selected. For more information, see “Specifying COPYDDN file parameters” on page 244.</td>
</tr>
<tr>
<td><strong>FORMAT</strong></td>
<td>Displays the SELECT Format panel, which is used to specify the output format for the SELECT statement. For more information, see page “Specifying the format of the SELECT statement” on page 255.</td>
</tr>
<tr>
<td><strong>FUSER</strong></td>
<td>Displays the Select User Format panel. For more information, see “Displaying and modifying the USER format of the SELECT statement” on page 262. The FUSER command (Select user format) command is available only if you select option 4 in the FORMAT field on the SELECT Format panel, as shown in Figure 35 on page 255.</td>
</tr>
<tr>
<td><strong>GENERAL</strong></td>
<td>Displays the General Options panel. This panel is used to specify general options for a Db2 HPU member. For more information, see “Specifying general options” on page 240 for more information.</td>
</tr>
<tr>
<td><strong>JCL</strong></td>
<td>Displays the JCL panel, which is used to generate JCL. This JCL can either be modified under the editor screen, saved, or directly submitted. The JCL command is available only if a table space was selected. For more information, see “Generating JCL to run a Db2 HPU job” on page 270.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>LOADDNN</td>
<td>Displays the LOADDNN File panel. For more information, see “Displaying and modifying LOADDNN file parameters” on page 269.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Displays the SELECT Statement Option panels, which are used to specify values for columns of type DATE, TIME, or TIMESTAMP. For more information, see “Specifying options for the SELECT statement” on page 247.</td>
</tr>
<tr>
<td>ORDERBY</td>
<td>Displays the SELECT Order By panel. For more information, see “Sorting columns of a SELECT statement” on page 261.</td>
</tr>
<tr>
<td>OUTDDN</td>
<td>Displays the OUTDDN File panel. For more information, see “Displaying the list of OUTDDN files” on page 266.</td>
</tr>
<tr>
<td>PART</td>
<td>Displays the Partition List panel, which is used to select partitions that are to be unloaded. This command is available only if a partitioned table space was selected. For more information, see “Including and excluding partitions” on page 242.</td>
</tr>
<tr>
<td>SELECT</td>
<td>Displays the SELECT Statement List panel. For more information, see “Updating the list of SELECT statements” on page 249.</td>
</tr>
<tr>
<td>SELECTALL</td>
<td>Selects all columns.</td>
</tr>
<tr>
<td>UNLDDN</td>
<td>Displays the UNLDDN File Parameters panel, which is used to specify parameters for the allocation of the data set that contains the data of unloaded table spaces or partitions. For more information, see “Specifying UNLOAD file parameters” on page 245.</td>
</tr>
<tr>
<td>WHERE</td>
<td>Displays the ISPF edit panel so you can enter an SQL statement.</td>
</tr>
</tbody>
</table>

### Related tasks

- **Specifying the format of the SELECT statement**
  - The **SELECT Format** panels are used to specify the output format for a SELECT statement.

- **Selecting columns for the SELECT statement**
  - The **SELECT Columns** panel is used to select columns for the SELECT statement.

- **Sorting columns of a SELECT statement**
  - The **SELECT ORDER BY** panel is used to specify the order in which each column in the SELECT statement is processed for the output data set and whether to sort the column in ascending or descending order.

- **Displaying and modifying the USER format of the SELECT statement**
  - The **Select user format** panels are used to update the USER formats.

- **Displaying the list of OUTDDN files**
  - The OUTDDN list panel is used to display the list of OUTDDN files.

- **Displaying and modifying LOADDNN file parameters**
  - The LOADDNN file panel is used to display and modify LOADDNN file parameters.

- **Generating JCL to run a Db2 HPU job**
You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a Db2 HPU job by invoking the JCL command.

**Starting the Db2 HPU interactive component in stand-alone mode**

You can start Db2 HPU interactive component in stand-alone mode without using the Db2 Administration Tool or the Db2 Tools Launchpad.

**Procedure**

Use one of the following methods:

- Run the INZHPU procedure that is located in the SINZCLST library. When you run the INZHPU procedure in stand-alone mode, do not pass an argument.
- Issue this command:

  ```
  EXEC 'HINZnnn.SINZCLST(INZHPU)'
  ```

  In this example, HINZnnn.SINZCLST is the name of the library that contains the INZHPU procedure. You can also associate this command with an option in a menu.

  **Restriction:** The TSO user ID that runs this command requires a minimum REGION of 4096K.

- Issue this command:

  ```
  TSO INZHPU
  ```

  **Requirement:** To use this method, you must have installed INZHPU as a TSO command.

**Related tasks**

Optional: Creating a TSO command for the Db2 HPU interactive component

You can make the Db2 HPU interactive component available as a TSO command.

**Starting the Db2 HPU interactive component by using the Db2 Administration Tool**

The Db2 HPU interactive component can be started by using the Db2 Administration Tool.

**Procedure**

Specify the HPU line command on the Db2 Admin panel.

Db2 HPU interactive component can be started from either the Table Spaces panel or from the Tables, Views, and Aliases panel.

If you invoke the HPU command from the Tables, Views, and Aliases panel, the SELECT Format panel is displayed. For more information about the SELECT Format panel, see “Specifying the format of the SELECT statement” on page 255.
Valid line commands are:
- T - Tables
- D - Database
- A - Auth
- G - Storage group
- ICS - Image copy status
- DIS - Display database
- STA - Start database
- STO - Stop database
- X - Indexes
- ? - Show all line commands

Select Name     Owner    DB Name  BP   L E S I C Ntable    N Active Segsz  T L
*        *        *        *    * * * * *      *           *     *  * *
------ -------- -------- -------- ---- - - - - - ------ ----------- ------ - -
SYSCOPY  SYSIBM   DSNDB06  BP0  A N A N N      1           0      0   Y
SYSDBASE SYSIBM   DSNDB06  BP0  A N A N N     14           0      0   Y
HPU  SYSDBAUT SYSIBM   DSNDB06  BP0  A N A N N     2           0      0   Y
        SYSDF  SYSIBM   DSNDB06  BP0  P N A N N     7           0      4   Y
        SYSGAUT SYSIBM   DSNDB06  BP0  A N A N N     1           0      0   Y
        SYSGROUP SYSIBM   DSNDB06  BP0  A N A N N     2           0      0   Y
        SYSOBJ  SYSIBM   DSNDB06  BP0  P N A N N     8           0      4   Y
        SYSPKAGE SYSIBM   DSNDB06  BP0  A N A N N     7           0      4   Y
        SYSPLAN SYSIBM   DSNDB06  BP0  A N A N N     5           0      0   Y
        SYSSTATS SYSIBM   DSNDB06  BP0  A N A N N     6           0      4   Y
        SYSSTR  SYSIBM   DSNDB06  BP0  A N A N N     4           0      4   Y
        SYSUUSR SYSIBM   DSNDB06  BP0  A N A N N     1           0      0   Y
        SYSVIEWS SYSIBM   DSNDB06  BP0  A N A N N     4           0      0   Y

Figure 16. Db2 Administration Table Spaces panel

If you invoke the HPU command from the Table Spaces panel, the General Options panel is displayed. See “Specifying general options” on page 240 for more information about the General Options panel.

Valid line commands are:
- C - Columns
- A - Auth
- L - List
- X - Indexes
- S - Table space
- D - Database
- V - Views
- T - Tables
- P - Plans
- Y - Synonyms
- SEL - Select prototyping
- ? - Show all line commands

Select Name     Owner    T DB Name  TS Name    Cols       Rows Checks
*                  *        *        *             *          *      *
----- ------------------ -------- -------- -------- ------ ---------- -----
SYSCOPY            SYSIBM   T DSNDB06  SYSCOPY      22         -1      0
SYSFIELDS          SYSIBM   T DSNDB06  SYSDBASE     13         -1      0
SYSTABLESPACE      SYSIBM   T DSNDB06  SYSDBASE     37         -1      2
SYSTABLES          SYSIBM   T DSNDB06  SYSDBASE     43         -1      4
SYSTABLEPART       SYSIBM   T DSNDB06  SYSDBASE     33         -1      0
SYSTABAUTH         SYSIBM   T DSNDB06  SYSDBASE     29         -1      0
SYSSYNONYMS        SYSIBM   T DSNDB06  SYSDBASE     7          -1      0
SYSSRELS           SYSIBM   T DSNDB06  SYSDBASE     13         -1      0
SYSSLINKS          SYSIBM   T DSNDB06  SYSDBASE     12         -1      0
SYSEKEYS           SYSIBM   T DSNDB06  SYSDBASE     7          -1      0
SYSSINDEXPART      SYSIBM   T DSNDB06  SYSDBASE     26         -1      0
SYSSINDEXES        SYSIBM   T DSNDB06  SYSDBASE     38         -1      7
SYSSFOREIGNKEYS    SYSIBM   T DSNDB06  SYSDBASE     7          -1     à

Figure 17. Db2 Admin Tables, Views and Aliases panel

Related tasks
Specifying the format of the SELECT statement
The SELECT Format panels are used to specify the output format for a SELECT statement.
Specifying general options
You can use the **General Options** panel to specify Db2 HPU general options.

## Starting Db2 HPU interactive component by using the Db2 Tools Launchpad

The Db2 HPU interactive component can be started by using the Db2 Tools Launchpad.

### Procedure

1. Specify the `S` line command on the **Db2 Tools Launchpad** panel, as shown in the following example. Press Enter.

```
------------------------------- DB2 Tools Launchpad ------------------------------- Row 1 of 8
Select the DB2 tool you wish to launch or enter its code in the command line.
Sel Code  Tool Name                                        Rel  Prog No.
---   ------------ ADMINISTRATION TOOLS -------------  ---  --------
ADM   DB2 Administration Tool                          710  5697-L90
---   --------- APPLICATION MANAGEMENT TOOLS --------  ---  --------
s   HPU   High PerformanceUnload                          410  5655-AA1
---   --------- PERFORMANCE MANAGEMENT TOOLS --------  ---  --------
---   -- RECOVERY AND REPLICATION MANAGEMENT TOOLS --  ---  --------
No table entries in this category
No table entries in this category
******************************* Bottom of data ********************************
```

*Figure 18. Db2 Tools Launchpad panel*

The **DB2 System** panel is displayed, as shown in the following figure:

```
INZ0005 ------------------------  DB2 System -------------------------- 11:44
Command ===>

Enter :
*DB2 system name . . ______ (mandatory field)

( PF3 to exit )
```

*Figure 19. Db2 subsystem name panel*

2. Specify a Db2 subsystem name, and press Enter.

The **Database And Tablespace Selection** panel is displayed, as shown in the following figure:
Figure 20. Database And Tablespace Selection panel

3. Use one of the following methods to select table spaces:

   • Issue the SELECT primary command to select one or more tables.
   • Issue the Select line command to select one table space. When you use the Select line command, the General options panel is displayed, which is used to specify parameters for the table space that you selected.
   • Specify the name of a database in the Database ==> field and the name a table space in the Tablespace ==> field.

Results

If you specify non-generic criteria and you do not include a percent sign (%) when table spaces are listed by generic names, a percent sign is automatically added to the end of the field.

Related tasks

Selecting tables

The Table List panel is used to create SELECT statements.

Specifying general options

You can use the General Options panel to specify Db2 HPU general options.

Procedure

1. Specify the HPU command on the Db2 Admin panel or after you specify a Db2 subsystem and table space.

   The General Options panel is displayed, as shown in the following figure:
2. Specify values for each of the following fields:

**Utility ID**
Specify the prefix that is used to build the unique Utility Identifier for the Db2 HPU job. The maximum length of the prefix is 11 bytes. This parameter is required.

**Part**
Specify information about the partitions of a table space.

- **L**
  Indicates that several partitions were selected by using the PART command. The L option is not used to display the list of partitions.

- *****
  Selects all partitions of a table space.

- **nnn**
  Indicates the number of a single partition of the table space.

**DB2**
Specify whether a given request can or must be processed through Db2.

- **No**
  The SELECT statements must be processed directly by Db2 HPU. If a SELECT statement is not supported by HPU, an error is generated and the program stops.

  **Attention:** The entire SYSIN file is parsed before any unload command is processed.

- **Yes**
  If the SELECT statement is too complex to be handled directly by the product, Db2 is called to extract the rows.

- **Force**
  DB2 must be used to extract the requested rows. This process is useful when the SELECT statement uses a filtering predicate that is efficiently processed through SQL and when the filtering factor is high.

  **Restriction:** DB2 FORCE cannot be used when attempting to unload from an image copy. Attempting to use DB2 FORCE to unload from an image copy results in an error and the program stops.

**LOCK**
Specify the lock mode of the table space.

- **Yes**
  The table space is accessed in read only mode when the Db2 HPU job is running.

- **No**
  The table space is processed without changing its access mode.
QUIESCE
Specify the processing of the QUIESCE command.

Yes
A QUIESCE command is processed if the table space is not in the COPY pending status; otherwise, processing of the QUIESCE command is stopped and restarted.

No
The table space is processed without a QUIESCE command.

Attention: Db2 HPU operates on the physical VSAM data set level that is outside of Db2. If you run Db2 HPU on a table where a row was just inserted, it is possible that the unloaded data might not contain the row that was inserted. The unloaded data does not show the row because Db2 might not have externalized the data to DASD yet. This situation can happen when you use Db2 HPU without issuing a QUIESCE (or STOP) on the object. Be careful when using QUIESCE NO.

UNLMAXROWS
Specify the maximum number of rows to be extracted for the physical unload. If the process involves a partitioned table space treated partition-by-partition, the limit applies to each partition.

UNLFREQROWS
Specify the sampling frequency for the physical unload. One row out of every \( n \) rows is. The value \( n \) is the argument of UNLFREQROWS.

PF3
Press PF3 to end the current unload JCL setup process. The following figure shows the Confirm to leave the current Unload panel. Specify Y to cancel the unload JCL setup process or N to continue.

What to do next
Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

Including and excluding partitions
The Partition List panel is used to include or exclude one or more partitions for a Db2 HPU job.

Procedure
1. Specify the PART command on the General Options panel and press Enter.
   The Partition List panel is displayed, as shown in the following figure:
The **Partition List** panel contains the following columns.

**S**
Use this column to specify the partitions to be selected or deselected. Enter a selection code in this column.

**Part**
This column contains the available partitions.
- To select a partition, specify the S selection code beside a partition number.
- To deselect a partition, specify the U selection code beside a partition number.
- To select all partitions, specify the S selection code beside **ALL** in the Part column.

**Requirement:** You must select at least one partition.

**Selected**
YES indicates that the partition was selected.

**High value of partition**
Displays the first 57 characters of the LIMITKEY column of the SYSIBM.SYSTABLEPART table.

2. Specify the appropriate line command.

You can specify the following line commands on the **Partition List** panel:

**S**
Select one or more partitions.

**U**
Undo the selection of one or more partitions.

**Z**
Displays the **Zoom Partition** panel, as shown in the following figure. Select this option when the value of the High value of partition column is too high to display in the field. The **Zoom Partition** panel shows the partition number, indicates if it was selected, and displays the entire value for the partition.
Press PF3 to return to the Partition List panel.

What to do next
Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

Specifying COPYDDN file parameters

The COPYDDN File Parameters panel is used to specify file parameters for the image copy file.

Procedure
1. Specify the COPYDDN command on the General Options panel.
   The COPYDDN File Parameters panel is displayed, as shown in the following figure:

   ![Figure 25. COPYDDN File Parameters panel](image)

   2. Specify values for each of the following fields:

   **COPYDDN**
   Displays the COPYDDN File Parameters panel, which is used to specify parameters for the image copy file. This command is available only if a table space was selected.

   **No**
   No image copy is used as source.

   **nn**
   The number of the last (most recent) full image copy to be used as source.

   **Last ic**
   Process the unload job by using the last valid registered full image copy on the object. This option is equivalent to a value of -1 for nn.

   **DDN**
   Process the unload by using the information that is provided on this panel.

   **IC**
   Specify the image copy type.

   **Non-inline**
   Indicates that the copy is not inline.
**Inline**
If the ddname points to an Inline Copy (that is a copy share level change), you must specify Inline. Failure to do so can result in duplicate or missing lines in the output, or failure of the job during the unload process itself.

**Check**
Check must be specified if the copy is Inline or is not known.

**Data Set Name**
Specify the name of the image copy data set. The following variables can be used, during the generation of the JCL, a substitution occurs.

- &DBNAME: this variable is replaced by the database name that contains the unloaded table space.
- &TSNAME: this variable is replaced by the name of the unloaded table space.
- &SPACE: this variable is replaced by the name of the unloaded table space. &SPACE is a synonym of &TSNAME.
- &ZUSER: this variable is replaced by the user name used to connect, where the interactive menu is running.
- &DB2: this variable is replaced by the SSID where the generated JCL will run.
- &PART: this variable is replaced by the partition number:
  - 000: when the unloaded table space is not partitioned or when ALL is specified as partition number on the panel you select the partition
  - nnnn: 4 digits for the partition number

For partitioned table spaces, to define one input file for each partition, use the &PART variable in the generic data set name, which allows Db2 HPU to process in parallel.

If Db2 HPU was requested to perform processing in parallel for partitioned table spaces, you must specify this processing for all files.

**Disp**
The disposition of the image copy data set. Enter SHR if the image copy data set is to be shared.

**Generic unit**
Specify the DASD unit name.

**Volume serial**
Specify the volume serial number. Use an asterisk (*) when the image copy data set is not cataloged.

**for tape unit/Label**
Specify the label sequence number when the data set resides on tape.

What to do next
Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

**Related reference**
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

---

**Specifying UNLOAD file parameters**
The **UNLDDN File Parameters** panel is used to specify parameters for allocating the data set that contains the data of the unloaded table spaces or partitions.

**Procedure**
1. Specify the **UNLDDN** command on the **General Options** panel.
The **UNLDDN File Parameters** panel is displayed, as shown in the following figure:

![Panel](image)

2. Specify values for each of the following fields:

**Data Set Name**

Specify the name of the data set that will contain the data of an unloaded table space or partition.

For partitioned table spaces, to define one input file for each partition, use the &PART variable in the generic data set name, which allows Db2 HPU to process in parallel.

If Db2 HPU was requested to perform a processing in parallel for partitioned table spaces, you must specify this processing for all files.

**Disposition**

Specify the disposition of the data set.

- **New/Old/Mod**
  Specify the disposition to use as the first parameter.

- **Del/Keep/Catlg**
  Specify the disposition to use as the second and third parameters.

**Generic unit**

Specify a generic unit if DISP=NEW or if the file is not cataloged.

**Volume serial**

Specify one to five volumes.

**Space Units**

Specify the volume allocation unit (BLKS, TRKS, CYLS).

**Primary quantity**

Specify the primary allocation for the data set.

**Secondary quantity**

Specify the secondary allocation for the data set.

**Record format**

Specify the format of the records as F, FB, V, or VB.

**Record length**

Specify the length of the records. The value that you enter must be less than or equal to 32760.
**Block size**
Specify the block size if FB or VB is specified for FORMAT. The BLKSIZE value must be less than or equal to 32760 and must be a multiple of LRECL (+4 if V or VB).

**SMS**
Specify the Data Class, Management Class, and Storage Class SMS parameters for allocation of data sets that are managed by SMS. The values that are entered for these parameters will be specified in the DD statement that is generated in the JCL.

**for tape unit Label**
Specify the tape sequence number if the file will be written to tape.

**What to do next**
Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

**Related reference**
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

---

**Specifying options for the SELECT statement**

The **SELECT Statement Option** panels are used to specify options for a SELECT statement.

**Procedure**

1. Specify the OPTIONS command on the **General Options** panel.
   The **SELECT Statement Option** panel (1 of 2) is displayed, as shown in the following figure:

   ![Figure 27. SELECT Statement Option panel (1 of 2)](image)

   **INZA501 ---- DZ7M DBDM7M   . TSTEST16 - SELECT Statement Option ------ 11:49**
   **Command ===>**
   **Commands : GENERAL   PART   COPYDDN   SELECT   UNLDDN   JCL**
   **Others options : LOADOPT**

   - **NULL** . . . . . . . _ (0 - Off)
   - or when NULL . . . . . . . __ (enter one character or an hexadecimal value)*
   - and when not NULL . . __ (enter one character or an hexadecimal value)*

   - **NULLID** . . . . . . . _ (Y - Yes, N - No)
   - **NULLPOS** . . . . . . . _ (A - After, B - Before)
   - **PIC sign** . . . . . . _ (sign or P)
   - **position** . . . . _ (L - Lead, T - Trail)
   - **decimal** . . . . _ (, or .)
   - **mask** . . . . ______ (enter ? for the list of authorized values)

   *(case sensitive)

   **(PF3 to exit, Enter : "Option" 2/2)**

   **Figure 27. SELECT Statement Option panel (1 of 2)**

2. Specify values for each of the following fields:

   **Other options: LoadOPT**
   Specify LoadOPT to display an ISPF edit panel that you use to enter load options.

   **NULL**
   Specify whether the null character is to be generated in the output file.

   **or when NULL**
   Specify the character to be used for columns with NULL value.

   **not NULL**
   Specify the character to be used for columns that have a value other than NULL.
**NULLID**
Specify whether the output file will contain NULL indicators. This option is considered only for FORMAT USER.

**NULLPOS**
Specify the position of the NULL indicator. It can be specified before or after the data field. This option is considered only for the USER format or if specified at the select level.

**PIC sign**
Specify the numeric data display format. The default value is a minus sign (-).

+ The plus sign (+) is used for positive values, and the minus sign (-) is used for negative values.

- The minus sign (-) is used for negative values; positive values are preceded by blanks.

P The padding character is used for positive values; negative values are identified by a minus sign (-).

**PIC position**
Specify the position of the sign. The default is Lead.

**Lead**
The sign is placed before the numeric value.

**Trail**
The sign is placed after the numeric value.

**PIC decimal**
Specify the decimal separator. The default value is a period (.).

. A period is written as the decimal separator.

, A comma is written as the decimal separator.

**PIC mask**
This optional parameter is used to define the mask. Enter a question mark (?) to display the list of authorized types. See “Selecting mask formats” on page 275 for more information.

3. Press Enter.
The second SELECT Statement Option panel is displayed, as shown in the following figure:

```
INZA502 ---- DZ7M DBDM7M   . TSTEST16 - SELECT Statement Option ---- 11:49
Command ===> Commands        :  GENERAL    PART    COPYDDN    SELECT    UNLDDN    JCL
Others options :  LOADOPT

DATE . . . . _____________
(enter ? for the list of authorized values)
DATEDELIM . . __
(enter one character or an hexadecimal value)*

TIME . . . . _____________
(enter ? for the list of authorized values)
TIMEDELIM . . __
(enter one character or an hexadecimal value)*

TIMESTAMP . . _____________
(enter ? for the list of authorized values)
LENGTHBYTE . . _
(Y - Yes, N - No)
LENGTH . . . _
(R - Real, M - Max)
*(case sensitive)

(PF3 to return to "Option" 1/2)
```

Figure 28. SELECT Statement Option panel (2 of 2)

4. Specify the values that you want for each of the following fields:
DATE
Specify the available type to encode dates in FORMAT USER. Specify a question mark (?) to display the list of authorized types.

DATEDELIM
Specify the default delimiter that is to be used in external date representation.

TIME
Specify the type to encode times in FORMAT USER. Specify a question mark (?) to display the list of authorized types.

TIMEDELIM
Specify the default delimiter that is to be used in external time representations.

TIMESTAMP
Specify the type to encode timestamps in FORMAT USER. Specify a question mark (?) to display the list of authorized types.

LENGTHBYTE
Specify whether you want the two length bytes for variable-length columns in an options block written to the output data set.

LENGTH
Specify whether the real or maximum length is used for variable-length fields in an options block.
  
  Real
  The length of the field does not change (value of the two length bytes). The default value is Real.
  
  Max
  The output field is padded to its maximum length with binary zeros.

The LENGTH keyword is valid only for variable-length fields.

What to do next
Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related tasks
Selecting date formats
The Type Values - DATE panel is used to select a date format.

Selecting time formats
The Type Values - TIME panel is used to select a time format.

Selecting timestamp formats
The Type Values - TIMESTAMP panel is used to select a timestamp format.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

---

**Updating the list of SELECT statements**

The SELECT statement list panel is used to update the SELECT statements of a Db2 HPU job.

**Procedure**

1. Specify the SELECT command.

   The SELECT Statement List panel is displayed, as shown in the following figure.
New Select statement?  
Specify Y and press Enter to create a SELECT statement.  
If the table space contains several tables, the Table List panel, as shown in Figure 32 on page 253, is displayed. If the table space contains only one table, the SELECT Format panel, as shown in the following figure, is displayed.

Interactive Select

Y

Specify Y to display the SELECT Format panel, as shown in the following figure. You can access all the primary commands from this panel.

N

If you specify N, the first limited SELECT Format panel is displayed. The commands on this panel are limited; only FORMAT, OUTDDN, LOADDDN, and JCL are available. You can use the EDIT command to access ISPF edit mode where you can enter the entire SELECT statement and cut and paste a SYSIN.

Press Enter to access the second limited SELECT Format panel, as shown in the following figure.
INZB102E --- DZ7M                               - SELECT Format ------- 14:52
Command ==> 

Commands : EDIT OUTDDN LOADDN JCL

ORIGINOBID . . . . .      (HexaDecimal value)
or . . . . .      (Decimal value)
OUTMAXROWS . . . .
OUTFREQROWS . . . .
SCHEME . . . . . . .      (E - Ebdic, A - Ascii, S - aSis, U - Unicode)
CCSID SBCS . . . . .
MIXED  . . . .
DBCS . . . .
OUTEXIT exitname . . . . . . . . . . . . . . . . . . . . . . (PF3 to return to "Format" 1/2)

Figure 31. SELECT Format panel (2 of 2)

The default value is Y.

Selection codes
You can select only one line. Specify one of the following codes:

S
Use the S selection code to select a statement. The SELECT description panel is displayed for further processing.

D
Use the D selection code to delete a SELECT statement. A confirmation panel is displayed.

C
Use the C selection code to copy a statement to a new statement that has the same characteristics and contents as the original statement.

The options you select determine the subsequent panels that are displayed. The following columns are on the SELECT Statement List panel:

S
Use this column to enter a selection code.

Select description
This column contains the description of the SELECT statement.

Inter. (Interactive)

Y
The SELECT statement is created by done using interactive panels.

N
The SELECT statement is written in freestyle.

Creator
This column contains the name of the creator of the SELECT statement.

Table or view
This column indicates the name of the table or view.

2. Press PF3 to end the current unload JCL set-up.

What to do next
Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related tasks
Selecting tables
The **Table List** panel is used to create SELECT statements.

**Related reference**
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

## Selecting tables

The **Table List** panel is used to create SELECT statements.

**Before you begin**
If you Specify non-generic criteria and you do not include a percent sign (%) when listing tables by generic names, a percent sign is automatically added to the end of the field.

**About this task**
The **Table List** panel is displayed when you have selected a table space and you Specify the **SELECT** command on the **Database and Tablespace Selection** panel or when you use the **SELECT Statement List** panel to create a new SELECT statement and there are several tables in the table space that you selected.

To select a table:

**Procedure**
Use the S selection code to select one or more tables, and press PF3 to access the **SELECT Statement List** panel so you can create SELECT statements.

a) If you select only one table, the **Table List** panel is displayed, as shown in the following figure:
Figure 32. Table List panel

The **Table List** panel contains the following columns:

**S**
- The selection code; enter one or more selection codes in this column.

**Creator**
- This column contains the name of the creator of the table.

**Table or view**
- This column contains the name of the table or of the view.
Database
This column contains the name of the database.

Table space
This column contains the name of the table space.

Type
This column contains the type (table or view).

b) If several tables are selected, the SELECT Statement List panel is displayed.

```
INZA601 ----- DZ7M DBDM7M   . TSTEST16 - SELECT Statement List Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Commands : GENERAL    PART    COPYDDN    OPTIONS    UNLDDN    JCL
Line commands: E - Edit, D - Delete, C - Copy

New Select statement ? . . . . (Y - Yes)
Use interactives panels . . Y (Y - Yes, N - No, default is Yes)
or
Use a line command on this table of select statements already created.
( PF3 to exit)

Select Select description Inter. Creator  Table or view
------ ----------------------------- ------ -----–—— ----------------->
selc 01                       Y    IBMUSER  TSTEST16
IBMUSER .TSTEST16               Y    IBMUSER  TSTEST16
IBMUSER .ZZZZZZZZz             Y    IBMUSER  TSTEST16
*****************************************************************************
```

Figure 33. SELECT Statement List panel

What to do next
Press PF3 to exit this panel.

Related tasks
Selecting time formats
The Type Values - TIME panel is used to select a time format.

Updating the list of SELECT statements
The SELECT statement list panel is used to update the SELECT statements of a Db2 HPU job.

Starting Db2 HPU interactive component by using the Db2 Tools Launchpad
The Db2 HPU interactive component can be started by using the Db2 Tools Launchpad.

Processing long names
Db2 HPU accepts long names for tables and columns. You can view and update the names by using the Complete table name panel.

Procedure
1. Position the cursor on the corresponding field, and press PF5. PF5 is available only when a table name is either displayed or can be entered on the panel; therefore, this option is not available on every panel.

The Complete table name panel is displayed, as shown in the following figure:
Specifying the format of the SELECT statement

The **SELECT Format** panels are used to specify the output format for a SELECT statement.

**Procedure**

1. Specify the **FORMAT** command to display the first **SELECT Format** panel.

2. Specify the values for the following fields in the first **SELECT Format** panel:

   **Other Option : PART**

   Use the PART option to access the **Partition Selection** panels to select a partition for the current SELECT statement. This SELECT format (list of partitions) panel is displayed when you specify PART in the **Other Option** field.

   Use this **SELECT format** panel to select one or more of the displayed partitions.
The selection codes for the **Partition List** panel are:

- **S**
  Specify S next to any partition you want to select.

- **U**
  Specify U next to any partition to undo its selection.

- **Z**
  Specify Z next to any partition to zoom in on a partition and to display on a single panel all information concerning that partition. An example is shown in Figure 24 on page 243.

**Requirement:** You must select at least one partition.

The columns on the **SELECT format (partition list)** panel are:

- **S**
  Use this column to specify the partitions to be selected or deselected. Enter a selection code in this column.

- **Part**
  This column indicates the available partitions. Only the partitions that were selected on the Partition list panel, Figure 23 on page 243, are displayed in this panel.

  To select all partitions, specify the S selection code beside **ALL** in the Part column.

  To select a partition, specify the S beside a partition number.

  To deselect a partition, specify the U selection code beside a partition number.

  **Requirement:** You must select at least one partition.

- **Selected**
  If YES is specified, the partition was selected. This field is displayed after you select a partition and press Enter.

- **High value of partition**
  Shows the first 57 characters of the LIMITKEY column of the SYSIBM.SYSTABLEPART table.

  Use the PF3 key to return to the first **SELECT format** panel, as shown in Figure 35 on page 255.

- **SELECT Description**
  Type a short description of the SELECT statement. The description is displayed in the Select description column of the **SELECT Statement List** panel, as shown in Figure 29 on page 250.

- **PART**
  Specify the following information about the partitions of a table space:
Indicates that several partitions have been selected by using the PART command. Selection code L is not used to display the list of partitions.

* 
Selects all partitions of a table space.

nnn 
Indicates the number of a single partition of the table space.

FORMAT 
Specify the format of the output file:

1 : DSNTIAUL 
Specifies that the file that is produced must be in the format required by DSNTIAUL.

2 : DELIMITED 
Specifies that the output file can be processed by your personal computer.

3 : VARIABLE 
Specifies that the output file must be compatible with the Db2 LOAD utility.

4 : USER 
Specifies that there is output in USER format.

DSNTIAUL STRICT 
Yes 
Specify Yes to obtain constant chains (if any are present in the SELECT statements) in variable format (VARCHAR). You can use only DSNTIAUL STRICT when you select option 1, FORMAT DSNTIAUL.

No 
Specify No if you do not want to obtain constant chains in variable format.

DELIMITED 
Specify that the format of the output data set is a comma-separated value file which corresponds to a .csv file that can be processed by your personal computer. You can specify only DELIMITED when you select option 2, FORMAT DELIMITED.

SEP 
Use this field to specify the character to be used to separate fields in the output data set.

DELIM 
Enter the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set.

NULL DELIM 
Yes 
Specifies that the delimiter character will be used at the beginning and the end of a field containing nulls.

No 
Specifies that the delimiter character will not be used at the beginning and the end of a field that contains nulls.

VARIABLE 
You can specify only VARIABLE when you select option 3, FORMAT VARIABLE. If you select option 3, you must enter a value in this field.

END 
Specify that the characteristics and the sequence of fields of the generated file are the same as the selected columns.

ALL 
Specify that the variable columns are to written according to their lengths.
LIKE creator
Specify the user ID of the creator. You can specify only LIKE creator when you select option 1, FORMAT DSNTIAUL or option 3, FORMAT VARIABLE.

LIKE table
Specify either the short or long table name that indicates the format to be used when creating the output file. You can specify only LIKE table when you select option 1, FORMAT DSNTIAUL or option 3, FORMAT VARIABLE.

3. Press Enter to display the second SELECT format panel or press PF3 to return to the Select List.
The second SELECT format panel is displayed, as shown in the following figure:

INZB102I --- DZ7M IBMUSER . TSTEST16           - SELECT Format ------- 11:59
Command ===>

Commands : COLUMNS WHERE ORDERBY OUTDDN FUSER LOADDN JCL
Other Option : PART
ORIGINOBID . . . . ___ (HexaDecimal value)
or . . . . ___ (Decimal value)
OUTMAXROWS . . . . __________
OUTFREQROWS . . . . __________
SCHEME . . . . . . (E - EBCDIC, A - ASCII, S - ASIS, U - Unicode)
CCSID SBCS . . . . _____
MIXED . . . . _____
DBCS . . . . _____
OUTEXIT exitname . . ________
in . . . . _ (1 - ASM, 2 - C, 3 - COBOL2)
( PF3 to return to "Format" 1/2)

Figure 37. SELECT format panel (2 of 2)

4. Specify the following values for the second SELECT format panel:

ORIGINOBID
Specify a value if the OBID table in the image copy is not equal to the OBID read in the catalog. A discrepancy can occur, for example, for an image copy of a table that was dropped and re-created with a new OBID.

OBID is the hexadecimal value of the OBID of the table in the image copy.
Use the appropriate line to enter a hexadecimal or a decimal value for ORIGINOBID.

OUTMAXROWS
Specify the maximum number of rows to select.

OUTFREQROWS
Specify the unload sampling frequency. One row out of every n rows is written into the OUTDDN data set.

SCHEME
This parameter is optional; use it to specify the format in which the data is unloaded.

E
EBCDIC
A
ASCII
S
ASIS
U
Unicode
CCSID
Specify as many as three valid optional code character set identifiers for the unloaded data: SBCS, MIXED, and DBCS values. If any of the fields are omitted or are specified as 0, the corresponding CCSID is assumed to be the same as the installation default CCSID.

You can load the FIELDPROC and EDITPROC directly from the DSNEXIT that is specified in the PARMLIB without allocating it in STEPLIB.

Attention: If the TRANSLAT parameter in the PARMLIB is set to inhibit data conversion, you can specify a conversion parameter (ASCII, EBCDIC, ASIS, or CCSID) in the SYSIN to override this inhibition.

If the unload format that is specified either in the SYSIN or in the PARMLIB by using the UNLSCHEM parameter is not identical to the system’s EBCDIC format, all constants that are specified in SYSIN will be translated to the unload format.

OUTEXIT exitname in exit_language
Use this field to specify the name and the language of the exit that handles the rows during the unload processing.

ASM
Assembler language (default)

C
C language

COBOL2
COBOL/2 language

5. Press Enter to accept the values, or press PF3 key to return to the first SELECT format panel.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

Selecting columns for the SELECT statement

The SELECT Columns panel is used to select columns for the SELECT statement.

Procedure

Specify the COLUMNS command.
The SELECT Columns panel is displayed, as shown in the following figure:
The SELECT columns panel contains the following fields. You can edit the SELECT Description, Sel, and Pos in Select fields.

**SELECT Description**
Type a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel, as shown in Figure 29 on page 250.

**Sel**
Specify one of the following options:

- **E** Edits the line. When you select **E**, the Edit Expression panel is displayed, as shown in the following figure. Use the Edit Expression panel to display the literal or the expression of a SELECT statement.

---

**Figure 38. SELECT Columns panel**

---

**Figure 39. Edit Expression panel**

Use the PF3 key to return to the SELECT columns panel.

- **D** Deletes the line.

**Pos in Select**
Specify the position of the column in the SELECT statement.

**Name/Value/Expr.**
Indicates the name of the column, the beginning of the literal value, or the expression.

**Trunc**
Indicates that the literal or column name is truncated on this panel.
L/C/E
Indicates that the line is related to a literal (L), to a column (C), or to an expression (E).

Type
Indicates the type of the column or the literal.

Length
Indicates the length of the column or the literal.

Scale
Indicates the decimal part of the column or the literal.

Pos in table
Indicates the position of the column in the table. Literals are indicated by a 0 in this column.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

Sorting columns of a SELECT statement

The SELECT ORDER BY panel is used to specify the order in which each column in the SELECT statement is processed for the output data set and whether to sort the column in ascending or descending order.

Procedure
Specify the ORDER BY command.
The SELECT Order By panel is displayed, as shown in the following figure:

```
n
```

Command ===>                                                  Scroll ===> CSR
Commands :  FORMAT    COLUMNS    WHERE    OUTDDN    FUSER    LOADDN    JCL
* Line commands: D - Delete column from the Order-By clause.
SELECT Description . . IBMUSER .TSTEST16

( PF3 to return to "Select List")

<table>
<thead>
<tr>
<th>Select Name</th>
<th>Columns description</th>
<th>Type</th>
<th>Length</th>
<th>Scale</th>
<th>Pos in order</th>
<th>(A/D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>_</td>
<td>COL4</td>
<td>VARCHAR</td>
<td>50</td>
<td></td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>_</td>
<td>COL2</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>_</td>
<td>COL1</td>
<td>INTEGER</td>
<td></td>
<td>1</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>_</td>
<td>COL3</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td>5</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 40. SELECT Order By panel

The SELECT ORDER BY panel contains the following fields. You can edit the SELECT description, Select, Pos in order, and (A/D) fields.

SELECT Description
Type a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel, as shown in Figure 29 on page 250.

Select
Specify D in this column to delete the column.

Name
Indicates the name of the column.

Type
Indicates the type of the column.
Length
Indicates the total length of the column. This field is only applicable for the CHAR, VARCHAR, LONGVAR, DECIMAL, FLOAT, GRAPHIC, VARG, and LONGVARG column types.

Scale
Indicates the length of the decimal part of the column. This field is authorized for columns of type DECIMAL.

Pos in Select
Indicates the position of the column in the SELECT statement.

Pos in Order
Specify the position of the column in creating the output data set.

A/D
Specify either A to sort in ascending order or D to sort in descending order.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

Displaying and modifying the USER format of the SELECT statement
The Select user format panels are used to update the USER formats.

Procedure
1. Specify the FUSER command.

The first Select User Format panel is displayed, as shown in the following figure:

```
INZB501 -- DZ7M IBMUSER . TTEST16 - Select User Forma Row 1 from 5
Command ==>                                                  Scroll ===> CSR
Commands :  FORMAT    COLUMNS    WHERE    ORDERBY    OUTDDN    LOADDDN    JCL
*         *                   *
Line commands: Z - Zoom
SELECT Description . . IBMUSER .TTEST16
                        ( PF3 to return to "Select List")

<-------- Columns description ---------> <------- Field description ------->

<table>
<thead>
<tr>
<th>S Name</th>
<th>Type</th>
<th>Length</th>
<th>Scale</th>
<th>Type</th>
<th>Length</th>
<th>Scale</th>
<th>Pad</th>
<th>Just</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL1</td>
<td>INTEGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL2</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AAAAAAA</td>
<td>CHAR</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL4</td>
<td>VARCHAR</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COL3</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 41. Select User Format panel (1 of 4)

The following fields are available on all four Select User Format panels. You can edit the SELECT description and the S fields.

SELECT Description
Type a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel.
Specify the Z line command in this column to specify the columns that you want to zoom. The zoom function displays two panels that contain all the information about a single column. The following figure shows the first User Format Zoom panel:

![Image of User Format Zoom panel](image)

**Figure 42. User Format Zoom panel (1 of 2)**

Use the PF3 key to return to the Select User Format panel, or press Enter to display the second User Format Zoom panel, as shown in the following figure:

![Image of User Format Zoom panel](image)

**Figure 43. User Format Zoom panel (2 of 2)**

Use the PF3 key to return to the Select User Format panel.

**Name**
Indicates the name of the column.

**Type**
Indicates the type of the column.

**Len.**
Indicates the length of the column.

**Scale**
Indicates the decimal part of the column.

2. Edit any of the following fields on the first panel and press Enter.
Type
Specify the type of the column after the conversion.

Len.
Specify the total length of the column. This field is authorized for columns of type CHAR, DECIMAL, ..., and its value must be less than the value shown in Length.

Scale
Specify the length of the decimal part of the column. This field is authorized for columns of type DECIMAL, ..., and its value must be less than the value shown in Length.

Pad
Specify the padding character in hexadecimal.

Just
Specify L for left justification or R for right justification.

The second Select User Format panel is displayed, as shown in the following figure:

```
INZB502 -- D27M IBMUSER . TSTEST16 - Select User Format Row 1 from 5
Command ====> Scroll ===> CSR
Commands : FORMAT COLUMNS WHERE ORDERBY OUTDDN LOADDDN JCL
* * * * *
Line commands: Z - Zoom
SELECT Description . . IBMUSER .TSTEST16
( PF3 to return to "Select List")

<table>
<thead>
<tr>
<th>S Name</th>
<th>Type</th>
<th>Length</th>
<th>Scale</th>
<th>Nul/def</th>
<th>Delim</th>
<th>Lengthbyte</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL1</td>
<td>INTEGER</td>
<td></td>
<td>N</td>
<td>____</td>
<td>____</td>
<td>____</td>
<td>____</td>
</tr>
<tr>
<td>COL2</td>
<td>CHAR</td>
<td>50</td>
<td>N</td>
<td>WD</td>
<td>____</td>
<td>____</td>
<td>____</td>
</tr>
<tr>
<td>AAAAAA</td>
<td>CHAR</td>
<td>15</td>
<td>N</td>
<td>____</td>
<td>____</td>
<td>____</td>
<td>____</td>
</tr>
<tr>
<td>COL4</td>
<td>VARCHAR</td>
<td>50</td>
<td>Y</td>
<td>WD</td>
<td>____</td>
<td>____</td>
<td>____</td>
</tr>
<tr>
<td>COL3</td>
<td>CHAR</td>
<td>50</td>
<td>N</td>
<td>WD</td>
<td>____</td>
<td>____</td>
<td>____</td>
</tr>
</tbody>
</table>

Figure 44. Select User Format panel (2 of 4)
```

The following field is available only on the second Select User Format panel. You cannot edit this field.

Nul/def
Indicates the Null option of the column.

N
The Null is not authorized.

Y
The Null is authorized.

WD
With Default.

3. Edit any of the following fields on the second Select User Format panel and press Enter.

Delim
Specify the delimiter character at the beginning and the end of a field.

Lengthbyte
Specify the length byte.

Length
Specify whether the real or maximum length is to be used for fields of variable length.

The third Select User Format panel is displayed, as shown in the following figure:
4. Edit any of the following fields on the third **Select User Format** panel and press Enter.

- **Sign**
  - Specify the numeric data display format.

- **Pos**
  - Specify the position of the sign.

- **Dec**
  - Specify the decimal separator.

- **Mask**
  - Specify the mask value.

- **CCSIDs**
  - You can specify up to three optional identifiers of the code character set code for the unloaded data: SBCS, MIXED, and DBCS values. If any field is omitted or is specified as 0, the corresponding CCSID is assumed to be the same as the installation default CCSID.

  You can load the FIELDPROC and EDITPROC directly from the DSNEXIT that is specified in the PARMLIB without allocating it in STEPLIB.

  **Attention:** If the TRANSLAT parameter in the PARMLIB is set to inhibit data conversion, you can specify a conversion parameter (ASCII, EBCDIC, ASIS, or CCSID) in the SYSIN to override this inhibition.

  If the unload format that is specified either in the SYSIN or in the PARMLIB by using the UNLSCHEM parameter is not identical to the EBCDIC format of the system, all constants that are specified in SYSIN are translated to the unload format.

The fourth **Select User Format** panel is displayed, as shown in the following figure:
INZB564 -- DZ7M IBMUSER . TSTEST16 - Select User Forma Row 1 from 5
Command ===> Scroll ===> CSR

Commands : FORMAT COLUMNS WHERE ORDERBY OUTDDN LOADDDN JCL
* * *

Line commands: Z - Zoom

SELECT Description . . IBMUSER .TTEST16

( PF3 to return to "Select List")

<-------- Columns description ---------> <------- Field description ------->
<-- (PF10)

S Name               Type     Length Scale Null When Null  Not Null  Null ID
- -----------------  -------- ------ ----- ---- ---------  --------  -------
_ COL1               INTEGER                _       __        __        _
_ COL2               CHAR         50        _       __        __        _
_ AAAAAAAA           CHAR         15        _       __        __        _
_ COL4               VARCHAR     50        _       __        __        _
_ COL3               CHAR         50        _       __        __        _

******************************* Bottom of data ********************************

Figure 46. Select User Format panel (4 of 4)

5. Edit any of the following fields on the fourth Select User Format panel, and press Enter.

**Null**
Specify whether the null character is to be generated in the output file.

**Not Null**
Specify the character to be used for columns having a value other than Null.

**Null ID**
Specify whether the output file will contain Null indicators. This option is available only for the USER format or, if specified, at the SELECT level.

**Related reference**
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

---

**Displaying the list of OUTDDN files**

The OUTDDN list panel is used to display the list of OUTDDN files.

**Procedure**

1. Specify the OUTDDN command.

The **OUTDDN List** panel is displayed, as shown in the following figure:
Figure 47. OUTDDN List panel

You can use the following selection codes on the OUTDDN List panel:

The OUTDDN List panel contains the following fields:

**S**
Use this column to specify the partitions to be selected or deselected. You can enter one of the following selection codes in this column:

- **S**: Use the S selection code to specify values for output data sets.

- **D**: Deletes a file.

- **C**: Copies a file to a new file.

**Data Set Name**
Indicates the name of the OUTDDN data set. The following variables can be used, during the generation of the JCL, a substitution occurs.

- **&DBNAME**: this variable is replaced by the database name that contains the unloaded tablespace.
- **&TSNAME**: this variable is replaced by the name of the unloaded tablespace.
- **&SPACE**: this variable is replaced by the name of the unloaded tablespace. &SPACE is a synonym of &TSNAME.
- **&ZUSER**: this variable is replaced by the user name used to connect, where the interactive menu is running.
- **&DB2**: this variable is replaced by the SSID where the generated JCL will run.
- **&PART**: this variable is replaced by the partition number:
  - 000: when the unloaded tablespace is not partitioned or
  - nnnn: 4 digits for the partition number
- **&CREATOR**: this variable is replaced by the owner of the table specified in the SELECT.
- **&TBNAME**: this variable is replaced by the owner of the table specified in the SELECT. This variable can be used only if the name of the table has at most 8 characters. When above, an ISPF message is issued.

Indicates the name of the data set that will contain the result of the SELECT statement.

For partitioned table spaces, to define one output file for each partition, use the &PART variable in the generic data set name, which allows Db2 HPU to process in parallel.
If Db2 HPU was requested to perform parallel processing for partitioned table spaces, you must specify this processing for all files.

**Disp**
- Indicates the disposition of the OUTDDN files.

**New/Old/Mod**
- Indicates the disposition to use as the first parameter.

**Del/Keep/Catlg**
- Indicates the disposition to use as the second and third parameters.

2. Specify the $S$ selection code in the $S$ field.

The **OUTDDN File** panel is displayed, as shown in the following figure:

![OUTDDN File panel](image)

3. Edit the following fields:

**Data Set Name**
- For partitioned table spaces, to define one output file for each partition, use the &PART variable in the generic data set name, which allows Db2 HPU to process in parallel.

If Db2 HPU was requested to perform a processing in parallel for partitioned table spaces, you must specify this processing for all files.

**Disposition**
- Specify whether the OUTDDN data set is shared.

**New/Old/Mod**
- Specify the disposition to use as the first parameter.

**Del/Keep/Catlg**
- Specify the disposition to use as the second and third parameters.

**Generic unit**
- Specify the DASD unit name.

**Volume serial**
- Specify the volume serial number.

**Space units**
- Specify the volume allocation unit (BLKS, TRKS, or CYLS).
Primary quantity
Specify the primary allocation for the data set.

Secondary quantity
Specify the secondary allocation for the data set.

Record format
Specify the format of the records as F, FB, V, or VB.

Record length
Specify the length of the records with a value less than or equal to 32760.

Block size
If FB or VB was specified in the FORMAT field, you must specify the block size. The BLKSIZE value must be less than or equal to 32760 and it must be a multiple of LRECL (+4 if V or VB).

SMS
Specify the Data Class, Management Class, and Storage Class parameters for allocation data sets that are managed by SMS. The values that are entered for these parameters are in the DD statement that is generated in the JCL.

for tape unit/Label
Specify the tape sequence number if the file is to be written to tape.

Related reference
Db2 HPU interactive component primary commands
Primary commands are used with the Db2 HPU interactive component.

Displaying and modifying LOADDIN file parameters
The LOADDIN file panel is used to display and modify LOADDIN file parameters.

Procedure
1. Specify the LOADDIN command.
   The LOADDIN File panel is displayed, as shown in the following figure:

   INZB70I ---- DZ7M IBMUSER . TSTEST16 - LOADDIN File -------- 12:05
   Command ===> Commands : FORMAT COLUMN WHERE ORDERBY OUTDDN FUSER JCL
   * * * * Other Option : LOADOPT
   Generated SYSIN LOAD data set
   SELECT Description . IBMUSER .TSTEST16
   SYSOUT class .
   Data set name .
   or Member name for PDS .
   Disposition . (New/Old/Mod) (Del/Keep/Catlg)
   Generic unit .
   Volume serial .
   for tape unit
   Label .
   ( PF3 to return to “SELECT List”)

   Figure 49. LOADDIN File panel

2. Edit the following fields:

   Other options: LoadOPT
   Specify LoadOPT to display an ISPF edit panel that you use to enter load options.
**SELECT Description**
Specify a short description of the SELECT statement. The description is displayed in the Select description column of the **SELECT Statement List** panel.

**SYSOUT class or Data Set Name**
Specify the SYSOUT class or the data set name for the LOADDDN file. If you enter a value in both fields, the value entered in the SYSOUT class field is used.

When entering a Data Set Name, the following variables can be used, during the generation of the JCL, a substitution occurs.

- **&DBNAME**: this variable is replaced by the database name that contains the unloaded tablespace.
- **&TSNAME**: this variable is replaced by the name of the unloaded tablespace.
- **&SPACE**: this variable is replaced by the name of the unloaded tablespace. **&SPACE** is a synonym of **&TSNAME**.
- **&ZUSER**: this variable is replaced by the user name used to connect, where the interactive menu is running.
- **&DB2**: this variable is replaced by the SSID where the generated JCL will run.
- **&PART**: this variable is replaced by the partition number:
  - 000: when the unloaded tablespace is not partitioned or
  - nnnn: 4 digits for the partition number
- **&CREATOR**: this variable is replaced by the owner of the table specified in the SELECT.
- **&TBNAME**: this variable is replaced by the owner of the table specified in the SELECT. This variable can be used only if the name of the table has at most 8 characters. When above, an ISPF message is issued.

**Member name for PDS**
Specify the PDS member name.

**Disp**
Specify the disposition and whether the LOADDDN data set is to be shared.

**Generic unit**
Specify the DASD unit name.

**Volume serial**
Specify the volume serial number.

**for tape unit Label**
Specify the tape sequence number, if the file will be written to tape.

**Related reference**
*Db2 HPU interactive component primary commands*
Primary commands are used with the Db2 HPU interactive component.

---

**Generating JCL to run a Db2 HPU job**
You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a Db2 HPU job by invoking the JCL command.

**About this task**

**Before submitting a JCL**

The typical use of HPU assumes that the SCEERUN and SCEERUN2 Language Environment libraries are available through the linklist. If this condition is not fulfilled in your environment, the Db2 HPU job generated by the interactive component will fail. To avoid this, make these libraries available to Db2 HPU
before you run the JCL. To do so, consider adding a JOBLIB control card in the user jobcards that can be specified through the "JCL/JobCard" option.

**Procedure**

1. Specify the **JCL** command.

   The **EDIT, SUBmit, SaveJCL, JobCard** window is displayed when you enter the JCL command.

   ![](image)

   **Figure 50. EDIT, SUBmit, SaveJCL, JobCard window**

   2. Position the cursor beside one of the following options and press Enter.

   **EDIT**
   
   Select this option to generate and edit JCL. An ISPF edit session is opened on the generated JCL. You can review it, modify it, and resubmit it.

   **SUBmit**
   
   Select this option to generate and submit JCL. The job is generated and submitted.

   **SaveJCL**
   
   Select this option to generate and save JCL.

   **JobCard**
   
   Select this option to update the job card that is used to generate JCL.

**Related tasks**

- **Saving JCL into a file**
  
  You can save the generated JCL into a file.

- **Updating the job card**
  
  This option is used to update the job card that is used for JCL generation.

**Related reference**

- **Db2 HPU interactive component primary commands**
  
  Primary commands are used with the Db2 HPU interactive component.

**Saving JCL into a file**

You can save the generated JCL into a file.

**About this task**

To save generated JCL into a file:

**Procedure**

1. Specify the **JCL** command.
2. Select the SAVE JCL option on the **EDIT, SUBmit, SaveJCL, JobCard** window. The **SAVE JCL** panel is displayed, as shown in the following figure:

![SAVE JCL panel](image)

**Figure 51. SAVE JCL panel**

3. Edit the following fields:

**DS name**
Specify the name of an existing PDS where you want to save the JCL. If the name is entered between single quotation marks, the TSO prefix is not added.

**Member**
Specify the member name. If this member exists, it is by the new one.

**Related tasks**
Generating JCL to run a Db2 HPU job
You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a Db2 HPU job by invoking the JCL command.

**Updating the job card**
This option is used to update the job card that is used for JCL generation.

**Procedure**
1. Specify the JCL command.
2. Select the JobCard option on the **EDIT, SUBmit, SaveJCL, JobCard** window. The **User Job Card** panel is displayed, as shown in the following figure:

![User Job Card panel](image)

**Figure 52. User Job Card panel**

3. Edit the following fields:

**Enter below the JOB card for the runs:**
Specify 1 - 4 lines for the job card. The syntax is identical to the standard job card syntax.
**REGION (MEG)**
Specify the maximum memory size (in MB) for the unload step.

**TIME (MIN)**
Specify the CPU time consumption limit (in minutes) for the unload step.

**Related tasks**
Generating JCL to run a Db2 HPU job
You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a Db2 HPU job by invoking the JCL command.

**Selecting date formats**
The **Type Values - DATE** panel is used to select a date format.

**About this task**
To select a date format:

**Procedure**
1. Specify a question mark (?) in the **DATE** field of a previous **SELECT Statement Option** panel.
   The **Type Values - DATE** panel is displayed, as shown in the following figure:

   ![Type Values - DATE panel](image)

   **Figure 53. Type Values - DATE panel**

   2. Specify the S selection code to select the date format.

   **Related tasks**
   Specifying options for the SELECT statement
   The **SELECT Statement Option** panels are used to specify options for a SELECT statement.
   Selecting timestamp formats
The **Type Values - TIMESTAMP** panel is used to select a timestamp format.

## Selecting time formats

The **Type Values - TIME** panel is used to select a time format.

### Procedure

1. Specify a question mark (?) in the **TIME** field of a previous **SELECT Statement Option** panel.

The **Type Values - TIME** panel is displayed, as shown in the following figure:

![Type Values - TIME panel](image)

2. Specify the S selection code to select the time format.

### Related tasks

- **Specifying options for the SELECT statement**
  - The **SELECT Statement Option** panels are used to specify options for a SELECT statement.
- **Selecting tables**
  - The **Table List** panel is used to create SELECT statements.

### Related reference

- **TIME format types**
  - Use the **TIME** format type to specify the output data format.

## Selecting timestamp formats

The **Type Values - TIMESTAMP** panel is used to select a timestamp format.

### Procedure

1. Specify a question mark (?) in the **TIMESTAMP** field of a previous **SELECT Statement Option** panel.

The **Type Values - TIMESTAMP** panel is displayed, as shown in the following figure:
Related tasks
Specifying options for the SELECT statement
The SELECT Statement Option panels are used to specify options for a SELECT statement.

Selecting date formats
The Type Values - DATE panel is used to select a date format.

Related reference
TIMESTAMP format types
Use the TIMESTAMP format type to specify the output data format.

Selecting mask formats
The Type Values - MASK panel is used to select a mask format.

Procedure
Specify a question mark (?) in the MASK field of a previous SELECT Statement Option panel.

The Type Values - MASK panel is displayed, as shown in the following figure:
**Figure 56. Type values - MASK panel**

<table>
<thead>
<tr>
<th>S Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>. <em>.</em></td>
<td>Unnecessary 0 will not be added</td>
<td></td>
</tr>
<tr>
<td>. *0</td>
<td>Decimal part will be padded with 0</td>
<td></td>
</tr>
<tr>
<td>. *Z</td>
<td>As .0 plus the decimal point is always printed</td>
<td></td>
</tr>
<tr>
<td>. 0.*</td>
<td>A digit must be present prior the decimal point</td>
<td></td>
</tr>
<tr>
<td>. 0.0</td>
<td>The int. part is present and dec. one is padded</td>
<td></td>
</tr>
<tr>
<td>. 0.Z</td>
<td>As 0.0 plus the decimal point is always printed</td>
<td></td>
</tr>
<tr>
<td>. 00.*</td>
<td>The integer part will be padded with 0</td>
<td></td>
</tr>
<tr>
<td>. 00.0</td>
<td>The integer and decimal parts are padded with 0</td>
<td></td>
</tr>
<tr>
<td>. 00.Z</td>
<td>As 00.0 plus the decimal point is always printed</td>
<td></td>
</tr>
</tbody>
</table>

******************************* Bottom of data ********************************
Chapter 7. Db2 HPU user exit

The Db2 HPU user exit is used to customize the output data set that is created by a SELECT statement. The exit can be written in any of the following languages:

- Assembler
- C
- COBOL/2

The Db2 HPU exit must reside in an authorized library and is loaded dynamically during Db2 HPU operation. This library must be in either the LINKLIST or in an authorized JOBLIB or STEPLIB.

For exits that are written in COBOL/2 and C, the STEPLIB, JOBLIB, or LINKLIST should also point to the LE/370 run-time libraries.

You do not need to modify Db2 HPU syntax to use the exit. Instead, specify the exit by using the OUTEXIT parameter. If this parameter is not present, Db2 HPU will not call the exit.

The Db2 HPU exit accesses the Db2 row in EXTERNAL format before application of the USER format. The exit can inspect, modify, or discard Db2 rows. However, the exit cannot modify the maximum length of variable-length columns for the unloaded tables.

The following topics provide additional information:

- “Register use” on page 277
- “Parameter block” on page 277
- “Sample Assembler user exit” on page 279
- “Sample COBOL/2 user exit” on page 282

Related reference

SELECT block syntax and description
The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

Register use

Db2 HPU communicates with the USER exit through registers 1 and 15. When the USER exit is called, register 1 contains the address of a parameter block, as described by the EXTXPLST DSECT in the sample member INFEXIT. Register 15 contains the return code of the exit.

Related concepts

Parameter block
When the USER exit is called, register 1 contains the address of a parameter block.

Related reference

Sample Assembler user exit
This sample user exit is provided in member INZEXIT (SINZSAMP library). It is written in Assembler.

Sample COBOL/2 user exit
This sample user exit is written in COBOL/2.

Parameter block

When the USER exit is called, register 1 contains the address of a parameter block.

The parameter block that is pointed to by register 1 contains the following parameters:
Parameter 1 (offset 0)
A function code (one binary word), which is one of the following values:

- **Function code 0 (process)**
  
  A function code of 0 indicates that all parameters are set, and that the data pointed to by the SQLDA can be checked and modified by the exit. The values of the columns are in EXTERNAL format, that is, they are not in Db2 internal format. For variable-length columns, the length can be changed, but must not be greater than the maximum length (field SQLLEN in the SQLDA).

  This function is not called if the initialization function returned a nonzero value.

  When the function code is 0, the return codes have the following meanings:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The row pointed to by the SQLDA is to be written to the output data set. A USER format can be used to modify the data after the exit is called.</td>
</tr>
<tr>
<td>4</td>
<td>This row is not to be written to the output data set.</td>
</tr>
<tr>
<td>8</td>
<td>The utility will stop.</td>
</tr>
</tbody>
</table>

- **Function code 1 (initialization)**
  
  A function code of 1 indicates that the exit was called before the table space was scanned and no data was available. The address of the SQLDA has a 0 value, and parameters 3, 4, and 5 are set. Parameter 6 contains the address of an 840-byte ready-to-use work area, which can be modified by the Exit if needed.

  When the function code is 1, the return codes have the following meanings:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The exit must be called (function code 0) for this SELECT.</td>
</tr>
<tr>
<td>4</td>
<td>The exit must not be called for this SELECT.</td>
</tr>
<tr>
<td>8</td>
<td>The utility must be stopped.</td>
</tr>
</tbody>
</table>

- **Function code 2 (end)**
  
  A function code of 2 indicates that only parameters 1, 5, or 6 should be used. The exit can, for example, deallocate the work area that was allocated during the initialization function.

  This function is called only if the initialization function return code was 0.

  When the function code is 2, any value of the return code is accepted.

**Parameter 2 (offset 4)**
The address of the SQLDA, which describes the selected columns and points to the data

**Parameter 3 (offset 8)**
The address of a table name

**Parameter 4 (offset 16)**
The address of the SSID

**Parameter 5 (offset 12)**
The user number, as shown in the example in “Sample Assembler user exit” on page 279 (one binary word)

**Parameter 6 (offset 52)**
The address of a work area. An 840-byte ready-to-use work area is allocated by the calling function, and the address is provided by this parameter. The Exit function can overwrite this address with the
work area address that is allocated during initialization, which is then unallocated at the end of the process.

**Parameter 7 (offset 28)**
The address of two varchar fields which point to the creator and table name.

**Related concepts**
Register use
Db2 HPU communicates with the USER exit through registers 1 and 15. When the USER exit is called, register 1 contains the address of a parameter block, as described by the EXTXPLST DSECT in the sample member INEXIT. Register 15 contains the return code of the exit.

**Related reference**
Sample Assembler user exit
This sample user exit is provided in member INZEXIT (SINZSAMP library). It is written in Assembler.

Sample COBOL/2 user exit
This sample user exit is written in COBOL/2.

---

**Sample Assembler user exit**

This sample user exit is provided in member INZEXIT (SINZSAMP library). It is written in Assembler.

```
***********************************************************************
* MEMBER : INZEXIT                                                  *
***********************************************************************
*
* 5655-AA1                                                        *
* © COPYRIGHT INFOTEL 1996, 2012 ALL RIGHTS RESERVED.               *
*
***********************************************************************
* THIS MEMBER IS CALLED 3 TIMES DURING THE UNLOADING OF A TABLE.    *
* R1 CONTAINS THE ADDRESS OF A PARAMETER BLOCK AS DESCRIBED BY      *
* THE EXTXPLST DSECT.                                             *
* THE MOST IMPORTANT FIELD IS THE POINTER TO THE SQLDA OF THE       *
* PROCESSED TABLE WHICH GIVES DATAS IN EXTERNAL FORMAT.             *
* FUNCTION 0 :                                                    *
* PROCESS OF THE DB2 ROW                                          *
* RETURN CODES :                                                  *
* 0 : ROW TO BE WRITTEN IN THE OUTPUT FILE                         *
* 4 : ROW DISCARDED                                               *
* FUNCTION 1 :                                                    *
* INITIALIZATION OF THE EXIT                                      *
* RETURN CODES :                                                  *
* 0 : EXIT ACTIVE FOR THIS SELECT STATEMENT                       *
* 4 : DESACTIVATION OF THE EXIT FOR THIS SELECT                   *
* FUNCTION 2 :                                                    *
* TERMINATION OF THE EXIT                                         *
* RETURN CODES NOT USED                                           *
***********************************************************************
R0   EQU  0
R1   EQU  1
R2   EQU  2
R3   EQU  3
R4   EQU  4
R5   EQU  5
R6   EQU  6
R7   EQU  7
R8   EQU  8
R9   EQU  9
R10  EQU 10
R11  EQU 11
R12  EQU 12
R13  EQU 13
R14  EQU 14
R15  EQU 15
```
* SQLDA DESCRIPTION
*
SQLDA DSECT
SQLDAID DS CL8 ID
SQLDABC DS F BYTE COUNT
SQLN DS H NUMBER OF COLUMNS
SOLD DS H NUMBER OF SELECTED COLUMNS
SQLVAR DS OF BEGIN COLUMNS
SQLDSIZ EQU *-SQLDA SIZE OF FIXED PART
*
SQLVARN DSECT COLUMN
SQLTYPE DS H TYPE
SQLLEN DS 0H LENGTH
SQLPRCSN DS X DEC PRECISION
SQLSCALE DS X DEC SCALE
SQLDATA DS A ADDRESS OF DATA
SQLIND DS A ADDRESS NULL FIELD INDICATOR
SQLNAME DS H,CL30 SIZE AND COLUMN NAME
SQLVSIZ EQU *-SQLVARN
*
EXTXPLST DSECT PARAMETERS PASSED TO THE EXIT
EXTXFUNC DS F FUNCTION (0 PROCESS, 1 INIT, 2 TERM)
EXTXASQL DS A ADDRESS OF SQLDA
EXTXATBN DS A ADDRESS OF TABLE IDENTIFICATOR
*
CREATOR(8) / TABLENAME(18)
EXTXH DS H RESERVED
EXTXNREF DS H REFERENCE NUMBERID
EXTXASSI DS A ADDRESS OF SSID(4)
EXTXAUTR DS A ADDRESS OF USER(8)
EXTXATID DS A ADDRESS OF UTILITY ID(16)
EXTXATB DS A ADDRESS OF TABLE IDENTIFICATOR
VARCHAR CREATOR/TABLENAME
EXTX5F DS 5F RESERVED
EXTX4F DS 4F FIELDS RESERVED FOR THE EXIT
EXTXAMSG DS A ADDRESSE OF MESSAGE
EXTXISM SG DS F LENGTH OF MESSAGE
EXTXUMSG DS CL100 EXIT MESSAGE AREA
EXTXUA WA DS OF EXIT WORK AREA
ORG EXTXPLST+1024 END OF LIST
*
GETM DSECT
SAVE DS 18F LOCAL SAVEAREA
TBCREATL DS H CREATOR LENGTH
TBCREAT DS CL160 TABLE CREATOR
TBNAMEL DS H TABLE NAME LENGTH
TBNAME DS CL160 TABLE NAME
WORK DS CL444 WORKAREA
GETML EQU *-GETM LENGTH
*
INZEXIT CSECT
STM R14,R12,12(R13) SAVE CALLERS REGISTERS
LR R10,R1 GET PARAMETERS
LR R10,R10 *
L R10,EXTXAUWA R10=WORKAREA ADDRESS
USING GETM,R11 WORKAREA ADDRESSABILITY
ST R13,4(R11) SAVE CALLERS SAVE AREA ADDRESS
L R2,EXTXFUNC GET FUNCTION
SLL R2,2 * 4 FOR DISPACHING
B **4(R2)
B PROCESS FUNCTION 0
B INIT FUNCTION 1
B TERM FUNCTION 2
*
INIT EQU *
*
A READY-TO-USE WORKAREA OF 840 BYTES IS ALLOCATED BY THE CALLING
FUNCTION. IF MORE SPACE IS NEEDED, A LARGER AREA SHOULD BE
ALLOCATED HERE.
* EXAMPLE:
* GETMAIN R,LV=GETML GET A WORKAREA
* ST R1,EXTXAUWA SAVE GETMAIN ADDRESS
* LR R11,R1
* ST R11,8(R13) GIVE CALLER MY SAVE AREA ADDRESS

280 Db2 High Performance Unload User's Guide
ST R13,4(R11)       SAVE CALLERS SAVE AREA ADDRESS
L  R3,EXTXATB       ADDRESS OF TABLE IDENTIFICATOR
L  R1,0 (R3)        TABLE CREATOR VARCHAR
LH R2,0( R1)        LENGTH
STH R2, TBCREATL    STORE LENGTH
BCTR R2, 0          FOR EXECUTE
EX R2, MVCOWNER     COPY OWNER

L  R1,4 (R3)        TABLE NAME VARCHAR
LH R2,0(R1)        LENGTH
STH R2,TBNAMEL     STORE LENGTH
BCTR R2,0          FOR EXECUTE
EX R2,MVCNAME      COPY TABLE NAME

*    B   RC0

MVCOWNER MVC TBCREAT, 2(R1)
MVCNAME MVC TBNAME, 2(R1)

*    PROCEDURE EQU *
L  R3,EXTXASQL       GET SQLDA ADDRESS
USING SQLDA,R3
LA R4,SQLDSIZ(,R3)  PASS FIXED AREA
USING SQLVARN,R4
XR R5,R5            GET NUMBER OF COLUMNS
LH R5,SQLN

LOOP EQU *
*    PROCESS OF THE TABLE COLUMNS
*    LA R4,SQLVSIZ(,R4)
BCT R5,LOOP          NEXT COLUMN
B   RC0              END PROCESS OK
*

TERM EQU *
*    IF A WORKING AREA WAS ALLOCATED, IT SHOULD BE FREED HERE
*    EXAMPLE:
*    L  R1,EXTXAUWA    FREE WORKAREA
*    FREEMAIN R,A=(R1),LV=GETML
*    B   RC0              END TERMINATION OK
*

RC0 EQU *
LA R15,0             B   RETURN

RC4 EQU *
LA R15,4             B   RETURN

RETURN EQU *
L  R13,SAVE+4
L  R14,12(R13)
LM R0,R12,20(R13)
BSM 0,R14
LTORG
END

Related concepts
Register use
Db2 HPU communicates with the USER exit through registers 1 and 15. When the USER exit is called, register 1 contains the address of a parameter block, as described by the EXTXPLST DSECT in the sample member INFEXIT. Register 15 contains the return code of the exit.

Parameter block
When the USER exit is called, register 1 contains the address of a parameter block.

Related reference
Sample COBOL/2 user exit
Sample COBOL/2 user exit

This sample user exit is written in COBOL/2.

*-------------------------------------
IDENTIFICATION DIVISION.
*-------------------------------------
PROGRAM-ID. INFEXIT0.

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
LINKAGE SECTION.
01 EXITPARMS.
  02 EXIT-FUNCTION-CODE PIC S9(9) BINARY.
     88 PROCESS-CALL VALUE ZERO.
     88 INITIALIZE-CALL VALUE 1.
     88 CLEANUP-CALL VALUE 2.
  02 EXIT-SQLDA-POINTER POINTER.
  02 FILLER POINTER.
  02 EXIT-REFERENCE-NUM PIC S9(4) BINARY.
  02 EXIT-SSID-POINTER POINTER.
  02 EXIT-USERID-POINTER POINTER.
  02 EXIT-UTILID-POINTER POINTER.
  02 EXIT-TBNAME-POINTER POINTER.
     02 FILLER PIC X(20).
  02 EXIT-WORKAREA-POINTER POINTER.
  02 FILLER PIC X(16).
  02 EXIT-USERMSG-POINTER POINTER.
  02 EXIT-MESSAGE-SIZE PIC S9(9) BINARY.
  02 EXIT-USER-MESSAGE PIC X(100).
01 WORKAREA.
  02 WORKAREA-TBCREATOR PIC X(8).
  02 WORKAREA-TBNAME PIC X(18).
  02 FILLER PIC X(818).
01 TABLE-INFO-VARCHAR
  02 TABLE-CREATOR-VARCHAR POINTER.
  02 TABLE-NAME-VARCHAR POINTER.
01 SSID PIC X(4).
01 USERID PIC X(8).
01 UTILID PIC X(16).
01 SQLDA.
  02 SQLDAX PIC X(8).
  02 SQLDABC PIC S9(8) COMP.
  02 SQLN PIC S9(4) COMP.
  02 SQLD PIC S9(4) COMP.
  02 SQLVAR OCCURS 1 TO 300 DEPENDING ON SQLN INDEXED BY I.
     03 SQLTYPE PIC S9(4) COMP.
     03 SQLLEN PIC S9(4) COMP.
     03 SQLDATA POINTER.
     03 SQLIND POINTER.
     03 SQLINDN REDEFINES SQLIND PIC S9(9) COMP.
  03 SQLNAME.
     04 SQLNAMEL PIC S9(4) COMP.
     04 SQLNAMEC PIC X(30).
*
PROCEDURE DIVISION USING EXITPARMS.
*
EVALUATE TRUE
  WHEN INITIALIZE-CALL
    PERFORM
      SET ADDRESS OF TABLE-INFO TO EXIT-TBNAME-POINTER
      SET ADDRESS OF WORKAREA TO EXIT-WORKAREA-POINTER
      MOVE 'INITIALISATION OK' TO EXIT-USER-MESSAGE
      MOVE ZERO TO RETURN-CODE
    END-PERFORM
  WHEN PROCESS-CALL
    PERFORM
      SET ADDRESS OF SQLDA TO EXIT-SQLDA-POINTER

Related concepts

Register use
Db2 HPU communicates with the USER exit through registers 1 and 15. When the USER exit is called, register 1 contains the address of a parameter block, as described by the EXTXPLST DSECT in the sample member INFEXIT. Register 15 contains the return code of the exit.

Parameter block
When the USER exit is called, register 1 contains the address of a parameter block.

Related reference
Sample Assembler user exit
This sample user exit is provided in member INZEXIT (SINZSAMP library). It is written in Assembler.
Chapter 8. Troubleshooting

Use these topics to diagnose and correct problems that you experience with Db2 HPU.

Decreasing the amount of necessary memory

The memory resources that are needed to run an unload significantly depends on the Db2 HPU settings, the specific unload to be processed, and the parallelism degree that is used.

About this task

The amount of memory that is needed to unload data increases along with the degree of parallelism that is used. If an unload job runs out of memory, consider reducing the parallelism degree.

Procedure

For the following conditions that apply to your environment, reduce the parallelism degree by using the specified method:

Multiple table spaces

When multiple table spaces are used in a single UNLOAD statement, such as multiple SELECT statements or the use of LIST, split the unique UNLOAD statement that uses multiple table spaces into several UNLOAD statements that uses fewer table spaces. Splitting the UNLOAD statement will prevent too many tasks from being run in parallel.

You can split the UNLOAD statement without changing it by adding the PARALLELISM keyword to the command by setting VUX030 UTLPARAL maximum_number, where maximum_number is the maximum number of table spaces that you want to process in parallel.

For example, the following statement ensure that the necessary amount of memory is reduced as much as possible by setting the parallelism for table space processing to 1.

```
TEMPLATE OUTPUT DSN MYFILES.TEST.&TS..S&TMP.
UNIT WORK SPACE(5,1) CYL
LISTDEF LIST INCLUDE TABLESPACE MYDB.TS%
UNLOAD TABLESPACE OPTIONS TEMPLATESET( TMP = :SELNUM )
PARALLELISM ( , , 1)
DB2 NO
SELECT * FROM LIST(LIST)
OUTDDN OUTPUT
FORMAT DSNTIAUL
```

Partitioned table spaces

When partitioned table spaces are involved, you can reduce the number of partitions that are processed in parallel by using several methods.

- Limit the scope of a single UNLOAD statement by using the PART () option. To perform the same unload, you must use multiple UNLOAD statements instead of one. For example, the following example shows a single UNLOAD statement that can be replaced with multiple statements that use the PART () option.

```
UNLOAD TABLESPACE MYDB.PART128
SELECT * FROM ME.MYTABLE
OUTDDN OUTPUT
```

To reduce the amount of necessary memory, replace the previous statement with the statements that are shown in the following example:

