3.5

*IBM Db2 Log Analysis Tool for z/OS
User's Guide*

IBM

Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

Subsequent editions of this PDF will not be delivered in the IBM Publications Center. Always download the latest edition from the Db2 Tools Product Documentation page.

Contents

Figures

Tables

About this information

This guide provides instructions for configuring IBM Db2 Log Analysis Tool for z/OS, and describes how to use it.

The typical user of this product is a database administrator who understands the issues relating to data integrity and monitoring data changes. This person must be familiar with Db2 and z/OS. The information in this document is designed to help data processing professionals perform these tasks:

- Monitor changes that are made to database tables.
- Create reports by various database criteria and use filters to view changes that are made to data in the tables.
- Undo changes that were made accidentally.
- Redo changes, such as for a testing environment.
- Determine if tighter controls are needed to ensure data is secure.

Chapter 1. Db2 Log Analysis Tool overview

This guide is intended for database administrators. It contains information on configuring and using Log Analysis Tool.

What's new in Db2 Log Analysis Tool

This section describes technical changes to Db2 Log Analysis Tool

New and changed information is marked with a vertical bar to the left of a change. Editorial changes that have no technical significance are not marked.

Older changes and enhancements are described in "What's new in previous editions" toward the back of this book.

New and changed functions

This topic provides a brief description of the recent enhancements and changes in Db2 Log Analysis Tool.

This section summarizes the significant improvements or enhancements for IBM Db2 Log Analysis Tool for z/OS V3.5 and refers you to relevant sections of this book for more information. Minor changes are not listed. For easier reading, the product name has been shortened to "Log Analysis Tool."

Older changes and enhancements are described in .

June, 2023

Description	Related APARs
The table versioning support is improved in the Log Analysis tool to work with Db2 Table. Before this enhancement, the tool used an internal dataset (TVFILE) to store the table versioning information in the binary format. With PH53523 release, the tool utilizes the Db2 table instead of the binary TVFILE to store and manage the table versioning information. This upgraded support helps in easy management of table versioning entries, provides customizable method for deleting the versioning entries, and eliminates the need to change the TVFILE name after the 'Track Schema Changes' run. For more information, see Table versioning support.	PH53523

November, 2022

Description	Related APARs
The messages in the Db2 Log Analysis Tool messages section are updated to match the product for components like 'Collect object data' parameter, panel commands, filters, zIIP processing, transaction report, parameter validation and so on.	

July, 2022

Description	Related APARs
A new section Specify Delimiters is added. For information, see "Specifying delimiters" on page 144.	

June, 2022

Description	Related APARs
Log Analysis Tool code has been changed to support Db2 13 function levels 500 and 501.	

April, 2022

Description	Related APARs
Option to print only changed columns has been added: This parameter is presented on the "Generate Database Activity Report (Details)" panel. This option allows you to customize the detail report so that only changed columns are printed. For more details, see "To run a Details Report" on page 129. This parameter can also be used in "Generate Database Activity Report (General)" panel workflow if the "Generate details" option is selected. For more details, see "Generate details" on page 98. The "Only changed columns" parameter also appears on the Process Archived Output panel, which is displayed when you choose option 6 on the main menu. For more details, see Chapter 10, "Process archived output," on page 161.	PH41975
An option to bypass use of the SYSLGRNX for Modify Recovery has been added. This option automatically turns off SYSLGRNX processing if it detects MODIFY RECOVERY utility usage during the reported timeframe.	PH38596

December, 2021

Description	Related APARs
A new parameter, MOVE TABLE, has been added for the ALTER TABLESPACE statement. This parameter enables you to move tables from deprecated table spaces to universal table spaces (UTS). See "Constraints" on page 9 for more information.	
When specifying values on the General Reports screens, more options have been added, which are the following: • A new "L" parameter on the General Reports screen (Misc. section) allows the use of high-speed processing even with a LOAD LOG YES setting. • A new "M" parameter on the General Reports screen (Output section) writes any Db2 commands found during the log scan to a command report. • A new "X" value on the General Reports screen (Generate Details field) allows you to run the "General" report without also running a "Details" report in order to save DASD. See "How to create a General Report" on page 85 and "Specifying values in the General Report fields" on page 86 for more information.	
A new parameter, Include load step in JCL, gives you the option to add the load step (LOADFILE) in the JCL that is generated. See Chapter 8, "Loading data into Db2," on page 139 for more information.	

Description	Related APARs
More information was added to the Mode File Report. To see examples, refer to "Mode File Report" on page 215.	
A new field, "Create JSON parameter file", has been added to the **User Settings** panel and the **Quick Start User Settings** panel to enable you to significantly minimize the amount of parameters you need to specify. See "Set user settings" on page 67 and "Quick start-specifying user settings" on page 185 for details.	
A new chapter, Chapter 23, "Modifying a JSON parameters file," on page 259, has been added to provide more details on how this file can be used.	