```
UNLOAD TABLESPACE MYDB.PART128 PART(1:32)
SELECT * FROM ME.MYTABLE
OUTDDN OUTPUT ;
```
UNLOAD TABLESPACE MYDB.PART128 PART(33:64)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;
UNLOAD TABLESPACE MYDB.PART128 PART(65:96)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;
UNLOAD TABLESPACE MYDB.PART128 PART(97:128)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;

• Add the MAXPART keyword to the UNLOAD statement or set the ULMAXPAR parameter. In this case, Db2 HPU splits the partitions to be processed into several groups, according to the maximum number of partitions that is specified by MAXPART or ULMAPAR. These groups of partitions are considered to be from different table spaces so they are allowed to be unloaded in parallel. To make sure the parallelism degree is actually reduced, the parallelism between table spaces must be set to 1 either by using the PARALLELISM(.,.,1) keyword or by setting the value of the VUX030 UTLPARAL parameter to 1, as shown in the following example.

UNLOAD TABLESPACE MAXPART 16
PARALLELISM( , , 1)
DB2 NO
SELECT * FROM ME.MYTABLE
OUTDDN (OUTP)

Also, consider using PARALLELISM(maximum_number,...) or VUX025 PARALLEL=maximum_number to limit partition parallelism to the value that you specify for maximum_number. For example, use the following UNLOAD statement to reduce the amount of memory that is needed to unload a table space with 128 partitions:

UNLOAD TABLESPACE PARALLELISM(32, , 1)
DB2 NO
SELECT * FROM ME.MYTABLE
OUTDDN (OUTP)

Tools Customizer troubleshooting

Use this information to diagnose and correct problems that you experience with Tools Customizer.

Gathering diagnostic information

Before you report a problem with Tools Customizer to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Tools Customizer problems:

• A clear description of the problem and the steps that are required to re-create the problem
• Relevant screen captures
• All messages that were issued as a result of the problem
• Product release number and the number of the last program temporary fix (PTF) that was installed
• The version of Db2 that you are using and the type and version of the operating system that you are using
• The Tools Customizer trace data set
• The Tools Customizer data store data set and the high_level_qualifier.SCQTEMPU data set
Determining the trace data set name

You will need to identify the name of the trace data set if you cannot allocate the trace data set, the trace data set runs out of space, or IBM Software Support asks for it.

The name of the trace data set depends on the prefix setting in the TSO profile. To identify the name of the trace data set, you must know the prefix setting.

- If PREFIX is set, the name of the trace data set is `prefix.CCQ.TRACE`, where `prefix` is the TSO prefix that you specified in the profile.
- If NOPREFIX is set, the name of the trace data set is `user_ID.CCQ.TRACE`, where `user_ID` is your TSO user ID.

Gathering diagnostic information

Before you report a problem with Db2 HPU to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Db2 HPU problems:
- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of Db2 that you are using and the type and version of the operating system that you are using

For errors in batch processing, provide the following information:
- The complete job log
- Print output
- The content of the PARMLIB, including all members or at least the INZUTIL member and the INZDSSID member, where `SSID` is the name of the Db2 subsystem or group attach name on which the issue occurs.

Messages and codes

Use the information in these messages to help you diagnose and solve Tools Customizer and Db2 HPU problems.

These topics describe the user abend codes and messages that are issued by Db2 HPU and Tools Customizer.

Topics:
- “Tools Customizer messages” on page 287
- “Db2 HPU user abend codes” on page 359
- “Db2 HPU messages” on page 364

Tools Customizer messages

Use the information in these messages to help you diagnose and solve Tools Customizer problems.

| CCQB000I | The product parameter data was saved in the data store. |

System action:
None.

User response:
No action is required.
The DB2 parameter data was saved in the data store.

Explanation:
Changes that were made to the Db2 parameters were saved in the data store.

System action:
None.

User response:
No action is required.

The LPAR parameter data was saved in the data store.

Explanation:
Changes that were made to the LPAR parameters were saved in the data store.

System action:
None.

User response:
No action is required.

At least one step must be selected in a selected task. The selected task is task_description.

Explanation:
When a task is selected, at least one step must be selected. A selected step is missing from the specified task.

System action:
Processing stops.

User response:
Select a step in the specified task or deselect the task.

The required information to run the Discover EXEC was saved in the data store.

Explanation:
The data store contains all the information that is required to run the Discover EXEC.

System action:
None.

User response:
No action is required.

The conflicting values for the parameter_name parameter must be resolved before the information can be saved.

Explanation:
Two values for one parameter conflict with each other, and they must be resolved to save the information.

System action:
Processing stops.

User response:
Resolve the conflicting values for the parameter.

One row must be selected.

Explanation:
One row in the table must be selected.

System action:
Processing stops.

User response:
Select one row.

Only one row can be selected.

Explanation:
Multiple rows in the table are selected, but only one row is allowed to be selected.

System action:
Processing stops.

User response:
Select only one row.

The jobs have been customized on the selected DB2 entries.

Explanation:
The jobs were customized on the Db2 entries that were selected.

System action:
None.

User response:
Press Enter to clear the message.

The jobs were not generated on one or more of the selected DB2 entries. Press PF3 to check the Db2 entries that were not customized.

Explanation:
The product was not customized on one or more of the Db2 entries that were selected.

System action:
None.

User response:
Press PF3 to see the Db2 entries on which the product was not customized. The status of these Db2 entries is Errors in Customization.

The edit session was started automatically because values for required parameters are missing or must be verified.

Explanation:
If product, LPAR parameters, or Db2 parameters are not defined or if parameter definitions must be verified, an editing session for the undefined or unverified parameters starts automatically.
User response:
Define values for all required product, LPAR parameters, or Db2 parameters.

CCQC003W The template_name template in the library_name metadata library does not contain any parameters.

Explanation:
The specified template does not have parameters.

System action:
None.

User response:
No action is required.

CCQC004S The value of the "type" attribute for the template_name template in the library_name metadata library does not match the value that was previously specified. The value is value_name, and the previously specified value is value_name.

Explanation:
The value of the "type" attribute must match the value that was previously specified.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC005S The template_name template exceeds the number of allowed templates for a customization sequence. The template is in the library_name metadata library.

Explanation:
The customization sequence can process only number templates. The specified template cannot be processed because the customization sequence already contains the maximum number of templates.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC006E The jobs could not be generated for the group_attach_name Db2 group attach name.

Explanation:
The customization jobs could not be generated for the specified Db2 group attach name.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC007E The jobs could not be generated for the subsystem_ID Db2 subsystem.

Explanation:
The customization jobs could not be generated for the specified Db2 subsystem.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC008E The jobs could not be generated for the member_name Db2 member.

Explanation:
The customization jobs could not be generated for the specified Db2 member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC009S The jobs were not generated for the DB2 entries.

Explanation:
One or more errors occurred while customization jobs were being generated for the selected Db2 entries.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC010S The template_name template could not be accessed in the library_name metadata library.

Explanation:
The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action:
Processing stops.

User response:
Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

**CCQC011S** The template_name template could not be written to the library_name customization library.

**Explanation:**
The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

**System action:**
Processing stops.

**User response:**
Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

**CCQC012W** The job card was generated with default values because the JOB keyword was missing.

**Explanation:**
Default values were used to generate the job card because the JOB keyword was not specified in the first line of the job card.

**System action:**
The job card was generated with default values.

**User response:**
No action is required. To generate the job card with your own values, add the JOB keyword in the first line of the job card.

**CCQC013W** The job card was generated with default values because the specified programmer name exceeded 20 characters.

**Explanation:**
Default values were used to generate the job card because the specified programmer name contained too many characters.

**System action:**
The job card was generated with default values.

**User response:**
No action is required. To generate the job card with your own values, add a valid programmer name in the job card. A valid programmer name is 1 - 20 characters.

**CCQC014W** The job card was generated with default values because the JOB keyword was not followed by a space.

**Explanation:**
Default values were used to generate the job card because a space did not follow the JOB keyword.

**System action:**
The job card was generated with default values.

**User response:**
No action is required. To generate the job card with your own values, add a space after the JOB keyword in the job card.

**CCQC015S** The template_name template in the library_name metadata library contains the following file-tailoring control statement: statement_name. This control statement is not valid in a template_type template.

**Explanation:**
The template_type template cannot contain the specified type of file-tailoring control statement.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQC016S** The )DOT file-tailoring control statement exceeded the number of allowed occurrences for the template_name template in the library_name metadata library.

**Explanation:**
The )DOT file-tailoring control statement can occur only a limited number of times in the specified template.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQC017S** The nested )DOT file-tailoring control statements exceeded the number of allowed occurrences in the template_name template in the library_name metadata library.

**Explanation:**
Nested )DOT file-tailoring control statements can occur only number times.
Chapter 8. Troubleshooting 291
- A screen capture of the Tools Customizer error. Ensure that the complete error message is displayed by pressing PF1.
- The Tools Customizer trace data set. It adheres to the following naming convention: TSO_ID.CCQ.TRACE, where TSO_ID is the TSO user ID that is running Tools Customizer.
- The ISPF file tailoring trace data set.

**CCQC025I** Customized jobs do not exist because they have not been generated.

**Explanation:**
The list of customized jobs cannot be displayed because the product has not been customized for any Db2 entries.

**System action:**
None.

**User response:**
Complete the steps to customize a product. Customized jobs are generated when all required product, LPAR parameters, and Db2 parameters are defined and at least one Db2 entry on which to customize the product has been selected.

**CCQC026S** The value of the "customized" attribute for the parameter_name parameter in the library_name metadata library template does not match the value that was previously specified. The value is value_name, and the previously specified value is value_name.

**Explanation:**
The value for the "customized" attribute for a parameter must match the value that was previously specified.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQC027S** The job_name customization job was not found in the library_name customization library.

**Explanation:**
The selected customization job does not exist in the customization library.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQC028S** The library_name customization library was not found.

**Explanation:**
The customization library does not exist.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQC029I** The customization jobs were generated for Product_name.

**Explanation:**
The customization jobs were generated for the specific product.

**System action:**
None.

**User response:**
No action is required.

**CCQC030S** The customization jobs cannot be generated because at least one DB2 entry must be associated with this product.

**Explanation:**
The product that you are customizing requires at least one Db2 entry to be associated with it before customization jobs can be generated.

**System action:**
None.

**User response:**
Associate a Db2 entry with the product that you are customizing, and regenerate the jobs.

**CCQC031I** The jobs were generated for the associated DB2 entries.

**Explanation:**
The customization jobs were generated for the Db2 entries that are associated with the product.

**System action:**
None.

**User response:**
No action is required.

**CCQC032S** The customization jobs were not generated for Product_name.

**Explanation:**
A severe error occurred while the jobs were being generated for the specified product.

**System action:**
None.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQC033S  The customization_library_name has no customized jobs.

Explanation:  
The specified customization library cannot be browsed or edited because it is empty.

System action:  
None.

User response:  
Generate customization jobs for the specified library, and browse or edit the library again.

CCQC034S  The specified operation is not allowed.

Explanation:  
Issuing commands against customization jobs from the customization library from an ISPF browse or edit session that was started on the Finish Product Customization panel is restricted.

System action:  
None.

User response:  
To make changes to customization jobs, follow the steps for recustomization.

CCQC035E  Before you generate customization jobs, edit the product parameters to select one or more tasks or steps, and then issue the G line command or the GENERATEALL command again.

Explanation:  
One or more tasks or steps must be selected before customization jobs can be generated.

System action:  
None.

User response:  
Edit the product parameters to select one or more tasks or steps. Then, issue the G line command or the GENERATEALL command again.

CCQC036E  Before you exit the Product Parameters panel, you must select one or more tasks or steps to generate customization jobs or issue the CANCEL command.

Explanation:  
One or more tasks or steps must be selected to generate customization jobs or the CANCEL command must be issued before you can exit the Product Parameters panel.

System action:  
None.

User response:  
Select one or more tasks or steps, or issue the CANCEL command.

CCQC037W  The customization information was not found.

Explanation:  
To use the JOBLIST command, the customization jobs must be regenerated by using the GENERATEALL command or the G line command.

System action:  
None.

User response:  
Issue the GENERATEALL command or the G line command to generate the customization jobs.

CCQC038W  The customization information cannot be accessed because the customization_member customization member is being used.

Explanation:  
The customization member that was specified on NNN is currently being used.

System action:  
None.

User response:  
Determine why the customization member is in use, release it, and redo the work.

CCQC039I  The VERIFY/VERIFYOFF command is not active in Generate mode.

Explanation:  
The VERIFY/VERIFYOFF command is not active in Generate mode because all values saved in Generate mode must already have been verified.

System action:  
None.

User response:  
No action is required.

CCQD000W  The member_name environment index member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:  
While determining if the specified environment index member is valid, the PL/I XML parser issued an exception warning code.

System action:  
Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQD001S The member_name environment index member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the specified environment index member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the error.

CCQD002S The XML structure of the member_name environment index member is not valid. The element_name element is unknown.

Explanation: The specified environment index member contains an unknown element.

System action: Processing stops.


CCQD003S The XML structure of the member_name environment index member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.


CCQD004S The XML structure of the member_name environment index member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.


CCQD005S The XML structure of the member_name environment index member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQD006S The XML structure of the member_name environment index member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation: The specified element occurs too many times in the environment index member.

System action: Processing stops.


CCQD007S The XML structure of the member_name environment index member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times in the environment index member.

System action: Processing stops.


CCQD008S The XML structure of the member_name environment index member is not valid. The
attribute_name attribute in the
element_name element cannot
call occur more than
maximum_number times.

Explanation:
The specified attribute occurs too many times in the
environment index member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQD009S The XML structure of the
member_name environment index
member is not valid. The
attribute_name attribute in the
element_name element must occur
at least minimum_number times.

Explanation:
The specified attribute does not occur enough times in the
environment index member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQD010S The XML structure of the
member_name environment index
member is not valid. Content is not
allowed for the attribute_name
attribute in the element_name
element, but content was found.

Explanation:
Content was found in an attribute that cannot contain
content. The name of the attribute and the name of the
element that contains it are indicated in the message
text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQD011S The XML structure of the
member_name environment index
member is not valid. Content is required for the attribute_name
attribute in the element_name
element, but content was not
found.

Explanation:
An attribute does not contain required content. The
name of the attribute and the name of the element
that contains it are indicated in the message text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQD012S The XML structure of the
member_name environment index
member is not valid. The content
length for the element_name
element exceeds
maximum_number characters.

Explanation:
An element contains too many characters. The name
of the element and the maximum number of allowed
characters are indicated in the message text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQD013S The XML structure of the
member_name environment index
member is not valid. The
attribute_name attribute in the
element_name element is
unknown.

Explanation:
The environment index member contains an unknown
attribute. The name of the unknown attribute and the
name of the element that contains it are indicated in
the message text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQD050S The following LPAR serial number
is duplicated in the environment
index member: serial_number.

Explanation:
The environment index member contains duplicate
LPAR serial numbers. The duplicate serial number is
indicated in the message text.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.
CCQD051S The following DB2 serial number is duplicated in the environment index member: serial_number.

Explanation: The environment index member contains duplicate DB2 serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.


CCQD052S The following DB2 group attach name is duplicated in the environment index member: group_attach_name.

Explanation: The environment index member contains duplicate group attach names.

System action: Processing stops.


CCQD053S The reference to the following DB2 subsystem for a DB2 group attach name is duplicated in the environment index member: subsystem_ID.

Explanation: The environment index member contains duplicate references to a DB2 subsystem for a DB2 group attach name.

System action: Processing stops.


CCQD054S The reference to the following DB2 subsystem for the LPAR_name LPAR is duplicated in the environment index member: subsystem_ID.

Explanation: The environment index member contains duplicate references to a DB2 subsystem for an LPAR. The duplicate subsystem ID is indicated in the message text.

System action: Processing stops.


CCQD055S The following Db2 group attach name was not found in the environment index member: group_attach_name.

Explanation: A group attach name that is referenced by a DB2 member does not exist in the environment index member.

System action: Processing stops.


CCQD056S The following LPAR was not found in the environment index member: LPAR_name.

Explanation: The LPAR does not exist in the environment index member.

System action: Processing stops.


CCQD057S The following LPAR is duplicated in the environment index member: LPAR_name.

Explanation: The environment index member contains duplicate LPARs. The name of the duplicate LPAR name is indicated in the message text.

System action: Processing stops.


CCQD100W The member_name product index member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.
User response:
See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception warning code.

**CCQD101S** The *member_name* product index member is not valid. The PL/I XML parser issued the following exception error code: 
*code_number*.

**Explanation:**
While determining if the product index member is valid, the PL/I XML parser issued the specified exception error code.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception error code. Ensure that the Tools Customizer data store data set DCB is the same as the sample SCCQSAMP(CCQCDATS) data set DCB.

**CCQD102S** The XML structure of the *member_name* product index member is not valid. The *element_name* element is unknown.

**Explanation:**
The specified product index member contains an unknown element.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD103S** The XML structure of the *member_name* product index member is not valid. Content is not allowed for the *element_name* element, but content was found.

**Explanation:**
Content was found for an element that cannot contain content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD104S** The XML structure of the *member_name* product index member is not valid. Content is required for the *element_name* element, but content was not found.

**Explanation:**
The specified element does not contain required content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD105S** The XML structure of the *member_name* product index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD106S** The XML structure of the *member_name* product index member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

**Explanation:**
The specified element occurs too many times in the product index member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD107S** The XML structure of the *member_name* product index member is not valid. The *element_name* element must occur at least *minimum_number* times.

**Explanation:**
The specified element does not occur enough times in the product index member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
The XML structure of the `member_name` product index member is not valid. The `attribute_name` attribute in the `element_name` element cannot occur more than `maximum_number` times.

**Explanation:**
An attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The XML structure of the `member_name` product index member is not valid. The `attribute_name` attribute in the `element_name` element must occur at least `minimum_number` times.

**Explanation:**
The specified attribute does not occur enough times in the product index member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The XML structure of the `member_name` product index member is not valid. Content is not allowed for the `attribute_name` attribute in the `element_name` element, but content was found.

**Explanation:**
An attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The content of the `member_name` product index member is not valid. The `configuration_ID` configuration ID for the `configuration-name` configuration name is not unique.

**Explanation:**
System action:
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
CCQD120S  The content of the member_name product index member is not valid. The pack ID pack_ID that is referenced by product prefix product_prefix in the metadata library library_name could not be found.

Explanation: The specified pack ID could not be found in the metadata library.

System action: Processing stops.


CCQD121I  The specified pack contains the component_name, which was previously specified as a stand-alone product.

Explanation: The specified component of the pack was previously specified as a stand-alone product.

System action: None.

User response: No action is required.

CCQD122I  The specified component metadata library was previously specified as part of the pack_name.

Explanation: The specified metadata library for the component was previously specified as part of a pack.

System action: None.

User response: No action is required.

CCQD123E  The customization library name library_name is being used by another product or component. Specify another customization library qualifier on the Tools Customizer Settings panel.

Explanation: A different product or component is using the specified customization library.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception error code.

CCQD124E  The customization library library_name is in use by another metadata library.

Explanation: A different product or component is using the specified customization library. Specify another metadata library in the Workplace panel.

System action: None.

User response: Specify another customization library qualifier in the Tools Customizer Settings panel.

CCQD300W  The member_name product environment member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception warning code.

CCQD301S  The member_name product environment member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception error code.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the specified exception error code.

CCQD302S  The XML structure of the member_name product environment member is not valid. The element_name element is unknown.

Explanation: A different product or component is using the specified customization library.

System action: None.

User response: None.
The specified product environment member contains an unknown element.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD303S** The XML structure of the member_name product environment member is not valid. Content is not allowed for the element_name element, but content was found.

**Explanation:**
Content was found for an element that cannot contain content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD304S** The XML structure of the member_name product environment member is not valid. Content is required for the element_name element, but content was not found.

**Explanation:**
The specified element does not contain required content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD305S** The XML structure of the member_name product environment member is not valid. The content length for the element_name element exceeds maximum_number characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD306S** The XML structure of the member_name product environment member is not valid. The element_name element cannot occur more than maximum_number times.

**Explanation:**
The specified element occurs too many times in the product environment member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD307S** The XML structure of the member_name product environment member is not valid. The element_name element must occur at least minimum_number times.

**Explanation:**
The specified element does not occur enough times in the product environment member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD308S** The XML structure of the member_name product environment member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

**Explanation:**
The specified attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD309S** The XML structure of the member_name product environment member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

**Explanation:**
The specified attribute does not occur enough times in the product environment member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQD310S**
The XML structure of the *member_name* product environment member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

**Explanation:**
The specified attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQD311S**
The XML structure of the *member_name* product environment member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

**Explanation:**
The specified attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQD312S**
The XML structure of the *member_name* product environment member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQD313S**
The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

**Explanation:**
The specified attribute in the product environment member is unknown.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQD350I**
The *subsystem_ID* Db2 subsystem is associated with this product.

**Explanation:**
The specified Db2 subsystem was added and saved in the Tools Customizer data store for the product to be customized.

**System action:**
Processing continues.

**User response:**
No action is required.

---

**CCQD351I**
The *member_name* Db2 member for the *group_attach_name* Db2 group attach name is associated with this product.

**Explanation:**
The specified Db2 member for the group attach name was added and saved in the Tools Customizer data store for the product to be customized.

**System action:**
Processing continues.

**User response:**
No action is required.

---

**CCQD352I**
The *group_attach_name* Db2 group attach name is associated with this product.

**Explanation:**
The specified Db2 group attach name was added and saved in the Tools Customizer data store for the product to be customized.

**System action:**
Processing continues.

**User response:**
No action is required.
The subsystem_ID Db2 subsystem is already associated with this product.

Explanation:
The specified Db2 subsystem cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action:
None.

User response:
Ensure that the Db2 subsystem is specified correctly. If the problem persists, contact IBM Software Support.

The member_name Db2 member for the group_attach_name Db2 group attach name is already associated with this product.

Explanation:
The specified Db2 member for the group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action:
None.

User response:
Ensure that the Db2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

The group_attach_name Db2 group attach name is already associated with this product.

Explanation:
The specified Db2 group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action:
Processing stops.

User response:
Ensure that the Db2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

The library_name metadata library is already associated with the maximum number of allowed Db2 entries for this product.

Explanation:
The specified metadata library cannot be associated with more Db2 entries because it is already associated with the number of Db2 entries that are allowed.

System action:
Processing stops.

User response:
Delete an associated Db2 entry, and associate the specified library with another Db2 entry again.

The subsystem_ID Db2 subsystem is unassociated with this product.

Explanation:
The specified Db2 SSID was unassociated with the product that you are customizing.

System action:
Processing continues.

User response:
No action is required.

The member_name Db2 member for the group_attach_name Db2 group attach name is unassociated with this product.

Explanation:
The specified Db2 member for the Db2 group attach name was unassociated with the product that you are customizing.

System action:
Processing continues.

User response:
No action is required.

The group_attach_name Db2 group attach name is unassociated with this product.

Explanation:
The specified Db2 group attach name was unassociated with the product that you are customizing.

System action:
Processing continues.

User response:
No action is required.

The library_name metadata library is not associated with the specified Db2 subsystem subsystem_ID.

Explanation:
The specified Db2 subsystem and metadata library are not associated with each other.

System action:
None.

User response:
Ensure that the Db2 subsystem and the metadata library are associated. If the problem persists, contact IBM Software Support.
CCQD361S  The library_name metadata library is not associated with the specified DB2 data sharing group member member_name for the group_attach_name Db2 group attach name.

Explanation:  The specified Db2 data sharing group member for the group attach name and metadata library are not associated with each other.

System action:  None.

User response:  Ensure that the Db2 data sharing group member for the group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD362S  The library_name metadata library is not associated with the specified group_attach_name Db2 group attach name.

Explanation:  The specified Db2 group attach name and metadata library are not associated with each other.

System action:  None.

User response:  Ensure that the Db2 group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD400W  The customization parser issued the code_number warning code while it parsed the product customization member member_name. See the PL/I programming guide for more information about this XML parser continuable exception code.

Explanation:  While determining if the specified member is valid, the PL/I XML parser issued an exception warning code.

System action:  Processing stops.

User response:  See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQD401S  The customization parser issued the code_number error code while it parsed the product customization member member_name. See the PL/I programming guide for more information about this XML parser terminating exception code.

Explanation:  While determining if the specified member is valid, the PL/I XML parser issued an exception error code.

System action:  Processing stops.

User response:  See the Enterprise PL/I for z/OS Programming Guide for more information about the error.

CCQD500W  The data_set_name data store data set was not found.

Explanation:  Tools Customizer could not find the specified data store data set.

System action:  None.

User response:  No action is required.

CCQD501W  The data_set_name data store data set was not found, so it was created.

Explanation:  Tools Customizer created the specified data set because it could not be found.

System action:  None.

User response:  No action is required.

CCQD502E  The data_set_name data store data set is not writable.

Explanation:  Tools Customizer cannot write to the specified data set.

System action:  None.

User response:  Ensure that the data set is writable.

CCQD503E  The data_set_name data store data set could not be opened with the disposition_type disposition.

Explanation:  Tools Customizer could not open the data set with the specified disposition.

System action:  Processing stops.

User response:  

Chapter 8. Troubleshooting 303
Ensure that you have WRITE authority access to this data set.

**CCQD504E** The *data_set_name* data store data set could not be opened with the *option_name* option.

**Explanation:**
Tools Customizer could not open the data set with the specified option.

**System action:**
Processing stops.

**User response:**
Ensure that you have WRITE authority access to this data set.

**CCQD505E** The data store data set *data_set_name* already exists in a different volume.

**Explanation:**
Tools Customizer could not create the specified data set because the specified data set already exists in a different volume. Data store data set names must be unique.

**System action:**
Processing stops.

**User response:**
Specify a different data store data set name.

**CCQD510I** The DB2 SSID and Db2 group attach name were created.

**Explanation:**
The Db2 SSID and Db2 group attach name were created and saved in the data store.

**System action:**
None.

**User response:**
No action is required.

**CCQD511E** The DB2 entry already exists in the list of Db2 entries to be associated.

**Explanation:**
The Db2 entry cannot be added because it already exists in the list of Db2 entries to be associated.

**System action:**
None.

**User response:**
Specify a different Db2 entry.

**CCQD512S** An error occurred while a DB2 entry was being created.

**Explanation:**
A severe error occurred while a Db2 entry was being created.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD513E** The specified DB2 entry already exists and is associated with the current product on the Customizer Workplace panel.

**Explanation:**
The Db2 entry cannot be added because it already exists, and it is already associated with the product to be customized.

**System action:**
None.

**User response:**
Press F3 to go to the Customizer Workplace panel to see the Db2 entry, or specify a different Db2 entry.

**CCQD514E** A value is required for a DB2 subsystem, a Db2 group attach name, or both before they can be created.

**Explanation:**
Required information is missing. A Db2 subsystem, a Db2 group attach name, or both must be specified.

**System action:**
None.

**User response:**
Specify a Db2 subsystem, a Db2 group attach name, or both.

**CCQD515E** The specified DB2 entry already exists in the list of Db2 entries and is already associated with the current product.

**Explanation:**
The Db2 entry has already been created and associated with the product that you want to customize.

**System action:**
None.

**User response:**
Specify a different Db2 entry.

**CCQD516E** The specified DB2 entry already exists in the list of Db2 entries on the Associate DB2 Entry with Product panel but is not associated with the current product.

**Explanation:**

**System action:**
None.

**User response:**
Specify a different Db2 entry.
The Db2 entry exists, but it must be associated with the product to be customized.

System action: None.

User response: On the Customizer Workplace panel, issue the ASSOCIATE command to associate the Db2 entry with the product.

CCQD517S  An error occurred while a DB2 entry was being copied.

Explanation: A severe error occurred while a Db2 entry was being copied

System action: Processing stops.


CCQD518E  A value is required for a DB2 subsystem, a Db2 group attach name, or both before they can be copied.

Explanation: Required information is missing. A Db2 subsystem, a Db2 group attach name, or both must be specified.

System action: None.

User response: Specify a Db2 subsystem, a Db2 group attach name, or both.

CCQD519I  The DB2 entry was copied.

Explanation: The Db2 entry was copied and saved in the Tools Customizer data store.

System action: None.

User response: No action is required.

CCQD520S  The DB2 entries were created.

Explanation: The Db2 entries were copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.

System action: Processing stops.

User response: No action is required.

Remove a Db2 entry from the list, and copy the specified Db2 entry again.

CCQD521E  Line_command is not a valid line command.

Explanation: The specified line command is not valid. Valid line commands are on the panel.

System action: Processing stops.

User response: Specify a valid line command.

CCQD522E  The subsystem_ID Db2 subsystem ID occurs more than once in the list. Each row must be unique.

Explanation: The specified Db2 subsystem ID can be used only once.

System action: Processing stops.

User response: Specify a different Db2 subsystem ID.

CCQD523E  The group_attach_name Db2 group attach name occurs more than once in the list. Each row must be unique.

Explanation: The specified Db2 group attach name can be used only once.

System action: Processing stops.

User response: Specify a different Db2 group attach name.

CCQD524E  The member_name Db2 member for the Db2 group attach name occurs more than once in the list. Each row must be unique.

Explanation: The specified Db2 member for the Db2 group attach name can be used only once.

System action: Processing stops.

User response: Specify a different Db2 member for the Db2 group attach name.

CCQD525I  The DB2 entries were created.

User response: No action is required.
CCQD526E  The subsystem_ID Db2 subsystem ID occurs more than once in the list. Each Db2 subsystem ID must be unique.

Explanation: The specified Db2 subsystem ID can be used only once.

System action: Processing stops.

User response: Specify a different Db2 subsystem ID.

CCQD527I  DB2 group attach names cannot be created during the copy process.

Explanation: The ability to create Db2 group attach names is not available during the copy process.

System action: None.

User response: Create Db2 group attach names by issuing the CREATE command on the Customizer Workplace panel.

CCQD528E  The metadata_library metadata library is already associated with number Db2 entries. The maximum number of associated Db2 entries for this metadata library is 256.

Explanation: A metadata library can be associated with a maximum of 256 Db2 entries. The specified metadata library is already associated with 256.

System action: Processing stops.

User response: Remove an existing association between the specified metadata library and a Db2 entry, and associate the specified metadata library with another entry.

CCQD529I  At least one row is required.

CCQD560E  The subsystem_ID Db2 subsystem already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The specified Db2 subsystem exists and is associated with the product that you are customizing.

System action: None.

User response: Specify another Db2 subsystem.

CCQD561E  The member_name Db2 member for the group_attach_name Db2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The specified Db2 data sharing group for the Db2 group attach name exists and is associated with the product that you are customizing.

System action: None.

User response: Specify another Db2 subsystem.

CCQD562E  The group_attach_name Db2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The specified Db2 group attach name exists and is associated with the product that you are customizing. The subsystem is in the table on the Customizer Workplace panel.

System action: None.

User response: Specify another Db2 group attach name.

CCQD563E  A value is required for a DB2 subsystem, a Db2 group attach name, or both before they can be created.

Explanation: A Db2 subsystem, a Db2 group attach name, or both are not specified so one or both of them cannot be created.

System action: None.

User response: Specify a value for the Db2 subsystem, the Db2 group attach name, or both.

CCQD565E  The subsystem_ID Db2 subsystem already exists in the list of Db2 entries and is already associated with the current product.

Explanation: The specified subsystem is already associated.

System action: None.

User response:
User response:
Specify a different Db2 subsystem.

CCQD566E The member_name Db2 member for the group_attach_name Db2 group attach name already exists in the list of Db2 entries and is already associated with the current product.

Explanation:
The specified Db2 member is already associated.

System action:
None.

User response:
Specify a different Db2 member.

CCQD567E The group_attach_name Db2 group attach name already exists in the list of Db2 entries and is already associated with the current product.

Explanation:
The specified Db2 group attach name is already associated.

System action:
None.

User response:
Specify another Db2 group attach name.

CCQD568I product_name is not associated with a Db2 entry.

Explanation:
The product that you are trying to customize is not associated with a Db2 entry. Before a product can be customized, it must be associated with at least one Db2 entry.

System action:
None.

User response:
Associate one or more Db2 entries with the product.

CCQD569I The product_name product configuration is not associated with a Db2 entry.

Explanation:
The configuration for the specified product is not associated with a Db2 entry.

System action:
None.

User response:
Associate one or more Db2 entries with the configuration.

CCQD577W The mode_name Db2 mode of the subsystem_ID Db2 subsystem is not supported by the product.

Explanation:
The product does not support the specified Db2 mode.

System action:
None.

User response:
Specify a supported Db2 mode.

CCQD578W The mode_name Db2 mode of the member_name Db2 member for the Db2 group is not supported by the product.

Explanation:
The product does not support the specified Db2 mode.

System action:
None.

User response:
Specify a supported Db2 mode.

CCQD579W The mode_name Db2 mode of the group_name Db2 group attach name is not supported by the product.

Explanation:
The product does not support the specified Db2 mode.

System action:
None.

User response:
Specify a supported Db2 mode.

CCQD580S The subsystem_ID Db2 subsystem was copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.

Explanation:
The copied Db2 subsystem was not associated with the product because the product is associated with the maximum number of Db2 entries.

System action:
None.

User response:
Remove an associated Db2 entry and associate the product with the copied Db2 subsystem.

CCQD581S The member_name Db2 member for the group_attach_name Db2 group attach name was copied to the list of Db2 entries but was not associated because the product is
already associated with the allowed number of Db2 entries.

Explanation:
The copied Db2 member for the Db2 group attach name was not associated with the product because the product is associated with the maximum number of Db2 entries.

System action:
None.

User response:
Remove an associated Db2 entry and associate the product with the copied Db2 member.

CCQD582S

The group_attach_name Db2 group attach name was copied to the list of Db2 entries but was not associated because the product is already associated with the allowed number of Db2 entries.

Explanation:
The copied Db2 group attach name was not associated with the product because the product is associated with the maximum number of Db2 entries.

System action:
None.

User response:
Remove an associated Db2 entry and associate the product with the copied Db2 group attach name.

CCQD583I

The from_DB2 Db2 subsystem was copied to the to_DB2 subsystem.

System action:
None.

User response:
No action is required.

CCQD584I

The member_name DB2 member for the group_attach_name DB2 group attach name is copied to the subsystem_ID DB2 subsystem.

Explanation:
The specified Db2 member was copied.

System action:
None.

User response:
No action is required.

CCQD585I

The group_attach_name Db2 group attach name cannot be copied because a Db2 member is required.

Explanation:
The specified Db2 group attach name was not copied because a Db2 member was missing.

System action:
None.

User response:
No action is required.

CCQD586S

The current LPAR is LPAR_name, but the data store contains information about the LPAR_name LPAR. You must use the LPAR_name LPAR to customize the product.

Explanation:
The LPAR that is stored in the data store data set must be used to customize the product.

System action:
Processing stops.

User response:
Use the LPAR that is stored in the data store data set.

CCQD587W

The level_number Db2 level of the subsystem_name Db2 subsystem is not supported by the product.

Explanation:
The product does not support the specified Db2 level.

System action:
Processing continues.

User response:
Specify a supported level of Db2.

CCQD588W

The level_number Db2 level of the member_name Db2 member of the group_name Db2 group is not supported by the product.

Explanation:
The product does not support the specified Db2 level.

System action:
Processing continues.

User response:
Specify a supported level of Db2.

CCQD589W

The level_number Db2 level of the group_name Db2 group attach name is not supported by the product.

Explanation:
The product does not support the specified Db2 level.

System action:
Processing continues.

User response:
Specify a supported level of Db2.

CCQD593I

The subsystem_ID Db2 subsystem was deleted.
User response: No action is required.

CCQD594I The member_name Db2 for the group_attach_name Db2 group attach name was deleted.

User response: No action is required.

CCQD595I The group_attach_name Db2 group attach name was deleted.

User response: No action is required.

CCQD596E The subsystem_ID Db2 subsystem was not deleted.

Explanation: An internal error occurred while the specified Db2 subsystem was being deleted.

System action: Processing stops.


CCQD597E The member_name Db2 member for the group_attach_name Db2 group attach name was not deleted.

Explanation: An internal error occurred while the specified Db2 member was being deleted.

System action: Processing stops.


CCQD598E The group_attach_name Db2 group attach name was not deleted.

Explanation: An internal error occurred while the specified Db2 group attach name was being deleted.

System action: Processing stops.


CCQD600W The member_name product customization member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQD601S The member_name product customization member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception error code.

CCQD602S The XML structure of the member_name product customization member is not valid. The element_name element is unknown.

Explanation: The data store member contains an unknown element.

System action: Processing stops.


CCQD603S The XML structure of the member_name product customization member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.


CCQD604S The XML structure of the member_name product customization member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

valid. Content is required for the \texttt{element} element, but content was not found.

**Explanation:**
The specified element is missing required content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD605S** The XML structure of the \texttt{member} product customization member is not valid. The content length for the \texttt{element} element exceeds \texttt{maximum} characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD606S** The XML structure of the \texttt{member} product customization member is not valid. The \texttt{element} element cannot occur more than \texttt{maximum} times.

**Explanation:**
The specified element occurs too many times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD607S** The XML structure of the \texttt{member} product customization member is not valid. The \texttt{element} element must occur at least \texttt{minimum} times.

**Explanation:**
The specified element does not occur enough times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD608S** The XML structure of the \texttt{member} product customization member is not valid. The \texttt{attribute} attribute in the \texttt{element} element cannot occur more than \texttt{maximum} times.

**Explanation:**
The specified attribute occurs too many times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD609S** The XML structure of the \texttt{member} product customization member is not valid. The \texttt{attribute} attribute in the \texttt{element} element must occur at least \texttt{minimum} times.

**Explanation:**
The specified attribute does not occur enough times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD610S** The XML structure of the \texttt{member} product customization member is not valid. Content is not allowed for the \texttt{attribute} attribute in the \texttt{element} element, but content was found.

**Explanation:**
The specified attribute cannot contain content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQD611S** The XML structure of the \texttt{member} product customization member is not valid. Content is required for the \texttt{attribute} attribute in the \texttt{element} element, but content was not found.

**Explanation:**
The specified attribute does not contain required content.

System action: Processing stops.


CCQD612S The XML structure of the member_name product customization member is not valid. The content length for the element_name element exceeds maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQD613S The XML structure of the member_name product customization member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute in the data store member is unknown.

System action: Processing stops.


CCQD614S The content of the member_name product customization member is not valid. The value of the element_name element is not valid. The value is value_name.

Explanation: The specified value is not valid.

System action: Processing stops.


CCQD700W The member_name Db2 data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the XML structure of the Db2 data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQD701S The member_name Db2 data member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the XML structure of the Db2 data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception error code.

CCQD750W The value_number value in the Db2 parameter parameter_name was skipped because only maximum_number values are allowed.

Explanation: The specified value was skipped because it exceeds the number of allowed values in the Db2 parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the Db2 parameter.

CCQD800W The member_name LPAR data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.
User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

**CCQD801S** The member_name LPAR data member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

**CCQD850W** The value_number value in the LPAR parameter parameter_name was skipped because only maximum_number values are allowed.

Explanation: The specified value was skipped because it exceeds the number of allowed values in the LPAR parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the LPAR parameter.

**CCQD851I** The subsystem_ID Db2 subsystem is copied to the member_name Db2 member for the group_attach_name Db2 group attach name.

User response: No action is required.

**CCQD852I** The member_name Db2 member for the group_attach_name Db2 group attach name is copied to the member_name Db2 member for the group_attach_name Db2 group attach name.

User response: No action is required.

**CCQD854I** The member_name Db2 member for the group_attach_name Db2 group 'attach name is copied to multiple Db2 entries.

User response: No action is required.

**CCQD900W** The member_name product data member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

**CCQD901S** The member_name product data member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

**CCQD950W** The value_number value in the product parameter parameter_name was skipped because only maximum_number values are allowed.

Explanation: The specified value was skipped because it exceeds the number of allowed values in the product parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the product parameter.

**CCQD960I** The subsystem_ID Db2 subsystem was changed to the member_name Db2 member for the groupAttachName Db2 group attach name.

User response: No action is required.
CCQD961I The member_name Db2 member for the group_attach_name Db2 group attach name was changed to the subsystem_ID Db2 subsystem.

User response: No action is required.

CCQD962I The member_name Db2 member for the group_attach_name Db2 group attach name was changed to the member_name Db2 member for the group_attach_name Db2 group attach name.

User response: No action is required.

CCQD963E The DB2 group attach name cannot be blank when the Db2 subsystem ID is blank.

Explanation:
A Db2 group attach name, Db2 subsystem ID, or both must be specified.

System action: Processing stops.

User response: Specify a Db2 group attach name, Db2 subsystem ID, or both.

CCQE000S The specified message field name or message message_ID was not found.

Explanation:
An error occurred while displaying a message field name or the specified message.

System action: Processing stops.


CCQE001E An incorrect trace level was specified. Valid trace levels are 0 - 4.

Explanation:
A wrong trace level was specified. Valid trace levels are 0 - 4.

System action: Processing stops.

User response: Specify a valid trace level 0 - 4.

CCQF028E An asterisk was improperly specified in a filter argument.

Explanation:
An asterisk, which is treated as data, is embedded in the filter arguments. A generic filter argument is specified by placing the asterisk in the last nonblank position of the argument. No rows match the filter arguments, so all rows will be shown.

System action: Processing stops.

User response: Specify a valid filter argument.

CCQF029I More Db2 entries are associated with the specified product. All Db2 entries are listed.

System action: None.

User response: No action is required.

CCQF080I The customized jobs for the product that you are customizing are stored in this data set.

System action: None.

User response: No action is required.

CCQF081I The JCL must be browsed or edited.

Explanation:
You can either browse or edit the JCL.

System action: None.

User response: No action is required.

CCQF082E The sort-command command has an invalid sort field or order. The valid fields are list-of-column-names. The valid sort orders are A (for ascending) or D (for descending).

Explanation:
An invalid sort field or order was specified.

System action: Processing stops.

User response: Specify a valid sort field or order.

CCQF083E The sort-command command is missing a sort field.

Explanation:
A sort field must be specified.
The sort-command command has more than two sort fields specified.

Explanation:
The specified sort command included more than two sort fields. The sort command can have up to two fields specified.

System action:
Processing stops.

User response:
Specify only one or two sort fields.

A sort order was specified incorrectly in the sort-command command. A sort order can be specified only after a field name.

Explanation:
Valid orders are A (for ascending) or D (for descending).

System action:
Processing stops.

User response:
Specify a valid sort order after a field name.

The sort-command command has an invalid sort field. The valid fields are list-of-the-table-column-names.

Explanation:
An invalid sort field was specified.

System action:
Processing stops.

User response:
Specify a valid sort field.

The sort-command command has an invalid sort order. The valid orders are A (for ascending) or D (for descending).

Explanation:
An invalid sort order was specified.

System action:
Processing stops.

User response:
Specify a valid sort order.

No row match the specified filter argument. All rows are shown.

Explanation:
No rows match the selected values.

System action:
Processing stops.

User response:
Specify a matched value for filtering.

Type the search arguments to filter objects. A generic filter argument is a search argument of the form AA*.

Explanation:
In a generic filter argument, only the characters up to the asterisk (*) are compared. The * must be placed in the last nonblank position of the argument. Asterisks embedded in the argument are treated as data.

System action:
None.

User response:
Specify a valid sort order.

To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.

System action:
None.

User response:
No action is required.

To show the Products to Customize section, specify a slash (/). To hide the Products to Customize section, remove the slash. The Product to Customize section can be shown or hidden only on the Customizer Workplace panel.

System action:
None.

User response:
No action is required.

To show the Usage Notes section, specify a slash (/). To hide the Usage Notes section, remove the slash. The Usage Notes section can be shown only on the Product Parameters, LPAR Parameters, and DB2 Parameters panels.

System action:
None.

User response:
No action is required.
The specified values have been saved.

System action: None.
User response: No action is required.

Displays the Panel Display Options panel. Use this panel to select which information to display on panels.

System action: None.
User response: No action is required.

The fully qualified name of the data set into which you want to copy the current user profile. If the data set name exceeds 42 characters, enclose the name in quotation marks. ALTER or UPDATE authorization to this data set is required.

System action: None.
User response: No action is required.

The volume name in which the data set will reside. If left blank, the volume name will be decided by the system.

System action: None.
User response: No action is required.

To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.

System action: None.
User response: No action is required.

The specified option option_name is not valid.

Explanation: The option that was specified is not a valid option on the panel.
System action: Tools Customizer stops.
User response: Specify a valid option on the panel.

Before you customize a product, verify your user settings.

Explanation: The user settings must be verified before a product can be customized.
System action: Tools Customizer stops.
User response: Verify the user settings.

Check the user settings. One or more current values are not valid.

Explanation: One or more of the values in the user settings is not valid.
System action: Tools Customizer stops.
User response: Ensure that the specified values for the user settings are valid.

Before you use Tools Customizer, you must select option 0 to verify your user settings.

Explanation: The user settings must be changed before a product can be customized.
System action: Tools Customizer stops.
User response: Change the user settings.

You must select option 0 to change your user settings.

Explanation: User settings must be changed before a product can be customized.
System action: Tools Customizer stops.
User response: Change the user settings.

The XML structure of the member_name Db2 parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:
While determining if the Db2 parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

**System action:**
Processing continues.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

### CCQI001S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

**Explanation:**
While determining if the Db2 parameter metadata member is valid, the PL/I XML parser issued an exception error code.

**System action:**
Processing continues.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

### CCQI002S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *element_name* element is unknown.

**Explanation:**
The specified element in the Db2 parameter metadata member is unknown.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

### CCQI003S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

**Explanation:**
The specified element cannot contain content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

### CCQI004S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

**Explanation:**
The specified element requires content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

### CCQI005S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

### CCQI006S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.

**Explanation:**
The specified element does not contain enough characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

### CCQI007S
The XML structure of the *member_name* Db2 parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

**Explanation:**
The specified element does not occur enough times.

**System action:**
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCQI008S</td>
<td>The XML structure of the member_name Db2 parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The specified attribute occurs too many times.</td>
</tr>
<tr>
<td></td>
<td><strong>System action:</strong> Processing stops.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> See “Gathering diagnostic information” on page 286. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI009S</td>
<td>The XML structure of the member_name Db2 parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The specified attribute did not occur enough times.</td>
</tr>
<tr>
<td></td>
<td><strong>System action:</strong> Processing stops.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> See “Gathering diagnostic information” on page 286. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI010S</td>
<td>The XML structure of the member_name Db2 parameter metadata member is not valid. The attribute_name attribute in the element_name element, but content was found.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The specified attribute cannot have content.</td>
</tr>
<tr>
<td></td>
<td><strong>System action:</strong> Processing stops.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> See “Gathering diagnostic information” on page 286. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI012S</td>
<td>The XML structure of the member_name Db2 parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The specified element contains too many characters.</td>
</tr>
<tr>
<td></td>
<td><strong>System action:</strong> Processing stops.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> See “Gathering diagnostic information” on page 286. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI013S</td>
<td>The XML structure of the member_name Db2 parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The specified attribute in the Db2 parameter metadata member is unknown.</td>
</tr>
<tr>
<td></td>
<td><strong>System action:</strong> Processing stops.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> See “Gathering diagnostic information” on page 286. Contact IBM Software Support.</td>
</tr>
<tr>
<td>CCQI014S</td>
<td>The content of the member_name Db2 parameter metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The specified value of the element is not a valid value.</td>
</tr>
<tr>
<td></td>
<td><strong>System action:</strong> Processing stops.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> See “Gathering diagnostic information” on page 286. Contact IBM Software Support.</td>
</tr>
</tbody>
</table>
The content of the DB2 parameter metadata member is not valid because the value of the \texttt{attribute\_name} attribute in the \texttt{element\_name} element is incorrect. The value of the attribute is \texttt{value\_name}.

**Explanation:**
The specified value of the attribute is not a valid value.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The content of the DB2 parameter metadata member is not valid because the data type of the \texttt{element\_name} element is incorrect. The value is \texttt{value\_name}.

**Explanation:**
The specified data type is not a valid data type.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The content of the DB2 parameter metadata member was not found in the data set.

**Explanation:**
Tools Customizer could not find the specified Db2 parameter metadata member.

**System action:**
Processing stops.

**User response:**
Contact IBM Software Support.

The parameter \texttt{name} LPAR parameter in the template \texttt{name} template does not have associated metadata in the \texttt{member\_name} LPAR parameter metadata member.

**Explanation:**
The specified template does not contain metadata for an LPAR parameter. The name of the LPAR parameter metadata member, the name of the LPAR parameter, and the name of the template are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The parameter \texttt{name} product parameter in the template \texttt{name} template does not have associated metadata in the \texttt{member\_name} product parameter metadata member.

**Explanation:**
The specified template does not contain metadata for a product parameter. The name of the product parameter metadata member, the name of the product parameter, and the name of the template are indicated in the message text.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The following metadata data set was not found: \texttt{data\_set\_name}.

**Explanation:**
Tools Customizer could not find the specified metadata data set.

**System action:**
Processing stops.

**User response:**
Ensure that the metadata data set is specified correctly. If the problem persists, contact IBM Software Support.

The following metadata data set could not be opened: \texttt{data\_set\_name}.

**Explanation:**

See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
Tools Customizer could not open the specified LPAR metadata data set.

**System action:**
Processing stops.

**User response:**
Ensure the metadata data set was specified correctly.

**Explanation:**
Tools Customizer could not find the Db2 parameter metadata member in the specified Tools Customizer metadata data set.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**Explanation:**
Tools Customizer could not find the specified LPAR parameter metadata member.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**Explanation:**
The product parameter metadata member was not found in the specified data set.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**Explanation:**
Db2 parameters are not required to customize the specified product.

**System action:**
Processing continues.

**User response:**
No action is required.

**Explanation:**
LPAR parameters are not required to customize the specified product.

**System action:**
Processing continues.

**User response:**
No action is required.

**Explanation:**
Associated metadata is missing for the specified Db2 parameter in a task.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**Explanation:**
Associated metadata is missing for the specified LPAR parameter in a task.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**Explanation:**
Associated metadata is missing for the specified product parameter in a task.

**System action:**
Processing stops.
CCQI063S  The parameter_name Db2 parameter in the task_description task and the step_description step does not have associated metadata in the member_name Db2 parameter metadata member.

Explanation:  
Associated metadata is missing for the specified Db2 parameter in a task and step.

System action:  
Processing stops.

User response:  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI064S  The parameter_name LPAR parameter in the task_description task and the step_description step does not have associated metadata in the member_name LPAR parameter metadata member.

Explanation:  
Associated metadata is missing for the specified LPAR parameter in a task and step.

System action:  
Processing stops.

User response:  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI065S  The parameter_name product parameter in the task_description task and the step_description step does not have associated metadata in the member_name product parameter metadata member.

Explanation:  
Associated metadata is missing for the specified product parameter in a task and step.

System action:  
Processing stops.

User response:  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI066S  The parameter_name Db2 parameter in the task_description task, step_description step, and template_name template condition does not have associated metadata in the member_name Db2 parameter metadata member.

Explanation:  
Associated metadata is missing for the specified Db2 parameter in a task, step, and template.

System action:  
Processing stops.

User response:  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI067S  The parameter_name LPAR parameter in the task_description task, step_description step, and template_name template condition does not have associated metadata in the member_name LPAR parameter metadata member.

Explanation:  
Associated metadata is missing for the specified LPAR parameter in a task, step, and template.

System action:  
Processing stops.

User response:  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI068S  The parameter_name product parameter in the task_description task, step_description step, and template_name template condition does not have associated metadata in the member_name product parameter metadata member.

Explanation:  
Associated metadata is missing for the specified product parameter in a task, step, and template.

System action:  
Processing stops.

User response:  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI069S  Product metadata does not support multiple configurations, but the template_name product template contains the parameter_name parameter. Enable multiple configurations support for this product, and try again.

Explanation:  

The specified template contains a parameter for multiple configurations, but the product is not enabled to support multiple configurations.

**System action:**
Processing stops.

**User response:**
Enable multiple configurations support, and try again.

---

**CCQI070E**

The parameter_name Db2 parameter metadata member is not valid. The default length for the parameter-element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

**Explanation:**
The specified length cannot be shorter than the default length.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQI071E**

The parameter_name LPAR parameter metadata member is not valid. The default length for the parameter-element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

**Explanation:**
The specified length cannot be shorter than the default length.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQI072E**

The parameter_name product parameter metadata member is not valid. The default length for the parameter-element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

**Explanation:**
The specified length cannot be shorter than the default length.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
CCQI076S  The XML structure of the member_name Db2 parameter metadata member is not valid. The parameter_name parameter refers to the section-name section. This section was not found in the Db2 parameter metadata member.

Explanation:
The specified value already exists for an attribute.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI077S  The XML structure of the member_name LPAR parameter metadata member is not valid. The parameter_name parameter refers to the section-name section. This section was not found in the LPAR parameter metadata member.

Explanation:
The specified parameter refers to a section that is not in the LPAR parameter metadata member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI078S  The XML structure of the member_name product parameter metadata member is not valid. The parameter_name parameter refers to the section-name section. This section was not found in the product parameter metadata member.

Explanation:
The specified parameter refers to a section that is not in the product parameter metadata member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI080S  The content of the member_name Db2 parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation:
The specified value for an attribute in the Db2 parameter metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI081S  The content of the member_name LPAR parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation:
The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI082S  The content of the member_name product parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation:
The specified value for an attribute in the product parameter metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI088I  The command command is not active in BROWSE mode.

Explanation:
The specified command can be entered only in Edit mode.

System action:
Processing continues.

User response:
Put the panel in Edit mode and reissue the command.

CCQI089I The command command is already active.

System action: Processing continues.
User response: No action required.

CCQI090S The product-defined Db2 parameter parameter_name in the member_name parameter metadata member references the section_ID section ID, but this ID does not exist in either the parameter metadata member or the Db2 parameter metadata member.

Explanation: A section that does not exist in the parameter metadata member or the Db2 parameter metadata member is referenced by the specified Db2 parameter.

System action: Processing stops.

CCQI091S The product-defined LPAR parameter in the member_name parameter metadata member references the section_ID section ID, but this ID does not exist in either the parameter metadata member or the LPAR parameter metadata member.

Explanation: A section that does not exist in the parameter metadata member or the LPAR parameter metadata member is being referenced by the specified LPAR parameter.

System action: Processing stops.

CCQI092S The overridden DB2 parameter parameter_name in the member_name parameter metadata member does not exist in the Db2 parameter metadata member.

Explanation: The specified parameter does not exist.

System action: Processing stops.

CCQI093S The overridden LPAR parameter parameter_name in the member_name parameter metadata member does not exist in the LPAR parameter metadata member.

Explanation: The specified parameter does not exist.

System action: Processing stops.

CCQI094S The CCQ$$PRD product customization parameter metadata member was not found in the data_set_name data set.

Explanation: The specified data set must contain the CCQ$$PRD product customization parameter metadata member.

System action: Processing stops.

CCQI100W The XML structure of the member_name LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.
User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI101S The XML structure of the member_name LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: The specified parameter does not exist.
**Explanation:**
While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception error code.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI102S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. The element_name element is unknown.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified element in the LPAR parameter metadata member is unknown.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI103S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified element cannot contain content.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI104S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. Content is required for the element_name element, but content was not found.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified element requires content.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI105S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI106S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified element does not contain enough characters.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI107S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. The element_name element must occur at least minimum_number times.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified element does not occur enough times.

**System action:**
Processing stops.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

<table>
<thead>
<tr>
<th>CCQI108S</th>
<th>The XML structure of the member_name LPAR parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.</th>
</tr>
</thead>
</table>

**Explanation:**
The specified attribute occurs too many times.

**System action:**
Processing stops.
CCQI109S  The XML structure of the member_name LPAR parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI110S  The XML structure of the member_name LPAR parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI111S  The XML structure of the member_name LPAR parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI112S  The XML structure of the member_name LPAR parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI113S  The XML structure of the member_name LPAR parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute in the LPAR parameter metadata member is unknown.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI114S  The content of the member_name LPAR parameter metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an element in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI115S  The content of the member_name LPAR parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value of the attribute is value_name.

Explanation: The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response:
The content of the member_name LPAR parameter metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation:
The specified data type value for an element in the LPAR parameter metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI117S

The content of the member_name LPAR parameter metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation:
The specified data type value for an attribute in the LPAR parameter metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI116S

The content of the member_name LPAR parameter metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation:
The specified data type value for an element in the LPAR parameter metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI122S

The XML structure of the member_name parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI120S

The XML structure of the member_name Db2 parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI121S

The XML structure of the member_name LPAR parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI123S

The XML structure of the member_name discover metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation:
An element contains the specified duplicate value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI124S

The XML structure of the member_name product customization parameter metadata member is not valid. The element_name element in the parameter_name parameter contains duplicate values for the element_name element. The duplicate value is value_name.

Explanation:
An element contains the specified duplicate value.
System action: Processing stops.


CCQI200W The XML structure of the member_name information metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation: While determining if the information metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI201S The XML structure of the member_name information metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation: While determining if the information metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI202S The XML structure of the member_name information metadata member is not valid. The element_name element is unknown.

Explanation: The specified element in the information metadata member is unknown.

System action: Processing stops.


CCQI203S The XML structure of the member_name information metadata member is not valid.

Explanation: The specified element is not allowed for the element_name element, but content was found.

System action: The specified element cannot contain content.

User response: Processing stops.

CCQI204S The XML structure of the member_name information metadata member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.


CCQI205S The XML structure of the member_name information metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI206S The XML structure of the member_name information metadata member is not valid. The content length for the element_name element must be at least minimum_number characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI207S**  The XML structure of the
*member_name* information
metadata member is not valid. The
*element_name* element must occur
at least *minimum_number* times.

Explanation:
The specified element does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI208S**  The XML structure of the
*member_name* information
metadata member is not valid. The
*attribute_name* attribute in the
*element_name* element cannot
occur more than *maximum_number* times.

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI209S**  The XML structure of the
*member_name* information
metadata member is not valid. The
*attribute_name* attribute in the
*element_name* element must
occur at least *minimum_number* times.

Explanation:
The specified attribute did not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI210S**  The XML structure of the
*member_name* information
metadata member is not valid. Content
is not allowed for the
*attribute_name* attribute in the
*element_name* element, but
content was found.

Explanation:
The specified attribute cannot have content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI211S**  The XML structure of the
*member_name* information
metadata member is not valid. Content
is required for the
*attribute_name* attribute in the
*element_name* element, but
content was not found.

Explanation:
The specified attribute is missing required content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI212S**  The XML structure of the
*member_name* information
metadata member is not valid. The
*attribute_name* attribute in the
*element_name* element is
unknown.

Explanation:
The specified attribute in the information metadata
member is unknown.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI213S**  The XML structure of the
*member_name* information
metadata member is not valid. The
*element_name* element cannot
exceed *maximum_number*
characters.

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI214S**  The content of the
*member_name* information
metadata member is not valid because the value of the
*element_name* element is
incorrect. The value is value_name.

Explanation:
The specified value for an element in the information metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI215S  The content of the member_name information metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation:
The specified value for an attribute in the information metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI216S  The content of the member_name information metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation:
The specified data type value for an element in the information metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI217S  The content of the member_name information metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation:
The specified data type value for an attribute in the information metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI218S  The content of the member_name information metadata member is not valid. The length of the value_name value that of the attribute_name attribute is longer than the value_name value of the attribute_name attribute.

Explanation:
The first specified value cannot be longer than the second specified value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI219S  The content of the member_name information metadata member is not valid. The value_name value of the attribute_name attribute contains the value_name value.

Explanation:
The first specified value cannot be longer than the second specified value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI220S  The XML structure of the member_name information metadata member is not valid. Content for the attribute_name attribute in the element_name element exceed maximum_number characters.

Explanation:
The specified attribute contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI223S  The XML structure of the member_name information metadata member is not valid. The value that is specified for the Db2 Level already exists. The value is value_name.
Explanation:
The specified value already exists.

System action:
Processing stops.

User response:
Specify a different Db2 level. If the problem persists, contact IBM Software Support.

CCQI224S  The XML structure of the member_name information metadata member is not valid. The value that is specified for the Db2 Mode already exists. The value is value_name.

Explanation:
The specified value already exists.

System action:
Processing stops.

User response:
Specify a different Db2 mode. If the problem persists, contact IBM Software Support.

CCQI250S  The information metadata member was not found in the data_set_name data set.

Explanation:
Tools Customizer could not find the information metadata member in the specified data set.

System action:
Processing stops.

User response
If this message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is hlq.SINZDENU.

Do not specify the Tools Customizer metadata library, which is hlq.SCCQDENU.

If the problem persists, identify the name of the Tools Customizer trace data set and contact IBM Software Support.

CCQI251E  The member_name member was not accessible in the data_set_name data set.

Explanation:
The specified member could not be accessed in the data set.

System action:
Processing stops.

User response:
Specify the correct metadata library.

CCQI252S  The information metadata member was not found in the library_name component metadata library that is part of the library_name pack metadata library. The name of the pack is pack_name.

Explanation:
The specified component metadata library does not contain the information metadata member.

System action:
Processing stops.

User response:
Specify the correct metadata library.

CCQI253E  The library_name Tools Customizer metadata library is not current. Update the metadata library on the Tools Customizer Settings panel.

Explanation:
The specified metadata library is not current.

System action:
Processing stops.

User response:
Specify a current metadata library on the Tools Customizer Settings panel.

CCQI300W  The XML structure of the member_name sequence metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:
While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:
Processing continues.

User response:
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

CCQI301S  The XML structure of the member_name sequence metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation:
While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception error code.
**System action:**
Processing stops.

**User response:**
See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code, and contact IBM Software Support.

**CCQI302S**
The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element is unknown.

**Explanation:**
The specified element in the sequence metadata member is unknown.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI303S**
The XML structure of the *member_name* sequence metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

**Explanation:**
The specified element cannot contain content.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI304S**
The XML structure of the *member_name* sequence metadata member is not valid. Content is required for the *element_name* element, but content was not found.

**Explanation:**
The specified element is missing required content.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI305S**
The XML structure of the *member_name* sequence metadata member is not valid. Content length for the *element_name* element cannot exceed *maximum_number* characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI306S**
The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

**Explanation:**
The specified element occurs too many times.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI307S**
The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

**Explanation:**
The specified element does not occur enough times.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI308S**
The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

**Explanation:**
The specified attribute occurs too many times.

**System action:**
Processing stops.

**User response:**
See "Gathering diagnostic information" on page 286. Contact IBM Software Support.

**CCQI309S**
The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the
element_name element must occur at least minimum_number times.

Explanation:
The specified attribute does not occur enough times.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI310S The XML structure of the member_name sequence metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation:
The specified attribute cannot contain content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI311S The XML structure of the member_name sequence metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation:
The specified attribute is missing required content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI312S The XML structure of the member_name sequence metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI313S The XML structure of the member_name sequence metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation:
The specified attribute in the sequence metadata member is unknown.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI314S The content of the member_name sequence metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.

Explanation:
The specified value for an element in the sequence metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI315S The content of the member_name sequence metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation:
The specified value for an attribute in the sequence metadata member is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI316S The content of the member_name sequence metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation:
The specified data type value for an element in the sequence metadata member is not valid.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI317S**
The content of the `member_name` sequence metadata member is not valid because the data type of the `attribute_name` attribute in the `element_name` element is incorrect. The value is `value_name`.

**Explanation:**
The specified data type value for an attribute in the sequence metadata member is not valid.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI350S**
The XML structure of the `member_name` sequence metadata member is not valid because the value of the `attribute_name` attribute in the `element_name` element is incorrect. The value is `value_name`.

**Explanation:**
A specified value for an attribute in the sequence metadata member is not valid.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI351S**
The `member_name` sequence metadata member was not found in the `data_set_name` metadata data set.

**Explanation:**
Tools Customizer could not find the specified product template in the data set.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI352S**
The `template_name` product template was not found in the `data_set_name` metadata data set.

**Explanation:**
Tools Customizer could not find the specified product template in the data set.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI353S**
The sequence metadata member was not found in the `data_set_name` component data set that is part of the `data_set_name` pack.

**Explanation:**
Tools Customizer could not find the sequence metadata member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI354S**
The XML structure of the `member_name` sequence metadata member is not valid. The value of the `attribute_name` attribute in the `element_name` element already exists.

**Explanation:**
The specified attribute contains a value that already exists.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI355S**
The XML structure of the `member_name` sequence metadata member is not valid. The condition element on the `level_type` level already contains a relational operator.

**Explanation:**
A relational operator already exists for the condition element on the specified level.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI362S** The XML structure of the *member_name* sequence metadata member is not valid. The condition element on the *level_type* level must contain only one content string or content number element.

Explanation: Only one content string element or content number element can be contained in the condition element on the specified level.

System action: Processing stops.


**CCQI363S** The XML structure of the *member_name* sequence metadata member is not valid. The condition element in the *element_name* element with the *attribute_name* attribute must contain either the content string element or the content number element.

Explanation: Either the content string element or the content number element must be in the condition element.

System action: Processing stops.

User response: Contact IBM Software Support.

**CCQI400W** The XML structure of the *member_name* parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

**CCQI402S** The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified element in the parameter metadata member is unknown.

System action: Processing stops.


**CCQI403S** The XML structure of the *member_name* parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

**CCQI404S** The XML structure of the *member_name* parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.

CCQI405S  The XML structure of the member_name parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQI406S  The XML structure of the member_name parameter metadata member is not valid. The content length for the element_name element must be at least minimum_number characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.


CCQI407S  The XML structure of the member_name parameter metadata member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.


CCQI408S  The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.


CCQI409S  The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.


CCQI410S  The XML structure of the member_name parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.


CCQI411S  The XML structure of the member_name parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.


CCQI412S  The XML structure of the member_name parameter metadata member is not valid. The content length for the
**element_name** element cannot exceed **maximum_number** characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI413S** The XML structure of the **member_name** parameter metadata member is not valid. The **attribute_name** attribute in the **element_name** element is unknown.

**Explanation:**
The specified attribute in the parameter metadata member is unknown.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI414S** The content of the **member_name** parameter metadata member is not valid because the value of the **element_name** element is incorrect. The value is **value_name**.

**Explanation:**
The specified value for an element in the parameter metadata member is not valid.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI415S** The content of the **member_name** parameter metadata member is not valid because the value of the **attribute_name** attribute in the **element_name** element is incorrect. The value is **value_name**.

**Explanation:**
The specified value for an attribute in the parameter metadata member is not valid.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI420S** The XML structure of the **member_name** parameter metadata member is not valid. The **element_name** element is unknown for the overridden Db2 parameter.

**Explanation:**

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI421S** The XML structure of the **member_name** parameter metadata member is not valid. The **element_name** element is unknown for the overridden LPAR parameter.

**Explanation:**

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI422S
The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown for the overridden Db2 parameter.

Explanation:

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI423S
The XML structure of the member_name parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown for the overridden LPAR parameter.

Explanation:

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI450S
The member_name product parameter metadata member was not found in the data_set_name data set.

Explanation:
Tools Customizer could not find the specified product parameter metadata member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI510W
The data_set_name data store data set does not exist.

Explanation:
The specified data store data set does not exist.

System action:
Processing continues.

User response:
Ensure that the data store data set exists.

CCQI511S
The data_set_name data store data set cannot be opened by using the disposition_type disposition.

Explanation:
The specified data store data set could not be opened with the specified disposition.

System action:
Processing continues.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI512S
The data_set_name data store data set cannot be opened by using the option-type option.

Explanation:
The specified data store data set was unable to be opened with the specified option.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

CCQI600W
The XML structure of the member_name product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:
While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:
Processing continues.

User response:
See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQI601S
The XML structure of the member_name product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation:
While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception error code.

**System action:**
Processing continues.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

**CCQI602S** The XML structure of the member_name product customization parameter metadata member is not valid. The element_name element is unknown.

**Explanation:**
The specified product customization parameter metadata member contains an unknown element.

**System action:**
Processing continues.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

**CCQI603S** The XML structure of the member_name product customization parameter metadata member is not valid. The content for the element_name element was found.

**Explanation:**
Content was found in an element that cannot contain content.

**System action:**
Processing continues.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

**CCQI604S** The XML structure of the member_name product customization parameter metadata member is not valid. The content for the element_name element was not found.

**Explanation:**
The specified element does not contain required content.

**System action:**
Processing continues.

**User response:**
See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

**CCQI605S** The XML structure of the member_name product customization parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

**Explanation:**
The specified element contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI606S** The XML structure of the member_name product customization parameter metadata member is not valid. The element_name element cannot occur more than maximum_number times.

**Explanation:**
The specified element occurs too many times in the product customization parameter metadata member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI607S** The XML structure of the member_name product customization parameter metadata member is not valid. The element_name element must occur at least minimum_number times.

**Explanation:**
The specified element does not occur enough times in the product customization parameter metadata member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI608S** The XML structure of the member_name product customization parameter metadata member is not valid. The
**Explanation:**
The specified attribute occurs too many times in the product customization parameter metadata member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI609S**

The XML structure of the member_name product customization parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

**Explanation:**
The specified attribute occurs too many times in the product customization parameter metadata member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI610S**

The XML structure of the member_name product customization parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

**Explanation:**
The specified attribute does not occur enough times in the product customization parameter metadata member.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI611S**

The XML structure of the member_name product customization parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

**Explanation:**
Content was found in an element that cannot contain content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI612S**

The XML structure of the member_name product customization parameter metadata member is not valid. The content length for the attribute_name attribute in the element_name element cannot exceed maximum_number characters.

**Explanation:**
The specified attribute contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI613S**

The XML structure of the member_name product customization parameter metadata member is not valid. The value of the element_name element is unknown.

**Explanation:**
The specified product customization parameter metadata member contains an unknown attribute.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQI614S**

The XML structure of the member_name product customization parameter metadata member is not valid. The value of the element_name element is not valid. The value value_name.

**Explanation:**
The specified value of the element is not a valid value.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI615S The XML structure of the member_name product customization parameter metadata member is not valid. The value of the attribute_name attribute for the element_name element is not valid. The value is value_name.

Explanation:
The specified value of the attribute is not a valid value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI616S The XML structure of the member_name product customization parameter metadata member is not valid. The data type of the element_name element is not valid. The value of the element is value_name.

Explanation:
The specified data type is not a valid data type.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI617S The XML structure of the member_name product customization parameter metadata member is not valid. The data type of the attribute_name attribute for the element_name element is not valid. The value of the attribute is value_name.

Explanation:
The specified data type is not a valid data type.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI650S The XML structure of the member_name product customization parameter metadata member is not valid. The following value of the attribute_name attribute in the element_name element already exists: value_name.

Explanation:
The specified value for an attribute already exists.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI651S The XML structure of the member_name product customization parameter metadata member is not valid. The parameter_name parameter refers to the following section, which was not found in the member_name product customization parameter metadata member: section-name.

Explanation:
The specified section is not in the product customization parameter metadata member.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI652S The member_name product customization metadata member not valid. The default length for the element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

Explanation:
The specified length cannot be shorter than the default length.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI653S The content of the member_name product customization parameter metadata member is not valid. The value of the attribute_name attribute in the element_name

Explanation:
The specified value of the attribute is not a valid value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
element is not valid. The value of the attribute is value_name.

Explanation:
The specified value of the attribute is not a valid value.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI700W The XML structure of the member_name solution pack metadata member is not valid. The PL/I XML parser issued the following exception warning code: code_number.

Explanation:
While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception warning code.

System action:
Processing continues.

User response:
See the Enterprise PL/I for z/OS Programming Guide for more information about the warning.

CCQI701S The XML structure of the member_name solution pack metadata member is not valid. The PL/I XML parser issued the following exception error code: code_number.

Explanation:
While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception error code.

System action:
Processing stops.

User response:
See the Enterprise PL/I for z/OS Programming Guide for more information about the error.

CCQI702S The XML structure of the member_name solution pack metadata member is not valid. The element_name element is unknown.

Explanation:
The specified solution pack metadata member contains an unknown element.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI703S The XML structure of the member_name solution pack metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation:
Content was found in an element that cannot contain content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI704S The XML structure of the member_name solution pack metadata member is not valid. Content is required for the element_name element, but content was not found.

Explanation:
The specified element does not contain required content.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI705S The XML structure of the member_name solution pack metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation:
The specified element contains too many characters.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQI706S The XML structure of the member_name solution pack metadata member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation:
The specified element occurs too many times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI707S**

The XML structure of the `member_name` solution pack metadata member is not valid. The `element_name` element must occur at least `minimum_number` times.

**Explanation:**
The specified element does not occur enough times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI708S**

The XML structure of the `member_name` solution pack metadata member is not valid. The `attribute_name` attribute in the `element_name` element cannot occur more than `maximum_number` times.

**Explanation:**
The specified attribute occurs too many times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI709S**

The XML structure of the `member_name` solution pack metadata member is not valid. The `attribute_name` attribute in the `element_name` element must occur at least `minimum_number` times.

**Explanation:**
The specified attribute does not occur enough times.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI710S**

The XML structure of the `member_name` solution pack metadata member is not valid. Content is not allowed for the `attribute_name` attribute in the `element_name` element, but content was found.

**Explanation:**
The specified attribute cannot have content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI711S**

The XML structure of the `member_name` solution pack metadata member is not valid. Content is required for the `attribute_name` attribute in the `element_name` element, but content was not found.

**Explanation:**
The specified attribute is missing content.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI712S**

The XML structure of the `member_name` solution pack metadata member is not valid. The content length for the `attribute_name` attribute in the `element_name` element cannot exceed `maximum_number` characters.

**Explanation:**
The specified attribute contains too many characters.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

---

**CCQI713S**

The XML structure of the `member_name` solution pack metadata member is not valid. The `attribute_name` attribute in the `element_name` element is unknown.

**Explanation:**
The specified attribute in the solution pack metadata member is unknown.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI714S**  The XML structure of the **member_name** solution pack metadata member is not valid because the value of the **element_name** element is incorrect. The value is **value_name**.

**Explanation:**
The specified value of the element is not a valid value.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI715S**  The XML structure of the **member_name** solution pack metadata member is not valid because the value of the **attribute_name** attribute in the **element_name** element is incorrect. The value of the attribute is **value_name**.

**Explanation:**
The specified value of the attribute is not a valid value.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI716S**  The XML structure of the **member_name** solution pack metadata member is not valid because the data type of the **element_name** element is incorrect. The value is **value_name**.

**Explanation:**
The specified data type is not a valid data type.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI717S**  The XML structure of the **member_name** solution pack metadata member is not valid because the data type of the **attribute_name** attribute in the **element_name** element is incorrect. The value of the attribute is **value_name**.

**Explanation:**
The specified data type is not a valid data type.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI720S**  The XML structure of the **member_name** solution pack metadata member is not valid. The **msg** element is required for the **component_name** component that is not customizable.

**Explanation:**
The **msg** element is required for the specified component, which cannot be customized by using Tools Customizer.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI750S**  The solution pack metadata member was not found in the **library_name** metadata library.

**Explanation:**
Tools Customizer could not find the solution pack metadata member in the specified library.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI751S**  The version in the **library_name** solution pack metadata library is different than the version in the **library_name** component metadata library. The name of the pack is **pack_name**, and the name of the component is **component_name**.

**Explanation:**
The version in the solution pack metadata library does not match the version in the component metadata library.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286.
Contact IBM Software Support.

**CCQI752S**  
The release in the *library_name* solution pack metadata library is different than the release in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.

**Explanation:**  
The release in the solution pack metadata library does not match the release in the component metadata library.

**System action:**  
Processing stops.

**User response:**  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQI753S**  
The modification level in the *library_name* solution pack metadata library is different than the modification level in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.

**Explanation:**  
The modification level in the solution pack metadata library does not match the modification level in the component metadata library.

**System action:**  
Processing stops.

**User response:**  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

---

**CCQI755S**  
The XML structure of the *member_name* discover parameter metadata member is not valid. When a default value is not specified in the metadata member, the "required" attribute with a value of "true" cannot be specified.

**System action:**  
Processing stops.

**User response:**  
Contact IBM Software Support.

---

**CCQM002E**  
The *command_name* line command is not valid: .

**Explanation:**  
The specified line command is not valid.

**System action:**  
Processing continues.

**User response:**  
Specify a valid line command on the panel.

---

**CCQ0000W**  
The XML structure of the *member_name* discover parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

**Explanation:**  
While determining if the discover parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

**System action:**  
Processing continues.

**User response:**  
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQ0001S**  
The XML structure of the *member_name* discover parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

**Explanation:**  
While determining if the Discover metadata member is valid, the PL/I XML parser issued an exception error code.

**System action:**  
Processing stops.

**User response:**  
See the Enterprise PL/I for z/OS Programming Guide for more information about the exception warning code.

---

**CCQ0002S**  
The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element is unknown.

**Explanation:**  
The specified element in the discover parameter metadata member is unknown.

**System action:**  
Processing stops.

**User response:**  
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
CCQ0003S  The XML structure of the member_name discover parameter metadata member is not valid. Content is not allowed for the element_name element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.


CCQ0004S  The XML structure of the member_name discover parameter metadata member is not valid. Content is required for the element_name element, but content was not found.

Explanation: The specified element is missing required content.

System action: Processing stops.


CCQ0005S  The XML structure of the member_name discover parameter metadata member is not valid. The content length for the element_name element cannot exceed maximum_number characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.


CCQ0006S  The XML structure of the member_name discover parameter metadata member is not valid. The element_name element cannot occur more than maximum_number times.

Explanation: The specified element occurs too many times.

System action: Processing stops.


CCQ0007S  The XML structure of the member_name discover parameter metadata member is not valid. The element_name element must occur at least minimum_number times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.


CCQ0008S  The XML structure of the member_name discover parameter metadata member is not valid. The attribute_name attribute in the element_name element cannot occur more than maximum_number times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.


CCQ0009S  The XML structure of the member_name discover parameter metadata member is not valid. The attribute_name attribute in the element_name element must occur at least minimum_number times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.


CCQ0010S  The XML structure of the member_name discover parameter metadata member is not valid. Content is not allowed for the attribute_name attribute in the element_name element, but content was found.

Explanation: The specified attribute cannot contain content.

CCQ0011S  The XML structure of the member_name discover parameter metadata member is not valid. Content is required for the attribute_name attribute in the element_name element, but content was not found.

Explanation: The specified attribute requires content.

System action: Processing stops.

CCQ0012S  The XML structure of the member_name discover parameter metadata member is not valid. The content length for the attribute_name attribute in the element_name element in the cannot exceed maximum_number characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

CCQ0013S  The XML structure of the member_name discover parameter metadata member is not valid. The attribute_name attribute in the element_name element is unknown.

Explanation: The specified attribute is unknown.

System action: Processing stops.

CCQ0014S  The content of the member_name discover parameter metadata member is not valid because the value of the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an element in the discover parameter metadata member is not valid.

System action: Processing stops.

CCQ0015S  The content of the member_name discover parameter metadata member is not valid because the value of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified value for an attribute in the discover parameter metadata member is not valid.

System action: Processing stops.

CCQ0016S  The content of the member_name discover parameter metadata member is not valid because the data type of the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an element in the discover parameter metadata member is not valid.

System action: Processing stops.

CCQ0017S  The content of the member_name product parameter metadata member is not valid because the data type of the attribute_name attribute in the element_name element is incorrect. The value is value_name.

Explanation: The specified data type value for an attribute in the product parameter metadata member is not valid.

System action: Processing stops.
User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQO050S The data_set_name Discover REXX EXEC data set could not be initialized or was not found.

Explanation:
Tools Customizer could not find or could not initialize the specified Discover REXX EXEC data set.

System action:
Processing stops.

User response:
Ensure that the Discover REXX EXEC is specified correctly.

CCQO051W The data_sharing_group_ID data sharing group ID cannot contain more than four characters.

Explanation:
The specified data sharing group ID contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified data sharing group ID does not exceed four characters.

CCQO052S The REXX_EXEC_name Discover REXX EXEC was not found in the data_set_name Discover data set.

Explanation:
Tools Customizer could not find the Discover REXX EXEC in the specified data set.

System action:
Processing stops.

User response:
Ensure that the Discover data set was specified correctly.

CCQO053W The LPAR_name LPAR name cannot contain more than eight characters.

Explanation:
The specified LPAR name contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified LPAR name does not exceed eight characters.

CCQO054W The subsystem_ID Db2 SSID cannot contain more than four characters. The record was not processed.

Explanation:
The specified Db2 SSID contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified Db2 SSID does not exceed four characters.

CCQO055W The parameter_name Db2 group attach name parameter is in the record_name Discover record, but a Db2 group attach name was not specified. The record was not processed.

Explanation:
The Discover record contains a data sharing group parameter, but a Db2 group attach name was not specified.

System action:
Processing continues.

User response:
Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO056W The parameter_name Db2 parameter in the record_name Discover record did not have a Db2 group attach name or a Db2 SSID. The record was not processed.

Explanation:
The Discover record did not have a Db2 group attach name or a Db2 subsystem ID in the Db2 parameter.

System action:
Processing continues.

User response:
Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO057W The Discover EXEC could not find the parameter_name parameter in the metadata for the product to be customized. The record was not processed.

Explanation:
The specified parameter could not be found in the metadata for the product to be customized.

System action:
Processing continues.

User response:
Ensure that information is specified correctly on the Discover Customized Product Information panel.
The parameter name in the record type Discover record does not start with CCQ_LPR_, CCQ_DB2_, or CCQ_PRD_. The record was not processed.

Explanation:
The parameter in the record does not start with CCQ_DB2_, CCQ_LPAR_, or CCQ_PRD_.

System action:
Processing continues.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The parameter name parameter cannot contain more than 72 characters. The record was not processed.

Explanation:
The specified product parameter contains too many characters.

System action:
Processing continues.

User response:
Ensure that the specified product parameter does not exceed 72 characters.

The Discover record from the REXX EXEC output must start with the following record type: record_type. The record was not processed.

Explanation:
A Discover record from the REXX EXEC output must start with the specified Db2 record type.

System action:
Processing continues.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

If you do not have a previously customized version of the product, do not run the Discover EXEC. Press END to go to the Customizer Workplace panel.

Explanation:
This message is issued when you customize a product for the first time. It prompts you to use the Discover EXEC to discover data from a previous customization of the specified product.

System action:
Processing continues.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The Discover EXEC could not find the following parameter_name parameter in the Db2 metadata. The record was not processed.

Explanation:
The specified parameter is missing in the Db2 metadata.

System action:
Processing continues.

User response:
If this parameter is required, contact IBM Software Support.

The Discover-record Discover record did not have a parameter name. The record was not processed.

Explanation:
A parameter name was missing in the Discover record.

System action:
Processing continues.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

The value for the parameter_name parameter is ignored because it has more than maximum_number characters, which is the maximum length that is defined in the metadata. The value is parameter_value.

Explanation:
The specified value exceeded the maximum allowed length, which was defined in the metadata. Tools Customizer truncated the extra characters.

System action:
Processing continues.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.
CCQ0066W  The record_name Discover record from the Discover REXX EXEC output does not have a parameter value. The record was not processed.

**Explanation:**
The Discover record was missing a parameter value from the Discover EXEC output.

**System action:**
Processing continues.

**User response:**
Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQ0067W  The parameter_name parameter is defined in the metadata to support one value, but more than one value was found. The last value was used.

**Explanation:**
The definition of the parameter in the metadata supports one value, but more than one value was specified. Only the last value was used.

**System action:**
Processing continues.

**User response:**
Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQ0068W  The value of the parameter_name parameter is ignored because the parameter is defined as internal=true. The value is value_name.

**Explanation:**
The specified value of the parameter is ignored because it is defined as internal=true.

**System action:**
Processing continues.

**User response:**
Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQ0069W  The Discover EXEC did not find the parameter_name parameter in the LPAR metadata. The record was not processed.

**Explanation:**
The specified parameter is missing from the LPAR metadata.

**System action:**
Processing continues.

**User response:**
Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQ0070W  The record_type Discover record contains an incorrect delimiter between the Environment section and the Data section. The record was not processed.

**Explanation:**
Tools Customizer found an incorrect delimiter between the Environment section and the Data section.

**System action:**
None.

**User response:**
No action is required.

CCQ0071W  The member_name member could not be found in the data_set_name Discover data set.

**Explanation:**
Tools Customizer could not find the specified Discover data set.

**System action:**
None.

**User response:**
No action is required.

CCQ0072S  The member_name discover metadata member was not found in the data_set_name metadata data set.

**Explanation:**
Tools Customizer could not find the specified metadata member in the data set.

**System action:**
Processing stops.

**User response:**
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQ0073E  The member_name discover metadata member is not valid because the default length for the element_name parameter element exceeds the length of the parameter. The default length is default_length, and the specified length is specified_length. The default length will be truncated accordingly.

**Explanation:**
The default length for the specified parameter element is longer than the parameter.
System action:
Processing continues.

User response:
No action is required.

**CCQO074S**
The content of the `member_name` discover metadata member is not valid. The value of the `attribute_name` attribute in the `element_name` element is not valid. The value of the attribute is `value_name`.

Explanation:
The specified value is not valid.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

**CCQO075W**
The `configuration_ID` configuration ID in the `record_name` Discover record is incorrect. The record was not processed.

Explanation:
The specified configuration ID is not correct.

System action:
Processing continues.

User response:
No action is required.

**CCQO076W**
The `configuration_ID` configuration ID cannot contain more than `maximum_number` characters. The record was not processed.

Explanation:
The specified configuration ID contains too many characters.

System action:
Processing continues.

User response:
No action is required.

**CCQO077S**
The discover metadata member was not found in the `data_set_name` component data set that is part of the `data_set_name` pack.

Explanation:
The discover metadata member was not found in the specified component data set.

System action:
Processing stops.

**CCQO078I**
Additional configurations were discovered and saved in the data store. All Db2 entries associated with this configuration are listed.

System action:
None.

User response:
No action is required.

**CCQO080I**
Product `name` does not support the Discover process.

Explanation:
The specified product does not support the Discover process.

System action:
None.

User response:
No action is required.

**CCQP000E**
The value of the `mode_name` Db2 mode is not valid for the `level_name` Db2 level.

Explanation:
The specified Db2 mode is not valid for the Db2 level.

System action:
Processing stops.

User response:
Specify a valid Db2 mode for the Db2 level.

**CCQP001E**
The value of the `mode_name` Db2 mode is missing.

Explanation:
The specified Db2 mode is not defined.

System action:
Processing stops.

User response:
Specify a value for the Db2 mode.

**CCQP002E**
The value of the `mode_name` Db2 level is missing.

Explanation:
The specified Db2 level is not defined.

System action:
Processing stops.

User response:
Specify a value for the Db2 level.

**CCQP003E**
The value of the `level_name` Db2 level is not valid.
Explanation:
The specified Db2 level does not have a valid name.

System action:
Processing stops.

User response:
Specify a valid value for the Db2 level.

CCQP004S  The parameter_name parameter does not exist in the CCQ$DB2 Db2 parameter metadata member.

Explanation:
The CCQ$DB2 Db2 parameter metadata member does not contain the specified parameter.

System action:
Processing stops.

User response:
See “Gathering diagnostic information” on page 286. Contact IBM Software Support.

CCQP005E  The value of the subsystem_ID Db2 SSID is missing.

Explanation:
The specified Db2 SSID is not defined.

System action:
Processing stops.

User response:
Specify a valid value for the Db2 SSID.

CCQP006E  The value of the group_attach_name Db2 group attach name is missing.

Explanation:
The specified Db2 group attach name is not defined.

System action:
Processing stops.

User response:
Specify a valid Db2 group attach name.

CCQQ000E  Specify a valid metadata library. Each qualifier of the library must start with an alphabetic character and must be 1-8 alphanumeric characters. The library name must be 1-44 characters.

Explanation:
The metadata library was not specified in the correct format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action:
Tools Customizer prompts for the correct library name.

User response:
Specify a valid metadata library. If the message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is hlq.SINZDENU. Do not specify the Tools Customizer metadata library, which is hlq.SCCQDENU.

CCQQ001E  The data_set_name data set name that was specified for the metadata library was not found.

Explanation:
The data set does not exist, or the data set name was written in the incorrect format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action:
Tools Customizer prompts for the correct data set name.

User response:
Specify a data set name in the correct format.

CCQQ002E  The data set name that was specified for the library_name metadata library cannot be opened.

Explanation:
Tools Customizer could not open the data set.

System action:
Tools Customizer prompts for an available data set.

User response:
Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ003E  The data_set_name data set name that was specified for the metadata sample library is not valid. The data set must be in the following format: HLQ.SxxxxSAMP.

Explanation:
The specified data set name was not specified in the correct format.

System action:
None.

User response:
Specify the data set name in the following format: HLQ.SxxxxSAMP, where xxx is the three-character prefix for the product.

CCQQ004E  The data_set_name data set is being used by another user. Try
again when the data set is not being used.

Explanation:
Another user is using the specified data set.

System action:
None.

User response:
Ensure that the specified data set is not being used.

CCQQ009E
The data set name data set name that was specified for the metadata library is not valid because the data set is empty.

Explanation:
The specified data set is empty.

System action:
Tools Customizer prompts for an available data set.

User response:
Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ011E
The library name metadata library for the component that is part of the library name pack was not found in the catalog. The name of the pack is pack name, and the name of the component is component name.

Explanation:
The specified metadata library is not in the catalog.

System action:
None.

User response:
Specify another metadata library.

CCQQ012E
The library name metadata library for the component that is part of the library name pack cannot be opened.

Explanation:
The specified metadata library cannot be opened.

System action:
None.

User response:
Ensure that the name of the library is specified correctly.

CCQS000I
Tools Customizer is being invoked for the first time or the previous ISPF session ended before Tools Customizer was exited. In both cases, the fields on this panel are populated with default values. Review these default values or specify new values to be used to customize products or packs.

Explanation:
When you customize a stand-alone product or a solution pack for the first time, or when an ISPF session unexpectedly ends before the ISPF profile is saved, you must specify or review your Tools Customizer user settings.

System action:
Processing stops.

User response:
Review and accept the default settings, or specify new settings.

CCQS001E
The following command is not valid: command name.

Explanation:
The specified command is not a valid command on the panel.

System action:
Processing stops.

User response:
Specify a valid command.

CCQS002W
The data set name Discover data set could not be found.

Explanation:
Tools Customizer could not find the specified data set.

System action:
The data set will be allocated, and processing continues.

User response:
Ensure that the data set name is specified correctly because the data set will be allocated with this name after the values are saved.

CCQS003W
The data set name Discover data set was not found so it was created.

Explanation:
Tools Customizer could not find the specified data set.

System action:
Processing continues.

User response:
Ensure that the data set name is specified correctly.

CCQS004I
The settings were saved.

Explanation:
The settings that you changed were saved.

System action:
Processing continues.

User response:
No action is required.

CCQS006W The length of a qualifier for the data_set_name customization library data set exceeds 26 characters.

Explanation:
The qualifier for the customization library data set is too long. The qualifier cannot exceed 26 characters.

System action:
Processing continues.

User response:
Specify a qualifier that is 26 characters or less.

CCQS007E The discover data set data_set_name could not be opened with the option-type option.

Explanation:
The specified option could not open the Discover data set.

System action:
None.

User response:
Specify a data set to which you have WRITE access.

CCQS008E The Discover data set data_set_name exists on a different volume.

Explanation:
The specified Discover data set must exist on the same volume as where it was created.

System action:
Processing continues.

User response:
Specify a different Discover data set name.

CCQS010E The customization library qualifier is not valid.

Explanation:
The customization library qualifier that was specified is not valid.

System action:
None.

User response:
Specify a valid qualifier for the customization library.

CCQS011E The group attach option is not valid.

Explanation:
The group attach option that was specified is not valid.

System action:
None.

User response:
Specify a valid option for the group attach option.

CCQS012E The Tools Customizer metadata library is not valid.

Explanation:
The metadata library that was specified is not a valid data set.

System action:
None.

User response:
Specify a valid data set for the metadata library.

CCQS013E The Discover data set is not valid.

Explanation:
The Discover data set that was specified is not a valid data set.

System action:
None.

User response:
Specify a valid Discover data set.

CCQS014E The data store data set is not valid.

Explanation:
The data set that was specified is not a valid data set.

System action:
None.

User response:
Specify a valid data store data set.

CCQS015E Tools Customizer is already running.

Explanation:
A session of Tools Customizer is already running in your environment. Only one Tools Customizer session is allowed.

System action:
None.

User response:
The trace data set is being used. Free the trace data set, and start Tools Customizer again.

CCQS018E Information on the first line of the job card exceeds 57 characters.

Explanation:
The first line of the job card can contain only 57 characters. This character limit includes a continuation character.

System action:
Tools Customizer clears the first line of the job card.

User response:
Specify information that does not exceed 57 characters on the first line of the job card.

**CCQS019E**  
The required trace data set, data_set_name, is currently not accessible.

**Explanation:**  
The trace data set must be accessible.

**System action:**  
Processing stops.

**User response:**  
Ensure that the trace data set is accessible.

**CCQS020E**  
An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

**Explanation:**  
To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

**System action:**  
None.

**User response:**  
Ensure that ALTER authority for the specified customization library data set is granted.

**CCQS021E**  
The value value_name in the field that contains the cursor position is not valid.

**Explanation:**  
The specified value is not valid.

**System action:**  
None.

**User response:**  
Specify a valid value.

**CCQS022E**  
An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

**Explanation:**  
To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

**System action:**  
None.

**User response:**  
Ensure that UPDATE authority for the specified customization library data set is granted.

**CCQS023E**  
An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

**Explanation:**  
To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

**System action:**  
None.

**User response:**  
Ensure that UPDATE authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

**CCQS024E**  
An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

**Explanation:**  
To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

**System action:**  
None.

**User response:**  
Ensure that ALTER authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

**CCQS025I**  
The display options were saved.

**System action:**  
None.

**User response:**  
No action is required.

**CCQS026E**  
The customization library data set data_set_name could not be opened because the requester does not have UPDATE authority on this data set.

**Explanation:**  
Users must have UPDATE authority to open the customization library data set. Users must have UPDATE authority to open the customization library data set.
System action:
None.

User response:
Ensure that UPDATE authority for the specified customization library data set is granted or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS027E The customization library data set data_set_name could not be created because the requester does not have ALTER authority on this data set.

Explanation:
To create the customization library data set, ALTER authority on the data set must be granted.

System action:
Processing stops.

User response:
Ensure that ALTER authority for the specific customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS029E The customization library data set is not valid. Enter a valid data set name or use the Tools Customizer default: data_set_name.

Explanation:
The specified data set is invalid.

System action:
Processing stops.

User response:
Specify a valid data set name.

CCQS030E The following command is not a valid CREATE statement: command_statement.

Explanation:
The specified CREATE command statement is invalid because it contains blanks or alphabetic characters.

System action:
Processing stops.

User response:
Specify a valid CREATE command statement. The correct syntax is CREATE nn, where nn is 1 - 99.

CCQS031E The following command is not a valid CREATE statement: command_statement. The number that can be specified with the CREATE command is 1 - 99.

Explanation:
The specified CREATE command statement is invalid because it contains either 0 or a number greater than 99.

System action:
Processing stops.

User response:
Specify a valid CREATE command statement. The correct syntax is CREATE nn, where nn is 1 - 99.

CCQS033E A user profile cannot be copied into the same user profile.

Explanation:
The specified data set cannot be copied into user's own user profile.

System action:
Processing stops.

User response:
Enter a different data set name.

CCQS034E The shared user profile data set data_set_name could not be created because the requester does not have UPDATE authority on this data set or because the data set already exists in another volume serial.

Explanation:
To create a shared user profile data set, the requester must have update authority on the data set, and the specified data set name must be unique.

System action:
Processing stops.

User response:
Ensure that the requester has UPDATE authority on the data set and ensure that the data set name is unique.

CCQS035E The specified data set already has a user profile. Specify a different data set, or press Enter again to replace the existing user profile.

Explanation:
Pressing Enter overwrites the previous user profile for the specified data set with user's own user profile.

System action:
Processing stops.

User response:
Specify a different data set name.

CCQS036E The customization library data_set_name already exists in volume and cannot be created in a
different volume. Enter a different customization library name.

Explanation:
The same data set name cannot exist in a different volume.

System action:
Processing stops.

User response:
Specify a different data set name.

CCQS037E The data set name was either not specified or invalid.

Explanation:
The data set name specified does not follow the IBM data set name convention.

System action:
Processing stops.

User response:
Specify a valid data set name.

CCQS038E The specified data set cannot be used.

Explanation:
The specified data sets contain information that supports Tools Customizer, but this data set cannot be used.

System action:
Processing stops.

User response:
Specify a different data set.

CCQS039E The specified data set has an invalid record format.

Explanation:
The specified data set should be saved as a different record format. For example, the record format should be FB (Formated Block) but it is set to VB (Variable Block).

System action:
Processing stops.

User response:
Specify a valid record format.

CCQT000I The product configuration ID configuration_ID was successfully copied from configuration_ID.

Explanation:
The specified configuration ID was copied.

System action:
None.

User response:

CCQT001E The command_name line command was specified more than once, which is not allowed.

Explanation:
The specified line command cannot be specified more than one time.

System action:
Processing stops.

User response:
Specify the line command only once.

CCQT002E The configuration_ID configuration_ID already exists. Specify a different configuration ID.

Explanation:
The specified configuration ID exists.

System action:
Processing stops.

User response:
Ensure that the specified configuration ID is unique.

CCQT003I The product configuration ID configuration_ID was created.

Explanation:
The specified configuration ID was created.

System action:
None.

User response:
No action is required.

CCQT004I The product configuration ID configuration_ID was removed.

Explanation:
The specified configuration ID was removed.

System action:
None.

User response:
No action is required.

CCQT005E The product configuration ID configuration_ID is not valid. The product configuration ID cannot contain a colon (:).

Explanation:
The specified configuration ID contains a colon (:), but a colon is not valid.

System action:
Processing stops.

User response:
Specify a configuration ID that does not contain a colon.
The configuration ID configuration ID exists. Specify a different configuration ID.

**Explanation:**
The specified configuration ID exists.

**System action:**
Processing stops.

**User response:**
Specify another configuration ID.

The configuration ID configuration ID exists but was removed from the list of configurations. To use this configuration ID, you must restore it.

**Explanation:**
The specified configuration ID exists but was removed from the list of available configuration.

**System action:**
Processing stops.

**User response:**
Specify another configuration ID. To restore the specified configuration ID, issue the CREATE command, and specify the same configuration ID again.

The configuration ID configuration ID exceeds maximum_number characters.

**Explanation:**
The specified configuration ID contains too many characters.

**System action:**
Processing stops.

**User response:**
Specify another configuration ID that does not exceed the maximum number of characters that was set by Db2 HPU.

Create request for configuration ID configuration was cancelled by user.

**Explanation:**
The request to create the specified configuration was canceled.

**System action:**
Processing stops.

**User response:**
No action is required.

The configuration ID configuration was not copied.

**System action:**
Processing stops.

**User response:**
No action is required.

The configuration ID configuration was not removed.

**Explanation:**
The specified configuration was not removed.

**System action:**
Processing stops.

**User response:**
No action is required.

None of the configurations were copied or removed. All of the previously selected configurations are deselected.

**Explanation:**
The selected configurations were not copied or removed, and they are deselected.

**System action:**
Processing stops.

**User response:**
No action is required.

Specify Y or N and press Enter to continue, or press End to cancel.

**Explanation:**
A function requires input.

**System action:**
Processing stops.

**User response:**
To continue, specify Y or N and press Enter. Otherwise, press End to cancel.

The command_name command is not allowed during the process of "Select" configuration line command.

**Explanation:**
The specified command is not allowed while the line command for selecting configurations is processing.

**System action:**
Processing stops.

**User response:**
Remove the specified line command.

The configuration ID configuration was not created.

**Explanation:**
The specified configuration was not created.
CCQT017I The configuration_ID configuration was not copied.

Explanation:
The specified configuration was not copied.

System action:
Processing stops.

User response:
No action is required.

CCQT018E Specify Y or N, and press Enter.

Explanation:
A function requires input.

System action:
Processing stops.

User response:
To continue, specify Y or N, and press Enter.

CCQT019I The select configuration_ID configuration process ended.

Explanation:
The select process for the specified configuration is finished.

System action:
Processing stops.

User response:
No action is required.

CCQT020E The configuration_ID configuration was not created because the data store was not accessible.

Explanation:
The specified configuration was not created because the data store could not be accessed.

System action:
Processing stops.

User response:
Ensure that the data store is accessible and create the configuration again.

CCQT021E The configuration_ID configuration was not copied because the data store was not accessible.

Explanation:
The specified configuration was not copied because the data store could not be accessed.

System action:
Processing stops.

User response:
Ensure that the data store is accessible and copy the configuration again.

CCQT025I The configuration_ID configuration was not updated.

Explanation:
The specified configuration was not updated because the edit process was canceled.

System action:
Processing stops.

User response:
No action is required.

CCQT027I The product configuration ID has been updated from edit_from_id to edit_to_id.

System action:
Processing continues.

User response:
No action is required.

CCQT028I The product configuration ID has been updated from edit_from_id to edit_to_id, and the description has been updated from edit_from_des to edit_to_des.

System action:
Processing continues.

User response:
No action is required.

CCQT029I The product configuration description has been updated from edit_from_des to edit_to_des.

System action:
Processing continues.

User response:
No action is required.

CCQX001S Product_name has already been customized by using values from data_set_name data store data set. Switch to the specified data store data set to continue customizing this product.

Explanation:
The specified product was customized by using values from the specified data store data set.

System action:
Processing stops.

User response:
Use the specified data store data set to continue customizing the product.
**CCQX002S**  
*component_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this component.

**Explanation:**
The specified component was customized by using values from the specified data store data set.

**System action:**
Processing stops.

**User response:**
Use the specified data store data set to continue customizing the component.

**CCQX011I**  
*Product_name* was not found.

**Explanation:**
The specified product was not found.

**System action:**
Processing stops.

**User response:**
Specify another product.

**Db2 HPU user abend codes**

This section contains a list of all of the abends that are issued by Db2 HPU.

**1003**
Db2 HPU was unable to open a sequential file. Correct the JCL to include the missing DD statement, and resubmit the job. This situation might occur when unloading data from image copies (COPYDDN option) of a partitioned table space stored on tape if some of the image copy data set are stored on the same volume. In such case, system messages requesting the same volume be mounted should be found in the system log, such as in the example below:

```
14.55.19 JOB09615*246 IEF455D MOUNT BRES19 ON 0581 FOR MZLFRDI UNLOAD OR REPLY
14.55.45 JOB09615 IEF234K 0581.BRES19,PVT,MZLFRDI,UNLOAD
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ FROM DDNAME IC001
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ FROM DDNAME IC002
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ FROM DDNAME IC003
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ FROM DDNAME IC004
14.55.45 JOB09615 IEF233D M 0581,BRES19,,MZLFRDI;UNLOAD, 939
939
LABO.F1P.DBTL000.FRD9D216.D13031.T1612444,
939 OR RESPOND TO IEF455D MESSAGE
14.55.45 JOB 09615*247 IEF455D MOUNT BRES19 ON 0581 FOR MZLFRDI UNLOAD OR REPLY
14.56.09 JOB 09615 IEA995I SYMPTOM DUMP OUTPUT 942
942 USER COMPLETION CODE=1003
942 TIME=14.56.07 SEQ=24286 CPU=0000
ASID=00D6
942 PSW AT TIME OF ERROR 078D1000 804BD3EE ILC 2 INTC 0D
942 ACTIVE LOAD MODULE
ADDRESS=000070000 OFFSET=004B
942 NAME=INZUTILK
942 DATAAT PSW 004BD3E8 - 00181610 0A0D9640 D26A0700
942 AR/GR 0: 00000000/80000000 1: 00000000/800003EB
942 2: 00000000/00606D0 3: 00000000/1254A8B0
942 4: 00000000/12386008 5: 00000000/12540000
942 6: 00000000/1254A548 7: 00000000/0000001
942 8: 00000000/1254A930 9: 00000000/12706000
942 A: 00000000/12541088 B: 00000000/004BCF58
942 C: 00000000/004BDF58 D: 00000000/0059F00
```
If you identify such a situation, disable the parallelism by adding an option such as MAXPART 1 PARALLELISM (,1).

1004
Db2 HPU attempted to allocate or open a VSAM file. The attempt was unsuccessful because an allocation or open failure occurred (see the register 15 return code).

See DFSMS Macro Instructions for Data Sets for z/OS for a complete description of the possible return codes and reason codes.

1005
An error occurred while accessing a VSAM file (see the register 15 return code).

See DFSMS Macro Instructions for Data Sets for z/OS for a complete description of the possible return codes and reason codes.

1006
A VSAM CLOSE failed (output). See the output messages.

1009
A return code greater than 4 was returned from an MVS sort. Error messages from sort are printed in the UTPRINT data set. Determine the cause of the error. In most cases, error messages from MVS sort identify the problem, and a dump is not needed. If the sort is unable to open the UTPRINT data set, only the messages and the abend are provided. For example, if the UTPRINT DD statement is omitted, add a UTPRINT DD statement in your JCL to obtain detailed error messages for analyzing the error.

1010
Db2 HPU detected that an internal subtask did not terminate at the end of processing. A possible cause might be that the DD cards do not correspond to the control statements. See the LOG messages. Contact IBM Software Support, if necessary.

1012
Db2 HPU was unable to decompress a row. Check the table space after restore.

1013
Db2 HPU detected an error in EDITPROC. Check the libraries and the table space.

1015
An attempt was made to perform an unload, but the ddname for the IMAGE COPY file was not in the JCL.

1017
A sub-task abended, which resulted in a general abend. Examine the error messages that were issued for the original abend (for example, B37 issued by a SORT utility).

1019
Db2 HPU detected that one of the WHERE clauses was too complex. Possible causes are that one of the keywords is not supported or is not compatible with SQL syntax.

1021
When register R15 is 0, the input image copy contains rows with several versions, and the active table version is 0. An input image copy that has rows with several versions means the table was altered before the image copy was taken and was not reorganized or reloaded in the meantime. When the active table version is 0, it has not been altered since it was created. When register R15 is not 0, an unexpected system page sequence was found.

If register R15 is 0, Db2 HPU cannot use the image copy as input. Select a copy that was run after the table space was reorganized.

If register R15 is 12, this means that Db2 HPU has found an unexpected page number (the page that was read was not the expected one). This control is done if CHECK_INTEGRITY LEVEL1 was requested. In such a case:

• R2 contains the expected page number
• R3 contains the number of the page that has been read
This can occur if the image copy to unload has one of the following types:

- inline image copy
- FlashCopy
- a copy created with the COPYTOCOPY utility
- is actually a concatenation of several Incremental Image Copies and an Full Image Copy

and no option allowing Db2 HPU to be aware of the specific type of the input image copy was specified. These options are:

- INLINE
- FROM_COPYTOCOPY
- CHECK
- FLASHCOPY
- MERGE_IIC

See the documentation of these options to determine which is relevant.

For any other value of R15, ensure the image copy is valid by creating an object and populating it with the image copy. You can use DSN1COPY. If the image copy is valid, contact IBM Software Support.

1022
Db2 HPU detected a problem with the version description (OBDREC) of a record from an input image copy or an input table space data set (LDS) which prevents the data from this record to be unloaded. Register R15 contains the version number of that record. Db2 HPU stops with this ABEND when one of the following situation is encountered:

- The version that is indicated by register R15 is not described in the image copy. The image copy might have been created with the SYSTEMPAGES NO option.
- When the image copy was created, the table had been altered \( n \) times (there are \( n \) records versions in the image copy), and when the table was unloaded, the table had been altered only \( m \) times, where \( m < n \) (the Db2 catalog only describes \( m \) records versions).
- When the image copy was created before a REORG that materializes a DROP COLUMN, use the DDLDDN option to provide Db2 HPU with a file that contains the DDL of the table as before the involved DROP COLUMN to unload the date from the image copy.
- The version of a record read from the table space data set (LDS) is not described in the table space data set; In such case, the INZX135 message is issued before Db2 HPU stops with the ABENDU1022. Refer to the explanation of this message.

If none of these situations occurred, contact IBM Software Support and provide the job log for a new run of the unload that failed and the related SYSABEND file.

When the image copy was created before a REORG that materializes a DROP COLUMN, use the DDLDDN option to provide Db2 HPU with a file that contains the DDL of the table as before the involved DROP COLUMN to unload the data from the image copy.

1024
Db2 HPU was unable to start. The reason code (register 15) is the code returned by the macro ATTACH. Contact IBM Software Support to get help on how to determine the root cause of the problem.

11xx
For abend codes that are not described below, Db2 HPU detected an internal error. Contact IBM Software Support.

1128
Db2 HPU has detected an internal error. The following reason code is issued:
A DATE TIME involved in a scalar expression needs to be converted by a DATE EXIT and this EXIT returned an error. Check that the data contained in the table is correct. If so, contact IBM Software Support.

If reason code 0F is associated with this abend, Db2 HPU detected an unsupported situation due to binary zeroes (i.e., x'00') in char columns to be unloaded. This unusual situation prevents an optimization mechanism to work correctly.

Check if your data may contain binary zeroes. If this is an abnormal content, fix the issue before running the unload job again. If this is a normal content, consider adding FIX(OPTCNV) to the ULOPTNS parameter to the Db2 HPU parmlib.

In any other case (i.e., either the reason code different from 0F or the reason code is 0F but either the above explanation or suggested solution does not apply), this might be an internal error. Contact IBM Software Support.

Db2 HPU has encountered an error when unloading LOB or XML in native mode.

If the reason code is among 5, 8, 15, 20, 22, 23, 24 and 25, Db2 HPU has encountered the following situation. For a specific row, the unload of LOB or XML data in native mode failed because of an inconsistency between the base table space and the LOB or XML related table space. This situation can occur if the consistency of these table spaces has not been ensured by having the Db2 buffer flushed to the underlying data sets (e.g., by specifying the QUIESCE YES option in the UNLOAD command) and no concurrent update of the object occurs during the unload (e.g., by specifying the LOCK YES option in the UNLOAD command). Submit the job again after having ensured of the table spaces consistency or run the unload in SQL mode by specifying the DB2 FORCE option. You can request Db2 HPU to issue a message and stop or to ignore such a situation and resume the processing by respectively setting the VUU084/ULLOBBHV to STOP or IGNORE.

For any other reason code or if the actions suggested above have not solved the problem, contact your IBM support.

If reason code 0A is associated with this abend, Db2 HPU detected an error while writing a LOB file into its target library. Db2 HPU received an error notification from the STOW macro (SVC21), and the registers are set to the following specifications:

R14
The return code of STOW (R15 from STOW)
The reason code of STOW (R0 from STOW)

The address of the member name (LOB file) that could not be created

The address of the library name for which the error occurred

This abend can be issued for the following reasons:

- The target library has run out of space (R14=16 and R2=2871=x'B37', or R2=3383=x'D37').
- Directory space for the target library does not exist (R14=12 and R2=0).
- The member (LOB file) to be created already exists in the target library (R14=4 and R2=0).

See *DFSMS Macro Instructions for Data Sets for z/OS* for a complete description of the possible return codes and reason codes.

If a reason code other than 0A is associated with this abend, Db2 HPU detected an internal error. Contact IBM Software Support.

1135

Db2 HPU has encountered an error when unloading LOB or XML in native mode.

If the reason code is 2, Db2 HPU has encountered the following situation. For a specific row, the unload of LOB or XML data in native mode failed because of an inconsistency between the base table space and the LOB or XML related table space. This situation can occur if the consistency of these table spaces has not been ensured by having the Db2 buffer flushed to the underlying data sets (e.g. by specifying the QUIESCE YES option in the UNLOAD command) and no concurrent update of the object occurs during the unload (e.g. by specifying the LOCK YES option in the UNLOAD command). Submit the job again after having ensured of the table spaces consistency or run the unload in SQL mode by specifying the DB2 FORCE option. You can request Db2 HPU to issue a message and stop or to ignore such a situation and resume the processing by respectively setting the VUU084/ULLOBBHV to STOP or IGNORE.

If the reason code is among 4 and D, increase the region size and submit again the JCL.

For any other reason code or if the actions suggested above have not solved the problem, contact your IBM support.

1136

Db2 HPU has encountered an error when unloading LOB or XML in native mode.

If the reason code is among 2, 4, 15, 1A, 29, 2B, 2D, 47, 48 and 4B, Db2 HPU has encountered the following situation. For a specific row, the unload of LOB or XML data in native mode failed because of an inconsistency between the base table space and the LOB or XML related table space. This situation can occur if the consistency of these table spaces has not been ensured by having the Db2 buffer flushed to the underlying data sets (e.g. by specifying the QUIESCE YES option in the UNLOAD command) and no concurrent update of the object occurs during the unload (e.g. by specifying the LOCK YES option in the UNLOAD command). Submit the job again after having ensured of the table spaces consistency or run the unload in SQL mode by specifying the DB2 FORCE option. You can request Db2 HPU to issue a message and stop or to ignore such a situation and resume the processing by respectively setting the VUU084/ULLOBBHV to STOP or IGNORE.

For any other reason code or if the actions suggested above have not solved the problem, contact your IBM support.

1137

An error occurred while Db2 Sort was running. The following reason code is issued:

2

The requested number of sort operations is incorrect because it exceeds the maximum number of sort operations that can be run in parallel or the requested number is out of range. See register R3 for the requested number of sort operations. If R3 contains a positive, small number, consider...
decreasing the parallelism degree that is requested by your unload process. If R3 contains an irrelevant value, such as a negative number or a positive number that is too large to be the accurate number of required sort operations, contact IBM Software Support.

If a reason code other than 2 is associated with this abend, Db2 HPU detected an internal error. Contact IBM Software Support.

11xx
For abend codes that were not previously described, Db2 HPU detected an internal error. Contact IBM Software Support.

12xx
Db2 HPU detected an internal error. Contact IBM Software Support.

20xx
Db2 HPU detected an internal error. Contact IBM Software Support.

3001-3005
Db2 HPU detected an internal error. Contact IBM Software Support.

4000
Expected information was not provided by the LISTCAT command. The following WTO is issued prior to the 4000 abend:

```
INZUT4-n : ERROR DURING LISTCAT FOR dsname
```

where `dsname` is the name of the data set for which the LISTCAT command was issued.

If `n`=1: the data set was located but no data set entry was found (information field length is equal to 0).

If `n`=2: for a CLUSTER component, the AMDSB control block could not be retrieved. The AMDSB control block contains the type of the data set (HURBA or HARBA).

Issue a manual LISTCAT command to determine the error.

40xx
Db2 HPU detected an internal error. Contact IBM Software Support.

**Db2 HPU messages**

This section contains a list of all of the messages that are issued by Db2 HPU.

Each Db2 HPU is accompanied by one of the following return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Informational. No error occurred.</td>
</tr>
<tr>
<td>4</td>
<td>Warning. A condition was detected; you might need to take further action.</td>
</tr>
<tr>
<td>8 or 16</td>
<td>Error. An error was detected; processing could not continue.</td>
</tr>
</tbody>
</table>

The return code shown for a message is the condition code from the job step in which the message is issued. If additional messages that have higher condition codes are issued during the same job step, the higher condition code is returned.

The message text uses the short variable name, such as Vxxnnnn, or the short parameter name. Use these short names to find descriptions about them in this user’s guide.

**INZC002E**

**FATAL ERROR WHILE CHECKING DDNAME ddname (RC=reason code)**

**User response:**
Contact IBM Software Support.

**Return Code:**
8

**Explanation:**
A unrecoverable error occurred when checking the ddname that was provided.
INZC003E  ERROR DDNAME ddname IS NOT ALLOCATED AND/OR NOT ROOT OF ANY ALLOCATED DDNAMES

Explanation:
The specified ddname is not allocated in the execution JCL.

User response:
Correct the SYSIN or the DD statements that are allocated in the JCL and resubmit the job.

Return Code: 8

INZC004E  INCOMPLETE OBJECT DEFINITION FOR TABLESPACE dbname.tsname message

Explanation:
The definition of the table space is incomplete. The partitioning index is missing.

User response:
Create the partitioning index.

Return Code: 8

INZC005E  IMAGE COPIES MUST BE SPECIFIED BY PARTITION, INVALID ALLOCATED DDNAME : ddname

Explanation:
Image copy files should be specified per partition for a partitioned table space.

User response:
Allocate a file per partition for image copy files.

Return Code: 8

INZC008E  UNBALANCED QUOTE

Explanation:
An unbalanced quotation mark is found in a file (SYSIN, TEMPLATE, LISTDEF, or DDL). Refer to the previous message to see which file contains the unbalanced quotation mark.

User response:
Review the file and correct the unbalanced quotation mark.

Return Code: 8

INZC019E  ERROR DDNAME ddname NOT ALLOCATED AND NO TEMPLATE FOUND

Explanation:
The indicated ddname used in the SYSIN does not fulfill any of the following:
- It is allocated via a DD card in the JCL
- It is the root name (prefix) of a ddname or of a set of ddname starting with the mentioned ddname appended with a numerical suffix allocated via JCL
- It is described by a TEMPLATE statement

Db2 HPU cannot process the unload command properly as an input or output data set associated to the indicated DD name is not allocated either by using a DD card or a TEMPLATE statement.

User response:
Check the JCL and the SYSIN and correct the problem.

Return Code: 8

INZC020E  stmt1 STATEMENT CANNOT BE MIXED WITHIN stmt2 STATEMENT

Explanation:
Incompatible statements were specified in the SYSIN. For example, UNLOAD PLUS cannot be used within a Db2 HPU UNLOAD statement.

User response:
Check the JCL and the SYSIN and correct the incompatibility.

Return Code: 8

INZC021E  LISTDEF list CANNOT BE FOUND

Explanation:
A LISTDEF or LISTDEFTBV that was specified in the SYSIN cannot be found. The LISTDEF or LISTDEFTBV definition is searched in the first place in the SYSIN, and then in the LISTDEF or LISTDEFTBV LIBRARY.

User response:
Verify that a LISTDEF or LISTDEFTBV definition exists in the SYSIN or the LISTDEF or LISTDEFTBV LIBRARY.

Return Code: 8

INZC022E  LISTDEF NAME list IS TOO LONG
**Explanation:**
A LISTDEF or LISTDEFTBV name that is longer than 18 characters is specified in the SYSIN.

**User response**
Change the length of the LISTDEF or LISTDEFTBV name to be 18 characters or less.

**Return Code**
8

**INZCE0322**  ERROR: THE MODULE INZUTILB IS NOT APF AUTHORIZED

**Explanation**
Db2 HPU determined that the INZUTILB module was not APF authorized.

**User response**
Provide the necessary authorization for the specified module, and resubmit the job.

**Return Code**
8

**INZCE0559**  ERROR: UNABLE TO LOAD DSNHDECP FROM THE DSNEXIT LIBRARY

**Explanation**
Db2 HPU was unable to load the Db2 installation options from the DSNEXIT LIBRARY.

**User response**
Check the value of the VZD007 installation parameter for the specified Db2 subsystem.

**Return Code**
8

**INZCI0559**  DDNAME: *ddname* ROWS: *number*

**Explanation**
The message is printed to the *ddname* HPUPRINT, if allocated, to inform about the number of unloaded rows per *ddname*.

**User response**
None.

**INZDSC01I**  The *nnn* value of VZD014 is ignored because it is ambiguous for the subsystem-ID SSID.

**Explanation**
Db2 HPU V4.1 and earlier releases did not require any details for Db2 V9 NFM and later releases. For such Db2 subsystems, VZD014 had the same value (i.e. 915). In Db2 HPU V4.2 and later releases, the exact Db2 level must be specified. This message is issued because the Discover process in Tools Customizer found the ambiguous 915 value and has not generated any value. The related value needs to be specified by using Tools Customizer.

**User response:**
No action is required.

**Return Code**
None.

**INZDSC02I**  The *nnn* value of VZD014 is not recognized for the subsystem-ID SSID.

**Explanation:**
The value that was discovered for the VZD014 is not valid. The value was not stored by the Discover process. The related value needs to be specified by using Tools Customizer.

**User response:**
No action is required.

**Return Code**
None.

**INZDSC03I**  The value-name value of Variable-ID contains a colon (:).

**Explanation:**
The specified value of the indicated variable contains a colon, which might cause incorrect processing in Tools Customizer. The value is kept as specified, and if incorrect processing occurs, Tools Customizer will issue the CCQO070W message.

**User response:**
No action is required.

**Return Code**
None.

**INZDSC07E**  INTERNAL ERROR, UNEXPECTED ERROR NUMBER *n*

**Explanation:**
An unexpected error has occurred

**User response:**
Contact IBM Software support.

**Return Code**
8

**INZDSC08E** ENTER VALUE EITHER IN FIELD 'NAME OF PARMLIB LIBRARY...' EITHER IN BOTH FIELDS 'PREVIOUS DATA SET...'

**Explanation:**
The discover process requires either the "name of the PARMLIB library of a previous version (4.2.0 and later) or both the name of the "Previous data set of variables to be retrieved (4.1.0 and earlier)" and the name of the "Previous data set with customized file dsnames (4.1.0 and earlier)". At least one piece of information has not been entered.

**User response:**
Fill in the fields where the cursor is located.

**Return Code**
8

**INZDSC09I** THE VALUE IN FIELDS 'PREVIOUS DATA SET...' IS IGNORED WHEN A VALUE IS SPECIFIED FOR 'NAME OF PARMLIB LIBRARY...'

**Explanation:**
Values have been entered for both a discover from a release 4.2.0 and later and a release "4.1.0 and earlier". Only the values from the later version (4.2.0 and later) are taken into account. The values entered in the fields "4.1.0 and earlier" are ignored.

**User response:**
None

**Return Code**
None

**INZDSC10E** DATASET NAME IS NOT THE NAME OF A PDS/PDSE

**Explanation:**
A non-PDS or PDSE data set name has been entered in the field "Name of PARMLIB library of a previous version (4.2.0 and above) " where a PDS or PDSE is required.

**User response:**
Correct the file "Name of PARMLIB library of a previous version (4.2.0 and above)".

**Return Code**
8

**INZDSC11E** INCONSISTENT Db2 HPU PARMLIB

**User response:**
The information for the discover exec process is inconsistent for the indicated reason.

**User response:**
Enter consistent information.

**Return Code**
8

**INZDSC12I** VARIABLE NOT RECOGNIZED: variable-name

**Explanation:**
An unknown variable has been read from the setting from which the discover process is done. The related line is ignored and the discover process continues.

**User response:**
None.

**Return Code**
None.

**INZDSC13E** DISCOVER VERSION current-vrm CANNOT WORK WITH AN OLD PARMLIB FROM VERSION previous-vrm.

**Explanation:**
The former settings provided as an input to the discover utility comes from a Db2 HPU version. The discover process cannot be done.

**User response:**
Provide the parmlib of a version not later than the current one.

**Return Code**
8

**INZDSC14I** VALUE INSTALL-SYSADM/INSTALL_SYSOPR IS NO LONGER ACCEPTED FOR DB2 PARAMETER VARIABLE-NAME FOR SSID DB0A. THE VALUE HAS TO BE CHANGED BEFORE TO GENERATE JCLs.

**Explanation:**
The value I INSTALL-SYSADM/INSTALL_SYSOPR formerly allowed for the indicated variable is no longer supported and must be changed after the discover process

**User response:**
Modify the setting for the indicated fvalue after the discover process is completed.

**Return Code**
None.
**INZDSC15I**  THE PREVIOUS PARMLIB HAS NOT BEEN BUILT THROUGH TCz. CHECK THE VALUE FOR PARAMETER VARIABLE-NAME

**Explanation:**
The data set entered in the “Name of the PARMLIB library of a previous version (4.2.0 and later)” actually comes from a version earlier than 4.2.0. Although the discover process works, the value for the indicated variable might not be the expected one.

**User response:**
Either check the value for the indicated variable or preferably rerun the discover process after having specified the data set name in the relevant field, i.e. "Previous data set variables to be retrieved (4.1.0 and earlier)".

**Return Code:**
None.

**INZGE00nn**  ERROR : WRONG nn PARAMETER SPECIFIED IN THE PARMLIB details

**Explanation**
An invalid value was specified in the PARMLIB for the specified parameter. The values for nn and details are:

- **nn**
  - **details**
  - **00** Refer to the line that follows this message.
  - **01** The variable is missing.
  - **02** The variable must be numeric.
  - **03** The variable must be an 8-digit number.

**User response**
Modify the parameter value, and resubmit the job.

**Return Code**
8

**INZGE0004**  ERROR: WRONG VALUE SPECIFIED FOR VUM028/DISPLUSR IN THE INZTVAR MEMBER, ITS LENGTH IS GREATER THAN THE MAXIMUM LENGTH FOR THIS PARAMETER

**Explanation**
Db2 HPU detected a length error in the PARMLIB for the VUM028/DISPLUSR parameter.

**User response**
Correct the length of the value that was specified for the VUM028/DISPLUSR parameter in the PARMLIB.

**Return Code**
8

**INZGE0102**  INTERNAL ERROR : FILE file, LINE line. PLEASE CONTACT YOUR TECHNICAL SUPPORT

**Explanation**
Db2 HPU detected an internal error.

**User response**
Contact IBM Software Support, and supply the return and reason codes.

**Return Code**
8

**INZGE0200**  ERROR : ddname DDCARD SHOULD BE LINE A FILE WITH RECFM=VBS

**Explanation**
The mentioned ddname does not relate to a file with the SPANNED format while this format is requested to process the unload request. Db2 HPU detected an internal error.

**User response**
Change the allocation parameters for the mentioned ddname by specifying the VBS RECFM or consider specifying DFSIGDCB YES to allow Db2 HPU to override the DCB of the provided output file and set its RECFM to VBS.

**Return Code**
ABENDU1003

**INZI006E**  NO CONVERSION WAS AVAILABLE BETWEEN CCSID ccsid AND CCSID ccsid

**Explanation**
A conversion could not be performed because no conversion service was available to perform the specified conversion.
Ensure that the appropriate conversion services are online and that the conversion for the specified ccsid is available.

Return Code
8

**INZI007I**
A CONVERSION WAS REQUESTED BETWEEN CCSID ccsid AND CCSID ccsid

Explanation
This is an informational message. Db2 HPU requested a conversion between the specified CCSIDs.

User response
No action is required.

Return Code
0

**INZI008E**
A CONVERSION FAILED BETWEEN CCSID ccsid source AND CCSID ccsid target

Explanation:
A character was found in the source string that cannot be converted into the CCSID target, and the PARMLIB parameter VZM008/SCUNSUB was set to NO.

User response:
Set the PARMLIB parameter VZUM028/SCUNSUB to YES to allow substitution character in conversions.

Return Code
8

**INZI009E**
CUNLCNV ERROR: RC n, REASON p CONVERSION BETWEEN CCSID ccsid1 AND CCSID ccsid2

Explanation
An error occurred when using Conversion Services to convert a string. The return code and reason code were returned from IBM Conversion Services.

User response
Check the return and reason codes in z/OS Support for Unicode: Using Conversion Services.

Return Code
8
The message displays the value of some parameters from the settings of the Db2 subsystem the utility is run against. The message text is formatted as follows:

- **DB2 SUB SYSTEM**: the subsystem identifier (DSNHDECP field: SSID if a single subsystem is involved - expected value is a valid subsystem name). When the subsystem is part of a data sharing group, the group attachment name <group> (DSNHDECP field: SSID if a member of a data sharing group is involved - expected value is a valid group attachment name) is also indicated.

- **DB2 CATALOG VERSION**: the version of the DB2 CATALOG and not exactly the Db2 version. For example, if a migration from V9 NFM to V10 CM has been attempted and a fallback to V9 has been done, the DB2 CATALOG VERSION field still displays V10 CM despite the fallback. A discrepancy between the DB2 CATALOG VERSION and the DB2 VERSION also occurs when only some of the members in a Db2 data sharing system have been migrated and some others still run the old version.

- **DB2 APPLCOMPAT**: displays the value of the Db2 application compatibility at the time Db2 HPU is run if applicable, i.e. from Db2 V11 NFM, and N/A otherwise. This value applies before Db2 HPU has taken any APPLCOMPAT option or APPLCOMPAT setting into account.

- **DSNEXIT**: name of the library that Db2 HPU considers as the DSNEXIT library, i.e. the library that contains the DSNHDECP load module.

- **DECIMAL POINT**: the character used as decimal point (DSNHDECP field: DECIMAL - expected values are PERIOD|COMMA)

- **SQL STRING DELIMITER**: the SQL string delimiter (DSNHDECP field: SQLDELI - expected values are DEFAULT|QUOTE|APOST)

- **MIXED**: mixed attribute of the subsystem (DSNHDECP field: MIXED - expected values are YES|NO)

- **ENCODING SCHEME**: subsystem encoding scheme (DSNHDECP field: ENSHEME- expected values are EBCDIC|ASCII|UNICODE)

- **EBCDIC CCSID**: EBCDIC CCSID (DSNHDECP fields: SCSCSID,MCCSID,GCCSID - expected values are valid CCSID values)

- **ASCII CCSID**: ASCII CCSID (DSNHDECP fields: ASCSCSID,AMCCSID,AGCCSID - expected values are valid CCSID values)

- **DATE FORMAT**: date format (DSNHDECP fields: DATE - expected values are ISO|USA|EUR|JIS|LOCAL)

- **TIME FORMAT**: time format (DSNHDECP fields: TIME - expected values are ISO|USA|EUR|JIS|LOCAL)

- **DECIMAL ARITHMETIC**: rules that are to be used when both operands in a decimal operation have precisions of 15 or less (DSNHDECP field: DECARTH - expected values are DEC15|DEC31)

**User response:**

No action is required. However, after the setting for a given Db2 has been completed, consider checking that the values displayed by UDBI020I/INZI020I are consistent with the Db2 subsystem settings. Also, if an unexpected behavior such as an unexpected format for the date or time fields in the output of an unload is encountered.

**Reason code**

None.

THE FOLLOWING USS DIRECTORY DOES NOT EXIST: directory_name.

**Explanation**

A directory must exist before it can be used in a template.

**User response**

Create the target directory before you reference it in a template.

**Return Code**

8

ERROR: DDNAME ddname NOT ALLOCATED

**Explanation**

An internal error occurred. Db2 HPU has determined that the specified ddname was not allocated.
Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZI143E  ERROR: DURING \textit{ddname} ALLOCATION

Explanation
Db2 HPU was unable to allocate a temporary data set under \textit{ddname}. If the \textit{ddname} is SYSIN, an extra blank line might be coded after the end of the in-stream SYSIN data set.

User response
Ensure that the parameters for the VUM018/WRKVOL PARMLIB parameter, the VUA007/WRKTUNIT PARMLIB parameter, or both are valid. Also, ensure that all the volumes that these parameters refer to have not run out of free space. Resubmit the job when the VUM018/WRKVOL PARMLIB parameter, the VUA007/WRKTUNIT PARMLIB parameter, or both refer to a set of existing volumes with enough free space.

If the \textit{ddname} is SYSIN and the in-stream SYSIN is followed by an extra blank line, remove the blank line, and resubmit the job.

If the problem persists, contact IBM Software Support, and supply the return codes and reason codes.

Return Code
8

INZI144E  ERROR: DURING reallocation of \textit{ddname} ddname

Explanation
Db2 HPU was unable to deallocate the ddname \textit{ddname}.

User response
Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZI146E  ERROR WHEN SUBMITTING A SQL QUERY TO DB2 ON PLAN planname

Explanation
Db2 cannot reply to an SQL query.

User response
Examine the Db2 messages in the SYSPRINT file. See \textit{Db2 Universal Database for z/OS Messages and Codes} for a description of the possible return codes and reason codes.

Return Code
8

INZI147E  ERROR WHEN CLOSING UTILITY PLAN, ERROR CODE: \textit{nn} REASON CODE: \textit{nn}

Explanation
A Db2 CAF closing error occurred.

User response
See the \textit{Db2 for z/OS Application Programming and SQL Guide} for CAF return code and reason code information.

Return Code
8
INZI148E  ERROR WHEN DISCONNECTING FROM DB2, ERROR CODE: nn  
REASON CODE: nn

Explanation:
Db2 HPU encountered an error while disconnecting from Db2.

User response
See the Db2 for z/OS Application Programming and SQL Guide for CAF return code and reason code information.

Return Code
8

INZI149S  DB2 CONNECTION ERROR, SSID: ssid, ERROR CODE: nn  
REASON CODE: nn

Explanation:
A Db2 CAF connection attempt failed with the indicated Db2 return code and reason code.

User response
Determine the reason of the error and its solution by checking the meaning of the indicated Db2 return and reason codes from the Db2 codes documentation.

Return Code
16

INZI150S  DB2 OPEN PLAN ERROR, PLAN: planname, ERROR CODE: nn  
REASON CODE: nn

Explanation:
A Db2 CAF open error occurred.

User response
See the Db2 for z/OS Application Programming and SQL Guide for CAF return code and reason code information.

Return Code
12

INZI151E  ERROR WHEN FINDING obj ERROR CODE: rc  
REASON CODE: rs

Explanation:
Db2 HPU cannot continue because an internal error occurred when trying to find information in the Db2 control blocks. The search object, obj, can be one of the following objects:

- SQLID
- SYSOPERATOR ID
- VCAT
- VERSION NUMBER
- DATACLASS
- MINIMUM DIVIDE SCALE
- DEFAULT FOR TS AND IX PRIMARY QUANTITY
- SYSADM ID
- AUTHID
- SECURITY LABEL

User response
Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZI154E  UNEXPECTED INTERNAL CONVERSION .KEY 0x

Explanation:
An error occurred during the internal conversion.

User response
Contact IBM Software Support, and supply the reason code.

Return Code
8

INZI157E  ERROR : UNABLE TO LOAD DSNHDECP FROM THE DSNEXIT LIBRARY dsname

Explanation:
The DSNHDECP module load from the dsname library failed.

User response
Check the value of the VZD007 installation parameter for the specified Db2 subsystem.

Return Code
8

INZI158E  ERROR WHEN TRYING TO OBTAIN A SHARED LOCK ON TABLE : name

Explanation:
An error occurred while positioning a shared lock.

**User response**
Resubmit the job. If the error occurs again, contact IBM Software Support and supply the return and reason codes.

**Return Code**
8

**INZI159E**
ERROR WHEN TRYING TO OBTAIN AN EXCLUSIVE LOCK ON TABLE: name

**Explanation:**
An error occurred while positioning an exclusive lock.

**User response**
Resubmit the job. If the error occurs again, contact IBM Software Support and supply the return and reason codes.

**Return Code**
8

**INZI161I**
TIMEOUT WAITING STOP STATUS OF THE objtype name

**Explanation:**
A timeout occurred while waiting for the result of a previous STOP command. The object type can be a TABLESPACE or INDEXSPACE. The STOP command is reissued.

**User response:**
No action is required.

**Return Code**
0

**INZI162E**
OPERATOR ISSUED CANCEL: DB2 CANNOT STOP THE objtype name

**Explanation:**
A CANCEL command was issued. The object type can be a TABLESPACE or INDEXSPACE. This message is issued when Db2 response time is slow.

**User response**
Increase the values of WAITQTY, WAITQTYM, and WAITUNIT, and resubmit the job.

**Return Code**
8

**INZI163E**
ERROR WHEN TRYING TO OBTAIN AN EXCLUSIVE LOCK ON TABLE: name

**Explanation:**
An error occurred while positioning an exclusive lock.

**User response**
Resubmit the job. If the error occurs again, contact IBM Software Support and supply the return and reason codes.

**Return Code**
8

**INZI164E**
ERROR WHEN TRYING TO STOP THE objtype name DB2 RETURN CODE: rc DB2 REASON CODE: rs

**Explanation:**
A STOP request was issued for a table space or index space, but the object might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

**User response**
See Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.

**Return Code**
8

**INZI165E**
ERROR WHEN TRYING TO STOP THE PARTITION(S) OF objtype name DB2 RETURN CODE: rc DB2 REASON CODE: rs

**Explanation:**
A STOP request was issued for a table space or index space that might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

**User response**
See Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.
Return Code
8

INZI166E ERROR WHEN TRYING TO START
THE objtype name DB2 RETURN CODE : rc DB2 REASON CODE : rs

Explanation
A START request was issued for a table space or index space that might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

User response
See Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.

Return Code
8

INZI167E ERROR WHEN TRYING TO START
THE PARTITION(S) OF objtype name DB2 RETURN CODE : rc DB2 REASON CODE : rs

Explanation:
A START request was issued for a table space or index space that might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

User response:
See Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.

Return Code: 8

INZI169E ERROR WHEN OPENING SYSIN

Explanation
An internal error occurred and Db2 HPU was unable to open the SYSIN data set.

User response
Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZI170E ERROR WHEN WRITING SYSIN

Explanation
An internal error occurred and Db2 HPU was unable to write to the SYSIN data set.

User response
Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZI171E ERROR WHEN CLOSING SYSIN

Explanation
An internal error occurred and Db2 HPU was unable to close the SYSIN data set.

User response
Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZI172E name DSNUTILB FAILED RETURN CODE = nn

Explanation
The specified Db2 DSNUTILB utility cannot continue because it encountered an error.

User response
Examine the Db2 messages in the SYSPRINT file and refer to Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.

Return Code
8

INZI173E name DSNUTILB ABENDED RETURN CODE = nn

Explanation
The Db2 DSNUTILB utility has abended.

User response
Examine the Db2 messages in the SYSPRINT file and refer to Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.
Return Code 8

INZI174E ERROR WHEN TRYING TO DETERMINE ENFM STATUS

Explanation
A failure occurred when determining the Db2 Version 8 subsystem mode. This error can occur because the DISPLAY GROUP command that was used for this purpose failed.

User response
Check for Db2 messages about the DISPLAY GROUP command. If authorization problems are reported by Db2, ensure that the user who performed the DISPLAY GROUP command has the necessary authorization to do so. An authorized user is the user who submitted the job and the user who is specified by the VUM028/DISPLUSR PARMLIB parameter.

Return Code 8

INZI175I PROCESSING SYSIN AS scheme

Explanation
This is an informational message that indicates the encoding scheme that is used to read the SYSIN.

User response:
No action is required.

Return Code 0

INZI180E ERROR WHEN DISPLAYING THE STATUS OF THE objtype name

Explanation
An error occurred while attempting to display the status of the table space or index space. This message is usually preceded by message INZI187E, which contains detailed information about the error. The object type can be a TABLESPACE or INDEXSPACE.

User response
Check the value of PARMLIB parameter VUM028/DISPLUSR. See message INZI187E for more information, and see the Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes.

Return Code 8

INZI181E THE PARTITION n OF THE objtype name IS NOT AVAILABLE : status

Explanation
An UNLOAD of a partitioned table space was requested but the specified partition of the object is unavailable. The object type can be a TABLESPACE or INDEXSPACE.

User response
Modify the status of the object, and resubmit the job.

Return Code 8

INZI182E THE objtype name IS NOT AVAILABLE : status

Explanation
An UNLOAD was requested but the specified object is unavailable. The object type can be a TABLESPACE or INDEXSPACE.

User response
Modify the status of the object and restart the utility.

Return Code 8

INZI183E FAILED TO SWITCH TSO USER userid TO EXECUTE DISPLAY COMMAND.

Explanation
The switch from the job user ID to the user ID that was specified in the VUM028/DISPLUSR PARMLIB parameter failed.

User response
Check the VUM028/DISPLUSR PARMLIB parameter and verify that a valid user ID was specified.

Return Code 8

INZI184E FAILED TO SWITCH BACK TSO USER userid
Explanation
The switch from the user ID that was specified in the VUMO28/DISPLUSR PARMLIB parameter to the initial job user ID failed.

User response
Contact IBM Software Support.

Return Code
8

INZI187E DB2 COMMAND ERROR, DB2 RETURN CODE: rc REASON CODE: rs EXPLANATION: text

Explanation
A Db2 command failed. When possible, the result of the command is printed in the SYSPRINT file.

User response
See Db2 Universal Database for z/OS Messages and Codes for a description of the possible return codes and reason codes. Check the SYSPRINT file for the result of the command.

Return Code
8

INZI188E UNABLE TO RETRIEVE VSAM INFORMATION FOR THE objtype name [PART n] LDS : dsname

Explanation
A LISTCAT command failed for the underlying LDS of a Db2 object. The object type can be TABLESPACE or INDEXSPACE. The dsname of the LDS is in error and, if the object is partitioned, the partition number is contained in the message.

User response
No action is required.

Return Code
8

INZI189I DISPLAY GROUP DETAIL INFORMATION : RC=rc REASON=rs

Explanation
This is an informational message. The DISPLAY GROUP DETAIL that was performed to determine the Db2 MODE through Db2 CAF interface returned a warning (RC=4). The return code and reason code from CAF are also contained in this message.

User response
No action is required.

Return Code
0

INZI190E ERROR WHEN TRYING TO QUIESCE THE object name

Explanation
The table space in a table space quiesce request might be the object of another active utility.

User response
See the Db2 Codes or Db2 Messages guide for a complete description of the possible error messages that are printed in SYSPRINT.

Return Code
8

INZI191E ERROR WHEN TRYING TO QUIESCE THE PARTITION(S) OF object name

Explanation
The table space in a table space quiesce request might be the object of another active utility.

User response
See the Db2 Codes or Db2 Messages guide for a complete description of the possible error messages that are printed in SYSPRINT.

Return Code
8

INZI192E UNABLE TO LOAD THE DB2 LIBRARY (system_explanation)

Explanation
Db2 HPU was not able to load a required Db2 module from the allocated Db2 libraries. The text reports the system explanation received when the attempt to load the module failed. For example, the system_explanation fields can be 'CEE3501S The module DSNAO64C was not found.' when the DSNAO64C Db2 module required for ODBC access was not available to Db2 HPU.
User response
Fix the situation reported by the system_explanation. You may need to ensure that the tasks described in "Authorizing and enabling Db2 HPU" section has been correctly performed.

Return Code
8

INZI193E ERROR: cli_description_error
Explanation:
Db2 HPU received error from Db2 CLI. This message contains the error description generated by the Db2 CLI.

User response
Refer to the "ODBC guide and reference" publication to determine the reason of the error reported by the cli_description_error field of the message and fix it. You may need to ensure that the tasks described in "Authorizing and enabling Db2 HPU" section has been correctly performed.

Return Code
8

INZI194E DB2 CONNEXION TO ssid BY [CAF | RRSAF] FAILED, RETURN CODE: return code REASON CODE: reason code
Explanation:
an attempt to connect to Db2, either by the CAF method through the CONNECT service or by the RRSAF method through the IDENTIFY service, failed with the indicated Db2 return and reason codes.

User response
Determine the reason of the error and its solution by checking the meaning of the indicated Db2 return and reason codes from the Db2 codes documentation.

Return Code
8

INZI195I DB2 STATUS OF THE PARTITION partition.number OF THE tablespace/indexspace dbname.tsname/dbname.isname : restricted_state
Explanation
The requested QUIESCE utility could not be run against the indicated partition of the object to unload because of the mentioned Db2 restricted state.

User response:
Check further messages on potential subsequent implication of the reported failure.

Return Code
4

INZI197I EXPLAIN PLAN SUCCESSFULL:
QUERYNO = number,
EXPLAIN_TIME = timestamp
Explanation
This message indicates that an EXPLAIN PLAN was performed successfully without automatic deletion of inserted rows in EXPLAIN tables. It gives the QUERYNO and EXPLAIN_TIME values to allow the manual deletion of the inserted rows.

User response:
No immediate action is required. As each execution of the job will insert a new set of data into the EXPLAIN tables, you may want to remove the just inserted row or gather the information for further deletion. You may also want not to keep the EXPLAIN tables rows by specifying the appropriate VUM034/EXPLNTB = DELETE in the parmlib or the DELETE_EXPLAIN_TABLES_ROWS option for this unload.

Return Code
0

INZI199W EXPLAIN TABLE: creator.name IS MISSING
Explanation
This message indicates that an EXPLAIN table required by the EXPLAIN PLAN statement is missing. This EXPLAIN PLAN statement was attempted by Db2 HPU because the OPTIMIZE_PART_RANGE feature was requested.

User response
If the indicated EXPLAIN table does not exist, create it.

Return Code
4
**Explanation**
The syntax contains a string with unbalanced delimiters.

**User response**
Check the SYSIN at the location that is indicated to find the string that contains the unbalanced delimiters. Correct the syntax and resubmit the job.

**Return Code**
8

**Explanation**
The syntax contains a hexadecimal string with unbalanced delimiters.

**User response**
Check the SYSIN at the location indicated to find the string that contains the unbalanced delimiters. Correct the syntax and resubmit the job.

**Return Code**
8

**Explanation**
A sequence of DBCS characters is not stopped with a SHIFT_IN character.

**User response**
Check the SYSIN at the location that is indicated to find the string with the missing SHIFT_IN character. Correct the syntax, and resubmit the job.

**Return Code**
8

**Explanation**
A DBCS identifier is not followed by a SHIFT_IN character.

**User response**
Check the SYSIN at the location that is indicated to find the string with the missing SHIFT_IN character. Correct the syntax, and resubmit the job.

**Return Code**
8

**Explanation**
This message indicates that the zIIP usage is requested. It precises which proportion of eligible tasks (SRB tasks) should be off-loaded. This value is set either by the operand of the ZIIP option or by the associated parmlib parameter, i.e. ZIIP usage (VUX039/ZIIP).

**User response:**
If enabling the zIIP was not meant, consider removing the ZIIP option from your unload command or specifying ZIIP OFF.

**Return Code**
0

**Explanation**
The syntax contains a graphic string with unbalanced delimiters.

**User response**
Check the SYSIN at the location that is indicated to find the string that contains unbalanced delimiters. Correct the syntax, and resubmit the job.

**Return Code**
8
Explanation
When ZIIP usage is requested, this message gives information about the type and numbers of available GENERAL, ZIIP and OTHER CPUs. The types of processors are deduced from the PCCAATTR flag of the value of the PCCA system control block indicated in brackets below:

- ZIIP: zIIP processor (PCCAATTR =PCCAZIIP)
- OTHER: Special Processor (PCCAATTR=PCCAIFA) or Discretionary Processor (PCCAATTR=PCCADSCR)
- GENERAL: Any processor not listed above ((PCCAATTR not in PCCAZIIP,CCAIFA, PCCADSCR)

See ""z/OS MVS Data Areas Volume 5" manual for more details about the PCCA control block.

User response:
None.

Return Code
0

INZI212I NO ZIIP AVAILABLE, ZIIP SET TO OFF

Explanation
The zIIP usage is requested but no zIIP processor has been found. The zIIP specific processing is not performed and no task will be off-loaded to a zIIP processor. Db2 HPU behaves as if ZIIP OFF was specified.

User response:
None.

Return Code
0

INZI213W ZIIP ACTIVATION FAILED, RC=return_code, REASON=reason_code

Explanation
The zIIP usage is requested but the ZIIP initialization phase failed. No task can be off-loaded to the zIIP processor(s). The unload process continues as if ZIIP OFF was specified.

User response:
Unless you are aware of malfunction of your zIIP processors, contact your IBM support.

Return Code
4

INZI214I THE SYNTAX KEYWORD option IS DEPRECATED AND IS IGNORED BY THE PROCESS.

Explanation
The specified option was valid in previous versions but is no longer used in the current version. The process does not take it into account.

User response
Refer to the “deprecated syntax” section and remove the keyword and its operands to prevent this message to be issued. See the table below that shows which keyword sequence to remove according to the keyword mentioned by the message. Option keyword sequences to be removed:

QUIESCECAT YES
QUIESCECAT NO
SRTVSMIN integer
SRTVSMAX integer
SQLACCES MINIMAL
SQLACCES AUTO
SQLACCES YES
SQLACCES NO

If none of the sequence mentioned in the table can be found, this means the syntax was not valid in any of the previous release and you must double check the current syntax diagram to find out how to correct the unload statement.

Return Code
0

INZI215I THE VALUE value OF THE PARMLIB VARIABLE variable IS DEPRECATED AND IS IGNORED BY THE PROCESS.

Explanation
The mentioned setting was applicable in previous version but is not longer taken into account by the process.

User response:
If variable is ONDEMAND_RESOURCE_ALLOCATION, remove the whole ONDEMAND_RESOURCE_ALLOCATION(value) for the ULOPTNS setting. For any variable, generate the customization jobs and submit the INZTVAR and INZCHECK jobs.
Return Code 0

INZI216E UNABLE TO DETERMINE CURRENT DB2 APPLCOMPAT

Explanation
The current DB2 APPLICATION COMPATIBILITY could not be retrieved for an undetermined reason.

User response
Contact your IBM support.

Return Code 8

INZI217I DB2 APPLCOMPAT IS SET TO value

Explanation
The Db2 current APPLCOMPAT was modified according to the APPLCOMPAT option specified in the GLOBAL OPTIONS block or by the VUM036/APPLCOMP parmlib parameter.

User response
None.

Return Code 0

INZI320I LISTDEF INFORMATION: LISTDEF NAME listdef-name

Explanation
This is an informational message that indicates a LISTDEF name that is used by Db2 HPU.

User response
No action is required.

Return Code 0

INZI321E LISTDEF ERROR: INVALID PARTLEVEL FOR objqual.objname VALID RANGE IS (1:n)

Explanation
The PARTLEVEL number that was specified in a LISTDEF or LISTDEFTBV is greater than the number of partitions within the specified table space.

User response
Modify the LISTDEF definition.

Return Code 8

INZI322E LISTDEF ERROR: THE option_name OPTION IS NOT SUPPORTED.

Explanation
Db2 HPU does not support the following options when they are specified in the LISTDEF:

- ALL
- XML
- LOB
- BASE
- HISTORY

User response
Remove the unsupported options from the LISTDEF definition.

Return Code 8

INZI323E TEMPLATE ERROR: DUPLICATE OPTION option IN TEMPLATE template-name

Explanation
A template option is defined multiple times.

User response
Modify the template definition to remove the duplicate options.

Return Code 8

INZI324E LISTDEF ERROR: OPTION PARTLEVEL CANNOT BE MIXED WITH RI OPTION

Explanation
PARTLEVEL and RI are defined in the same LISTDEF.

User response
Modify the LISTDEF definition.
Return Code
8

INZI325I  TEMPLATE INFORMATION:
TEMPLATE NAME template-name

Explanation
This is an informational message that indicates a template name that was used by Db2 HPU.

User response:
No action is required.

Return Code
8

INZI326E  TEMPLATE ERROR: VOLCNT
PARAMETER MUST BE FROM 1 TO
59 FOR DASD DATASET

Explanation
An invalid VOLUME COUNT is set in a template for a DASD data set.

User response
Change the VOLCNT parameter in the template definition.

Return Code
8

INZI327E  TEMPLATE ERROR: WHEN
SPECIFYING A DEVICE NUMBER,
UNCNT MUST BE SET TO 1.

Explanation
An invalid UNIT COUNT is set in a template, when a physical DEVICE NUMBER is used.

User response
Remove the UNCNT parameter from the template or set UNCNT to 1.

Return Code
8

INZI328I  DATASET ALLOCATED.
TEMPLATE=template-name
DDNAME=ddname DSN=dsname

Explanation
This is an informational message. A data set that was generated from a TEMPLATE is allocated by Db2 HPU.

User response:
No action is required.

Return Code
0

INZI329I  DATASET ALLOCATED FOR
PARTITION part.
TEMPLATE=template-name
DDNAME=ddname DSN=dsname

Explanation
This is an informational message. A data set that was generated from a TEMPLATE for a specific partition is allocated by Db2 HPU.

User response:
No action is required.

Return Code
0

INZI330E  DYNAMIC ALLOCATION FAILED.
TEMPLATE = template_name
DSN=dsname RC=rc, REASON=rs.
SVC99 MESSAGES ARE REPORTED
IN JESMSGL CARD

Explanation
A dynamic allocation through SVC 99 failed. The SVC 99 return code and the corresponding reason code are contained in the return and reason codes. Dynamic allocation messages IKJnnnnnI are issued in WTO in the JESMSGL card. This message is followed by message INZI332I, which contains dynamic allocation parameters.

User response
Refer to the allocation messages that were issued in WTO to determine the reason for the allocation error. If the dsname is generated from a template, correct the template, and resubmit the job. Otherwise, contact IBM Software Support and supply them with the content of messages INZI330E, INZI332I, and IKJnnnnnI.

Return Code
8

INZI331E  FREE OF DDNAME ddname
FAILED, RC=rc, REASON=rs.
SVC99 MESSAGES ARE REPORTED IN JESMSGL CARD

Explanation
A dynamic free through SVC 99 failed. The SVC 99 return code and corresponding reason code are contained in the return and reason codes. Dynamic allocation messages IKJnnnnnI are issued in WTO in the JESMSGL card. This message is followed by information message INZI332I, which displays dynamic allocation parameters.

User response
Contact IBM Software Support and supply them with the content of messages INZI330E, INZI332I, and IKJnnnnnI.

Return Code
8

INZI332I DYNAMIC ALLOCATION PARAMETER LIST (SVC99): text

Explanation
This message displays the dynamic allocation parameters when a dynamic allocation or free failed.

User response:
No action is required.

Return Code
0

INZI333E SPACE ESTIMATION FAILED

Explanation
Db2 HPU failed to estimate the allocation of the file that was generated by the TEMPLATE statement because a LISTCAT command on the underlying LDS file failed.

User response
Check the status of the table space LDS files.

Return Code
8

INZI336E DYNAMIC ALLOCATION FAILED. TEMPLATE=template_name GDG BASE=dsn_gdg IS NOT DEFINED AND GDGLIMIT IS SET TO 0

Explanation:
Db2 HPU failed to allocate a GDS file from a template. The GDG base is not defined and the GDGLIMIT is set to zero to prevent its definition.

User response
Do one of the following:
- Change the GDGLIMIT setting to a non-zero value in the SYSIN and then resubmit the job.
- Create the GDG using an IDCAMS DEFINE GENERATIONDATAGROUP command and then resubmit the job.

Return Code
8

INZI337E DYNAMIC ALLOCATION FAILED FOR TEMPORARY FILE=temp_dsn RC=rc, REASON=reason

Explanation:
The dynamic allocation of a temporary file by using DYNALLOC services (SVC 99) failed. RC and REASON contain the dynamic allocation return and reason codes. Dynamic allocation messages IKJnnnnnI are issued in WTO in the JESMSGL card.

User response:
Refer to the allocation messages that were issued in WTO to determine the reason for the allocation error.

Return Code
8

INZI338E IDCAMS ERROR RC=rc, sysprint of the IDCAMS execution which failed

Explanation:
The IDCAMS program invoked by Db2 HPU has failed. This error message is followed by the SYSPRINT of the IDCAMS which failed. RC contains the IDCAMS return code.

User response:
Refer to the content of the IDCAMS to determine the reason of the error.

Return Code
8

INZI339E ALLOCATION OF BASE GDG=gdg_dsn FAILED

Explanation:
This message occurs when Db2 HPU failed to create a base GDG for a template file. This error message is
generally preceded by error message INZU337E or INZU338E.

**User response:**
Refer to the content of message INZU337E or INZU338E and check the template definition.

**Return Code**
8

**INZI340E**  
MISSING MODELDCB OR DATACLAS IN DEFINITION OF TEMPLATE=template_name

**Explanation:**
This message occurs when a template that corresponds to GDG files is defined without a MODELDCB or DATACLAS parameter and when the parmlib parameter TMPLDSCB is left blank.

**User response:**
Check the template definition and parmlib parameter TMPLDSCB, and resubmit the job.

**Return Code**
8

**INZI341E**  
MODELDCB=dsname IN TEMPLATE=template_name IS NOT CATALOGUED

**Explanation:**
This message occurs when a template contains a MODELDCB that does not correspond to a catalogued entry.

**User response:**
Correct the setting of the MODELDCB parameter in the template definition, and resubmit the job.

**Return Code**
8

**INZI344I**  
LISTDEF LISTDEF_name: CLAUSE - INCLUDE_or_EXCLUDE_clause - IDENTIFIES NO OBJECTS

**Explanation:**
A LISTDEF or LISTDEFTBV clause returned no object. LISTDEF_name is the name of the list defined the LISTDEF or LISTDEFTBV statement, and INCLUDE_or_EXCLUDE_clause indicates which clause returned no object.

**User response:**
No action is required.

**Return Code**
0

**INZI346E**  
LISTDEF INVALID SQLCODE=sqlcode FOR sql_order_type_

**Explanation:**
An SQL error occurred during the search of Db2 objects for a LISTDEF or LISTDEFTBV statement. sql_order_type_ can be one of the following: DECLARE/PREPARE CURSOR, OPEN CURSOR, FETCH CURSOR or CLOSE CURSOR

**User response:**
Refer to the previous sql error message. If it suggests a site-dependent or temporary issue, solve the issue or simply run the job again. Otherwise, contact the IBM support.

**Return Code**
12

**INZI348W**  
The option_name OPTION IN THE template_name TAPE TEMPLATE IS IGNORED.

**Explanation:**
Db2 HPU ignored an option in a template. The option name and template name are indicated in the message text.

**User response:**
Remove the specified option to suppress this warning.

**Return Code**
4

**INZI368W**  
The CONFIGURATION PARAMETER name IS OUT OF RANGE (min_value - max_value).

**Explanation:**
The value specified for the indicated PARMLIB parameter is not within the allowed range. The parameter must be greater than or equal to min_value and less than or equal to max_value.

User response:
Set the indicated parameter with a supported value and resubmit the job.

Return Code
4

INZI436E SYNTAX ERROR IN DDLDDN FILE
STARTING AT (line1,column1) AND
ENDING AT (line2,column2)

Explanation
A syntax error occurred while analyzing the DDL DDN file. Db2 HPU was unable to determine the cause of the error but the error message contains the location where the error occurred.

User response
Verify that valid syntax was used in the indicated area of the DDLDDN file.

Return Code
8

INZI458E utility COULD NOT START A NEW THREAD OR PROCESS (CODE reason) (reason text)

Explanation
An error occurred when trying to start a new thread. The system returned the mentioned reason code along with the indicated reason text when the attempt to start a new thread failed.

User response
Determine the root cause of the indicated system reason code and fix it. If the reason is the lack of resources to create a new thread, consider decreasing the Db2 HPU need for threads with the help of either the PARALLELISM or the MAXPART option or both. You may also want to use the corresponding parmlib parameters instead, i.e. VUX025/PARALLEL, VUX031/DB2PARAL, VUX030/UTLPARAL and VUU060/ULMAXPAR PARMLIB but as the optimal parallelism configuration might apply especially to the current unload, changing the unload statement should be preferred.

Return Code
8

INZI500I DDL ANALYSING: PROCESSING DDL FILE DDN=ddname

Explanation
This is an informational message. The analysis of the DDL file has started.

User response: No action is required.

Return Code
0

INZI501I DDL ANALYSING: IN BLOCK
STARTING AT POS(line1, column1)
ENDING AT POS(line2, column2)

Explanation
This is an informational message that provides the position of the DDL file that is currently being analyzed.

User response: No action is required.

Return Code
0

INZI502I DDL ANALYSING: ITEM STARTING
AT POS(line, column)

Explanation
This is an informational message that provides the position of the DDL file that is currently being analyzed.

User response: No action is required.

Return Code
0

INZI503E DDL ANALYSING: SYNTAX ERROR
AT POS(line, column)

Explanation
A syntax error occurred in a DDL file during analysis.
Correct the syntax error. Contact IBM Software Support if you are unable to determine the cause of the error.

**Return Code**
8

**INZI510E**  
**DDL ANALYSING: DUPLICATE DESCRIPTION OF objtype name**

**Explanation**
An object is defined twice in the DDLDDN file. This message is preceded by informational message INZI502I, which contains the location of the duplication in the DDL file.

**User response**
Check the CREATE clauses in the DDLDDN file and remove the duplicate definition.

**Return Code**
8

**INZI511E**  
**DDL ANALYSING: DUPLICATE DEFINITION OF PARTITION IN CREATE INDEX STATEMENT**

**Explanation**
The CREATE INDEX clause specifies a clustering index partition more than once.

**User response**
Check the CREATE INDEX clause in the DDLDDN file and remove the duplicate definition.

**Return Code**
8

**INZI512E**  
**DDL ANALYSING: INVALID INDEX KEY IN CREATE INDEX STATEMENT**

**Explanation**
The CREATE INDEX clause specifies an incorrect key.

**User response**
Check the CREATE INDEX clauses in the DDLDDN file and remove duplicate definitions.

**Return Code**
8

**INZI513E**  
**DDL ANALYSING: UNDEFINED COLUMN IN THE CREATE INDEX STATEMENT**

**Explanation**
The CREATE INDEX clause specifies an unknown column in the DDLDDN file.

**User response**
Check the CREATE TABLE and CREATE INDEX clauses in the DDLDDN file for invalid column names.

**Return Code**
8

**INZI514E**  
**DDL ANALYSING: UNDEFINED TABLE IN CREATE INDEX STATEMENT**

**Explanation**
An index is defined on a table that was not previously described in the DDLDDN file.

**User response**
Check the CREATE TABLE and CREATE INDEX clauses in the DDLDDN file for the undefined index.

**Return Code**
8

**INZI515E**  
**DDL ANALYSING: INVALID BUFFER POOL: name**

**Explanation**
An invalid buffer pool is specified in the DDL.

**User response**
Check the CREATE TABLESPACE or CREATE DATABASE clauses in the DDLDDN file.

**Return Code**
8

**INZI519W**  
*reason FOR [TABLESPACE/INDEXSPACE/TABLE] object [PART] part WAIT(wait time)/RETRY(nb retry) MECHANISM IS STARTING*
Explanation
Db2 HPU detects an unavailable resource. The wait/retry mechanism is activated for one of the following reasons:

- ALLOCATION FAILED
- DB2 STATUS (status) NOT SUPPORTED
- LOCK TABLE FAILED

If the resource becomes available, processing continues.

User response:
No action is required.

Return Code
None.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZI520E</td>
<td>DDL ANALYSING: MANDATORY CLAUSE &quot;IN DATABASE-NAME.TABLESPACE-NAME&quot; IS MISSING IN DDL OF TABLE</td>
</tr>
</tbody>
</table>

Explanation:
The CREATE statement for the specified table does not contain database and table space names, which Db2 HPU requires.

User response:
Add the IN DATABASE-NAME.TABLESPACE-NAME clause to the CREATE statement for the specified table.

Return Code
8

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZI523I</td>
<td>UNIT COUNT IS SET TO unit_count_value</td>
</tr>
</tbody>
</table>

Explanation:
The unit count was set by Db2 HPU for the template file that is indicated in the preceding INZI328I message or the INZI329I message.

User response:
No action is required.

Return Code
0

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZI524E</td>
<td>UNABLE TO READ PARMLIB pds name/ddname (rc = return-code)</td>
</tr>
</tbody>
</table>

Explanation:
Db2 HPU could not read the INZUTIL member from the specified partitioned data set or ddname to determine the installation parameters. The return-code is only meant for IBM support.

User response:
Check if the mentioned partitioned data set or the data set allocated under the mentioned DDname has a readable INZUTIL member. If not, either check the PARMLIB library (INZPLIB/INFPPLIB) parameter and make sure the customization step has been completed by running the INZPARM customization job or allocate a partitioned data set containing a relevant INZUTIL member under the INFPPLIB ddname. If a readable INZUTIL member can actually be found at the mentioned location (PDS or ddname), contact the IBM support.

Return Code
8

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZI527E</td>
<td>DDL ANALYSING: option1 OPTION REQUIRES MISSING option2 OPTION.</td>
</tr>
</tbody>
</table>

Explanation:
This message indicates that the DDL provided for the DDLDDN option is incorrect because an option required by the first option is missing.

User response:
Correct the syntax error.

Return Code
8

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZI528E</td>
<td>DDL ANALYSING: option1 OPTION IS INCOMPATIBLE WITH option2 OPTION.</td>
</tr>
</tbody>
</table>

Explanation:
This message indicates that the DDL provided for the DDLDDN option is incorrect because of a conflict between 2 options in the DDL.

User response:
Correct the syntax error.

Return Code
8

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZI529I</td>
<td>THE PARMLIB PARAMETERS HAVE BEEN READ FROM: path</td>
</tr>
</tbody>
</table>

Explanation:
This message indicates the name of the PDS or DDN from where the parmlib has been read.

User response:
No action is required

Return Code
0
INZI531E TEMPLATE ERROR: INVALID TEMPLATE DEFINITION FOR TEMPLATE alternate_template_name POINTED BY TEMPLATE base_template_name

Explanation:
This message indicates that the alternate template specified by the LIMIT option of a base template is incompatible with the base template because the limit value it specifies for size is less than the main template size limit.

User response:
Check and modify the template(s) definition(s) to ensure the rule concerning size limits is fulfilled.

Return Code: 8

INZI532W TEMPLATE WARNING: LIMIT OPTION FOR TEMPLATE template_name IS IGNORED DUE TO USAGE OF SPACE OPTION

Explanation:
This message indicates that the LIMIT option of the TEMPLATE statement is ignored because a SPACE option is also specified.

User response:
Check and modify the template(s) definition(s)

Return Code: 8

INZI534E CALL TO STORED PROCEDURE proc FAILED.

Explanation
An error occurred while attempting to use the indicated stored procedure for one of the following reasons:
• case 1: the stored procedure is not reachable, in which case preceding Db2 error messages give more details on the reason of the error.
• case 2: the stored procedure returned an error code, in which case the message text contains the explanation of the IFI or SQL error.

User response
Proceed as described below, depending on the applicable case:
• case 1: Ensure that stored procedures and corresponding WLM environments are correctly created on the involved Db2 subsystem or data sharing group. If the involved stored procedure is ADMIN_INFO_SYSPARM, you may want to disallow its use by specifying USE_ADMIN_INFO_SYSPARM(NO) for the VUU030/ULOPTNS parameter.
• case 2: Refer to the documentation of the related stored procedure to determine the root cause of the error.

Return Code: 8

INZI535E INTERNAL ERROR WHEN ACCESSING TO dsname PROCESSING IN NATIVE MODE IS IMPOSSIBLE.

Explanation:
An internal error occurred when accessing the indicated data set name during an unload processed in native mode. A potential subsequent an unpredictable ABEND can occur.

User response:
Contact your IBM support and consider using the DB2 FORCE option if applicable.

Return Code: 8

INZI666F FATAL ERROR : explanation

Explanation
An unexpected exception has occurred. The explanation text, if any, is intended only for support.

User response
Report this failure to your IBM Software Support.

Return Code: 16

INZI800E ERROR ACCESSING TO A DATASET. ACTION: service RC: return-code REASON: reason DDN: ddname DSN: dsname

Explanation
Explanation: an error occurred when allocating or accessing to a dataset. The concerned ddname, dsname, service and related reason and return code are indicated in the message. The service affected by the issue can be one of the following:
• OPEN
• CLOSE
• GET
• PUT
• MODCB
• STOW
• CHECK
• POINT
• ALLOCATE
• FREE

If the concerned service is either ALLOCATE or FREE, the return and reason codes indicated by the message are those from dynalloc. For the other services, the return and reason codes are those from the related VSAM macro or QSAM macro.

User response
Determine the reason of the allocation error with the help of the explanation of:
• the mentioned return and reason codes for the specified 'service',
• the potential system error messages concerning the involved ddname or dsname.

Return Code
8

INZRE40  PAGE NO. page number IN LDS NO. LDS number OF PAGESET pageset IS NOT IN SEQUENCE, SCAN ABORTED

Explanation
Db2 HPU encountered a row that is invalid.

User response
Use the REPAIR command to correct the condition; then resubmit the job. This message is issued if Db2 HPU encounters an error that is related to the row structure while it is reading the rows of a table space. To limit the number of error messages that are issued, use the VUX018/LDSERRLM variable in the PARMLIB. When the limit that is specified in VUX018/LDSERRLM is reached, message INZRE41 is issued with the following text: INZRE41 MESSAGE LIMIT REACHED, ONE OR MORE MESSAGE(S) NOT PRINTED.

Return Code
8

INZRE42  ROW AT OFFSET nnnn ON PAGE (HEX) nnnn OF PAGESET xxxx IS INVALID

Explanation
Db2 HPU encountered a row that is invalid.

User response
Use the REPAIR command to correct the condition; then resubmit the job. This message is issued if Db2 HPU encounters an error that is related to the row structure while it is reading the rows of a table space. To limit the number of error messages that are issued, use the VUX018/LDSERRLM variable in the PARMLIB. When the limit specified in VUX018/LDSERRLM is reached, message INZRE42 is issued with the following text: INZRE42 MESSAGE LIMIT REACHED, ONE OR MORE MESSAGE(S) NOT PRINTED.

Return Code
8

INZRE45  IMAGE COPY FROM DD ddname CONTAINS NO ROW WITH SPECIFIED OBID(S). SCAN ABORTED

Explanation:
The image copy data set that is to be unloaded and allocated to the ddname ddname contains a table that does not belong to the requested table. The OBID report provides information about the image copy. This

INZRE40  PAGE NO. page number IN LDS NO. LDS number OF PAGESET pageset IS NOT IN SEQUENCE, SCAN ABORTED

Explanation
Db2 HPU encountered a row that is invalid.

User response
Use the REPAIR command to correct the condition; then resubmit the job. This message is issued if Db2 HPU encounters an error that is related to the row structure while it is reading the rows of a table space. To limit the number of error messages that are issued, use the VUX018/LDSERRLM variable in the PARMLIB. When the limit specified in VUX018/LDSERRLM is reached, message INZRE41 is issued with the following text: INZRE41 MESSAGE LIMIT REACHED, ONE OR MORE MESSAGE(S) NOT PRINTED.

Return Code
8

INZRE42  ROW AT OFFSET nnnn ON PAGE (HEX) nnnn OF PAGESET xxxx IS INVALID

Explanation
Db2 HPU encountered a row that is invalid.

User response
Use the REPAIR command to correct the condition; then resubmit the job. This message is issued if Db2 HPU encounters an error that is related to the row structure while it is reading the rows of a table space. To limit the number of error messages that are issued, use the VUX018/LDSERRLM variable in the PARMLIB. When the limit specified in VUX018/LDSERRLM is reached, message INZRE42 is issued with the following text: INZRE42 MESSAGE LIMIT REACHED, ONE OR MORE MESSAGE(S) NOT PRINTED.

Return Code
8

INZRE45  IMAGE COPY FROM DD ddname CONTAINS NO ROW WITH SPECIFIED OBID(S). SCAN ABORTED

Explanation:
The image copy data set that is to be unloaded and allocated to the ddname ddname contains a table that does not belong to the requested table. The OBID report provides information about the image copy. This
report is created only when an image copy of a table space that contains a single table is processed.

**User response:**
Specify the correct OBID value or ORIGINOBID 0.

**Return Code**
8

**INZR1007**  ERROR WHEN OPENING VIRTUAL FILE : DB2FVNNN

**Explanation**
An internal error occurred when opening virtual file DB2FVNNN.

**System action**
Processing ends.

**User response**
Contact IBM Software Support, and supply the return codes and reason codes.

**Return Code**
8

**INZR1008**  ERROR WHEN WRITING VIRTUAL FILE : DB2FVNNN

**Explanation**
An internal error occurred when writing virtual file DB2FVNNN.

**System action**
Processing ends.

**User response**
Contact IBM Software Support, and supply the return codes and reason codes.

**Return Code**
8

**INZR2000**  DB2 ERROR: PREPARE / DESCRIBE

**Explanation**
This is a Db2 error. The lines that follow this message contain the error message that is generated by Db2.

**System action**
Processing ends.

**User response**
Correct the error.

**Return Code**
8

**INZR2001**  DB2 ERROR: OPEN CURSOR

**Explanation**
This is a Db2 error. The lines that follow this message contain the error message that is generated by Db2.

**System action**
Processing ends.

**User response**
Correct the error.

**Return Code**
8

**INZR2004**  DB2 ERROR: FETCH CURSOR

**Explanation**
This is a Db2 error. The lines that follow this message contain the error message that is generated by Db2.

**System action**
Processing ends.

**User response**
Correct the error.

**Return Code**
8

**INZR2010**  DB2 ERROR: CONNECTING TO ssid, RETURN CODE=rc REASON CODE=rs

**Explanation**
A Db2 CAF connection error occurred.

**User response**
See the CAF return codes and reason codes in the *Db2 for z/OS Application Programming and SQL Guide*.  

Chapter 8. Troubleshooting 389
Return Code 12

INZR2011 DB2 ERROR: OPENING PLAN planname, RETURN CODE=rc
REASON CODE=rs

Explanation
A Db2 CAF open error occurred.

User response
See the CAF return codes and reason codes in the Db2 for z/OS Application Programming and SQL Guide.

Return Code 12

INZR2012 DB2 ERROR: DISCONNECTING FROM ssid, RETURN CODE=rc
REASON CODE=rs

Explanation
An error occurred while Db2 HPU was disconnecting from Db2.

User response
See the CAF return codes and reason codes in the Db2 for z/OS Application Programming and SQL Guide.

Return Code 8

INZR2013 DB2 ERROR: CLOSING PLAN planname, RETURN CODE=rc
REASON CODE=rs

Explanation
A Db2 CAF closing error occurred.

User response
See the CAF return codes and reason codes in the Db2 for z/OS Application Programming and SQL Guide.

Return Code 8

INZR2014 DB2 ERROR: CLOSE CURSOR

Explanation:
The Db2 error described by the Db2 message reprinted in the following lines occurred.

User response:
Fix the error described the indicated Db2 message.

Return Code: 8

INZR3000 NUMBER OF ROWS EXTRACTED COPIED ON ddname: rows-count

Explanation:
This message indicates the total number of records that were written into an output dataset by an unload processed in SQL mode.

User response:
No action is required.

Return Code: 0

INZR3001 NUMBER OF ROWS DISCARDED DUE TO DATA TRUNCATION: number

Explanation:
During an unload processed in SQL mode, some rows have not been retrieved as a truncation occurred (the process received an SQLCODE +354).

User response:
Consider changing the SQL statement to avoid any truncation error.

Return Code: 4

INZT003E TEMPLATE ERROR: INVALID EXPIRATION DATE FORMAT.
VALID FORMATS ARE YYYYDDD, YYYY/DDD

Explanation
An invalid expiration date is specified in a template.

User response
Specify a valid date and resubmit the job.

Return Code 8

INZT004E TEMPLATE ERROR: INVALID OPERAND operand

Explanation
An invalid operand is specified in a template.

User response
Specify a valid operand and resubmit the job.
INZT005E  TEMPLATE ERROR: VOLSER volser IS TOO LONG

Explanation
An invalid VOLSER is specified in a template.

User response
Specify a valid VOLSER and resubmit the job.

INZT006I  TEMPLATE INFORMATION: TAPE-OPTIONS ARE IGNORED BY UTILITY

Explanation
Db2 HPU does not support TAPE-OPTIONS of TEMPLATE. These options are ignored.

User response
No action is required.

INZT007E  TEMPLATE ERROR: INVALID GDG PARAMETER

Explanation
The GDG that is specified in a TEMPLATE statement is incorrect.

User response
Correct the incorrect GDG and resubmit the job.

INZT008E  TEMPLATE ERROR: OPERAND n OF SUBSTRING substring IS OUT OF RANGE

Explanation
The parameters specified in a SUBSTRING expression in a template are out of range.

User response
Check the syntax of the DSN in the template, verify that they are in the correct range, and resubmit the job.

INZT009E  TEMPLATE ERROR: DUPLICATE TEMPLATE NAME template-name

Explanation
A redefinition of a TEMPLATE occurs in the SYSIN.

User response
Correct the syntax and resubmit the job.

INZT010E  TEMPLATE ERROR: TEMPLATEDD ddname CANNOT BE OPEN

Explanation
The TEMPLATEDD library of a TEMPLATE cannot be opened.

User response
Verify that the TEMPLATEDD library is spelled correctly in the SYSIN and the JCL.

INZT011E  TEMPLATE ERROR: INVALID DSN VARIABLE var IN TEMPLATE

Explanation
An invalid variable is used in a dsname definition of a TEMPLATE.

User response
Modify the variable or use the TEMPLATESET option to define it.

INZT012E  TEMPLATE ERROR: INVALID SUBSTRING NOTATION substring IN DSN OPERAND. DSN OPERAND

Chapter 8. Troubleshooting 391
**Explanation**

The dsname expression of a TEMPLATE that uses the SUBSTR expression must be enclosed in single quotation marks.

**User response**

Use single quotation marks around the dsname specification.

**Return Code**

8

**INZT013E**  TEMPLATE ERROR: FATAL SYNTAX ERROR

**Explanation**

A syntax error occurred in a TEMPLATE definition. A previous message contains the position in the SYSIN where the analysis failed.

**User response**

Check the syntax of the TEMPLATE statement, correct any errors, and resubmit the job.

**Return Code**

8

**INZT014E**  TEMPLATE ERROR: DSN OR PATH KEYWORD EXPECTED

**Explanation**

A DSN or PATH keyword is missing in a TEMPLATE definition.

**User response**

Correct the syntax of the TEMPLATE statement, and resubmit the job.

**Return Code**

8

**INZT015E**  TEMPLATE ERROR: OPERAND operand IS TOO LONG

**Explanation**

A DSN expression of a TEMPLATE definition contains a qualifier or member that is too long.

**User response**

Modify the syntax of the DSN definition in the TEMPLATE statement, and resubmit the job.

**Return Code**

8

**INZT016E**  TEMPLATE ERROR: INVALID OPERAND operand INTEGER IS EXPECTED

**Explanation**

An invalid operand is used in a TEMPLATE definition. An integer value was expected.

**User response**

Correct the TEMPLATE statement, and resubmit the job.

**Return Code**

8

**INZT017E**  TEMPLATE ERROR: INVALID OPERAND operand FOR OPTION option

**Explanation**

An invalid operand is used in one of the options of a TEMPLATE definition.

**User response**

Correct the TEMPLATE statement, and resubmit the job.

**Return Code**

8

**INZT018E**  TEMPLATE ERROR: SYNTAX ERROR FOR OPTION option

**Explanation**

A syntax error occurred in a TEMPLATE definition.

**User response**

Correct the TEMPLATE statement, and resubmit the job.

**Return Code**

8
**INZT019E**  TEMPLATE ERROR: NUMBER OF VOLSER IS GREATER THAN 255

**Explanation**
The number of VOLSER that were specified in a TEMPLATE definition is greater than the allowable limit of 255.

**User response**
correct the TEMPLATE statement, and resubmit the job.

**Return Code**
8

**INZT020E**  TEMPLATE ERROR: DSNTYPE PDS REQUIRES DIR OPTIONS

**Explanation**
A DSNTYPE PDS is specified in a TEMPLATE without a specified directory block option DIR.

**User response**
correct the TEMPLATE statement, and resubmit the job.

**Return Code**
8

**INZT021E**  TEMPLATE ERROR: OPERAND IS OUT OF RANGE (val1, val2) FOR OPTION option

**Explanation**
An operand in the TEMPLATE definition is out of range. The message specifies which option is involved and the authorized range (val1-val2).

**User response**
correct the TEMPLATE statement, and resubmit the job.

**Return Code**
8

**INZT022E**  TEMPLATE ERROR: option1 OPTIONS CANNOT BE MIXED WITH option2 OPTIONS

**Explanation**
Incompatible options are used in the TEMPLATE definition.

**User response**
correct the TEMPLATE statement, and resubmit the job.

**Return Code**
8

**INZT023W**  TEMPLATE WARNING: TEMPLATE template-name FOR STORAGE ON TAPE DEVICE IS IGNORED BY UTILITY

**Explanation**
Db2 HPU does not support the use of templates that are defined with TAPE options. This TEMPLATE definition is ignored and processing continues. If no other TEMPLATE definition exists and if the corresponding DDNAME is not allocated in the JCL, Db2 HPU will fail with error message INZC019E.

**User response**
If necessary, change the TEMPLATE definition or use another TEMPLATE.

**Return Code**
4

**INZT024E**  TEMPLATE ERROR: INVALID DISP SPECIFICATION

**Explanation**
A disposition option in the TEMPLATE definition is invalid.

**User response**
correct the TEMPLATE statement, and resubmit the job.

**Return Code**
8

**INZT025E**  TEMPLATE ERROR: INVALID OPERAND operand FOR OPTION option. VALID OPERAND(S) MAY BE values
Explanation
An invalid operand is used in a TEMPLATE definition. The message specifies which TEMPLATE option is invalid and lists valid values.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT026E  TEMPLATE ERROR: INVALID PARENTHETICAL EXPRESSION

Explanation
An invalid parenthetical expression is used in a TEMPLATE definition.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT027E  TEMPLATE ERROR: COMMA IS EXPECTED

Explanation
A comma is expected in a TEMPLATE definition.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT028E  TEMPLATE ERROR: RETPD AND EXPDL OPTIONS ARE EXCLUSIVES

Explanation
Options RETPD and EXPDL are exclusives in a TEMPLATE definition.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT029E  TEMPLATE ERROR: INVALID DSNAME QUALIFIER qualifier

Explanation
An invalid qualifier occurs in the dsname expression of a TEMPLATE definition.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT030E  TEMPLATE ERROR: DOT IS EXPECTED

Explanation
A DOT character is expected in the dsname expression of a TEMPLATE definition.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT032E  TEMPLATE ERROR: VARIABLE var IS NOT SUPPORTED IN CONTEXT OF utility

Explanation
A dsname variable that was used in the dsname expression of a TEMPLATE definition cannot be used by Db2 HPU.

User response
Refer to the list of supported variables, correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT033E  TEMPLATE ERROR: DSNAME GENERATED FROM TEMPLATE template-name IS INVALID DSN=dsname
Explanation
The dsname that was generated from a TEMPLATE definition is invalid. This message is usually issued when the length of the generated dsname is greater than 44 characters.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT034E TEMPLATE ERROR: TEMPLATE template-name1 REDEFINES DSNAME dsname PREVIOUSLY GENERATED FROM TEMPLATE template-name2

Explanation
The dsname that was generated from a TEMPLATE statement redefines a dsname that was previously generated. Each dsname generated from a TEMPLATE statement must be unique.

User response
Correct the TEMPLATE statement to ensure each dsname is unique, then resubmit the job.

Return Code
8

INZT035E TEMPLATE ERROR: DSN VARIABLE var HAS NO VALUE

Explanation
A dsname variable in the dsname expression of a TEMPLATE definition cannot be substituted.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT036I TEMPLATE template AT POS(line, column) IS READ FROM LIBRARY ddname

Explanation
This is an informational message. A template is defined in a TEMPLATE library.

User response:
No action is required.

Return Code
0

INZT038E TEMPLATE ERROR: UNSUPPORTED OPTION option

Explanation
An unsupported option is used in a TEMPLATE definition.

User response
Correct the TEMPLATE statement, and resubmit the job.

Return Code
8

INZT040E TEMPLATE ERROR: DSNAME REQUIRES MEMBER NAME WITH PDS OR LIBRARY DSNTYPE

Explanation:
A TEMPLATE that corresponds to an HFS file is specified without option DSNTYPE HFS.

User response:
Specify option DSNTYPE HFS in the TEMPLATE statement and resubmit the job.

Return Code:
8

INZT041E TEMPLATE ERROR: DSNAME GENERATED FROM TEMPLATE template_name IS TOO LONG DSN=dsn

Explanation:
The dsname that is generated from a TEMPLATE definition is too long. The message displays the generated dsname.

User response:
Correct the TEMPLATE statement and resubmit the job.

Reason Code
8

INZT043E TEMPLATE ERROR: THE option_name TEMPLATE OPTION

Chapter 8. Troubleshooting 395
WITH THE DSN KEYWORD IS NOT VALID.

Explanation:
The PATH options FILEDATA, PATHOPTS, PATHMODE, and PATHDISP cannot be specified with the DSN keyword in the TEMPLATE statement.

User response:
Remove the specified PATH option that is not valid.

Return Code
8

INZT044E TEMPLATE ERROR: THE PATH NAME AT POS(line, col) IS NOT VALID.

Explanation:
A DSN file name cannot be specified as a path name in the PATH option.

User response:
Correct the SYSIN, and rerun the job.

Return Code
8

INZT045E TEMPLATE ERROR: THE DSN OPTION AT POS(line, col) IS NOT VALID.

Explanation:
A path name that is specified with the DSN option must be enclosed in single quotation marks.

User response:
Enclose the path name in single quotation marks, and rerun the job.

Return Code
8

INZT047I TEMPLATE INFORMATION: THE option_name TEMPLATE OPTION IS IGNORED BECAUSE THE PATH OPTION IS SPECIFIED.

Explanation:
When the PATH option is used to create z/OS USS files, the template option is ignored.

User response:
If you want to use the PATH option, remove the template option that is ignored. If you do not want to use the PATH option, remove it.

Return Code
0

INZT048E TEMPLATE ERROR: THE option_name TEMPLATE OPTION CANNOT BE SPECIFIED WITH THE PATH OPTION.

Explanation:
The specified template option is incompatible when the PATH option is used to allocate a z/OS USS file.

User response:
Correct the SYSIN, and rerun the job.

Return Code
8

INZT049E TEMPLATE ERROR: THE alternate_template_name TEMPLATE SPECIFIED IN LIMIT OPTIONS OF base_template_name TEMPLATE IS NOT DEFINED

Explanation:
As the alternate template specified via the LIMIT option of the indicated base template is not defined, the processing of the base template cannot be completed.

User response:
Remove the LIMIT option if specified by mistake or provide the alternate template definition.

Return Code
8

INZT051E TEMPLATE ERROR: TEMPLATE template_name_1 AND template_name_2 LINKED BY LIMIT OPTION REDEFINED DSNAMES dsname USING [POTENTIALY] THE SAME UNIT OR STORAGE CLASS unit_or_storage_class_name

Explanation:
The indicated templates that are directly or indirectly linked by the LIMIT option of TEMPLATE statements:
- generate the same data set name
- (potentially) on the same UNIT or with the same STORAGE CLASS. This situation where TEMPLATE statements from a template switching chain do not ensure that the generated data set are distinct, in terms of data set name or location is not supported. The text message mentions POTENTIALLY if the templates definitions do not allow to conclude if both templates share a common UNIT or STORAGE CLASS. This occurs if either of the following situation occurs:
– one of the indicated TEMPLATE statements does not specify any UNIT or STORAGE CLASS.

– one of the indicated template statements specifies a UNIT but no STORAGE CLASS and the other template specifies a STORAGE CLASS but no UNIT.

**User response:**
Specify either distinct DSN patterns or distinct UNIT/STORAGE CLASS for one of the indicated templates.

**Return Code:**
8

### INZT101I
LISTDEF list-name AT POS(line,column) IS READ FROM LIBRARY ddname

**Explanation**
This is an informational message. A LISTDEF is defined in a LISTDEF library.

**User response:**
No action is required.

**Return Code**
0

### INZT102E
LISTDEF ERROR: NAME list-name IS TOO LONG.

**Explanation**
A LISTDEF name exceeds the 18 character limit.

**User response**
Correct the LISTDEF statement, and resubmit the job.

**Return Code**
8

### INZT103E
LISTDEF ERROR: INVALID DB-SPEC KEYWORD AT POS (line,column)

**Explanation**
An invalid DB-SPEC keyword is used to define a LISTDEF or LISTDEFTBV.

**User response**
Correct the LISTDEF or LISTDEFTBV statement, and resubmit the job.

**Return Code**
8

### INZT104E
LISTDEF ERROR: PARTITION NUMBER OUT OF RANGE FOR PARTLEVEL OPTION

**Explanation**
A partition number in the PARTLEVEL option of the LISTDEF definition is out of range.

**User response**
Correct the LISTDEF statement, and resubmit the job.

**Return Code**
8

### INZT105E
LISTDEF ERROR: INVALID DB2 IDENTIFIER identifier AT POS(line, column)

**Explanation**
An invalid Db2 identifier is used in a LISTDEF definition.

**User response**
Correct the LISTDEF statement, and resubmit the job.

**Return Code**
8

### INZT106E
LISTDEF ERROR: OPTIONS ALL, BASE, LOB, XML ARE EXCLUSIVES

**Explanation**
The ALL, BASE, LOB, and XML options are exclusives in a LISTDEF definition.

**User response**
Correct the LISTDEF statement, and resubmit the job.

**Return Code**
8

### INZT107E
LISTDEF ERROR: DUPLICATE LISTDEF NAME list-name

**Explanation**
A duplicate LISTDEF or LISTDEFTBV name is used in the SYSIN.

**User response**
Correct the LISTDEF or LISTDEFTBV statement, and resubmit the job.

**Return Code**
8
User response
Check the syntax, correct the error, and resubmit the job.

Return Code
8

INZT108E LISTDEF ERROR: LISTDEF list-name AT POS(line, column) IS NOT DEFINED

Explanation
A LISTDEF that was used in the LISTDEF definition is not defined.

User response
Check the name of the included LISTDEF or the name of LISTDEF library that was used, correct the error, and resubmit the job.

Return Code
8

INZT109E LISTDEF ERROR: LISTDEFDD ddname CANNOT BE OPEN

Explanation
The LISTDEFDD library of LISTDEF cannot be opened.

User response
Check the syntax or the JCL for errors, and resubmit the job.

Return Code
8

INZT110E LISTDEF ERROR: TABLESPACES OR INDEXSPACES MUST BE SPECIFIED WITH DATABASE KEYWORD

Explanation
The filter DATABASE in the LISTDEF definition requires an INDEXSPACES or TABLESPACES keyword.

User response
Correct the LISTDEF statement, and resubmit the job.

Return Code
8

INZT111E LISTDEF or LISTDEFTBV ERROR: INVALID PATTERN pattern FOR object-type AT POS (line, column)

Explanation
An invalid pattern is used in the LISTDEF or LISTDEFTBV definition.

User response
Correct the LISTDEF or LISTDEFTBV statement, and resubmit the job.

Return Code
8

INZT112E LISTDEF ERROR: INVALID OPERAND operand

Explanation
An invalid operand is used in a LISTDEF or LISTDEFTBV definition.

User response
Correct the LISTDEF or LISTDEFTBV statement, and resubmit the job.

Return Code
8

INZT113E LISTDEF ERROR: UNSUPPORTED OPTION option

Explanation
An unsupported option is used in the LISTDEF definition.

User response
Correct the LISTDEF statement, and resubmit the job.

Return Code
8

INZU001E SYSIN CANNOT BE OPENED

Explanation
Db2 HPU was unable to open the SYSIN data set.
User response
Ensure that you have a SYSIN DD statement in your JCL. Otherwise, contact IBM Software Support, and supply the return and reason codes.

**Return Code**
16
**INZU002E**  EXECUTION ARGUMENTS MISSING

**Explanation**
The execution arguments are missing.

**User response**
Provide the missing parameters, and resubmit the job.

**Return Code**
16
**INZU003E**  CANNOT DETERMINE THE SSID.

**Explanation**
An error was detected while determining the SSID parameter.

**User response**
Correct the parameters, and resubmit the job.

**Return Code**
16
**INZU005E**  ERROR CONVERTING A CONSTANT IN AN EXPRESSION

**Explanation**
An error occurred while doing operations on a literal in an expression (WHERE clause or other).

**User response**
Check that the literal is correctly specified. Also, ensure that there are no conversion issues. Check to see whether message INZI006E was issued.

**Return Code**
8
**INZU009E**  PART() HAS BEEN SPECIFIED BUT THE OBJECT IS NOT PARTITIONED

**Explanation**
A PART() statement has been detected, but the table space to unload is not partitioned.

**User response**
Correct the SYSIN.

**Return Code**
8
**INZU010I**  ITEM STARTING AT position ENDING AT position

**Explanation**
This is an informational message. Indicates the position in the SYSIN of the lexical or grammatical element that is causing the message that follows this message.

**User response**
See the message that follows this message.

**Return Code**
0
**INZU011E**  PARTITION NUMBER IS OUT OF RANGE

**Explanation**
The partition number that was specified by the PART keyword is greater than the total number of partitions of the table space.

**User response**
Correct the SYSIN and resubmit the job.
INZU012E  CONFLICT BETWEEN SELECT PART() AND UNLOAD PART() SPECIFICATIONS

Explanation
The PART specification of the SELECT statement conflicts with the one in the UNLOAD block. When both are specified, the SELECT PART specification must be a subset of the UNLOAD PART specification.

User response
Correct the SYSIN.

INZU014E  INVALID LEXEME AFTER KEYWORD(S) keyword-name VALID LEXEME(S) IS(ARE) lexeme

Explanation
A lexical element was followed by an invalid value.

User response
Correct the SYSIN and rerun the job.

INZU015E  name STATEMENT UNEXPECTED KEYWORD

Explanation
The end of the specified statement was reached without the presence of a semicolon.

User response
Check that the statement is terminated with a semicolon and that the syntax of the statement is valid.

INZU016E  name STATEMENT SYNTAX IS INVALID

Explanation
The specified statement is coded incorrectly.

User response
Correct the statement syntax; for example, a parenthesis or comma might be missing.

INZU017E  INVALID LEXEME. VALID LEXEME(S) MAY BE xxxxx

Explanation
An invalid or unexpected lexeme was found.

User response
Correct the statement syntax.

INZU018E  INVALID PARMLIB PARAMETER name

Explanation
An error occurred while retrieving the indicated PARMLIB parameter.

User response
Check the PARMLIB.

INZU019E  TABLESPACE dbname.tsname NOT FOUND IN THE DB2 CATALOG

Explanation
The information for the specified table space was not found.

User response
Check that the table space name, if specified in the SYSIN, is correct. Make sure that the catalog information is quiesced in the catalog.
INZU020E  ERROR WHILE RETRIEVING TABLESPACE INFO FROM THE CATALOG (dbname.tsname)

Explanation
An error occurred while retrieving the information from the catalog.

User response
Check the installation. Ensure that the proper authorizations are in place. Contact IBM Software Support, and supply the return and reason codes.

Return Code
8

INZU021E  UNSUPPORTED CONVERSION type TO type

Explanation
An incorrect conversion type was requested.

User response
See “USER format - deprecated” on page 70 for inconsistency between input and output column type. If you specified the LIKE clause, ensure that it is compatible with the input data.

Return Code
8

INZU022E  INVALID LEXEME

Explanation
An invalid or unexpected lexeme was found.

User response
Check the statement syntax.

Return Code
8

INZU023E  INVALID PIC SPECIFICATION.

Explanation
An invalid PIC() specification was found.

User response
Check the statement syntax.

Return Code
8

INZU024E  COLUMN name IS NOT PART OF THE object.

Explanation
The indicated column used in the statement is not a column of the selected object (TABLE or VIEW).

User response
Check the statement syntax.

Return Code
8

INZU025E  INVALID OPERAND FOR OPERATOR name.

Explanation
An invalid operand was found for the specified operator.

User response
Check the statement syntax.

Return Code
8

INZU026E  MISSING PARENTHESIS.

Explanation
A missing parenthesis has been detected.

User response
Check the statement syntax.

Return Code
8

INZU027W  CNTLCARDS DB2 NOT SUPPORTED, WILL USE CNTLCARDS DB2LOAD INSTEAD

Explanation
CNTLCARDS Db2 is not supported. CNTLCARDS DB2LOAD was used instead.
User response
Only the LOAD SYSIN will be generated. To suppress the message, replace Db2 by DB2LOAD.

Return Code
4

INZU028W CNTLCARDS OPTIONS NOT SUPPORTED, OPTION IGNORED

Explanation
CNTLCARDS options are not supported. The option is ignored.

User response
The CNTLCARDS option that was specified is not supported; no SYSCNTL is generated.

Return Code
4

INZU030E OUTPUT DDN NOT FOUND FOR PARTITION number

Explanation
Output ddname was not found for the specified partition.

User response
Correct the syntax.

Return Code
8

INZU035E PARAMETER IS OUT OF RANGE (rangevalue)

Explanation
The parameter is out of range.

User response
The specified parameter is out of range. A valid range might be given as part of the message.

Return Code
8

INZU036E MISSING PARAMETERS FOR name

Explanation
One or more parameters are missing.

User response
Correct the syntax.

Return Code
8

INZU037E IN() CLAUSE MUST CONTAIN CONSTANTS ONLY

Explanation
The two specified keywords or statements are incompatible.

User response
Correct the syntax.
**Explanation**
A variable or an expression has been detected in an IN() clause.

**User response**
Correct the syntax. Change the expression to avoid having a non-constant value in an IN() clause, or authorize the processing by Db2 (DB2 YES, or DB2 FORCE).

**Return Code**
8

**INZU038E** UNSUPPORTED TYPE CONVERSION FOR COLUMN name

**Explanation**
Unsupported type conversion.

**User response**
The requested output type is incompatible with the actual input type. Correct the syntax. Check the LIKE clause and the COLUMN description of the underlying table or expressions.

**Return Code**
8

**INZU039E** FATAL SYNTAX ERROR BEFORE LINE nnnn COLUMN nnnn

**Explanation**
A fatal syntax error was detected. The syntax analysis stops and the program ends.

**User response**
The message contains the end of the last lexeme that was read from the syntax. This is not necessarily the cause for the failure. Check the syntax before that point.

**Return Code**
8

**INZU040E** INVALID ORDER BY: SORT KEY reason

**Explanation**
An ORDER BY clause cannot be processed natively by Db2 HPU for one of the following reasons:

- i IS NOT A COLUMN OF TABLE

Db2 HPU can only issue a sort on columns in the unloaded table, not on expressions. In this case, i is the sequence number of the sort key in the ORDER BY clause.

- **EXCEEDS 4088 BYTES LENGTH**
Because of SORT limitations, the total length of the sort keys cannot exceed 4088 bytes.

**User response**
Modify the ORDER BY clause or use DB2 YES to process the SELECT statement using Db2 access.

**Return Code**
8

**INZU041E** ERROR DURING ALLOCATION OF FIC FOR xxxx DSN: xxxx

**Explanation**
The dynamic allocation of a full image copy for a given table space failed.

**User response**
Make sure that the indicated file is accessible.

**Return Code**
8

**INZU042E** NO GLOBAL FULL IMAGE COPY FOUND FOR TABLESPACE dbname.tsname

**Explanation**
No valid full image copy was found for the specified table space. If the message INZU417I is also issued, see the related explanation for this message for more details on the origin of the issue.

**User response**
Ensure that Db2 can find the full image copy or copies that are required to process the unload request that you specified in the SYSIN. If you are using a partitioned table space, only complete FICs (no partial copies) are considered. Use the PARTITIONED keyword or the ANYTYPE keyword to accept full image copies per partition as input.

**Return Code**
8

**INZU043E** CONVERSION ERROR BETWEEN CCSID ccsid AND CCSID ccsid

**Chapter 8. Troubleshooting** 403
Explanation
Db2 HPU used or might use a conversion between the specified CCSIDs.

User response:
No action is required.

Return Code
8

INZU044W reason FOR TABLESPACE $dbname.tsname ORDER CLUSTER STATEMENT IS IGNORED FOR SELECT select-number

Explanation
An ORDER CLUSTER statement was specified, but is ignored for one of the following reasons:

NO CLUSTER INDEX FOUND
No cluster index can be found for the table in the SELECT statement or for at least one of the tables resulting from a LISTDEF used in the SELECT statement.

CLUSTER INDEX WITH RANDOM KEY
The cluster index defined on the selected table has a random key and cannot be used for sorting the unloaded data.

CLUSTER INDEX KEY NOT SUPPORTED
FORMAT INTERNAL is specified, the table space is in basic row format (BRF), and the cluster index that is defined on the table in the SELECT statement has a column located after the first variable columns of the table.

User response
Remove the ORDER CLUSTER from the SELECT statement and resubmit the job.

Return Code
4

INZU047E INVALID EXECUTION ARGUMENTS

Explanation
The execution arguments are incorrect. This error occurs when the PARM parameter of the EXEC statement does not start with "EP=UTLGLCTL", and the UNLOAD statement specifies FASTUNLOAD.

User response
Correct either the execution arguments or the UNLOAD statement.

Return Code
16

INZU048E WHERE CLAUSE IS NOT BOOLEAN

Explanation
The WHERE clause must evaluate as Boolean.

User response
Correct the WHERE clause so it evaluates to 1 or 0.

Return Code
16

INZU049W DATEFMT date-format FORMAT IS NOT SUPPORTED

Explanation
The date-format that is specified in DATEFMT is not supported. The SSID-default date format will be used.

User response
Change the date format to a compatible format, or do nothing if the Db2 default is acceptable.

Return Code
4

INZU050W TIMEFMT time-format FORMAT IS NOT SUPPORTED

Explanation
The time format that is specified in TIMEFMT is not supported. The SSID-default time format is used.

User response
Change the time format to a compatible format, or do nothing if the Db2 default is acceptable.

Return Code
4

INZU051E DDN ddname TOO LONG

Explanation
The specified ddname is too long.

User response
Choose a shorter ddname. Most situations require a DDN of six characters or less.
Return Code
8

INZU052E    UNSUPPORTED SYNTAX

Explanation
The syntax is not supported. This message is displayed when you use syntax that is compatible with Db2 HPU syntax, but is not syntax supported by Db2 HPU.

User response
Modify the compatible syntax to a supported syntax, or convert the statement into Db2 HPU syntax.

Return Code
8

INZU053W    TSFMT timestamp-format FORMAT IS NOT SUPPORTED

Explanation
The TIMESTAMP format that is specified in TSFMT is not supported. The default TIMESTAMP format of the current Db2 subsystem is used.

User response
Change the TIMESTAMP format to a compatible format, or do nothing if the Db2 default is acceptable.

Return Code
4

INZU055W    OUTPUT-FORMAT LOAD NOT SUPPORTED, DSNTIAUL WILL BE USED INSTEAD

Explanation
FASTUNLOAD LOAD output format is not supported. DSNTIAUL is used instead, and processing continues.

User response:
No action is required.

Return Code
4

INZU056E    TABLE name NOT FOUND IN TABLESPACE tsname

Explanation
The specified table was not found in the table space. A possible explanation is that you specified a SELECT statement with the keyword FROM table, and the user's name is not the same as the table's creator.

User response
Specify a fully qualified name for the table, if necessary.

Return Code
8

INZU057E    EMPTY TABLESPACE name

Explanation
The specified table space did not contain any tables.

User response
The table space that is being unloaded must contain at least one table. Check your object selection in the SYSIN.

Return Code
8

INZU058E    OUTPUT DDN ddname NOT FOUND FOR SELECT number

Explanation
Output ddname not found for the specified select.

User response
Correct the syntax.

Return Code
8

INZU059W    LOADDNN: ddname NOT FOUND. SKIP LOAD STATEMENT GENERATION FOR SELECT STARTING AT POS(line,column)

Explanation
The LOADDNN ddname was not allocated. No load JCL will be generated for any statement that uses this LOADDNN.

User response
Correct the JCL, the syntax, or both.

Return Code
0
**INZU060E** TOO MANY COLUMNS IN INTO CLAUSE FOR SELECT number

**Explanation**

Too many columns were specified in the INTO clause compared to the sum of columns in the SELECT clause.

**User response**

Correct the INTO Clause.

**Return Code**

8

**INZU061E** INVALID OR UNSUPPORTED DATA TYPE SPECIFICATION

**Explanation**

Invalid or unsupported data type specification.

**User response**

Correct the INTO clause.

**Return Code**

8

**INZU063I** UNSUPPORTED SELECT

**Explanation**

The SELECT statement is not supported by Db2 HPU. An attempt will be made to process the SELECT through Db2.

**User response**

No action is required.

**Return Code**

4

**INZU065E** INCONSISTENT PARAMETER LIST

**Explanation**

A multi-parameters function or predicate has inconsistent parameters type.

**User response**

Check the syntax. Change the expression to eliminate the conflict. Use explicit conversion if necessary or possible.

**Return Code**

8

**INZU066E** THE TABLE creator.tbname DOES NOT EXIST

**Explanation**

The specified table has not be found in the Db2 catalog table SYSIBM.SYSTABLES.

**User response**

Correct the syntax to specify the name of a table that can be found in the Db2 catalog, or, if the data is to be unloaded from an image copy, specify the table description with the help of the DDLDDN feature.

**Return Code**

8

**INZU067E** COLUMN FORMAT SPECIFICATION MISSING

**Explanation**

A column format specification is expected after the comma.

**User response**

Correct the syntax.

**Return Code**

8

**INZU068E** INVALID COLUMN FORMAT PARAMETER

**Explanation**

A column format specification contains an invalid parameter.

**User response**

Correct the syntax.

**Return Code**

8

**INZU069E** INVALID COLUMN FORMAT IDENTIFIER

**Explanation**

A column format specification does not describe a proper column name or number.

**User response**

Correct the syntax.
Return Code 8

INZU070E  element MISSING

Explanation
The specified element was expected at this point in the syntax.

User response
Correct the syntax.

Return Code 8

INZU071E  UNUSED FORMAT USER COLUMN SPECIFICATION, COLUMN NAME col_name

Explanation
A column description was given in a FORMAT USER but could not be associated with any column of the SELECT statement.

User response
Verify the SELECT statement syntax.

Return Code 8

INZU072E  UNUSED FORMAT USER COLUMN SPECIFICATION, COLUMN NUMBER col_number

Explanation
A column description was given in a FORMAT USER but could not be associated with any column of the SELECT statement.

User response
Verify the SELECT statement syntax.

Return Code 8

INZU073E  LIKE-TABLE FORMAT FAILED: SQL_order

Explanation
A LIKE formatting clause could not be processed. The SQL statement that was used to extract the LIKE information is given.

User response
Verify that the LIKE table exists.

Return Code 8

INZU074E  INVALID MASK VALUE FOR PIC(): mask_value

Explanation
An invalid value was specified for the mask of a PIC() definition.

User response
Check the PIC() definition.

Return Code 8

INZU075E  UNUSED LIKE COLUMN SPECIFICATION, COLUMN NAME col_name

Explanation
A column description was given in a LIKE clause, but it could not be associated with any column of the SELECT statement.

User response
Verify the SELECT statement syntax.

Return Code 8

INZU076E  UNUSED LIKE COLUMN SPECIFICATION, COLUMN NUMBER col_number

Explanation
A column description was given in a LIKE clause but could not be associated with any column of the SELECT statement.

User response
Verify the SELECT statement syntax.

Chapter 8. Troubleshooting 407
Return Code
8
INZU077E NO OBJECT TO UNLOAD

Explanation
An object (table or table space) to unload was not found in an UNLOAD statement.

User response
Check and modify the UNLOAD and SELECT statements, and resubmit the job.

Return Code
8
INZU078E CASE STATEMENT CANNOT CONTAIN ONLY NULL EXPRESSION

Explanation
At least one result-expression of a case statement must be not null.

User response
Correct the CASE statement.

Return Code
8
INZU079E THE TARGET TABLE table_name HAS AN INVALID NUMBER OF COLUMN

Explanation
The LIKE table has a wrong column number and the unloaded table does not match the LIKE table.

User response
The LIKE table must have more or as many columns than the table to be unloaded.

Return Code
8
INZU080E ERROR : SELECT FROM VIEW 'view_name' IS NOT SUPPORTED BY THE UTILITY

Explanation
The utility does not support the specified view.

User response
See “SELECT block syntax and description” on page 183 for information about defining views.

Return Code
8
INZU081E ERROR : CLAUSE INTO IS NOT ALLOWED WITH OUTPUT-FORMAT LOAD OR COMMA-DELIMITED

Explanation
Fast Unload does not support an INTO clause with OUTPUT-FORMAT LOAD and COMMA-DELIMITED.

User response
Remove the INTO clause.

Return Code
8
INZU082W WARNING : CLAUSE STARTING AT (line, column) ENDING AT (line, column) IS NOT SUPPORTED BY UTILITY

Explanation
An option of Fast Unload or UNLOAD PLUS syntax is not supported by the utility.

User response
Remove the conflicting statement.

Return Code
4
INZU083E FORMAT format IS NOT SUPPORTED

Explanation
The specified format is not supported by Db2 HPU.

User response
Choose another format.

Return Code
8
INZU084W WARNING : LOAD-CONTROL FASTLOAD NOT SUPPORTED, DB2LOAD WILL BE USED INSTEAD
**Explanation**

FASTUNLOAD LOAD-CONTROL FASTLOAD is not supported; DB2LOAD will be used instead.

**User response:**
No action is required.

**Return Code**
4

**INZU085W** WARNING: FIRST ARGUMENT OF SAMPLE STATEMENT IS IGNORED BY UTILITY

**Explanation**

You can use the CA FASTUNLOAD SAMPLE keyword with one or two numeric values. However, Db2 HPU only allows you to use this keyword with one value.

**User response:**
No action is required.

**Return Code**
4

**INZU086E** ERROR: 'ORDER BY' ON EXPRESSION IS NOT SUPPORTED

**Explanation**

It is not possible to sort output data according to the result of expression. Only Db2 columns are allowed in the ORDER BY clause.

**User response**
Either specify a Db2 column in the ORDER BY clause or authorize Db2 processing.

**Return Code**
8

**INZU087E** ERROR: MULTIPLE SELECTS WITH ORDER BY CLAUSE UNLOADING IN THE SAME DDNAME ARE NOT ALLOWED

**Explanation**

Multiple SELECTs on the same table are specified in the same UNLOAD clause. It is not possible to sort the results of multiple SELECT clauses when unloading into the same ddname. Only one single ORDER BY clause is allowed by ddname.

**User response**
Either specify different ddnames for each SELECT with the ORDER BY clause or split the unload in multiple unload clauses.

**Return Code**
8

**INZU088E** ERROR: SELECTS ON MULTIPLE TABLES WITH ORDER BY CLAUSE UNLOADING IN THE SAME DDNAME ARE NOT ALLOWED

**Explanation**

Multiple SELECTs on the same table are specified in the same UNLOAD clause. It is not possible to sort the result of a SELECT on one table and to get the result of a SELECT on another table by using the same ddname.

**User response**
Either specify a different ddname for the SELECT with the ORDER BY clause or split the UNLOAD clause into two UNLOAD clauses; one UNLOAD clause contains the SELECT with the ORDER BY clause and the other UNLOAD clause contains the remaining SELECT clauses.

**Return Code**
8

**INZU089E** ERROR: MULTIPLE SELECTS ON A PARTITIONED TABLESPACE WITH DISTINCT ORDER BY CLAUSE UNLOADING IN THE SAME DDNAME ARE NOT ALLOWED

**Explanation**

Multiple SELECTs on a partitioned table space are specified in the same UNLOAD. It is not possible to have different types of sorts when partitions are unloaded into the same ddname. In this case, all ORDER BY clauses should be the same.

**User response**
Either specify a different ddname for each type of ORDER BY clauses or split the UNLOAD clause into several UNLOAD clauses.

**Return Code**
8
INZU090E UNRECOVERABLE SYNTAX ERROR STARTING AT (line, column) AND ENDING AT (line, column)

Explanation
An unrecoverable syntax error occurred. The incorrect syntax was found within the area indicated by the message, but the exact reason of the error could not be determined.

User response
Check the syntax in the indicated area.

Return Code
8

INZU091E OBID OPTIONS WITHOUT INFILE DDNAME OPTION

Explanation
This error occurs in UNLOAD PLUS syntax when using the OBID option without using INFILE options to specify the IMAGE COPY ddname.

User response
Specify the IMAGE COPY ddname with INFILE options.

Return Code
8

INZU092W OPTION NULLSTRING IS IGNORED BY THE UTILITY

Explanation
The option NULLSTRING (UNLOAD PLUS syntax) used the OBID option without using INFILE options to specify the IMAGE COPY ddname.

User response
Specify the IMAGE COPY ddname with INFILE options.

Return Code
8

INZU093E THE STRING STARTING AT (line, column) MUST BE ONE CHARACTER LONG

Explanation
The specified string must be one character in length.

User response
Replace the string with a single character.

Return Code
8

INZU094E DDNAME ddname NOT FOUND

Explanation
The output ddname that was specified in the SYSIN was not found in the JCL.

User response
Check the output ddname in the JCL.

Return Code
8

INZU095E SYNTAX ERROR - keyword KEYWORD ONLY VALID WITH block BLOCK

Explanation
The specified keyword cannot be used out of the block that is named in the message.

User response
Verify the statement syntax.

Return Code
8

INZU096E REQUIRED KEYWORD, keyword, NOT SPECIFIED, PROCESSING TERMINATED

Explanation
The specified keyword was expected at this point in the syntax.

User response
Verify the statement syntax.

Return Code
8
Explanation
A syntax element cannot be recognized in the specified syntax block. It might be a misspelled keyword or an invalid argument for the current syntax context.

User response
Verify the statement syntax.

Return Code
8

INZU098E  SYNTAX ERROR - OPERAND, operand, IS USED OUT OF CONTEXT

Explanation
Syntax elements, which are usually used as arguments, were found out of context. Arguments are associated with a keyword that precedes the operand.

User response
Verify the statement syntax.

Return Code
8

INZU099E  SYNTAX ERROR - EMPTY OPTIONS BLOCK

Explanation
An options block cannot be empty.

User response
Verify the statement syntax.

Return Code
8

INZU100E  SYNTAX ERROR - UNLOAD KEYWORD MUST BE FOLLOWED BY TABLESPACE KEYWORD

Explanation
The UNLOAD keyword must be followed by the TABLESPACE keyword.

User response
Verify the statement syntax.

Return Code
8

INZU101E  DB2 NO SPECIFIED, PROCESSING TERMINATED

Explanation
An invalid or an unsupported SYSIN was detected. The selected option DB2 NO implied termination of process.

User response
Correct the SYSIN, or try DB2 YES in case of unsupported SYSIN.

Return Code
8

INZU102I  DB2 FORCE SPECIFIED WILL PROCESS THROUGH DB2

Explanation
The DB2 FORCE option was specified. The UNLOAD will be processed through Db2.

User response:
No action is required.

Return Code
0

INZU103W  DB2 YES SPECIFIED, WILL ATTEMPT DB2 PROCESSING

Explanation
An invalid or an unsupported SYSIN was detected. The UNLOAD will be processed through Db2 because the DB2 YES option was specified.

User response:
No action is required.

Return Code
4

INZU104I  WARNING IN SELECT STARTING AT (line, column), keyword AND keyword POINT THE SAME BASE DDNAME: ddname

Explanation
This warning occurs when two of these directives, UNLDDN, OUTDDN, and LOADDN, share the same base ddname.
<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><strong>INZU105I</strong> <em>keyword</em> <strong>OPTION IS IGNORED BY THE UTILITY</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;An unsupported <em>OPTION</em> was encountered for FASTUNLOAD or UNLOAD PLUS syntax.</td>
<td>No action is required.</td>
</tr>
<tr>
<td>0</td>
<td><strong>INZU106E</strong> <strong>THE TABLESPACE TO UNLOAD CANNOT BE DETERMINED</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;Db2 HPU cannot determine which table space to unload.</td>
<td>Specify the name of the table space to be unloaded in the SYSIN.</td>
</tr>
<tr>
<td>8</td>
<td><strong>INZU107E</strong> <strong>UNLDDN NOT SPECIFIED</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;A physical unload without UNLDDN was found.</td>
<td>Specify an UNLDDN clause.</td>
</tr>
<tr>
<td>8</td>
<td><strong>INZU109E</strong> <strong>SELECT STATEMENT STARTING AT (line, col) IS TOO LARGE</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;The size of a SELECT statement is larger than the maximum size that is allowed by Db2.</td>
<td>Reduce the size of the indicated SELECT statement.</td>
</tr>
<tr>
<td>8</td>
<td><strong>INZU110E</strong> <strong>ESCAPE IS NOT ALLOWED WITH MIXED STRING IN LIKE PREDICATE</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;The ESCAPE clause is not allowed in the LIKE predicate for a mixed string.</td>
<td>Correct the SYSIN, and resubmit the job.</td>
</tr>
<tr>
<td>8</td>
<td><strong>INZU111E</strong> <strong>RESULT OF SQL EXPRESSION CONCAT IS TOO LONG</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;The result of the CONCAT SQL expression is too long. The maximum length of the resulting string is 65535.</td>
<td>Modify SQL expression CONCAT.</td>
</tr>
<tr>
<td>8</td>
<td><strong>INZU112W</strong> <strong>STRING IN ITEM STARTING AT (line, columns) IS TOO LONG AND IS TRUNCATED.</strong>&lt;br&gt;&lt;br&gt;Explanation&lt;br&gt;The string that was specified in the SYSIN is too long for the item. The string is truncated to the maximum number of characters and processing continues.</td>
<td>Correct the string length in the SYSIN.</td>
</tr>
</tbody>
</table>
**Explanation**

The CCSID target of the specified output column has been set to 0 by the operator. The column will be unloaded in the source CCSID.

**User response:**
No action is required.

**Return Code**
4

**INZU121W**  
CCSID TARGET OF COLUMN  
col_name IS 0. COLUMN WILL BE UNLOADED IN EBCDIC

**Explanation**

The CCSID target of the pointed output column has been set to 0 by the operator. The column will be unloaded in the source CCSID, which is EBCDIC CCSID due to internal conversion.

**User response:**
No action is required.

**Return Code**
4

**INZU122W**  
charset SUBSYSTEM CCSID IS SET TO 0. NO CONVERSION WILL BE DONE FOR COLUMN col_name

**Explanation**

The CCSID target of the selected output column is set to 0 and comes from subsystem CCSIDs. The column will be unloaded in the source CCSID. The message indicates which CCSID subsystem is set to 0.

**User response**
Check if the requested CCSID conversion can be avoided, modify the SYSIN, and resubmit the job.

**Return Code**
4

**INZU123W**  
charset SUBSYSTEM CCSID IS SET TO 0. COLUMN col_name WILL BE UNLOADED IN EBCDIC

**Explanation**

Check if the requested CCSID conversion can be avoided, modify the SYSIN, and resubmit the job.

**System action**
Modify the installation parameter to get available the CCSID pointed out.

**User response:**
No action is required.

**Return Code**
4

**INZU124E**  
ERROR IN SELECT STARTING AT (line, column), keyword AND keyword POINT THE SAME BASE DDNAME: ddname

**Explanation**

Two of these directives, UNLDDN, OUTDDN, and LOADDN, share the same base ddname.

**User response**
Specify a unique ddname in the SELECT statement.

**Return Code**
8

**INZU127W**  
DIRECT ACCESS TO DB2 CATALOG FAILED: SWITCH TO SQL ACCESS

**Explanation**

This is a warning message that indicates that an error occurred when accessing the Db2 catalog by using the DIRECT ACCESS method. The access method is switched to SQL ACCESS.

**User response**
Contact IBM Software Support to determine the reason of the failure.

**Return Code**
0

**INZU128E**  
SYSTEM ERROR: DB2 CATALOG EXTRACTION FAILED

**Explanation**

The catalog extraction failed.

**User response**
Contact IBM Software Support to determine the reason of the failure.
**Explanation**
An unload from an image copy was performed with a SELECT statement that is not supported by Db2 HPU.

**User response**
Correct the specified SELECT statement.

**Return Code**
12

**INZU130I**  
CHECK FAILED FOR THE FIC FOR name, DEFAULT VALUE WAS TAKEN

**Explanation**
Due to the parameter VUU032/ULCHKCPY in the PARMLIB, a check was done on the COPYDDN parameter, and no valid full image copy was found for the specified table space in the SYSIBM.SYSCOPY table. The FIC was considered as a NON INLINE FIC.

**User response**
Ensure that Db2 can find the full image copy or copies that are required to process the unload request that you specified in the SYSIN. If working on a partitioned table space, only complete FICs are considered (no partial copies).

**Return Code**
4

**INZU131W**  
encoding ccsid_type SUBSYSTEM CCSID IS NOT DEFINED COLUMN column_designation WILL BE UNLOADED IN DEFAULT SBCS CCSID

**Explanation**
The MIXED CCSID target for the column_designation (column name or column number) is not defined and comes from subsystem CCSIDs. The column will be unloaded in the SBCS subsystem CCSID. The message specifies which subsystem CCSID is not defined.

**System action**
None.

**User response**
Verify that the encoding scheme is correctly specified in the UNLOAD command or in the PARMLIB parameter UNLSCHEM. If not, correct it and resubmit the job.

**Return Code**
8

**INZU134E**  
SYNTAX ERROR: INVALID CORRELATION-NAME correlation-name

**Explanation:**
This is a syntax error. An invalid correlation-name is specified in the select statement.

**User response:**
Check the select statement.

**Return Code**
8

**INZU145E**  
ERROR: WRONG CPU NUMBER. EXECUTION STOPPED

**Explanation**
A module could not run on a specific CPU.
User response
This is a new or upgraded CPU. Contact IBM Software Support to obtain a new confidential code.

Return Code
8

INZU146E  ERROR: EXPIRATION DATE EXCEEDED. EXECUTION STOPPED

Explanation
Db2 HPU's expiration date was reached.

User response
Contact IBM Software Support.

Return Code
8

INZU147E  ERROR: DUPLICATE COLUMN column IN ORDER BY CLAUSE

Explanation
Duplicate column is not authorized in an ORDER BY clause.

User response
Check the syntax and remove the duplicate column.

Return Code
8

INZU168E  ERROR WHEN READING DEFAULT VALUE IN CATALOG FOR COLUMN name

Explanation
The default value for the specified column in the Db2 catalog cannot be interpreted.

User response
Check the default value for the specified column in the catalog.

Return Code
8

INZU170E  ERROR: UNABLE TO UNLOAD MORE THAN 1296 REGISTERED IMAGE COPIES IN THE SAME EXECUTION

Explanation
The maximum number of registered image copies that Db2 HPU can unload at one time was reached. Db2 HPU is unable to allocate more than 1296 ddnames.

User response
Split the SYSIN into smaller parts so that the limit is not reached.

Return Code
8

INZU172E  ERROR: UNABLE TO QUIESCE TABLESPACEdbname.tsname

Explanation
QUIESCE YES LOCK YES was specified but Db2 HPU was unable to quiesce the table space because of its status. When the PARMLIB parameter VUU028/ULQSCEBH is TRY, quiesce is not taken if the object status does not allow it. This can happen when another utility is running at the same time.

User response
If the table space is already quiesced, rerun the job with QUIESCE NO. Otherwise, try to manually quiesce the table space.

Return Code
8

INZU176I  INFORMATION: QUIESCE IMPOSSIBLE. A STOP/START SEQUENCE IS PERFORMED ON TABLESPACE ixname.tsname

Explanation
The QUIESCE is impossible on the object because of its status. In this case, the object is stopped and then restarted in order to flush Db2 buffers. The STOP/START sequence is not done when the PARMLIB parameter value of VUU028/ULQSCEBH is TRY.

User response: No action is required.

Return Code
0

INZU180I  UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE PARMLIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN
Explanation
The default UTPRINT DDN is not used when the VUX020/SORTCLAS parameter is specified in the PARMLIB or when SORTCLASS is specified in the SYSIN. If a sort is invoked, a UTPRTnn file is dynamically allocated.

User response:
No action is required.

Return Code
0

INZU184E  ERROR: COLUMN TYPE column type IS NOT SUPPORTED

Explanation
Db2 HPU cannot unload columns of the indicated type. This message occurs when a SELECT statement includes a column with a specific type and the DB2 NO option was specified. The column type is one of the following:

- LOB or XML (IN STREAM): in-stream unloading of LOB or XML data is only supported in SQL mode.
- TIMESTZ: timestamp with time zone is only supported in SQL mode.

Db2 HPU cannot unload column type BLOB, CLOB and DBCLOB. This message occurs when a SELECT STATEMENT includes a column type LOB and SPANNED YES is not used.

User response:
Remove the non supported column from SELECT statement if its selection was not intended or use the DB2 YES or DB2 FORCE option to allow the unloading of this column type. Remove the column type BLOB, CLOB, DBCLOB from the SELECT statement or specify SPANNED YES.

Return Code
8

Example
Unload the whole content of a table having a LOB column in spanned format:

```
UNLOAD TABLESPACE
SELECT *
FROM me.mytable
OUTDDN (SYSREC)
FORMAT VARIABLE ALL
OPTIONS SPANNED YES
LOADDN LOAD
```

INCLUDES UNSUPPORTED COLUMN TYPE

Explanation
The SELECT statement has one or more unsupported column types. Db2 HPU cannot process the statement in any mode (DB2 NO, DB2 YES, DB2 FORCE).

User response
Remove the unsupported column from the SELECT statement, and resubmit the job.

Return Code
8

INZU186E  ERROR: GRAPHIC COLUMNS WITH FIELDPROC ARE NOT SUPPORTED IN ORDER CLAUSE

Explanation
A graphic (GRAPHIC or VARGRAPHIC type) column defined with a FIELDPROC is used in an ORDER clause (ORDER BY or ORDER CLUSTER) and the native mode was requested. Db2 HPU does not support such a request.

User response
As Db2 HPU can process a logical unload with an ORDER BY statement involving a GRAPHIC/ VARGRAPHIC column in SQL mode but cannot process a logical unload with an ORDER CLUSTER clause in SQL mode, you can do the following changes in order to get the expected result. Allow Db2 HPU to use the SQL mode by specifying the DB2 YES or DB2 FORCE option and, if the unsupported ORDER clause is an ORDER CLUSTER clause, replace the ORDER CLUSTER clause by an equivalent explicit ORDER BY clause.

Return Code
8

INZU187E  LOAD STATEMENT ERROR: DUPLICATE DELIMITERS IN FORMAT DELIMITED CHARDEL = 'X'_hexval', COLDEL = 'X'_hexval', DECPT = 'X'_hexval'

Explanation
The LOAD SYSIN GENERATION was requested with DELIMITED output format. You cannot specify the same character for more than one type of delimiter (COLDEL, CHARDEL, and DECPT).
Specify a distinct delimiter in the DELIMITED (SEP, DELIM) format. SEP and DELIM delimiters must be different than the decimal separator.

Return Code
8

INZU188E LOAD STATEMENT ERROR: NO STRING DELIMITERS SPECIFIED

Explanation:
LOAD SYSIN GENERATION was requested with DELIMITED output format requested via the LOADDDN option or COMMA-DELIMITED with Fast Unload, CSV with UNLOAD PLUS and DELIMITED with DB2 UNLOAD. No string delimiters are specified, which causes an error in the LOAD UTILITY. For example, when the string contains delimeter characters. This situation is reported via a warning message because ULRLDRC WARNING applies.

User response:
Specify a string delimiter (option DELIM) to get generated LOAD control cards fully compliant with the LOAD utility.

Return Code: 4

INZU189E LOAD STATEMENT ERROR: INVALID VALUE X 'hexval' FOR _delimiter_type DELIMITER IN FORMAT DELIMITED

Explanation:
LOAD SYSIN GENERATION was requested with DELIMITED output format. An invalid value was specified for a LOAD UTILITY delimiter (CHARDEL, COLDEL, or DECPT). This message indicates the incorrect value and type of delimeter used.

User response:
Change the incorrect value to one that is supported by the LOAD UTILITY.

Return Code: 8

INZU190I IN SELECT POS (line, column), nnnn CONVERSION(S) NO NEEDED DUE TO EXPRESSION IN COLUMN NO number, FROM CCSID csid TO CCSID ccsid

Explanation:
This message is issued when a conversion is requested for a column due to an expression. Such a conversion could decrease unload performance. The column is identified by its number in the SELECT clause. The message indicates how many conversions are requested for the column due to an expression.

User response:
No action is required.
Return Code
0

INZU191I IN SELECT POS (line, column), convname CONVERSION NEEDED FOR COLUMN col_name, FROM CCSID ccsid TO CCSID ccsid

Explanation
This message is issued when a conversion is requested for a column. The purpose of this message is to inform you that such a conversion could decrease unload performance. The column is identified by its name in the SELECT clause.

User response:
No action is required.

Return Code
0

INZU192I IN SELECT POS (line, column), A CONVERSION NEEDED FOR COLUMN NO number, FROM CCSID ccsid TO CCSID ccsid

Explanation
This message is issued when a conversion is requested for a column. The purpose of this message is to inform you that such a conversion could decrease unload performance. The column is identified by its number in the SELECT clause.

User response:
No action is required.

Return Code
0

INZU193I IN WHERE CLAUSE OF SELECT POS (line, column), nnn CONVERSION(S) NEEDED FROM CCSID ccsid TO CCSID ccsid

Explanation
This message is issued when a conversion is requested in a WHERE clause. The purpose of this message is to inform you that such a conversion could decrease unload performance. The message tells you how many conversions are requested due to the WHERE clause.

User response:
No action is required.

Return Code
0

INZU194W FOR SELECT POS (line, column), SYSTEM EBCDIC WILL BE TAKEN AS DEFAULT SCHEME. REQUESTED SCHEME ASIS IS OVERridden.

Explanation
A SELECT was processed by Db2 and no default CCSID can be determined. Default CCSIDs are determined by the scheme that was specified in the PARMLIB (VUU023/UNLSCHM) or in the SYSIN. When the scheme is ASIS, default CCSIDs are set to the ones of the table space of the unload task. When SELECT statements are processed by Db2, Db2 HPU sets the default CCSID to the one of the EBCDIC system scheme. These CCSIDs are used to write items such as NULL and NOT NULL indicator values, separator character and string delimiter for DELIMITED FORMAT and output record padding.

User response
Specify the correct scheme in the SYSIN according to the behavior you require for default CCSIDs.

Return Code
4

INZU195I INFORMATION : detail RC FORCED TO rc DUE TO USER PARAMETER PREFERENCES SEE THE PARMLIB PARAMETER variable

Explanation
This message is issued when the return code is not null (no error). This message informs you that the value of the return code is the value that was specified in the settings. Detail is NO ROW UNLOADED and variable is VUU24/UNLZLRC, if rc indicates that no rows were unloaded.

Details is SWITCH TO SQL MODE and variable is VUU069/SQLSWIRC, if rc indicates that at least one non supported select has been processed in SQL mode. In all other cases, detail is not given.

User response:
No action is required.
INZU200E  LOAD STATEMENT ERROR : DATE FORMAT INVALID FOR COLUMN name

Explanation
This message is issued when LOAD SYSIN GENERATION is requested. Format of DATE type column is not compatible with the format that is allowed in Db2 LOAD.

User response:
Choose an appropriate DATE format for the UNLOAD SYSIN if you want to reload your data.

Return Code
8

INZU201E  LOAD STATEMENT ERROR : TIME FORMAT INVALID FOR COLUMN name

Explanation
This message is issued when LOAD SYSIN GENERATION is requested. Format of TIME type column is not compatible with the format that is allowed in Db2 LOAD.

User response
Choose an appropriate TIME format for the UNLOAD SYSIN if you want to reload your data.

Return Code
8

INZU202E  LOAD STATEMENT ERROR : TIMESTAMP FORMAT INVALID FOR COLUMN name

Explanation
This message is issued when LOAD SYSIN GENERATION is requested. The format of TIMESTAMP type column is not compatible with the format that is allowed in Db2 LOAD.

User response
Choose an appropriate TIMESTAMP format for the UNLOAD SYSIN if you want to reload your data.

Return Code
8

INZU203E  LOAD STATEMENT ERROR : COLUMN VARIABLE WITHOUT HEADER COLUMN name

Explanation
This message is issued when LOAD SYSIN GENERATION is requested. It is not possible to load a variable column without a length header.

User response
Choose an appropriate format for the UNLOAD SYSIN if you want to reload your data.

Return Code
8

INZU204E  LOAD STATEMENT ERROR: COLUMN col_name WITH CCSID ccsid_col DIFFERENT FROM LOAD CCSID ccsid_load

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. The CCSID of the unloaded column is different from the CCSID used to load data. This situation is reported via an error message because ULRLDRC REJECT applies. However, the unload job continues and produces the data set containing the unloaded data and the related LOAD control cards.

User response:
Use a single CCSID for a same UNLOAD file to get generated LOAD control cards fully compliant with the LOAD utility or specify ULRLDRC IGNORE so that this situation is only reported via an informational message and a return code of 0.

Return Code:
8

INZU204I  LOAD STATEMENT INFO: COLUMN col_name WITH CCSID ccsid_col DIFFERENT FROM LOAD CCSID ccsid_load

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. The CCSID of the unloaded column is different from the CCSID used to load data. This situation is reported via an informational message only because ULRLDRC IGNORE applies.

User response:
Use a single CCSID for a same UNLOAD file to get generated LOAD control cards fully compliant with the LOAD utility.
Return Code: 0

INZU204W LOAD STATEMENT WARNING: COLUMN col_name WITH CCSID ccsid_col DIFFERENT FROM LOAD CCSID ccsid_load

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. The CCSID of the unloaded column is different from the CCSID used to load data. This situation is reported via a warning message because ULRLDRC WARNING applies.

User response:
Use a single CCSID for a same UNLOAD file to get generated LOAD control cards fully compliant with the LOAD utility or specify ULRLDRC IGNORE so that this situation is only reported via an informational message and a return code of 0.

Return Code: 4

INZU205W LOAD STATEMENT WARNING: STRING DATA ARE PADDED FOR COLUMN name

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. Unloaded data is padded, so data might be different from its original values.

User response:
To prevent data modification, do not use the option PADDING.

Return Code: 4

INZU206E LOAD STATEMENT ERROR: REAL LENGTH COLUMN WITH NULL INDICATOR AT END ARE NOT SUPPORTED, COLUMN name.

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. Db2 HPU cannot load a real length column with a null indicator at the end. This situation is reported via a warning message because ULRLDRC WARNING applies.

User response:
Specify a NULL indicator at the beginning to reload such a column.

Return Code: 12

INZU209E LOAD STATEMENT ERROR: INCOMPATIBLE NULL POSITION FOR COLUMN name

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. Db2 HPU cannot load a nullable column with a null ID at the end with the variable format (if the file contains variable columns).

User response:
Specify the option NULLPOS BEFORE for such a format.

Return Code: 12

INZU211I LOAD STATEMENT WARNING: UNABLE TO DETERMINE TABLE NAME FOR SELECT STARTING AT POS (line, column). LOAD STATEMENT MUST BE MODIFIED

Explanation:
This message is issued when LOAD statement generation (LOADDN option) is requested. As it is not possible to determine the target table name to be used for the LOAD statement, it is substituted with xxxxxxx.xxxxxx.

User response:
Modify the generated LOAD statement to specify the table name to LOAD or consider specifying an INTO TABLE clause so that the generated LOAD statement can be used as generated.

Return Code: 0

INZU212E LOAD STATEMENT ERROR: COLUMN col_name IS A CONSTANT. LOAD STATEMENT MUST BE MODIFIED TO BE USABLE.

Explanation:
This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded is a constant column and does not match a column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via an error message because ULRLDRC REJECT applies. However, the unload job continues and produces the data set
containing the unloaded data and the related LOAD control cards.

**User response:**
Modify the generated LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control cards compliant with the LOAD utility or specify ULRLDRC IGNORE so that this situation is only reported via an informational message only and a return code of 0.

**Return Code:**
8

INZU212I LOAD STATEMENT INFO: COLUMN col_name IS A CONSTANT. LOAD STATEMENT MUST BE MODIFIED TO BE USABLE.

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded is a constant column and does not match a column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via an informational message only because ULRLDRC IGNORE applies.

**User response:**
Modify the generated LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control cards compliant with the LOAD utility.

**Return Code:**
0

INZU212W LOAD STATEMENT WARNING: COLUMN col_name IS A CONSTANT. LOAD STATEMENT MUST BE MODIFIED TO BE USABLE.

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded is a constant column and does not match a column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via a warning message because ULRLDRC WARNING applies.

**User response:**
Modify the generated LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control cards compliant with the LOAD utility.

**Return Code:**
4

INZU213E LOAD STATEMENT ERROR: COLUMN col_name RESULTS FROM A SQL EXPRESSION. LOAD STATEMENT MUST BE MODIFIED TO BE USABLE.

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded results from an SQL expression and does not match a column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via an error message because ULRLDRC REJECT applies. However, the unload job continues and produces the data set containing the unloaded data and the related LOAD control cards.

**User response:**
Modify the LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control card compliant with the LOAD utility or specify ULRLDRC IGNORE so that this situation is only reported via an informational message only and a return code of 0.

**Return Code:**
8

INZU213I LOAD STATEMENT INFO: COLUMN col_name RESULTS FROM A SQL EXPRESSION. LOAD STATEMENT MUST BE MODIFIED TO BE USABLE.

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded results from an SQL expression and does not match a column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via an informational message only because ULRLDRC IGNORE applies.

**User response:**
Modify the LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control card compliant with the LOAD utility.

**Return Code:**
0

INZU213W LOAD STATEMENT WARNING: COLUMN col_name RESULTS FROM A SQL EXPRESSION. LOAD STATEMENT MUST BE MODIFIED TO BE USABLE.

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded results from an SQL expression and does not match a column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via a warning message because ULRLDRC WARNING applies.

**User response:**
Modify the LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control card compliant with the LOAD utility.

**Return Code:**
4
column from the table to load. The LOAD statement is not usable as it is specified. This situation is reported via a warning message because ULRLDRC WARNING applies.

**User response:**
User response: Modify the LOAD statement before using it with the LOAD utility or add an INTO clause that matches the UNLOAD command to get LOAD control card compliant with the LOAD utility or specify ULRLDRC IGNORE so that this situation is only reported via an informational message only and a return code of 0.

**Return Code:**
4

**INZU214E** LOAD STATEMENT ERROR: FAILED TO access FILE ddname

**Explanation:**
An error occurred accessing to the LOADDN dataset specified by the 'ddname'. The access type can be 'OPEN', 'WRITE INTO' or 'CLOSE'.

**Return Code:**
8

**Action:**
Check system messages to determine the reason of the problem. If the access type is OPEN, check if several distinct ddnames do not allocate the same data set. In this case, use allocate the data set with a single DD statement and refer to its ddname in any LOADDN option you wish.

**INZU215E** LOAD STATEMENT ERROR: INVALID CONVERSION FOR COLUMN col_name

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. An invalid conversion is requested between the data type of the column in the unloaded file and the data type of the column in the table. The LOAD SYSIN cannot be generated.

**User response**
Modify the output field type of the UNLOAD SYSIN.

**Return Code:**
8

**INZU216I** LOAD STATEMENT CAN NOT BE GENERATED IN FORMAT format

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. The LOAD SYSIN cannot be generated in format CSV or XML (UNLOAD PLUS syntax).

**User response:**
No action is required.

**Return Code:**
0

**INZU217E** LOAD STATEMENT ERROR: UNABLE TO COMPUTE keyword FOR SELECT AT Pos (line, column) keyword IS SET TO 0

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. Option SORTKEYS or NUMRECS is requested by an unsupported SELECT statement and cannot be computed or EXECUTE NO is specified for this unload. The default SORTKEYS or NUMRECS value is 0. This situation is reported via an error message because ULRLDRC REJECT applies. However, the unload job continues and produces the data set containing the unloaded data and the related LOAD control cards.

**User response:**
If necessary, manually change the SORTKEYS or NUMRECS value in LOAD SYSIN before using it with the LOAD utility.

**Return Code:**
8

**INZU217I** LOAD STATEMENT INFO : UNABLE TO COMPUTE keyword FOR SELECT AT POS (line, column) keyword IS SET TO 0

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. Option SORTKEYS or NUMRECS is requested by an unsupported SELECT statement and cannot be computed or EXECUTE NO is specified for this unload. The default SORTKEYS or NUMRECS value is 0. This situation is reported via an informational message only because ULRLDRC IGNORE applies.

**User response:**
If necessary, manually change the SORTKEYS or NUMRECS value in LOAD SYSIN before using it with the LOAD utility.

**Return Code:**
0
**INZU217W**  LOAD STATEMENT WARNING: UNABLE TO COMPUTE **keyword** FOR SELECT AT POS (line, column) **keyword** IS SET TO 0

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. Option SORTKEYS or NUMRECS is requested by an unsupported SELECT statement and cannot be computed or EXECUTE NO is specified for this unload. The default SORTKEYS or NUMRECS value is 0. This situation is reported via a warning message only because ULRLDRC WARNING applies.

**User response:**
If necessary, manually change the SORTKEYS or NUMRECS value in LOAD SYSIN before using it with the LOAD utility or specify ULRLDRC IGNORE so that this situation is only reported via an informational message and a return code of 0.

**Return Code:**
4

**INZU218I**  UNABLE TO GENERATE LOAD STATEMENT FOR DELIMITED FORMAT IN SELECT STARTING AT (line, column)

**Explanation:**
This message is issued when LOAD SYSIN GENERATION is requested. In format COMMA-DELIMITED, syntax FAST UNLOAD, the LOAD SYSIN cannot be generated.

**User response:**
No action is required.

**Return Code**
0

**INZU219I**  PTFLEVEL=aaaaaa-nnnnn

**Explanation:**
This is an informational message that displays the number of the latest APAR (nnnnn) that was applied on the INZUTILB module. The nnnnn value, if any, is intended for internal IBM use only.

**User response:**
No action is required.

**Return Code**
0

**INZU220E**  THE SETTINGS FOR DB2 db2id IN THE PARMLIB ARE INCOMPLETE OR ARE MISSING

**Explanation:**
Db2 HPU could not find any value of VZD007/DSNEXIT associated with the subsystem or group name Db2 ID that is specified within the PARM of the EXEC card.

**User response:**
Check that the Db2 ID that is specified within the PARM of the EXEC card is a valid Db2 subsystem or group name. Verify that one of the VZD001 parameters was set to Db2 ID and that the VZD007/DSNEXIT associated parameter is present.

**Return Code**
8

**INZU221I**  PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS SET TO n

**Explanation:**
Parallelism of LDS processing was activated by indicating the parallelism degree. This message is issued when unloading a partitioned table space with parallelism or when several SELECT statements that were processed by using SQL are specified in the same UNLOAD block.

**User response:**
No action is required.

**Return Code**
0

**INZU222I**  ddname, TOTAL NUMBER OF RECORDS WRITTEN n

**Explanation:**
This message indicates the total number of records that were written into an output file.

**User response:**
No action is required.

**Return Code**
0

**INZU223I**  PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF
PARALLELISM IS LIMITED TO \( n \) BY MAXSORT PARAMETER

**Explanation**
Parallelism of LDS processing was activated by indicating the parallelism degree. A parallelism parameter was specified in SYSIN but could not be used because SORT had to be activated. The maximum degree of parallelism was then limited by parameter MAXSORT.

**User response:**
No action is required.

**Return Code**
0

**INZU224I** IBM DB2 HIGH PERFORMANCE UNLOAD Vn.n

**Explanation**
This informational message contains the Db2 HPU version and release number.

**User response:**
No action is required.

**Return Code**
0

**INZU225I** PARALLELISM ACTIVATED FOR TABLESPACE PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS SET TO \( n \)

**Explanation**
Parallel processing of the table space was activated because a parallelism degree was specified in the PARALLELISM option or in the PARMLIB parameter VUX030/UTLPARAL. This message is issued when multiple table spaces with parallelism are unloaded.

**User response:**
No action is required.

**Return Code**
0

**INZU226I** PARALLELISM ACTIVATED FOR SELECT PROCESSED BY DB2. MAXIMUM DEGREE OF PARALLELISM IS SET TO \( n \)

**Explanation**
Parallel processing for Db2-processed SELECT statements was activated because a parallelism degree was specified in the PARALLELISM option or in the PARMLIB parameter VUX031/DB2PARAL. This message is issued when multiple SELECT statements are processed by Db2.

**User response:**
No action is required.

**Return Code**
0

**INZU227I** OPTION option value FOR FORMAT format WILL BE IGNORED BY UTILITY

**Explanation**
The option for the SELECT statement is ignored for the specified format.

**User response:**
No action is required.

**Return Code**
0

**INZU228E** text EXPRESSION IS NOT SUPPORTED BY UTILITY

**Explanation:**
The SELECT that is coded in SYSIN contains an SQL expression (text) which is not natively supported by Db2 HPU, and the SYSIN specified DB2 NO. The SELECT cannot be processed natively.

**User response:**
Specify DB2 YES in SYSIN and rerun the job.

**Return Code**
8

**INZU230E** LOAD STATEMENT ERROR: OPTIONS list of options ARE NOT COMPATIBLE

**Explanation**
Incompatible load options are specified for LOAD SYSIN GENERATION.

**User response**
Modify the LOAD options.
INZU231E COPYDDN CHECK ERROR: GENERIC DDN CHECK IS NOT SUPPORTED

Explanation
The COPYDDN option CHECK is used with a generic ddname (for example, DDNIN*).

User response
Remove the CHECK option and specify that the image copy is INLINE by using the INLINE keyword.

INZU234E COPYDDN NOT FOUND FOR PARTITION i

Explanation
When unloading a partitioned image copy with parallelism, this message indicates that the selected partition has no COPYDDN in JCL.

User response
Allocate the missing COPYDDN for the indicated partition in the JCL and rerun the job.

INZU232E PARAMETER IS INCONSISTENT WITH ITEM AT (line, column) ENDING AT (line, column)

Explanation
The parameter that was specified in the previous message has a type that is inconsistent with the parameter that is specified in this message.

User response
Change the syntax to eliminate the conflict. Be as specific as possible.

INZU237I IN unload / select n PARTITION p CANNOT BE UNLOADED IN DDN ddnx dp DDN ddnx dp IS ALSO USED AS A BASE DDN IN select/unload n PARTITION WILL BE UNLOADED IN BASE DDN ddn

Explanation:
A suffixed DDN that is used to unload a partition with parallelism is also being used as a base DDN by either a SELECT or an UNLOAD statement. To avoid mixing of data, the partition is unloaded in the base DDN instead of the suffixed DDN.

User response:
Check the output data distribution to verify that it is correct. If necessary, change the base ddnames to avoid mixing the ddnames.

INZU233W LOAD STATEMENT WARNING: PARMLIB PARAMETER VUU039/UNLLDER IS SET TO IGNORE. LOAD STATEMENT ERRORS FOR SELECT STARTING AT (line, column) ARE IGNORED. LOAD STATEMENT IS BYPASSED

Explanation
A LOADDOWN keyword requested a LOAD SYSIN generation but an error prevents it from generating a correct LOAD statement for one specific SELECT because PARMLIB parameter VUU039/UNLLDER is set to IGNORE. Processing continues.

User response:
No action is required.

INZU238E UNEXPECTED UNSUPPORTED SELECT UTILITY CANNOT SWITCH TO SQL PROCESSING REASON return_code / reason_code

Explanation
A supported SELECT statement cannot be processed natively by Db2 HPU.

• condition 1) If the return_code is 0X30000, this means the SELECT statement involves a too complex SQL expression or a too complex WHERE clause to be processed in native mode.

• condition 2) For any other return_code or if no complex SQL expression or WHERE clause is involved, this might be an internal error.
If condition 1 applies, consider running the unload in SQL mode by specifying `DB2 FORCE` or try to reword the SQL expression or the `WHERE` clause to make it simpler, if possible. For example, if the `WHERE` clause involved a very big number of `MYCOL = myvalue` predicates combined with OR, consider replacing it by a single IN predicate.

If condition 2 applies, or if the solution for condition 1 does not solve the issue, contact IBM Software Support, and provide the content of the PARMLIB, including all members or at least the INZUTIL member and the INZDSSID member, where SSID is the name of the Db2 subsystem or group attach name on which the issue occurs, the complete job log, and the DDL of the unloaded object.

### Return Code

**8**

**INZU239I**  UNLOAD OF TABLESPACE *dbname.tsname* IS SPLIT DUE TO THE OPTION MAXPART SET TO *n*.

**Explanation:**
Db2 HPU cannot perform an ORDER BY or ORDER CLUSTER on a partitioned table because of the MAXPART parameter. The MAXPART parameter is allowed only with an ORDER BY statement when each partition is unloaded in a separate output file.

**User response:**
Modify the SYSIN by either removing the ORDER BY or MAXPART specification, or specify an output file per partition, and rerun the job.

**Return Code**
None.

**INZU240I**  PARMLIB VARIABLE VUU058/ULIDENT SET TO 'IGNORE' DUE TO DB2 VERSION PRIOR TO V9.

**Explanation:**
The PARMLIB parameter VUU058/ULIDENT contains OVERRIDE, which can only be accepted in Db2 Version 9 or later. This parameter is changed to IGNORE for the current execution.

**User response:**
No action is required.

**Return Code**
None.

**INZU241I**  DB2 HPU IS NOT RUNNING UNDER KEY 7.

**Explanation:**
The PSW key of the utility is not 7. This occurs because Db2 HPU was not invoked via a JCL EXEC statement with `PGM=INZUTILB`. In such a case, restrictions apply. An ABEND can occur if the unloaded table is defined with a procedure, EDITPROC or FIELDPROC, which needs to run in key 7. If no ABEND occurs in such a case, the data is correctly unloaded.

**User response:**
If a subsequent ABEND occurs and the unload table involves an EDITPROC or a FIELDPROC, consider invoking Db2 HPU via a JCL EXEC card.

**Return Code**
None.

**INZU242E**  IN SELECT STARTING AT (line, column) TABLESPACE *ts_name.db_name* MAXPART *n* AND ORDER BY REQUIRE AN OUTPUT FILE PER PARTITION.

**Explanation:**
Db2 HPU cannot perform an ORDER BY or ORDER CLUSTER on a partitioned table because of the MAXPART parameter. The MAXPART parameter is allowed only with an ORDER BY statement when each partition is unloaded in a separate output file.

**User response:**
Modify the SYSIN by either removing the ORDER BY or MAXPART specification, or specify an output file per partition, and rerun the job.

**Return Code**
8
User response
Correct the constant in the SYSIN, and resubmit the job.

Return Code
8

INZU251E INVALID SEMI-COLON BETWEEN SELECT STATEMENT

Explanation
An unauthorized comma was found between SELECT statements.

User response
Correct the SYSIN, and resubmit the job.

Return Code
8

INZU252W LOAD STATEMENT WARNING: INDDN CANNOT BE DETERMINED FOR SELECT STARTING AT (line, column) DEFAULT SYSREC IS USED

Explanation
The output data ddname cannot be determined. This message occurs when LOAD SYSIN GENERATION is requested. This situation can happen when using a TEMPLATE for the output data set.

User response:
No action is required.

Return Code
4

INZU255E ESTIMATION OF SPACE ALLOCATION FOR TEMPLATE template-name FAILED. UTILITY FAILED TO ESTIMATE SIZE OF TABLESPACE dbname.tsname

Explanation
Db2 HPU needs to estimate space allocation for output files that were allocated by using a TEMPLATE statement. This error occurs when the estimates for the space that is required to unload the table fails.

User response
Specify the SPACE parameter in the TEMPLATE definition, and resubmit the job.

Return Code
12

INZU256I PROCESSING UNLOAD FROM LISTDEF list-name

Explanation
This informational message provides the LISTDEF name when an UNLOAD statement is generated from a LISTDEF.

User response:
No action is required.

Return Code
0

INZU257I GENERATING STATEMENTS FROM SELECT STARTING AT POS(line, column) USING LISTDEF list-name

Explanation
This informational message provides the LISTDEF name and the SELECT position in the SYSIN when a SELECT statement is generated from a LISTDEF.

User response:
No action is required.

Return Code
0

INZU259E LOAD STATEMENT ERROR: GENERATION FAILED FOR SELECT STARTING AT POS(line, column)

Explanation
A LOAD statement generation has failed. Subsequent messages will provide more details about the cause of the failure.

User response
Refer to the subsequent messages for more information about the cause of the error.

Return Code
8

INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(line, column)
Explanation
This informational message is issued during the generation of a load statement that corresponds to a LOADDDN keyword.

User response:
No action is required.

Return Code
0

INZU261I PROCESSING GLOBAL LOAD STATEMENT

Explanation
This informational message is issued when generation of a GLOBAL LOAD statement begins. A GLOBAL LOAD statement occurs when several LOAD statements that correspond to several SELECT statements are written in the same output file. Subsequent messages provide more information about which SELECT statements are being written.

User response:
No action is required.

Return Code
0

INZU262I UNLOAD nnnnn: TABLESPACE dbname.tsname

Explanation
This is an informational message. A physical table space UNLOAD that uses an UNLDDN keyword was requested.

User response:
No action is required.

Return Code
0

INZU263I SELECT nnnnn

Explanation
This informational message indicates the SELECT statement that informational or error messages will be issued for. Subsequent messages provide additional information. The sequential number corresponds to the occurrence of the SELECT statement in an UNLOAD statement.

User response:
No action is required.

Return Code
0

INZU264E UNSUPPORTED SELECT WITH LISTDEF list-name

Explanation
A LISTDEF and a SELECT statement are not compatible. Some options are restricted when you use a LISTDEF in a SELECT statement. For example, a WHERE clause is not allowed. Refer to “Fast select, fast listdef and fast listdeftbv select blocks syntax and description” on page 188 for more information. This message indicates the name of the LISTDEF. Message INZU010I precedes this message and contains the position of the SELECT.

User response
Modify the SELECT statement, and resubmit the job.

Return Code
8

INZU265W TEMPLATE OPTIONS LIMIT IS IGNORED BY UTILITY

Explanation
This message indicates that the LIMIT option specified for a TEMPLATE statement was ignored by Db2 HPU.

User response:
Remove the LIMIT option from template to avoid warning or add TEMPLATE_LIMIT(YES) to your setting of the VUU030/ULOPTNS parmlib parameter to enable this feature if a Db2 HPU statement is used.

Return Code
4

INZU266E INVALID ORDER BY IN SELECT FROM A LISTDEF

Explanation
A LISTDEF is used in a SELECT statement that specifies an ORDER BY clause. When using a LISTDEF in a SELECT statement, you can use an ORDER CLUSTER clause, but not an ORDER BY clause.

User response
Modify the SELECT statement, and resubmit the job.

Return Code
8
Explanation
The PART keyword of the UNLOAD statement has no effect in SELECT or UNLOAD statements that use a LISTDEF. To control partition unloading, use the PARTLEVEL option of the LISTDEF.

User response:
No action is required.

Return Code
0

Explanation
The UNLDDN keyword was used without specifying a table space name or a LISTDEF.

User response:
Specify a table space name or a LISTDEF in the UNLOAD statement.

Return Code
8

Explanation
An undefined variable is used in a DSN TEMPLATE expression.

User response
Correct the variable name or define it with the TEMPLATESET option.

Return Code
8

Explanation
Db2 HPU cannot process physical UNLOAD and SELECT statements when they both use a LISTDEF in the same UNLOAD command.

User response
Specify the ORIGINOBID in the SELECT statement, and resubmit the job.
INZU274I  UNLOAD nnnnn IS PROCESSED THROUGH DB2

Explanation
This is an informational message. A SELECT statement is processed through Db2. The UNLOAD number nnnnn corresponds to the occurrence of the UNLOAD statement in the SYSIN.

User response:
No action is required.

Return Code
8

INZU275I  SELECT sssss IN UNLOAD uuuuu IS PROCESSED THROUGH DB2

Explanation
This is an informational message. A SELECT statement is processed through Db2. The UNLOAD number uuuuu corresponds to the occurrence of the UNLOAD statement in the SYSIN, and the SELECT number sssss corresponds to the occurrence of the SELECT statement in the UNLOAD.

User response:
No action is required.

Return Code
0

INZU276E  SPACE ALLOCATION IS REQUIRED IN TEMPLATE template-name

Explanation
Db2 HPU needed to evaluate the amount of data to be stored in the associated data set in order to set the allocation because the SPACE option was not specified in the template definition. However, Db2 HPU could not determine the expected amount of data. This error usually occurs when Db2 HPU processes a logical unload that includes an unsupported SELECT statement.

User response
Check the template definition, define the space allocation, and resubmit the job.

Return Code
8

INZU277I  PROCESSING UNLOAD nnnnn FROM TABLESPACE dbname.tsname

Explanation
This is an informational message. Db2 HPU is processing the dbname.tsname table space unload.

User response:
No action is required.

Return Code
0

INZU278I  UNLOAD STATEMENT FROM SYSIN STARTING AT POS(line, column)

Explanation
This is an informational message. A physical unload (using UNLDDN) was requested from a table space name that is explicitly specified in the SYSIN.

User response:
No action is required.

Return Code
0

INZU279I  SELECT statements using single table specification

Explanation
This is an informational message. Db2 HPU is processing a SELECT statement from a table name that is explicitly specified in the SYSIN. Message INZU280I follows this message.

User response:
No action is required.

Return Code
0

INZU280I  SELECT nnnnn starting at POS(line, column)

Explanation
This is an informational message. Db2 HPU is processing a SELECT statement from a table name that is explicitly specified in the SYSIN.

User response:
No action is required.
INZU281I  UNLOAD STARTING AT POS(line, column)

Explanation
This informational message provides the UNLOAD command position in the SYSIN. This message is issued for each UNLOAD command that is coded in the SYSIN.

User response:
No action is required.

INZU282I  OUTPUT DDNAME=ddname

Explanation
A JCL allocated ddname is used to process a physical or logical unload. This message is issued after message INZU280I or INZU278I.

User response:
No action is required.

INZU283E  ERROR IN TEMPLATE LIBRARY ddname

Explanation
A TEMPLATE library with an incorrect or unexpected content was provided to Db2 HPU. Subsequent messages may provide additional details. The process continues as the a TEMPLATE Library is not mandatory.

User response
Refer to the subsequent messages to determine the reason of the failure or check the TEMPLATE library to see if it has an obviously unexpected content such as no TEMPLATE keyword at all. If the TEMPLATE library is actually needed for the unload statement, correct the error, otherwise consider to simply remove the TEMPLATE library.

Return Code
0

INZU284E  ERROR IN LISTDEF LIBRARY ddname

Explanation
An error occurred when parsing a LISTDEF library. Subsequent messages provide the reason of failure.

User response
Refer to the subsequent messages to determine the reason of the failure.

Return Code
8

INZU287I  NO TABLE FOUND IN LISTDEF list-name FOR SELECT STARTING AT POS(line, column)

Explanation
A SELECT statement from a LISTDEF or LISTDEFTBV is coded in the SYSIN, but the generated list does not contain a table. The SELECT statement that uses this LISTDEF or LISTDEFTBV cannot generate any SELECT statements.

User response:
No action is required.

Return Code
0

INZU288I  NO TABLESPACE FOUND IN LISTDEF list-name FOR UNLOAD STARTING AT POS(line, column)

Explanation:
A physical UNLOAD statement that uses a LISTDEF is coded in the SYSIN, but the generated list does not contain a table space with at least one table. The UNLOAD statement that uses this LISTDEF does not generate a physical UNLOAD.

User response:
No action is required.

Return Code
0

INZU289I  PROCESSING UNLOAD nnnnn FROM UNDETERMINED TABLESPACE

Return Code
8
Explanation
An unsupported SELECT statement is processed through Db2. This message is followed by one INZU280I message for each unsupported SELECT statement. All unsupported SELECT statements are processed with the same unload number.

User response:
No action is required.

Return Code
0

INZU280I TABLESPACE UNLOADED FROM IMAGE COPY

Explanation
No action is required.

Return Code
0

INZU290I TABLESPACE UNLOADED FROM IMAGE COPY

Explanation
An UNLOAD is being performed from an image copy. A subsequent message provides the file name of the image copy that was used.

User response:
No action is required.

Return Code
0

INZU291I TABLESPACE UNLOADED FROM LAST IMAGE COPY

Explanation
An UNLOAD is performed from the most recent full image copy of the table space that was found in the SYSIBM.SYSCOPY when COPYDDN LAST_IC was used in the SYSIN.

User response:
No action is required.

Return Code
0

INZU292I TABLESPACE UNLOADED FROM IMAGE COPY n

Explanation
An UNLOAD is performed from the nth most recent full image copy of the table space that was found in the SYSIBM.SYSCOPY, when COPYDDN n is used in the SYSIN.

User response:
No action is required.

Return Code
0

INZU293E UNLOAD FROM IMAGE COPY STARTING AT POS(line, column). NO OUTPUT DDNAME FOUND FOR PARTITION n

Explanation
An output file is missing that is needed to unload a partitioned image copy with parallelism. The partition for which the ddname is identified. This type of unload requires one distinct file for each unloaded partition.

User response
Add the missing DDNAME.

Return Code
8

INZU297I LOGICAL RECORD LENGTH OF THE OUTPUT FILE name IS SET TO 32756. RECORDS MIGHT BE TRUNCATED

Explanation
The maximum length of the output record exceeds the physical limit. PARMLIB parameter VU020/ULTR32K is set to YES and the output file record format is variable.

User response:
No action is required.

Return Code
0

INZU298E LOGICAL RECORD LENGTH OF THE OUTPUT FILE ddname IS lrecl AND EXCEEDS THE PHYSICAL LIMIT

Explanation
The maximum length of the output record exceeds the physical limit. This problem usually occurs when a fixed output format is used and when unloading tables in 32KB-page table space.

User response
Use one of the following methods to reduce the length of the output record:
• Use a variable format (VARIABLE ALL, VARIABLE END, DELIMITED).
• Reduce the length of larger columns by forcing the length in USER format or by using the SUBSTR() function.
• Unload only the necessary columns.

Return Code
8

INZU299E DUPLICATE COLUMN DEFINITIONS. USE EITHER INTO-CLAUSE, FORMAT USER OR LIKE CLAUSE TO SPECIFY COLUMN OUTPUT FORMAT.

Explanation
Output column definitions cannot be specified in both the INTO clause and FORMAT USER clause or in the INTO clause with the LIKE clause.

User response
Suppress one of the format definitions, and resubmit the job.

Return Code
8

INZU304E UNABLE TO RETRIEVE TABLESPACE OR DATABASE CCSID FROM THE DB2 CATALOG

Explanation
The table space or database CCSID retrieval failed.

User response
Check the SQLCODE and SQL messages in the job.

Return Code
8

INZU305E UNSUPPORTED ROW_VALUE_EXPRESSION

Explanation:
A row_value_expression occurs in a SELECT statement, and DB2 NO is used in the UNLOAD command. Db2 HPU does not natively support the row_value_expression.

User response:
Specify DB2 YES or DB2 FORCE, or modify the expression and rerun the job.

Return Code
8

INZU306E INPUT IMAGE COPY IS NOT SUPPORTED WHEN UNLOADING COLUMN TYPE LOB OR XML

Explanation:
An unload from COPY with an input DDNAME specified could not be performed because an XML or LOB column is involved. In such a case, only the LAST_IC keyword is supported by Db2 HPU. The COPYDDN dname option was specified. In such a case and after maintenance for MR0411126733, only the LAST_IC keyword is supported by Db2 HPU.

User response:
Specify COPYDDN LAST_IC or COPYDDN -1.

Return Code
8

INZU307E COPYDDN -n IS NOT SUPPORTED WHEN UNLOADING COLUMN TYPE LOB OR XML

Explanation:
The COPYDDN -n option was specified for a logical unload requesting a LOB or XML column be unloaded. For the later situation, only the LAST_IC option is supported by Db2 HPU.

User response:
Specify COPYDDN LAST_IC or COPYDDN -1.

Return Code
8

INZU308E NO ELIGIBLE FLASHCOPY WAS FOUND FOR COLUMN column_name FROM TABLE qualifier.table_name

Explanation:
Unloading of an XML or a LOB column from FlashCopy image copies was requested but the mandatory set of copies for the indicated column (see the requirements for using COPYDDN LAST_IC or COPYDDN -1 when a LOB or XML column is involved) was not found in the Db2 catalog.

User response:
Create a set of FlashCopy image copies that fulfills the requirements for the COPYDDN LAST_IC or COPYDDN -1 feature when a LOB or XML column is involved.

Return Code
8
INZU309E  COPYDDN OPTION ANYTYPE IS NOT SUPPORTED WHEN UNLOADING COLUMN TYPE LOB OR XML

Explanation:
The requested ANYTYPE option of the COPYDDN feature is invalid since an XML or LOB column must be unloaded. In such a case, only GLOBAL or PARTITIONED keywords are supported by Db2 HPU.

User response:
Replace the ANYTYPE option by either GLOBAL or PARTITIONED according to the available set of images copies.

Return Code
8

INZU310E  CCSID ccсид IS NOT ccсид_class.

Explanation:
A target CCSID with an irrelevant class was specified.

User response:
Specify a CCSID from the indicated class.

Return Code
8

INZU311E  CONVERSION INTO DBCS CCSID ccсид IS NOT SUPPORTED FOR COLUMN column_name

Explanation:
This message indicates that an invalid conversion into a DBCS CCSID was requested.

User response:
Specify a SBCS or VBCS CCSID at the target CCSID/Return Code
8

INZU351I  DATA SET GENERATED.

Explanation:
This message provides information about a data set that was generated from a TEMPLATE in EXECUTE NO mode.

User response:
No action is required.

Return Code
0

INZU352I  DATA SET GENERATED.

Explanation:
This message provides information about a data set that was generated from a TEMPLATE in EXECUTE NO mode.

User response:
No action is required.

Return Code
0

INZU353I  DATA SET GENERATED FOR PARTITION n

Explanation:
This message provides information about a data set that was generated from a TEMPLATE in EXECUTE NO mode.

User response:
No action is required.

Return Code
0

INZU354E  AUTOTAG OVERFLOW. IN SELECT STARTING AT POS(line,col) MAXIMUM AUTOTAG VALUE integer REACHED

Explanation:
The generated value for the autotag has reached the maximum limit.

User response:
Increase the size of the autotag.

Return Code
8

INZU355E  AUTOTAG ERROR FOR SELECT num_select STARTING AT POS(line,col). NO AUTOTAG/AUTOTAG CONFLICT. MERGING RECORDS WITH INCONSISTENT AUTOTAG SPECIFICATIONS INTO OUTPUT DDN=ddd NOT ALLOWED

Explanation:
The AUTOTAG option in one SELECT statement is set to a value other than zero, and the AUTOTAG option in a different SELECT statement is set to zero or is not specified. These inconsistent AUTOTAG values are shared by the same output file. This output file is allocated under ddname ddd. The location of one of the conflicting SELECT statements is provided in the message.
User response:
Ensure that an output data set is the target of SELECT statements with compatible AUTOTAG specifications by either separating the output data sets of the conflicting statements or by making their AUTOTAG specifications consistent. To make the AUTOTAG specifications consistent, either add or remove the AUTOTAG specifications in the UNLOAD statement.

Return Code

8

INZU356E  AUTOTAG ERROR FOR SELECT num_select STARTING AT POS(line,col). DIFFERENT SIZES MERGING RECORDS WITH INCONSISTENT AUTOTAG SPECIFICATIONS INTO OUTPUT DDN=ddd NOT ALLOWED

Explanation:
Inconsistent sizes of AUTOTAG fields are requested for SELECT statements that have the same output file. The output file is allocated under ddname ddd. The location of one of the conflicting SELECT statements is provided in the message.

User response:
Ensure that an output data set is the target of SELECT statements with compatible AUTOTAG specifications by either separating the output data sets of the conflicting statements or by making their AUTOTAG specifications consistent. To make the AUTOTAG specifications consistent, specify the same AUTOTAG specifications in the UNLOAD statement.

Return Code

8

INZU357E  IN SELECT POS(line, col), SQLPART OPTION IS INVALID FOR TABLE SPACE dbname.tsname

Explanation:
The SQLPART option is available only for SELECT statements for which tables and table spaces can be determined.

User response:
Remove the SQLPART option, or modify the SELECT statement and run the job again.

Return Code

8

INZU359E  IN SELECT POS(line, col), INVALID OUTDDN SQLPART OPTION REQUIRED SAME PROPERTIES TO BOTH OUTPUT DDN

Explanation:
DDN1 and DDN2 of OUTDDN(DDN1, DDN2) must be built identically. If OUTDDN uses templates, DDN1 and DDN2 must both be templates with the &PART variable to unload partitions in distinct files, or DDN1 and DDN2 must be templates without the &PART variable to unload partitions in one file. If OUTDDN uses files that are allocated in JCL, each partition or range of partitions must be unloaded in the same build as ddname (for example, DDN100i, ddn DDN200i).

User response:
Modify the definition of the output file to ensure that DDN1 and DDN2 are built the same way.

Return Code

8

INZU360I  SELECT STATEMENT PROCESSING THROUGH DB2

Explanation:
This message indicates that the SELECT statement is processed through Db2.

User response:
No action is required.

Return Code

0

INZU362E  UNLOAD nnnn FROM TABLE SPACE dbname.tsname FAILED POS(line, column)“ RC = 0xnnnnn

Explanation:
This message indicates that an unload that was natively processed by Db2 HPU failed. This unload was skipped to allow the next unload to process. The indicated reason code is meant for advanced diagnosis by the IBM support.

User response:
Check for other Db2 HPU messages or system messages that might explain why the unload failed. Fix the problem, and run the job again. If you can't find the cause of the issue, contact your IBM support.

**Return Code**

8

**INZU363E** UNLOAD nnnn FROM DB2 FAILED
POS(line, column)" RC = 0xnnnnnn

**Explanation:**
An unload that was processed through Db2 failed. This unload was skipped to allow the next unload to process.

**User response:**
Check for other Db2 HPU messages or system messages that might explain why the unload failed. Fix the problem, and run the job again.

**Return Code**

8

**INZU364E** UNSUPPORTED DEFAULT VALUE FOR COLUMN colname IN LIMIT KEY OF PARTITION part number OF TABLE SPACE dbname.tsname

**Explanation:**
Db2 HPU cannot build the SQL filter because the default value for the LIMIT KEY of a partition is unsupported.

**User response:**
Specify an explicit partition limit for the indicated columns, or remove the SQLPART option.

**Return Code**

8

**INZU365E** SELECT STARTING AT POS(line, column) CANNOT BE PROCESSED

**Explanation:**
A SELECT statement cannot be processed.

**User response:**
See other Db2 HPU messages that were issued before INZU365E to determine why the SELECT statement cannot be processed. Fix the problem, and run the job again.

**Return Code**

8

**INZU366I** utility RETURN CODE rc (REASON CODE rs)

**Explanation**
This informational message contains the utility return and reason code when an error occurs. The reason code is intended to be used only by support personnel for diagnosis if the cause of the error cannot be determined.

**User response**
Refer to the SYSPRINT for information about the return code. If you cannot determine the cause of the error, contact IBM Software Support and supply the return and reason codes.

**Return Code**

8, 12, or 16

**INZU368I** PHYSICAL AND LOGICAL UNLOAD FROM GLOBAL FIC WITH PARTITION FILTERING WILL BE PROCESSED IN TWO SEPARATE TASKS

**Explanation:**
A physical unload and a logical unload were requested while unloading from a global image copy with partition filtering or using one output file per partition. The physical unload and the logical unload are run separately. The input image copy is read twice to process the UNLOAD command.

**User response:**
No action is required.

**Return Code**

None.

**INZU369W** WARNING : CLAUSE INTO OR REFORMAT IS IGNORED WITH FORMAT USER

**Explanation:**
The FORMAT USER is requested and incompatible REFORMAT or INTO clause is also specified. As a consequence, the formatting specification from the REFORMAT or INTO clause are ignored.

**User response**
- Case 1: if the output format is the expected one and you want the unload to complete with a completion code of zero, consider the following changes:
  - if an INTO clause is ignored, remove it.
  - if a REFORMAT clause is ignored, remove it if it does not apply to any other logical unload that does not request the USER format. Otherwise, rewrite the UNLOAD command so that the
REFORMAT clause does not apply to any logical unload requesting the USER format.

- Case 2: if the output format is not the expected one because the INTO or REFORMAT clause is ignored, consider the following changes:
  - if an INTO clause is ignored, merge the USER format and the INTO clause format specifications into a single INTO clause that specifies the actual desired output format.
  - if a REFORMAT clause is ignored, replace the USER format by another format clause and an INTO clause.

Return Code

4

INZU370I LDS ALLOCATION FAILED FOR [TABLE SPACE|INDEX SPACE] dbname.spacename [PART nn], DB2 CATALOG IS ACCESSED TO UPDATE LDS NAME

Explanation:
Dynamic allocation of the LDS that corresponds to the table space or index space, and partition has failed. Db2 HPU will refresh the IPREFIX information from the Db2 catalog before dynamic allocation is reissued.

User response:
No action is required.

Return Code
None.

INZU371I THE LDS NAME HAS NOT BEEN MODIFIED BY THE SWITCH PHASE OF AN UTILITY

Explanation:
Message INZU370I was previously issued, but the value of the IPREFIX that was read from the Db2 catalog is identical to the value that was initially read. Dynamic allocation is not reissued.

User response:
No action is required.

Return Code
None.

INZU372I OBID REPORT FOR UNLOAD unload-number

Explanation:
This message is part of the OBID report. It contains the number for the unload that unloaded the image copy.

User response:
No action is required.

Return Code
None.

INZU373I IMAGE COPY INFORMATION: SSID ssid, DBID n X'nnnnn' PSID n X'nnnn'

Explanation:
This message is part of the OBID report. It contains the SSID and the DBID/PSID that identifies the table space.

User response:
No action is required.

Return Code
None.

INZU374I n OBID(s) ENCOUNTERED IN THE IMAGE COPY

Explanation:
This message is part of the OBID report. It contains the number of OBIDs that were found in the image copy.

User response:
No action is required.

Return Code
None.

INZU375I OBID n X'nnnn'

Explanation:
This message is part of the OBID report. It contains an OBID that identifies a table that was found in the image copy.

User response:
No action is required.

Return Code
None.

INZU376I UNLOAD n SELECT n [PARTITION n] NUMBER OF RECORDS WRITTEN n

Explanation:
This message indicates the total number of records that were written for each logical unload (SELECT statement) or each physical unload (UNLDDDN) into the output file that is specified in message INZU222I. The unload (physical or logical) the message is related to is referred to as follows:
- the number of the UNLOAD command it belongs to
  - SELECT 0 for a physical unload
- the number of the SELECT statement for a logical unload

The numbers are related to the writing sequence and the SELECT sequence numbers relate a given UNLOAD command

User response:
No action is required.

Return Code
None.

INZU377E  INVALID PADDING SPECIFICATION.
PADDING_STRICT IS SET IN PARMLIB PARAMETER VUU030/ULOPTNS

Explanation:
An old style of syntax for PADDING OPTIONS was used, but PARMLIB parameter VUU030/ULOPTNS uses only the new style of syntax.

User response:
Change the SYSIN to match the PADDING (SBCS padding [, DBCS padding [, record padding ]]) syntax.

Return Code
8

INZU378I  DEPRECATED PADDING SYNTAX: THIS SYNTAX WILL BE NOT SUPPORTED IN NEXT RELEASE. PLEASE UPDATE HPU SYSIN TO CONFORM TO NEW BEHAVIOUR OR USE FLAG PADDING_STRICT IN PARMLIB PARAMETER VUU030/ULOPTNS

Explanation:
The PADDING 'c'/x'hh'/x'hhhh' syntax was used, and PADDING_STRICT(NO) was specified in PARMLIB parameter VUU030/ULOPTNS to support the PADDING 'c'/x'hh'/x'hhhh' syntax. However, the PADDING 'c'/x'hh'/x'hhhh' syntax will not be supported in later releases.

User response:
Consider using the PADDING (SBCS padding [, DBCS padding [, record padding ]]) syntax for PADDING to manage the padding of both CHAR and GRAPHIC data types. Change the SYSIN syntax.

Return Code
8

INZU380I  INFORMATION: DEFAULT OUTPUT FORMAT IS SET TO format DUE TO PARMLIB PARAMETER VUU045/ULFORMAT

Explanation:
This message specifies the default format that will be used by Db2 HPU when FORMAT has not been specified for a SELECT statement.

User response:
No action is required.

Return Code
None.

INZU381I  THE COPYDDN KEYWORD IS NOT VALID. DB2 NO MUST BE SPECIFIED. THIS SYNTAX IS DEPRECATED AND WILL NOT BE SUPPORTED IN LATER RELEASES.

Explanation:
The UNLOAD statement specifies COPYDDN and none of the following conditions is fulfilled:
- the DB2 NO option is specified
- no DB2 option is specified and ULSEDB2 is set to NO.

As the unload from an image copy can only be performed in native mode (DB2 NO), this is an ambiguous request. In versions up to V3R2, Db2 HPU ignored the COPYDDN option in such a case. From version V4R1, a more straightforward behavior considering such a request as invalid can be enabled by specifying COPYDDN_STRICT(YES) in the VUU030/ULOPTNS parmlib parameter.
As the Db2 HPU settings specified does not specify COPYDDN STRICT(NO), for compatibility reasons, the former ambiguous behavior - i.e. the data might not be unloaded from the image copy but from the "online" table - applies to the current unload.

**User response:**
As the former behavior corresponding to COPYDDN STRICT(NO) is deprecated and will be removed in a future Db2 HPU release, consider changing the unload statement and/or the Db2 HPU settings to meet the enhanced behavior requirements as follows. If unloading from an image copy is desired, specify the DB2 NO option or make sure the ULSEDB2 parmlib parameter is set to NO. The latter change should be done only after checking that no other job with COPYDDN option and without a DB2 option exists. If the unload from the online table is desired, remove the unintended or superfluous COPYDDN option. More generally speaking, consider setting COPYDDN STRICT(YES) to make sure such an ambiguous request (unload from image copy and SQL mode) can no longer be specified. Be aware that this setting change must be done after reviewing all the unload jobs specifying an UNLOAD statement with the COPYDDN option.

**Return Code**

0

**INZU382E** THE COPYDDN OPTION AND THE DB2 YES OR DB2 FORCE OPTIONS ARE INCOMPATIBLE.

**Explanation:**
The COPYDDN keyword cannot be specified with DB2 YES or DB2 FORCE when COPYDDN STRICT(YES) is specified for the VUU030/ULOPTNS PARMLIB parameter. To unload from an image copy, both DB2 NO and the COPYDDN keyword must be specified.

**User response:**
To unload from an image copy, specify COPYDDN and DB2 NO in the SYSIN.

**Return Code**

8

**INZU384I** THE DB2 PARAMETER WAS FORCED TO NO BECAUSE COPYDDN WAS SPECIFIED.

**Explanation:**
The DB2 parameter was forced to NO for one of the following reasons:
- A value for the DB2 parameter was not specified in the SYSIN.
- A request was made for data to be unloaded from an image copy by using the COPYDDN parameter.
- COPYDDN STRICT(YES) was specified for the VUU030/ULOPTNS PARMLIB variable.

**User response:**
No action is required.

**Return Code**

0

**INZU386I** THE UNLOAD PROCESS IS SERIALIZED BECAUSE OF TAPE MANAGEMENT.

**Explanation:**
An unload process is serialized when a logical unload or a physical unload is processed by using output files that are allocated on tape devices. The tape devices are defined by the TAPEUNIT option.

**User response:**
No action is required.

**Return Code**

0

**INZU386E** THE FOLLOWING UNIT IS DUPLICATED IN THE TAPEUNIT OPTION: unit_name.

**Explanation:**
The definition of a tape unit or a storage class is duplicated in the TAPEUNIT option.

**User response:**
Remove one of the duplicate definitions.

**Return Code**

8

**INZU387E** THE ORDER BY CLAUSE IS NOT SUPPORTED WHEN DATA IS UNLOADED IN FORMAT INTERNAL. THE TABLE SPACE IS IN BASIC ROW FORMAT, AND THE column_name COLUMN IN THE table_name TABLE IS POSITIONED AFTER A VARYING-LENGTH COLUMN.

**Explanation:**
The ORDER BY clause cannot be processed in the INTERNAL format because the records of the table space are in basic row format (BRF), and the specified column that is used in the ORDER BY clause is after a variable length column.

**User response:**
No action is required.
INZU388W  THE INDEX KEY IS NOT SUPPORTED WHEN DATA IS UNLOADED IN FORMAT INTERNAL. THE TABLE SPACE IS IN BASIC ROW FORMAT AND THE INDEX KEY column_name COLUMN IN THE table_name TABLE IS POSITIONED AFTER A VARYING-LENGTH COLUMN. CLUSTER INDEX KEY NOT SUPPORTED

Explanation:
The ORDER CLUSTER clause cannot be processed in the INTERNAL format because the records of the table space are in basic row format (BRF), and the specified column that is used in the key of the clustering index is after a variable length column. The ORDER CLUSTER clause is ignored for the SELECT statement. This message is preceded by message INZU044W.

User response:
No action is required.

INZU389E  THE FOLLOWING CLAUSE IS INCOMPATIBLE WITH FORMAT INTERNAL: clause_name.

Explanation:
The clause cannot be used with the INTERNAL format.

User response:
Remove the clause, or unload the data in another type of format.

INZU390E  FORMAT INTERNAL AND ANOTHER FORMAT CANNOT BE SPECIFIED WITHIN THE SAME UNLOAD COMMAND.

Explanation:
The INTERNAL format cannot be used with other format types in the same UNLOAD command.

User response:
Use FORMAT INTERNAL for all logical unloads that are processed natively in the same UNLOAD command, or split the unload by using at least two UNLOAD commands.
The MAXPART parameter was forced to 1 to unload the specified table space.

User response:
No action is required.

Return Code
0

INZU394I DATA FROM THE IMAGE COPY WILL BE UNLOADED WITHOUT PARTITION INFORMATION BECAUSE OF THE FOLLOWING REASON: reason.

Explanation
An output file per partition has been requested for a table space that was unloaded from an image copy. The structure of the image copy file or the structure of the unloaded table space in the Db2 catalog or in the DDLDDN file does not allow unloading per partition. All the unloaded rows will be written in the file that corresponds to the first selected partition. The message text indicates one of the following reasons:

- THE IMAGE COPY CONTAINS DATA FROM A NONPARTITIONED TABLE SPACE. THE SPECIFIED TABLE SPACE IS A PARTITIONED TABLE SPACE.
- THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLE SPACE. THE SPECIFIED TABLESPACE IS A PARTITION-BY-GROWTH TABLE SPACE.

User response:
No action is required.

Return Code
None.

INZU395E THE IMAGE COPY CANNOT BE USED FOR THIS TABLE SPACE FOR THE FOLLOWING REASON: reason.

Explanation
The input image copy file has a structure that is not compatible with the unloaded table space as described in the Db2 catalog or in the DDLDDN file. Db2 HPU cannot unload from such image copy. The message text indicates one of the following reasons:

THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLESPACE
THE SPECIFIED TABLESPACE IS A NON-PARTITIONED TABLESPACE

- THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED-BY-GROWTH TABLESPACE
THE SPECIFIED TABLESPACE IS A NON-PARTITIONED-BY-GROWTH TABLESPACE

- THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLESPACE OF number PARTITIONS
THE SPECIFIED TABLESPACE IS A PARTITIONED TABLESPACE AND HAS LESS PARTITIONS:
number

- THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLESPACE OF number PARTITIONS
THE SPECIFIED TABLESPACE IS A PARTITIONED TABLESPACE AND HAS MORE PARTITIONS:
number AND UNLOAD BY PARTITION HAS BEEN REQUESTED

- THE IMAGE COPY CONTAINS SEVERAL VERSIONS FOR THE TABLE table__name WHILE THE TARGET TABLE HAS NO VERSION

User response:
Correct the structure mismatch either by providing another input image copy or by unload another table space. If the structures are compatible (both are partitioned but the table space specified in SYSIN has a bigger number of partition), consider changing the list of partitions to be unloaded or request a single output data set to be created.

Return Code
8

INZU398E DB2 YES CANNOT BE SPECIFIED FOR A TABLE WITH CONTROLLED ACCESS ON A ROW OR COLUMN. THE TABLE NAME IS table_name, AND THE COLUMN NAME IS column_name.

Explanation:
Access to the specified table is controlled at the row or column level. For such a table, output depends on whether data is accessed natively or by Db2. DB2 YES is restricted because it does not explicitly set the type of access mode to be used.

**User response:**
Specify DB2 NO or DB2 FORCE. For a table whose access is controlled at the column level, consider removing the protected column from the SELECT statement.

**Return Code**
8

**INZU399E**
A TABLE WITH CONTROLLED ACCESS ON A ROW OR COLUMN CANNOT BE UNLOADED IN NATIVE MODE. THE TABLE NAME IS table_name, AND THE COLUMN NAME IS column_name.

**Explanation:**
Access to the specified table is controlled at the row or column level. According to the specified value of VUU062/ULACCTRL, unloading data in native mode (when DB2 NO is specified) is restricted for this table.

**User response:**
Specify DB2 YES or change the setting of the VUUXX/ULACCTRL parameter to allow data to be unloaded in native mode. For a table whose access is controlled at the column level, consider removing the protected column from the SELECT statement.

**Return Code**
8

**INZU400I**
The following output CCSID for an XML column might not match encoding XML standards: ccsid.

**Explanation:**
The chosen CCSID for an XML column might generate XML with an encoding name that is does not match XML standards.

**User response:**
Check the output CCSID for the XML data.

**Return Code**
0

**INZU401E**
UNLOADING A VIEW WITH FORMAT INTERNAL IS NOT SUPPORTED.

**Explanation:**
FORMAT INTERNAL cannot be used to unload a view.

**User response:**
Specify a different type of format.

**Return Code**
8 or 0 if a later INZU538I message has been issued for the same unload task

**Note:** When an ALTERNATE FORMAT is specified for the INTERNAL format and the message INZU538I is present, this error message is actually ignored but its text explains the reason why the INTERNAL format has been generated for the SELECT statement in question.

**INZU402E**
SELECT WITH COLUMN LIST IS NOT SUPPORTED WHEN DATA IS UNLOADED IN FORMAT INTERNAL.

**Explanation:**
FORMAT INTERNAL cannot be used with a partial selection of the columns of the table.

**User response:**
Specify a SELECT * statement.

**Return Code**
8

**INZU403E**
FLASHCOPY AND NON FLASHCOPY CANNOT BE MIXED.
FLASHCOPY: dsname
NON FLASHCOPY: dsname.

**Explanation:**
A FlashCopy and a non-FlashCopy have been specified as input for a single UNLOAD statement for a partitioned table space. Db2 HPU cannot process both image copies.

**User response:**
Issue one UNLOAD statement with the FlashCopy as input and a separate UNLOAD statement with the non-FlashCopy as input.

**Return Code:**
8

**INZU404E**
LOB AND XML FILES CANNOT BE UNLOADED ON TAPE.

**Explanation:**
Db2 HPU cannot unload LOB and XML files on tape.

**User response:**
Specify a DASD output file for LOB and XML files.

**Return Code**
8

**INZU405E**
LOADDDN CANNOT BE GENERATED ON TAPE TEMPLATE.
**Explanation:**
The LOADDN cannot use a template that explicitly defines the dataset on a TAPE unit. A TEMPLATE statement explicitly defines a data set on tape if it specifies the STACK option or a unit name mentioned by the Db2 HPU TAPEUNIT feature (either through the VZM010/TAPEUNIT or the TAPEUNIT option from the GLOBAL OPTIONS block).

**User response:**
Change the template parameters so that it does not explicitly define a tape dataset by removing any STACK option or by specifying a unit name not mentioned by the TAPEUNIT feature.

**Return Code**
8

**INZU406E** THE REQUESTED QUIESCE OF THE TABLESPACE COULD NOT BE PERFORMED. CHECK DB2 STATUS OF THE TABLESPACE AND RELATED INDEXES.

**Explanation:**
The requested QUIESCE utility could not be performed because of the Db2 status of the table space or its related indexes.

**User response:**
Consider removing the QUIESCE request or perform the relevant Db2 operation to remove the restricted status that is preventing the QUIESCE utility from completing. Also, consider setting the VUU028/ULQSCEBH PARMLIB variable to FORCE so that a STOP/START sequence is attempted when the QUIESCE utility fails because of the Db2 status. Doing so enlarges the number of cases where the unload QUIESCE request can be performed.

**Return Code**
8

**INZU407W** UNLOAD n SELECT n [PARTITION n] MAX_EXPECTED_ROWS (n) IS LESS THAN THE UNLOADED ROWS (n).

**Explanation:**
The number of effective unloaded rows is higher than the number of expected rows that is specified in the MAX_EXPECTED_ROWS keyword. Further processing might be affected because the sort resources or the size of the output data set might be insufficient. The unload (physical or logical) the message is related to is referred to as follows:

- the number of the UNLOAD command it belongs to
- the number of the SELECT statement for a logical unload

The numbers are related to the writing sequence and the SELECT sequence numbers relate a given UNLOAD command.

**User response:**
Modify the value of the MAX_EXPECTED_ROWS keyword so that it reaches at least the number of unloaded rows.

**Return Code**
4

**INZU408I** INDEX SCAN ACCESS HAS BEEN DISABLED DUE TO MAX_EXPECTED_ROWS SPECIFICATION.

**Explanation:**
The index scan access to sort rows by using the cluster index has been disabled because the MAX_EXPECTED_ROWS keyword has been specified in the SELECT statement.

**User response:**
If necessary, remove the MAX_EXPECTED_ROWS keyword to enable the index scan access. The index scan access is more efficient than the sort utility, but it depends on the number of output rows.

**Return Code**
0

**INZU409E** INTERNAL ERROR WHEN PREPARING DB2SORT INVOCATION - REASON 0xnnnn

**Explanation:**
Db2 HPU unexpectedly failed when preparing the Db2 Sort utility invocation.

**User response:**
Contact IBM Software Support.

**Return Code**
8

**INZU410E** INSUFFICIENT RESOURCES FOR DB2SORT INVOCATION. THE NUMBER OF SORTS TO PROCESS IS n MAXSORT IS SET TO p DB2SORT CAN PROCESS A MAXIMUM OF q SORTS IN PARALLEL

**Explanation:**
Not enough resources are available to process the requested sorts with Db2 Sort. The number of sorts to be processed in parallel (i.e. \( \min(p, n) \)) is greater than the number of sorts that Db2 Sort can handle in parallel (\( q \)).

**User response:**
Reduce the sort parallelism degree by decreasing the MAXSORT parameter value \( p \) to a value that is lower than \( q \). If a partitioned table space is used, consider reducing the unload tasks parallelism by decreasing the first operand in the PARALLELISM keyword or by using the MAXPART keyword.

**Return Code**

8

**INZU415I**  
DB2 PARAMETER SET TO FORCE BECAUSE SQLID WAS SPECIFIED.

**Explanation:**
The DB2 parameter was forced to FORCE for the following reason: a request was made to change the CURRENT SQLID.

**User response:**
No action is required

**Return Code**

0

**INZU416E**  
THE SQLID KEYWORD AND THE DB2 YES OR DB2 NO OPTIONS ARE INCOMPATIBLE

**Explanation:**
The SQLID keyword cannot be specified with DB2 YES or DB2 NO. To specify an SQLID, both DB2 FORCE and the SQLID keyword must be specified.

**User response:**
Remove the undesired SQLID option or specify DB2 FORCE in the SYSIN.

**Return Code**

8

**INZU417I**  
IMAGE COPIES PRIOR TO BRA rba CANNOT BE USED FOR UNLOADING database.tablespace

**Explanation**
This message informs that a line describing a DROP COLUMN materialization is found for the tablespace. Image COPY created with a smaller rba value cannot be used for UNLOAD. Database, tablespace and RBA are those that are printed in the Db2 message DSNU556I when trying to do a recover with a too old image copy.

**User response:**
If the message INZU042E is also issued, look at the explanation for this message.

**Return Code**

None

**INZU418E**  
THE POSITION OF A LOB/XML COLUMN IN THE SELECT LIST-OF-COLUMNS/* IS INCOMPATIBLE WITH SPANNED YES OPTION, PROCESS CANNOT CONTINUE BECAUSE ENFORCE_COLUMN_ORDER YES WAS SPECIFIED.

**Explanation**
The SPANNED YES option was requested. This option requires the LOB or XML data to be placed at the end of the result table (defined by the column list of the SELECT clause and by the optional INTO clause). As some LOB or XML columns positions do not meet this requirement, the unload can only be processed if ENFORCE_COLUMN_ORDER NO which allows Db2 HPU to rearrange the columns of the result table. As ENFORCE_COLUMN_ORDER YES applies to the unload in progress, the processing stops.

If INZU417I message has been previously issued, see the related explanation for more details on the origin of the issue.

**User response:**
Use the value NO of the parameter ENFORCE_COLUMN_ORDER, or change the order of selected columns so that the LOB/XML columns are the last columns of the SELECT statement.

**Return Code**

8

**INZU419I**  
THE POSITION OF A LOB/XML COLUMN IN THE SELECT columnlist IS INCOMPATIBLE WITH SPANNED YES OPTION, COLUMNS WERE REORDERED BECAUSE ENFORCE_COLUMN_ORDER NO WAS SPECIFIED.

**Explanation**
The SPANNED YES option was requested. This option requires the LOB or XML data to be placed at the end of the result table (defined by the column list of the
SELECT clause and by the optional INTO clause). As some LOB or XML columns positions do not meet this requirement and ENFORCE_COLUMN8ORDER NO applies to the unload in progress, Db2 HPU rearranges the columns of the result table to allow the unload to be processed.

User response: None

Return Code 0

INZU420I SPANNED YES OPTION APPLIES TO SELECT STARTING AT (x,x) ENDING AT (x,x). OUTPUT FILE HAS RECFM VBS.

Explanation
As the SPANNED YES option is specified for a logical unload (SELECT clause) which result table has LOB data columns, the mandatory VBS RECFM is needed for the OUTDDN output data set. Further processing will make sure this requirement is fulfilled.

User response: None

Return Code 0

INZU421W SPANNED YES AT POS (x,y) WILL BE IGNORED BECAUSE THERE IS NO LOB DATA TO BE UNLOADED IN SPANNED FORMAT

Explanation
The SPANNED YES option is specified for a logical unload which result table has no LOB data. SPANNED YES does not apply in such a case and is therefore ignored.

User response: Consider removing the SPANNED YES option of you actually do not mean to unload any LOB data.

Return Code 4

INZU422E THE FORMAT REQUESTED FOR THE LAST SELECTED COLUMN, WHICH IS NOT A LOB OR AN XML, IS NOT COMPATIBLE WITH THE SPANNED YES OPTION.

Explanation
The requested format prevents from unloading data with the SPANNED YES option (i.e. into a SPANNED file).

User response: Select one of the formats supported when SPANNED YES option is requested or remove the SPANNED YES option of it is not intended.

Return Code: 8

INZU423I SPANNED YES AT POS (x,y) WILL BE IGNORED BECAUSE OF clause.

Explanation
This message informs that the requested clause (ex. FORMAT DELIMITED) is not compatible with the SPANNED YES option. As SPANNED option was specified at the GLOBAL level, it is ignored. The place of the clause indicated by this message is given by an immediately preceding INZU010I message.

User response: None

Return Code: 0

INZU424E SPANNED YES AT POS (x,y) IS INCOMPATIBLE WITH clause

Explanation
The requested clause (ex. FORMAT DELIMITED) is not compatible with the SPANNED YES option. As the SPANNED option was not specified at the GLOBAL level, the ambiguous request cannot be performed. The place of the clause indicated by this message is given by an immediately preceding INZU010I message.

User response: Either remove the SPANNED YES option or do not specify the mentioned clause.

Return Code: 8

INZU425E AT POS (x,x) explanation WHEN SPANNED YES OPTION IS SPECIFIED

Explanation
A situation prevents a logical unload with the SPANNED YES option to be performed.

The failure cause can be one of the following:

1. HFS FILE NOT ALLOWED: The OUTDDN file allocation (either via JCL DD card of TEMPLATE) HFS file.
2. CCSID CONVERSION NOT ALLOWED FOR LOB/XML COLUMN: the unload command requires a CCSID conversion for a LOB or XML column
3. SPECIFY LOBFILE FOR XML COLUMN
4. OUTFREQROWS NOT ALLOWED FOR SELECT PROCESSED BY DB2
5. SORT EXTERNAL NOT ALLOWED
6. USE OF BOTH LENGHT REAL AND LENGTHBYTE NO NOT ALLOWED
7. MULTI-BYTES LENGTH INDICATOR NOT ALLOWED

User response
Consider removing the cause of the failure or the SPANNED YES option.
1. do not allocate the OUTDDN file as an HFS file
2. remove the CCSID conversion request that affects the LOB or XML columns
3. specify a REFORMAT or an INTO clause for XML columns to unload XML data as LOBFILEs
4. remove the OUTFREQROWS option
5. remove the SORT EXTERNAL option
6. remove either the LENGTH REAL or the LENGTHBYTE NO option
7. specify a one-byte NULL indicator field

Return Code: 8

INZU426E SPANNED DATASET dsname CANNOT BE USED TWICE FOR SELECT STARTING AT (x,x) ENDING AT (x,x) AND SELECT STARTING AT (x,x) ENDING AT (x,x)

User response:
A dataset used as the output of a logical unload (i.e. a SELECT statement) specifying the SPANNED option is also specified at the output dataset for another logical unload. This is forbidden as a spanned dataset cannot be shared by two unloads. A separate dataset must be specified for each dataset to be written in spanned format.

User response:
Specify distinct output datasets

Return Code: 8

INZU427I LOAD OPTIONS options IS TRANSFORMED INTO A COMMENT BECAUSE ITS DISALLOWED BY DB2 WHEN OPTION SPANNED YES IS USED.

Explanation:
A LOADOPT clause was specified for an unload request which output is to be written in SPANNED format. The mentioned option specified by the LOADOPT is not supported by the Db2 LOAD utility when the data to be loaded is a file with RECFM=VBS (spanned file). This incompatible option was transformed into a comment in the generated LOAD command.

User response:
None

Return Code: None

INZU428E THE UNLOAD COMMAND, INCLUDING SELECT STARTING position, REQUESTS TO UNLOAD THE LOB COLUMN column name INTO INCOMPATIBLE OUTPUT FORMATS.

Explanation:
At least two logical unloads from a single UNLAD command request the same LOB column be unloaded in spanned instream format on one hand and in a different format on the other hand (i.e. non-spanned in-stream or LOB file).

User response:
Separate the two conflicting logical unloads into two distinct UNLOAD commands.

Return Code: 8

INZU429E XMLSET ERROR: INVALID VARIABLE/STRING - string - IN XMLTAG DEFINITION

Explanation:
An invalid variable or string is used to define an XML tag.

Operator response:
None

System programmer response:
Check XML tag definition.

Return Code: 8

INZU430E XMLSET ERROR: DUPLICATED DEFINITION OF VARIABLE - string -

Explanation:
An XMLSET variable is duplicated

Operator response:
None

System programmer response:
Check XMLSET option.

Return Code:
User response
Consider one or more of the following options to reduce the length of the non-LOB data:

- Reduce the length of larger columns by specifying a smaller length (if the USER format is used or by adding an INTO clause).
- Use the SUBST() SQL scalar function
- unload only the necessary columns

Return Code: 8

INZU437E ALL TABLE ARE NOT IN SAME TABLESPACE

Explanation
An IBM UNLOAD statement involving tables belonging to distinct table spaces was verified, which is not allowed. When an IBM UNLOAD statement is specified, all involved tables must belong to the same table space.

Action
Consider splitting up the unload statement into separate statements involving only tables from the same table space.

Return Code: 8

INZU439W MULTI ROW FETCH (rows number) HAS BEEN FORCED DUE TO QUERY ACCELERATION REQUEST.

User response:
The QUERY_ACCELERATION parameter is specified with a value different from NONE and the UNLEOWSET or its default VUU035/UNLROWSET is set to 1. In this case, the multi-row fetch mode is forced with the number of rows (rows number) specified in the message.

User response:
Consider specifying ULROWSET to the value mentioned in the message if you do not want the warning message and its related return code 4 be issued.

Return Code: 4

Chapter 8. Troubleshooting 447
User response:
Consider removing the QUERY_ACCELERATION option if you do not want the warning message and its related return code 4 be issued.

Return Code:
4

**INZU441I** PART OPTION IS IGNORED AS A DATA-UNLOAD PROCESS USE SQL MODE

Explanation:
The PART keyword is specified but as the request is processed by SQL, the keyword is ignored.

User response:
Remove the PART keyword.

Return Code:
None.

**INZU442E** OPTION option IS NOT SUPPORTED BY THE UTILITY

Explanation:
An UNLOAD statement with either the IBM UNLOAD or FASTUNLOAD or UNLOAD PLUS syntax was specified and an option incompatible with this syntax was specified. Db2 HPU stops as it cannot process the ambiguous statement.

Action:
Remove the unsupported option from the unload statement.

Return Code
8

**INZU443I** SELECT sequential_number STARTING AT POS(line, column) FOR TABLE tbname

Explanation:
This message is issued before other messages to identify to which part of the statement the subsequent messages refer to. The sequential_number is sequence number of the SELECT statement in the UNLOAD statement.

Action:
None.

Return Code
0

**INZU445E** FORMAT DELIMITED REQUIRES ONE AND ONLY ONE FROM TABLE

System action:
An IBM UNLOAD statement requesting output data in DELIMITED format has been specified. The statement cannot be processed because the unload statement involves several tables, which is incompatible with the IBM UNLOAD DELIMITED format. With the IBM UNLOAD DELIMITED format, only one table can be unloaded at a time.

Action:
Consider splitting up the unload statement into separate statements involving only one table.

Return Code
8

**INZU447E** AN ERROR OCCURRED DURING UNLOAD PROCESSING - REFER TO PREVIOUS EXPLANATION MESSAGES

Explanation:
An error occurred while unloading the data. The error usually affects only some of the records to be processed. More details about the error is given by previous messages.

User response:
Check if the situations reported by the detailed error messages can be solved by changing the unload request.

Return Code
8

**INZU448E** SECOND OR THIRD ARGUMENT OF SUBSTR OUT OF RANGE

Explanation:
An error was encountered during the SELECT/WHERE clause evaluation. The line concerned was discarded.

User response:
Check the SELECT/WHERE clause, SUBSTR function.

Return Code
8

**INZU449E** INVALID FILED NAME name FOR CONSTANT

System action:
An IBM UNLOAD statement that selects a constant named with the same name as the name of a column
from the unloaded table was specified. This is not supported.

**Action:**
Consider naming the constant with a different name.

**Return Code**
8

**INZU449W** OPTION option is not supported by the utility

**Explanation**
An unload statement with either the IBM ULOAD or FASTUNLOAD or UNLOAD PLUS syntax was specified and an option incompatible with the chosen syntax was specified. The unexpected option is ignored.

**Action**
Consider removing the unexpected option to get a return code 0.

**Return Code**
4

**INZU450I** The image copy used to unload the table table_name contains data from several tables and ORIGINOBID 0 was specified. This ambiguous request can lead to unpredictable results. Check the OBID report generated in SYSPRINT and specify the OBID of the table to be unloaded.

**Explanation**
A request to unload the data from an image copy has been done. Db2 HPU has detected that the image copy contains the data of several tables. As ORIGINOBID 0 has been specified, Db2 HPU unload the data from the first table encountered while reading the image copy. As this first table might not be the relevant one, this might lead to unpredictable results. Specifying an explicit value for ORIGINOBID is strongly recommended. In this case, Db2 HPU issues an OBID report which list the OBID of the tables contained in the image copy.

**User response**
Specify the obid of the table to be unloaded.

**Return Code**
0

**INZU451E** OPTION FORMAT INTERNAL is not valid when used with more than one table in the same dataset.

**Explanation**
Unloading data from several SELECT statements in INTERNAL format was requested. As the LOAD utility can only load the data from a single table, merging output from several SELECT statements is not allowed.

**Return Code**
8

**Action**
Specify separated output data sets.

**INZU452E** LOGICAL RECORD LENGTH OF OUTPUT RECORD EXCEEDED THE LIMIT FOR TABLE table_name

**System action:**
An IBM UNLOAD statement has been specified and the output record exceeds the maximal output record length.

**Action**
The unload request is not supported as is. To circumvent the size limit issue, consider the following changes according to the context:

- with an inline LOB unload, use the SPANNED option;
- if the table has varchar columns unloaded, with their maximum size, try to unload the varchar data on their actual size. This might solve the issue especially if the varcolumns are significantly oversized.

**Return Code**
8

**INZU454I** MAXERR ignored for control_type loss control as reject mode has been requested.

**Explanation:**
As data loss control for the indicated control type (CHARACTER or NUMERIC) is processed in REJECT mode (see VUU071/LOSSCHAR or VUU072/LOSSNUM in parmlib or LOSSCHAR or LOSSNUM in SYSIN), the MAXERR option specified in SYSIN is ignored. This parameter is only relevant in WARNING mode.
INZU463E  ERROR WHEN GENERATING THE
CREATE TABLE STATEMENT OF
THE TABLE : creator.table RETURN
CODE : rc REASON CODE : reason
SQLCODE : sqlcode

Explanation:
An error occurred when generating the CREATE TABLE statement requested by the LOADDL TABLE option.

User response:
If the SQLCODE is -805, check that the Db2 HPU BIND job, i.e. corresponding to the INZBIND template, has been run successfully for the Db2 subsystem or data sharing group involved. For any other situation, contact IBM Software Support.

Return Code:
8

INZU464W  NO CREATE TABLE STATEMENT
HAS BEEN GENERATED FOR
SELECT number AS THE NAME OF
THE TABLE HAS NOT BEEN
IDENTIFIED

Explanation:
The CREATE TABLE statement requested by the LOADDL TABLE option was not written into the LOADDN data set because the name of the table has not been identified. The LOADDL TABLE does not support this situation. The unload process continues and the LOADDN data set contains the LOAD statement only. This is the case when the unload is processed in SQL mode and the table description has not been obtained from the Db2 catalog.

User response:
If the SQL statement is supported in native mode, specify DB2 NO to get the LOADDL TABLE option be processed successfully. Otherwise, consider removing the LOADDL TABLE option to prevent this message from being issued.

Return Code:
4

INZU465W  NO CREATE TABLE STATEMENT
HAS BEEN GENERATED FOR
SELECT number AS THE TYPE OF
THE TABLE creator.table name IS
NOT SUPPORTED.

Explanation:
The CREATE TABLE statement requested by the LOADDL TABLE option was not written into the LOADDN data set because the type of the table is not supported. The unload process continues and the LOADDN data set contains the LOAD statement only.

User response:
Check if the Db2 object involved in the unload is an actual table, i.e. created by a CREATE TABLE statement. You can check this in the Db2 catalog after the table has been created. For such a table, the value of the TYPE column from the SYSIBM.SYSTABLES catalog table is 'T'. If this is not the case, rewrite the SELECT statement to use an actual table instead of a view or an alias or any other object. If rewriting the SELECT statement is not possible, remove the LOADDL TABLE option to prevent this message from being issued.

Return Code:
4

INZU474I  DATA TYPE FOR TARGET COLUMN
column-name NOT SPECIFIED BY
THE COLUMN-LEVEL FORMATTING
REQUEST, DEFAULT TYPE type-
name APPLIES.

Explanation:
A logical unload requests column-level formatting (e.g. an INTO clause or the USER format option) but does not specify any output data type for the mentioned column of the resulting table. Therefore, the indicated default data type is assumed as an output data type for this column, according to the rules applicable to the mentioned reformating feature (INTO clause or USER format). This default data type is determined by the SQL rules that apply to the SQL expression that describes the column of the resulting table.

User response:
none

Return code
0

INZU475E  FLASH COPY DATASET MUST BE
ALLOCATED FOR EACH UNLOADED
PARTITION

Explanation
COPYDDN ddname FLASHCOPY was specified for an unload involving a partitioned table space and the required Flash image copy data set of at least one partition is not allocated.

User response
Allocate a FlashCopy data set for each partition to unload if FlashCopy data set are to be used and
remove the type FLASHCOPY option from the COPYDDN option otherwise.

Return Code
8

INZU476I PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS LIMITED TO n BY ZIIPMXPL PARAMETER.

Explanation: The processing of partitions/LDS is done with parallelism and the related tasks are offloaded to zIIP processors. The maximum degree of parallelism for LDS reading tasks is limited by the parmlib parameter VUX042/ZIIPMXPL or by its counterpart ZIIPMXPL option from the Technical Parameters options block.

User response: No action is required.

Return code
0

INZU480I MONO-ROW FETCH READ ACTIVATED.

Explanation: A logical unload has been processed in SQL mode. The involved SQL SELECT statement has been submitted to Db2 with the single-row fetch feature as requested by either the UNLROWSET 1 option or the value 1 set for the VUU035/ULROWSET parameter.

User response: No action is required. Take this information into account for analysis and fine tuning of the Db2 performance.

Return code
0

INZU481I PART RANGE OPTIMIZATION ENABLE FOR dbname.tsname

Explanation: This message informs that OPTIMIZATION_PART_RANGE is set to YES and an attempt to reduce the set of partitions from the mentioned table space is made as at least one SELECT statement from a table belonging to this table space has a WHERE clause that might exclude some partitions. The optimization process is based on the output of an EXPLAIN PLAN statement.

User response: None.

Return Code
0

INZU484I THE VSAM DATA SET DOES NOT EXIST FOR TABLE SPACE dbname.tsname

Explanation: Table space dbname.tsname was created with DEFINE NO, and Db2 has not defined the underlying LDS files. No rows are unloaded for this object.

User response: No action is required.

Return Code
None.

INZU485I NUMBER OF PARTITIONS SELECTED PASSED FROM number TO number PARTITIONS: n1, n2, [n4 - n5]...
**Explanation:**
When the option OPTIMIZE_PART_RANGE is set to YES, this message indicates that the SELECT statement specified by preceding INZU280I has been successfully optimized, i.e. the number of partitions that will be read has been decreased according to the specified WHERE clause. It indicates how many and which partitions remain selected for processing. The list of partitions consists in partition numbers and/or partition ranges ([Part_Start - Part_End]) separated by commas.

**User response:**
None.

**Return Code**
0

**INZU486I** NO OPTIMIZATION FOUND. number_of_partitions PARTITIONS KEPT

**Explanation:**
When option OPTIMIZE_PART_RANGE is set to YES, this message indicates that the analysis of the SELECT statement specified by INZU280I led to no optimization. The number of partitions to unload remains the same. Optimization may be disabled to avoid useless query.

**User response:**
As the impossibility to optimize the query might change even if the related table data is updated, consider disabling the OPTIMIZE_PART_RANGE for this unload statement. As disabling this feature affects all the unload sysin, make sure this does not disable the actual optimization of another unload statement.

**Return Code**
0

**INZU487W** PART RANGE OPTIMIZATION FAILED FOR dbname.tsname OR creator.tablename

**Explanation:**
This message informs that OPTIMIZATION_PART_RANGE cannot be performed for the mentioned resource. This message specifies the tablespace or the table for which the EXPLAIN PLAN statement failed thus preventing the optimization from completing successfully. The reason may be an invalid userid (specified by parameter VUM035/EXPLNUSR) or the lack of the required EXPLAIN tables. The unload process continues with the OPTIMIZE_PART_RANGE feature disabled.

**User response**
Check if the VUM035/EXPLNUSR parmlib parameter is correctly set and if the specified userid has the necessary privilege to run the EXPLAIN PLAN statement. Check if the PLAN_TABLE and DSN_PGRANGE_TABLE exist.

**Return Code**
4

**INZU488W** NO PARTITION TO UNLOAD, SELECT select_number IS CANCELED

**Explanation**
When option OPTIMIZE_PART_RANGE is set to YES, this message indicates that the SELECT statement specified by INZU280I has been successfully optimized but the result of the optimization is that the WHERE clause predicates imply that no data would ever be returned by this statement regardless of the data of the involved tables. For example, this can occur if a non-sense trivial predicate such as 1 = 2 is specified or if the combination of several non trivial predicates, such as (x > 10 AND x < 5) systematically returns no row.

**User response**
Correct the WHERE clause of the involved SELECT statement or remove it if it was non intentionally specified.

**Return Code**
4

**INZU492W** WARNING: DATA FROM COLUMN column WAS TRUNCATED TO n BYTES FROM m

**Explanation**
Data from a column has been truncated. Truncation occurs for instream BLOB, CLOB, and DBLOB data types to limit the record size to the maximum size allowable.

**User response**
To avoid truncation, use LOB file reference (datatype BLOBF, CLOBF, DBCLOBF) in a REFORMAT option, INTO clause, or USER FORMAT definition.

**Return Code**
4
INZU493W  WARNING: CLONE KEYWORD IGNORED : text

Explanation
The CLONE keyword has been ignored. The CLONE keyword has no effect when the table space has no clone or when no physical unload (UNLDDN) is requested. The CLONE keyword is supported only for physical unloads. The text field might contain one of the following messages:

• THE TABLESPACE DOES NOT CONTAIN ANY CLONE TABLE
• ONLY SUPPORTED FOR PHYSICAL UNLOAD

To unload a CLONE table using a SELECT statement, specify the name of the CLONE table in the SELECT.

User response:
No action is required.

Return Code
4

INZU494E  ERROR: LOBFILE TYPE FOR COLUMN colname NOT ALLOWED

Explanation
LOB file data types (CLOBF, DBCLOBF, BLOBF) are supported only for CLOB, DBCLOB, and BLOB columns. You cannot convert any other data type to a LOB file, and you cannot mix a LOB type with a non-corresponding LOB file type (for example you cannot convert a BLOB column into a CLOBF data type).

User response
Check the conversion that is specified in the INTO clause, the REFORMAT option of USER format definition. Correct the invalid conversion, and resubmit the job.

Return Code
8

INZU495E  ERROR: INCORRECT LOBFILE TEMPLATE template name FOR COLUMN col: Mandatory VARIABLE missing. The TEMPLATE MUST CONTAIN AT LEAST ONE OF THE FOLLOWING ITEMS reason.

Explanation
A TEMPLATE statement is used to unload LOB data in a LOB file reference, but the DSN template does not contain either the &UNIQ or a user variable set to the :RECNUM predefined variable.

User response
Modify the TEMPLATE definition, and resubmit the job.

Return Code
8

INZU496E  ERROR: INCORRECT OUTPUT TYPE FOR COLUMN colname. OUTPUT TYPE MUST BE A CHAR OR VARCHAR LONG ENOUGH (in CAR.) TO CONTAIN LOB FILE NAME

Explanation
A LOB file field (subtype BLOBF, CLOBF, or DBCLOBF) is either not a CHAR or VARCHAR, or is not long enough to contain the LOB file reference. This message indicates the required length, which corresponds to the maximum length of the TEMPLATE statement that was used in the LOB file specification, is indicated in the message.

User response
No action is required.

Return Code
8

INZU497E  ERROR: INCORRECT LOBFILE TEMPLATE template name FOR COLUMN col. VARIABLE &UNIQ OR USER DEFINED VARIABLE RECNUM BASED ON : MUST ONLY BE PART OF THE MEMBER NAME

Explanation
The syntax contains an invalid TEMPLATE specification for a LOB file data type. The &UNIQ variable or a user variable that is set to the :RECNUM predefined variable cannot be used in the name of the PDS. They should be used in the member name.

User response
Modify the template definition, and resubmit the job.

Return Code
8

INZU498E  ERROR: INCORRECT LOBFILE TEMPLATE template name FOR COLUMN col. VARIABLE &PA.
&TSNAME. OR &SPACENAME. MUST BE PART OF THE DATASET NAME

Explanation
The template is invalid for a LOBFILE: because a separate output file is used for each partition. The template that is used for LOB file reference should contain either the partition number (variable &PA.) or the space name (&TS. or &SN. variable).

User response
Modify the template definition, and resubmit the job.

Return Code
8

INZU499E ERROR: INFORMATION WAS NOT FOUND IN THE table_name TABLE FOR THE FOLLOWING column_type COLUMN: column_name.

Explanation
Db2 HPU did not find information in the specified catalog table for a LOB column or an XML column. The table name is SYSIBM.SYSAUXRELS for a LOB column and SYSIBM.SYSXMLRELS for an XML column.

User response
Determine if the definition of the unloaded object is complete. If the object definition is complete, contact IBM Software Support.

Return Code
8

INZU501E ERROR: PARTITIONS CANNOT BE UNLOADED IN SEPARATE PDS MEMBERS

Explanation
Unloading a partitioned table space in a PDS file with a separate member for each partition is not supported.

User response
Modify the template definition and resubmit the job.

Return Code
8

INZU502W WARNING: DATA AFTER CONVERSION CAN BE TOO LONG

FOR FIELD column name, TABLE table name

Explanation
The size of the output field might be too short to contain the converted data when that data is converted in UTF8. As the data can be expand from one character to four characters, the target length must be equal or higher than the source length * 4.

User response
Accept the truncation or make one of the following changes:

- increase the length of the output field using a REFORMAT or INTO specification to get a target length equal to the source length * 4
- specify UNICODE_EXPANSION_RATIO 4 or set the VUU068/ULUNIEXP parmlib parameter to 4 to automatically resize the output length. Be aware that the latter has an impact of any Db2 HPU job that involves an encoding conversion into UTF8.

Return Code
4

INZU503E ERROR: 'ORDER BY' ON LOB OR XML COLUMN IS NOT SUPPORTED

Explanation
You cannot sort output data using an LOB or XML column as sort criteria.

User response
Modify the SYSIN and resubmit the job.

Return Code
8

INZU504I INFORMATION: COLUMN TYPE type IS NOT SUPPORTED NATIVELY

Explanation
A SELECT statement included a column type that Db2 HPU cannot natively unload. If DB2 YES is also specified, the SELECT statement is processed using SQL access.

User response
Modify the template definition and resubmit the job.

Return Code
8

INZU505I INFORMATION: COLUMN TYPE type IS NOT SUPPORTED NATIVELY

Explanation
A SELECT statement included a column type that Db2 HPU cannot natively unload. If DB2 YES is also specified, the SELECT statement is processed using SQL access.

User response
No action is required.
Return Code
0

INZU505E ERROR: SORT SIZE CANNOT BE ESTIMATED FOR SELECT STARTING AT POS (n,n) YOU MUST SPECIFY THE ESTIMATED NUMBER OF ROWS TO BE SORTED INSTEAD OF ‘VSAMSIZE’ KEYWORD

Explanation
Db2 HPU cannot estimate the number of rows to be sorted because of the characteristics of the SELECT statement. The VSAMSIZE keyword is accepted only with natively supported SELECT statements that are processed in DB2 FORCE.

User response
Specify the SORT size parameter of the SORT SYSIN KEYWORD or modify the SELECT statement.

Return Code
8

INZU506W WARNING: EXTERNAL SORT CANNOT BE PROCESSED FOR SELECT STARTING AT POS (n,n) THE COLUMN (column name) OF THE ‘ORDER BY’ CLAUSE CANNOT BE FOUND IN THE SELECTED COLUMNS

Explanation
External sort is not supported because a column that was specified in the ORDER BY clause is not in the list of selected columns.

User response
The SORT is processed internally by Db2.

Return Code
4

INZU507I HIDDEN OPTION IGNORED FOR SELECT PROCESSED BY DB2 STARTING AT POS (n,n)

Explanation
The hidden columns cannot be unloaded when the SELECT statement is processed by Db2. The HIDDEN option is ignored.

User response:
No action is required.

Return Code
0

INZU508I TABLE creator name HAS BEEN LOCKED IN SHARE MODE DURING UNLOAD PROCESS [LIMITED TO UNLOADED PARTITIONS]

Explanation
The LOCK YES parameter changed the access to the specified table. The text “LIMITED TO UNLOADED PARTITIONS” is indicated when the lock was performed only for a subset of partitions of the involved table space.

User response:
No action is required.

Return Code
0

INZU509W UNIT NAME SPECIFIED IN TEMPLATE templatename IS IGNORED WHEN USING LOB FILE REFERENCE VARIABLES IN SQL

Explanation
The TEMPLATE definition for templatename specifies a UNIT name. This template is used to generate a LOB file reference by using a file reference variable that is processed by Db2. In such a case, the UNIT name is not used to create the file that is pointed to by the file reference. To create the file on a specific unit, use DFSMS ACS routines to assign a unit name based on the dsname of the output file.

User response
Modify the TEMPLATE definition and resubmit the job.

Return Code
4

INZU510W WARNING: EXTERNAL SORT CANNOT BE PROCESSED FOR SELECT STARTING AT POS (line, col) SORT KEY EXCEEDS 4088 BYTES LENGTH

Explanation
An ORDER BY clause with a sort key that is longer than 4088 bytes was specified in a SELECT statement.
SORT EXTERNAL is ignored and the SORT is processed by Db2.

**User response:**
No action is required.

**Return Code**
4

**INZU511W** WARNING: EXTERNAL SORT CANNOT BE PROCESSED FOR SELECT STARTING AT POS (line, col) THE ORDER BY CLAUSE CONTAINS AN EXPRESSION

**Explanation**
SORT(EXTERNAL) is ignored and the ORDER BY clause is processed by Db2.

**Return Code**
4

**INZU512I** INFORMATION: IN SELECT STARTING AT POS(line, col), SORT(EXTERNAL) OPTION IS SET BY UTILITY TO PERFORM ORDER BY

**Explanation:**
The utility set the SORT(EXTERNAL) option to ORDER BY a SELECT statement because an SQL-partitioned UNLOAD with an ORDER BY statement is performed in a single output file. The SORT(EXTERNAL) option allows sorting data after SQL unloads partitions.

**User response:**
No action is required.

**Return Code**
None.

**INZU513E** ERROR: 'IFERROR NULL' CANNOT BE SPECIFIED FOR THE COLUMN name. THERE IS NO NULL INDICATOR IN THE OUTPUT FILE FOR THIS COLUMN

**Explanation:**
A column format option that specifies IFERROR NULL is defined for an output field that does not contain a null-byte indicator.

**User response:**
Either specify NULLBYTE YES to force the null-byte indicator, or specify a different value in the IFERROR option.

**Return Code**
8

**INZU515E** ERROR: EXPRESSION WITH DECFLOAT DATA CANNOT BE NATIVELY PROCESSED WITHOUT DECIMAL-FLOATING-POINT FACILITY HARDWARE

**Explanation:**
An expression that uses a DECFLOAT operand cannot be processed natively by Db2 HPU because the Decimal-Floating-Point Facility is not available on the current processor.

**User response:**
Modify the expression in the SELECT statement to allow native processing by Db2 HPU, or specify DB2 FORCE to use Db2 processing.

**Reason Code**
8

**INZU516E** NO CONSISTENT (SAME RBA) FULL IMAGE COPY FOUND FOR EACH REQUESTED PARTITION OF THE TABLE SPACE dbname.tsname

**Explanation:**
The CONSISTENT keyword was used, but Db2 HPU did not find consistent input full image copies for the unloaded partitions.

**User response:**
If a consistent image copy is not necessary, remove the CONSISTENT keyword. If a consistent input image copy is necessary, contact your DBA to ensure that such copies are created for the unloaded object. If you suspect a problem in Db2 HPU, contact IBM Software Support.

**Return Code**
8

**INZU517E** NO FULL IMAGE COPY FOUND FOR EACH REQUESTED PARTITION OF THE TABLE SPACE dbname.tsname

**Explanation:**
The PARTITIONED keyword was used, but Db2 HPU could not find a partitioned full image copy for each unloaded partition.

**User response:**
Ensure that a copy data set is available for each partition to be unloaded, or remove the PARTITIONED keyword from the UNLOAD statement, if unloading from a global image copy is appropriate.
THE template_name XML TEMPLATE FOR THE column_name COLUMN IS INCORRECT. THE &PA. VARIABLE MUST BE PART OF THE PDS OR THE LIBRARY NAME.

Explanation:
When XML data is natively unloaded from a partitioned table space in a PDS or PDSE, the template definition must contain the &PA. variable. When XML data is unloaded by using Db2 (DB2 FORCE), &PA. is not required.

User response:
Correct the template definition, and rerun the job.

The format does not start with a correct element. The allowed elements are:
• for DATE type: DD, MO, YY, YYYY
• for TIME type: HH, MI, SS
• for TIMESTAMP type: DD, MO, YY, YYYY, HH, MI, SS, NN, NNNN, NNNNNN

User response:
Correct the format.

The requested format contains an invalid element. The allowed elements are:
• for DATE type: DD, MO, YY, YYYY
• for TIME type: HH, MI, SS
• for TIMESTAMP type: DD, MO, YY, YYYY, HH, MI, SS, NN, NNNN, NNNNNN

User response:
Remove the irrelevant element from the format.

The MM element is not accepted to avoid misunderstandings, since it can stand for either months or minutes.

User response:
Replace MM by MO for month or MI for minutes according to your need.

The APPLCOMPAT string is ignored for DB2 version previous to V11 NFM.
Explaination
The request to set the CURRENT APPLICATION
COMPATIBILITY, either via the APPLCOMPAT option
specified in the GLOBAL OPTIONS block or the
VUM036/APPLCOMP parmlib parameter was ignored
because it is not supported by the Db2 version Db2
HPU is run against.

User response
Check if the unload statement makes sense and gives
an acceptable result without the CURRENT
APPLICATION COMPATIBILITY being supported.
Modify your unload statement and/or your setting of
VUM036/APPLCOMP accordingly.

Return Code
4

INZU530I LOAD STATEMENT : INDDN FROM LOADOPT IGNORED BECAUSE OF LOADINDDN(YES).

Explanation:
This message indicates that both the LOADINDDN YES
and INDDN options or parameters have been
specified. In this case, only the LOADINDDN option is
taken into account.

User response:
If you want the INDDN load option to be processed,
then you need to specify LOADINDDN NO in an option
block of the unload statement.

Return Code
0

INZU531I CHECK IF THE SEMICOLON REQUIRED AT THE END OF GLOBAL STATEMENT IS CORRECTLY LOCATED.

Explanation:
The terminating semicolon character of the GLOBAL
statement might be missing.

User response:
Check the global statement

Return Code
0

INZU532I BIF_COMPATIBILITY SETTINGS :
native_mode_value FOR NATIVE MODE AND sql_mode_value FOR SQL MODE AS A RESULT OF THE SETTING FOR BIF_COMPATIBILITY AND

Explanation:
This message informs that the BIF_COMPATIBILITY
values used for native and SQL modes are different,
which might lead the Db2 HPU output to be
undesirably depending on the execution mode (SQL
mode vs native mode). native_mode_value and the
sql_mode_value can be either V9_DECIMAL_VARCHAR or CURRENT.

The mentioned value explains what the behavior for
CHAR and VARCHAR function is in terms of equivalent
settings of Db2 BIF_COMPATIBILITY parameter. The
indicated situation results from the settings of the
following parameters:

• for the native mode, the Db2 HPU
  BIF_COMPATIBILITY and Db2 HPU
  BIF_COMPATIBILITY_FROM_DB2_SETTINGS
  subparameters of the Db2 HPU parmlib parameter
  VUU030/ULOPTNS

• for the SQL mode, the Db2 BIF_COMPATIBILITY
  parameter of the concerned Db2 subsystem.

User response
Unless you have specific needs requiring the
mentioned discrepancy between SQL and native
modes, consider changing the value of some of the
following parameters

• the Db2 HPU BIF_COMPATIBILITY or Db2 HPU
  BIF_COMPATIBILITY_FROM_DB2_SETTINGS
  subparameters of the Db2 HPU parmlib parameter
  VUU030/ULOPTNS to change the native mode
  behavior

• the Db2 BIF_COMPATIBILITY parameter of the
  concerned Db2 subsystem to change the SQL mode
  behavior For Db2 version V11 NFM and later, the
  most efficient way to ensure consistency is to set the
  Db2 HPU
  BIF_COMPATIBILITY_FROM_DB2_SETTINGS
  (V11_AND_LATER) in the parmlib parameter
  VUU030/ULOPTNS.

Return Code
0

INZU537E TEMPLATE ERROR: TEMPLATE 

template_base_name IS INCOMPATIBLE WITH 
alternate_template TEMPLATE
Explanation
A base template and its alternate template specified through the LIMIT option are not compatible due to one of the following inconsistent situation:

• The data set name pattern for one of them includes the &PART variable while the other does not. This is not supported as the same policy concerning partitions, i.e. whether the data set name depends on the partition number or not, must apply to all the TEMPLATE statements chained via the LIMIT option.

• one of the TEMPLATE statement request stacking with the STACK YES option while the other TEMPLATE statement does not or explicitly specifies STACK NO.

User response
Change the TEMPLATE statements to solve the inconsistent situation and make sure:

• the data set name patterns of the indicated templates are consistent regarding the use of the &PART variable. To do so, either add the &PART variable to the data set name pattern that does not include it or remove it from the data set name pattern that includes it.

• the indicated TEMPLATE statements are consistent regarding the stacking.

Return Code
8

| INZU538I | SWITCH TO THE ALTERNATE FORMAT alternate_format FOR THE SELECT # select-number TABLESPACE dbname.tsname |

Explanation:
FORMAT INTERNAL was specified along with the ALTERNATE FORMAT sub-option for a SELECT statement specifying a FROM LISTDEF clause. As the INTERNAL format is not suitable or supported for the indicated table space, the related data is unloaded in the indicated alternate format.

User response:
None. If you don’t understand why the alternate format was chosen, search for preceding error messages (either INZU392E, INZU401E or INZU391E) that gives an explanation.

Return Code
0

| INZU539E | ALTERNATE FORMAT IS ONLY ALLOWED UNLOADING FROM LISTDEF |

Explanation:
The ALTERNATE FORMAT option is specified and the required FROM LIST clause is missing. As specifying two formats via the ALTERNATE FORMAT option does not make much sense for a SELECT statement specifying a FROM table_name clause, ALTERNATE FORMAT is allowed only if a FROM LIST clause is specified.

User response:
Remove the ALTERNATE FORMAT option and replace the FORMAT INTERNAL specification by the format specified as an alternate format if the internal is not suitable for the single SELECT statement involved. If you want to take benefit from the run time determination of the relevant format for a single unload statement, replace the FROM table_name clause by a FROM LIST(listdef_name) clause with the name of a LISTDEF statement matching only the table to unload.

Return Code
8

| INZU540E | ALTERNATE FORMAT IS ONLY ALLOWED FOR INTERNAL FORMAT |

Explanation:
ALTERNATE FORMAT was specified but the required FORMAT INTERNAL option is missing.

User response:
either specify FORMAT INTERNAL or remove the unintended ALTERNATE FORMAT option.

Return Code
8

| INZU541I | DISPLAY OF IMPLICIT PARAMETERS ACTIVATED IN DDN ddname. |

Explanation:
This message indicates that the values of the implicit parameters are written to the indicated ddname as requested by the value specified either by the VUU083/DISPPARM parmlib parameter or the SYSIN DISPLAY_PARMLIB option from the technical parameters options block.

User response:
None

Return Code
0

| INZU542E | FORMAT DELIMITED ERROR FOR SELECT select_number STARTING AT POS(line,column) MERGING |

Chapter 8. Troubleshooting 459
INCONSISTENT RECORDS WITH OPTION TITLE ON INTO OUTPUT DDN=outdd IS NOT ALLOWED

Explanation:
This message is issued when the indicated output ddname for a SELECT statement in DELIMITED format with the TITLE ON option is merged with the output of another unload. This operation is not allowed.

User response:
Remove the TITLE ON option from DELIMITED format block or specify an output dedicated to the unload with the TITLE ON option.

Return Code
8

INZU543W  TEMPLATE WARNING: IN TEMPLATE template_name LIMIT OPTION IS IGNORED FOR data_set_type

Explanation:
The LIMIT option of the indicated TEMPLATE statement was ignored as it is not supported for the data set type to be allocated. The data set types for which the LIMIT option of the TEMPLATE statements is not supported are:

- LOB file (LOB file reference)
- XML file (LOB file reference)
- LOAD control file (which generation is requested by the LOADDN option).

User response:
Remove the LIMIT option from the TEMPLATE statement if it is not used for other data sets for which the LIMIT option is supported. Otherwise, specify a specific TEMPLATE statement with no LIMIT option for the indicated data set type.

Return Code
4

INZU544W  THE OUTPUT FIELD WILL BE EXPANDED WITH A RATIO OF ratio AS REQUESTED BY THE SETTING OF THE PARMLIB PARAMETER VUU068/ULUNIEXP. THE OUTPUT FIELD MIGHT NOT BE LONG ENOUGH FOR FIELD column-name, TABLE table-name

Explanation:
This message indicates that the output length for the specified field has been expanded as requested by the parmlib parameter VUU068/ULUNIEXP to a size that can be too short to unload the converted data.

User response:
If you are sure that the output length is large enough, no action is required. If you want to avoid the return code 4, set the UNICODE_EXPANSION_RATIO in the HPU command to the same value than the parmlib parameter VUU068/ULUNIEXP. If the output length is not suitable as truncation can occur, set the expansion ratio to 4, either setting the UNICODE_EXPANSION_RATIO parameter in the HPU command, either setting the parmlib parameter VUU068/ULUNIEXP.

Return Code
4

INZU545I  THE OUTPUT FIELD WILL BE EXPANDED WITH A RATIO OF ratio AS REQUESTED BY THE SETTING OF THE PARAMETER UNICODE_EXPANSION_RATIO. THE OUTPUT FIELD MIGHT NOT BE LONG ENOUGH FOR FIELD column-name, TABLE table-name

Explanation:
This message indicates that the output length for the specified field has been expanded as requested by the keyword UNICODE_EXPANSION_RATIO in the HPU command to a size that can be too short to unload the converted data.

User response:
if you are sure that the output length is large enough, no action is required, otherwise set the parameter UNICODE_EXPANSION_RATIO to 4.

Return Code
0

INZU546I  THE OUTPUT FIELD WILL BE EXPANDED WITH A RATIO OF ratio AS REQUESTED BY THE SETTING OF THE PARAMETER UNICODE_EXPANSION_RATIO OR THE PARMLIB PARAMETER VUU068/ULUNIEXP. FIELD column-name, TABLE table-name

Explanation:
This message indicates that the output length for the specified field has been expanded as requested by the keyword UNICODE_EXPANSION_RATIO in the HPU.
command or by the setting of the parmlib parameter VUU068/ULUNIEXP.

User response:
None.

Return Code
0

INZU547W  lob/xml NOT CORRECTLY UNLOADED DUE TO INCONSISTENCY WITH BASE TABLESPACE. REASON abend-reason. INCORRECT DATA WAS WRITTEN INTO output-location

Explanation
For a specific row, the unload of LOB or XML data in native mode failed because of an inconsistency between the base table space and the LOB or XML related table space. This situation can occur if the consistency of these table spaces has not been ensured by having the Db2 buffer flushed to the underlying data sets (e.g. by specifying the QUIESCE YES option in the UNLOAD command) and no concurrent update of the object occurs during the unload (e.g. by specifying the LOCK YES option in the UNLOAD command). The output-location indicates the location of the LOB or XML data affected by the issue. The output-location is specified as indicated in the table below:

<table>
<thead>
<tr>
<th>output-location</th>
<th>the incorrect data was written to</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN=dataset-name(member-name)</td>
<td>the indicated PDS member</td>
</tr>
<tr>
<td>path-name</td>
<td>the indicated path</td>
</tr>
<tr>
<td>DSN=dataset-name - RECORD NUMBER = nnn</td>
<td>the nnn th record of the indicated data set name that was allocated through a TEMPLATE statement</td>
</tr>
<tr>
<td>DDN=ddname - RECORD NUMBER = nnn</td>
<td>the nnn th record of the data set allocated through a JCL DD statement which ddname is indicated</td>
</tr>
</tbody>
</table>

The mentioned reason code, i.e. abend-reason, is meant for Advanced troubleshooting by the IBM support if you suspect the reported situation is not true; especially if you have made sure of the consistency of the involved table spaces. If the

VUU084/ULLOBBHV parmlib is set to IGNORE, the unload process continues and the subsequent INZU548I message sums up the number of such errors encountered during the unload.

If the VUU084/ULLOBBHV parmlib is set to STOP, the unload process will stop as reported by the subsequent INZU549E message.

User response:
Unless this situation where the base table data is correct and some of the LOB or XML data is incorrect is acceptable for the subsequent use of the output data, no action is required. Otherwise, submit the job again after having ensured of the table spaces consistency or run the unload in SQL mode by specifying the DB2 FORCE option.

Return Code
4

INZU548I  nnn LOB/XML NOT CORRECTLY UNLOADED DUE TO INCONSISTENCY BETWEEN LOB/XML TABLESPACE AND dbname.tsname BASE TABLESPACE

Explanation
This message indicates how many times the situation reported by the INZU547W message – i.e. inconsistent LOB or XML unloaded without stopping the unload processing - was encountered during the unload of a base table contained in the indicated table space. Db2 HPU does not issue the INZU547W message every time it encounters the situation as the maximum number allowed for these messages is set by the VUU085/ULLOBBHM parmlib parameter. This message allows you to determine the amount of inconsistent LOB or XML data actually in the unloaded data.

User response:
None.

Return Code
0

INZU549E  INCONSISTENCY BETWEEN LOB/XML TABLESPACE AND BASE TABLESPACE. PROCESSING STOPS ACCORDING TO PARAMETER VUU084/ULLOBBHV.

Explanation:
For a specific row, the unload of LOB or XML data in native mode failed because of an inconsistency between the base table space and the LOB or XML
related table space. This situation can occur if the consistency of these table spaces has not been ensured by having the Db2 buffer flushed to the underlying data sets (e.g. by specifying the QUIESCE YES option in the UNLOAD command) and no concurrent update of the object occurs during the unload (e.g. by specifying the LOCK YES option in the UNLOAD command). As the parameter VUU084/ULLOBBHV is set to STOP, the unload processing stops.

**User response:**
Submit the job again after having ensured of the table spaces consistency or run the unload in SQL mode by specifying the DB2 FORCE option. You may also want to skip this error and let the unload process continue by setting the VUU084/ULLOBBHV to IGNORE.

**Return Code**
8

**INZU550E** TAPEUNIT PARAMETER: DEFAULT UNIT COUNT CAN NOT BE GREATER THAN THE MAXIMUM NUMBER OF DEVICES FOR ONE UNIT.

**Explanation:**
This message is displayed when TAPEUNIT(unit(maximum_devices_number, default_devices_number)) is specified either in parmlib or in sysin, and default_devices_number is greater than the maximum number of devices for this unit. default_devices_nb must be less than or equal to maximum_devices_number.

**User response:**
Change the TAPEUNIT parameter/option to ensure that default_devices_number is less than or equal to max_devices_nb.

**Return Code**
8

**INZU551E** THE "SET CURRENT APPLICATION COMPATIBILITY = applcomp_string." STATEMENT REQUESTED THROUGH APPLCOMPAT OPTION OR VUM036/APPLCOMP PARAMETER FAILED. SEE PRECEDING DB2 MESSAGES FOR DETAILS.

**Explanation:**
A request to set the CURRENT APPLICATION COMPATIBILITY register was made either by specifying the APPLCOMPAT applcomp_string or by setting the VUM036/APPLCOMP parmlib parameter to applcomp_string. The attempt to run the corresponding "SET CURRENT APPLICATION COMPATIBILITY = applcomp_string" statement failed. The unload process stops.

**User response:**
Check the SQL messages issued above for more details about the reason why the statement failed. If an SQLCODE=-713 error is reported, replace the value of applcomp_string by a value supported by your Db2 subsystem. For any other sql code error, refer to related Db2 documentation to determine the relevant solution.

**Return Code**
8

**INZU552E** LOCK CANNOT BE PERFORMED FOR SELECT no AT POS(line,col)

**Explanation:**
The LOCK feature requested for the indicated SELECT statement processed in SQL mode could not be performed. It occurred because the EXPLAIN command required to determine the tables involved in the SELECT statement failed.

**User response:**
Check previous messages about the failure of the EXPLAIN command to determine and solve the related problem.

**Return Code**
8

**INZU553I** LOCK IS NOT PERFORMED FOR SELECT no AT POS(line,col) AS VUU088/ULLOCKMO IS SET TO NATIVE_ONLY

**Explanation:**
The LOCK feature requested for the indicated SELECT statement processed in SQL mode is not honored because the VUU088/ULLOCKMO parmlib parameter is set to NATIVE_ONLY that disables the LOCK feature in SQL mode.

**User response:**
Set VUU088/ULLOCKMO to ALL to allow Db2 HPU to perform the LOCK feature or remove the LOCK feature request (e.g. by specifying LOCK NO) to prevent this message from being issued.

**Return Code**
0

**INZU555W** THE PARMLIB PARAMETER VUU074/ULLOOKTE IS FORCED TO LOCK DUE TO UNSUPPORTED SELECT TO LOCK
**Explanation:**  
The SQL LOCK technique was chosen to honor the LOCK feature despite the setting of the VUU074/ULLOCKTE parameter that requests another technique. This occurred because Db2 HPU can only use one locking technique per Db2 HPU invocation (i.e. JCL step) and the LOCK feature requested for a SELECT statement processed in SQL mode can only be honored with the SQL LOCK technique.

**User response:**  
As the setting of VUU074/ULLOCKTE is ignored if different from LOCK when the LOCK feature is requested for both SELECT statements processed in SQL mode and SELECT statements processed in SQL mode, consider running unload steps only involving SELECT statements processed with the same mode (either native or SQL). If the LOCK feature is actually not desired for SELECT statements processed in SQL mode, consider disabling the feature for the SQL mode only by setting VUU088/ULLOCKMO to NATIVE_ONLY.

**Return Code**  
4

---

**INZU556E**  
THE DDNAME ddname DOES NOT ALLOCATE AN IMAGE COPY OF THE EXPECTED PARTITION NUMBER (part-num)

**Explanation:**  
According to the UNLOAD command, the mentioned ddname must allocate the image copy data set for the indicated partition number but Db2 HPU determined that the allocated data set does not contain the image copy of a partition with the expected number.

**User response:**  
Change the DD statement to allocate the indicated partition number or correct the UNLOAD command (e.g. the COPYDDN option or the PART/SQLPART option) so that the ddname in question is associated to the partition number of the currently allocated data set.

**Return Code**  
8

---

**INZU557I**  
OPTIONAL 'ORDER CLUSTER' CLAUSE IGNORED FOR location_of_select_statement

**Explanation:**  
The indicated SELECT statement being processed specifies an ORDER CLAUSE incompatible with the SQL mode that applies to this SELECT statement. As either ORDER CLUSTER OPTIONAL was specified or ORDER CLUSTER MANDATORY was not specified and ORDER_CLUSTER_BHV(OPTIONAL) is specified in the value of the VUU030/ULOPTNS parmlib parameter, the ORDER CLAUSE is considered as optional. As a consequence, Db2 ignores the ORDER CLUSTER clause by removing it before passing the SELECT statement to Db2.

**User response:**  
None.

**Return Code**  
0

---

**INZX005**  
PARTITION NO. nnnn IS BEING UNLOADED

**Explanation:**  
A partition is being unloaded.

**User response:**  
No action is required.

**Return Code**  
0

---

**INZX006**  
TABLESPACE UNLOAD PHASE STARTED

**Explanation:**  
The TABLESPACE UNLOAD phase started.

**System action**  
Utility processing continues normally.

**User response:**  
No action is required.

**Return Code**  
0

---

**INZX007**  
TABLESPACE UNLOAD PHASE ENDED, nnnn ROWS PROCESSED

**Explanation:**  
This message indicates normal completion of the TABLESPACE UNLOAD phase and shows the number of rows processed.

**User response:**  
No action is required.

**Return Code**  
0

---
Explanation
The specified partition was unloaded, and nnnn records were processed.

User response:
No action is required.

Return Code
0

INZX062  SORT ACTIVATED VIA E15

Explanation
The SORT phase was activated by using input exit E15.

User response:
No action is required.

Return Code
None.

INZX063  SORT ACTIVATED VIA SORTIN

Explanation
The SORT phase was activated by using a dynamically allocated SORTIN file.

User response:
No action is required.

Return Code
None.

INZX064  SORT ACTIVATED VIA CA-SRAM

Explanation
The SORT phase was activated by using the CA-SRAM interface.

User response:
No action is required.

Return Code
None.

INZX068  sort-identifier SORT FAILED, RC=return code, sort-parameters, L=nnn

Explanation
The sort for sort-identifier failed.

sort-identifier
Specifies (in most of the cases) the name of the table space or index space which rows are sorted.

return code
Specifies the return code that was issued by the SORT utility.

sort-parameters
Specifies the parameters that were used when activating the SORT utility.

nnn
Specifies the length of the records to be sorted.
For example, INZX068 T1SQXXX-04 SORT FAILED, RC=16, FILSZ=E616068, DYNALLOC=(SYSDA,2), L=107.

User response
Examine the job output, and determine the cause of the error.

Return Code
Abend.

INZX073  ONE OR MORE PAGES IN ERROR WERE FOUND, RC = nn

Explanation
Db2 HPU encountered one or more pages in error, which it could not correct.

User response
Examine the job output, and determine the cause of the error.

Return Code
8

INZX081  IMAGE COPY IS BEING READ FROM ddname or table_space_name FLASHCOPY IS BEING READ

Explanation
Db2 HPU is reading an image copy of the table space from a file that is identified either by the ddname or from a FlashCopy.

User response:
No action is required.

Return Code
None.

INZX089  DDN ddname [TS tsname][PART part] : n RECORDS WRITTEN [IN
**Explanation**

This message indicates how many rows have been unloaded into the indicated ddname, for the mentioned partition or set of partitions (part) if relevant (no partition number is mentioned if a non-partitioned table space is involved or if all the partitions are unloaded), of the table space named tsname (the tablespace name might not be indicated for an unload processed in SQL mode for which a table space could not be identified). This message is issued because VUU050/ULFRQMSG is set to a value greater than zero. It is issued each time M new rows have been unloaded, where M is the value of PARMLIB parameter VUU050/ULFRQMSG.

Depending on the content of PARMLIB parameter VUU050/ULFRQMSG, the following optional information can be displayed:

- **IN hh:mm:ss**: Indicates the elapsed time since the first record of the mentioned table was unloaded. This information is displayed if TOTAL is specified in PARMLIB parameter VUU050/ULFRQMSG or if neither TOTAL nor DELTA is specified.
- **INTERVAL hh:mm:ss**: Indicates the elapsed time since the previously issued message or since the first record was unloaded. This information is displayed if DELTA is specified for VUU050/ULFRQMSG.

**User response:**

No action is required.

**Return Code**

None.

**Note:** The message is preceded/prefixed by the time when it was issued. It is either written by the system time when it is issued to the console or by a field formatted as hh:mm:ss: when the message is written in a file.

**INZX090**

**DDN ddname [TS tsname][PART part] : n RECORDS WRITTEN [IN hh:mm:ss][INTERVAL hh:mm:ss], UNLOAD DONE.**

**Explanation**

This message, issued at the end of the unload, indicates how many rows have been unloaded into the indicated ddname, for the mentioned partition or set of partitions (part) if relevant (no partition number is mentioned if a non-partitioned table space is involved or if all the partitions are unloaded), of the table space named tsname (the tablespace name might not be indicated for an unload processed in SQL mode for which a table space could not be identified).

Depending on the content of PARMLIB parameter VUU050/ULFRQMSG, the following optional information can be displayed:

- **IN hh:mm:ss**: Indicates the elapsed time since the first record of the mentioned table was unloaded. This information is displayed if TOTAL is specified in PARMLIB parameter VUU050/ULFRQMSG or if neither TOTAL nor DELTA is specified.
- **INTERVAL hh:mm:ss**: Indicates the elapsed time since the previously issued message or since the first record was unloaded. This information is displayed if DELTA is specified for VUU050/ULFRQMSG.

**User response:**

No action is required.

**Return Code**

None.
Return Code
None.

INZX104  indexspace INDEX PARTITION nnn SCANNED, ppp RIDS EXTRACTED

Explanation:
This message is issued for informational purposes only.

System action:
Processing continues.

User response:
No action is required.

Return Code:
None.

INZX105  indexspace INDEX SCAN TURNED OFF, REASON 0Xreason-seq, nnn RIDS EXTRACTED

Explanation:
The indexscan feature that was triggered while processing a logical unload with an ORDER clause has been turned off. This occurs because Db2 HPU has determined the indexscan feature was not worth being used. This normally occurs when the clustering index of the involved table space is not well organized. The message indicates the number (nn) of RIDs sequentially read from the index before Db2 HPU decided to turn the indexscan feature off, reason and seq codes that are meant from advanced troubleshooting by IBM support if needed.

User response:
No action is required if the performance of the related unload is not a problem but consider reorganizing or rebuilding the indicated partition of the index to make sure the indexscan feature is used. If Db2 HPU persists in turning the index scan feature off, contact your IBM support and provide the joblog of the unload including the SYSnnnnn ddname file.

Return Code:
None.

INZX110  ESTIMATED SIZES (ROWS):

Explanation
This message is issued for informational purposes only.

System action
Processing continues.

User response:
No action is required.

Return Code
None.

INZX111  nnnnK / ppppK REQUESTED ON uu UNIT(S), SIZE=(aaaa * bbbb)

Explanation
This message is issued for informational purposes only.

System action
Processing continues.

User response:
No action is required.

Return Code
None.

INZX112  ddname UNIT-COUNT = cnt USED INSTEAD OF OPTIMAL UNCNT = cnt AS PER limit SPECIFICATION

Explanation
The allocation of the data set handled by the mentioned ddname could not be done with the number (nn) of RIDs sequentially read from the index before Db2 HPU decided to turn the indexscan feature off, reason and seq codes that are meant from advanced troubleshooting by IBM support if needed.
estimated number of units because it exceeded one of the following limits:

- **LIMUNIT**: set via the VUX010/LIMUNIT parmlib parameter or its counterpart LIMUNIT in SYSIN, for dataset to allocate on disk
- **MAXTUNIT**: set via the VUX017/MAXTUNIT parmlib parameter or its counterpart MAXTUNIT in SYSIN, for dataset to allocate on tape

**User response**

If subsequent ABENDSx37 affects the data set, consider the following actions depending on the estimated size mentioned in a previous INZX111 message concerning the same ddname:

- if the estimated size is obviously much bigger than needed, either do nothing or specify YES for the RTSESTIM feature or force directly specify an estimated of the expected amount of data to unload with the MAX_EXPECTED_ROWS option.
- if the estimated size if accurate, change the value specified for the either the WRKMXPR and/or LIMUNIT feature. If no ABENDSx37 occur, consider changing the value anyway to prevent such likely issues from occurring.

**Return Code**

0

---

**INZX113**

**Explanation**

The allocation of the data set handled by the mentioned ddname could not be done with the estimated primary quantity because it exceeded the following limit:

- **WRKMXPR** : set via the VUX019/WRKMXPR parmlib parameter or its counterpart WRKMXPR in SYSIN

**User response**

If subsequent ABENDSx37 affects the data set, consider the following actions depending on the estimated size mentioned in a previous INZX111 message concerning the same ddname:

- if the estimated size is obviously much bigger than needed, either do nothing or specify YES for the RTSESTIM feature or force directly specify an estimated of the expected amount of data to unload with the MAX_EXPECTED_ROWS option.
- if the estimated size if accurate, change the value specified for the either the WRKMXPR and/or LIMUNIT feature. If no ABENDSx37 occur, consider changing the value anyway to prevent such likely issues from occurring.

**Return Code**

0

---

**INZX133**

**Explanation**

During the unload processing of partition **nnn** of a partitioned table space, Db2 HPU was unable to find the data set with the ddname that corresponds to the image copy of partition **nnn**. The partition has not been unloaded and no row has been written into the output file.

**User response**

This situation might be intentional. Otherwise check that the ddname corresponding to the image copy of partition **nnn** has been allocated in the JCL. When unloading the partitions of a table space into independent data sets, one image copy data set per partition must be provided as input data sets. (See “Db2 HPU-allocated ddnames” on page 87.)

**Return Code**

4

---

**INZX134**

**Explanation**

This message is issued to inform that the processing of multiple LDS table space has led to an allocation error message against a non-existing LDS that has been issued and that Db2 HPU has checked that this was a normal situation considering the actual number of LDS the table space is made of.

This situation occurs when the last LDS of the table space has exactly reached the maximum size of the data set defined for this table space. The reported situation has no impact on the correctness of the unload output.

**User response**

None.

**Return Code:**

None
INZX135

VERSION 0 MISSING FOR dbname.tsname. IF ALTER TABLE PROCESSED RECENTLY, TRY LATER OR REORGANIZE THE TABLESPACE

Explanation
This message is sent to the console before abending with an USER ABEND 1022 (R15=0) in the following situation:

- the unloaded table is versioned as it has been altered to modify its DDL
- the LDS VSAM dataset contains ONLY record in version 0 (no row has been updated since the ALTER TABLE process)
- no description of the version 0 has been found in the Db2 catalog. This case occurs when the insertion of the version 0 is delayed so the job can run successfully later. However, to insure a successful execution, the table space has to be reorganized.

User response:
Rerun the job later or reorganize the table space.

Return Code:
ABEND U1022 with R15=0

INZX206

tsname DSNUTILB FAILED
RETURN CODE = nn ***

Explanation
The Db2 DSNUTILB utility cannot continue because it encountered an error.

User response
Examine the Db2 messages in the SYSPRINT file; see the Db2 Universal Database for z/OS Messages and Codes manual for a complete description of possible return codes and reason codes.

Return Code
8

INZX207

tsname DSNUTILB ABENDED ***
reason code = X'hhhhhhhhhh' ***

Explanation
The Db2 DSNUTILB utility cannot continue because it abended.

User response
Examine the Db2 messages in the SYSPRINT file; see the Db2 Universal Database for z/OS Messages and Codes manual for a complete description of possible return codes and reason codes.

Return Code
Db2 reason code

INZZ001S

(abend_sequence) ABEND IN
program_name -
CODE=system_completion_code
user_completion_code
REASON=reason_code

Explanation
One or more abends occurred, where

abend_sequence
Indicates the sequence in which this abend occurred during the current process. If a dump is taken for this abend, message INZZ100I(abend_sequence) SDWA is issued at the beginning of the dump.

program_name
Indicates the name of the program that was running when the abend occurred. If the program name could not be determined, ******** is displayed.

system_completion_code
Indicates the system completion code.

user_completion_code
Indicates the user completion code.

reason_code
Indicates the hexadecimal reason code.

If a dump is not taken for the abend, this message is followed by either INZZ002I, INZZ003I, or INZZ004I messages. If a problem occurs while the dump is being taken, this message is followed by either INZZ011W, INZZ012W, or INZZ019W messages. If the dump was successfully taken, a message is not issued.

User response
Complete the steps for the following system completion codes:

S000
See the explanation for the user completion code that is indicated in the message text in “Db2 HPU user abend codes ” on page 359.

S878
Not enough memory was available to complete the process.

- If IEA705I mentions a problem during FREEMAIN, after you ensure that the VZM009 parameter has the correct value, collect a
SYSABEND DUMP if the problem recurs, and contact IBM Software Support.

- If IEA705I mentions a problem during GETMAIN, increase the REGION value or reduce the memory that is necessary for your job by using the instructions in “Decreasing the amount of necessary memory” on page 285.

For other system completion codes, see z/OS MVS System Codes.

Return Code
None.

INZZ002I  (abend_sequence) NO DUMP REQUESTED

Explanation:
A dump was not requested for this abend, so a dump cannot be taken. The abend was specified in the preceding message INZZ001S with the same abend sequence. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
No action is required.

Return Code
None.

INZZ003I  (abend_sequence) NO DUMP PRODUCED FOR THIS SYSTEM CODE

Explanation:
The settings indicated that a dump was not to be taken. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
If a dump is necessary to diagnose the problem, change the settings so that a dump will be taken, and rerun the job with a SYSABEND DD card.

Return Code
None.

INZZ004I  (abend_sequence) SYSABEND UNAVAILABLE, NO DUMP TAKEN

Explanation:
A dump could not be taken because the SYSABEND ddname was not available. A SYSABEND DD card might not be in the JCL. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
Check system messages that might give details about the problem with the SYSABEND ddname. Correct the problem, and rerun the job.

Return Code
None.

INZZ005I  (abend_sequence) NO DUMP PRODUCED DUE TO PREVIOUS ERROR

Explanation:
A dump was not taken. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
Check previous messages INZZ004I, INZZ01nx, where n is a number and x is a character, for the reason why a dump was not taken.

Return Code
None.

INZZ009S  (????) ABEND OCCURED - NO DETAIL AVAILABLE

Explanation:
An abend occurred and was trapped. However, more detailed information is unavailable. In the message text, ???. indicates that the sequence of this abend could not be determined.

User response:
Check other system messages for more information about solving the problem.

Return Code
None.

INZZ011W  (abend_sequence) ERROR DURING DUMP, SYSABEND IS TRUNCATED

Explanation:
An error occurred while opening the SYSABEND file. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
Check message INZZ013I and other system messages to determine how to solve the problem.

Return Code
None.
Explanation:
A failure occurred while the dump was being written. The SYSABEND file does not contain the whole DUMP. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
Check message INZZ013I and other system messages to determine how to solve the problem.

Return Code
None.

INZZ013I (abend_sequence)
CODE=system_completion_code
user_completion_code
REASON=reason_code

Explanation
This message is issued after message INZZ011W or INZZ012W and explains which abend prevented the dump from being taken correctly.

abend_sequence
Indicates the sequence in which this abend occurred during the current process. If a dump is taken for this abend, message INZZ100I(number_of_abends) SDWA is issued at the beginning of the dump.

system_completion_code
Indicates the system completion code of the abend that prevented the dump from being taken correctly.

user_completion_code
Indicates the user completion code of the abend that prevented the dump from being taken correctly.

reason_code
Indicates the hexadecimal reason code of the abend that prevented the dump from being taken correctly.

If a dump is not taken for the abend, this message is followed by either INZZ021I, INZZ003I, or INZZ004I messages. If a problem occurs while the dump is being taken, this message is followed by either INZZ011W, INZZ012W, or INZZ019W messages. If the dump was successfully taken, a message is not issued.

User response:
Follow the instructions in the related INZZ011W or INZZ012W messages.

Return Code
None.

INZZ019W (????) NO DUMP TAKEN - ABEND DURING OPENING SYSABEND - NO DETAIL AVAILABLE

Explanation:
A dump could not be taken, and the reason could not be determined. In the message text, ???? indicates that the sequence of this abend could not be determined.

User response:
Check other system messages to determine why the abend occurred. If message INZZ021I was issued, the job might remain in a wait state. In this case, consider cancelling the job.

Return Code
None.

INZZ021I (abend_sequence) PROCESSING. ANOTHER PROCESS IS DELAYED

Explanation:
An abend occurred while another one was being processed. Processing of the new one is postponed. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
No action is required.

Return Code
None.

INZZ022I (abend_sequence) PROCESSING. ANOTHER PROCESS IS WAITING

Explanation:
An abend is being processed while at least one other abend is waiting to be processed. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.

User response:
No action is required.

Return Code
None.

INZZ023I (abend_sequence) PROCESS ENDING. ANOTHER PROCESS IS RESUMED

Explanation:
The processing of an abend finished, and the processing of an abend that had been postponed is about to start. In the message text, abend_sequence indicates the sequence in which this abend occurred during the current process.
**User response:**
No action is required.

**Return Code**
None.

**INZZ100I (abend_sequence) SDWA :**

**Explanation:**
This message is written at the beginning of the data that is related to the dump that was taken for one of the abends that occurred during the process. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

**User response:**
No action is required.

**Return Code**
None.

**INZZ120S POSIX ENVIRONMENT INITIALIZATION FAILED.**

**Explanation:**
An error occurred during initialization of the POSIX environment when the LE directive POSIX(ON) has been set. This might be caused by a lack of RACF authorization as the user that invoked Db2 HPU must be permitted to the BPX.MAP FACILITY class profile. The reason code indicated by the message is for advanced diagnosis by your IBM technical support.

**User response:**
Look for prior RACF messages and fix the mentioned issue, if any. Otherwise, Contact IBM Software Support, supply the joblog and mention this message and the indicated reason code.

**Return Code**
8

**INZ0666F FATAL ERROR : text**

**Explanation**
An unexpected exception has occurred. A string might be included at the end of the message. These messages are intended for the IBM Software Support, and are not documented.

**User response**
Contact IBM Software Support.

**Return Code**
16

**PLI200 *** SYSIN FILE COULD NOT BE OPENED**

**Explanation**
Db2 HPU was unable to open the SYSIN data set.

**User response**
Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL. Or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

**Return Code**
12

**PLI201 *** COMMAND OUT OF SEQUENCE**

**Explanation**
Db2 HPU detected that the specified command is out of sequence.

**User response**
Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

**Return Code**
8

**PLI202 *** ERROR READING SYSIN FILE**

**Explanation**
Db2 HPU encountered an error when reading the SYSIN data set.

**User response**
Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

**Return Code**
12

**PLI203 *** INVALID COMMAND**
Explaination
Db2 HPU encountered an invalid command.

User response
Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code
None.

PLI210 *** REFERENCED MODULE DOES NOT CONTAIN INZPLIB CSECT

Explaination
Db2 HPU was unable to locate the INZPLIB CSECT within the referenced module.

User response
Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tools Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code
8

PLI217 *** UPDATE ERROR, EXECUTION ABORTED

Explaination
Db2 HPU encountered an internal update error.

User response
Contact IBM Software Support, and supply the return and reason codes.

Return Code
12

PLI218 *** REFERENCED MODULE NOT FOUND IN LIBRARY

Explaination
Db2 HPU was unable to find the referenced module within the specified library.

User response
Check the names of the load module libraries.

Return Code
8

PLI300 *** SYSPRINT FILE COULD NOT BE OPENED

Explaination
Db2 HPU was unable to open the SYSPRINT data set.

User response
Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tools Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code
8

PLIB001E PARMLIB NOT FOUND. INFPLIB ddname MISSING OR PROGRAM NOT CUSTOMIZED

Explaination
Db2 HPU is unable to find the PARMLIB and, therefore, cannot be started.

System action
None.

User response
Insert a DD card with the ddname INFPLIB into the JCL, into the CLIST or into the REXX procedure. Or use the INZPLIB0 program to customize the DSNAME of the PARMLIB (see the INZPARM sample JCL).

PLIB002E SYNTAX ERROR ON LINE nnnn IN THE MEMBER mmmmmm of THE PARMLIB

Explaination
Db2 HPU is unable to start because an error was detected in the PARMLIB. See message PLIB100I for the DSNAME of the PARMLIB.

System action
None.

User response
Modify the syntax error in the specified line, then restart Db2 HPU.
PLIB003E MEMBER mmmmmmm NOT FOUND IN THE PARMLIB

Explanation
Db2 HPU is unable to start because one member is not in the PARMLIB; see message PLIB100 for the DSNAME of the PARMLIB.

System action
None.

User response
Verify that the specified member is in the PARMLIB.

PLIB011E SYNTAX ERROR IN THE PARMLIB FOR THE VARIABLE vvvvvvv

Explanation
Db2 HPU detected a syntax error for one of the PARMLIB's variables.

System action
None.

User response
Correct the syntax error of the specified PARMLIB variable, then restart Db2 HPU.

PLIB012E vvvvvvv MUST BE A VARIABLE OF THE PARMLIB OR A JCL DD CARD

Explanation
Db2 HPU is unable to start because a mandatory PARMLIB parameter is missing.

System action
None.

User response
Insert the missing variable (designing a file) into the PARMLIB, or insert a DD card that specifies the variable name as ddname into the JCL. Then restart Db2 HPU.

PLIB013E ALLOCATION FAILED FOR THE DSNAME dataset-name (VARIABLE vvvvvvv IN THE PARMLIB)

Explanation
Db2 HPU is unable to start because a file allocation failed.

System action
None.

User response
Verify that the file exists, or specify another file name into the corresponding variable of the PARMLIB. Then restart Db2 HPU.

PLIB100I PARMLIB =dataset-name

Explanation
A PARMLIB name error has been detected.

System action
None.

User response
Refer to the previous message that describes this error.

PLIB999S PARMLIB SEVERE ERROR : MODULE = mmmmmmmmm RETURN CODE = rr - REASON CODE = cc CONTACT THE IBM SUPPORT CENTER

Explanation
Db2 HPU encountered an internal error.

System action
None.

User response
Contact IBM Software Support, and supply the return and reason codes.
Chapter 9. Reference

These reference topics are designed to provide you with quick access to information about Db2 HPU syntax and data.

Customization reference

Refer to information about Db2 HPU parameters, dsnames, and templates during the customization process.

Topics:
- “Recommended settings” on page 475
- “Discover process parameters” on page 479
- “Db2 HPU library dsnames” on page 480
- “Db2 HPU Db2 parameters” on page 481
- “Db2 HPU output data parameters” on page 499
- “Db2 HPU Db2 Administration Tool and Db2 Launchpad parameters” on page 534
- “Db2 HPU conversion parameters” on page 534
- “Db2 HPU file management parameters” on page 535
- “Db2 HPU tuning parameters” on page 542
- “Db2 HPU sort parameters” on page 550
- “Customization templates” on page 552

Recommended settings

The behavior of Db2 HPU can be adjusted to your needs by means of many parameters.

For easier setting, many parameters have a default value chosen to be either usual or optimal. Except for specific requirements, the default value can be kept.

However, the behavior of Db2 HPU is always enriched and improved by offering new features but also new behaviors for an existing feature. In the latter case, the new behavior is enabled by a new value of an existing parameter or by a new parameter. In order to keep the behavior of Db2 HPU jobs, the former default behavior is not changed, even though the new behavior is an improvement.

As a consequence, the default value of some parameters is not the more advisable value.

When using Db2 HPU for the first time or in order to benefit from its improvements, consider the recommendations concerning the settings below.
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Sub-Parameter Name</th>
<th>Default Value</th>
<th>Recommended Value</th>
<th>Benefit of recommended value</th>
<th>Advice applicable from Db2 HPU 5.1 maintenance level (1)</th>
<th>Scope of the recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VUM033/OPTRGPRT</td>
<td>/</td>
<td>NO</td>
<td>YES</td>
<td>Better performance of logical unload involving partitions and run in native mode</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>BIF_COMPATIBILITY _FROM_DB2_SETTNGS</td>
<td>NO</td>
<td>V11_AND_LATER</td>
<td>Consistency with the Db2 settings and a consistency of SQL and native modes</td>
<td>PI92055</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>COPYDDN_STRICT</td>
<td>NO</td>
<td>YES</td>
<td>More rigorous behavior</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>DSNNTAULStrict</td>
<td>NO</td>
<td>YES</td>
<td>Better compliance with DSNNTAUL</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>SEARCH_UCATALOGED_IC</td>
<td>NO</td>
<td>YES</td>
<td>Capability to exploit uncatalogued image copies through the COPYDDN LAST_IC or COPYDDN - integer option</td>
<td>PH15431</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>TAPELBI</td>
<td>NOSPANDED</td>
<td>ALL</td>
<td>Wider support of LBI</td>
<td>PH04468</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>TEMPLATE_LIMIT</td>
<td>NO</td>
<td>YES</td>
<td>Compliance with Db2</td>
<td>PH03781</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>TEMPLATE_SUBSTITUTION</td>
<td>MEMBER_NAME</td>
<td>GROUP_NAME</td>
<td>Compliance with Db2</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>TRUE_UNICODE</td>
<td>NO</td>
<td>YES</td>
<td>More rigorous behavior</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>USE_ADMIN_INFO_SYSPARM</td>
<td>NO</td>
<td>YES</td>
<td>Compliance with Db2</td>
<td>PH15568</td>
<td>All</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>USER_SWITCH_METHOD</td>
<td>AS</td>
<td>RACF</td>
<td>Compliance with Db2</td>
<td>PH03885</td>
<td>All</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Sub-Parameter Name</td>
<td>Default Value</td>
<td>Recommended Value</td>
<td>Benefit of recommended value</td>
<td>Advice applicable from Db2 HPU 5.1 maintenance level (1)</td>
<td>Scope of the recommendation</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>VUU033/ULCHKCPY</td>
<td></td>
<td>NO</td>
<td>YES</td>
<td>Better reliability by enforcing automated controls of the input image copies</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU036/GBLPARAL</td>
<td></td>
<td>NO</td>
<td>YES</td>
<td>Better overall performance expected for unloads run in native mode from several partitions into a single data set.</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU044/SUBTKSOF</td>
<td></td>
<td>NO</td>
<td>YES</td>
<td>Better overall performance expected for unloads run in native mode from several partitions into a single data set.</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU054/CTRLLIBM</td>
<td></td>
<td>NO</td>
<td>YES</td>
<td>Compliance with Db2 by ensuring full compatibility of generated LOAD statements with Db2 LOAD</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU057/OPALLFMT</td>
<td></td>
<td>NO</td>
<td>YES</td>
<td>Better consistency - Db2 HPU settings for USER format also apply to all Db2 HPU native built-in formats</td>
<td>GA</td>
<td>All</td>
</tr>
<tr>
<td>VUU089/INTORUL1</td>
<td></td>
<td>COLTYPE</td>
<td>REFORMAT</td>
<td>More intuitive and consistent behavior of the REFORMAT clause feature</td>
<td>PH19262</td>
<td>All</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Sub-Parameter Name</td>
<td>Default Value</td>
<td>Recommended Value</td>
<td>Benefit of recommended value</td>
<td>Advice applicable from Db2 HPU 5.1 maintenance level (1)</td>
<td>Scope of the recommendation</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>------------------------------</td>
<td>--------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>AUTO_UNC</td>
<td>0</td>
<td>Value greater than zero according to your storage policy</td>
<td>Better control of resources allocated by Db2 HPU</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td></td>
<td>NT_MAX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>CNTLCARDS_DB2</td>
<td>NO_DDL</td>
<td>DDL</td>
<td>Compliance with UNLOAD PLUS</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>U_NULLCHAR</td>
<td>6F</td>
<td>Value specified in your former UNLOAD PLUS settings</td>
<td>Better compliance with UNLOAD PLUS</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>U_NULLCHAR</td>
<td>6F</td>
<td>Value specified in your former UNLOAD PLUS settings</td>
<td>Compliance with UNLOAD PLUS</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>VUU030/ULOPTNS</td>
<td>U_NULLTY</td>
<td>T1</td>
<td>Value specified in your former UNLOAD PLUS settings</td>
<td>Compliance with UNLOAD PLUS</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>VUU073/MAXERR</td>
<td></td>
<td>0</td>
<td>Any non zero value</td>
<td>Lower system resource consumption by limiting the number of messages that are written into the spool</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>VUU083/DISPPARM</td>
<td></td>
<td>NO</td>
<td>IF,ddname or YES,ddname with the ddname of your choice</td>
<td>Easier trouble shooting More comprehensiv report</td>
<td>PH01897</td>
<td>Site</td>
</tr>
<tr>
<td>VUU087/DISPXPL</td>
<td></td>
<td>NO</td>
<td>IF,ddname or YES,ddname with the ddname of your choice</td>
<td>Easier trouble shooting More comprehensiv e reporting</td>
<td>PH19457</td>
<td>Site</td>
</tr>
<tr>
<td>VUX039/ZIIP</td>
<td></td>
<td>OFF</td>
<td>Any value different from OFF</td>
<td>Savings of regular CPU time by enabling zIIP processor exploitation</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>Parameter name</td>
<td>Sub-Parameter Name</td>
<td>Default Value</td>
<td>Recommended Value</td>
<td>Benefit of recommended value</td>
<td>Advice applicable from Db2 HPU 5.1 maintenance level (1)</td>
<td>Scope of the recommendation</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>VUX042/ZIIPMXPL</td>
<td>/</td>
<td>0</td>
<td>Value greater than the number of available zIIP processors - optimal value to be determined from benchmarking</td>
<td>Better performance regarding elapse time</td>
<td>GA</td>
<td>Site</td>
</tr>
<tr>
<td>VUX043/ZIIPMXPU</td>
<td>/</td>
<td>0</td>
<td>Value greater than the number of available zIIP processors - optimal value to be determined from benchmarking</td>
<td>Better performance regarding elapse time</td>
<td>GA</td>
<td>Site</td>
</tr>
</tbody>
</table>

(1) GA stands for General Availability and means that the advice is applicable to Db2 HPU 5.1 without any specific maintenance.

**Discover process parameters**

The Discover process parameters on the Discover Customized Product Information panel (CCQPDSC) in Tools Customizer are required to run the Discover EXEC.

You can accept the default values for the Discover process parameters, or you can customize them based on your specific requirements. All Discover process parameters are required.

The following list describes the Discover process parameters.

**Name of the PARMLIB library of a previous version (4.2.0 and later)**

The name of PARMLIB library (INZPLIB/INFPPLIB) of a previous version. Use this field when the previous installation was done through TCz (Db2 HPU version 4.2.0 and later). Example: oldhlq.SINZAMP. There is no default value.

**Previous data set of variables to be retrieved (4.1.0 and earlier)**

The name of the data set that contains the variable to be discovered (INZRSAVE or INZTVAR member). Use this field when the previous installation was done without TCz (Db2 HPU version 4.1.0 and earlier). Example: oldhlq.SINZRAV(INZRSAVE). There is no default value.

**Previous data set with customized file dsnames (4.1.0 and earlier)**

The name of the data set that contains customized files dsnames (INZTDSN, member). Use this field when the previous installation was done without TCz (Db2 HPU version 4.1.0 and earlier). Example: oldhlq.SINZAMP(INZTDSN). There is no default value.

**Verbose mode for discover procedure........OFF (OFF,ON)**

Specify ON to enable display of the trace information issued by the discover process. Otherwise, specify OFF.
Db2 HPU library dsnames

The Db2 HPU library dsnames section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the dsnames of the Db2 HPU libraries.

The following list describes the dsnames:

**Load modules library (SINZLOAD)**
This parameter is required. It specifies the dsname of the load modules library (SINZLOAD).

In previous releases of Db2 HPU, the name of this parameter was VIZ003.

**APF load modules library (SINZLINK)**
This parameter is required. It specifies the dsname of the APF load modules library (SINZLINK).

In previous releases of Db2 HPU, the name of this parameter was VIZ004.

**Samples library (SINZSAMP)**
The dsname of the samples library (INZSAMP).

**PARMLIB library (INZPLIB/INFPLIB)**
This parameter is required. It specifies the dsname of the PARMLIB library (INZPLIB). Do not use the SINZSAMP library. This library is handled by Db2 HPU under the INFPLIB ddname.

**Requirement:** This library must be the same as the one that is specified in the INZTDSN member. If you do not use this library, Db2 HPU cannot read the customization correctly.

In previous releases of Db2 HPU, the name of this parameter was VIZ007.

**Product's DBRM library (SINZDBRM)**
This parameter is required. It specifies the dsname of the DBRM library (SINZDBRM).

In previous releases of Db2 HPU, the name of this parameter was VIZ012.

**REXX EXEC library (SINZCLST)**
This parameter is required. It specifies the dsname of the REXX EXEC library (SINZCLST).

**ISPF LOAD modules library (SINZLLIB)**
This parameter is required to run the Db2 HPU interactive component. It specifies the dsname of the ISPF load module library (SINZLLIB) that contains the load modules.

In previous releases of Db2 HPU, the name of this parameter was VIZ013.

**ISPF messages library (SINZMLIB)**
This parameter is required to run the Db2 HPU interactive component. It specifies the dsname of the ISPF messages library (SINZMLIB) that contains the messages.

In previous releases of Db2 HPU, the name of this parameter was VIZ015.

**ISPF panels library (SINZPLIB)**
This parameter is required to run the Db2 HPU interactive component. It specifies the dsname of the ISPF panels library (SINZPLIB) that contains the ISPF panels.

In previous releases of Db2 HPU, the name of this parameter was VIZ016.

**ISPF skeletons library (SINZSLIB)**
This parameter is required to run the Db2 HPU interactive component. It specifies the dsname of the ISPF skeletons library (SINZSLIB) that contains the skeletons.

In previous releases of Db2 HPU, the name of this parameter was VIZ017.

**ISPF tables library (SINZTLIB)**
This parameter is required to run the Db2 HPU interactive component. It specifies the dsname of the ISPF tables library (SINZTLIB) that contains the tables.

In previous releases of Db2 HPU, the name of this parameter was VIZ018.
Db2 HPU Db2 parameters

The Db2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the Db2 parameters that are used by Db2 HPU.

The Db2 HPU parameters are designated in three ways.
- a long designation that describes what the parameter is used for
- two short names used as a reference especially within the messages texts:
  - one has the following format Vxxnnnn where xx are characters and nnnn are figures
  - the other one is a short label of 8 characters

You can accept the default values for the Db2 parameters, or you can configure them based on your specific requirements. Some Db2 parameters are mandatory and must be configured.

The following list describes the common Db2 parameters. The parameters are listed in the following format:

description (Vxxnnnn/parameter-name)

**Percentage of displaced pages in inline FIC (VUX023/PGDFIN)**

This parameter is optional. It specifies the estimated value of the percentage of displaced pages in inline full-image copies (FIC).

Valid values are 0 - 100. The default value is 20.

The corresponding SYSIN keyword is PGDFIN in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX023/PGDFIN.

**Use real-time statistics tables for size estimation (VUX036/RTSESTIM)**

This parameter is optional. It specifies whether the content of the real-time statistics tables must be taken into account to estimate the number of rows of the table space to be unloaded. This estimate is used to allocate the output data set when a TEMPLATE is used, to set the sort program parameters when an ORDER BY clause is specified, or both.

Valid values are YES and NO:
- YES: the content of the real-time statistics tables, when available, is used to estimate the number of rows of the table space to be unloaded. This normally gives the more accurate estimate.
- NO: the content of the real-time statistics tables is not used to estimate the number of rows of the table space to be unloaded. Choose this value only if instructed to do so by the IBM support or if this setting is inherited from a functional installation.

This parameter is also referred to as VUX036/RTSESTIM.

The default value is YES.

**Maximum degree of parallelism for SELECT (VUX031/DB2PARAL)**

This parameter is optional. It specifies the parallelism degree for an UNLOAD command when several SELECT statements are processed by using Db2 access (using DB2 FORCE or DB2 YES with unsupported SELECT statements). The value that you specify indicates the maximum number of SELECT statements that are processed in parallel. The value that you specify indicates the maximum number of SELECT statements that are processed in parallel.

This parameter is also referred to as VUX031/DB2PARAL.

Valid values are 1 - 65535. The default value is 5.

**Db2 plan name of the Call Level Interface (VUX044/CLIPLAN)**

This parameter is optional. It specifies the plan name of the Db2 for z/OS Call Level Interface (CLI).

**Note:** The DSNACLI plan can be created using the sample job in member DSNTIJCL of the DB2 SDSNSAMP data set. Refer to the Db2 for z/OS Call Level Interface Guide and Reference for further information.
The default value is DSNACLI.
This parameter is also referred to as VUX044/CLIPLAN.

**Application plan for Db2 HPU (VUM011/PLANOBJT)**
This parameter is required. It specifies the plan name for the interactive application and the batch jobs.
Valid values are valid Db2 plan names.
This parameter is also referred to as VUM011/PLANOBJT.

**Package collection for Db2 HPU (VUM030/COLLOBJT)**
This parameter is required. It specifies the name of the collection that is used to bind the Db2 HPU packages.
This parameter is also referred to as VUM030/COLLOBJT.

**Partitions range filter optimization (VUM033/OPTRGPRT)**
This parameter is optional. It specifies whether Db2 HPU has to activate the OPTIMIZE_PART_RANGE functionality. This feature allows to enhance the performance of logical unloads run in native mode against a partitioned table space and that specify a WHERE clause. This feature exploits the Db2 for z/OS part range table populated by the EXPLAIN PLAN statement to determine in which partitions the rows matching the WHERE clause are located so that Db2 HPU only reads the relevant partitions. The use of this feature requires a PLAN_TABLE and a DSN_PGRANGE_TABLE tables are available at run time. The "User who issues EXPLAIN PLAN statements (VUM035/EXPLNUSR)" setting determines what PLAN_TABLE and DSN_PGRANGE_TABLE tables are populated and then read by Db2 HPU when the feature is enabled. Make sure the setting for "User who issues EXPLAIN PLAN statements (VUM035/EXPLNUSR)" also suits your needs. Specify one of the following values:
- YES: to enable the OPTIMIZE_PART_RANGE functionality.
- NO: to disable the OPTIMIZE_PART_RANGE functionality.
The default value is NO.
This parameter is also referred to as VUM033/OPTRGPRT

**DELETE or KEEP rows inserted in EXPLAIN tables (VUM034/EXPLNTB)**
This parameter is optional. It specifies whether the rows inserted by Db2 HPU into the EXPLAIN tables when the OPTIMIZE_PART_RANGE or the LOCK functionality is used have to be deleted or kept afterwards.
Specify one of the following values:
- DELETE: to delete the rows inserted by Db2 HPU from the EXPLAIN tables.
- KEEP: to keep the rows inserted by Db2 HPU into the EXPLAIN tables.
The default value is DELETE
This parameter is also referred to as VUM034/EXPLNTB

**User who issues EXPLAIN PLAN statements (VUM035/EXPLNUSR)**
This parameter is optional. It specifies on behalf of which userid the EXPLAIN PLAN statements are issued. Db2 HPU may issue EXPLAIN PLAN statements when the OPTIMIZE_PART_RANGE or the LOCK functionality is used. The userid specified by this parameter must:
- own a PLAN_TABLE and a DSN_PGRANGE_TABLE EXPLAIN tables.
- have the execute privilege on the DSNACLI plan.
- have the privilege to run an EXPLAIN statement on the SELECT statements involved in the UNLOAD commands. This is ensured if either one of the following conditions is fulfilled:
  - the userid has the privilege to run the SELECT statements from the UNLOAD command
  - the userid has the EXPLAIN privilege VZD001.
Specify one value for each Db2 subsystem that you defined with variable
Specify one of the following values:
• CURRENT_USER: the user who submits the unload job is used to run the EXPLAIN PLAN on the table to be unloaded. With this settings, every user that submits a unload job triggering the OPTIMIZE_PART_RANGE feature must own a PLAN_TABLE and a DSN_PGRANGE_TABLE EXPLAIN tables.

• user_name: a specific user name is used to run the EXPLAIN PLAN on the table to be unloaded. The name can be 1 - 7 characters. With this setting, any unload job triggering the OPTIMIZE_PART_RANGE feature uses the PLAN_TABLE and DSN_PGRANGE_TABLE EXPLAIN tables owned by this user_name.

This parameter is also referred to as VUM035/EXPLNUSR.

Application compatibility (VUM036/APPLCOMP)

This parameter is optional. It specifies the compatibility level behavior for SQL. Specify a value valid for the CURRENT APPLICATION COMPATIBILITY special register. This parameter is ignored by Db2 HPU for Db2 version prior to V11 NFM.

User who reads encrypted VSAM data set (VUM037/CRYPTUSR)

The userid on behalf of which Db2 HPU directly accesses to the encrypted Db2 VSAM data set involved in the unload jobs run in native mode.

For a Db2 like behavior - i.e. any user that has the requested privilege to run a given SELECT statement involving encrypted table spaces or index spaces can also run a Db2 HPU unload specifying the same SELECT statement -, set this parameter with a userid that has the authorization to access the key label for any of the encrypted data set involved by your Db2 HPU jobs.

userid

the access to the encrypted data set is done on behalf of this specified userid.

CURRENT_USER

the access to the encrypted data set is done on behalf of the userid of the submitter of the Db2 HPU job.

Valid values are CURRENT_USER and a valid 1-8 character userid.

The default value is CURRENT_USER.

Owner of the plan created for Db2 HPU (VUM012/PLANOWN)

This parameter is required. It specifies the name of the owner of the plan to be used to bind the Db2 HPU plan.

Specify one value for each Db2 subsystem that you defined.

This parameter is also referred to as VUM012/PLANOWN.

Grant on the plan created for Db2 HPU (VUX011)

This parameter is required. It specifies whether to use the GRANT TO PUBLIC or the GRANT TO USER statement to grant privileges to the plan that was created for Db2 HPU.

Specify one value for each Db2 subsystem that you defined with variable VZD001.

Valid values are PUBLIC and USER.

This parameter is also referred to as VUX011.

User who quiesces the table space to be unloaded (VUM031/QUITUSR)

This parameter is optional. It specifies on behalf of which userid the following Db2 actions are performed:

• QUIESCE utility against the table space to be unloaded
• sequence of START/STOP commands against the table space to be unloaded
• SQL LOCK against the table to be unloaded and a START RO command against the related table space

Specify one of the following values:
**CURRENT_USER**

The user who submits the job is used to run the QUIESCE utility on the table space to be unloaded.

**USER name**

A specific user name is used to run the QUIESCE utility on the table space to be unloaded. The name can be 1 - 7 characters.

This parameter is also referred to as VUM031/QUITUSR.

The default value is CURRENT_USER.

**User who issues SELECT by using dynamic SQL (VUM032/ACTLGUSR)**

This parameter is optional. It specifies the user who will issue the dynamic SQL statements against the Db2 catalog. This userid must have the EXECUTE privilege on the DSNACL plan and user profile that contains an OMVS segment.

**CURRENT_USER**

The user who submits the job is used to issue the SQL statements against the Db2 catalog.

**user_name**

A specific user name is used to issue the SQL statements against the Db2 catalog. This userid name can be 1 - 7 characters.

**PLAN_OWNER**

The owner of the plan that was created for Db2 HPU - specified through VUM012/PLANOWN - is used as the userid to access the Db2 catalog. The value of VUM012/PLANOWN must be a valid TSO user ID of 1 - 8 characters.

Specify one value for each Db2 subsystem that you defined with variable VZD001.

The default value is CURRENT_USER.

This parameter is also referred to as VUM032/ACTLGUSR.

**User who runs the DISPLAY command (VUM028/DISPLUSR)**

This parameter is optional. It specifies the user who will run the DISPLAY command.

**CURRENT_USER**

The user who submitted the job is used to run the DISPLAY command. CURRENT_USER is the default value.

**USER name**

A specific user name is used to run the DISPLAY command. The name can be 1 - 7 characters.

Specify one value for each Db2 subsystem that you defined with variable VZD001.

The default value is CURRENT_USER.

This parameter is also referred to as VUM028/DISPLUSR.

**Use Db2 to process unsupported SELECT statements (VUU011/ULSEDB2)**

This parameter is optional. It specifies whether Db2 will process SELECT statements when the statements are not supported by Db2 HPU. This parameter is used only if the DB2 option is not specified in the SYSIN. When COPYDDN STRICT(YES) is specified in the VUU030/ULOPTNS parameter, unloading data from the table by using Db2 instead of unloading data from the image copy is not allowed. Therefore, if you are unloading from an image copy and COPYDDN STRICT(YES) is specified, the value of the VUU001 parameter is forced to NO.

Specify one of the following values:

**NO**

SELECT statements that are not supported by Db2 HPU will not be processed by Db2.

**YES**

SELECT statements that are not supported by Db2 HPU will be processed by Db2, unless DB2 NO was specified in the SYSIN of Db2 HPU.

The default value is YES.
This parameter is also referred to as VUU011/ULSEDB2.

**Lock the tables in the table space (VUU012/ULLOCK)**
This parameter is optional. It specifies whether to lock the tables in the table space.

Specify one of the following values:

- **NO**
  Tables in the table space are not to be locked unless YES was specified in the SYSIN of Db2 HPU.

- **YES**
  Tables in the table space are locked unless NO was specified in the SYSIN of Db2 HPU.

  Specifying LOCK NO does not preclude Db2 from taking locks if SQL Access is used to access the data.

  The default value is NO.

This parameter is also referred to as VUU012/ULLOCK.

**Quiesce the table space (VUU013/ULQSCE)**
This parameter is optional. It specifies whether to quiesce the table space.

Specify one of the following values:

- **NO**
  The table space is not quiesced unless YES was specified in the SYSIN of Db2 HPU.

- **YES**
  The table space is quiesced unless NO was specified in the SYSIN of Db2 HPU.

  The default value is NO.

*Attention:* Db2 HPU operates on the physical VSAM data set level that is outside of Db2. If you run Db2 HPU on a table in which a row was just inserted, the unloaded data might not contain the row that was inserted. The unloaded data does not show the row because Db2 might not have externalized the data to DASD yet. This situation can happen when you use Db2 HPU without issuing a QUIESCE (or STOP) on the object. Be careful when using QUIESCE NO.

This parameter is also referred to as VUU013/ULQSCE.

**Degree of parallel processing (VUU021/ULDEGREE)**
This parameter is applicable only when Db2 extracts data. It specifies the number of parallel tasks or I/O operations that Db2 can use to extract data from a partitioned table space.

Specify one of the following values:

- **1**
  Parallelism is not used.

- **ANY**
  Db2 HPU decides whether parallelism will be used.

- **CURRENT_DEGREE**
  Db2 HPU must keep the default value for the current degree that is set in the Db2 customization. If the PARMLIB variable is set to CURRENT_DEGREE, Db2 HPU will not issue any SET CURRENT DEGREE statements before issuing the SELECT statement.

  The default value is CURRENT_DEGREE.

  The corresponding SYSIN parameter is ULDEGREE in the GLOBAL OPTIONS block.

  This parameter is also referred to as VUU021/ULDEGREE.

**Default scheme for UNLOAD TABLESPACE (VUU023/UNLSCHEM)**
This parameter is optional. It specifies the unload format for the data.

Specify one of the following values:
ASCII
Specifies that the unloaded data must be in ASCII format. Db2 HPU uses the ASCII CCSID of the subsystem, unless you override it by specifying the CCSID option in the SYSIN of Db2 HPU.

ASIS
Specifies that the data is unloaded in its original format. If the specification for the underlying table space cannot be determined (for example, if the data is processed by Db2), the CCSID that is returned by a standard prepare statement in SQLDA is used. You can also override ASIS by specifying the CCSID keyword.

Specifying ASIS does not mean that a conversion is not required. Conversion might still be required when columns that are not CHAR, VARCHAR, GRAPHIC, and VARGRA types are unloaded in an external format and when the schema of the unloaded table space is not system-EBCDIC.

Db2 HPU first converts the value to the external format (in system-EBCDIC), and the result is converted to the table space schema.

Similarly, the padding characters or field separators (FORMAT DELIMITED) are in system-EBCDIC by default. They are also converted to the table space schema if the table space schema is not EBCDIC.

Conversion is also required when the table space CCSID is not the same as the SYSIN CCSID and when the SELECT statement causes C'constants'.

EBCDIC
Indicates that the data is unloaded in EBCDIC format. Db2 HPU uses the EBCDIC CCSID of the subsystem unless you override it by specifying the CCSID keyword.

UNICODE
Indicates that the data is unloaded in UNICODE format. Db2 HPU uses the UNICODE CCSID of the subsystem unless you override it by specifying the CCSID option.

The default value is EBCDIC.

This parameter is also referred to as VUU023/UNLSCHNM.

Quiesce process (VUU028/ULQSCEBH)
This parameter is optional. It specifies whether the process of updating physical objects in linear data sets (LDS) must be forced or only attempted.

Specify one of the following values:

FORCE
The process of updating physical objects in LDSs is forced, which means that the object is quiesced if possible; otherwise, a STOP/START is forced. The default value is FORCE.

TRY
The process of updating physical objects in LDSs is attempted, which means that the object is quiesced if possible; otherwise, the processing terminated in error.

The default value is FORCE.

This parameter is also referred to as VUU028/ULQSCEBH.

Db2 HPU additional features (VUU030/ULOPTNS)
This parameter is optional. Use this parameter to activate functions for Db2 HPU syntax. This parameter can receive a list of optional parameters, separated by commas, that modify Db2 HPU behavior. Use a continuation comma at the end of each line or fill in the line up to the end.

LOADINDDN(YES)
INDDN ddname cards are automatically generated according to the ddnames specified by the UNLOAD command for the data sets containing the unloaded data. If a TEMPLATE is used to allocate the output file, Db2 HPU also generates a TEMPLATE statement to allocate the ddname used by the INDDN cards generated by the LOADINDDN(YES) feature. To avoid any invalid duplicate specification of INDDN cards, any INDDN option specified via the tableoptions of a LOADOPT option from the OPTIONS block or via the “Options for global LOAD statement (VUU019/ULOPTLDT)” parmlib parameter is ignored.
LOADINDDN(NO)
No INDDN card is automatically generated according to the ddnames specified by the UNLOAD command for the data sets containing the unloaded data. If the LOADINDDN(YES) option is not specified by the UNLOAD command, the INDDN option specified via the tableoptions of a LOADOPT option from the OPTIONS block or via the “Options for global LOAD statement (VUU019/ULOPTLDT)” parmlib parameter are taken into account.

DSNTIAULSTRICT(YES)
The SELECT statements that are coded with FORMAT DSNTIAUL will function as if they are coded as FORMAT DSNTIAUL STRICT.

DSNTIAULSTRICT(NO)
The SELECT statements that are coded with FORMAT DSNTIAUL are not changed into FORMAT DSNTIAUL STRICT.

Attention: Changing the behavior of the DSNTIAUL format might affect the content of the unloaded data. Use the DSNTIAUL(YES) parameter and DSNTIAUL(NO) parameter with caution. See “DSNTIAUL block syntax and description” on page 210 for a description of the difference between specifying FORMAT DSNTIAUL and specifying FORMAT DSNTIAUL STRICT.

DELIMITEDGRAPHEXT(YES)
In the DELIMITED format, GRAPHIC and VARGRAPHIC columns are unloaded in ASCII or in EBCDIC as GRAPHIC EXTERNAL data, including the SO/SI characters.

DELIMITEDGRAPHEXT(NO)
In the DELIMITED format, GRAPHIC and VARGRAPHIC columns are unloaded in ASCII or in EBCDIC as GRAPHIC data, without the SO/SI characters.

The DELIMITEDGRAPHEXT parameter is used only for Db2 HPU syntax. It has no impact on Fast Unload orUnload Plus syntax and has no impact when GRAPHIC data is unloaded in UNICODE.

PADDING_STRICT(YES)
The previous syntax for the PADDING option that allowed you to code a two-byte hexadecimal constant, such as PADDING x'hhhh', is not accepted. The default value for the DBCS padding character is always the DBCS space that was converted in the output CCSID.

COPYDDN_STRICT(YES) or COPYDDN_STRICT(NO)
You can use the COPYDDN option to require that Db2 HPU unloads the data from an image copy. Through the DB2 option or its counterpart ULSEDB2 parmlib parameter, you can also give directions concerning the processing mode (native mode or SQL mode) to use when unloading. The unload from an image copy is not possible in SQL mode. Therefore, COPYDDN and SQL mode are incompatible.

However, the early Db2 HPU versions (i.e. up to version V3.2) did not reject such an incompatible request and considered the processing mode request as the more meaningful/prevailing request, thus ignoring the COPYDDN option when the SQL mode was required.

This led to the undesirable following effects:
• the data from the table was unexpectedly unloaded from the table instead of from the image copy when:
  − DB2 FORCE was specified in the unload statement
  − DB2 YES applied (either through the DB2 option in the UNLOAD statement or the ULSEDB2 parmlib parameter) and the SQL statement is not supported by Db2 HPU.
• the data previously unloaded from the table was suddenly unloaded from the image copy because a Db2 HPU maintenance had made the involved SELECT statement supported.

From the version V4.1, Db2 HPU was fitted with a clearer and more reliable behavior when the COPYDDN feature is concerned. With this straightforward behavior, COPYDDN and DB2 YES (or ULSEDB2=YES if no DB2 option) or DB2 FORCE are considered incompatible. When COPYDDN is specified, the native must be requested, either by specifying the DB2 NO option or by default if the ULSEDB2 parmlib parameter is set to NO.
Specify COPYDDN\_STRICT(YES) to enable the straightforward behavior and make sure the data is unloaded from an image copy whenever COPYDDN is specified.

Specify COPYDDN\_STRICT(NO) to keep the former behavior if you have existing jobs that might specify COPYDDN and require the SQL mode.

A later Db2 HPU release will make the straightforward behavior corresponding to COPYDDN\_STRICT(YES) the unique behavior. The COPYDDN\_STRICT(NO) setting will then be ignored.

**BY\_SQL\_ONLY(WITH\_UR)**

SELECT statements with the WITH\_UR clause are always processed in SQL mode. SELECT statements without the WITH\_UR clause are always processed in native mode.

**GBLPARAL\_MAXPART\_SET\_NULL(YES) or GBLPARAL\_MAXPART\_SET\_NULL(NO)**

Processing subsets of partitions (when you use MAXPART or ULMAXPAR when the specified value is less than the number of partitioned to be unloaded) and unloading without parallelism because GBLPARAL (NO) is specified are incompatible.

When VUU060/ULMAXPAR n and VUU036/GBLPARAL NO are specified together and are applicable to the unload in progress, GBLPARAL NO is ignored, and the data is unloaded without parallelism at the partition level unless GBLPARAL\_MAXPART\_SET\_NULL(YES) is specified.

Specify GBLPARAL\_MAXPART\_SET\_NULL(YES) to disable the processing of the subset of partitions and unload the data without partition parallelism instead. If you specify either the PARALLELISM or the MAXPART keyword in the UNLOAD statement, specifying GBLPARAL\_MAXPART\_SET\_NULL(YES) does not affect parallelism.

**AUTO\_UNCNT\_MAX(unit\_count\_value)**

Sets the maximum value to be used as the unit count when the unit count is dynamically determined by Db2 HPU. The value is determined by Db2 HPU when the value is not specified for the UNCNT option in the TEMPLATE statement.

Valid values are 0 - 59.

**CHECK\_CCSID\_STRICT(YES|NO)**

Specifies with CCSID classes, such as SBCS or MIXED or DBCS, are allowed.

**YES**

Using an irrelevant CCSID class, such as SBCS or MIXED or DBCS, in the CCSID(ccsid\_sbc, ccsid\_mixed, ccsid\_dcb) option is not allowed.

**NO**

NO Using an irrelevant CCSID class, such as SBCS or MIXED in the CCSID(ccsid\_sbc, ccsid\_mixed[, ccsid\_dcb]) option is allowed except for ccsid\_dcb.

The default value is YES.

The default values are LOADINDDN(NO), DSNTIAULSTRICT(NO), DELIMITEDGRAPHEXT(NO), COPYDDN\_STRICT(NO), and BY\_SQL\_ONLY().

**INSTREAM\_XML\_AS\_CLOB(YES|NO)**

Specifies with CCSID classes, such as SBCS or MIXED or DBCS, are allowed.

**INSTREAM\_XML\_AS\_CLOB(YES)**

XML data unloaded in-stream are unloaded as CLOB data.

**INSTREAM\_XML\_AS\_CLOB(NO)**

XML data unloaded in-stream are unloaded as XML unless FORMAT DSNTIAUL STRICT is requested in which case the XML data is unloaded as CLOB.

The default value is YES.

**INSTREAM\_XML\_AS\_CLOB(YES|NO)**

Specifies with CCSID classes, such as SBCS or MIXED or DBCS, are allowed.
INSTREAM_XML_AS_CLOB(YES)
xml data unloaded in-stream are unloaded as CLOB data.

INSTREAM_XML_AS_CLOB(NO)
xml data unloaded in-stream are unloaded as XML unless FORMAT DSNTIAUL STRICT is requested in which case the XML data is unloaded as CLOB.

The default value is YES.

TRUE_UNICODE(YES|NO)
TRUE_UNICODE(YES)
When conversion into unicode is requested with no target CCSID specification, the CSSID UTF8 is used as a target CCSID.

TRUE_UNICODE(NO)
When conversion into unicode is requested with no target CCSID specification, the default SBCS CSSID of the subsystem is used as a target CCSID.

The default value is NO.

CHECK_CCSID_STRICT(YES|NO)
CHECK_CCSID_STRICT(YES)
The use of irrelevant CCSID class (i.e. SBCS or MIXED or DBCS) in the CCSID (ccsid_sbc, ccsid_mixed, ccsid_dbcs) option is prohibited.

CHECK_CCSID_STRICT(NO)
The use of irrelevant CCSID class (i.e. SBCS or MIXED or DBCS) in the CCSID (ccsid_sbc, ccsid_mixed, ccsid_dbcs) option is allowed.

The default behavior of Db2 HPU is CHECK_CCSID_STRICT (YES).

U_NULLCHAR
UNLOAD PLUS NULLCHAR parameter default. This parameter is used for UNLOAD PLUS syntax and allows you to change the null indicator character in the output record.

Allowed values: xx where xx is any valid hexadecimal constant, except X'00'.

The default value is 6F.

U_NULLTYPE
UNLOAD PLUS NULLCHAR parameter default. This parameter is used for UNLOAD PLUS syntax and allows you to specify the location and length of the null indicator field in the output record.

Allowed values: T1/L1/T2/L2

The default value is T1.

DB2_SETTINGS_SOURCE (AUTO|DSNHDECP)
AUTO
The way Db2 HPU determines the Db2 settings depends on the Db2 version. Prior to Db2 V9, the DSNHDECP member from the DSNEXIT library is loaded; in Db2 V9 and later versions, the GETVARIABLE SQL function is used.

DSNHDECP
the DSNHDECP member from the DSNEXIT library is used regardless of the Db2 release.

The default value is DB2_SETTINGS_SOURCE (AUTO).

TEMPLATE_SSID_SUBSTITUTION (MEMBER_NAME/GROUP_NAME)
Specifies how the &SSID and &SS variables used in the name-expression of DSN option from the TEMPLATE statements are evaluated.

MEMBER_NAME
The SSID template variable is substituted with the current Db2 member name to which Db2 HPU has connected.
GROUP_NAME

The SSID template variable is substituted with the Db2 member name in a non data sharing environment and with the group attachment name for data sharing environment.

The default value is TEMPLATE_SSID_SUBSTITUTION (MEMBER_NAME).

CNTLCARDS_DB2(DDL/NO_DDL)

This parameter specifies the behavior of HPU for the UNLOAD PLUS options CNTLCARDS DB2 or CNTLCARDS DB2DDL:

NO_DDL

The above mentioned options are considered as CNTLCARDS DB2LOAD and no CREATE TABLE statement is created and the job ends with return code 4 with the message INZU027W.

DDL

When either of the above mentioned options are specified, Db2 HPU generates both the LOAD statement and the CREATE TABLE statements.

Default value: NO_DDL

BIF_COMPATIBILITY_FROM_DB2_SETTINGS(V11_AND_LATER/NO)

This parameter is optional. It specifies whether the Db2 BIF_COMPATIBILITY settings must be taken into account to determine the behavior of the CHAR and VARCHAR SQL functions evaluated in native mode. Select one of the following allowed values.

NO (default value)

the behavior of the CHAR and VARCHAR functions is driven by the value specified for the Db2 HPU BIF_COMPATIBILITY subparameter from the VUU030/ULOPTNS parameter of its default value (i.e. CURRENT) if it is not specified. Choosing this value can lead to discrepancies between the Db2 HPU behavior in native and SQL modes. Such a discrepancy is reported through the INZU532I message.

V11_AND_LATER

if the Db2 version is V11 NFM or later, the behavior of the CHAR and VARCHAR functions is driven by the value of the Db2 BIF_COMPATIBILITY parameter according to the table below:

<table>
<thead>
<tr>
<th>Db2 BIF_COMPATIBILITY (applies to SQL mode)</th>
<th>CHAR and VARCHAR functions behavior in native mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>V9 or V9_DECIMAL_VARCHAR or V9_TRIM</td>
<td>same as described for Db2 HPU BIF_COMPATIBILITY(V9)</td>
</tr>
<tr>
<td>CURRENT</td>
<td>CURRENT</td>
</tr>
</tbody>
</table>

If the Db2 version is prior to V11 NFM, the same behavior as described for NO applies.

Important: if BIF_COMPATIBILITY_FROM_DB2_SETTINGS(NO) is chosen or if BIF_COMPATIBILITY_FROM_DB2_SETTINGS(V11_AND_LATER) is chosen and Db2 HPU is using a Db2 version prior to V11 NFM, make sure to specify the appropriate value for the BIF_COMPATIBILITY subparameter from VUU030/ULOPTNS.

The default value is BIF_COMPATIBILITY_FROM_DB2_SETTINGS(NO).

BIF_COMPATIBILITY(CURRENT/V9)

This parameter is optional. It applies to the native processing and only applies if one of the following condition if fulfilled:

• the Db2 release is prior to V11
• the BIF_COMPATIBILITY_FROM_DB2_SETTINGS(NO) is specified for VUU030/ULOPTNS.

Otherwise, i.e. if BIF_COMPATIBILITY_FROM_DB2_SETTINGS(V11_AND_LATER) is specified for VUU030/ULOPTNS and the Db2 release is later than V11 NFM, the value specified for this parameter (BIF_COMPATIBILITY from VUU030/ULOPTNS) is ignored and the behavior of the CHAR and VARCHAR functions in native mode is driven by the Db2 BIF_COMPATIBILITY settings. See the description of BIF_COMPATIBILITY_FROM_DB2_SETTINGS(V11_AND_LATER) for more details.
When applicable, this parameter specifies the behavior for the CHAR and VARCHAR SQL functions when evaluated in native mode. It is the counterpart of the Db2 BIF_COMPATIBILITY parameter that only affects the behavior of the CHAR and VARCHAR SQL function in SQL mode.

Specify one of the following allowed values to get the desired CHAR and VARCHAR behavior:

**CURRENT (default value)**

For decimal input, the CHAR and VARCHAR built-in functions return data in the Db2 Version 10 format if the Db2 version is V10 NFM or higher. When this value is specified, the CHAR and VARCHAR functions evaluated in native mode have the same behavior as evaluated by Db2 when the Db2 BIF_COMPATIBILITY is set to CURRENT.

**V9**

For decimal input, the CHAR and VARCHAR built-in functions return data in the Db2 Version 9 format. When this value is specified, the CHAR and VARCHAR functions evaluated in native mode have the same behavior as evaluated by Db2 when the Db2 BIF_COMPATIBILITY is set to V9_DECIMAL_VARCHAR.

**Note:** unlike the Db2 BIF_COMPATIBILITY parameter, this parameter does not affect the behavior of the TRIM function.

When this parameter is applicable, the behaviors in SQL mode and native mode are fully identical for either of the following situations:

- Db2 HPU BIF_COMPATIBILITY specifies CURRENT and Db2 BIF_COMPATIBILITY specifies CURRENT.
- Db2 HPU BIF_COMPATIBILITY specifies V9 and Db2 BIF_COMPATIBILITY specifies V9_DECIMAL_VARCHAR.

**Important:** If the Db2 BIF_COMPATIBILITY setting changes, Db2 HPU must be bound so that the changes are also visible in Db2 HPU.

**FMT_VAR_FB_IF_FIXED**

When specified, the default RECFM for an unload in FORMAT VARIABLE of only fixed length columns is FB instead of VB. If this parameter is not specified, the default RECFM for unload of only fixed length data is VB. This parameter applies only to FORMAT VARIABLE.

The default value is BIF_COMPATIBILITY(CURRENT).

**TAPEUNIT_DEFAULT_DEV_TO_MAX**

This keyword allows to specify how the default_devices_number operand of the TAPEUNIT option or VZM010/TAPEUNIT parmlib parameter is determined when not explicitly specified. In other words, it allows to specify the rule used to determine the default value of default_devices_number operand. By default, i.e. when this keyword is not specified, the default value of 1 is used. Otherwise, when TAPEUNIT_DEFAULT_DEV_TO_MAX is specified, the default value is the same value as specified for the maximum_number_of_devices. In a nutshell, if TAPEUNIT(unit(maxdev) is specified, Db2 HPU behaves as if:

- TAPEUNIT(unit(maxdev,1) was specified if TAPEUNIT_DEFAULT_DEV_TO_MAX is not specified
- TAPEUNIT(unit(maxdev,maxdev) was specified if TAPEUNIT_DEFAULT_DEV_TO_MAX is specified

**TAPELBII(INPUT|NOSPANNED|ALL)**

Specifies in which cases Db2 HPU handles large block when a data set located on tape and allocated with BLKSZLIM is involved.

**INPUT**
Db2 HPU only supports large block for input data sets

**NOSPANNED**
Db2 HPU writes large blocks into output data sets except for SPANNED files

**ALL**
Db2 HPU supports large block for both input and output data sets even for spanned output data sets
The default value is NOSPANNED.

**TEMPLATE_LIMIT(NO)**
The keyword LIMIT of the TEMPLATE statement is ignored but a warning message is issued to inform the user that the keyword was ignored.

**TEMPLATE_LIMIT(YES)**
The keyword LIMIT of the TEMPLATE statement is supported. See TEMPLATE block syntax and description section for details about this feature.

**SEARCH_UNCATALOGED_IC(YES|NO)**
This sub-parameter specifies whether the uncatalogued image copies are eligible when an unload from image copy is requested through COPYDDN LAST_IC or COPYDDN -integer options.

- **SEARCH_UNCATALOGED_IC"YES"**: both the cataloged and non cataloged images copies referenced in the Db2 catalog (in the SYSIBM.SYSCOPY table) are considered.
- **SEARCH_UNCATALOGED_IC"NO"**: only the cataloged images copies referenced in the Db2 catalog (in the SYSIBM.SYSCOPY table) are considered.

The default value is SEARCH_UNCATALOGED_IC(NO).

**ORDER_CLUSTER_BHV (OPTIONAL|MANDATORY)**
Specify whether the 'ORDER CLUSTER' clause, when specified, is considered, by default, as a mandatory or an optional request. When an ORDER CLUSTER clause is considered as optional, Db2 HPU ignores it when it is specified for a logical unload processed in SQL mode.

- **OPTIONAL**: the 'ORDER CLUSTER' clause is considered as optional, thus ignored if the unload is processed in SQL mode. Select this value if your usually specify an ORDER CLUSTER clause for performance purpose to prepare the data for sub-sequence reload operation but none of the subsequent programs that use the unloaded data require the data be ordered according to the order cluster key.
- **MANDATORY**: the 'ORDER CLUSTER' clause is considered as mandatory. As a consequence, any unload to be run in SQL mode that specifies an incompatible ORDER CLUSTER clause stops reporting an error. Select this value you usually specify the ORDER CLUSTER clause to get data unloaded ordered according to the order cluster key and that order as required by the sub-sequence programs that take this data as input.

The default value is MANDATORY.

**USER_SWITCH_METHOD(AS|RACF)**
Db2 HPU does not perform every action on behalf of the same userid but may switch between userids from the list below:

- the submitter of the job that contains the Db2 HPU step
- the User who execute DB2 processes on the table space (VUM031/QUITUSR)
- the User who issues SELECT by using dynamic SQL (VUM032/ACLTGUSR)
- the User who issues EXPLAIN PLAN statements (VUM035/EXPLNUSR)
- the User who runs the DISPLAY command (VUM028/DISPLUSR)

Two methods can be used to switch between the userids. This parameter allows you to specify which method to use.

**AS**
The new userid is set at the address space level. This gives the unload job the same privileges as the privileges directly given to the userid (primary authorizations) but none of the privileges that the given userid might inherit from secondary authorizations. This method is suitable if all the specific userid specified by the parmlib parameters mentioned in the above list explicitly have a privilege required for the related operation.

**RACF**
RACF is invoked to switch to the new userid. This gives the unload job the same privileges as the privileges directly given to the userid (primary authorizations) and the privileges that the given userid might inherit from secondary authorizations. This method is suitable if at least
one of the userid specified by the parmlib parameters mentioned in the above list inherits the privilege required for the related operation via secondary authorizations.

**Note:** the following situation is not supported if USER_SWITCH_METHOD is set to RACF.

- the SYSPRINT dd statement is specified more than once and the first card allocates it as a non-SYSOUT data set
- the QUIESCE of the table to unload is requested
- the VUM031/QUITUSUSR parameter is not set to CURRENT_USER

When such an unlikely situation occurs, Db2 fails to QUIESCE the table space.

The default value is USER_SWITCH_METHOD(AS).

**USE_ADMIN_INFO_SYSPARM (YES|NO)**

This subparameters allows Db2 HPU to use ADMIN_INFO_SYSPARM stored procedure to retrieve DSNZPARM parameters. It concerns BIF_COMPATIBILITY and TEMPLATE_TIME parameters. Enabling this feature requires the stored procedure ADMIN_INFO_SYSPARM to be correctly installed and usage of the procedure also requires specifics privileges.

- The owner of the Db2 HPU plan (specified through VUM012/PLANOWN parameter) must have one of the following privileges on each package that the stored procedure uses:
  - The EXECUTE privilege on the package for DSNADMIZ
  - Ownership of the package
  - PACKADM authority for the package collection
  - SYSADM authority

- The user who execute Db2 HPU must have MONITOR1 privilege.

**YES**

Enable usage of ADMIN_INFO_SYSPARM stored procedure.

**NO**

Enable usage of ADMIN_INFO_SYSPARM stored procedure.

**Important:** Impact in the evaluation of date/time variables for data set names generated for TEMPLATE statements: If USE_ADMIN_INFO_SYSPARM(YES) is specified for VUU030/ULOPTNS, the default value for template time is the value of TEMPLATE_TIME parameter in the Db2 subsystem. If USE_ADMIN_INFO_SYSPARM(NO) is specified for VUU030/ULOPTNS, the default value for template time is set to UTC.

The default behavior of Db2 HPU is USE_ADMIN_INFO_SYSPARM(NO).

Here is an example that shows how to specify a very long value for ULOPTNS:

**MAXERR_STRICT(YES/NO)**

The MAXERR feature (requested either by the MAXERR option or the VUU073/MAXERR parmlib parameter) allows to save up systems resources such as CPU and JES spool spaces by stopping the unload when too many data related errors - in other words, errors affecting a specific row - are encountered. The specifications of this feature that apply in Db2 HPU up to version 4.3 prevent to take benefit from the performance enhancements brought by the Db2 HPU internal architecture introduced by Db2 HPU 5.1. The MAXERR feature has be fitted with a new behavior that allows to exploit the current Db2 HPU version optimized performance. The *MAXERR_STRICT* subparameter allows to specify whether the genuine MAXERR feature behavior must be kept.

**NO**

specify NO to allow the genuine MAXERR feature behavior to be modified in order to take benefit from the best Db2 HPU performance. When this value is specified, the modified MAXERR feature differs from the genuine one as follows: if the last row processed by Db2 HPU is affected by a row level error, the unload completes with a completion code of 8 and the bottom row of the unloaded data contains this invalid record.

**YES**

specify YES to keep the genuine MAXERR feature only if you have the following requirements:
• completion code: the unload completes with completion code of 4 when at least one error at row level has been encountered.

• unloaded data: even though errors at the row level occur, the dataset containing the unloaded rows contain only valid records.

The default value is NO.

Here is an example that shows how to specify a very long value for ULOPTNS: >>>>>>> de99468... [HPUZ-3140]

This parameter is also referred to as VUU030/ULOPTNS.

Check image copy before unloading (VUU033/ULCHKCPY)

This parameter is optional. It specifies whether the dsname that was specified by the COPYDDN parameter in the SYSIN should be checked.

Specify one of the following values:

YES

The dsname that was provided by the COPYDDN parameter in the SYSIN is checked. This PARMILIB parameter is considered if CHECK or INLINE is not specified in the SYSIN. The return code is 4, with a warning message in the SYSOUT if the check against the SYSIBM.SYSCOPY failed, and the FIC corresponding to the dsname is considered as a non-inline FIC.

NO

The dsname is not checked.

The default value is NO.

This parameter is also referred to as VUU033/ULCHKCPY.

Option to modify the like behavior in SQL statement (VUU034/ULLIKE)

This parameter is optional. It specifies how Db2 HPU handles UNICODE MIXED strings in a LIKE predicate.

Specify one of the following values:

STRICT

Indicates that SQL rules are used to handle UNICODE mixed strings. To handle UNICODE mixed strings by using SQL rules, Db2 HPU must convert all operands (columns and masks of the LIKE predicate) in UNICODE DBCS, which increases CPU consumption due to conversions for each unloaded row.
**FAST**
Indicates that UNICODE mixed strings are considered as SBCS strings in LIKE predicates. The FAST option avoids MIXED to DBCS conversions and improves UNLOAD performances.

The default value is FAST.

This parameter is also referred to as VUU034/ULLIKE.

**Db2 HPU response when a LOADDDN cannot be generated (VUU038/UNLLDER)**
This parameter is optional. It specifies whether Db2 HPU will stop when it is unable to generate a requested LOADDDN. This situation can occur when the output format that was specified for DATE, TIME, or TIMESTAMP is not supported by the Db2 LOAD utility.

Specify one of the following values:

**STOP**
Db2 HPU stops with return code 8.

**IGNORE**
Db2 HPU issues a warning message and continues processing.

The default value is STOP.

This parameter is also referred to as VUU039/UNLLDER.

**Unload HIDDEN Column by select * (VUU042/ULHIDDEN)**
This parameter is optional. It specifies whether hidden columns will be selected when a SELECT * statement from a table name or from a LISTDEF is used. This parameter applies only to SELECT statements that are processed natively. Use the HIDDEN parameter of the EXEC card or the HIDDEN option of the OPTIONS block to override this parameter.

Specify one of the following values:

**NO**
Specifies that hidden columns will not be unloaded when a SELECT * statement is used.

**YES**
Specifies that hidden columns will be unloaded when a SELECT * statement is used. For SELECT statements that are processed by Db2, the HIDDEN option is ignored.

The default value is NO.

This parameter is also referred to as VUU042/ULHIDDEN.

**Unload controlled access table in native mode (VUU062/ULACCTRL)**
This parameter is optional. Use it to unload data from tables whose access is controlled at the row level or the column level. This parameter applies only when the data of the table is accessed in native mode, which means a physical unload or a logical unload with DB2 NO specified and a supported SQL statement. To unload a row-controlled table, specify DB2 FORCE or specify DB2 NO, and set the VUU062 parameter to YES.

**YES**
Tables that are controlled at the row or column level can be unloaded in native mode.

**NO**
Tables that are controlled at the row or column level cannot be unloaded.

The default value is NO.

This parameter is also referred to as VUU062/ULACCTRL.

**Technical parameter to unload XML columns (VUU063/ULOPTLX)**
This parameter is optional. Use this parameter to tune the low-level technical algorithm that Db2 HPU uses to unload XML data. If the default optimized values do not apply to your specific case, you can use the VUU063 parameter to enhance performance.

**Important:** Do not specify a value for this parameter unless IBM Software Support recommends a specific value.
This parameter does not have a default value.

An equivalent SYSIN keyword for this parameter does not exist.

This parameter is also referred to as VUU063/ULOPTLX.

**Parameter to identify ROW-TRANSACTION-TIMESTAMP (VUU065/ULRTTST)**

This parameter is optional. It specifies the technique to be used when generating the load SYSIN to identify the columns that are defined as ROW-TRANSACTION-TIMESTAMP.

This parameter applies only to natively processed SELECT statements.

Unsupported SELECT statements and SELECT statements that are processed with the DB2 FORCE option are ignored. In this case, the INGOREFIELDS and PERIODOVERRIDE keywords are not generated.

**IGNORE**

The load generates the column data. The generated load control statement contains the combination of IGNOREFIELDS keyword and a dummy field name for the identify column.

**IGNOREOVERRIDE**

The unloaded data is loaded into the identify column. The PERIODOVERRIDE keyword is added to the load control statement, and the exact name of the identify column is kept. This option is available for Db2 Version 9 and later releases. For earlier version of Db2, the ignore version is used.

**IGNORE** the default value.

This parameter is also referred to as VUU065/ULRTTST.

**Parameter to identify ROW-TRANSACTION-START-ID (VUU066/ULRTSID)**

This parameter is optional. It specifies the technique to be used when generating the load SYSIN to identify the columns that are defined as ROW-TRANSACTION-START-ID.

This parameter applies only to natively processed SELECT statements.

Unsupported SELECT statements and SELECT statements that are processed with the DB2 FORCE option are ignored. In this case, the INGOREFIELDS and PERIODOVERRIDE keywords are not generated.

**IGNORE**

The load generates the column data. The generated load control statement contains the combination of IGNOREFIELDS keyword and a dummy field name for the identify column.

**TRANSIDOVERRIDE**

The unloaded data is loaded into the identify column. The TRANSIDOVERRIDE keyword is added to the load control statement, and the exact name of the identify column is kept. This option is available for Db2 Version 9 and later releases. For earlier version of Db2, the ignore version is used.

**IGNORE** the default value.

This parameter is also referred to as VUU066/ULRTSID.

**Option to specify conversion truncation policy (VUU067/ULCNVTRC)**

This parameter is optional. It allows you to specify what policy to apply when data truncation is needed because of a CCSID conversion. The value of ULCNVTRC is overridden by the CONVERSION_TRUNCATION_ALLOWED sysin option (see OPTIONS block). Allowed values: YES/NO

**YES**

Truncation of the output field after a CCSID conversion is allowed.

**NO**

Truncation of the output field after a CCSID conversion is forbidden. Records requiring truncation are discarded and the return code is set to 4.

The default value is YES.

This parameter is also referred to as VUU067/ULCNVTRC.
Expansion ratio for conversion to UTF8 CCSID (VUU068/ULUNIEXP)

This parameter is optional. It allows you to specify what expansion ratio (i.e. target data length divided by source data length) to apply to char/varchar data when a conversion from a non-UTF8 CCSID to a UTF8 CCSID is done. The value of ULUNIEXP is overridden by the\nUNICODE_EXPANSION_RATIO sysin option (see OPTIONS block).

Allowed values: integer from 1 to 4
Default value: 1

This parameter is also referred to as VUU068/ULUNIEXP.

Technique to lock the unloaded tablespace (VUU074/ULLOCKTE)

This parameter is optional. It specifies the technique used to lock the unloaded tablespace when LOCK(YES) is requested for a logical unload processed in native mode.

**LOCK**

A SQL LOCK statement is processed on the tables to be unloaded (only on partitions to be unloaded for partitioned tablespace). The lock is released at the end of the UNLOAD process.

**START_RO**

The following steps are performed:

- a SQL LOCK statement is processed on the tables to be unloaded (only on partitions to be unloaded for partitioned tablespace)
- a QUIESCE of the table space is performed if QUIESCE YES applies
- the table space is started in READ ONLY mode (only partitions to be unloaded for partitioned table space)
- locks are released by issuing a COMMIT
- the table space is started in its initial status (RW UT) when the released datasets have been read (for partitioned table space, a partition is started in its initial status when its related dataset has been read)

**Important:** this LOCK technique is not honored and the SQL LOCK technique is used instead, if your Db2 HPU step involves a logical unload processed in SQL mode. See the explanation of the “Eligible processing mode(s) for locking data feature (VUU088/ULLOCKMO)” parmlib parameter for more explanation.

**Note:** if you specify LOCK YES QUIESCE YES and the parameter VUU074/ULLOCKTE = START_RO, the following steps are processed:

- LOCK on the table of the table space to be unloaded
- QUIESCE of the table space
- START RO of the table space (or the unloaded partition of the table space)
- LOCK removed by processing an SQL COMMIT
- START in its original status (RW or UT) when the read process ends. For partitioned table space, each time a partition dataset is read, the partition is started in its original status (RW or UT).

The default value is LOCK.

This parameter is also referred to as VUU074/ULLOCKTE.

Image copies selected for partitioned table spaces (VUU076/ULICTYPE)

This parameter is optional. It specifies the criteria to determine the image copies to be unloaded when COPYDDN LAST_IC or COPYDDN integer is requested for a partitioned table space and none of the GLOBAL, PARTITIONED, ANYTYPE or CONSISTENT sub-options is specified.

Specify one of the following values:

**GLOBAL**

The LAST_IC processing looks for a global FIC of the complete table space.
PARTITIONED
The LAST_IC processing looks only for FICs that were created for each partition.

ANYTYPE
The LAST_IC processing looks for the most recent global FIC or FIC per partition. If the ANYTYPE keyword is coded without partitions, Db2 HPU looks for the latest FIC for each partition and for the latest global FIC. If all FICs for each partition are more recent than the global FIC, the more recent FICs are used. Otherwise, the global FIC is used. The ANYTYPE value applies only to the COPYDDN LAST_IC behavior (not for COPYDDN integer).

PARTITIONED_CONSISTENT
The LAST_IC processing uses the same FIC selection criterion as for PARTITIONED and also makes sure the image copies of the involved partitions are consistent by checking these copies have the same START_RBA value in the SYSIBM.SYSCOPY table.

ANYTYPE_CONSISTENT
The LAST_IC processing uses the same FIC selection criterion as for ANYTYPE and also makes sure the image copies of the involved partitions are consistent by checking these copies have the same START_RBA value in the SYSIBM.SYSCOPY table.

The default value is the value of VUU076/ULICTYPE.

The corresponding SYSIN options are the OPTIONS COPYDDN sub options.

This parameter is also referred to as VUU076/ULICTYPE.

Oldest year assumed for DECIMAL to DATE conversions when no explicit century given by the DECIMAL source data (VUU077/ULCENT)

This parameter specifies the first year (from 1 to 9999) of a 100-year period that allows to determine the century of a DATE resulting from a conversion of DECIMAL(6,0) data that do not provide an explicit value of the century. The dates are considered to range from the value of this parameter to 99 years later. In other words, if this parameter value expressed as a 4-digit value CCYY where CC are called the century digits and YY the year digits, any date resulting from a conversion of a DECIMAL(6,0) that has no century digits is added century digits as follows:

• If the year digits from the DECIMAL(6,0) data are greater or equal to YY, the CC century digits are added to build the full date

• If the year digits from the DECIMAL(6,0) data are less than YY, the CC + 1 century digits are added to build the full date.

This parameter can also be set to 0, in which case DB2 HPU is not allowed to guess the century digits when not available in the source decimal data thus prohibiting any DECIMAL(6,0) to DATE conversions.

The default value is 0.

The corresponding SYSIN parameter is CENTURY.

This parameter is also referred to as VUU077/ULCENT.

Eligible processing mode(s) for locking data feature (VUU088/ULLOCKMO)

This parameter is optional. It specifies which processing mode(s) is eligible for the LOCK feature (LOCK YES option or by setting the VUU012/ULLOCK parameter to YES).

Specify one of the following values:

ALL
The LOCK feature is enabled for any SELECT statement regardless of the processing mode (native or SQL).

Relationship with other parameters: when the LOCK feature is requested for a SELECT statement processed in SQL mode, Db2 HPU needs to determine the list of the tables involved in the SELECT statement. This determination is done by exploiting the result of an EXPLAIN on the SELECT statement which execution requires that the VUM035/EXPLNUSR parmlib parameter is set and the PLAN_TABLE and DSN_PGRANGE_TABLE tables belonging to the userid specified by the
VUM035/EXPLNUSR exist. The VUM034/EXPLNTB parmlib parameter is also taken into account to keep or delete the rows from the PLAN_TABLE and DSN_PGRANGE_TABLE related to the EXPLAIN performed to determine the tables list. The LOCK of the involved tables is done through an SQL LOCK in share mode against the tables and partitions involved in the unload regardless of the value specified for VUU074/ULLOCKTE. As Db2 HPU allows only one LOCK technique per Db2 HPU invocation (i.e. JCL step), if the LOCK feature is requested for a SELECT statement processed in SQL mode, which requires the SQL LOCK technique, the LOCK feature for SELECT statements processed in native mode are also performed through an SQL LOCK.

**NATIVE_ONLY**

The LOCK feature is enabled only in native mode and ignored in SQL mode.

**Related reference**

User-allocated ddnames

To run unload jobs, you must allocate certain ddnames in the Db2 HPU JCL.

### Db2 HPU output data parameters

The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

The Db2 HPU parameters are designated in three ways:

- a long designation that describes what the parameter is used for
- two short names used as a reference especially within the messages texts:
  - one has the following format Vxxnnnn where xx are characters and nnnn are figures
  - the other one is a short label of 8 characters

You can accept the default values for the Db2 HPU output data parameters, or you can configure them based on your specific requirements.

The following list describes the Db2 HPU parameters. The parameters are listed in the following format:

```
description (Vxxnnnn/parameter-name)
```

#### Override DCB Db2 HPU parameters (VUM029/DFSIGDCB)

This parameter is optional. It specifies whether the DCB JCL parameter can be overridden in Db2 HPU syntax.

Specify one of the following values:

- **YES**
  - For jobs that use Db2 HPU syntax, Db2 HPU ignores the DCB parameter in the JCL and sets the DCB attributes to the appropriate values.

- **NO**
  - For jobs that use Db2 HPU syntax, Db2 HPU uses the DCB parameter in the JCL.

The default value is NO.

The corresponding SYSIN keyword is DFSIGDCB in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUM029/DFSIGDCB.

#### Override DCB UNLOAD PLUS syntax (VUM029/DFSIGDCB)

This parameter is optional. It specifies whether the DCB JCL parameter can be overridden in UNLOAD PLUS syntax.

Specify one of the following values:

- **YES**
  - For jobs that use UNLOAD PLUS syntax, Db2 HPU ignores the DCB parameter in the JCL and sets the DCB attributes to the appropriate values.
NO
For jobs that use UNLOAD PLUS syntax, Db2 HPU uses the DCB parameter in the JCL.
The default value is NO.
The corresponding SYSIN keyword is DFSIGDCB in the Technical Parameters options block, which can
be coded in the GLOBAL OPTIONS block.
This parameter is also referred to as VUM029/DFSIGDCB.

Override DCB FAST UNLOAD syntax (VUM029/DFSIGDCB)
This parameter is optional. It specifies whether the DCB JCL parameter can be overridden in Fast
Unload syntax.
Specify one of the following values:

YES
For jobs that use Fast Unload syntax, Db2 HPU ignores the DCB parameter in the JCL and sets the
DCB attributes to the appropriate values.

NO
For jobs that use Fast Unload syntax, Db2 HPU uses the DCB parameter in the JCL.
The default value is NO.
The corresponding SYSIN keyword is DFSIGDCB in the Technical Parameters options block, which can
be coded in the GLOBAL OPTIONS block.
This parameter is also referred to as VUM029/DFSIGDCB.

NULL indicator, format USER (VUU014/ULNULL)
This parameter is optional. Use this parameter to modify the null or not-null indicator.

OFF
The null indicator is not present in the output data set.

hhhh
The first two digits (one hexadecimal character) represent the null indicator for a null column. The
last two digits (one hexadecimal character) represent the null indicator for a not-null column.
The default value is FF00.
The corresponding SYSIN parameter is OPTIONS NULL.
This parameter is also referred to as VUU014/ULNULL.

NULL indicator, format DSNTIAUL (VUU014/ULNULL)
This parameter is optional. Use this parameter to modify the null or not-null indicator.

OFF
The null indicator is not present in the output data set.

hhhh
The first two digits (one hexadecimal character) represent the null indicator for a null column. The
last two digits (one hexadecimal character) represent the null indicator for a not-null column.
The default value is FF00.
The corresponding SYSIN parameter is OPTIONS NULL.
This parameter is also referred to as VUU014/ULNULL.

NULL indicator, format VARIABLE (VUU014/ULNULL)
This parameter is optional. Use this parameter to modify the null or not-null indicator.

OFF
The null indicator is not present in the output data set.

hhhhh
The first two digits (one hexadecimal character) represent the null indicator for a null column. The
last two digits (one hexadecimal character) represent the null indicator for a not-null column.
The default value is FF00.
The corresponding SYSIN parameter is OPTIONS NULL.
This parameter is also referred to as VUU014/ULNULL.

**NULL indicator, format EXTERNAL (VUU014/ULNULL)**
This parameter is optional. Use this parameter to modify the null or not-null indicator.

**OFF**
The null indicator is not present in the output data set.

**hhhh**
The first two digits (one hexadecimal character) represent the null indicator for a null column. The last two digits (one hexadecimal character) represent the null indicator for a not-null column.
The default value is FF00.
The corresponding SYSIN parameter is OPTIONS NULL.
This parameter is also referred to as VUU014/ULNULL.

**DATE, format USER (VUU015/ULDATE)**
This parameter is optional. It specifies the default conversion type for a date column.
The default value is DATE_C.
The corresponding SYSIN parameter is OPTIONS DATE.
This parameter is also referred to as VUU015/ULDATE.

**DATE, format DSNTIAUL (VUU015/ULDATE)**
This parameter is optional. It specifies the default conversion type for a date column.
The default value is DATE_DB2.
The corresponding SYSIN parameter is OPTIONS DATE.
This parameter is also referred to as VUU015/ULDATE.

**DATE, format DELIMITED (VUU015/ULDATE)**
This parameter is optional. It specifies the default conversion type for a date column.
The default value is DATE_C.
The corresponding SYSIN parameter is OPTIONS DATE.
This parameter is also referred to as VUU015/ULDATE.

**DATE, format VARIABLE (VUU015/ULDATE)**
This parameter is optional. It specifies the default conversion type for a date column.
The default value is DATE_C.
The corresponding SYSIN parameter is OPTIONS DATE.
This parameter is also referred to as VUU015/ULDATE.

**DATE, format EXTERNAL (VUU015/ULDATE)**
This parameter is optional. It specifies the default conversion type for a date column.
The default value is DATE_DB2.
The corresponding SYSIN parameter is OPTIONS DATE.
This parameter is also referred to as VUU015/ULDATE.

**TIME, format USER (VUU016/ULTIME)**
This parameter is optional. It specifies the default conversion type for a time column.
The default value is TIME_A.
The corresponding SYSIN parameter is OPTIONS TIME.
This parameter is also referred to as VUU016/ULTIME.

**TIME, format DSNTIAUL (VUU016/ULTIME)**
This parameter is optional. It specifies the default conversion type for a time column.
TIME, format DELIMITED (VUU016/ULTIME)
This parameter is optional. It specifies the default conversion type for a time column.
The default value is TIME_A.
The corresponding SYSIN parameter is OPTIONS TIME.
This parameter is also referred to as VUU016/ULTIME.

TIME, format VARIABLE (VUU016/ULTIME)
This parameter is optional. It specifies the default conversion type for a time column.
The default value is TIME_A.
The corresponding SYSIN parameter is OPTIONS TIME.
This parameter is also referred to as VUU016/ULTIME.

TIME, format EXTERNAL (VUU016/ULTIME)
This parameter is optional. It specifies the default conversion type for a time column.
The default value is TIME_DB2.
The corresponding SYSIN parameter is OPTIONS TIME.
This parameter is also referred to as VUU016/ULTIME.

TIMESTAMP, format USER (VUU017/ULTMSTP)
This parameter is optional. It specifies the default conversion type for a timestamp column.
The default value is TMSTP_B
The corresponding SYSIN parameter is OPTIONS TIMESTAMP.
This parameter is also referred to as VUU017/ULTMSTP.

TIMESTAMP, format DSNTIAUL (VUU017/ULTMSTP)
This parameter is optional. It specifies the default conversion type for a timestamp column.
The default value is TMSTP_B
The corresponding SYSIN parameter is OPTIONS TIMESTAMP.
This parameter is also referred to as VUU017/ULTMSTP.

TIMESTAMP, format DELIMITED (VUU017/ULTMSTP)
This parameter is optional. It specifies the default conversion type for a timestamp column.
The default value is TMSTP_B
The corresponding SYSIN parameter is OPTIONS TIMESTAMP.
This parameter is also referred to as VUU017/ULTMSTP.

TIMESTAMP, format VARIABLE (VUU017/ULTMSTP)
This parameter is optional. It specifies the default conversion type for a timestamp column.
The default value is TMSTP_B
The corresponding SYSIN parameter is OPTIONS TIMESTAMP.
This parameter is also referred to as VUU017/ULTMSTP.

TIMESTAMP, format EXTERNAL (VUU017/ULTMSTP)
This parameter is optional. It specifies the default conversion type for a timestamp column.
The default value is TMSTP_B
The corresponding SYSIN parameter is OPTIONS TIMESTAMP.
This parameter is also referred to as VUU017/ULTMSTP.

Decimal picture, format USER (VUU018/ULPIC)
This parameter is optional. It specifies the numeric data display format.
Specify a value in the following format: `signpositionseparator`.

**sign**

Specify one of the following values to print the sign:

- `+`
  
  The plus sign (+) is used for positive values.

- `-`
  
  The minus sign (-) is used for negative values.

- `P`
  
  The padding character is used for positive values, and the minus sign (-) is used for negative values.

- `V`
  
  The sign byte used depends on the data type as follows:
  - for non-float numeric data:
    - no sign byte is written for values greater than or equal to zero.
    - the minus sign (-) is written for values less than zero.
  - for float data:
    - the plus sign (+) is written for values greater than or equal to zero.
    - the minus sign (-) is written for values less than zero.

The default value for specifying the rules for printing is the minus sign.

**position**

Specify one of the following values to position the sign:

- `LEAD`
  
  The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

- `TRAIL`
  
  The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

**separator**

Specify one of the following values for the decimal separator:

- `.`
  
  Use a period (.) as the decimal separator.

- `,`
  
  Use a comma (,) as the decimal separator.

The default value `-LEAD.,` which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU018/ULPIC.

**Decimal picture, format DSNTIAUL (VUU018/ULPIC)**

This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: `signpositionseparator`.

**sign**

Specify one of the following values to print the sign:

- `+`
  
  The plus sign (+) is used for positive values.
The minus sign (-) is used for negative values.

P
The padding character is used for positive values, and the minus sign (-) is used for negative values.

V
The sign byte used depends on the data type as follows:

• for non-float numeric data:
  – no sign byte is written for values greater than or equal to zero.
  – the minus sign (-) is written for values less than zero.
• for float data:
  – the plus sign (+) is written for values greater than or equal to zero.
  – the minus sign (-) is written for values less than zero.

The default value for specifying the rules for printing is the minus sign.

position
Specify one of the following values to position the sign:

LEAD
The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

TRAIL
The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

separator
Specify one of the following values for the decimal separator:

. Use a period (.) as the decimal separator.

, Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.
This parameter is also referred to as VUU018/ULPIC.

Decimal picture, format DELIMITED (VUU018/ULPIC)
This parameter is optional. It specifies the numeric data display format.
Specify a value in the following format: signpositionseparator.

sign
Specify one of the following values to print the sign:

+ The plus sign (+) is used for positive values.

- The minus sign (-) is used for negative values.

P The padding character is used for positive values, and the minus sign (-) is used for negative values.
The sign byte used depends on the data type as follows:

- for non-float numeric data:
  - no sign byte is written for values greater than or equal to zero.
  - the minus sign (-) is written for values less than zero.

- for float data:
  - the plus sign (+) is written for values greater than or equal to zero.
  - the minus sign (-) is written for values less than zero.

The default value for specifying the rules for printing is the minus sign.

**position**

Specify one of the following values to position the sign:

**LEAD**

The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

**TRAIL**

The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

**separator**

Specify one of the following values for the decimal separator:

- Use a period (.) as the decimal separator.
- Use a comma (,) as the decimal separator.

The default value -LEAD,, which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU018/ULPIC.

**Decimal picture, format VARIABLE (VUU018/ULPIC)**

This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: `signpositionseparator`.

**sign**

Specify one of the following values to print the sign:

- The plus sign (+) is used for positive values.
- The minus sign (-) is used for negative values.

**V**

The sign byte used depends on the data type as follows:

- for non-float numeric data:
  - no sign byte is written for values greater than or equal to zero.
  - the minus sign (-) is written for values less than zero.
• for float data:
  – the plus sign (+) is written for values greater than or equal to zero.
  – the minus sign (-) is written for values less than zero.

The default value for specifying the rules for printing is the minus sign.

**position**
Specify one of the following values to position the sign:

**LEAD**
The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

**TRAIL**
The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

**separator**
Specify one of the following values for the decimal separator:

• .
  Use a period (.) as the decimal separator.

• ,
  Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU018/ULPIC.

**Decimal picture, format EXTERNAL (VUU018/ULPIC)**
This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: *signpositionseparator*.

**sign**
Specify one of the following values to print the sign:

• +
  The plus sign (+) is used for positive values.

• -
  The minus sign (-) is used for negative values.

• P
  The padding character is used for positive values, and the minus sign (-) is used for negative values.

• V
  The sign byte used depends on the data type as follows:
  • for non-float numeric data:
    – no sign byte is written for values greater than or equal to zero.
    – the minus sign (-) is written for values less than zero.
  • for float data:
    – the plus sign (+) is written for values greater than or equal to zero.
    – the minus sign (-) is written for values less than zero.

The default value for specifying the rules for printing is the minus sign.
**position**

Specify one of the following values to position the sign:

**LEAD**

The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

**TRAIL**

The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

**separator**

Specify one of the following values for the decimal separator:

- Use a period (.) as the decimal separator.
- Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU018/ULPIC.

**Display format for numeric values, format USER (VUU032/ULMASK)**

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with non significant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

**Important:**

1. The ULMASK variable has no impact on the formatting of FLOAT values.

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>.</em></td>
<td>No</td>
<td>Only when the value is 0</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>0.*</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>*.0</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>0.0</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 32. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.0</td>
<td>Yes</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>*.Z</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>0.Z</td>
<td>No</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>00.Z</td>
<td>Yes</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 143.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is `*.*`

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU032/ULMASK.

**Display format for numeric values, format DSNTIAUL (VUU032/ULMASK)**

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with non-sig
  nificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when
  the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal
  separator column applies only to decimal data. The decimal separator is never displayed for
  SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal
  separator with zeroes, up to the number of digits of the decimal scale.

**Important:**

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 33. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>.</em></td>
<td>No</td>
<td>Only when the value is 0</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>0.*</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>*.0</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>0.0</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 33. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.0</td>
<td>Yes</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>*.Z</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>0.Z</td>
<td>No</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>00.Z</td>
<td>Yes</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 143.

The following example shows how to specify a ULMASK value for the USER format:

00.0

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is **.***

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU032/ULMASK.

**Display format for numeric values, format DELIMITED (VUU032/ULMASK)**

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with non significant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

**Important:**

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 34. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>.</strong>*</td>
<td>No</td>
<td>Only when the value is 0</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>0.*</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>*<strong>.0</strong></td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>0.0</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 34. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.0</td>
<td>Yes</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>*.Z</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>0.Z</td>
<td>No</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>00.Z</td>
<td>Yes</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 143.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is **.*.**

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU032/ULMASK.

**Display format for numeric values, format VARIABLE**

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with nonsignificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

**Important:**

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 35. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>.</strong>*</td>
<td>No</td>
<td>Only when the value is 0</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>0.*</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>*.0</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>0.0</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 35. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.0</td>
<td>Yes</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>*.Z</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>0.Z</td>
<td>No</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>00.Z</td>
<td>Yes</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 143.

The following example shows how to specify a ULMASK value for the USER format:

```
 00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is *.*

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU032/ULMASK.

Display format for numeric values, format EXTERNAL (VUU032/ULMASK)

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with non significant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

Important:

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 36. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>.</em></td>
<td>No</td>
<td>Only when the value is 0</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>0.*</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal value is not 0</td>
<td>No</td>
</tr>
<tr>
<td>*.0</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>0.0</td>
<td>No</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 36. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

<table>
<thead>
<tr>
<th>ULMASK value</th>
<th>Left padding</th>
<th>Left zero</th>
<th>Decimal separator</th>
<th>Right padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>00.0</td>
<td>Yes</td>
<td>Always</td>
<td>Only if the decimal scale is not 0</td>
<td>Yes</td>
</tr>
<tr>
<td>*.Z</td>
<td>No</td>
<td>Only when the decimal scale is 0</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>0.Z</td>
<td>No</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
<tr>
<td>00.Z</td>
<td>Yes</td>
<td>Always</td>
<td>Always</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 143.

The following example shows how to specify a ULMASK value for the USER format:

00.0

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is **.*.**

The corresponding SYSIN parameter is OPTIONS PIC.

This parameter is also referred to as VUU032/ULMASK.

**Options for global LOAD statement (VUU019/ULOPTLDT)**

This parameter is optional. It defines the parameters of the LOAD statement that were generated at the table space level.

Enclose all sub-parameters in parentheses, as shown in the following example:

`${SORTDEVT(SYSDA) }`  `SORTNUM(32)`

To generate a parenthesis in the LOADDDN, include two parentheses in this parameter, as shown in the following statement:

`(ENFORCE(NO), LOG(NO), COPYDDN((SYSCOPY)))`

This statement generates this option in the LOAD statement:

`ENFORCE(NO) LOG NO COPYDDN(SYSCOPY)`

To code the load options on several lines, follow these rules:

- Use a continuation comma at the end of each line.
- Begin the first string with a left parenthesis, and code the corresponding right parenthesis only on the last line used.

The following parameters are valid. If you code a parameter that is not in this list, it is not controlled, and it is copied in the generated LOAD SYSIN:

- COPYDDN
- DISCARDDN
- DISCARDS
- ENFORCE(CONSTRAINTS/NO)
- INDDN
- KEEPDICTIONARY
- LOG(YES/NO)
- RECOVERYDDN
• REPLACE
• RESUME(YES|NO)
• SORTDEVT
• SORTKEYS
• SORTNUM
• PARALLEL or PARALLEL (num-subtask) where num-subtask specifies the maximum number of sub-tasks that are to be processed in parallel by the LOAD utility. The value must be an integer between 0 and 32767, inclusive.

**Note:** The INDDN keyword is ignored if the LOADINDDN(YES) option applies, either by the parmlib settings or by the LOADINDDN(YES) options from the unload command. If you want the INDDN keyword to be taken into account, do not specify the LOADINDDN(YES) for ULOPTNS or disable it by specifying LOADINDDN(NO) option from the unload command.

**Important:** If you specify the SORTKEYS keyword, the value that you specify is substituted with a value that is calculated according to the number of unloaded records, except when you run Db2 HPU in EXECUTE NO mode. When you run Db2 HPU in EXECUTE NO mode, the &SORTKEYS variable is replaced with 0.

If you code a parameter that is not in the previous list, it will not be controlled, and will only be copied in the generated LOAD SYSIN.

When the output file is a spanned file, NUMRECS is always generated in the SYSIN for the LOAD utility, therefore the SORTKEYS parameter is not taken into account.

The default value is (LOG (NO), ENFORCE(NO)).

This parameter is also referred to as VUU019/ULOPTLDT.

**Options of partition LOAD statement (VUU020/ULOPTLDP)**
This parameter is optional. Defines the parameters of the LOAD statement that was generated at the partition level.

To code the load options on several lines, follow these rules:
• Use a continuation comma at the end of each line.
• Begin the first string with a left parenthesis, and code the corresponding right parenthesis only on the last line used.

The following parameters are accepted:
• RESUME(YES|NO)
• REPLACE
• KEEPDICIONARY

The default value is (RESUME(YES)).

This parameter is also referred to as VUU020/ULOPTLDP.

**Options of table LOAD statement (VUU070/ULOPTLDB)**
This parameter is optional. Defines the parameters of the LOAD statement to be generated at the table level.

To code the load options on several lines, follow these rules:
• Use a continuation comma at the end of each line.
• Begin the first string with a left parenthesis, and code the corresponding right parenthesis only on the last line used.

The following parameters are accepted:
• NUMREC(&NUMREC): the number of unloaded records is specified as the operand of the NUMRECS options written in the generated LOAD statement
• NUMRECS(integer): the specified integer value is used as the operand of the NUMRECS options written in the generated LOAD statement.

If you code a parameter that is not in the above, it will not be controlled and will only be copied in the generated LOAD SYSIN as specified.

The default value is (NUMRECS(&NUMRECS)).

This parameter is also referred to as VUU070/ULOPTLDB

Position for NULL indicator, format USER (VUU022/NULLPOS)

This parameter is optional. It specifies the position of the NULL indicator within the Db2 HPU output data sets in USER format. This parameter does not apply to a LOB column unloaded in a spanned format (SPANNED YES option) in which case the NULL indicator is written before the output field (containing the data) whatever the value of NULLPOS.

Specify one of the following values:

AFTER
  The NULL indicator is placed after the field containing the related data.

BEFORE
  The NULL indicator is placed before the field containing the related data.

FRONT
  The NULL indicators are gathered at the front of each unloaded record. The sequence of the NULL indicators is the same as the sequence of the related columns.

The default value is BEFORE.

The corresponding SYSIN parameter is OPTIONS NULLPOS.

This parameter is also referred to as VUU022/NULLPOS.

Position for NULL indicator, format DSNTIAUL (VUU022/NULLPOS)

This parameter is optional. It specifies the position of the NULL indicator within the Db2 HPU output data sets in DSNTIAUL format. This parameter does not apply to a LOB column unloaded in a spanned format (SPANNED YES option) in which case the NULL indicator is written before the output field (containing the data) whatever the value of NULLPOS.

Specify one of the following values:

AFTER
  The NULL indicator is placed after the field containing the related data.

BEFORE
  The NULL indicator is placed before the field containing the related data.

FRONT
  The NULL indicators are gathered at the front of each unloaded record. The sequence of the NULL indicators is the same as the sequence of the related columns.

The default value is AFTER.

The corresponding SYSIN parameter is OPTIONS NULLPOS.

This parameter is also referred to as VUU022/NULLPOS.

Position for NULL indicator, format VARIABLE (VUU022/NULLPOS)

This parameter is optional. It specifies the position of the NULL indicator within the Db2 HPU output data sets in VARIABLE format. This parameter does not apply to a LOB column unloaded in a spanned format (SPANNED YES option) in which case the NULL indicator is written before the output field (containing the data) whatever the value of NULLPOS.

Specify one of the following values:

AFTER
  The NULL indicator is placed after the field containing the related data.
BEFORE
The NULL indicator is placed before the field containing the related data.

FRONT
The null indicators are gathered at the front of each unloaded record. The sequence of the NULL indicators is the same as the sequence of the related columns.

The default value is BEFORE.

The corresponding SYSIN parameter is OPTIONS NULLPOS.

This parameter is also referred to as VUU022/NULLPOS.

Position for NULL indicator, format EXTERNAL (VUU022/NULLPOS)
This parameter is optional. It specifies the position of the NULL indicator within the Db2 HPU output data sets in EXTERNAL format. This parameter does not apply to a LOB column unloaded in a spanned format (SPANned YES option) in which case the NULL indicator is written before the output field (containing the data) whatever the value of NULLPOS.

Specify one of the following values:

AFTER
The NULL indicator is placed after the field containing the related data.

BEFORE
The NULL indicator is placed before the field containing the related data.

FRONT
The null indicators are gathered at the front of each unloaded record. The sequence of the NULL indicators is the same as the sequence of the related columns.

The default value is AFTER.

The corresponding SYSIN parameter is OPTIONS NULLPOS.

This parameter is also referred to as VUU022/NULLPOS.

Use the same DDN for UNLDDN, OUTDDN, and LOADDNN (VUU029/UNLDDDN)
This parameter is optional. Use this parameter to prevent the use of the same DDN for UNLDDN, OUTDDN, and LOADDNN.

Specify one of the following values:

YES
If you use the same DDN in a single SELECT statement with the keywords UNLDDN, OUTDDN, and LOADDNN, Db2 HPU issues error INZU124E.

NO
An error is not issued when you use the same DDN in a single SELECT statement with the keywords UNLDDN, OUTDDN, and LOADDNN.

The default value is NO.

This parameter is also referred to as VUU029/UNLDDDN.

DATE/TIME delimiter option (VUU031/DTDELIM)
This parameter is optional. It applies to FORMAT DELIMITED to delimit the column types DATE, TIME, and TIMESTAMP in the output data set.

Specify one of the following values:

YES
Column types DATE, TIME, and TIMESTAMP are delimited by the column delimiter that is specified by the option DELIM literal.

NO
If the NULL DELIM option is used in FORMAT DELIMITED, this parameter will not apply to the DATE, TIME, and TIMESTAMP columns.

The default value is NO.
This parameter is also referred to as VUU031/DTDELIM.

**NULL DATE/TIME delimiter option (VUU043/DTNULDLM)**

This parameter is optional. Specifies that the NULL DELIM option of FORMAT DELIMITED should also be used for DATE, TIME, and TIMESTAMP columns. This parameter is used only when PARMLIB variable VUU031/DTDELIM is set to YES.

**YES**

If the NULL DELIM option is used in FORMAT DELIMITED, it will also apply to the DATE, TIME, and TIMESTAMP columns.

**NO**

If the NULL DELIM option is used in FORMAT DELIMITED, it will not apply to the DATE, TIME, and TIMESTAMP columns.

The default value is NO.

This parameter is also referred to as VUU043/DTNULDLM.

**Technique to generate SYSIN for identity column (VUU058/ULIDENT)**

This parameter is optional. It specifies the technique to be used when generating the load SYSIN for identity columns that are defined as GENERATED ALWAYS.

This parameter applies only to natively processed SELECT statements.

Unsupported SELECT statements and SELECT statements that are processed with the DB2 FORCE option are ignored. In this case, the IGNOREFIELDS and IDENTITYOVERRIDE keywords are not generated.

Specify one of the following values:

**IGNORE**

The load generates the column data. The generated load control statement contains the combination of IGNOREFIELDS keyword and a dummy field name for the identity column.

**OVERRIDE**

The unloaded data is loaded into the identity column. The IDENTITYOVERRIDE keyword is added to the load control statement, and the exact name of the identity column is kept. This option is available for Db2 Version 9 and later releases. For earlier versions of Db2, the IGNORE option is used.

The default value is IGNORE.

This parameter is also referred to as VUU058/ULIDENT.

**Sign for zoned-decimal numeric values (VUU037/ULSIGZ)**

This parameter is optional. It specifies the sign for zoned-decimal values in hexadecimal format. Valid values consist in one or two characters of the following list: A, B, C, D, E, and F. The first character represents the positive sign, the second one the negative sign. Using this parameter helps you create a customized format that meets the requirements of the programs that process the unloaded data.

The default value is CD.

This parameter is also referred to as VUU037/ULSIGZ.

**Positive sign for date-time packed values (VUU038/ULSIGDTP)**

This parameter is optional. It specifies the positive sign for date- and time-packed values. Valid values are A, B, C, D, E, and F. Each value represents the positive sign. Using this parameter helps you create a customized format that meets the requirements of the programs that process the unloaded data.

The default value is F.

This parameter is also referred to as VUU038/ULSIGDTP.

**Truncate variable records exceeding 32756 (VUU040/ULTR32K)**

This parameter is optional. Specifies whether Db2 HPU will truncate the variable records that exceed the physical limit of 32756.
Specify one of the following values:

**YES**
- The output file record format is variable, and the LRECL is greater than 32756 (LRECL > 32756). LRECL is truncated to 32756, and message INZU297I is issued.

**NO**
- The output file record format is fixed, and the LRECL is greater than 32756 (LRECL > 32756). Processing ends, and message INZU298E is issued.

The default value is NO.

This parameter is also referred to as VUU040/ULTR32K.

**Default output format for a logical unload (VUU045/ULFORMAT)**
This parameter is optional. It specifies the value of the output format when the FORMAT parameter is not specified in the SYSIN for a SELECT statement.

This parameter applies only to Db2 HPU syntax.

⚠️ **Attention:** Changing this value in the PARMLIB will affect existing production jobs that do not specify the FORMAT parameter. Use caution when changing the value of this parameter in the PARMLIB.

Specify one of the following values:

**DSNTIAUL**
- Specifies that the default output format for SELECT statements is FORMAT VARIABLE. This format is the default value.

**STRICT**
- Use this value to unload data in the same format that the DSNTIAUL program produces. DSNTIAUL STRICT affects the formatting of constant character strings that are specified in SELECT statements.

- Specifying STRICT in VUU045/ULFORMAT overrides the value of DSNTIAUL STRICT in the PARMLIB variable ULOPTNS. If STRICT is not specified in VUU045/ULFORMAT, the content of the PARMLIB variable ULOPTNS is used.

⚠️ **Important:** FORMAT DSNTIAUL STRICT applies only to Db2 HPU syntax.

**DELIMITED**
- Specifies that the default output format for SELECT statements is FORMAT DELIMITED.

**SEP=val/BLANK**
- Specifies the separator character, *val*, to separate fields in the output data set. *val* can be specified in character (’c’) or hexadecimal (X’hh’) format. Values that are specified in hexadecimal are not converted in CCSID conversion. Values that are specified in character are converted, if necessary, into the output CCSID.

- Use BLANK to specify a space character. BLANK is the default value for SEP.

**DELIM=val/NONE**
- Specifies the delimiter character, *val*, to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set. *val* can be specified in character (’c’) or hexadecimal (X’hh’) format. Values that are specified in hexadecimal are not converted in CCSID conversion. Values that are specified in character are converted, if necessary, into the output CCSID.

- BLANK
  - Specifies a space character.

- NONE
  - Specifies that you do not want to use a delimiter.

The default value for the delimiter is NONE.
**NULLDELIM**
Specifies that null values are not enclosed by the delimiter character that is specified by the val of DELIM.

The DELIM and NULL DELIM keyword can also apply to DATE, TIME, TIMESTAMP format depending on the settings of VUU043/DTNULDLBM.

**VARIABLE**
Specifies that the default output format for SELECT statements is FORMAT VARIABLE.

**ALL**
If the column that was selected last is variable, the output data set is VB, and this last column is written on its effective length. The effective length is the actual length of the data that is contained in a column that has a variable type. Both length bytes precede the column.

**END**
All the variable columns are written by using their actual length.
The default value is END if only VARIABLE is specified.

**EXTERNAL**
Specifies that the default output format for SELECT statements is FORMAT EXTERNAL.

**USER**
Specifies that the default output format for SELECT statements is FORMAT USER.

This parameter is also referred to as VUU045/ULFORMAT.

**Size of the autotag in the output file (VUU046/ULAUTAG)**
This parameter is optional. Use this parameter to specify a number for the value of the autotag that is generated in the output file. It applies only to Db2 HPU syntax, and it applies to all output formats.

Valid values are 0 - 8.
The default value is 0. When the default value is specified, an autotag is not generated.
The corresponding SYSIN parameter is OPTIONS AUTOTAG.

**Attention:** Changing this value in the PARMLIB will affect existing production jobs. Use caution when you change this value.

This parameter is also referred to as VUU046/ULAUTAG.

**Full compatibility of LOAD statement with Db2 LOAD (VUU054/CTRLLIBM)**
This parameter is optional. It specifies whether the LOAD statement that is generated by Db2 HPU must be fully compatible with the Db2 LOAD utility.

Valid values are YES and NO.

When this parameter is set to YES, additional controls are made when the DELIMITED format is requested, and the LOAD statement is generated only if it is compatible with the Db2 LOAD utility.

This parameter is also referred to as VUU054/CTRLLIBM.

**Delimiter for external date, format USER (VU055/DATEDEL)**
This parameter is optional. It specifies the default delimiter that is used in external date representations.

Specify one of the following values:

`'c'`
The specified value must be in character format, coded between single quotation marks, and one byte long.

**BLANK**
Specifies that a space character is used as the default delimiter for external date representation. By default, this variable applies to FORMAT USER. It can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.
The default value is '-'.
The corresponding SYSIN parameter is OPTIONS DATEDELIM.
This parameter is also referred to as VUU055/DATEDEL.

**Delimiter for external date, format DSNTIAUL (VU055/DATEDEL)**
This parameter is optional. It specifies the default delimiter that is used in external date representations.

Specify one of the following values:

'c'
   The specified value must be in character format, coded between single quotation marks, and one byte long.

**BLANK**
   Specifies that a space character is used as the default delimiter for external date representation.
   By default, this variable applies to FORMAT USER. It can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.

The default value is '-'.
The corresponding SYSIN parameter is OPTIONS DATEDELIM.
This parameter is also referred to as VUU055/DATEDEL.

**Delimiter for external date, format DELIMITED (VU055/DATEDEL)**
This parameter is optional. It specifies the default delimiter that is used in external date representations.

Specify one of the following values:

'c'
   The specified value must be in character format, coded between single quotation marks, and one byte long.

**BLANK**
   Specifies that a space character is used as the default delimiter for external date representation.
   By default, this variable applies to FORMAT USER. It can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.

The default value is '-'.
The corresponding SYSIN parameter is OPTIONS DATEDELIM.
This parameter is also referred to as VUU055/DATEDEL.

**Delimiter for external date, format VARIABLE (VU055/DATEDEL)**
This parameter is optional. It specifies the default delimiter that is used in external date representations.

Specify one of the following values:

'c'
   The specified value must be in character format, coded between single quotation marks, and one byte long.

**BLANK**
   Specifies that a space character is used as the default delimiter for external date representation.
   By default, this variable applies to FORMAT USER. It can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.

The default value is '-'.
The corresponding SYSIN parameter is OPTIONS DATEDELIM.
This parameter is also referred to as VUU055/DATEDEL.

**Delimiter for external date, format EXTERNAL (VUU055/DATEDEL)**

This parameter is optional. It specifies the default delimiter that is used in external date representations.

Specify one of the following values:

- **'c'**
  - The specified value must be in character format, coded between single quotation marks, and one byte long.

- **BLANK**
  - Specifies that a space character is used as the default delimiter for external date representation. By default, this variable applies to FORMAT USER. It can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.

  The default value is `'-'`.

  The corresponding SYSIN parameter is OPTIONS DATEDELIM.

This parameter is also referred to as VUU055/DATEDEL.

**Delimiter in external time, format USER (VUU056/TIMEDEL)**

This parameter is optional. It specifies the default delimiter that will be used in external time representations.

- **'c'**
  - The specified value must be in character format, coded between single quotation marks, and one byte long.

- **BLANK**
  - Specifies that a space character will be used as the default delimiter for external time representation.

  The default value is `'.'`.

  The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

This parameter is also referred to as VUU056/TIMEDEL.

**Delimiter in external time, format DSNTIAUL (VUU056/TIMEDEL)**

This parameter is optional. It specifies the default delimiter that will be used in external time representations.

- **'c'**
  - The specified value must be in character format, coded between single quotation marks, and one byte long.

- **BLANK**
  - Specifies that a space character will be used as the default delimiter for external time representation.

  The default value is `'.'`.

  The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

This parameter is also referred to as VUU056/TIMEDEL.

**Delimiter in external time, format DELIMITED (VUU056/TIMEDEL)**

This parameter is optional. It specifies the default delimiter that will be used in external time representations.

- **'c'**
  - The specified value must be in character format, coded between single quotation marks, and one byte long.
BLANK
Specifies that a space character will be used as the default delimiter for external time representation.
The default value is '\.'.
The corresponding SYSIN parameter is OPTIONS TIMEDELIM.
This parameter is also referred to as VUU056/TIMEDEL.

Delimiter in external time, format VARIABLE (VUU056/TIMEDEL)
This parameter is optional. It specifies the default delimiter that will be used in external time representations.
'c'
The specified value must be in character format, coded between single quotation marks, and one byte long.

BLANK
Specifies that a space character will be used as the default delimiter for external time representation.
The default value is '\.'.
The corresponding SYSIN parameter is OPTIONS TIMEDELIM.
This parameter is also referred to as VUU056/TIMEDEL.

Delimiter in external time, format EXTERNAL (VUU056/TIMEDEL)
This parameter is optional. It specifies the default delimiter that will be used in external time representations.
'c'
The specified value must be in character format, coded between single quotation marks, and one byte long.

BLANK
Specifies that a space character will be used as the default delimiter for external time representation.
The default value is '\.'.
The corresponding SYSIN parameter is OPTIONS TIMEDELIM.
This parameter is also referred to as VUU056/TIMEDEL.

Options apply to all formats (VUU057/OPALLFMT)
This parameter is optional. It specifies whether the formatting options that are specified in SYSIN in the OPTIONS block at the GLOBAL level or the UNLOAD level apply only to FORMAT USER or to all output formats. VUU057/OPALLFMT applies to the following SYSIN parameters of the OPTIONS block:
• NULL
• DATE
• TIME
• TIMESTAMP
• PIC
• DATEDELIM
• TIMEDELIM
• NULLID
• NULLPOS
• NULLPAD
• PADDING
TRIM

Specify one of the following values:

NO

The SYSIN parameters in the previous list apply only to FORMAT USER when they are specified at the GLOBAL level or at the UNLOAD level.

YES

The SYSIN parameters in the previous list apply to all output formats when they are specified at the GLOBAL level or at the UNLOAD level.

Attention: Specifying YES for this PARMLIB variable might affect the output data that is generated by existing production jobs. Change this variable to YES only after you have verified that existing Db2 HPU production jobs will not be affected by the change.

When the SYSIN parameters in the previous list are specified at the SELECT level (after the OUTDDN keyword), they always apply to all output formats. In this case, the value of OPALLFMT is ignored.

The default value is NO.

This parameter is also referred to as VUU057/OPALLFMT.

Trim the trailing blanks, format USER (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether Db2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO

Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

This parameter is also referred to as VUU059/ULTRIM.

Trim the trailing blanks, format DSNTIAUL (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether Db2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO

Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

This parameter is also referred to as VUU059/ULTRIM.

Trim the trailing blanks, format DELIMITED (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether Db2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:
NO
Trailing blanks are not removed from variable length strings.

YES
Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

This parameter is also referred to as VUU059/ULTRIM.

Trim the trailing blanks, format VARIABLE (VUU059/ULTRIM)
This parameter is optional. Use it to specify whether Db2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO
Trailing blanks are not removed from variable length strings.

YES
Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

This parameter is also referred to as VUU059/ULTRIM.

Trim the trailing blanks, format EXTERNAL (VUU059/ULTRIM)
This parameter is optional. Use it to specify whether Db2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO
Trailing blanks are not removed from variable length strings.

YES
Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

This parameter is also referred to as VUU059/ULTRIM.

String constant is unloaded as CHAR/VARCHAR data types (VUU064/STRNGCST)
This parameter is optional. It specifies whether string constants in the SELECT statements are handled as CHAR or VARCHAR data when the SELECT statement is processed in native mode.

CHAR
The string constants are handled as CHAR data unless the DSNTIAUL STRICT format is requested either by specifying DSNTIAULSTRICT(NO) as a subparameter of the ULOPTNS parameter or by using the FORMAT DSNTIAUL STRICT option in the UNLOAD command. When you specify CHAR, the output for expressions that have string constants might depend on whether the unload is performed in native or in SQL mode.

VARCHAR
The string constants are handled as VARCHAR data. When you specify VARCHAR, the output for expressions that have string constants does not depend on whether the unload is processed in native or in SQL mode. Consider specifying VARCHAR for unloads that use the DB2 YES option.

Important: If some of your processes use strings that are handled as CHAR data when Db2 HPU processes the unload in native mode, use the default value. Otherwise, specify VARCHAR to ensure
that the output of the unload for expressions that have string constants do not depend on the processing mode of the unload.

If you want to specify VARCHAR so that string constants are processed as VARCHAR data in native mode or SQL mode but you have some jobs that expect string constants to be processed as CHAR data, consider changing the SELECT statements for these unloaded by replacing any string constant `string_constant` with CHAR(`string_constant`).

The default value is CHAR.

This parameter is also referred to as VUU064/STRNGCST.

**Manage character loss, Db2 HPU syntax (VUU071/LOSSCHAR)**

This parameter is optional. It specifies the behavior of the product when significant data from a character column is lost during unload formatting because of an INTO or REFORMAT clause and a Db2 HPU syntax is specified.

Db2 HPU considers a data loss has occurred for a character column if the data truncation removes any character that does not belong to the padded part of the character string, i.e. a sequence of padding characters that ends at the right end of the character string.

The data loss is checked for the following data types:

- CHAR
- VARCHAR
- GRAPHIC
- VARGRAPHIC
- BINARY
- VARBINARY

Specify one of the following values:

**IGNORE**
- No control is done. The return code of the job is 0 unless another issue occurs.

**WARNING**
- A warning message is issued for any column affected by a data loss. The return code is 4 unless another issue occurs.

**REJECT**
- An error message is issued for any column affected by a data loss. The return code is 8 unless another issue occurs.

The default value is IGNORE.

The corresponding SYSIN parameter is CHARACTER_LOSS from the OPTIONS block.

This parameter is also referred to as VUU071/LOSSCHAR.

**Manage character loss, UNLOAD PLUS syntax (VUU071/LOSSCHAR)**

This parameter is optional. It specifies the behavior of the product when significant data from a character column is lost during unload formatting because of an INTO or REFORMAT clause and an UNLOAD PLUS syntax is specified.

Db2 HPU considers a data loss has occurred for a character column if the data truncation removes any character that does not belong to the padded part of the character string, i.e. a sequence padding characters that ends at the right end of the character string.

The data loss is checked for the following data types:

- CHAR
- VARCHAR
- GRAPHIC
- VARGRAPHIC
• BINARY
• VARBINARY

Specify one of the following values:

**IGNORE**
No control is done. The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss. The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss. The return code is 8 unless another issue occurs.

The default value is IGNORE.

The corresponding SYSIN parameter is CHARACTER_LOSS from the OPTIONS block.

This parameter is also referred to as VUU071/LOSSCHAR.

Manage character loss, FAST UNLOAD syntax (VUU071/LOSSCHAR)
This parameter is optional. It specifies the behavior of the product when significant data from a character column is lost during unload formatting because of an INTO or REFORMAT clause and a FAST UNLOAD syntax is specified.

Db2 HPU considers a data loss has occurred for a character column if the data truncation removes any character that does not belong to the padded part of the character string, i.e. a sequence padding characters that ends at the right end of the character string.

The data loss is checked for the following data types:
• CHAR
• VARCHAR
• GRAPHIC
• VARGRAPHIC
• BINARY
• VARBINARY

Specify one of the following values:

**IGNORE**
No control is done. The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss. The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss. The return code is 8 unless another issue occurs.

The default value is IGNORE.

The corresponding SYSIN parameter is CHARACTER_LOSS from the OPTIONS block.

This parameter is also referred to as VUU071/LOSSCHAR.

Manage numeric loss, Db2 HPU syntax (VUU072/LOSSNUM)
This parameter is optional. It specifies the behavior of the product when significant data from a numeric column is lost during unload formatting (via INTO or REFORMAT clause) and a Db2 HPU syntax is specified.

Db2 HPU considers a data loss has occurred for a numeric column the numeric value has changed.

The data loss is checked for the following data types:
• SMALLINT
• INTEGER
• BIGINT
• DECIMAL

Specify one of the following values:

**IGNORE**
No control is done. The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss. The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss. The return code is 8 unless another issue occurs.

The default value is IGNORE.

The corresponding SYSIN parameter is NUMERIC_LOSS from the OPTIONS block.

This parameter is also referred to as VUU072/LOSSNUM.

**Manage numeric loss, UNLOAD PLUS syntax (VUU072/LOSSNUM)**
This parameter is optional. It specifies the behavior of the product when significant data from a numeric column is lost during unload formatting (via INTO or REFORMAT clause) and an UNLOAD PLUS syntax is specified.

Db2 HPU considers a data loss has occurred for a numeric column the numeric value has changed.

The data loss is checked for the following data types:
• SMALLINT
• INTEGER
• BIGINT
• DECIMAL

Specify one of the following values:

**IGNORE**
No control is done. The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss. The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss. The return code is 8 unless another issue occurs.

The default value is IGNORE.

The corresponding SYSIN parameter is NUMERIC_LOSS from the OPTIONS block.

This parameter is also referred to as VUU072/LOSSNUM.

**Manage numeric loss, FAST UNLOAD syntax (VUU072/LOSSNUM)**
This parameter is optional. It specifies the behavior of the product when significant data from a numeric column is lost during unload formatting (via INTO or REFORMAT clause) and a FAST UNLOAD syntax is specified.

Db2 HPU considers a data loss has occurred for a numeric column the numeric value has changed.

The data loss is checked for the following data types:
• SMALLINT
• INTEGER
• BIGINT
• DECIMAL

Specify one of the following values:

**IGNORE**
No control is done. The return code of the job is 0 unless another issue occurs.

**WARNING**
A warning message is issued for any column affected by a data loss. The return code is 4 unless another issue occurs.

**REJECT**
An error message is issued for any column affected by a data loss. The return code is 8 unless another issue occurs.

The default value is **IGNORE**.

The corresponding SYSIN parameter is **NUMERIC_LOSS** from the OPTIONS block.

This parameter is also referred to as **VUU072/LOSSNUM**.

**Example: Manage numeric loss, with Db2 HPU, UNLOAD PLUS, FAST UNLOAD syntaxes**

Assume C1 has the **INTEGER** data-type and is unloaded into a **SMALLINT** data-type with the help of a **REFORMAT** clause:

• a row for which C1’s value is 100 and is unloaded without error as the value 100 is kept (not altered) in the SMALLINT data-type output.
• a row for which C1’s value is 1 000 000 (the internal representation is x'000F4240') is processed as follows according to the parameter value:
  – it is unloaded with C1 field containing the value 16960 (the internal representation is x'4240') without any message related to data loss if **IGNORE** is specified,
  – it is unloaded with C1 field containing the value 16960 (with internal representation is x'4240') with a warning message if **WARNING** is respected,
  – it is rejected with an error message if **REJECT** is specified.

**Error message limit, Db2 HPU parameters (VUU073/MAXERR)**

This parameter is optional. It specifies the maximum number of messages that are issued if, while processing an unload requested via the xxx syntax, Db2 HPU encounters errors (conversion, formatting...) affecting a specific row, i.e. a data related error. Specifying 0 means no limit applies. Use this parameter to limit the number of messages that are written into the JES spool.

The way Db2 HPU behaves and ends when such errors are encountered depends on the value of the **MAXERR_STRICT** subparameter of the VUU060/ULOPTNS parmlib parameter.

• if **MAXERR_STRICT(YES)** is set
  When row level errors occur, the unload
  – ends with completion code of 4 if the maximum number of errors specified by **MAXERR** has not been reached. The data set that contains the unloaded rows contains all the valid rows corresponding to the SELECT statement.
  – ends with completion code of 8 if the maximum number of errors specified by **MAXERR** has been reached. The data set that contains the unloaded rows contains a subset of the valid rows corresponding to the SELECT statement.

• if **MAXERR_STRICT(NO)** is set
  When row level errors occur, the unload:
  – ends with completion code of 4 if the maximum number of errors specified by **MAXERR** has not been reached and the latest row processed by Db2 HPU is not affected by such an error. The data
set that contains the unloaded rows contains all the valid rows corresponding to the SELECT
statement.
– ends with completion code of 8 if the maximum number of errors specified by MAXERR has been
reached or the latest row processed by Db2 HPU is affected by such an error. The data set that
contains the unloaded rows contains a subset of the valid rows corresponding to the SELECT
statement and an extra invalid row at the bottom of the data set.

Valid values are: 0-2147483647 (0 means 'no limit').
The default value is 0.
The corresponding SYSIN parameter is MAXERR.
This parameter is also referred to as VUU073/MAXERR.

Error message limit, UNLOAD PLUS syntax (VUU073/MAXERR)
This parameter is optional. It specifies the maximum number of messages that are issued if, while
processing an unload requested via the UNLOAD PLUS syntax, Db2 HPU encounters errors
(conversion, formatting...) affecting a specific row, i.e. a data related error. Specifying 0 means no limit
applies. Use this parameter to limit the number of messages that are written into the JES spool. The
way Db2 HPU behaves and ends when such errors are encountered depends on the value of the
MAXERR_STRICT subparameter of the VUU060/ULOPTNS parmlib parameter.
• if MAXERR_STRICT(YES) is set
  When row level errors occur, the unload
  – ends with completion code of 4 if the maximum number of errors specified by MAXERR has not
    been reached. The data set that contains the unloaded rows contains all the valid rows
    corresponding to the SELECT statement.
  – ends with completion code of 8 if the maximum number of errors specified by MAXERR has been
    reached. The data set that contains the unloaded rows contains a subset of the valid rows
    corresponding to the SELECT statement.
• if MAXERR_STRICT(NO) is set
  When row level errors occur, the unload
  – ends with completion code of 4 if the maximum number of errors specified by MAXERR has not
    been reached and the latest row processed by Db2 HPU is not affected by such an error. The data
    set that contains the unloaded rows contains all the valid rows corresponding to the SELECT
    statement.
  – ends with completion code of 8 if the maximum number of errors specified by MAXERR has been
    reached or the latest row processed by Db2 HPU is affected by such an error. The data set that
    contains the unloaded rows contains a subset of the valid rows corresponding to the SELECT
    statement and an extra invalid row at the bottom of the data set.

Valid values are: 0-2147483647 (0 means 'no limit').
The default value is 0.
The corresponding SYSIN parameter is DISCARDS.
This parameter is also referred to as VUU073/MAXERR.

Error message limit, FASTUNLOAD syntax (VUU073/MAXERR)
This parameter is optional. It specifies the maximum number of messages that are issued if, while
processing an unload requested via the FASTUNLOAD syntax, Db2 HPU encounters errors
(conversion, formatting...) affecting a specific row, i.e. a data related error. Specifying 0 means no limit
applies. Use this parameter to limit the number of messages that are written into the JES spool. The
way Db2 HPU behaves and ends when such errors are encountered depends on the value of the
MAXERR_STRICT subparameter of the VUU060/ULOPTNS parmlib parameter.
• if MAXERR_STRICT(YES) is set
  When row level errors occur, the unload
  – ends with completion code of 4 if the maximum number of errors specified by MAXERR has not
    been reached. The data set that contains the unloaded rows contains all the valid rows
    corresponding to the SELECT statement.
  – ends with completion code of 8 if the maximum number of errors specified by MAXERR has been
    reached or the latest row processed by Db2 HPU is affected by such an error. The data set that
    contains the unloaded rows contains a subset of the valid rows corresponding to the SELECT
    statement and an extra invalid row at the bottom of the data set.

Valid values are: 0-2147483647 (0 means 'no limit').
The default value is 0.
The corresponding SYSIN parameter is DISCARDS.
This parameter is also referred to as VUU073/MAXERR.
– ends with completion code of 4 if the maximum number of errors specified by MAXERR has not been reached. The data set that contains the unloaded rows contains all the valid rows corresponding to the SELECT statement.

– ends with completion code of 8 if the maximum number of errors specified by MAXERR has been reached. The data set that contains the unloaded rows contains a subset of the valid rows corresponding to the SELECT statement.

• if MAXERR.Strict(NO) is set

When row level errors occur, the unload

– ends with completion code of 4 if the maximum number of errors specified by MAXERR has not been reached and the latest row processed by Db2 HPU is not affected by such an error. The data set that contains the unloaded rows contains all the valid rows corresponding to the SELECT statement.

– ends with completion code of 8 if the maximum number of errors specified by MAXERR has been reached or the latest row processed by Db2 HPU is affected by such an error. The data set that contains the unloaded rows contains a subset of the valid rows corresponding to the SELECT statement and an extra invalid row at the bottom of the data set.

Valid values are: 0-2147483647 (0 means 'no limit').

The default value is 0.

The corresponding SYSIN parameter is DISCARDS.

This parameter is also referred to as VUU073/MAXERR.

Fill NULL with padding character, format USER (VUU075/NULLPAD)

This parameter is optional. It specifies whether the padding character is used to fill the output field when a null value is unloaded. Specify one of the following values:

NO
When a null value is unloaded, the output field is filled with binary zeroes(x'00').

YES
When a null value is unloaded, the output field is filled with the padding character.

The default value is YES.

The corresponding SYSIN parameter is OPTIONS NULLPAD.

Fill NULL with padding character, format DSNTIAUL (VUU075/NULLPAD)

This parameter is optional. It specifies whether the padding character is used to fill the output field when a null value is unloaded. Specify one of the following values:

NO
When a null value is unloaded, the output field is filled with binary zeroes(x'00').

YES
When a null value is unloaded, the output field is filled with the padding character.

The default value is NO.

The corresponding SYSIN parameter is OPTIONS NULLPAD.

Fill NULL with padding character, format VARIABLE (VUU075/NULLPAD)

This parameter is optional. It specifies whether the padding character is used to fill the output field when a null value is unloaded. Specify one of the following values:

NO
When a null value is unloaded, the output field is filled with binary zeroes(x'00').

YES
When a null value is unloaded, the output field is filled with the padding character.

The default value is YES.
The corresponding SYSIN parameter is OPTIONS NULLPAD.

**Fill NULL with padding character, format EXTERNAL (VUU075/NULLPAD)**

This parameter is optional. It specifies whether the padding character is used to fill the output field when a null value is unloaded. Specify one of the following values:

- **NO**
  - When a null value is unloaded, the output field is filled with binary zeroes (x'00').

- **YES**
  - When a null value is unloaded, the output field is filled with the padding character.

The default value is **YES**.

The corresponding SYSIN parameter is OPTIONS NULLPAD.

**NULL processing in DELIMITED format (VUU078/ULNULLFL)**

This parameter is optional. Use it to specify processing of NULL value for the DELIMITED format. Specify one of the following values:

- **UNDELIMITED_EMPTY_STRING**
  - Specifies that null values for strings will not be enclosed by the delimiter character that is specified in the DELIM delimiter-character.

- **DELIMITED_EMPTY_STRING**
  - Specifies that null values for strings will be enclosed by the delimiter character that is specified in DELIM delimiter-character.

- **NULLVAL(string)**
  - Specifies the string value that is written in the output field whenever the corresponding column table has a NULL value. The specified string must be surrounded between apostrophes but it is not enclosed in any delimiter in the output data set. The string is written as is regardless of any format. The string is converted into the output file CCSID. The value NULLVAL() specifies that the NULL value is an empty string. The string must be free of any space character and its length must not exceed 50 characters.

The valid values are UNDELIMITED_EMPTY_STRING, DELIMITEDEMPTY_STRING and NULLVAL(string) where string is a up to 50 characters blank free character string.

The default value is DELIMITED_EMPTY_STRING.

This parameter is also referred to as VUU078/ULNULLFL.

**LISTDEF behavior for table selection (VUU081/UNLLDTBB)**

Specifies how the "table spec" (TABLE creator-ID.table-name) of a LISTDEF statement is used to determine the list of table spaces associated with the LISTDEF statement. Specify one of the following values to get the selection mode described below:

- **TABLE**
  - a table space is selected if it contains at least:
    - a table which name matches the "table spec"

- **VIEW**
  - a table space is selected if it contains at least:
    - a table which name matches the "table spec"
    - a table involved in a view which name matches the "table spec"

The valid values are: TABLE, VIEW.

The default value is **TABLE**.

**Processing when LOB/XML not retrieved (VUU084/ULLOBBHV)**

The processing of a logical unload run in native mode and involving LOB or XML data might fail to retrieve the LOB or XML data associated with some of the records of the base table if no action to ensure the consistency of the table space containing the base table and the related LOB or XML table.
spaces is taken. This parameter allows specifying what action to take whenever such a situation occurs.

**STOP**
the processing stops with a return code of 8. The unloaded data are unusable.

**IGNORE**
the processing continues and the step ends with a return code 4. The unloaded data is available but some LOB or XML data are erroneous.

**ABEND**
a user ABEND is issued. The unloaded data are unusable.

The default value is STOP.

**Maximum number of issued messages for LOB/XML errors (VUU085/ULLOBBHM)**
When IGNORE is specified for the VUU084/ULLLOBHV parmlib parameter, Db2 HPU may unload inconsistent LOB or XML data for some records of the unloaded table and normally issues a message for every affected record. This parameter specifies the maximum number of messages issued in such a case. This number may be chosen from 1 to 100000.

The default value is 30.

**Default exclusion criteria for EXCLUDE_FIC option (VUU086/EXCLFIC)**
Specifies an exclusion criterion to apply when the eligible full image copies (FIC) are searched as requested by the COPYDDN LAST_IC or COPYDDN -integer options.

**NONE**
no kind of FIC is excluded.

**TO_SORT**
the images copies which processing by Db2 HPU requires a sort process are not eligible. This kind of image copies is created by the LOAD or REORG utilities (inline image copies) or by the COPYTOCOPY utility. The eligible image copy(ies) is(are) selected according to the selection criteria expressed by the GLOBAL, PARTITIONED,ANYTPE or CONSISTENT suboptions if any.

The default value is NONE.

This parameter is also referred to as VUU086/EXCLFIC.

For more details about the impact of the chosen value, the explanation of the counterpart EXCLUDE_FIC option.

**Request the display of a EXPLAIN report (VUU087/DISPEXPL)**
This parameter is optional. It specifies whether and where to display the report about the EXPLAIN statements performed during an unload. The report gives the following information:

- Excerpt of the rows of PLAN_TABLE and DSN_PGRANGE_TABLE resulting from an EXPLAIN statement related to a process of a SELECT.
- It also gives information about the information extracted: TABLES, PARTITIONS.

Specify either of the following allowed values:

**NO**
no report is displayed.

**IF,ddname**
the report is displayed to the specified ddname, only if this ddname is allocated. The specified ddname may be any valid ddname including SYSPRINT but should not be set with any other Db2 HPU reserved ddname (see “Ddname allocation” on page 87).

**YES,ddname**
the report is displayed to the specified ddname which is allocated by Db2 HPU as a SYSOUT data set if the corresponding ddname is not allocated. The specified ddname may be any valid ddname including SYSPRINT but should not be set with any other Db2 HPU reserved ddname (see “Ddname allocation” on page 87).
The content of the report produced by the DISEXPL feature is described below.

A header:

************************************************************************
*                          HPU EXPLAIN REPORT                          *
*           AS REQUESTED BY VUU087/DISPEXPL =value of VUU087/DISPEXPL  *
************************************************************************

that reminds what the current file contains.

It is followed by one section per logical unload which processing involved the EXPLAIN statement mechanism. Each section contains the following subsections:

A first subsection with information related to a given logical unload:

- SELECT LOCATION IN SYSIN : NO select_number POS(line,col)
- FEATURE REQUIRING EXPLAIN : requesting_feature
- SET QUERYNO EXPLAIN STATEMENT : query_number
- EXPLAINED STATEMENT : SELECT_statement

where:

- select_number is the number assigned by Db2 HPU to the select located at the line and columns mention in the POS(line,col) part
- line,col are the line and column in the SYSIN data set where the SELECT in question starts
- requesting_feature is the Db2 HPU feature that required the use of the EXPLAIN statement. It can be either LOCK YES IN SQL MODE or OPTIMIZE_PART_RANGE
- query_number is the query_number assigned by Db2 HPU to the SELECT statement
- SELECT_statement is the text of the SELECT statement on which the logical unload in question is based

A second subsection displays the excerpt of the PLAN_TABLE related to the logical unload

| PLAN_TABLE ROWS FOR QUERYNO: query_number |
| column_name : value                           |
| ...                                           |

where:

- query_number is the query_number assigned by Db2 HPU to the SELECT statement
- column_name is the name of the column from the DSN_PRGANGE_TABLE
- value is the value of the related column surrounded by | and | for value in external format and in binary format for binary values

A third subsection displays the excerpt of the PLAN_TABLE related to the logical unload

| DSN_PRGANGE_TABLE ROWS FOR QUERYNO: query_number |
| column_name : value                           |
| ...                                           |

where:

- query_number is the query_number assigned by Db2 HPU to the SELECT statement
- column_name is the name of the column from the DSN_PRGANGE_TABLE
- value is the value of the related column surrounded by | and | for value in external format and in binary format for binary values

A last subsection displays the feature conclusion which is formatted as described below.

If the feature is OPTIMIZE_PART_RANGE:
- FEATURE CONCLUSION:
  SELECTED PARTITIONS : list_of_partitions
  where:
  • list_of_partitions is the subset of partition to unload.

If the feature is LOCK YES IN SQL MODE:

- FEATURE CONCLUSION:
  TABLES AND PARTITIONS TO LOCK :
    TABLE : creator.table
    +- PARTITIONS : list_of_partitions
    ....
    TABLE : creator.table
    +- PARTITIONS : list_of_partitions
  where:
  • creator.table is the name of a table involved in the SELECT statement which has to be locked
  • list_of_partitions is the subset of partitions of the table to be locked when the table is partitioned

Default value: NO

**INTO clause rule if field_name without output-data-type(VUU089/INTORUL1)**

This parameter is optional. It allows to change the behavior of INTO clause, when the field-name is explicitly specified, but its output-data-type is not specified. Set one of the following values:

**COLTYPE**
The INTO clause get the output-data-type from the column type of the resulting table, the column-format-options specified in the INTO clause apply. Any REFORMAT clause is ignored.

**REFORMAT**
If a REFORMAT clause applies to the column type of the resulting table, the output-data-type specified through the REFORMAT clause applies, as well as the column-format-options except if a format option already exists in the INTO clause. If no REFORMAT clause applies, the output-data-type is the one of the column of the resulting table, and options from the INTO clause apply.

The default value is COLTYPE.

**Related concepts**

**DELIMITED format**
When you create output in the DELIMITED format, you can specify a separator character and a delimiter character.

**VARIABLE format**
When you create output in the VARIABLE format, the output is compatible with the Db2 LOAD utility input data set.

**EXTERNAL format**
When you create output in the EXTERNAL format, output fields are in the EXTERNAL format that corresponds to their default type, output records are fixed, and a field separator is not used.

**Related reference**

**FORMAT block syntax and description**
Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

**DATE format types**
Use the DATE format type to specify the output data format.

**TIME format types**
Use the TIME format type to specify the output data format.

**TIMESTAMP format types**
Db2 HPU Db2 Administration Tool and Db2 Launchpad parameters

The Db2 Administration Tool and Db2 Launchpad parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters that are required to integrate Db2 HPU with Db2 Admin.

The Db2 HPU parameters are designated in three ways:

• a long designation that describes what the parameter is used for
• two short names used as a reference especially within the messages texts:
  – one has the following format Vxxnnnn where xx are characters and nnnn are figures
  – the other one is a short label of 8 characters

The following list describes the Db2 HPU Db2 Administration Tool and Db2 Launchpad parameters. The parameters are listed in the following format:

description (Vxxnnnn/parameter-name)

High-level qualifier for the Db2 Admin data sets

This parameter is optional. It specifies the high-level qualifier of the Db2 Administration Tool libraries. This information is used by the sample programs INZADBI and INZDB2IX (in the SINZCLIST library) to update the Db2 Administration Tool and the Data Management Tools Launchpad tables.

This parameter is also referred to as VUU025.

Library which contains Db2 Admin commands tables

This parameter is optional. It specifies the name of the library that contains the Db2 Administration Tool command tables. This information is used by the sample programs INZADBI and INZDB2IX (in the SINZCLIST library) to update the Db2 Administration Tool and the Db2 Tools Launchpad tables.

This parameter is also referred to as VUU026.

Library which contains the ADBDMTI EXEC

This parameter is optional. It specifies the name of the library that contains the ADBMTI EXEC. This information is used by the sample programs INZADBI and INZDB2IX (in the SINZSAMP library) to update the Db2 Administration Tool and the Db2 Tools Launchpad tables.

This parameter is also referred to as VUU027.

Db2 HPU conversion parameters

The Db2 HPU conversion parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the conversion parameters that are used by Db2 HPU.

The Db2 HPU parameters are designated in three ways:

• a long designation that describes what the parameter is used for
• two short names used as a reference especially within the messages texts:
  – one has the following format Vxxnnnn where xx are characters and nnnn are figures
  – the other one is a short label of 8 characters

The following list describes the Db2 HPU conversion parameters. The parameters are listed in the following format:

description (Vxxnnnn/parameter-name)

Unicode Conversion Services load library (VZM006/SCUNMOD)

This parameter is optional. It specifies the name of the IBM Conversion Service Load Library. If you want to perform conversions that imply non-SBCS CCSIDs or pairs of SBCS CCSIDs that are not supported by the SYSSTRINGS catalog table, you must first install IBM OS/390® Support for Unicode. For more information about this program, see Program Directory for z/OS Support for Unicode and z/OS Support for Unicode Using Conversion Services.
This parameter is also referred to as VZM006/SCUNMOD.

**Unicode Conversion Services technique search order (VZM007/SCUNTSO)**

This parameter is optional. It specifies a list of technique search orders to be searched for the z/OS Unicode Services. Separate each value with a comma. When a conversion between two CCSIDs is required, Db2 HPU looks for a conversion that specifies one of the listed technique search orders until it finds one valid conversion.

**Example:** SCUNTSO = ER,,RE means that Db2 HPU will for the ER technique search order, followed by the blank technique search order, and then the RE technique search order.

The default value is blank.

If Db2 Version 8 or later is installed, usually the technique search order is equal to ER.

See z/OS Support for Unicode Using Conversion Services for more information.

This parameter is also referred to as VZM007/SCUNTSO.

**Unicode Conversion Services substitution character mode (VZM008/SCUNSUB)**

This parameter is optional. It specifies how the UNLOAD utility manages conversions through z/OS Unicode Services.

**YES**

Indicates that if a character in a string cannot be converted between two CCSIDs, it is replaced by the substitution character of the converter.

**NO**

Indicates that if a character in a string cannot be converted, the conversion of all the strings fails.

The default value is YES.

This parameter is also referred to as VZM008/SCUNSUB.

**Db2 HPU file management parameters**

The Db2 HPU file management parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the file management parameters that are used by Db2 HPU.

The Db2 HPU parameters are designated in three ways.

• a long designation that describes what the parameter is used for

• two short names used as a reference especially within the messages texts:
  • one has the following format Vxxnnnn where xx are characters and nnnn are figures
  • the other one is a short label of 8 characters

The following list describes the Db2 HPU file management parameters. The parameters are listed in the following format:

description (Vxxnnnn/parameter-name)

**Volume name of migrated object (VZM005/INFVSMIG)**

This parameter is optional. It specifies the volume name that is located in the ICF catalog for migrated files. The default value is MIGRAT, which corresponds to the value that is used by the DFSMShsm component of the IBM Data Facility Storage Management Subsystem (DFSMS).

This parameter is also referred to as VZM005/INFVSMIG.

**Numbers of devices for tape unit/storage class (VZM010/TAPEUNIT)**

This parameter is optional. Use this parameter to specify the maximum and default numbers of devices that can be used by Db2 HPU when you request that output files be allocated on tape by specifying a tape unit name or a storage class name. These devices numbers must be specified for each tape unit name or storage class name you intend to use for allocating the Db2 HPU output data sets. The value of this parameter must be formed according to the following syntax diagram:
Maximum number of disk units to allocate a work file (VUX010/LIMUNIT)

This parameter is optional. It specifies the maximum number of disk units to be used when Db2 HPU allocates a temporary work file. It is also taken into account to determine the number of work data sets specified as the DYNALLOC first operand of the SORT program when a Db2 HPU needs to perform a sort task.

The default value is 9.

The corresponding SYSIN keyword is LIMUNIT in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX010/LIMUNIT.

Note: When allocations are performed by Db2 HPU, the LIMUNIT value is used as the volume count (VOLCNT) allocation parameter unless it exceeds the maximum value accepted by the system for the device type involved. For example, if LIMUNIT is set to 255 and the works data sets are to be allocated on DASD devices for which the limit is 59, Db2 HPU allocates the work data sets with VOLCNT=59.

Maximum size for primary allocation of a work data set (VUX019/WRKMXPR)

This parameter is optional. It specifies the maximum size, in kilobytes, for the primary allocation of a work data set on DASD. When very large work data sets are used, the primary allocation might be distributed among several volumes according to the limit that was specified in the VUX010/LIMUNIT parameter.

Requirement: Regardless of the limit that was specified on the VUX010/LIMUNIT parameter, the value that you provide for the VUX019/WRKMXPR parameter must be lower than the capacity of the units that is used for these work data sets (VUM013). You must also consider that these units might be distributed among several volumes.

Valid values are 1 - 16777215.

The default value is 500000.

The corresponding SYSIN keyword is WRKMXPR in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX019/WRKMXPR.

Note: When Db2 HPU invokes the sort program, the number of work datasets to be used by the sort program is determined according to the following rules:

- The size of a work data set is assumed to be VUX019/WRKMXPR.
- The number of work files per allowed volume (i.e. VUX010/LIMUNIT) must not exceed 2.

These rules are implemented via the following formula: Number of work data sets = min(upper_round(2 * FILSZ * RECORD_LENGTH / WRKMXPR / 1024), 2 * LIMUNIT) where FILSZ stands for the estimated number of records to be sorted and RECORD_LENGTH for the mean value of the record length.

Although the above formula is always used to determine the allowed number of work data sets, the actual number of data sets might be lower if this calculated number exceeds the number of file the sort program can handle. For example, DFSORT V1R12.0 cannot use more than 255. Therefore,
although LIMUNIT is set to a value greater than or equal to 128, which would theoretically allow the sort program, to use more than 256 work data sets, no more than 255 files will be actually used.

**DSCB model for allocation of GDS (VUX033/TMPDSCB)**

This parameter is optional. It specifies the model data set control block (DSCB) to use when allocating generation data sets (GDS) and when a TEMPLATE is used to allocate new generations of a generation data group (GDG).

This parameter is also referred to as VUX033/TMPDSCB.

**Volumes for allocation of temporary data sets (VUM018/WRKVOL)**

This parameter is optional. It specifies the name of the volume where temporary data sets will reside.

This parameter is also referred to as VUM018/WRKVOL.

**Tape unit where the work data sets must be allocated (VUA007/WRKTUNIT)**

This parameter is optional. It specifies the name of the tape unit that is used to allocate temporary files.

If you use temporary files on tape, specify a tape unit or DASD device on this parameter. If you specify a DASD device, ensure that the pool of volumes that are associated with that unit has enough free space to store large data sets. If you do not specify a value, the utility allocates temporary files on the unit that is specified on the WRKUNIT parameter.

The corresponding SYSIN keyword is WRKTUNIT of the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUA007/WRKTUNIT.

**Maximum size for work data set on DASD (VUX016/WRKUNTSW)**

This parameter is optional. It specifies a threshold size (in kilobytes) for work data sets. All work data sets that exceed this threshold size will be allocated on the unit that is specified on the VUA007/WRKTUNIT parameter.

The corresponding SYSIN keyword is WRKUNTSW in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX016/WRKUNTSW.

**Maximum number of unit for tape temporary data set (VUX017/MAXTUNIT)**

This parameter is optional. It specifies the maximum number of tape units that are provided for work data sets that are used by a Db2 HPU job.

Valid values are 1 - 255.

The default value is 2.

The corresponding SYSIN keyword is MAXTUNIT in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX017/MAXTUNIT.

**Number of volumes for temporary data set on tape (VUX032/WRKTVCNT)**

This parameter is optional. It specifies the number of volumes to use for temporary data sets that are allocated on tape. Specifying 0 indicates that no VOLCOUNT parameter will be used for allocating tape files. In this case, up to five volumes are used.

Valid values are 0 - 255.

The default value is 0.

The corresponding SYSIN keyword is WRKTVCNT in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX032/WRKTVCNT.

**BUFNO for sequential QSAM (VUM022/QSBUFNO)**

This parameter is optional. It specifies the number of data buffers for sequential QSAM (the BUFNO parameter of the DCB for QSAM).
Valid values are 1 - 255.
The default value is 60.
This parameter is also referred to as VUM022/QSBUFNO.

**BUFND for sequential VSAM (VUM023/VSBUFND)**
This parameter is optional. It specifies the number of data buffers for sequential VSAM (the BUFND parameter of the ACB for VSAM).

**Note:** Specifying a large value might increase the amount of memory that Db2 HPU requires.
Valid values are 0 - 65535.
The default value is 360.
This parameter is also referred to as VUM023/VSBUFND.

**Db2 HPU reporting parameters**
The Db2 HPU reporting parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the reporting parameters that are used by Db2 HPU.

The Db2 HPU parameters are designated in three ways.

- a long designation that describes what the parameter is used for
- two short names used as a reference especially within the messages texts:
  - one has the following format Vxxnnnn where xx are characters and nnnn are figures
  - the other one is a short label of 8 characters

The following list describes the reporting parameters. The parameters are listed in the following format: 
*description (Vxxnnnn/parameter-name)*

**List of system codes for which no dump is produced (VZM009)**
This parameter is optional. It defines a list of system codes for which no dump is produced if an abend occurs during execution of Db2 HPU. You can specify up to eight lines. Each line can contain up to eight values. Separate each value by a comma.

After changing the content of this variable, you must regenerate the installation job related to the INZZSCOD template to activate the change.

The list can contain specific system codes that are coded on three hexadecimal digits or generic system codes that begin with X and include all the system codes for the number that follows it. For example, 0c1 is a specific system code, and X78 is a generic system code that includes all system codes that end with 78 (B78, D78, E78, and so on).

The default value is X22, X06, X37, 913, X78, X0A.
This parameter is also referred to as VZM009.

**Generate a tape usage report in SYSPRINT (VZM011/TAPERPT)**
This parameter is optional. Use it to specify whether to generate a report about the tape usage. The report displays the data set names (DSN) and their positions on the tape (FILESEQ) for each volume that is used.

Specify one of the following values:

**YES**
The tape usage report is generated.

**NO**
The tape usage report is not generated.

The default value is NO.
This parameter is also referred to as VZM011/TAPERPT.
Maximum number of messages for row structure errors (VUX018/LDSERRLM)
This parameter is optional. It specifies the maximum number of messages that are issued if Db2 HPU encounters a row structure error while reading the rows of a table space. Use this parameter to limit the number of messages that are written into the spool.

Valid values are 0 - 2147483647.

Important: Specifying a large number of records might increase the amount of storage that Db2 HPU requires.

This parameter does not have a default value.
This parameter is also referred to as VUX018/LDSERRLM.

Return code for unloaded rows (VUU024/UNLZLRC)
This parameter is optional. It specifies the return code that applies when no rows are unloaded by all SELECT statements of the UNLOAD command.

Valid values are 0 - 4 095.
The default value is 4.
Alteration applies only when original return code is zero unless keyword MAXRC is specified.
This parameter is also referred to as VUU024/UNLZLRC.

Return code if an object is missing (VUU024/UNLZLRC)
This parameter is optional. Specifies the return code that applies when an object included in a LISTDEF or LISTDEFTBV does not exist or when a LISTDEF or LISTDEFTBV expands to contain no objects. When this parameter is not specified, no specific return code applies for this situation.

Valid values are 1 - 4095.
There is no default value.
Alteration applies only when original return code is zero unless keyword MAXRC is specified.
This parameter is also referred to as VUU024/UNLZLRC.

Return code if zero line is unloaded (VUU024/UNLZLRC)
This parameter is optional. It specifies the return code when a LISTDEF or LISTDEFTBV clause returns no objects. When this parameter is not specified, no specific return code applies for this situation.

Valid values are 1 - 4095.
There is no default value.
Alteration applies only when original return code is zero unless keyword MAXRC is specified.
This parameter is also referred to as VUU024/UNLZLRC.

MAXRC alteration return code (VUU024/UNLZLRC)
This parameter is optional. When MAXRC is not specified, the original return code can be altered only when it is 0. When MAXRC is specified, it can also be altered when its value is 1 - 8.

Action when fully compliant LOADDNN cannot be generated (VUU082/ULRLDRC)
This parameter specifies how to handle the situation when Db2 HPU cannot generate LOAD control cards (or the corresponding features for the non HPU-native syntaxes) fully compliant with the DB2 LOAD utility requirements.

The concerned situations are:
• a column of the target table is different from a column of the source tables: i.e. it is defined by an SQL expression with at least a constant or an operator
• all the columns of the target table are not unloaded with the same CCSID
• no string delimiter is defined for an unload in DELIMITED format (or corresponding formats for non HPU-native syntaxes)
SORTKEYS or NUMRECS is specified for an unload involving unsupported SELECT statement and can therefore not be computed.

Specify one of the following value:

**IGNORE**
Db2 HPU reports the situation by issuing information messages. The completion code related to this situation is 0 and does not affect the global completion code.

**WARNING**
Db2 HPU reports the situation by issuing warning messages. The completion code related to this situation is 4 and affects the global completion code if lower.

**REJECT**
Db2 HPU reports the situation by issuing error messages. The completion code related to this situation is 8 and affects the global completion code if lower.

The data set containing the unloaded data and the LOAD control statements are written regardless of the value specified for this parameter.

The default value is **WARNING**.

This parameter is also referred to as VUU082/ULRLDRC.

**Frequency to display information messages (VUU050/ULFRQMSG)**
This parameter is optional. It specifies that Db2 HPU issues informational messages that display the current number of unloaded rows for each output file.

\( n \)
At the end of processing, Db2 HPU issues message INZX089, which indicates the total number of rows for each output file. A non-null integer value means that this message is issued for each \( n \) rows.

The default value is 0.

This parameter is also referred to as VUU050/ULFRQMSG.

**Additional parameter for information messages (VUU050/ULFRQMSG)**
This parameter is optional. It specifies interval time since the previous message was issued or since the first record was unloaded.

Specify one of the following values:

**DELTA**
Messages INZX089 and INZX090 are issued to indicate the elapsed time since the previously issued message.

**TOTAL**
Messages INZX089 and INZX090 are issued to indicate the total elapsed time since the beginning of the unload process.

**TOTAL** is the default when a non-null value is specified and neither **DELTA** nor **TOTAL** is specified.

**TOTAL** and **DELTA** are mutually exclusive.

The default value is **TOTAL**.

This parameter is also referred to as VUU050/ULFRQMSG.

**The location where Db2 HPU writes information messages (VUU051/PROCMSG)**
This parameter is optional. It specifies where to write informational messages that correspond to the unload process of the table space, image copy, or partitions and indexes.

Specify one of the following values:

**WTO**
Messages are issued as write to operator messages in the system log.
**ddname**
Messages are issued in the corresponding ddname. If the corresponding ddname is not already allocated in the JCL, it is dynamically allocated as SYSOUT=*.
This ddname can be equal to SYSPRINT, in which case these messages might be mixed with other messages that Db2 HPU issues.

All messages that are issued in the output file are prefixed with the system time in *hh:mm:ss* format.

The default value is WTO.

The corresponding SYSIN parameter is OPTIONS PROCMMSG.
This parameter is also referred to as VUU051/PROCMSG.

**Location of messages issued for each SELECT statement (VUU052/SELMSG)**
This parameter is optional. It specifies where to write informational messages INZX089 and INZX090 that are issued for each SELECT statement.

Specify one of the following values:

**WTO**
Messages that correspond to the SELECT statements are issued as write to operator messages in the system log.

**NUMBERED**
Messages that correspond to the SELECT statements are issued in a ddname for each SELECT statement. If this ddname is not already allocated in the JCL, it is dynamically allocated as SYSOUT=*.
The format of the generated ddname is UxxSyyyy, where *xx* is the two-digit UNLOAD number and *yyyy* is the SELECT number for this unload.

**ddname**
Messages that correspond to the SELECT statements are issued in the corresponding ddname. If this ddname is not already allocated in the JCL, it will be dynamically allocated as SYSOUT=*.
This ddname can be equal to SYSPRINT, in which case these messages might be mixed with other messages that Db2 HPU issues.

The default value is WTO.

The corresponding SYSIN parameter is OPTIONS SELMSG.
This parameter is also referred to as VUU052/SELMSG.

**Generate a list of all OBIDs in the FIC (VUU053/OBIDRPT)**
This parameter is optional. It specifies whether to generate a list of all object IDs (OBIDs) that are found in the full image copy (FIC). It applies only when you are unloading from an image copy.

**YES**
Generates a list of all OBIDs that are found in the FIC.

**NO**
Does not generate a list of all OBIDs that are found in the FIC.

The default value is NO.

The corresponding SYSIN parameter is OBID_REPORT in the COPYDDN option.
This parameter is also referred to as VUU053/OBIDRPT.

**Return code when the utility has switched to SQL mode (VUU069/SQLSWIRC)**
This parameter is optional. It specifies the return code that applies when the processing of a select statement has automatically switched to SQL mode when the statement is not supported in native mode. It is not applicable when the SQL mode is explicitly requested (e.g. Db2 HPU FORCE is specified).

Valid values are 0 - 4 095.

The default value is 4.
This parameter is also referred to as VUU069/SQLSWIRC.

Db2 HPU tuning parameters

The Db2 HPU tuning parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the tuning parameters that are used by Db2 HPU.

The Db2 HPU parameters are designated in three ways.

• a long designation that describes what the parameter is used for
• two short names used as a reference especially within the messages texts:
  – one has the following format Vxxnnnn where xx are characters and nnnn are figures
  – the other one is a short label of 8 characters

The following list describes the tuning parameters. The parameters are listed in the following format:
description (Vxxnnnn/parameter-name)

Default size of the input file or object for Db2 HPU (VUX003/SIZE)

This parameter is optional. It specifies the default number of records when allocating the resources (work data sets and FILSZ parameter) that are required by the SORT processing. The default value is used only when Db2 HPU is unable to estimate the number of rows to be unloaded, which happens only in very specific cases.

This parameter is ignored when the rows are selected by Db2. Therefore, set VUX003/SIZE to a typical or maximum number of rows for the table spaces that are usually processed by Db2 HPU.

VUX003/SIZE is expressed as a number of records.

Valid values are 1000 - 2147483647

Important: Specifying a large number of records might increase the amount of storage that Db2 HPU requires.

The default value is 1000000 records.

This parameter is also referred to as VUX003/SIZE.

Minimum memory size for each sort process (VUX004/LOWMEM)

This parameter is optional. It specifies the memory size, in bytes, below the 16 MB line that is used by the sort process. Db2 HPU considers LOWMEM only during parallel processing. When Db2 HPU prepares to start one or more SORT tasks (with the exception of the first SORT), it checks whether the amount of remaining memory under the line is greater or equal to LOWMEM.

Take the following into account in order to determine the MAXSORT and LOWDEM parameters:

1. Determine the maximum storage that is allowed for a SORT below the 16 MB limit; this value is referred as SORTMAXMEMBELOW hereafter. For example, SORTMAXMEMBELOW is the value of VSCORE parameter of the DFSORT installation and the value of VSCORE parameter of the SYNCSORT installation. Refer to the documentation of your SORT program to determine the SORTMAXMEMBELOW value if you use another one.

2. Determine the minimum amount of memory required below the 16 MB line by your SORT program: This value is referred as SORTMINMEMBELOW hereafter. To do so, either refer to your SORT program documentation or run a sample of SORT program sorting no rows or a very little number of rows and check the amount of memory this program has used below the 16MB line (see the SYS value given by the IEF371I message). SORTMINMEMBELOW is about 2070 000 bytes for DFSORT VR1R10.

3. Determine the remaining memory below the 16 MB limit when the INZUTILB module is loaded in memory: this value is referred as HPUMEMBELOW hereafter. To do so, run a sample Db2 HPU job requesting a basic unload (the IVP job is suitable) with the QUIESCE YES option and no SORT request. Check the amount of memory this program has used below the 16 MB line (see the SUS value given by the IEF374I message).

Set the LOWMEM and MAXSORT according to the following rules:
• \( \text{SORTMINMEMBELOW} < \text{LOWMEM} < \text{SORTMAXMEMBELOW} \)
• \( 1 < \text{MAXSORT} < (16 \text{ MB} - \text{HPUMEMBELOW}) / \text{LOWMEM} \)

To prevent failures, do not set LOWMEM lower than \( \text{SORTMINMEMBELOW} \).

To maximize the amount of memory that is used, set LOWMEM to a value less than or equal to the value of \( \text{SORTMAXMEMBELOW} \). Setting LOWMEM to a higher value would lead to a waste of memory equal to \( \text{LOWMEM} - \text{SORTMINMEMBELOW} \). Setting LOWMEM equal to \( \text{LOWMAXMEMBELOW} \) ensures that any SORT task launched by Db2 HPU can use an amount of memory up to that value of \( \text{SORTMAXMEMBELOW} \). However, as \( \text{SORTMAXMEMBELOW} \) might not always be needed, consider setting LOWMEM to a lower value so that Db2 HPU will attempt to launch more SORT tasks. Do not decrease LOWMEM to a too low value as it can lead to failures because some SORT tasks might run out of memory. The default setting (in parmlib) might not be relevant for some complex unload jobs where a lot of tasks are involved and parallelism is enabled. In such a case, it is recommended to adjust the value of LOWMEM and MAXSORT via the Technical Parameters option block in the GLOBAL block in SYSIN.

To improve performance, increase MAXSORT and decrease LOWMEM.

To minimize the consumption of memory below the 16 MB line, decrease MAXSORT and increase LOWMEM.

Valid values are \( 1 - 2147483647 \).

The default value is 270000.

The corresponding SYSIN keyword is LOWMEM in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUX004/LOWMEM.

**Wait unit in seconds/100 (VUX007/WAITUNIT)**

This parameter is required. It specifies the wait time (in hundredths of seconds) between two unsuccessful tests of the STOP command.

Valid values are \( 1 - 2147483647 \).

The default value is 100 (1 second).

This parameter is also referred to as VUX007/WAITUNIT.

**Number of wait periods before issuing a WTOR (VUX008/WAITQTY)**

This parameter is required. It specifies the number of times an unsuccessful STOP command will be tested before a WTOR is sent to the console. If the operator answers CANCEL (C), Db2 HPU will stop with return code 8. If the operator answers WAIT (W), the wait process starts again.

Valid values are \( 1 - 2147483647 \).

The default value is 20.

This parameter is also referred to as VUX008/WAITQTY.

**Wait time for the WTOR reply (VUX009/WAITQTYM)**

This parameter is required. It specifies the maximum wait time (in seconds) before an answer is sent to the WTOR message. Utility execution will stop beyond this limit (return code 8).

Valid values are \( 1 - 2147483647 \).

The default value is 60.

This parameter is also referred to as VUX009/WAITQTYM.

**Maximum degree of parallelism for LDS reading (VUX025/PARALLEL)**

This parameter is optional. It specifies the parallelism degree for an unload job when you are unloading a partitioned table space. This parameter indicates the maximum number of partitions that are processed in parallel. However, if the unload job includes a sort process, the VUX025 parameter is limited to the use of the VUX005/MAXSORT parameter. When a sort process is included, if the value of the VUX005/MAXSORT parameter is lower than the value that is specified in the VUX025/
PARALLEL parameter, the VUX005/MAXSORT parameter is used. You can specify a value of 1 or greater.

Valid values are 1 - 65535.

The default value is 20.

This parameter is also referred to as VUX025/PARALLEL.

Maximum degree of parallelism between unload tasks (VUX030/UTLPARAL)

This parameter is optional. It specifies the parallelism degree when you are starting several unload tasks for the same UNLOAD command. An unload task can be when you unload a table space (each table space that is accessed counts for one task, whether it is partitioned or not), or a SELECT statement that is processed by Db2. If more than one SELECT statement is processed by using Db2, Db2 HPU will count one task for all of these SELECT statements. You can specify a value of 1 or greater.

Valid values are 1 - 65535.

The default value is 5.

This parameter is also referred to as VUX030/UTLPARAL.

Number of rows retrieved by an SQL FETCH function (VUU035/ULROWSET)

This parameter is optional. It specifies the number of rows that will be retrieved by a single SQL FETCH. Specifying the number of rows that will be retrieved improves the performance for SELECT statements that are processed by Db2 (using DB2 FORCE or DB2 YES with unsupported SELECT statements). When you specify 1, the multi-fetch function is not used. A standard single-row FETCH is used instead. This function is available only in Db2 Version 8 and later releases.

Valid values are 1 - 32767.

The default value is 100.

This parameter is also referred to as VUU035/ULROWSET.

Enforce partition parallelism for unloading a table into a single file (VUU036/GBLPARAL)

This parameter is optional, or unloads run in native mode from data sets per partition into a single output file, it specifies whether parallelism regarding partitions must be used when reading the input data.

Specify one of the following values:

YES
Parallelism is enforced when table spaces are unloaded. The parallelism degree is set in variables VUX025/PARALLEL and VUX005/MAXSORT when a sort is requested. YES is the recommended value.

NO
Parallelism is disabled unless the PARALLELISM(n, ,) option (Db2 HPU syntax only) is specified with n > 1. This value is meant for the unexpected case where the overall benefit for such jobs would not be sufficient compared to the extra cost of parallelism in terms of CPU and memory resources.

For compatibility reasons, the default value is NO.

This parameter is also referred to as VUU036/GBLPARAL.

ZIIP usage (VUX039/ZIIP)

This parameter is optional. It allows to enable or disable the usage of ZIIP processors. Valid values are a percentage from 0 to 100 or OFF.

percentage
Specify an integer value from 0 to 100 to specify the percentage of eligible tasks (SRB tasks) Db2 HPU requests the system to offload to ZIIP processors

OFF
Disables the usage of zIIP processors.
The default value is OFF.

**Important:** do not set this parameter to 0 to disable the usage of zIIP processors. Specifying 0 would request Db2 HPU to prepare the tasks eligible for zIIP processing which implies an extra CPU cost but would not request any offload of those tasks to zIIP processor because of the percentage set to 0. This would lead to an undesirable increase of the Global Purpose CPU use. This value is meant for diagnosis purpose. Ask your system specialist to determine the best value of the percentage to choose.

This parameter is also referred to as VUX039/ZIIP.

**Size of work areas buffers (VUX040/UBUFSIZE)**

This parameter is optional. It allows to specify the size expressed in kB of the buffers that are used for communication between some Db2 HPU tasks, such as reading LDS, formatting data, and sorting data. A higher value has little impact on performance, but a lower value forces Db2 HPU to change tasks more frequently, which increases the wait and CPU times. The shipped value, i.e. the default value, is suitable for most of the use cases. Do not change the shipped value unless directed to do so by the IBM support.

Valid values are 256 to 8192.

The default value is 512.

This parameter is also referred to as VUX040/UBUFSIZE.

**Number of buffers of work areas (VUX041/UBUFNO)**

This parameter is optional. It allows to specify the number of buffers that are used for communication between some Db2 HPU tasks, such as reading LDS, formatting data, and sorting data. The shipped value, i.e. the default value, is suitable for most of the use cases. Do not change the shipped value unless directed to do so by the IBM support.

Default value is 3. This settings is meant for balanced performance over all kind of unloads.

Valid values are 2 to 8.

The default value is 512.

This parameter is also referred to as VUX041/UBUFNO.

**ZIIP maximum parallelism (VUX042/ZIIPMXPL)**

This parameter is optional. It specifies the maximum parallelism degree applicable to Linear Data Set reading when the zIIP processors use is put into action. This parameter might be needed to tune the sharing of the zIIP processors with other processes. Valid values are integers 0 - 65535.

0

This specific value means the maximum parallelism degree is given by the number of zIIP processors available on the LPAR

integer from 1 to 65535

The specified number is taken, as is, as the maximum parallelism degree.

The default value is 0.

This parameter is also referred to as VUX042/ZIIPMXPL.

**ZIIP maximum parallelism (VUX043/ZIIPMXPU)**

This parameter is optional. It specifies the maximum parallelism degree applicable when several unload tasks are started for the same UNLOAD command and zIIP processors is put into action. This parameter might be needed to tune the sharing of the zIIP processors with other processes. Valid values are 0 - 65535.

0

This specific value means the maximum parallelism degree is given by the number of zIIP processors available on the LPAR.

integer from 1 to 65535

The specified number is taken, as is, as the maximum parallelism degree.

The default value is 0.
This parameter is also referred to as VUX043/ZIIPMXPU.

**Support parallelism for unloading several partitions (VUU044/SUBTKSOF)**

This parameter is optional. For unloads run in native mode from data sets per partition into a single output file, it specifies whether parallelism regarding partitions must be used when writing the output data and applies only if VUU036/GBLPARAL is set to YES.

Specify one of the following values:

- **NO**
  - Does not support parallelism when data is unloaded from several partitions into a single output data set.

- **YES**
  - Supports parallelism when data is unloaded from several partitions into a single output data set.
  - YES is the recommended value.

For compatibility reasons, the default value is NO.

This parameter is also referred to as VUU044/SUBTKSOF.

**Maximum number of partitions processed in one unload (VUU060/ULMAXPAR)**

This parameter is optional. It specifies the maximum number of partitions that Db2 HPU can process in a single unload operation. ULMAXPAR affects logical unload operations (SELECT) that are processed natively and physical unload operations only. You can use this parameter to reduce the memory consumption of Db2 HPU.

If the total number of partitions that have to be processed for a table space is larger than the value that is specified in ULMAXPAR, Db2 HPU automatically splits the single unload operation into several unload operations to respect the value of ULMAXPAR. If the value of ULMAXPAR is smaller than the value of PARALLELISM at the partition level, the effective partition parallelism is limited by ULMAXPAR.

Valid values are 0 - 4096.

The default value is 0, which means that splitting is not done. When you specify a non-zero value, ORDER BY and ORDER CLUSTER clauses can be used only when each partition is unloaded into a separate file. When you specify a non-zero value and a split is done, the OUTMAXROWS or UNLMAXROWS setting applies to each partition.

The corresponding SYSIN keyword is MAXPART, which can be specified in the GLOBAL OPTIONS block or the UNLOAD block.

This parameter is also referred to as VUU060/ULMAXPAR.

**Size of the buffer used to retrieve LOB data (VUU041/ULOCSIZE)**

This parameter is optional. It applies when LOB data is unloaded in SQL mode and the internal process to retrieve the LOB data uses LOB locators which occurs as indicated in the table below:

<table>
<thead>
<tr>
<th>LOB data in output</th>
<th>Retrieval mode</th>
<th>VUU041/ULOCSIZE applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-stream</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>LOB file</td>
<td>LOB file reference</td>
<td>No</td>
</tr>
<tr>
<td>SPANNED</td>
<td>LOB locator</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Keep the default value or consider cautious changes to improve the performance of unload of LOB data in SQL mode. Make sure to run a comprehensive set of unload run in SQL mode and involving LOB where VUU041/ULOCSIZE applies afterward.

Valid values are 1 - 16000000.

The default value is 1000000.

This parameter is also referred to as VUU041/ULOCSIZE.
Wait/retry function when resources are unavailable (VUU047/RETRYMOD)

This parameter is optional. If a resource is not available, it specifies whether Db2 HPU uses wait/retry logic. This parameter controls the following availability criteria:

- The status of the Db2 object (table space, partition, or index) that prevents processing.
- If LOCK YES is specified in the SYSIN and a LOCK cannot be taken on the corresponding object.
- If the dynamic allocation of the LDS files of the table space or index to be unloaded fails.

Specify one of the following values:

**NONE**
Db2 HPU does not use wait/retry logic.

**ALL**
Db2 HPU uses wait/retry logic for the three kinds of situations in the previous list.

**STATUS**
Db2 HPU uses wait/retry logic when the status of the Db2 object, table space, partition, or index prevents processing. The following statuses prevent processing:

- CHKP
- GRECP
- UTUT
- PSRBD
- WEPR
- RBDP
- RBDP*
- REORP
- DBETE
- RESTP
- PSRCP
- LPL
- RELDP
- REFP
- RECP
- RECP*

**LOCK**
Db2 HPU uses wait/retry logic when a LOCK is requested by the LOCK YES option and the LOCK cannot be taken.

**LDS**
Db2 HPU uses wait/retry logic when LDS allocation fails.

You can specify multiple values except when you specify ALL or NONE. If you specify multiple values, separate them by using commas without spaces. Specify the wait time (in hundredths of a second) between two unsuccessful tests of the resource availability with PARMLIB parameter VUU048/RETRYW, and specify the number of retries with PARMLIB parameter VUU049/RETRYNB.

The default value is NONE.

This parameter is also referred to as VUU047/RETRYMOD.

Wait time between tries to access unavailable resources (VUU048/RETRYW)

This parameter is optional. It specifies the wait time (in hundredths of seconds) between two unsuccessful tests of the resource availability that is defined by the VUU047/RETRYMOD parameter.

Valid values are 1 - 32767.
The default value is 100 (1 second).

This parameter is also referred to as VUU048/RETRYW.

**Number of retries when resources are unavailable (VUU049/RETRYNB)**

This parameter is optional. It specifies the number of times to test the availability of a resource before stopping with return code 8.

Valid values are 1 - 32767.

The default value is 20.

This parameter is also referred to as VUU049/RETRYNB.

**Concurrent access resolution for prepared statement (VUU061/ACCPREP )**

This parameter is optional. Use it to specify the CONCURRENT ACCESS RESOLUTION option for the PREPARE statement that Db2 HPU will use to process SELECT statements by using Db2.

Specify one of the following values:

**WAIT**

The WAIT FOR OUTCOME clause is specified to wait for the commit or rollback of data that is being updated or deleted.

**USE**

The USE CURRENTLY COMMITTED clause is specified to use the currently committed version of the data that is being updated or deleted. The USE option is supported only in Db2 10 and later releases.

The VUU061 parameter does not have a default value.

This parameter is also referred to as VUU061/ACCPREP.

**Management of exit written in C or COBOL (VUU079/EXITHLL)**

This parameter is optional. It specifies whether your unload jobs invoke user-exits written in high-level language (C, COBOL) or exploit the Db2 HPU DDL generation feature that invokes a high-level language embedded exit. The later is request either via the UNLOAD PLUS syntax with the CNTLCARDS DB2 or CNTLCARDS DB2DDL options or via the HPU native syntax via the LOADDDL TABLE option).

**YES**

Some of your unload jobs invoke user-exits written in high-level language (C, COBOL) or exploit the Db2 HPU DDL generation feature.

**NO**

None of your unload jobs invoke user-exits written in high-level language (C, COBOL) or exploit the Db2 HPU DDL generation feature. Specifying NO saves up memory.

The default value is NO.

**Note:** When the setting for this parameter is changed, the installation job for the INZALIAS template must be regenerated and submitted to activate the change.

This parameter is also referred to as VUU079/EXITHLL.

**Number of simultaneously opened LOB LDS(VUU080/MXOPNLLD)**

This optional parameter specifies how many LDS, from 1 to 254, of a multi- LDS LOB tablespace can be kept allocated and opened simultaneously for unloading LOB data in native mode.

When the number of LDSes of the LOB table space is less than this value, each LDS of the LOB tablespace is allocated only once. Otherwise, some LDSes of the LOB table might be opened several times depending on where the data to be unloaded is located. For better performance, consider setting this parameter to the highest value but be aware that setting a too big value might lead to resource shortage when many LOB LDSes are involved.

The default value is 16.

The corresponding SYSIN keyword is MXOPNLLD in the Technical Parameters block. This parameter is also referred to as VUU080/MXOPNLLD.
Request the display of implicit parameters (VUU083/DISPPARM)

This parameter allows to specify whether and, when relevant, where to display the values of the implicit parameters (i.e. not specified by the UNLOAD command but set via the parmlib).

Specify either of the following allowed values:

NO
no comprehensive list of the values of the implicit parameters is displayed.

IF,ddname
a comprehensive list of the values of the implicit parameters is printed to the specified ddname, only if this ddname is allocated. The specified ddname may be any valid ddname including SYSPRINT except for any other Db2 HPU reserved ddname (see the Db2 HPU batch mode operation/Ddname allocation section).

YES,ddname
a comprehensive list of the values of the implicit parameters is printed to the specified ddname which is allocated by Db2 HPU as a SYSOUT data set if the corresponding ddname is not allocated. The specified ddname may be any valid ddname including SYSPRINT except for any other Db2 HPU reserved ddname (see the Db2 HPU batch mode operation/Ddname allocation section).

When either IF,ddname is specified and ddname is allocated or YES,ddname is specified the values of the implicit parameters are written to the specified ddname as described below.

The output consists of the following parts:

1. a section that starts with

```
**************************************************
* PARMLIB PARAMETERS
* READ FROM
* parmlib origin
**************************************************
```

shows the value of the parameters as set via the parmlib that includes the following subsections in the mentioned sequence:

a. the "Product parameters" subsection that starts with

```
************** PRODUCT PARAMETERS **************
```

and ends with the beginning of the next section.

b. the "LPAR parameters" subsection that starts with

```
************** LPAR PARAMETERS **************
```

and ends with the beginning of the next section.

c. the "Db2 parameters" subsection that starts with

```
************** DB2 PARAMETERS FOR ssid ***********
```

and ends with

```
**************************************************
* END OF PARMLIB PARAMETERS *
**************************************************
```

The format for the parameters displayed is this section is : "VXXnnn/PARMNAME = parameter_value where VXXnnn/PARMNAME is the short name for a given parameter and parameter_value is the value of the parameter as set via the parmlib for the concerned LPAR and Db2 subsystem or group attach.

2. a second section that starts with

```
****** PARAMETERS OVERRIDEN BY SYSIN OPTIONS *****
```
and ends at the bottom of the file. This section displays the values of the parameters that are explicitly overridden by a specification from the UNLOAD command as a table with the following columns:

**PARMLIB PARAM**
this column contains the short name of the overridden parmlib parameter.

**OVERRIDDEN BY SYSIN OPTION**
this column contains the keywords from the UNLOAD command responsible for the parmlib parameter to be overridden. It allows to determine the value that applies for a given parameter for the unload processes the SYSIN option applies to. The applicability of the value is given by the "location" columns.

**LOCATION**
a set of three columns gives the location of the SYSIN option that overrides the value of the related parameter. The three columns give the following details

- **LEVEL**: the hierarchical level where the option is specified (i.e. GLOBAL or UNLOAD or SELECT)
- **UNL#**: Unload number (if the option is specified at the UNLOAD or SELECT level)
- **SEL#**: Select number (if the option is specified at the SELECT level)

An additional note entry (*) at the end of the row (see the row for ULDATE in the following example) is added to indicate that the value only applies to the USER format as the VUU057/OPALLFMT parameter is set to NO.

<table>
<thead>
<tr>
<th>PARM LIB PARAM</th>
<th>OVERRIDDEN BY SYSIN OPTION</th>
<th>LOCATION</th>
<th>LEVEL</th>
<th>UNL#</th>
<th>SEL#</th>
</tr>
</thead>
<tbody>
<tr>
<td>VUU011/ULSEDB2</td>
<td>DB2 NO</td>
<td>UNLOAD</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VUU014/ULNULL</td>
<td>NULL OFF</td>
<td>SELECT</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>VUU015/ULDATE</td>
<td>DATE DATE_E</td>
<td>UNLOAD</td>
<td>1</td>
<td></td>
<td>(+)</td>
</tr>
</tbody>
</table>

**Note**: The effect of the REFORMAT, INTO and FORMAT clauses do not explicitly override the parmlib parameter but actually set the value of some parameters to predetermined values is not reported by this section.

The default value is NO.

The corresponding SYSIN keyword is DISPLAY_PARM in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

This parameter is also referred to as VUU083/DISPPARM.

**Db2 HPU sort parameters**
The Db2 HPU sort parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the sort parameters that are used by Db2 HPU.

The Db2 HPU parameters are designated in three ways.

- a long designation that describes what the parameter is used for
- two short names used as a reference especially within the messages texts:
  - one has the following format Vxxnnnn where xx are characters and nnnn are figures
  - the other one is a short label of 8 characters

The following list describes the common sort parameters. The parameters are listed in the following format:

**description** (Vxxnnnn/parameter-name)

**Maximum number of active sorts processes (VUX005/MAXSORT)**
This parameter is optional. It specifies the maximum number of active sorts that can run in the same step when parallel processing is involved. IBM DFSORT for z/OS supports MAXSORT values 1 - 20, and all of its modules are reusable.
Most of the non-IBM sort products contain modules that require MAXSORT=1 because not all of their modules are reusable.

To improve performance, increase MAXSORT and decrease LOWMEM.
To minimize the consumption of memory below the line, decrease MAXSORT and increase LOWMEM.
Set VUX005/MAXSORT to a value lower than or equal to the value of VUX025/PARALLEL to get the optimal behavior.
Valid values are 1 - 32767.
The default value is 20.
The corresponding SYSIN keyword is MAXSORT in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.
This parameter is also referred to as VUX005/MAXSORT.

Allocation distribution for the sort input file (VUX006/WRKSPACE)
This parameter is optional. It specifies the percentage of space that is allocated for the sort input file. WRKSPACE can be one of the following values:
Specify one of the following values:

**PARTIAL**
Corresponds to a primary and secondary allocation that is equal to 50 percent of the estimated size of the file to be sorted. Two volumes are allowed for this allocation.

**FULL**
Corresponds to a primary allocation that is equal to 100 percent of the estimated size of the file to be sorted, and to a secondary allocation of 33 percent.
The default value is PARTIAL.
The corresponding SYSIN keyword is WRKSPACE in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.
This parameter is also referred to as VUX006/WRKSPACE.

Sort program that Db2 HPU uses to sort data (VUX037/SORTUTIL)
This parameter is optional. It allows you to specify which sort program to use when Db2 HPU needs to perform a sort operation outside of Db2.
Specify one of the following values:

**SORT**
The on-site sort program is used. The load module name is DFSORT.

**DB2SORT**
The IBM Db2 Sort for z/OS utility (Db2 Sort) is used. Db2 Sort can improve the performance of utility sort processing, especially in environments with large volumes of data, large table spaces, large indexes, or batch window constraints. These environments require a more sophisticated sorting approach than the approach that is used by tools that are used for general sorting purposes.
The load module name is DB2SORT. The relevant entry point load module of Db2 Sort must be installed in the system libraries. Specifying DB2SORT does not affect the Db2 settings. Therefore, if you want the sorts performed by Db2 (for unsupported SELECT statements or when DB2 FORCE is specified), you must change the appropriate Db2 settings.

**Tip:** Specify DB2SORT when you need to perform several sort operations in parallel.
For more information about Db2 Sort, see the *Db2 Sort for z/OS User’s Guide*.
The default value is SORT.
The corresponding SYSIN keyword is SORTUTIL in the GLOBAL OPTIONS block.
This parameter is also referred to as VUX037/SORTUTIL.
Customization templates

Tools Customizer uses the following templates to generate the customization jobs for Db2 HPU.

Prerequisites

The typical installation and use of Db2 HPU assumes that the SCEERUN and SCEERUN2 Language Environment libraries are available through the linklist. If this condition is not fulfilled in your environment, the installation job generated for the INZCHECK template will fail. To avoid this, consider adding these libraries to the linklist or, if this cannot be done, consider adding a JOBLIB control card in the user jobcards that can be specified in Tools Customizer in the "User Job Card Settings for Customization Jobs " field accessible through the "0. User settings for Tools Customizer" option.

The following table lists and describes the templates:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INZTVAR</td>
<td>Generates the INZUTIL member containing the non Db2-dependent settings</td>
</tr>
<tr>
<td>INZPARM</td>
<td>Defines the dsname of the Db2 HPU PARMLIB to the load modules</td>
</tr>
<tr>
<td>INZHPUCL</td>
<td>Generates the members required to run the Db2 HPU ISPF interface</td>
</tr>
<tr>
<td>INZLAUNC</td>
<td>Generates the clist to be run to add Db2 HPU to Db2 Tools Launchpad</td>
</tr>
<tr>
<td>INZADTOO</td>
<td>Generates a member with instructions for adding Db2 HPU into Db2 Admin</td>
</tr>
<tr>
<td>INZBIND</td>
<td>Binds the plans and packages that are used by Db2 HPU and grants access</td>
</tr>
<tr>
<td>INZEXEUE</td>
<td>Generates a sample job for running Db2 HPU against a given subsystem</td>
</tr>
<tr>
<td>INZZSCOD</td>
<td>Defines the list of abend codes trapped for dump</td>
</tr>
<tr>
<td>INZCHECK</td>
<td>Generates members with the Db2-dependent settings and runs Db2 HPU IVP</td>
</tr>
<tr>
<td>INZFFREE</td>
<td>Frees the Db2 HPU plan for a Db2 subsystem or a data sharing group</td>
</tr>
</tbody>
</table>

Locating a parameter in Tools Customizer panels

About this task

You may need to set a single parameter as instructed to by your IBM support or by the explanation of a Db2 HPU feature or message. The following sections explain how parameters are organized and gives information to help you locate a given parameter in the Tools Customizer panels.

One panel per category

The parameters are accessible through several Tools Customizer's panels. Each panel is dedicated to a specific category of parameters as described by the list below.

- **Product Parameters: Db2 HPU**: to set or browse parameters applicable to any LPAR or Db2 subsystem or group attach (See “<<< Product parameters” in the figure below).
• **LPAR parameters: Db2 HPU**: to set or browse parameters applicable to the local LPAR. See "<<<LPAR parameters" in the figure below).

• **DB2 Parameters: Db2 HPU**: to set or browse parameters applicable to a given Db2 subsystem or group attach. See "<<< Db2 parameters" in the figure below).

**The Customizer Workplace; Db2 HPU panel**

<table>
<thead>
<tr>
<th>TCUSTMR</th>
<th>Customizer Workplace: DB2 HPU</th>
<th>Row 1 to 2 of 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>&gt;&gt;&gt;&gt;</td>
<td>Scroll &gt;&gt;&gt;&gt; CSR</td>
</tr>
</tbody>
</table>

Enter a line command under the Cmd column for each Db2 entry and press Enter.

Commands: ASSOCIATE DISCOVER GENERATEALL JOBLIST

Product to Customize

Product metadata library . : HINZ510.FLB.SINZDENU > LPAR . : S0W1
Product name . . . . . . . : IBM DB2 High Performance > Version : 5.1.0

Customization lib MZLFLB.TCZ.5S0W1$.INZ510.FLB2

Vol INFO5

Product and LPAR Parameter Status

Line commands: E - Edit B - Browse

- Product parameters . : Ready to Customize <<< Product parameters
- LPAR parameters . . : Ready to Customize <<< LPAR parameters

Associated Db2 Entries and Parameter Status

Line commands: G - Generate jobs E - Edit B - Browse C - Copy R - Remove

Cmd SSID GrpAttch Lvl Mode User ID Date Status Message

- DZBA -- 111 NFM MZLFLB 2019/05/20 Ready to Customize <<< Db2 parameters
- DB0A -- 111 NFM MZLFLB 2019/05/20 Ready to Customize <<< Db2 parameters

**Tools Customizer panels and sections**

On each Tools Customizer’s panel, the parameters are gathered under sections. The bullet list is organized the following way:

- **Panel**
  - **Section**
    - **Parameter**

The parameter layout is as follows:

- **Db2 parameters**
  - (header)
    - N/A/N/A >DB2 subsystem ID "
    - N/A/N/A >Group attach name
  - General Db2 Information - common
    - N/A/N/A >Mode"
    - N/A/N/A >Level number
    - N/A/N/A >Mixed mode
  - DB2 Libraries - common
    - N/A/N/A >Load library"
    - N/A/N/A >Run library
    - N/A/N/A >Exit library
  - Utilities - Common
    - N/A/N/A >SYSAFF for DB2 utilities"
    - N/A/N/A >Plan name for the DSNTIAD utility
  - Db2 HPU DB2 parameters
    - VUM012/PLANOWN >Owner of the plan created for Db2 HPU "
    - VUX011/________ >Grant on the plan created for Db2 HPU
    - VUM031/QUITUSR >User who execute DB2 processes on the table space
- VUM032/ACTLGUSR >User who issues SELECT by using dynamic SQL
- VUM033/OPTTRGPR >Partition range filter optimization
- VUM034/EXPLNTB >DELETE or KEEP rows inserted in EXPLAIN tables
- VUM035/EXPLNUSR >User who issues EXPLAIN PLAN statements
- VUM036/APPLCOMP >Specifies the compatibility level for SQL
- VUM028/DISPLUSR >User who runs the DISPLAY command
- VUU062/ULACCTRL >Unload controlled access table in native mode
- VUX044/CLIPLAN >Db2 Plan name of the CLI
- VUM037/CRYPTUSR >User who reads encrypted VSAM data set

- LPAR parameters
  - Other parameters - Common
    - N/A/N/A >Unit name for batch work data sets"
  - HPU Library Dsnames
    - VIZ003/SINZLOAD >Load modules library"
    - VIZ004/SINZLINK >APF load modules library
    - VIZ006/SINZSAMP >Samples library
    - VIZ007/SINZPARM >PARMLIB library
    - VIZ012/SINZDBRM >Product's DBRM library
    - VIZ014/SINZCLST >REXX EXEC library
    - VIZ013/SINZLLIB >ISPF load modules library
    - VIZ015/SINZMLIB >ISPF messages library
    - VIZ016/SINZPANL >ISPF panels library
    - VIZ017/SINZSLIB >ISPF skeletons library
    - VIZ018/SINZTLIB >ISPF tables library
  - HPU File Management parameters
    - VZM005/INFVSMIG >Volume name of migrated objects "
    - VZM010/TAPEUNIT >Numbers of devices for tape unit/storage class
    - VUX010/LIMUNIT >Maximum number of disk units to allocate a work file
    - VUX019/WRKMXPR >Maximum size for primary allocation of a work data set
    - VUX033/TMPLDSCB >DSCB model for allocation of GDS
    - VUM018/WRKVOL >Volumes for allocation of temporary data sets
    - VUA007/WRKTUNIT >Tape unit where the work data sets must be allocated
    - VUX016/WRKUNTSW >Maximum size for work data set on DASD
    - VUX017/MAXTUNIT >Maximum number of unit for tape temporary data set
    - VUX032/WRKTVCNT >Number of volumes for temporary data set on tape
    - VZM006/SCUNMOD >Unicode Conversion Services load library
  - Db2 HPU Tuning Parameters
    - VUX004/LOWMEM >Minimum memory size for each sort process "
    - VUX007/WAITUNIT >Wait unit in seconds/100
    - VUX008/WAITQTY >Number of wait periods before issuing a WTOR
    - VUX009/WAITQTYM >Wait time for the WTOR reply
    - VUX025/PARALLEL >Maximum degree of parallelism for LDS reading
    - VUX030/UTLPARAL >Maximum degree of parallelism for unload tasks
- VUX040/UBUFSIZE >Size of work areas buffers
- VUX041/UBUFNO >Number of buffers of work areas
- VUU041/ULOCUSIZE >Size of the buffer used to retrieve LOB data
- VUX039/ZIIP »zIIP usage
- VUX042/ZIPPMXPL >Maximum parallelism on zIIP for LDS
- VUX043/ZIPPMXPU >Maximum parallelism on zIIP for unload tasks
- VUU080/MXOPNLLD >Number of simultaneously opened LOB LDS

- HPU Sort parameters
  - VUX005/MAXSORT >Maximum number of active sorts processes
  - VUX006/WRKSPACE >Allocation distribution for the sort input file
  - VUX037/SORTUTIL >Sort program that Db2 HPU uses to sort data

- Product parameters
  - VUM011/PLANOBJT >Application plan for Db2 HPU
  - VUM030/COLLOBJT >Package collection for Db2 HPU
  - VUU079/EXITHLL >Management of exit written in C or COBOL.
  - VUM022/QSBUFNO >BUFNO for sequential QSAM
  - VUM023/VSBUFND >BUFND for sequential VSAM
  - VUM029/DFSIGDCB >Override DCB DB2, HPU parameters
  - VUM029/DFSIGDCB >Override DCB, UNLOAD PLUS syntax
  - VUM029/DFSIGDCB >Override DCB, FAST UNLOAD syntax
  - VUU011/ULSEDB2 >Use DB2 to process unsupported SELECT statements
  - VUU012/ULLOCK >Lock the tables in the table space
  - VUU074/ULLOCKTE >Specify the technique used to lock the unloaded object
  - VUU013/ULQSCE >Quiesce the table space
  - VUU014/ULNULL >NULL indicator, format USER
  - VUU014/ULNULL >NULL indicator, format DSNTIAUL
  - VUU014/ULNULL >NULL indicator, format VARIABLE
  - VUU014/ULNULL >NULL indicator, format EXTERNAL
  - VUU015/ULDATE >DATE, format USER
  - VUU015/ULDATE >DATE, format DSNTIAUL
  - VUU015/ULDATE >DATE, format DELIMITED
  - VUU015/ULDATE >DATE, format VARIABLE
  - VUU015/ULDATE >DATE, format EXTERNAL
  - VUU016/ULTIME >TIME, format USER
  - VUU016/ULTIME >TIME, format DSNTIAUL
  - VUU016/ULTIME >TIME, format DELIMITED
  - VUU016/ULTIME >TIME, format VARIABLE
  - VUU016/ULTIME >TIME, format EXTERNAL
  - VUU017/ULTMSTP >TIMESTAMP, format USER
  - VUU017/ULTMSTP >TIMESTAMP, format DSNTIAUL
  - VUU017/ULTMSTP >TIMESTAMP, format DELIMITED
  - VUU017/ULTMSTP >TIMESTAMP, format VARIABLE
  - VUU017/ULTMSTP >TIMESTAMP, format EXTERNAL
- VUU018/ULPIC >Decimal picture, format USER
- VUU018/ULPIC >Decimal picture, format DSNTIAUL
- VUU018/ULPIC >Decimal picture, format DELIMITED
- VUU018/ULPIC >Decimal picture, format VARIABLE
- VUU018/ULPIC >Decimal picture, format EXTERNAL
- VUU019/ULOPTLDT >Options for global LOAD statement
- VUU020/ULOPTLDP >Options of partition LOAD statement
- VUU070/ULOPTLDI >Options of table LOAD statement
- VUU071/LOSSCHAR >Manage character loss, Db2 HPU parameters
- VUU071/LOSSCHAR >Manage character loss, UNLOAD PLUS syntax
- VUU071/LOSSCHAR >Manage character loss, FAST UNLOAD syntax
- VUU072/LOSSNUM >Manage numeric loss, Db2 HPU parameters
- VUU072/LOSSNUM >Manage numeric loss, UNLOAD PLUS syntax
- VUU072/LOSSNUM >Manage numeric loss, FAST UNLOAD syntax
- VUU073/MAXERR >Error message limit, Db2 HPU parameters
- VUU073/MAXERR >Error message limit, UNLOAD PLUS syntax
- VUU073/MAXERR >Error message limit, FAST UNLOAD syntax
- VUU021/ULDEGREE >Degree of parallel processing
- VUU022/NULLPOS >Position for NULL indicator, format USER
- VUU022/NULLPOS >Position for NULL indicator, format DSNTIAUL
- VUU022/NULLPOS >Position for NULL indicator, format VARIABLE
- VUU022/NULLPOS >Position for NULL indicator, format EXTERNAL
- VUU075/NULLPAD >Fill NULL with padding character, format USER
- VUU075/NULLPAD >Fill NULL with padding character, format DSNTIAUL
- VUU075/NULLPAD >Fill NULL with padding character, format VARIABLE
- VUU075/NULLPAD >Fill NULL with padding character, format EXTERNAL
- VUU023/UNLSCHEM >Default scheme for UNLOAD TABLESPACE
- VUU024/UNLZLRC >Return code if zero row is unloaded
- VUU024/UNLZLRC >Return code if zero row is unloaded
- VUU024/UNLZLRC >Return code if no object is returned by LISTDEF
- VUU024/UNLZLRC >Return code if SELECT statement retrieves no row
- VUU024/UNLZLRC >MAXRC alteration return code
- VUU028/ULQSCEBH >Quiesce process
- VUU029/UNLDDDN >Use the same DDN for UNLDDN, OUTDDN, and LOADDDN
- VUU030/ULOPTNS >Db2 HPU additional features
- VUU031/DTDELIM >DATE/TIME delimiter option
- VUU032/ULMASK >Display format for numeric values, format USER
- VUU032/ULMASK >Display format for numeric values, format DSNTIAUL
- VUU032/ULMASK >Display format for numeric values, format DELIMITED
- VUU032/ULMASK >Display format for numeric values, format VARIABLE
- VUU032/ULMASK >Display format for numeric values, format EXTERNAL
- VUU033/ULCHKCPY >Check image copy before unloading
- VUU076/ULICTYPE >Image copies selected for a partitioned table space
- VUU077/ULCENT >Year to set the century for DECIMAL to DATE conversion
- VUU078/ULNULFL >Output field for NULL values in DELIMITED format
- VUU081/UNLLDTBB >LISTDEF behavior for table selection
- VUU082/ULRLDRC >Action if fully compliant LOADDDN cannot be generated
- VUU083/DISPPARM >Request the display of implicit parameters
- VUU084/ULLOBBHV >Processing when LOB/XML not retrieved
- VUU085/ULLOBBHM >Maximum number of issued messages for LOB/XML errors
- VUU034/ULLLIKE >Option to modify the like behavior in SQL statement
- VUU035/ULROWSET >Number of rows retrieved by an SQL FETCH function
- VUU036/GBLPARAL >Enforce partition parall. unloading into a single file
- VUU037/ULSIGZ >Sign values for zoned-decimal numeric values
- VUU038/ULSIGDTP >Positive sign for date-time packed values
- VUU039/UNLLDER >Db2 HPU response when a LOADDDN cannot be generated
- VUU040/ULHIDDEN >Unload HIDDEN Column by select *
- VUU042/DTNULDLM >NULL DATE/TIME delimiter option
- VUU044/SUBTKSOF >Support parallelism for unloading several partitions
- VUU045/ULFORMAT >Default output format for a logical unload
- VUU046/ULAUTAG >Size of the autotag in the output file
- VUU047/RETRYMOD >Wait/retry function when resources are unavailable
- VUU048/RETRYW >Wait time between tries to access unavailable resource
- VUU049/RETRYNB >Number of retries when resources are unavailable
- VUU050/ULFRQMSG >Frequency to display information messages
- VUU050/ULFRQMSG >Additional parameter for information messages
- VUU051/PROCMSG >The location where Db2 HPU writes information messages
- VUU052/SELMNG >Location of messages issued for each SELECT statement
- VUU053/OBIDRPT >Generate a list of all OBIDs in the FIC
- VUU055/DATEDEL >Delimiter for external date, format USER
- VUU055/DATEDEL >Delimiter for external date, format DSNTIAUL
- VUU055/DATEDEL >Delimiter for external date, format DELIMITED
- VUU055/DATEDEL >Delimiter for external date, format VARIABLE
- VUU055/DATEDEL >Delimiter for external date, format EXTERNAL
- VUU056/TIMEDEL >;Delimiter in external time, format USER
- VUU056/TIMEDEL >Delimeter in external time, format DSNTIAUL
- VUU056/TIMEDEL >Delimeter in external time, format DELIMITED
- VUU056/TIMEDEL >Delimeter in external time, format VARIABLE
- VUU056/TIMEDEL >Delimeter in external time, format EXTERNAL
- VUU057/OPALLFMT >Options apply to all formats
- VUU058/ULIDENT >LOAD technique for identity column
- VUU065/ULRTTST >LOAD technique for ROW-TRANSACTION-TIMESTAMP column
- VUU066/ULRTSID >LOAD technique for ROW-TRANSACTION-START-ID col
- VUU059/ULTRIM >Trim the trailing blanks, format USER
- VUU059/ULTRIM >Trim the trailing blanks, format DSNTIAUL
- VUU059/ULTRIM >Trim the trailing blanks, format DELIMITED
- VUU059/ULTRIM >Trim the trailing blanks, format VARIABLE
- VUU059/ULTRIM >Trim the trailing blanks, format EXTERNAL
- VUU060/ULMAXPAR >Maximum number of partitions processed in one unload
- VUU061/ACCPREP >Concurrent access resolution for prepared statement
- VUU063/ULOPTLX >Technical parameter to unload XML columns
- VUU064/STRNGCST >String constant is unloaded as CHAR/VARCHAR data types
- VUU067/ULCNVTRE >Option to specify conversion truncation policy
- VUU068/ULUNIEXP >Expansion ratio for conversion to UTF8 CCSID
- VUU069/SQLSWIRC >Return code when the utility has switched to sql mode
- VUX003/SIZE >Default size of the input file or object for Db2 HPU
- VUX018/LDSERRLM >Maximum number of messages for row structure errors
- VUX023/PGDFIN >Percentage of displaced pages in inline FIC
- VUX031/DB2PARAL >Maximum degree of parallelism for SELECT
- VUX036/RTSESTIM >Use real-time statistics tables for size estimation
- VUX038/CHECKINT >Level of data integrity control of input data
- VZM007/SCUNTSO >Unicode Conversion Services technique search order
- VZM008/SCUNSUB >Unicode Conversion Services character substitution
- VZM011/TAPERPT >Generate a tape usage report in SYSPRINT
- VUU025/________ >High-level qualifier for the DB2 Admin data sets
- VUU027/________ >Library that contains the ADBDMTI EXEC
- VUU026/________ >Library that contains DB2 Admin commands tables
- VUU027/________ >Library that contains the ADBDMTI EXEC
- VZM009/________ >List of system codes for which no dump is produced

- Db2 parameters
  - (header)
    - N/A/N/A >DB2 subsystem ID "
    - N/A/N/A >Group attach name
  - General Db2 Information - common
    - N/A/N/A >Mode"
    - N/A/N/A >Level number
    - N/A/N/A >Mixed mode
  - DB2 Libraries - common
    - N/A/N/A >Load library"
    - N/A/N/A >Run library
    - N/A/N/A >Exit library
  - Utilities - Common
    - N/A/N/A >SYSAFF for DB2 utilities"
    - N/A/N/A >Plan name for the DSNTIAD utility
  - Db2 HPU DB2 parameters
    - VUM012/PLANOWN >Owner of the plan created for Db2 HPU "
    - VUX011/________ >Grant on the plan created for Db2 HPU
    - VUM031/QUITUSR >User who execute DB2 processes on the table space
- VUM032/ACTLGUSR >User who issues SELECT by using dynamic SQL
- VUM033/OPTRGPRT >Partition range filter optimization
- VUM034/EXPLNTB >DELETE or KEEP rows inserted in EXPLAIN tables
- VUM035/EXPLNUSR >User who issues EXPLAIN PLAN statements
- VUM036/APPLCOMP >Specifies the compatibility level for SQL
- VUM028/DISPLUSR >User who runs the DISPLAY command
- VUU062/ULACCTRL >Unload controlled access table in native mode
- VUX044/CLIPLAN >Db2 Plan name of the CLI
- VUM037/CRYPTUSR >User who reads encrypted VSAM data set

• LPAR parameters
  - Other parameters - Common
    - N/A/N/A >Unit name for batch work data sets"
    - HPU Library Dsnames
      - VIZ003/SINZLOAD >Load modules library"
      - VIZ004/SINZLINK >APF load modules library
      - VIZ006/SINZSAMP >Samples library
      - VIZ007/SINZPARM >PARMLIB library
      - VIZ012/SINZDBRM >Product's DBRM library
      - VIZ014/SINZCLST >REXX EXEC library
      - VIZ013/SINZLLIB >ISPF load modules library
      - VIZ015/SINZMLIB >ISPF messages library
      - VIZ016/SINZPANL >ISPF panels library
      - VIZ017/SINZSLIB >ISPF skeletons library
      - VIZ018/SINZTLIB >ISPF tables library
  - HPU File Management parameters
    - VZM005/INFVSMIG >Volume name of migrated objects "
    - VZM010/TAPEUNIT >Numbers of devices for tape unit/storage class
    - VUX010/LIMUNIT >Maximum number of disk units to allocate a work file
    - VUX019/WRKMXPR >Maximum size for primary allocation of a work data set
    - VUX033/TMPLDSCB >DSCB model for allocation of GDS
    - VUM018/WRKVLQ >Volumes for allocation of temporary data sets
    - VUA007/WRKTUNIT >Tape unit where the work data sets must be allocated
    - VUX016/WRKUNITW >Maximum size for work data set on DASD
    - VUX017/MAXTUNIT >Maximum number of unit for tape temporary data set
    - VUX032/WRKTVCNT >Number of volumes for temporary data set on tape
    - VZM006/SCUNMOD >Unicode Conversion Services load library
  - Db2 HPU Tuning Parameters
    - VUX004/LOWMEM >Minimum memory size for each sort process "
    - VUX007/WAITUNIT >Wait unit in seconds/100
    - VUX008/WAITQTY >Number of wait periods before issuing a WTOR
    - VUX009/WAITQTYM >Wait time for the WTOR reply
    - VUX025/PARALLEL >Maximum degree of parallelism for LDS reading
    - VUX030/UTLPARAL >Maximum degree of parallelism for unload tasks

Chapter 9. Reference 559
- VUX040/UBUFSIZE >Size of work areas buffers
- VUX041/UBUFNO >Number of buffers of work areas
- VUU041/ULOCFSIZE >Size of the buffer used to retrieve LOB data
- VUX039/ZIIP >zIIP usage
- VUX042/ZIIPMXPL >Maximum parallelism on zIIP for LDS
- VUX043/ZIIPMXPU >Maximum parallelism on zIIP for unload tasks
- VUU080/MXOPNLLD >Number of simultaneously opened LOB LDS

- HPU Sort parameters
  - VUX005/MAXSORT >Maximum number of active sorts processes
  - VUX006/WRKSPACE >Allocation distribution for the sort input file
  - VUX037/SORTUTIL >Sort program that DB2 HPU uses to sort data

- Product parameters
  - Common parameters
    - VUM011/PLANOBJT >Application plan for Db2 HPU
    - VUM030/COLOBJT >Package collection for Db2 HPU
    - VUU079/EXITHLL >Management of exit written in C or COBOL.
    - VUM022/QSBUFNO >BUFNO for sequential QSAM
    - VUM023/VSBUFND >BUFND for sequential VSAM
    - VUM029/DFSGDCB >Override DCB, DB2, HPU parameters
    - VUM029/DFSGDCB >Override DCB, UNLOAD PLUS syntax
    - VUM029/DFSGDCB >Override DCB, FAST UNLOAD syntax
    - VUU011/ULSED2B >Use DB2 to process unsupported SELECT statements
    - VUU012/ULLOCK >Lock the tables in the table space
    - VUU074/ULLOCKTE >Specify the technique used to lock the unloaded object
    - VUU013/ULQSCE >Quiesce the table space
    - VUU014/ULNULL >NULL indicator, format USER
    - VUU014/ULNULL >NULL indicator, format DSNTIAUL
    - VUU014/ULNULL >NULL indicator, format VARIABLE
    - VUU014/ULNULL >NULL indicator, format EXTERNAL
    - VUU015/ULDATE >DATE, format USER
    - VUU015/ULDATE >DATE, format DSNTIAUL
    - VUU015/ULDATE >DATE, format DELIMITED
    - VUU015/ULDATE >DATE, format VARIABLE
    - VUU015/ULDATE >DATE, format EXTERNAL
    - VUU016/ULTIME >TIME, format USER
    - VUU016/ULTIME >TIME, format DSNTIAUL
    - VUU016/ULTIME >TIME, format DELIMITED
    - VUU016/ULTIME >TIME, format VARIABLE
    - VUU016/ULTIME >TIME, format EXTERNAL
    - VUU017/ULTMSTP >TIMESTAMP, format USER
    - VUU017/ULTMSTP >TIMESTAMP, format DSNTIAUL
    - VUU017/ULTMSTP >TIMESTAMP, format DELIMITED
    - VUU017/ULTMSTP >TIMESTAMP, format VARIABLE
- VUU017/ULTMSTP >TIMESTAMP, format EXTERNAL
- VUU018/ULPIC >Decimal picture, format USER
- VUU018/ULPIC >Decimal picture, format DSNTIAUL
- VUU018/ULPIC >Decimal picture, format DELIMITED
- VUU018/ULPIC >Decimal picture, format VARIABLE
- VUU018/ULPIC >Decimal picture, format EXTERNAL
- VUU019/ULOPTLDT >Options for global LOAD statement
- VUU020/ULOPTLDP >Options of partition LOAD statement
- VUU070/ULOPTLDB >Options of table LOAD statement
- VUU071/LOSSCHAR >Manage character loss, Db2 HPU parameters
- VUU071/LOSSCHAR >Manage character loss, UNLOAD PLUS syntax
- VUU071/LOSSCHAR >Manage character loss, FAST UNLOAD syntax
- VUU072/LOSSNUM >Manage numeric loss, Db2 HPU parameters
- VUU072/LOSSNUM >Manage numeric loss, UNLOAD PLUS syntax
- VUU072/LOSSNUM >Manage numeric loss, FAST UNLOAD syntax
- VUU073/MAXERR >Error message limit, Db2 HPU parameters
- VUU073/MAXERR >Error message limit, UNLOAD PLUS syntax
- VUU073/MAXERR >Error message limit, FAST UNLOAD syntax
- VUU021/ULDEGREE >Degree of parallel processing
- VUU022/NULLOPOS >Position for NULL indicator, format USER
- VUU022/NULLOPOS >Position for NULL indicator, format DSNTIAUL
- VUU022/NULLOPOS >Position for NULL indicator, format VARIABLE
- VUU022/NULLOPOS >Position for NULL indicator, format EXTERNAL
- VUU075/NULLPAD >Fill NULL with padding character, format USER
- VUU075/NULLPAD >Fill NULL with padding character, format DSNTIAUL
- VUU075/NULLPAD >Fill NULL with padding character, format VARIABLE
- VUU075/NULLPAD >Fill NULL with padding character, format EXTERNAL
- VUU023/UNLSCHEM >Default scheme for UNLOAD TABLESPACE
- VUU024/UNLZLRC >Return code if zero row is unloaded
- VUU024/UNLZLRC >Return code if an object is missing
- VUU024/UNLZLRC >Return code if no object is returned by LISTDEF
- VUU024/UNLZLRC >Return code if SELECT statement retrieves no row
- VUU024/UNLZLRC >MAXRC alteration return code
- VUU028/ULQSCEBH >Quiesce process
- VUU029/UNLDDDN >Use the same DDN for UNLDDN, OUTF邓, and LOADDDN
- VUU030/ULOPTNS >Db2 HPU additional features
- VUU031/DTDELIM >DATE/TIME delimiter option
- VUU032/ULMASK >Display format for numeric values, format USER
- VUU032/ULMASK >Display format for numeric values, format DSNTIAUL
- VUU032/ULMASK >Display format for numeric values, format DELIMITED
- VUU032/ULMASK >Display format for numeric values, format VARIABLE
- VUU032/ULMASK >Display format for numeric values, format EXTERNAL
- VUU033/ULCHKCPY >Check image copy before unloading
- VUU076/ULICTYPE >Image copies selected for a partitioned table space
- VUU077/ULCENT >Year to set the century for DECIMAL to DATE conversion
- VUU078/ULNULLFL >Output field for NULL values in DELIMITED format
- VUU081/UNLLDTBB >LISTDEF behavior for table selection
- VUU082/ULRLDRC >Action if fully compliant LOADDDN cannot be generated
- VUU083/DISPPARM >Request the display of implicit parameters
- VUU084/ULLOBBHV >Processing when LOB/XML not retrieved
- VUU085/ULLOBBHM >Maximum number of issued messages for LOB/XML errors
- VUU034/ULLIKE >Option to modify the like behavior in SQL statement
- VUU035/ULROWSET >Number of rows retrieved by an SQL FETCH function
- VUU036/GBLPARAL >Enforce partition parall. unloading into a single file
- VUU037/ULSIGZ >Sign values for zoned-decimal numeric values
- VUU038/ULSIGDTP >Positive sign for date-time packed values
- VUU039/UNLLDER >Db2 HPU response when a LOADDDN cannot be generated
- VUU040/ULTR32K >Truncate variable records exceeding 32766
- VUU042/ULHIDDEN >Unload HIDDEN Column by select *
- VUU043/DTNULDL >NULL DATE/TIME delimiter option
- VUU044/SUBTKSOF >Support parallelism for unloading several partitions
- VUU045/ULFORMAT >Default output format for a logical unload
- VUU046/ULAUTAG >Size of the autotag in the output file
- VUU047/RETRYMOD >Wait/retry function when resources are unavailable
- VUU048/RETRYW >Wait time between tries to access unavailable resource
- VUU049/RETRYNB >Number of retries when resources are unavailable
- VUU050/ULFRQMSG >Frequency to display information messages
- VUU050/ULFRQMSG >Additional parameter for information messages
- VUU051/PROCMSG >The location where Db2 HPU writes information messages
- VUU052/SELMMSG >Location of messages issued for each SELECT statement
- VUU053/OBIDRPT >Generate a list of all OBIDs in the FIC
- VUU055/DATEDEL >Delimiter for external date, format USER
- VUU055/DATEDEL >Delimiter for external date, format DSNTIAUL
- VUU055/DATEDEL >Delimiter for external date, format DELIMITED
- VUU055/DATEDEL >Delimiter for external date, format VARIABLE
- VUU055/DATEDEL >Delimiter for external date, format EXTERNAL
- VUU056/TIMEDEL >Delimiter in external time, format USER
- VUU056/TIMEDEL >Delimiter in external time, format DSNTIAUL
- VUU056/TIMEDEL >Delimiter in external time, format DELIMITED
- VUU056/TIMEDEL >Delimiter in external time, format VARIABLE
- VUU056/TIMEDEL >Delimiter in external time, format EXTERNAL
- VUU057/OPALLFMT >Options apply to all formats
- VUU058/ULIDENT >LOAD technique for identity column
- VUU065/ULRTTST >LOAD technique for ROW-TRANSACTION-TIMESTAMP column
- VUU067/ULRTSID >LOAD technique for ROW-TRANSACTION-START-ID col
- VUU059/ULTRIM >Trim the trailing blanks, format USER
- VUU059/ULTRIM >Trim the trailing blanks, format DSNTIAUL
- VUU059/ULTRIM >Trim the trailing blanks, format DELIMITED
- VUU059/ULTRIM >Trim the trailing blanks, format VARIABLE
- VUU059/ULTRIM >Trim the trailing blanks, format EXTERNAL
- VUU060/ULMAXPAR >Maximum number of partitions processed in one unload
- VUU061/ACCPREP >Concurrent access resolution for prepared statement
- VUU063/ULOPTLX >Technical parameter to unload XML columns
- VUU064/STRINGCST >String constant is unloaded as CHAR/VARCHAR data types
- VUU067/ULCNVTRC >Option to specify conversion truncation policy
- VUU068/ULUNIEXP >Expansion ratio for conversion to UTF8 CCSID
- VUU069/SQLSWIRC >Return code when the utility has switched to sql mode
- VUX003/SIZE >Default size of the input file or object for Db2 HPU
- VUX018/LDSERRLM >Maximum number of messages for row structure errors
- VUX023/PGDFIN >Percentage of displaced pages in inline FIC
- VUX031/DB2PARAL >Maximum degree of parallelism for SELECT
- VUX036/RTSESTIM >Use real-time statistics tables for size estimation
- VUX038/CHECKINT >Level of data integrity control of input data
- VZM007/SCUNTSO >Unicode Conversion Services technique search order
- VZM008/SCUNSUB >Unicode Conversion Services character substitution
- VZM011/TAPERPT >Generate a tape usage report in SYSPRINT
- VUU025/________ >High-level qualifier for the DB2 Admin data sets
- VUU027/________ >Library that contains the ADBDMTI EXEC
- VUU026/________ >Library that contains DB2 Admin commands tables
- VUU027/________ >Library that contains the ADBDMTI EXEC
- VZM009/________ >List of system codes for which no dump is produced

**Parameters location**

The table below lists all the parameters used by Db2 HPU and for each parameters, provides the name of the Tools Customizer’s panel and the name of the Section in the panel where the parameter is located. The rank column displays the rank of the parameter within the panel described as n/N which mean the parameter is the n th parameter starting from the top of the panel that contains a total of N parameters.

<table>
<thead>
<tr>
<th>Key</th>
<th>Short name</th>
<th>Parameter label</th>
<th>Panel</th>
<th>Section</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>DB2 subsystem ID</td>
<td>Db2 parameters</td>
<td>(header)</td>
<td>1/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Exit library</td>
<td>Db2 parameters</td>
<td>DB2 Libraries - common</td>
<td>8/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Group attach name</td>
<td>Db2 parameters</td>
<td>(header)</td>
<td>2/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Level number</td>
<td>Db2 parameters</td>
<td>General Db2 Information - common</td>
<td>4/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Load library</td>
<td>Db2 parameters</td>
<td>DB2 Libraries - common</td>
<td>6/22</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>--------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Mixed mode</td>
<td>Db2 parameters</td>
<td>General Db2 Information - common</td>
<td>5/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Mode</td>
<td>Db2 parameters</td>
<td>General Db2 Information - common</td>
<td>3/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Plan name for the DSNTIAD utility</td>
<td>Db2 parameters</td>
<td>Utilities - Common</td>
<td>10/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Run library</td>
<td>Db2 parameters</td>
<td>DB2 Libraries - common</td>
<td>7/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>SYSAFF for DB2 utilities</td>
<td>Db2 parameters</td>
<td>Utilities - Common</td>
<td>9/22</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Unit name for batch work data sets</td>
<td>LPAR parameters</td>
<td>Other parameters - Common</td>
<td>1/39</td>
</tr>
<tr>
<td>VIZ003</td>
<td>SINZLOAD</td>
<td>Load modules library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>2/39</td>
</tr>
<tr>
<td>VIZ004</td>
<td>SINZLINK</td>
<td>APF load modules library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>3/39</td>
</tr>
<tr>
<td>VIZ006</td>
<td>SINZSAMP</td>
<td>Samples library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>4/39</td>
</tr>
<tr>
<td>VIZ007</td>
<td>SINZPARM</td>
<td>PARMLIB library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>5/39</td>
</tr>
<tr>
<td>VIZ012</td>
<td>SINZDBRM</td>
<td>Product's DBRM library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>6/39</td>
</tr>
<tr>
<td>VIZ013</td>
<td>SINZLLIB</td>
<td>ISPF load modules library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>8/39</td>
</tr>
<tr>
<td>VIZ014</td>
<td>SINZCLST</td>
<td>REXX EXEC library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>7/39</td>
</tr>
<tr>
<td>VIZ015</td>
<td>SINZMLIB</td>
<td>ISPF messages library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>9/39</td>
</tr>
<tr>
<td>VIZ016</td>
<td>SINZPANL</td>
<td>ISPF panels library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>10/39</td>
</tr>
<tr>
<td>VIZ017</td>
<td>SINZSLIB</td>
<td>ISPF skeletons library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>11/39</td>
</tr>
<tr>
<td>VIZ018</td>
<td>SINZTLIB</td>
<td>ISPF tables library</td>
<td>LPAR parameters</td>
<td>HPU Library Dnames</td>
<td>12/39</td>
</tr>
<tr>
<td>VUA007</td>
<td>WRKTUNIT</td>
<td>Tape unit where the work data sets must be allocated</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>19/39</td>
</tr>
<tr>
<td>VUM011</td>
<td>PLANOBJT</td>
<td>Application plan for Db2 HPU</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>1/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-----------------------------------------------------</td>
<td>---------------------</td>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>VUM012</td>
<td>PLANOWN</td>
<td>Owner of the plan created for Db2 HPU</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>11/22</td>
</tr>
<tr>
<td>VUM018</td>
<td>WRKVOL</td>
<td>Volumes for allocation of temporary data sets</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>18/39</td>
</tr>
<tr>
<td>VUM022</td>
<td>QSBUFNO</td>
<td>BUFNO for sequential QSAM</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>4/141</td>
</tr>
<tr>
<td>VUM023</td>
<td>VSBUFND</td>
<td>BUFND for sequential VSAM</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>5/141</td>
</tr>
<tr>
<td>VUM028</td>
<td>DISPLUSR</td>
<td>User who runs the DISPLAY command</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>19/22</td>
</tr>
<tr>
<td>VUM029</td>
<td>DFSIGDCB</td>
<td>Override DCB DB2, HPU parameters</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>6/141</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Override DCB, FAST UNLOAD syntax</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>8/141</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Override DCB, UNLOAD PLUS syntax</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>7/141</td>
</tr>
<tr>
<td>VUM030</td>
<td>COLLOBJT</td>
<td>Package collection for Db2 HPU</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>2/141</td>
</tr>
<tr>
<td>VUM031</td>
<td>QUITSSUSR</td>
<td>User who execute DB2 processes on the table space</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>13/22</td>
</tr>
<tr>
<td>VUM032</td>
<td>ACTLGUSR</td>
<td>User who issues SELECT by using dynamic SQL</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>14/22</td>
</tr>
<tr>
<td>VUM033</td>
<td>OPTGPRT</td>
<td>Partition range filter optimization</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>15/22</td>
</tr>
<tr>
<td>VUM034</td>
<td>EXPLNTB</td>
<td>DELETE or KEEP rows inserted in EXPLAIN tables</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>16/22</td>
</tr>
<tr>
<td>VUM035</td>
<td>EXPLNUSR</td>
<td>User who issues EXPLAIN PLAN statements</td>
<td>Db2 parameters</td>
<td>DB2 HPU DB2 parameters</td>
<td>17/22</td>
</tr>
<tr>
<td>VUM036</td>
<td>APPLCOMP</td>
<td>Specifies the compatibility level for SQL</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>18/22</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>------</td>
</tr>
<tr>
<td>VUM037</td>
<td>CRYPTUSR</td>
<td>User who reads encrypted VSAM data set</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>19/22</td>
</tr>
<tr>
<td>VUU011</td>
<td>ULSEDB2</td>
<td>Use DB2 to process unsupported SELECT statements</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>9/141</td>
</tr>
<tr>
<td>VUU012</td>
<td>ULOCK</td>
<td>Lock the tables in the table space</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>10/141</td>
</tr>
<tr>
<td>VUU013</td>
<td>ULQSCE</td>
<td>Quiesce the table space</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>12/141</td>
</tr>
<tr>
<td>VUU014</td>
<td>ULNULL</td>
<td>NULL indicator, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>14/141</td>
</tr>
<tr>
<td>VUU014</td>
<td>ULNULL</td>
<td>NULL indicator, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>16/141</td>
</tr>
<tr>
<td>VUU014</td>
<td>ULNULL</td>
<td>NULL indicator, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>13/141</td>
</tr>
<tr>
<td>VUU014</td>
<td>ULNULL</td>
<td>NULL indicator, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>15/141</td>
</tr>
<tr>
<td>VUU015</td>
<td>ULDATE</td>
<td>DATE, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>19/141</td>
</tr>
<tr>
<td>VUU015</td>
<td>ULDATE</td>
<td>DATE, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>18/141</td>
</tr>
<tr>
<td>VUU015</td>
<td>ULDATE</td>
<td>DATE, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>21/141</td>
</tr>
<tr>
<td>VUU015</td>
<td>ULDATE</td>
<td>DATE, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>17/141</td>
</tr>
<tr>
<td>VUU015</td>
<td>ULDATE</td>
<td>DATE, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>20/141</td>
</tr>
<tr>
<td>VUU016</td>
<td>UTIME</td>
<td>TIME, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>24/141</td>
</tr>
<tr>
<td>VUU016</td>
<td>UTIME</td>
<td>TIME, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>23/141</td>
</tr>
<tr>
<td>VUU016</td>
<td>UTIME</td>
<td>TIME, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>22/141</td>
</tr>
<tr>
<td>VUU016</td>
<td>UTIME</td>
<td>TIME, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>26/141</td>
</tr>
<tr>
<td>VUU016</td>
<td>UTIME</td>
<td>TIME, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>25/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>------------------------------------------</td>
<td>------------------------</td>
<td>--------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>VUU017</td>
<td>ULTSTP</td>
<td>TIMESTAMP, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>29/141</td>
</tr>
<tr>
<td>VUU017</td>
<td>ULTSTP</td>
<td>TIMESTAMP, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>28/141</td>
</tr>
<tr>
<td>VUU017</td>
<td>ULTSTP</td>
<td>TIMESTAMP, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>31/141</td>
</tr>
<tr>
<td>VUU017</td>
<td>ULTSTP</td>
<td>TIMESTAMP, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>27/141</td>
</tr>
<tr>
<td>VUU017</td>
<td>ULTSTP</td>
<td>TIMESTAMP, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>30/141</td>
</tr>
<tr>
<td>VUU018</td>
<td>ULPIC</td>
<td>Decimal picture, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>33/141</td>
</tr>
<tr>
<td>VUU018</td>
<td>ULPIC</td>
<td>Decimal picture, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>32/141</td>
</tr>
<tr>
<td>VUU018</td>
<td>ULPIC</td>
<td>Decimal picture, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>34/141</td>
</tr>
<tr>
<td>VUU018</td>
<td>ULPIC</td>
<td>Decimal picture, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>36/141</td>
</tr>
<tr>
<td>VUU018</td>
<td>ULPIC</td>
<td>Decimal picture, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>35/141</td>
</tr>
<tr>
<td>VUU019</td>
<td>ULOPTLDT</td>
<td>Options for global LOAD statement</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>37/141</td>
</tr>
<tr>
<td>VUU020</td>
<td>ULOPTLDP</td>
<td>Options of partition LOAD statement</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>38/141</td>
</tr>
<tr>
<td>VUU021</td>
<td>ULDEGREE</td>
<td>Degree of parallel processing</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>49/141</td>
</tr>
<tr>
<td>VUU022</td>
<td>NULLPOS</td>
<td>Position for NULL indicator, format DSNTIAU</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>51/141</td>
</tr>
<tr>
<td>VUU022</td>
<td>NULLPOS</td>
<td>Position for NULL indicator, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>53/141</td>
</tr>
<tr>
<td>VUU022</td>
<td>NULLPOS</td>
<td>Position for NULL indicator, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>50/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------------------------------------------------------</td>
<td>------------------------------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>VUU022</td>
<td>NULLPOS</td>
<td>Position for NULL indicator, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>52/141</td>
</tr>
<tr>
<td>VUU023</td>
<td>UNLSCHEM</td>
<td>Default scheme for UNLOAD TABLESPACE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>58/141</td>
</tr>
<tr>
<td>VUU024</td>
<td>UNLZLRC</td>
<td>Return code if an object is missing</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>60/141</td>
</tr>
<tr>
<td>VUU024</td>
<td>UNLZLRC</td>
<td>Return code if no object is returned by LISTDEF</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>61/141</td>
</tr>
<tr>
<td>VUU024</td>
<td>UNLZLRC</td>
<td>Return code if zero row is unloaded</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>59/141</td>
</tr>
<tr>
<td>VUU024</td>
<td>UNLZLRC</td>
<td>MAXRC alteration return code</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>63/141</td>
</tr>
<tr>
<td>VUU024</td>
<td>UNLZLRC</td>
<td>Return code if SELECT statement retrieves no row</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>62/141</td>
</tr>
<tr>
<td>VUU025</td>
<td>__________</td>
<td>High-level qualifier for the DB2 Admin data sets</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>137/141</td>
</tr>
<tr>
<td>VUU026</td>
<td>__________</td>
<td>Library that contains DB2 Admin commands tables</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>139/141</td>
</tr>
<tr>
<td>VUU027</td>
<td>__________</td>
<td>Library that contains the ADBDMTI EXEC</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>138/141</td>
</tr>
<tr>
<td>VUU027</td>
<td>__________</td>
<td>Library that contains the ADBDMTI EXEC</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>140/141</td>
</tr>
<tr>
<td>VUU028</td>
<td>ULQSCEBH</td>
<td>Quiesce process</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>64/141</td>
</tr>
<tr>
<td>VUU029</td>
<td>UNLDDDN</td>
<td>Use the same DDN for UNLDDN, OUTDDN, and LOADDDN</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>65/141</td>
</tr>
<tr>
<td>VUU030</td>
<td>ULOPTNS</td>
<td>Db2 HPU additional features</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>66/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------</td>
<td>------------------------</td>
<td>---------------------</td>
<td>--------</td>
</tr>
<tr>
<td>VUU031</td>
<td>DTDELIM</td>
<td>DATE/TIME delimiter option</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>64/141</td>
</tr>
<tr>
<td>VUU032</td>
<td>ULMASK</td>
<td>Display format for numeric values, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>69/141</td>
</tr>
<tr>
<td>VUU032</td>
<td>ULMASK</td>
<td>Display format for numeric values, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>68/141</td>
</tr>
<tr>
<td>VUU032</td>
<td>ULMASK</td>
<td>Display format for numeric values, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>70/141</td>
</tr>
<tr>
<td>VUU032</td>
<td>ULMASK</td>
<td>Display format for numeric values, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>72/141</td>
</tr>
<tr>
<td>VUU032</td>
<td>ULMASK</td>
<td>Display format for numeric values, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>71/141</td>
</tr>
<tr>
<td>VUU033</td>
<td>ULCHKCPY</td>
<td>Check image copy before unloading</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>73/141</td>
</tr>
<tr>
<td>VUU034</td>
<td>ULLIKE</td>
<td>Option to modify the like behavior in SQL statement</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>82/141</td>
</tr>
<tr>
<td>VUU035</td>
<td>ULRWSET</td>
<td>Number of rows retrieved by an SQL FETCH function</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>83/141</td>
</tr>
<tr>
<td>VUU036</td>
<td>GBLPARAL</td>
<td>Enforce partition parall. unloading into a single file</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>84/141</td>
</tr>
<tr>
<td>VUU037</td>
<td>ULSIGZ</td>
<td>Sign values for zoned-decimal numeric values</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>85/141</td>
</tr>
<tr>
<td>VUU038</td>
<td>ULSIGDTP</td>
<td>Positive sign for date-time packed values</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>86/141</td>
</tr>
<tr>
<td>VUU039</td>
<td>UNLLDER</td>
<td>Db2 HPU response when a LOADDDDN cannot be generated</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>87/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>------------------</td>
<td>------</td>
</tr>
<tr>
<td>VUU040</td>
<td>ULTR32K</td>
<td>Truncate variable records exceeding 32756</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>88/141</td>
</tr>
<tr>
<td>VUU041</td>
<td>ULOCSIZE</td>
<td>Size of the buffer used to retrieve LOB data</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>32/39</td>
</tr>
<tr>
<td>VUU042</td>
<td>UHIDDEN</td>
<td>Unload HIDDEN Column by select *</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>89/141</td>
</tr>
<tr>
<td>VUU043</td>
<td>DTNULDLM</td>
<td>NULL DATE/TIME delimiter option</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>90/141</td>
</tr>
<tr>
<td>VUU044</td>
<td>SUBTKSOF</td>
<td>Support parallelism for unloading several partitions</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>91/141</td>
</tr>
<tr>
<td>VUU045</td>
<td>ULFORMAT</td>
<td>Default output format for a logical unload</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>92/141</td>
</tr>
<tr>
<td>VUU046</td>
<td>ULAUTAG</td>
<td>Size of the autotag in the output file</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>93/141</td>
</tr>
<tr>
<td>VUU047</td>
<td>RETRYMOD</td>
<td>Wait/retry function when resources are unavailable</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>94/141</td>
</tr>
<tr>
<td>VUU048</td>
<td>RETRYW</td>
<td>Wait time between tries to access unavailable resource</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>95/141</td>
</tr>
<tr>
<td>VUU049</td>
<td>RETRYNB</td>
<td>Number of retries when resources are unavailable</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>96/141</td>
</tr>
<tr>
<td>VUU050</td>
<td>ULFRQMSG</td>
<td>Additional parameter for information messages</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>98/141</td>
</tr>
<tr>
<td>VUU050</td>
<td>ULFRQMSG</td>
<td>Frequency to display information messages</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>97/141</td>
</tr>
<tr>
<td>VUU051</td>
<td>PROCMSG</td>
<td>The location where Db2 HPU writes information messages</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>99/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------------------------------------------</td>
<td>------------------------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>VUU052</td>
<td>SELMSG</td>
<td>Location of messages issued for each SELECT statement</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>100/141</td>
</tr>
<tr>
<td>VUU053</td>
<td>OBIDRPT</td>
<td>Generate a list of all OBIDs in the FIC</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>101/141</td>
</tr>
<tr>
<td>VUU055</td>
<td>DATEDEL</td>
<td>Delimiter for external date, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>104/141</td>
</tr>
<tr>
<td>VUU055</td>
<td>DATEDEL</td>
<td>Delimiter for external date, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>103/141</td>
</tr>
<tr>
<td>VUU055</td>
<td>DATEDEL</td>
<td>Delimiter for external date, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>106/141</td>
</tr>
<tr>
<td>VUU055</td>
<td>DATEDEL</td>
<td>Delimiter for external date, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>102/141</td>
</tr>
<tr>
<td>VUU055</td>
<td>DATEDEL</td>
<td>Delimiter for external date, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>105/141</td>
</tr>
<tr>
<td>VUU056</td>
<td>TIMEDEL</td>
<td>Delimiter in external time, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>108/141</td>
</tr>
<tr>
<td>VUU056</td>
<td>TIMEDEL</td>
<td>Delimiter in external time, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>107/141</td>
</tr>
<tr>
<td>VUU056</td>
<td>TIMEDEL</td>
<td>Delimiter in external time, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>109/141</td>
</tr>
<tr>
<td>VUU056</td>
<td>TIMEDEL</td>
<td>Delimiter in external time, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>111/141</td>
</tr>
<tr>
<td>VUU056</td>
<td>TIMEDEL</td>
<td>Delimiter in external time, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>110/141</td>
</tr>
<tr>
<td>VUU057</td>
<td>OPALLFMT</td>
<td>Options apply to all formats</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>112/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------------------------------------------------</td>
<td>-------------------------</td>
<td>--------------------</td>
<td>--------</td>
</tr>
<tr>
<td>VUU058</td>
<td>ULIDENT</td>
<td>LOAD technique for identity column</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>113/141</td>
</tr>
<tr>
<td>VUU059</td>
<td>ULTRIM</td>
<td>Trim the trailing blanks, format DELIMITED</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>118/141</td>
</tr>
<tr>
<td>VUU059</td>
<td>ULTRIM</td>
<td>Trim the trailing blanks, format DSNNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>117/141</td>
</tr>
<tr>
<td>VUU059</td>
<td>ULTRIM</td>
<td>Trim the trailing blanks, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>120/141</td>
</tr>
<tr>
<td>VUU059</td>
<td>ULTRIM</td>
<td>Trim the trailing blanks, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>119/141</td>
</tr>
<tr>
<td>VUU059</td>
<td>ULTRIM</td>
<td>Trim the trailing blanks, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>116/141</td>
</tr>
<tr>
<td>VUU060</td>
<td>ULMAXPAR</td>
<td>Maximum number of partitions processed in one unload</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>121/141</td>
</tr>
<tr>
<td>VUU061</td>
<td>ACCPREP</td>
<td>Concurrent access resolution for prepared statement</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>122/141</td>
</tr>
<tr>
<td>VUU062</td>
<td>ULACCTRL</td>
<td>Unload controlled access table in native mode</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>20/22</td>
</tr>
<tr>
<td>VUU063</td>
<td>ULOPTLX</td>
<td>Technical parameter to unload XML columns</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>123/141</td>
</tr>
<tr>
<td>VUU064</td>
<td>STRNGCST</td>
<td>String constant is unloaded as CHAR/VARCHAR data types</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>124/141</td>
</tr>
<tr>
<td>VUU065</td>
<td>ULRTTST</td>
<td>LOAD technique for ROW-TRANSACTION-TIMESTAMP column</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>114/141</td>
</tr>
<tr>
<td>VUU066</td>
<td>ULRTSID</td>
<td>LOAD technique for ROW-TRANSACTION-START-ID col</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>115/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>-----------------</td>
<td>-------</td>
</tr>
<tr>
<td>VUU067</td>
<td>ULCNVTRC</td>
<td>Option to specify conversion truncation policy</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>125/141</td>
</tr>
<tr>
<td>VUU068</td>
<td>ULUNIEXP</td>
<td>Expansion ratio for conversion to UTF8 CCSID</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>126/141</td>
</tr>
<tr>
<td>VUU069</td>
<td>SQLSWIRC</td>
<td>Return code when the utility has switched to sql mode</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>127/141</td>
</tr>
<tr>
<td>VUU070</td>
<td>ULOPTLDB</td>
<td>Options of table LOAD statement</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>39/141</td>
</tr>
<tr>
<td>VUU071</td>
<td>LOSSCHAR</td>
<td>Manage character loss, Db2 HPU parameters</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>40/141</td>
</tr>
<tr>
<td>VUU072</td>
<td>LOSSNUM</td>
<td>Manage numeric loss, Db2 HPU parameters</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>41/141</td>
</tr>
<tr>
<td>VUU072</td>
<td>LOSSNUM</td>
<td>Manage numeric loss, FAST UNLOAD syntax</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>42/141</td>
</tr>
<tr>
<td>VUU072</td>
<td>LOSSNUM</td>
<td>Manage numeric loss, UNLOAD PLUS syntax</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>43/141</td>
</tr>
<tr>
<td>VUU073</td>
<td>MAXERR</td>
<td>Error message limit, Db2 HPU parameters</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>44/141</td>
</tr>
<tr>
<td>VUU073</td>
<td>MAXERR</td>
<td>Error message limit, FAST UNLOAD syntax</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>45/141</td>
</tr>
<tr>
<td>VUU073</td>
<td>MAXERR</td>
<td>Error message limit, UNLOAD PLUS syntax</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>46/141</td>
</tr>
<tr>
<td>VUU074</td>
<td>ULLOCKTE</td>
<td>Specify the technique used to lock the unloaded object</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>11/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>---------------------------------------------------------------</td>
<td>---------------------</td>
<td>--------------------</td>
<td>--------</td>
</tr>
<tr>
<td>VUU075</td>
<td>NULLPAD</td>
<td>Fill NULL with padding character, format DSNTIAUL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>55/141</td>
</tr>
<tr>
<td>VUU075</td>
<td>NULLPAD</td>
<td>Fill NULL with padding character, format EXTERNAL</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>57/141</td>
</tr>
<tr>
<td>VUU075</td>
<td>NULLPAD</td>
<td>Fill NULL with padding character, format USER</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>54/141</td>
</tr>
<tr>
<td>VUU075</td>
<td>NULLPAD</td>
<td>Fill NULL with padding character, format VARIABLE</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>56/141</td>
</tr>
<tr>
<td>VUU076</td>
<td>ULICTYPE</td>
<td>Image copies selected for a partitioned table space</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>74/141</td>
</tr>
<tr>
<td>VUU077</td>
<td>ULCENT</td>
<td>Year to set the century for DECIMAL to DATE conversion</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>75/141</td>
</tr>
<tr>
<td>VUU078</td>
<td>ULNULLFL</td>
<td>Output field for NULL values in DELIMITED format</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>76/141</td>
</tr>
<tr>
<td>VUU079</td>
<td>EXITHLL</td>
<td>Management of exit written in C or COBOL.</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>3/141</td>
</tr>
<tr>
<td>VUU080</td>
<td>MXOPNLLD</td>
<td>Number of simultaneously opened LOB LDS</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>36/39</td>
</tr>
<tr>
<td>VUU081</td>
<td>UNLLDTBB</td>
<td>LISTDEF behaviour for table selection</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>77/141</td>
</tr>
<tr>
<td>VUU082</td>
<td>ULRLDRC</td>
<td>Action if fully compliant LOADDDN cannot be generated</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>78/141</td>
</tr>
<tr>
<td>VUU083</td>
<td>DISPPARM</td>
<td>Request the display of implicit parameters</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>79/141</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------</td>
<td>--------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>VUU084</td>
<td>ULLOBBHV</td>
<td>Processing when LOB/XML not retrieved</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>80/141</td>
</tr>
<tr>
<td>VUU085</td>
<td>ULLOBBHM</td>
<td>Maximum number of issued messages for LOB/XML errors</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>81/141</td>
</tr>
<tr>
<td>VUX003</td>
<td>SIZE</td>
<td>Default size of the input file or object for Db2 HPU</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>128/141</td>
</tr>
<tr>
<td>VUX004</td>
<td>LOWMEM</td>
<td>Minimum memory size for each sort process</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>24/39</td>
</tr>
<tr>
<td>VUX005</td>
<td>MAXSORT</td>
<td>Maximum number of active sorts processes</td>
<td>LPAR parameters</td>
<td>HPU Sort parameters</td>
<td>37/39</td>
</tr>
<tr>
<td>VUX006</td>
<td>WRKSPACE</td>
<td>Allocation distribution for the sort input file</td>
<td>LPAR parameters</td>
<td>HPU Sort parameters</td>
<td>38/39</td>
</tr>
<tr>
<td>VUX007</td>
<td>WAITUNIT</td>
<td>Waitunit in seconds/100</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>25/39</td>
</tr>
<tr>
<td>VUX008</td>
<td>WAITQTY</td>
<td>Number of wait periods before issuing a WTOR</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>26/39</td>
</tr>
<tr>
<td>VUX009</td>
<td>WAITQTYM</td>
<td>Wait time for the WTOR reply</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>27/39</td>
</tr>
<tr>
<td>VUX010</td>
<td>LIMUNIT</td>
<td>Maximum number of disk units to allocate a work file</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>15/39</td>
</tr>
<tr>
<td>VUX011</td>
<td>__________</td>
<td>Grant on the plan created for Db2 HPU</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>12/22</td>
</tr>
<tr>
<td>VUX016</td>
<td>WRKUNTSW</td>
<td>Maximum size for work data set on DASD</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>20/39</td>
</tr>
<tr>
<td>VUX017</td>
<td>MAXTUNIT</td>
<td>Maximum number of unit for tape temporary data set</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>21/39</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>----------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>VUX018</td>
<td>LDSERLML</td>
<td>Maximum number of messages for row structure errors</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>129/141</td>
</tr>
<tr>
<td>VUX019</td>
<td>WRKMXPR</td>
<td>Maximum size for primary allocation of a work data set</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>16/39</td>
</tr>
<tr>
<td>VUX023</td>
<td>PGDFIN</td>
<td>Percentage of displaced pages in inline FIC</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>130/141</td>
</tr>
<tr>
<td>VUX025</td>
<td>PARALLEL</td>
<td>Maximum degree of parallelism for LDS reading</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>28/39</td>
</tr>
<tr>
<td>VUX030</td>
<td>UTLPARAL</td>
<td>Maximum degree of parallelism for unload tasks</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>29/39</td>
</tr>
<tr>
<td>VUX031</td>
<td>DB2PARAL</td>
<td>Maximum degree of parallelism for SELECT</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>131/141</td>
</tr>
<tr>
<td>VUX032</td>
<td>WRKTVCNT</td>
<td>Number of volumes for temporary data set on tape</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>22/39</td>
</tr>
<tr>
<td>VUX033</td>
<td>TMPLDSCB</td>
<td>DSCB model for allocation of GDS</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>17/39</td>
</tr>
<tr>
<td>VUX036</td>
<td>RTSESTIM</td>
<td>Use real-time statistics tables for size estimation</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>132/141</td>
</tr>
<tr>
<td>VUX037</td>
<td>SORTUTIL</td>
<td>Sort program that Db2 HPU uses to sort data</td>
<td>LPAR parameters</td>
<td>HPU Sort parameters</td>
<td>39/39</td>
</tr>
<tr>
<td>VUX038</td>
<td>CHECKINT</td>
<td>Level of data integrity control of input data</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>133/141</td>
</tr>
<tr>
<td>VUX039</td>
<td>ZIIP</td>
<td>zIIPusage</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>33/39</td>
</tr>
<tr>
<td>VUX040</td>
<td>UBUFSIZE</td>
<td>Size of work areas buffers</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>30/39</td>
</tr>
<tr>
<td>VUX041</td>
<td>UBUFNO</td>
<td>Number of buffers of work areas</td>
<td>LPAR parameters</td>
<td>Db2 HPU Tuning Parameters</td>
<td>31/39</td>
</tr>
<tr>
<td>Key</td>
<td>Short name</td>
<td>Parameter label</td>
<td>Panel</td>
<td>Section</td>
<td>Rank</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-------------------------------------------------------------</td>
<td>----------------</td>
<td>--------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>VUX042</td>
<td>ZIIPMXPL</td>
<td>Maximum parallelism on zIIP for LDS</td>
<td>LPAR parameters</td>
<td>Db2 HPU Parameters</td>
<td>34/39</td>
</tr>
<tr>
<td>VUX043</td>
<td>ZIIPMXPU</td>
<td>Maximum parallelism on zIIP for unload tasks</td>
<td>LPAR parameters</td>
<td>Db2 HPU Parameters</td>
<td>35/39</td>
</tr>
<tr>
<td>VUX044</td>
<td>CLIPLAN</td>
<td>Db2 Plan name of the CLI</td>
<td>Db2 parameters</td>
<td>Db2 HPU DB2 parameters</td>
<td>21/22</td>
</tr>
<tr>
<td>VZM005</td>
<td>INFVSMIG</td>
<td>Volume name of migrated objects</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>13/39</td>
</tr>
<tr>
<td>VZM006</td>
<td>SCUNMOD</td>
<td>Unicode Conversion Services load library</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>23/39</td>
</tr>
<tr>
<td>VZM007</td>
<td>SCUNTSO</td>
<td>Unicode Conversion Services technique search order</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>134/141</td>
</tr>
<tr>
<td>VZM008</td>
<td>SCUNSUB</td>
<td>Unicode Conversion Services character substitution</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>135/141</td>
</tr>
<tr>
<td>VZM009</td>
<td>-----------</td>
<td>List of system codes for which no dump is produced</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>141/141</td>
</tr>
<tr>
<td>VZM010</td>
<td>TAPEUNIT</td>
<td>Numbers of devices for tape unit/storage class</td>
<td>LPAR parameters</td>
<td>HPU File Management parameters</td>
<td>14/39</td>
</tr>
<tr>
<td>VZM011</td>
<td>TAPERPT</td>
<td>Generate a tape usage report in SYSPRINT</td>
<td>Product parameters</td>
<td>Common parameters</td>
<td>136/141</td>
</tr>
</tbody>
</table>

To locate and access a specific Db2 HPU parameter in Tools Customizer's panels, search the key of the parameter in the "key" column of the table and access the Tools Customizer's panel indicated in the "Panel" column. Browse this panel until you reach the section indicated by the "Section" column. Browse the section till you find the searched parameter. The value in the "Rank in the panel" column helps you figure out how much to browse. The "key/shortname" value on the right side of the panel helps you check you have reached the searched location/parameters.

**INZDB21X member**

Use the INZDB21X member to integrate Db2 HPU with Db2 Administration Tool.

The following example shows the INZDB21X member. It is not customized.
INSTRUCTIONS:
HPU installation : Please read all the procedure before beginning.

1) FIRST PART : make a backup
   a) Please back up the following db2 rexx procedures before
      beginning any modification:
      - &VUU027(ADB21S)
      - &VUU027(ADB21T)

ADDRESS ISPEXEC "LIBDEF ISPTABL DATASET",
   "ID('&VUU026')",
   "STACK"
ADDRESS ISPEXEC "LIBDEF ISPTLIB DATASET",
   "ID('&VUU026')",
   "STACK"

/* If you do so, the two following lines must be added at the end of */
/* the rexx procedures before the exit command */
ADDRESS ISPEXEC "LIBDEF ISPTABL DATASET"
ADDRESS ISPEXEC "LIBDEF ISPTLIB DATASET"

/* 2) SECOND PART : update the ADB21S table. */
/* ( Command lines table for the DB2ADMIN tablespace panel ) */
/* Look for HPU in the file */
&VUU027(ADB21S)

/* If it is not in, then, after backing up this file, insert the */
/* following lines. */
/* These lines should be added before the TBSORT and TCLOSE */
/* command, near the end of the file. Be careful not to insert */
/* these lines before the TBADD statement of another command. */
ACMD=''
PAN=''
SQL=''
CMD='HPU'
DESCRIP='Additional command - High Performance Unload'
ISPf='SELECT CMD(INZHPU D TS &DB2IN &DBNAME &NAME)' "TBADD" table

578 Db2 High Performance Unload User's Guide
This example contains the following variables:

&VUU026
The library that contains the Db2 Admin Tool commands tables, such as DBTOOL.SADBTLIB.

&VUU027
The library that contains the ADBDMTI EXEC, such as DBTOOL.SADBEXEC.

Related tasks
Optional: Integrating Db2 HPU into Db2 Administration Tool
Optionally, you can integrate Db2 HPU into Db2 Administration Tool (Db2 Admin). Tools Customizer will create the necessary JCL, but you must manually complete some steps after you submit the customization job.

Data types for output (TYPE keyword)

The TYPE keyword of the SELECT statement (OPTION block for FORMAT USER) is used to create several types of data in the output. These types are declared in the keyword TYPE. The use of this keyword implies that data is to be converted from the original column type to the type that is declared in the TYPE keyword.

The output data types that are allowed are described in the following topics:

- “Numeric data” on page 580
- “Nonnumeric data” on page 580
- “DATE, TIME, and TIMESTAMP data” on page 581
- “Supported conversions” on page 583
**Numeric data**

The numeric data type declared in the TYPE keyword is used to specify the output numeric data type.

You can use the following formats to code numeric data:

**INTEGER or INT**
Whole numbers in a binary word of 31 bits plus the sign.

**SMALLINT**
Whole numbers in a binary halfword of 15 bits plus the sign.

**DECIMAL(\(n,m\)) or DEC(\(n,m\))**
Standard decimal value that is contained in \((n/2+1)\) bytes. The default value is DECIMAL or DEC and is equivalent to DECIMAL (5,0).

**DECIMAL ZONED(\(len, scale\))**

The format of a zoned number is \(znznzn...zn/sn\), where \(n\) denotes a 4-bit decimal digit (called the numeric bits); \(z\) is the digit’s zone (the left four bits of a byte); \(s\) is the right-most operand that can be a zone \((z)\) or a sign value (hexadecimal A, C, E, or F for a positive number, and hexadecimal B or D for a negative number).

- **\(len\)** specifies the number bytes (decimal digits) in the output field. The length must be 1 - 31. If the source data type is DECIMAL and the length parameter is omitted, the default length is determined by the column attribute that is defined in the table. Otherwise, the default length is 31 bytes.

- **\(scale\)** specifies the number of digits to the right of the decimal point (a decimal point is not included in the output field in this case). There must be an integer that is greater than or equal to zero and must be less than or equal to the length. The default depends on the column attributes that are defined in the table. If the source data type is DECIMAL, the defined scale value is taken as the default value; otherwise, the default is 0.

An error occurs if the output field size that is specified is less than the data length. If the size of the output field is less than the data length, X'F0' is padded on the left.

**FLOAT(\(n\))**
Number \((n)\) is simple floating point precision if \((0<n<22)\) in a fullword. Number is double floating point precision if \((21<n<54)\).

The default type for FLOAT is double precision.

**Nonnumeric data**

The nonnumeric data type declared in the TYPE keyword is used to specify the output nonnumeric data type.

You can use the following formats to code nonnumeric data:

**CHARACTER(\(n\)) or CHAR(\(n\))**
Character string of length \(n\) \((0<n<255)\) bytes.

**VARCHAR(\(n\))**
A two-byte length field that is followed by \(n\) characters. The size equals \(n+2\) bytes. The Db2 type LONG VARCHAR is not used in a sequential data set.

**GRAPHIC(\(n\))**
Graphic character string that is coded on \(2n\) bytes. One character equals 2 bytes.

**VARGRAPHIC(\(n\))**
Variable-length graphic character string that is coded on \(2n+2\) bytes. The Db2 type LONG VARGRAPHIC is not used in a sequential data set.
DATE, TIME, and TIMESTAMP data

The DATE, TIME, and TIMESTAMP data types that are declared in the TYPE keyword are used to specify the output data types.

DATE format types

Use the DATE format type to specify the output data format.

The formats, data types, and default length for DATE are shown in the following table:

<table>
<thead>
<tr>
<th>Format</th>
<th>Output data</th>
<th>Data type</th>
<th>Default length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE_A</td>
<td>MM-DD-YYYY</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>DATE_B</td>
<td>MM-DD-YY</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_C</td>
<td>YYYY-MM-DD</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>DATE_D</td>
<td>YY-MM-DD</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_E</td>
<td>DD-MM-YYYY</td>
<td>Character</td>
<td>10</td>
</tr>
<tr>
<td>DATE_F</td>
<td>DD-MM-YY</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_G</td>
<td>YYYY-DDD</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_H</td>
<td>YY-DDD</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>DATE_I</td>
<td>MMDDYYYY</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_J</td>
<td>MMDDYY</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>DATE_K</td>
<td>YYYYMMDD</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_L</td>
<td>YYMMDD</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>DATE_M</td>
<td>DDMMYYYY</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>DATE_N</td>
<td>DDMMYY</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>DATE_O</td>
<td>YYYYDDDD</td>
<td>Character</td>
<td>7</td>
</tr>
<tr>
<td>DATE_P</td>
<td>YYDDD</td>
<td>Character</td>
<td>5</td>
</tr>
<tr>
<td>DATE_Q</td>
<td>YYDDMM</td>
<td>Zoned decimal</td>
<td>6</td>
</tr>
<tr>
<td>DATE_R</td>
<td>YYYYDDMM</td>
<td>Zoned decimal</td>
<td>8</td>
</tr>
<tr>
<td>DATE_0</td>
<td>YYYYMMDD</td>
<td>Packed decimal</td>
<td>5</td>
</tr>
<tr>
<td>DATE_1</td>
<td>MMDDYY</td>
<td>Packed decimal</td>
<td>4</td>
</tr>
<tr>
<td>DATE_2</td>
<td>DDMMYY</td>
<td>Packed decimal</td>
<td>4</td>
</tr>
<tr>
<td>DATE_3</td>
<td>YYYYMDD</td>
<td>Packed decimal</td>
<td>4</td>
</tr>
<tr>
<td>DATE_4</td>
<td>YYDDMM</td>
<td>Packed decimal</td>
<td>4</td>
</tr>
<tr>
<td>DATE_5</td>
<td>MMDDYYYY</td>
<td>Packed decimal</td>
<td>5</td>
</tr>
<tr>
<td>DATE_6</td>
<td>DDMMYYYY</td>
<td>Packed decimal</td>
<td>5</td>
</tr>
<tr>
<td>DATE_7</td>
<td>YYYYDDMM</td>
<td>Packed decimal</td>
<td>5</td>
</tr>
<tr>
<td>DATE_DB2</td>
<td>Date format of the site</td>
<td>Character</td>
<td>10 or the length of the LOCAL date format</td>
</tr>
</tbody>
</table>
By default, Db2 HPU uses the hyphen character (-) or content of the PARMLIB variable VUU055/DATEDEL as a delimiter. To change this delimiter, code DATEDELIM 'd', where d is the new delimiter, in the OPTIONS block.

Related reference
OPTIONS block syntax and description
Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

Db2 HPU output data parameters
The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

TIME format types
Use the TIME format type to specify the output data format.

The formats, data types, and default length for TIME are shown in the following table:

<table>
<thead>
<tr>
<th>Format</th>
<th>Output data</th>
<th>Data type</th>
<th>Default length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME_A</td>
<td>HH.MM.SS</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>TIME_B</td>
<td>HH.MM</td>
<td>Character</td>
<td>5</td>
</tr>
<tr>
<td>TIME_C</td>
<td>HH.MM AM</td>
<td>Character</td>
<td>8</td>
</tr>
<tr>
<td>TIME_D</td>
<td>HHMMSS</td>
<td>Character</td>
<td>6</td>
</tr>
<tr>
<td>TIME_E</td>
<td>HHMM</td>
<td>Character</td>
<td>4</td>
</tr>
<tr>
<td>TIME_0</td>
<td>HHMMSS</td>
<td>Packed decimal</td>
<td>4</td>
</tr>
<tr>
<td>TIME_DB2</td>
<td>Time format of the site Character</td>
<td>8 or the length of the LOCAL time format</td>
<td></td>
</tr>
</tbody>
</table>

By default, Db2 HPU uses the '.' delimiter character or content of PARMLIB variable VUU056/TIMEDEL. To change the delimiter, code TIMEDELIM 'd', where d is the new delimiter, in the OPTIONS block.

Related tasks
Selecting time formats
The Type Values - TIME panel is used to select a time format.

Related reference
OPTIONS block syntax and description
Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

Db2 HPU output data parameters
The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

TIMESTAMP format types
Use the TIMESTAMP format type to specify the output data format.

The formats, data types, and default length for TIMESTAMP are shown in the following table:

<table>
<thead>
<tr>
<th>Format</th>
<th>Output data</th>
<th>Data type</th>
<th>Default length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMESTAMP_A</td>
<td>YYYY-MM-DD-HH.MM.SS</td>
<td>Character</td>
<td>19</td>
</tr>
<tr>
<td>TIMESTAMP_B</td>
<td>YYYY-MM-DD-HH.MM.SS.NNN...NNN</td>
<td>Character</td>
<td>19 - 31(^1)</td>
</tr>
</tbody>
</table>
Table 40. TIMESTAMP format types (continued)

<table>
<thead>
<tr>
<th>Format</th>
<th>Output data</th>
<th>Data type</th>
<th>Default length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMESTAMP_C</td>
<td>YYYYMMDDHHMMSS</td>
<td>Character</td>
<td>14</td>
</tr>
<tr>
<td>TIMESTAMP_D</td>
<td>YYYYMMDDHHMMSS</td>
<td>Character</td>
<td>12</td>
</tr>
<tr>
<td>TIMESTAMP_E</td>
<td>YYYYMMDDHHMMSSSSSSSSSSSS...SSSSSS</td>
<td>Character</td>
<td>14 - 26¹</td>
</tr>
<tr>
<td>TIMESTAMP_F</td>
<td>YYYYMMDDHHMMSSSSSSSSSSSS</td>
<td>Character</td>
<td>12 - 24¹</td>
</tr>
<tr>
<td>TIMESTAMP_G</td>
<td>YYYY-MM-DD HH:MM:SS.SSSS</td>
<td>Character</td>
<td>23</td>
</tr>
<tr>
<td>TIMESTAMP_0</td>
<td>YYYYMMDDHHMMSSSSSSSSSSSSS</td>
<td>Packed decimal</td>
<td>8 - 14¹</td>
</tr>
</tbody>
</table>

Note:

1. The default length depends on the precision of the input data type.

Related tasks

Selecting timestamp formats

The Type Values - TIMESTAMP panel is used to select a timestamp format.

Related reference

OPTIONS block syntax and description

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

Db2 HPU output data parameters

The Db2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

Supported conversions

Db2 HPU supports conversions to specific output data types.

The following table shows all the supported conversions for Db2 data types and Db2 HPU data types. Conversions that are not contained in this table are not supported by Db2 HPU.

Table 41. Supported conversions

<table>
<thead>
<tr>
<th>Db2 data type</th>
<th>Output data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>SMALLINT, DECIMAL, FLOAT, CHAR</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>INTEGER, DECIMAL, FLOAT, CHAR</td>
</tr>
<tr>
<td>DECIMAL(m,n)</td>
<td>SMALLINT, INTEGER, CHAR, DECIMAL(p,q), FLOAT</td>
</tr>
<tr>
<td>FLOAT</td>
<td>SMALLINT, INTEGER, CHAR, DECIMAL, FLOAT</td>
</tr>
<tr>
<td>CHAR(n)</td>
<td>CHAR(m), VARCHAR(m)</td>
</tr>
<tr>
<td>VARCHAR(n)</td>
<td>CHAR(m), VARCHAR(m)</td>
</tr>
<tr>
<td>LONG VARCHAR(n)</td>
<td>CHAR(m), VARCHAR(m)</td>
</tr>
<tr>
<td>GRAPHIC(n)</td>
<td>GRAPHIC(m), VARGRAPHIC(m), CHAR(m)</td>
</tr>
<tr>
<td>VARGRAPHIC(n)</td>
<td>GRAPHIC(m), VARGRAPHIC(m)</td>
</tr>
<tr>
<td>LONG VARGRAPHIC(n)</td>
<td>GRAPHIC(m), VARGRAPHIC(m)</td>
</tr>
</tbody>
</table>
Syntax compatibility

Db2 HPU provides limited support for Fast Unload syntax and UNLOAD PLUS syntax.

Topics:
• “Fast Unload syntax compatibility” on page 584
• “UNLOAD PLUS syntax compatibility” on page 589
• “Db2 unload compatibility” on page 595

Fast Unload syntax compatibility

Db2 HPU provides limited support for the JCL that is used with Fast Unload for Db2, Version 3.1; however, some features of the Fast Unload product might be ignored or be interpreted differently when they are issued by Db2 HPU. In most cases, the amount of work that is required to convert Fast Unload JCL to Db2 HPU JCL is significantly reduced.

For a description of the syntax shown here, refer to the Fast Unload documentation.

Important:
• Do not use this syntax to create new JCL. Fast Unload JCL support is provided only to simplify converting from Fast Unload to Db2 HPU. This syntax is supported to the extent that Db2 HPU can perform processing that is like the processing that is described in the syntax. Many keywords are ignored. Some options are automatically converted to Db2 HPU syntax. For example, OUTPUT-FORMAT LOAD is converted to OUTPUT-FORMAT DSNTIAUL.
• If you specify FASTUNLOAD in SYSIN to use the Fast Unload syntax, precede subsystem_name/group_attachment_name/job_id in the PARM field with the EP=UTGLCTL positional parameter, as shown in the following example:

```
PARM='EP=UTGLCTL/DSN5, DB2UNLOAD'
```

fast unload

```
FASTUNLOAD fast unload options fast unload select statement
```

fast unload options
Notes:

1 Keywords that are ignored are identified by an asterisk (*). Dependent keywords and variables are also ignored.
Keywords that are not supported are identified by two asterisks (**). Dependent keywords and variables are also not supported.

You can specify a maximum of four exits.

---

**Notes:**

1. Keywords that are ignored are identified by an asterisk (*). Dependent keywords and variables are also ignored.
fast unload tablename spec

PART

( integer

ALL

 integer

)

fast unload where clause

ORDER BY

colname

num

ASC

DESC

ORDER CLUSTER

fast unload tablename spec

tablename

creator.tablename

 INTO TABLE creator.tablename

fast unload select options
Notes:

1 You can specify a maximum of four exits.

2 Keywords that are ignored are identified by an asterisk (*). Dependent keywords and variables are also ignored.
UNLOAD PLUS syntax compatibility

Db2 HPU provides limited support for the JCL that is used with UNLOAD PLUS for Db2, Version 2.1.01; however, some features of UNLOAD PLUS might be ignored or be interpreted differently when issued by Db2 HPU. This restrictions are mentioned by notes at the bottom of the syntax diagram. In most cases, the amount of work that is required to convert UNLOAD PLUS JCL to Db2 HPU JCL is significantly reduced.

For a description of the syntax shown here, refer to the UNLOAD PLUS for Db2 documentation.

Important: Do not use this syntax to create new JCL. UNLOAD PLUS JCL support is provided only to simplify converting from UNLOAD PLUS to Db2 HPU. This syntax is supported to the extent that Db2 HPU can perform processing that is like the processing that is described in the syntax. Many keywords are ignored. Some options are automatically converted to Db2 HPU syntax.
Notes:

1. This keyword is ignored. Dependent keywords and variables are also ignored.
2. This keyword is not supported. Dependent keywords and variables are also not supported.
Unlike UNLOAD PLUS, only the cataloged image copy datasets are taken into account. The same behavior as with the Db2 HPU syntax COPYDDN LAST_IC or COPYDDN -integer UNLOAD block syntax and description applies.

### Unload plus csv format

`TERMINATEDBY` 
- `'','`
- `'char'`
- `x'hexValue'`

`SELECT_ELEMENT` 
- `AUTO`
- `NONE`
- `'slemonlemonde.frtring'`

`ENCLOSEDBY` 
- `'''`
- `'char'`
- `x'hexValue'`
- `AND`
- `'''`
- `'char'`
- `x'hexValue'`

`NULLSTRING` 
- `'NULL'`
- `'string'`

Notes:
1. This keyword is not supported. Dependent keywords and variables are also not supported.

### Unload plus message block

`ON MESSAGE` 
- `50251`
- `50253`
- `50254`
- `STOP UTILITY`
- `CONTINUE UTILITY`

Notes:
1. This keyword is not supported. Dependent keywords and variables are also not supported.

### Unload plus failure block

`ON FAILURE` 
- `UTILINIT`
- `UNLOAD`
- `UTILTERM`
- `TERMINATE UTILITY`
- `RETCODE` 
- `integer`
- `STOP UTILITY`

Notes:
1. This keyword is not supported. Dependent keywords and variables are also not supported.
unload plus control card

Notes:
1 This keyword is ignored. Dependent keywords and variables are also ignored.

unload plus select
SELECT * INTO unload plus record, unload plus field
FROM tablename
WHERE unload plus condition
ORDER BY colname ASC DESC

unload plus record
NAME qualifier .
RECORDID 'string'

unload plus field
**Notes:**

1. This keyword is ignored. Dependent keywords and variables are also ignored.
2. This keyword is not supported. Dependent keywords and variables are also not supported.

**unload plus condition**

Notes:

1. Optionally, you can enclose the unload plus predicate in parentheses.
### Db2 unload compatibility

Db2 HPU supports the control cards compliant with the Db2 UNLOAD utility for Db2 for z/OS, version 11 and can be operated with a prior Db2 version unless it involves a Db2 UNLOAD utility feature that relies on Db2 features that are not available in the prior Db2 version.

See the related Db2 publication (Utility Guide and Reference) for more details on the Db2 Unload control statements.

However, some features of the Db2 UNLOAD utility are ignored or implemented differently by Db2 HPU. In most cases, the amount of work that is required to convert a Db2 UNLOAD to Db2 HPU JCL is low.

### Restrictions

The Db2 UNLOAD utility features that are ignored or implemented differently by Db2 HPU are listed in this section. The Db2 HPU behavior when such a feature is requested is described.

### Options

Ignored features:
- NOSUBS: ignored. The message INZU105I is issued. This feature is dealt through VZM008/SCUNSUB.
- DECFLOAT_ROUND_MODE: ignored. The message INZU105I is issued.
- IMPLICIT_TZ: ignored. The message INZU105I is issued.

Altered features:
- MAXERR: MAXXER 1 is supported. MAXERR n with n > 1 is not supported. The default value is MAXERR 1.

Restrictions:
- SPANNED YES: The SPANNED YES option is only supported for LOB data.
Field specification
Restriction:
• POSITION: only POSITION(*) is supported. End with completion code 8 and a message INZU052E is issued.

Not supported:
• timestamp with time zone-spec: not supported. End with completion code 8 and a message INZU052E is issued.

Important:
Do not use this syntax to create new Db2 HPU JCL. The Db2 UNLOAD support is provided only to simplify converting from Db2 UNLOAD to Db2 HPU. This syntax is supported to the extent that Db2 HPU can perform processing that is like the processing that is described in the syntax. Several keywords are ignored. Some options are automatically converted to Db2 HPU features.

Example: Usage of a Db2 UNLOAD JCL with Db2 HPU
In this example below, the unload from a table is requested by using control cards valid for the Db2 UNLOAD (module DSNUTILB).

Note:
The genuine Db2 UNLOAD JCL has been altered to replace PGM=DSNUTILB by PGM=INZUTILC.

Instead of changing the JCL this way, you may consider defining DSNUTILB as an alias for INZUTILC. Be aware that this change would affect any JCL invoking DSNUTILB and lead to failures if DSNUTILB is not used for the UNLOAD utility but for any other online DSNUTILB allows to run.

```
//UNLOAD EXEC PGM=INZUTILC,PARM='DB81,Db2UNLOAD',
//REGION,=0M,TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
//   DD DSN=DSNA10.SDSNLOAD,DISP=SHR
//   DD DSN=DSNA10.SDSNLOAD2,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSREC  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSIN   DD*
UNLOAD TABLESPACE DSN8D10A.DSN8S10E
   FROM TABLE DSN81010.EMP
```
Notices

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

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

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

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

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

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

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

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

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

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

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

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

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, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.
Index

Special Characters

&part keyword 97

A

abend codes 359
accessibility 5
ACCPREP parameter 542
ACTLGUSR parameter 481
Admin Tool
  starting Db2 HPU interactive component 237
ALTERNATE keyword 215
APF load modules library 480
APF load modules parameter 480
APPLCOMP parameter 481
APPLCOMPAT keyword 113
ASCII keyword 183
ASIS keyword 183
authority privileges 81
authorization 25
AUTOTAG keyword 143

B

basic predicate 188
batch utility
  enabling 25
  generating jobs 26, 235
  overview 81
benefits 1
between predicate 188
BLOBF parameter 143
block syntax 90
buffering technique 6
BUFFNO keyword 97

C

CCSID keyword 183
CCSID parameter 143
changing display options 39
CHECK_INTEGRITY keyword 120
CLONE keyword 120
CLONED keyword 92
codes
  abend 359
  parameter block 277
  COL keyword 214
  COLOBJT parameter 481
  COLUMNS primary command 235
compatibility (continued)
  overview 6
  UNLOAD PLUS syntax 589
Complete table name panel 584
complex processing 120
components 2
concatenation of image copies 67
concurrent image copies (CIC) 67
CONCURRENT_ACCESS keyword 113
configuration
  authorizing 25
  enabling batch utility 25
Confirm to leave the current Unload panel 240
consistency
  output data 77
  unloading 77
CONVERSION_TRUNCATION_ALLOWED keyword 143
conversions
  DATE format 143
  supported output data types 583
  TIME format 143
  TIMESTAMP format 143
cookie policy 597
Copy DB2 Entries panel 57
COPYDDN
  specifying file parameters 244
COPYDDN File Parameters panel 244
COPYDDN keyword
  ANYTYPE keyword 120
  CHECK keyword 120
  DDLDDN keyword 120
  FROM COPYTOCOPY 120
  GLOBAL keyword 120
  INLINE keyword 120
  MERGE_IIC 120
  PARTITIONED keyword
  CONSISTENT 120
  SIZE keyword 120
  using with TEMPLATE statements 111
COPYDDN primary command 235
CRYPTUSR parameter 481
CTRLIBM parameter 499
customization
  associated list
    adding Db2 entries 47
    overview 27
  associating Db2 entries 47
  browsing parameters 56
  changing display options 39
  changing parameters 40
  component 27
  copying Db2 entries 57
  Create a DB2 Entry panel 47
  creating Db2 entries 47
  customization jobs
    deleting 59
    displaying 59
customization (continued)
customization jobs (continued)
generating 54
maintaining 59
regenerating 54
renaming 59
sort sequence 54
submitting 54, 59
customization library
deleting jobs 59
maintaining 59
overview 29
re-customizing 59
renaming jobs 59
customization library qualifier
specifying 36
Customized status 27
Customizer Workplace panel 54
customizing a new version of a product 40
customizing a product for the first time 40
customizing settings 36
data sets
customization library 29
data store 29, 31
DATASTOR 31
Discover EXEC library 29
metadata library 29, 31
data store
overview 29
data store data set
specifying 36
Db2 Administration Tool parameters 534
Db2 data sharing members
adding 47
associating 47
copying 57
creating 47
Db2 entries
adding 47
associating 47
copying 57
creating 47
defining 54
deleting 58
generating jobs for 54
removing 58
selecting 54
specifying 54
unassociating 58
Db2 group attach field
specifying 36
Db2 group attach names
adding 47
associating 47
copying 57
creating 47
Db2 HPU conversion parameters 534
Db2 HPU Db2 parameters 481
Db2 HPU file management parameters 535
Db2 HPU library dnames 480
Db2 HPU output data parameters
output data 499
Db2 HPU sort parameters 550
Db2 HPU tuning parameters 542
customization (continued)
Db2 parameters
defining 52
editing 52
DB2 Parameters panel 52
Db2 subsystems
adding 47
associating 47
copying 57
creating 47
defining Db2 parameters 52
defining LPAR parameters 50
defining parameters 48, 54
defining product parameters 49
deleting Db2 entries 58
deleting jobs 43
Discover Customized Product Information panel 45
Discover EXEC
customizing a new version of a product 40, 42
overview 29
retrieving product information automatically 45
Discover process parameters 479
Discovered status 27
discovering previous versions 42
discovering product information 45
display options 39
displaying jobs 59
displaying panel text 39
editing LPAR parameters 50
editing parameters 40
editing product parameters 49
Errors in Customization status 27
finding trace data set 287
Finish Product Customization panel 54
first-time 41
first-time customization 40
generating jobs 54
hiding panel text 39
high-level qualifier 27
Incomplete status 27
Integrating Db2 HPU into Db2 Admin 61, 534
Integrating Db2 HPU into Db2 Launchpad 61
job sort order 54
jobs
deleting 59
displaying 59
maintaining 59
renaming 59
sort order 54
submitting 54, 59
LPAR parameters
defining 50
editing 50
LPAR Parameters panel 50
LPARs 60
main list
adding Db2 entries 47
Associate DB2 Entry for Product panel 47
overview 27
maintaining jobs 59
maximizing information panels 39
metadata libraries
specifying 44
metadata library
customization (continued)
metadata library (continued)
maintenance, best practices 31
overview 29
specifying 36
modifying parameters 40
modifying settings 36
multiple instances 36
multiple-LPAR environment 60
Not Required status 27
options 39
overview 25
panel display options 39
panels
Associate DB2 Entry for Product 47
Create a DB2 Entry 47
Customizer Workplace 54
DB2 Parameters 52
Discover Customized Product Information 45
Finish Product Customization 54
LPAR Parameters 50
Product Parameters 49
Specify the Metadata Library 44
parameters
browsing 56
defining 48, 54
viewing 56
preparing to use Tools Customizer 36
prerequisites 25
product 27
product parameters
changing 43
defining 49
editing 43, 49
modifying 43
Product Parameters panel 49
Ready to Customize status 27
Recommended settings 475
recustomization 40, 43
recustomizing 43
recustomizing a product 40
removing Db2 entries 58
roadmaps
customizing for the first time 41
first-time customization 41
recustomizing 43
Specify the Metadata Library panel 44
specifying data sets 36
specifying metadata libraries 44
starting Tools Customizer 35
status types
Customized 27
Discovered 27
Errors in Customization 27
Incomplete 27
Not Required 27
Ready to Customize 27
steps 25
submitting jobs 54
terminology 27
trace data set 287
troubleshooting
finding trace data set 287
user job card settings
customization (continued)
user job card settings (continued)
specifying 36
viewing parameters 56
customization library
overview 29
customization library qualifier
specifying 36
customizing settings 36
D
data consistency
unloading 77
data conversion
DATE format 143
TIME format 143
TIMESTAMP format 143
data sets
output 183
data sources
full image copies (FIC) 64
incremental image copies 64
last full image copies 66
nonpartitioned table spaces 63
partitioned table spaces 63
data store
maintenance, best practices 31
overview 29
data store data set
specifying 36
data types
nonnumeric 580
numeric 580
Database and Tablespace Selection panel 239
DATACLAS keyword 97
DATE format types
specifying output data 581
DATE keyword 143
DATEDEL parameter 499
DATEDELIM keyword 143
Db2 Admin
integrating Db2 HPU 61
Db2 Admin Tool
starting Db2 HPU interactive component 237
Db2 Administration Tool
integrating with Db2 HPU 577
Db2 Administration Tool parameters
customizing 534
Db2 catalog
access 77
Direct 77
SQL 77
Db2 catalog access 17
Db2 compliance 11
Db2 connection mode 17
Db2 group attach field
specifying 36
Db2 HPU conversion parameters
customizing 534
Db2 HPU Db2 parameters
customizing 481
Db2 HPU file management parameters
customizing 535
Db2 HPU library dsnames
  customizing 480
Db2 HPU output data parameters
  customizing 499
Db2 HPU sort parameters
  customizing 550
Db2 HPU tuning parameters
  customizing 542
DB2 keyword 113, 120
Db2 Launchpad
  integrating Db2 HPU 61
  starting Db2 HPU interactive component 239
Db2 parameter 240
Db2 QUIESCE authority 81
Db2 SELECT privilege 81
Db2 subsystems
  specifying 239
Db2 System panel 239
Db2 Tools Launchpad panel 239
Db2 UNLOAD utility compatibility 81
DB2DBRM 480
DB2PARAL parameter 481
DBCLOBF parameter 143
DBRM library 480
DBRM parameter 480
DDLDDN keyword
  using with TEMPLATE statements 111
ddnames
  allocating by Db2 HPU
  dispparm 87
  ICxx-ICxxnnnn 87
  LBFnnnnn 87
  SORT 87
  allocating by user
  copydd 88
  dispparm 88
  INFPLIB 88
  INZPLIB 88
  JOBLIB 88
  loaddd 88
  outdd 88
  STEPLIB 88
  SYSBIND 88
  SYSIN 88
  SYSPRINT 88
  SYSTEM 88
  unldd 88
  UTPRINT 88
decreasing memory 285
DELIM parameter 143
DELIMITED block syntax 211
DELIMITED format
  description 1
  examples 69
  overview 211
DELIMITED keyword 206, 211
deprecated functions 20
DFSIGDCB parameter 499
diagnostic information
  gathering 286, 287
DIR keyword 97
disability 9
Discover EXEC
  overview 29
Discover process parameters
  customizing 479
DISP keyword 97
display options 39
displaying panel text 39
DISPLUS parameter 481
dispparm 87, 88
DISPPARM parameter 542
documentation
  accessing 4, 8
  sending feedback 4
documentation changes 17
DSFIGDCB keyword 113
DSN keyword
  &variable 97
DSN1 Table Spaces panel 237
DSN1 Tables, Views, and Aliases panel 237
dsnames
  installation libraries 480
  DSNTIAUL block syntax 210
  DSNTIAUL format
    creating sequential data sets 218
    description 1
    examples 69, 225
    overview 210
  DSNTIAUL keyword 206, 210
  DSNTYPE keyword 97
  DTDELIM parameter 499
  DTNUMLDM parameter 499
  DYNALLOC macro (SVC 99) 97
dynamic SQL 17

E

EBCDIC keyword 183
Edit Expression panel 259
EDIT, SUBmit, SaveJCL, JobCard panel 270
encoding schemes 73
ENFORCE COLUMN ORDER parameter 143
example
  Generating the access path during the data extraction
  using CURRENT OPTIMIZATION HINT 181
  Requesting the input data integrity to be checked with
  the LEVEL1 controls 143
  Unloading in XML format using variables 177
  Unloading into a data set allocated via a TEMPLATE with
  DSNTYPE EXTREQ 113
examples
  APPLCOMPAT 120
  Assembler user exit 279
  calling user exit 75
  changing column names 225
  changing columns 70
  changing data types 225
  COBOL/2 user exit 282
  committed data 117, 119
  concatenating columns 70
  consistent image copies 66
  creating Db2 LOAD-compatible sequential data sets 220
  creating sequential data sets
    DSNTIAUL format 218
    USER format 217
  DELIMITED format 69
  DSNTIAUL format 69
input data sources (continued)
full image copies (FIC) 64
incremental image copies 64
last full image copies 66
nonpartitioned table spaces 63
partitioned table spaces 63
install library dsnames 480
interactive component
displaying LOADDDN file parameters 269
displaying OUTDDN files list 266
excluding partitions 242
generating
batch utility jobs 26
JCL 270
including partitions 242
modifying LOADDDN file parameters 269
modifying SELECT statement USER format 262
overview 235
primary commands 235
processing long names 254
saving JCL into file 271
selecting
tables 252
selecting columns 259
selecting formats
date 273
SELECT statement 255
time 274
timestamp 274
selecting mask formats 275
sorting SELECT statement columns 261
specifying options
general 240
SELECT statement 247
specifying parameters
COPYDDN file 244
UNLOAD file 245
specifying SELECT statement format 255
starting
Db2 Admin Tool 237
Db2 Tools Launchpad 239
stand-alone mode 237
updating SELECT statements list 249
updating the job card 272
internal behavior tuning 20
INTERNAL block syntax 215
INTERNAL format
overview 215
INTERNAL keyword 206, 215
INTERNAL_FORMAT keyword 120
INTO keyword 188
INTO_RULES 113
INZ0005 panel 239
INZ0010 panel 239
INZA10 panel 244
INZA201 panel 242
INZA21 panel 242
INZA30 panel 244
INZA40 panel 245
INZA501 panel 247
INZA502 panel 247
INZA601 panel 249

G
GBLPARAL parameter 542
GDGLIMIT keyword 97
general options
specifying 240
General Options panel 240, 270
GENERAL primary command 235
GLOBAL OPTIONS block syntax 113

H
hardware requirements 7
HIDDEN parameter 143
hiding panel text 39

I
ICxx-ICxxnnnn ddname 87
IFERROR keyword 143
IFERROR parameter 143
image copies
full 64
incremental 64
last full 66
unload consistency 77
in predicate 188
INCLUDE keyword 92
incremental copies
examples 64
INDEXSCAN keyword 113, 120
INDEXSPACES keyword 92
INFEXIT 277
INFEXIT member 277
INFVSMIG parameter 535
Input 11
input data
unloading LOB 76
unloading XML 76
input data sources
INZA61 panel 252
INZADBI member 61
INZB101E panel 249
INZB101I panel 255
INZB12 panel 255
INZB201 panel 259
INZB21 panel 259
INZB40 panel 261
INZB501 panel 262
INZB502 panel 262
INZB503 panel 262
INZB504 panel 262
INZB51 panel 262
INZB52 panel 262
INZB601I panel 266
INZB61 panel 266
INZB701 panel 269
INZDB21X 61
INZDB21X member 577
INZEEXECU member 82
INZEXIT 279
INZLINK 480
INZLOAD 480
INZP006 panel
  DATE 273
  MASK 275
  TIME 274
  TIMESTAMP 274
INZP041 panel 254
INZP057 panel 272
INZP060 panel 240
INZPLIB ddname 88
INZPLIB library 480
ISPF messages library 480
ISPF messages parameter 480
ISPF panels library 480
ISPF panels parameter 480
ISPF skeletons library 480
ISPF skeletons parameter 480
ISPF tables library 480
ISPF tables parameter 480

J
JCL
  running two SELECT statements sample 82
JCL (job control language)
    generating
      interactive component 270
JCL primary command 235
JOBLIB ddname 88
JUST parameter 143

K
keyboard shortcuts 9
KEYLABEL keyword 97
keywords
  &part variable 97
  ALTERNATE 215
  APPLCOMPAT 113
keywords (continued)
  AUTOTAG 143
  BUFFNO 97
  CHECK_INTEGRITY 120
  CLONE 120
  CLONED 92
  COL 214
  CONCURRENT_ACCESS 113
  CONVERSION_TRUNCATION_ALLOWED keyword 143
  COPYDDN
    ANYTYPE 120
    CHECK 120
    DDLDDN 120
    FROM_COPYTOCOPY 120
    GLOBAL 120
    INLINE 120
    MERGE_IIC 120
    PARTITIONED CONSISTENT 120
    SIZE 120
  DATABASE 92
  DATACLASS 97
  DATE 143
  DATEDELIM 143
  DB2 113, 120
  DDLDDN 120
  DELIMITED 206, 211
  DIR 97
  DISP 97
  DSFIGDCB 113
  DSNTIAUL 206, 210
  DSNTYPE 97
  EXCLUDE 92
  EXECUTE 113, 120
  EXPDL 97
  EXTERNAL 206
  FLOAT 143
  FORMAT 206
  FROM 188
  GDGLIMIT 97
  IfERROR 143
  INCLUDE 92
  INDEX 92
  INDEXSCAN 113, 120
  INDEXSPACE 92
  INDEXSPACES 92
  INTERNAL 206, 215
  INTERNAL_FORMAT 120
  INTO 188
  INTO_RULES 113
  KEYLABEL 97
  LAST_IC 120
  length 97
  LENGTH 143
  LENGTHBYTE 143
  LIMIT 97
  LIMUNIT 113
  LIST 92, 120, 188
  LISTDEF 92
  LISTDEFD 91
  LISTDEFDD 92
  LOADDDN 206
  LOADINDDDN 143
LOADOPT 143
LOCK 113, 120
LOWMEM 113
LRECL 97
MAXPART 113, 120
MAXPRIME 97
MAXSORT 113
MGMTCLAS 97
MODELDCB 97
NBRSECND 97
NULL 143
NULLID 143
NULLPAD 143
NULLPOS 143
OBID_REPORT 120
OFF 91
OPTIMIZATION HINT keyword 143
OPTIMIZE PART RANGE 113
ORDER BY 188
ORDER CLUSTER 188
ORIGINOBID 188
PADDING 143
PARALLELISM 113, 120
parenthetical expression 97
PART 120, 188
PARTLEVEL 92
PATH
  FILEDATA 97
  PATH_VARIABLES 91
  PATHDISP 97
  PATHMODE 97
  PATHOPTS 97
  PCTPRIME 97
  PIC 143
  PROCMSG 113
  PROGRESS_MSG_FREQ
    integer 113
  QSAM BUFFERS 120
  QSAM-BUFFER 113
  QUERY_ACCELERATION keyword 143
  QUIESCE 113, 120
  QUIESCECAT 120
  RECFM 97
  REFORMAT
    BLOF keyword 143
    CCSID keyword 143
    CLOBF keyword 143
    DBCLOBF keyword 143
    DELIM keyword 143
    IFERROR keyword 143
    INTO keyword 143
    LENGTH keyword 143
    LENGTHBYTE keyword 143
    NULLID keyword 143
    PADDING keyword 143
    PIC keyword 143
    TRIM keyword 143
    TYPE keyword 143
  RETPD 97
  RI
    COPY 92
  S
  TOAT 143
  TOATPRT 143
  TYPE 214, 579
  ULDEGREE 113
  UNLDDN 120
  UNLFREQROWS 120
  UNLROWSET 143
  USER 206, 214
  USING with TEMPLATE statements 111
  VARIABLE 206, 213
  UNLMAXROWS 120
  UNLROWSET 143
  USER 206, 214
  using with TEMPLATE statements 111
  VARIABLE 206, 213
  VOLCNT 97
  VOLUMES 97
  VSAM BUFFERS 120
  VSAM-BUFFER 113
  WITH UR 188
  WRKMXPR 113
  WRKSPACE 113
  WRKTVCNT 113
  WRKUNIT 113
  WRKUNTSW 113
  XML 206, 216
  XMLSET 143
  ZIIP 113
  ZONEDDECOP keyword 143
  last full image copies (FIC)
example 66
LAST_IC keyword 120
Launchpad
  integrating Db2 HPU 61
Launchpad (continued)

starting Db2 HPU interactive component 239
LBFnnnnn ddbname 87

legal notices
cookie policy 597
notices 597
programming interface information 597
terms and conditions 598
trademarks 597

length keyword 97
LENGTH keyword 143
LENGTH parameter 143
LENGTHBYTE keyword 143
LENGTHBYTE parameter 143

libraries
APF load modules 480
authorizing SINZLINK 25

DBRM 480
dsnames 480
INZPLIB 480
ISPF messages 480
ISPF panels 480
ISPF skeletons 480
ISPF tables 480
LIBUTIL 480
LOAD modules 480
PARMLIB 480
SINZDBRM 480
SINZLINK 480
SINZLOAD 480
LIBUTIL library 480
like predicate 188
LIKE table-name 210
LIMIT keyword 97
LIMUNIT keyword 113
LIMUNIT parameter 535
LIST keyword
DATABASE keyword 92
INDEX keyword 92
INDEXSPACE keyword 92
PARTLEVEL keyword 92
TABLE keyword 92
TABLESPACE keyword 92
LISTDEF
examples 222
LISTDEF and LISTDEFTBV block syntax 92
LISTDEF keyword 92
LISTDEF specification 97
LISTDEFTDD keyword 91
LISTDEFTBV keyword 92
LISTDEFTBV specification 97
LOAD modules library 480
LOAD modules parameter 480
LOADDDN
statement example 82
LOADDDN keyword
using with TEMPLATE statements 111
LOADDDN keywordXML keyword
DELIM keyword 183
LOADOPT keyword 183
NULLVAL keyword 183
LOADDDN panel 269
LOADDDN primary command 235
LOADINDDN keyword 143
LOADOPT
parameter example 82
LOADOPT keyword 143
LOB columns
unloading 224
LOB input data 76
LOB output data 76
LOCK keyword 113, 120
LOCK parameter 240
logical unload
description 67
DSNTIAUL example 69
performing 68
specifying LISTDEF 97
LOWMEM keyword 113
LOWMEM parameter 542
LRECL keyword 97

M

MAX_EXPECTED_ROWS keyword 183
maximizing information on panels 39
maximizing performance, See performance
MAXPART keyword 113, 120
MAXPRIME keyword 97
MAXPROCSYS 85
MAXPROCUSER 85
MAXSORT keyword 113
MAXSORT parameter 550
MAXTHREADTASKS 85
MAXUNIT parameter 535
MAXUNIT keyword 113
members
INEXIT 277
INZADBI 61
INZDB21X 577
INZEXECU 82
memory
decreasing 285
Memory 83
messages
return codes 364
metadata library
maintenance, best practices 31
overview 29
specifying 36
MGMTCLAS keyword 97
MODELDCB keyword 97
modifying settings 36
MXOPNLLD parameter 542
MySupport 8

N

NBRSECND keyword 97
new messages 21, 23
nonnumeric data 580
nonpartitioned table spaces 63
notices 597
NULL
specifying by using interactive component 247
NULL keyword 143
NULL parameter 143
null predicate 188
NULLID
  specifying by using interactive component 247
NULLID keyword 143
NULLID parameter 143
NULLPAD keyword 143
NULLPOS
  specifying by using interactive component 247
NULLPOS keyword 143
NULLPOS parameter 499
numeric data 580

O

OBID_REPORT keyword 120
OFF keyword 91
OMVS 85
OPALLFMT parameter 499
OPTIMIZATION HINT keyword 143
OPTIMIZE PART RANGE keyword 113
options
  specifying 240
OPTIONS block syntax 143
OPTIONS primary command 235
ORDER BY keyword 188
ORDER CLUSTER keyword 188
ORDERBY primary command 235
ORIGINOBID keyword 188
OUTDDN File panel 266
OUTDDN keyword
  using with TEMPLATE statements 111
OUTDDN List panel 266
OUTDDN primary command 235
OUTEXIT keyword
  ASM keyword 183
  C keyword 183
  COBOL2 keyword 183
OUTFREQROWS keyword
  ON_RESULT_TABLE keyword 183
OUTMAXROWS keyword
  ON_RESULT_TABLE keyword 183
Output 11
output data
  consistency 77
  DATE type 581
  nonnumeric 580
  numeric 580
  TIME type 582
  TIMESTAMP type 582
  types 579
  unloading LOB 76
  unloading XML 76
output data consistency
  unloading 77
output data sets 183
output encoding scheme 73
output files
  allocating 74
  calling a user exit 75
  limiting rows unloaded 75
  overriding 74
  partitions 74
  specifying one output file per partition 74
output formats (continued)
  DELIMITED 1, 211
  DELIMITED format 69
  DSNTIAUL 1, 210
  DSNTIAUL format 69
  encoding scheme 73
  EXTERNAL 1, 73
  INTERNAL 215
  logical unload 68
  physical unload 68
  REORG unload-only 1, 68
  specifying parameters 499
  USER 1, 70, 214
  VARIABLE 1, 70, 213
XML 216
output types
  TEMPLATESET specification 143
overview 1

P

PADDING keyword 143
PADDING parameter 143
panel display options 39
panels
  Copy DB2 Entries 57
PARALLEL parameter 542
parallel processing 6, 63, 88, 120, 183
PARALLELISM keyword 113, 120
parameter block 277
parameters
  ACCPREP 542
  ACTLGUSR 481
  APF load modules 480
  APPLCOMP 481
  BLOBF 143
  CCSID 143
  COLLOBJT 481
  CRYPTUSR 481
  CTLLIBM 499
  customizing Db2 481
  DATEDEL 499
  DB2PARAL 481
  DBCLOBF 143
  DBRM 480
  DELIM 143
  DFSIGDCB 499
  DISPLUSR 481
  DISPPARM 542
  DTDELIM 499
  DTNULDLM 499
  ENFORCE_COLUMN_ORDER 143
  EXITHLL 542
  EXPLNUSR 481
  GBLPARAL 542
  HIDDEN 143
  IFERROR 143
  INFVSMIG 535
  ISPF messages 480
  ISPF panels 480
  ISPF skeletons 480
  ISPF tables 480
  JUST 143
  LENGTH 143
parameters (continued)

LENGTHBYTE 143
LIMUNIT 535
LOAD modules 480
LOWMEM 542
MAXSORT 550
MAXUNIT 535
MXOPNL2D 542
NULL 143
NULLID 143
NULLPOS 499
OPALLFMT 499
PADDING 143
PARALLEL 542
PAMLIB 480
PGDFIN 481
PIC block 143
PLANOBJT 481
PLANOWN 481
QSBUNNO 535
QUITSR 481
RETRYMOD 542
RETRYN 542
RETYW 542
RTSESTIM 481
SCUNMOD 534
SCUNSUB 534
SCUNTSO 534
SIZE 542
SORTUTIL 550
SPANNED 143
STRINGCST 499
SUBTKSOF 542
TAPEUNIT 535
TEMPLATE 143
TIMEDEL 499
TMPLDSCB 535
ULACCTRL 481
ULAUTAG 499
ULCHKCPY 481
ULCNVTRC 481
ULDATE 499
ULDEGREE 481
ULFORMAT 499
ULHIDDEN 481
ULIDENT 499
ULLIKE 481
ULLOCK 481
ULMASK 499
ULMAXPAR 542
ULNULL 499
ULOSIZE 542
ULOPTLDP 499
ULOPTLDT 499
ULOPTLX 481
ULOPTNS 481
ULPIC 499
ULQSC 481
ULQSCBH 481
ULROWSET 542
ULRSTID 481
ULRTST 481
ULSDB2 481
ULSIGDTP 499

parameters (continued)

ULTIME 499
ULTMSTP 499
ULTR32K 499
ULTRIM 499
UNLQDTO 499
UNLQSCBH 481
ULPUBLIC 481
UNLSCHEM 481
UTLPARAL 542
VIZ003 480
VIZ004 480
VIZ007 480
VIZ012 480
VIZ013 480
VIZ015 480
VIZ016 480
VIZ017 480
VIZ018 480
VSBUFND 535
VUA007 535
VUM011 481
VUM012 481
VUM018 535
VUM022 535
VUM023 535
VUM028 481
VUM029 499
VUM030 481
VUM031 481
VUM032 481
VUM033 481
VUM034 481
VUM035 481
VUM036 481
VUM037 481
VUM038 481
VUM039 481
VUM040 542
VUM041 542
parameters (continued)

VyU042 481
VyU043 499
VyU044 542
VyU045 499
VyU046 499
VyU047 542
VyU048 542
VyU049 542
VyU054 499
VyU055 499
VyU056 499
VyU057 499
VyU058 499
VyU059 499
VyU060 542
VyU061 542
VyU062 481
VyU063 481
VyU064 499
VyU065 481
VyU066 481
VyU067 481
VyU079 542
VyU080 542
VyU083 542
VyX003 542
VyX004 542
VyX005 550
VyX006 550
VyX007 542
VyX008 542
VyX009 542
VyX010 535
VyX011 541
 VyX016 535
VyX017 535
VyX019 535
VyX023 481
VyX025 542
VyX030 542
VyX031 542
VyX032 535
VyX033 535
VyX036 481
VyX037 550
VyX039 542
VyX042 542
VyX043 542
VzM005 535
VzM006 534
VzM007 534
VzM008 534
VzM010 535
WaitQTY 542
WaitQTYM 542
WaitUnit 542
WRKMXPR 535
WRKSPACE 550
WRKTUNIT 535
WRKTVCNT 535
WRKUNTSW 535
WRKVOL 535
ZiP 542

parenthetical expression keyword 97
PARMLIB library 480
PARMLIB parameter 480
PART keyword 120, 188
PART primary command 235
Partition List panel 242
partitions
specifying one output file 74
PATH keyword
FILEDATA keyword 97
LRECL keyword 97
RECFM keyword 97
PATH_VARIABLES keyword 91
PATHDISP keyword 97
PATHMODE keyword 97
PATHOPTS keyword 97
PCTPRIME keyword 97
performance
buffering 6
maximizing 6
parallel processing 6
synchronization technique 6
Performance 11
PGDFIN keyword 113
PGDFIN parameter 481
physical unload
description 67
specifying LISTDEF 97
PIC block parameter 143
PIC keyword 143
PIC sign
specifying by using interactive component 247
PLANOBJT parameter 481
PLANOWN parameter 481
predicates
basic 188
between 188
in 188
like 188
null 188
preparing to use Tools Customizer 36
primary commands
COLUMNS 235
COPYDDN 235
FORMAT 235
FUSER 235
GENERAL 235
JCL 235
LOADDDN 235
OPTIONS 235
ORDERBY 235
OUTDDN 235
PART 235
SELECT 235
SELECTALL 235
UNLDDN 235
WHERE 235
privileges 81
problems
diagnostic information about 286, 287
process 2
PROCESS OPTIONS block syntax 91
processing complex SELECT statements 183
Processing mode considerations 84
PROCMSG keyword 113
programming interface information 597
PROGRESS_MSG_FREQ keyword
integer 113
Q
QSAM BUFFERS keyword 120
QSAM-BUFFER keyword 113
QSBUFNO parameter 535
QUERY_ACCELERATION keyword 143
QUIESCE keyword 113, 120
QUIESCE parameter 240
QUIESCECAT keyword 120
QUITUSR parameter 481
R
RACF READ authority 81
reader comment form 4
RECFM keyword 97
Recommended settings
settings 475
REFORMAT
example 224
REFORMAT keyword
BLOF keyword 143
CCSID keyword 143
CLOBF keyword 143
DBCLOBF keyword 143
DELM keyword 143
IFERROR keyword 143
INTO keyword 143
LENGTH keyword 143
LENGTHBYTE keyword 143
NULLID keyword 143
PIC keyword 143
TRIM keyword 143
TYPE keyword 143
registers
specifying supported 188
USER exit communication 277
removed messages 24
REORG unload-only format 68
requirements
hardware 7
software 7
reserved words
using value of 120
RETPD keyword 97
RETRYMOD parameter 542
RETRYNB parameter 542
RETRYW parameter 542
RI keyword
COPY keyword 92
roadmaps
customizing for the first time 41
first-time customization 41
rows
limiting unloaded 75
RTSESTIM parameter 481
S
sample parameters
unload 499
SAVE JCL panel 271
scalar functions 188
scenarios 5
screen readers and magnifiers 5, 9
SCUNMOS parameter 534
SCUNSUB parameter 534
SCUNTSO parameter 534
search condition 188
SELECT block syntax 183
SELECT Columns panel 259
SELECT Format panel 249, 255
SELECT Order By panel 261
SELECT primary command 235
SELECT statement
examples 82, 217, 220
SELECT Statement List panel 249
SELECT Statement Option panel 247
SELECT statements
overview 183
processing complex 183
Select User Format panel 262
SELECTALL primary command 235
SELMESG keyword 113
service information 4
SINZDBRM library 480
SINZLINK library
authorizing 25
description 480
SINZLOAD library 480
SIZE parameter 542
software requirements 7
SORT ddname 87
SORT keyword 113, 120
SORTCLASS keyword 113
sorting data 40
SORTUTIL keyword 113
SORTUTIL parameter 550
SPACE keyword 97
SPANNED keyword 143
SPANNED NO keyword 113
SPANNED parameter 143
SPANNED YES keyword 113
special registers 188
specifying data sets 36
SQL mode 20
SQLID keyword 113
SQLPART keyword 188
STACK keyword 97
stand-alone mode
starting 237
standard Db2 type data 579
start keyword 97
statements
EXEC 82, 84
SELECT 82, 120, 217
UNLOAD TABLESPACE 120
STEPLIB ddname 88
STORCLAS keyword 97
STRINGCST parameter 499
STORCLAS keyword 97
SUBSYS keyword 97
Index 613
subsystems
  specifying 239
SUBTKSOF parameter 542
summary of changes 17
support
  required information 286, 287
support information 4
supported conversions 583
SVC 99 97
synchronization technique 6
syntax compatibility 584
syntax compatibility
  Fast Unload 584
UNLOAD PLUS 589
syntax diagrams
  DELIMITED block 211
  DSNTIAUL block 210
  fast select block 188
  FORMAT block 206
  GLOBAL OPTIONS block 113
  INTERNAL block 215
  LISTDEF block 92
  main 90
  OPTIONS block 143
  PROCESS OPTIONS block 91
  SELECT block 183
  TEMPLATE block 97
  UNLOAD block 120
  USER block 214
  VARIABLE block 213
  XML block 216
SYSABEND ddname 88
SYsin ddname 88
SYSPRINT
dname 88
example 232
SYSPRINT example 232
SYSTERM ddname 88

T

Table List panel 252
table spaces
  nonpartitioned input data 63
  partitioned input data 63
  selecting 239
TABLESPACES keyword 92
tape devices 120
tape options 97
TAPEREPORT keyword 113, 120
TAPEUNIT keyword 113, 120
TAPEUNIT parameter 535
technote 4, 8
TEMPLATE
  examples 222
  block syntax 97
  keyword 97
  parameter 143
  statements
    keywords 111
    using to unload data from a table with LOBs 112
    using to unload data from a table with XML 112
  TMPLDSCB keyword 91

TEMPLEASET (continued)
  processing multiple UNLOAD and SELECT statements 221
TEMPLEASET keyword 143
terminology 7
TIME format types
  specifying output data 582
TIME keyword 143
TIMEDEL parameter 499
TIMEDELIM keyword 143
TIMESTAMP format types
  specifying output data 582
TIMESTAMP keyword 143
TMPLDSCB parameter 535
Tools Customizer
  associated list
    adding Db2 entries 47
    overview 27
  associating Db2 entries 47
  browsing parameters 56
  changing display options 39
  component 27
  Copy DB2 Entries panel 57
  copying Db2 entries 57
  Create a DB2 Entry panel 47
  creating Db2 entries 47
  customization jobs
    deleting 59
    displaying 59
    generating 54
    maintaining 59
    renaming 59
    sort sequence 54
    submitting 54, 59
  customization library
    deleting jobs 59
    maintaining 59
    re-customizing 59
    renaming jobs 59
  customization library qualifier
    specifying 36
  customization templates 552
  Customized status 27
  Customizer Workplace panel 54
customizing a new version of a product 40, 42
customizing a product for the first time 40, 41
customizing conversion parameters 534
customizing Db2 HPU Db2 parameters 481
customizing file management parameters 535
customizing output data parameters 499
customizing sort parameters 550
customizing tuning parameters 542
data sets
customization library 29
data store 29, 31
DATASTOR 31
Discover EXEC library 29
metadata library 29, 31
data store data set
  specifying 36
Db2 data sharing members
  adding 47
  associating 47
  copying 57
Tools Customizer (continued)

Db2 data sharing members (continued)
creating 47
Db2 entries
adding 47
associating 47
copying 57
creating 47
defining 54
deleting 58
generating jobs for 54
removing 58
selecting 54
specifying 54
unassociating 58
DB2 entries 27
Db2 group attach field
specifying 36
Db2 group attach names
adding 47
associating 47
copying 57
creating 47
Db2 parameters
defining 52
editing 52
DB2 Parameters panel 52
Db2 subsystems
adding 47
associating 47
copying 57
creating 47
DBb2 entries
deleting 58
unassociating 58
defining Db2 parameters 52
defining LPAR parameters 50
defining parameters 48, 54
defining product parameters 49
deleting Db2 entries 58
deleting jobs 43
Discover Customized Product Information panel
customizing parameters 479
Discover EXEC
customizing a new version of a product 40, 42
retrieving product information automatically 45
Discovered status 27
discovering product information 45
displaying jobs 59
displaying panel text 39
editing LPAR parameters 50
editing product parameters 49
Errors in Customization status 27
features 3
finding trace data set 287
Finish Product Customization panel 54
first-time customization 40, 41
generating jobs 54
hiding panel text 39
high-level qualifier 27
Incomplete status 27
integrating Db2 HPU into Db2 Admin 534
job sort order 54
jobs
Tools Customizer (continued)

jobs (continued)
deleting 59
displaying 59
maintaining 59
renaming 59
submitting 59
locating a parameter 552
LPAR Parameters panel 50
main list
adding Db2 entries 47
Associate DB2 Entry for Product panel 47
overview 27
maintaining jobs 59
maximizing information on panels 39
metadata libraries
specifying 44
metadata library
specifying 36
multiple instances 36
multiple-LPAR environment 60
Not Required status 27
overview 3
panels
Associate DB2 Entry for Product 47
Copy DB2 Entries 57
Create a DB2 Entry 47
Customizer Workplace 54
DB2 Parameters 52
Discover Customized Product Information 45
Finish Product Customization 54
LPAR Parameters 50
Product Parameters 49
Specify the Metadata Library 44
parameters
browsing 56
viewing 56
preparing to use 36
product 27
product parameters
changing 43
editing 43
modifying 43
Product Parameters panel
customizing conversion parameters 534
customizing Db2 Admin parameters 534
customizing Db2 HPU Db2 parameters 481
customizing file management parameters 535
customizing library dsnames 480
customizing output data parameters 499
customizing sort parameters 550
customizing tuning parameters 542
Ready to Customize status 27
recustomization 40
recustomizing a product 40, 43
removing Db2 entries 58
roadmaps
customizing a new version of a product 42
recustomizing a product 43
using the Discover EXEC 42
Specify the Metadata Library panel 44
specifying metadata libraries 44
starting 35
status types
Tools Customizer (continued)

status types (continued)

Customized 27
Discovered 27
Errors in Customization 27
Incomplete 27
Not Required 27
Ready to Customize 27
submitting jobs 54
terminology 27
trace data set 287
troubleshooting 286
user job card settings
specifying 36
using the Discover EXEC 42
viewing parameters 56

trace data set
finding 287
trademarks 597
translations 73
TRIM keyword 143
troubleshooting
decreasing memory 285

Troubleshooting 11
TRTCH keyword 97
TYPE keyword 214, 215, 579
Type Values panel
DATE 273
MASK 275
TIME 274
TIMESTAMP 274

U

ULACCTRL parameter 481
ULAUTAG parameter 499
ULCHKCPY parameter 481
ULCNVTRC parameter 481
ULDATE parameter 499
ULDEGREE keyword 113
ULDEGREE parameter 481
ULHIDDEN parameter 481
ULIDENT parameter 499
ULLIKE parameter 481
ULLOCK parameter 481
ULMASK parameter 499
ULMAXPAR parameter 542
ULNULL parameter 499
ULOCSIZE parameter 542
ULOPTLDP parameter 499
ULOPTLDT parameter 499
ULOPTLX parameter 481
ULOPTNS parameter 481
ULOPTNS/ONDEMAND_RESOURCE_ALLOCATION 20
ULPIC parameter 499
ULQSC parameter 481
ULQSCEBH parameter 481
ULRLDRC keyword 113
ULROWSET parameter 542
ULRSID parameter 481
ULRTST parameter 481
ULSEDB2 parameter 481
ULSIGDTP parameter 499
ULTIME parameter 499
ULTMSTP parameter 499
ULTR32K parameter 499
ULTRIM parameter 499
UNCNT keyword 97
UNICODE keyword 183
UNICODE_EXPANSION_RATIO keyword 143
UNIT keyword 97
UNLDDDN parameter 499
UNLDDN
specifying file parameters 245
UNLDDN File Parameters panel 245
UNLDDN keyword
using with TEMPLATE statements 111
UNLDDN primary command 235
UNLFREQROWS keyword 120
UNLFREQROWS parameter 240
UNLDER parameter 481
UNLDTBB keyword 113
UNLMAXROWS keyword 120
UNLMAXROWS parameter 240
UNLOAD block syntax 120
UNLOAD command
overview 90
unloading nonpartitioned table spaces 63
unload parameters 499
unload plus syntax 589
UNLOAD TABLESPACE statement 120
UNLROWSET keyword 143
UNLSCHEM parameter 481
upgrading
upgrading Db2 for z/OS 60
user abend codes 359
USER block syntax 214
user exit
calling 75
communication 277
overview 277
parameter block 277
USER format
creating sequential data sets 217
description 1
examples 70
overview 214
User Format Zoom panel 262
User Job Card panel 272
user job card settings
specifying 36
USER keyword 206, 214
utilities
management solutions 8
utility management
comprehensive solutions 8
UTLPARAL parameter 542
UTPRINT ddname 88

V

VARIABLE block syntax 213
VARIABLE format
creating Db2 LOAD-compatible sequential data sets 220
description 1
examples 70
overview 213
VARIABLE keyword 206, 213
VIZ003 parameter 480
VIZ004 parameter 480
VIZ007 parameter 480
VIZ012 parameter 480
VIZ013 parameter 480
VIZ015 parameter 480
VIZ016 parameter 480
VIZ017 parameter 480
VIZ018 parameter 480
VOLCNT keyword 97
VOLUMES keyword 97
VSAM BUFFERS keyword 120
VSAM-BUFFER keyword 113
VSBUFND parameter 535
VUA007 parameter 535
VUM011 parameter 481
VUM012 parameter 481
VUM018 parameter 535
VUM022 parameter 535
VUM023 parameter 535
VUM028 parameter 481
VUM029 parameter 499
VUM030 parameter 481
VUM031 parameter 481
VUM032 parameter 481
VUM033 parameter 481
VUM034 parameter 481
VUM035 parameter 481
VUM036 parameter 481
VUM037 parameter 481
VUM038 parameter 481
VUM039 parameter 499
VUM040 parameter 499
VUM041 parameter 499
VUM042 parameter 481
VUM043 parameter 499
VUM044 parameter 499
VUM045 parameter 499

VUU046 parameter 499
VUU047 parameter 542
VUU048 parameter 542
VUU049 parameter 542
VUU054 parameter 499
VUU055 parameter 499
VUU056 parameter 499
VUU057 parameter 499
VUU058 parameter 499
VUU059 parameter 499
VUU060 parameter 542
VUU061 parameter 542
VUU062 parameter 481
VUU063 parameter 481
VUU064 parameter 499
VUU065 parameter 481
VUU066 parameter 481
VUU067 parameter 481
VUU079 parameter 542
VUU080 parameter 542
VUU083 parameter 542
VUU003 parameter 542
VUU004 parameter 542
VUU005 parameter 550
VUU006 parameter 550
VUU007 parameter 542
VUU008 parameter 542
VUU009 parameter 542
VUU010 parameter 535
VUU011 parameter 481
VUU012 parameter 481
VUU013 parameter 481
VUU014 parameter 499
VUU015 parameter 499
VUU016 parameter 499
VUU017 parameter 499
VUU018 parameter 499
VUU019 parameter 499
VUU020 parameter 499
VUU021 parameter 481
VUU022 parameter 499
VUU023 parameter 481
VUU025 parameter 534
VUU026 parameter 534
VUU027 parameter 534
VUU028 parameter 481
VUU029 parameter 499
VUU030 parameter 481
VUU031 parameter 499
VUU032 parameter 499
VUU033 parameter 481
VUU034 parameter 481
VUU035 parameter 542
VUU036 parameter 542
VUU038 parameter 499
VUU039 parameter 481
VUU040 parameter 499
VUU041 parameter 542
VUU042 parameter 481
VUU043 parameter 499
VUU044 parameter 542
VUU045 parameter 499

W
WAITQTY parameter 542
WAITQTYM parameter 542
WAITUNIT parameter 542
what's new 11
WHERE clause
search condition 188
WHERE primary command 235
WITH UR keyword 188
WRKMXPR keyword 113
WRKMXPR parameter 535
WRKSPACE keyword 113

Index 617
WRKSPACE parameter 550
WRKTUNIT parameter 535
WRKTCNT keyword 113
WRKTCNT parameter 535
WRKUNIT keyword 113
WRKUNTSW keyword 113
WRKUNTSW parameter 535
WRKVOL parameter 535

X
XML block syntax 216
XML format
  overview 216
XML input data 76
XML keyword 206, 216
XML output data 76
XML parameter 143
XMLSET keyword 143

Z
ZIIP keyword 113
ZIIP parameter 86, 542
ZONED_DEC_SIGN keyword 143
Zoom Partition panel 242