2021-04-28

Description	Related APARs
You can now generate a job that detects changes to table versions, and you can run the job regularly to enable automated table versioning. For more information, see Table versioning support.	PH20156
Data encryption using Huffman compression is now supported. The topic "Constraints" on page 9 was updated and message ALAA741E was removed.	PH24753
The documentation was updated to add the Utilities menu to both the Log Analysis Tool Classic Menu and the Quick Start Menu. The topics Chapter 11, "Using utilities," on page 163, "Converting a timestamp to an LRSN" on page 165, and "Converting an LRSN to a timestamp" on page 166 were added. In addition, topics "Starting to use Db2 Log Analysis Tool" on page 79, "Accessing the tutorial" on page 83, and "Selecting quick start activity" on page 183 were updated.	PH03886

2020-02-04

Description	Related APARs
The documentation for the Misc Flag field on the general reports panel was revised to clarify the P option setting, and to better describe the potential issues when a point-in-time recovery is encountered for objects. The topic "Specifying values in the General Report fields" on page 86 and message "ALAA156I" on page 277 were updated.	None
Data encryption information was updated to clarify the types of data compression supported and not supported by Log Analysis Tool. The topic "Constraints" on page 9 was updated and message ALAA741E was added.	None
Instructions were added for disabling multi-row insert when running dynamic SQL in batch. Multi-row insert is enabled by Fast Apply when there are more than two inserts per commit in a statement. The topic Chapter 9, "Executing dynamic SQL in batch," on page 155 was updated.	None

2019-09-24

Description	Related APARs
A topic was added that describes the layout of the archive output data set that is generated when output is archived from running detail reports. The topic "Archive output data set description" on page 411 was added and the topic Chapter 10, "Process archived output," on page 161 was updated.	None
An option to continue processing after Log Analysis Tool encounters certain errors was added to the **Continue processing after warnings** field on the **User Settings** panel. The topic "Set user settings" on page 67 was updated.	PI85940

2019-07-19

Description	Related APARs
If the Db2 system that you will use Log Analysis Tool with has an application defaults load module name other than DSNHDECP, you can now specify that name under option 11.2 from the classic menu or the quick start menu. The topic "Entering Db2 Log Analysis Tool specific information" on page 82 was updated. Also, the control file report example was updated in the topic "Control File Report" on page 204.	PH11341
An enhancement was added to allow you to reorder the columns for both delimited and fixed output LOAD formats. The topic "Using the COLUMNS command" on page 148 was added, the topic "Edit column order" on page 147 was changed, and messages were added and removed for this feature.	PH13447
If you want the LOADFILE to be split into one data set per object, you can modify data set node names for the split load files and the load cards so that object IDs are not used in the data set names. The topic "Modifying split LOADFILE data set names" on page 146 was added and the topic Chapter 8, "Loading data into Db2," on page 139 was changed.	PH00879
Several changes were made to ALA objects to support Db2 12 function level 504. The topics "Worksheets: Gathering parameter values" on page 24, "Specifying a Quiet time report" on page 109, and "Product parameters, steps and tasks for Log Analysis Tool" on page 420 were updated and messages were added.	PH10288
Information was added to describe the Setup option from the Db2 Log Analysis Tool main menu. The topics "Db2 Log Analysis Tool setup options" on page 81, "Entering ZPARM, BSDS, and load library information" on page 81, and "Entering Db2 Log Analysis Tool specific information" on page 82 were added.	None

New messages

Description	Related APARs
The following messages were added or updated: ALAA032E, ALAA144E, ALAA163E, ALAA164E, ALAA171E, ALAA173E, ALAA178E, ALAA187E, ALAA194E, ALAA197E, ALAA320E-ALAA322E, ALAA329I, ALAA337E, ALAA345E, ALAA346W, ALAA360E, ALAA408E, ALAA419I, ALAA482E, ALAA498E, ALAA499E, ALAA519E, ALAA530E, ALAA539E, ALAA631E, ALAA634E, ALAA640E-ALAA662E, ALAA664E, ALAA667E, ALAA672E, ALAA678W, ALAA679W, ALAA689E-ALAA736I.	None
ALAA737E	PH01288

Description	Related APARs
ALAA738I-ALAA740I	PH11341
ALAA742W-ALAA750W	PH20156

Db2 12 function level support

When you activate new Db2 12 function levels in a Db2 subsystem or data sharing group, enhancements might become available that impact Db2 Log Analysis Tool.

The levels of function level support are defined as follows:

Tolerated
    The product works as it did on a previous release or function level of Db2 for z/OS, but it does not support the new features of this function level.

Supported
    The product supports most, but not necessarily all, of the new function-level features that IBM deems the most significant for this product.

The following function levels are tolerated or supported by Db2 Log Analysis Tool and are provided with the corresponding PTF, if any. For more information about a specific function level, click the link in the "Db2 function level" column.

Table 1. Db2 Log Analysis Tool PTFs in support of Db2 12 function levels		
Db2 12 function level	**Toleration PTF**	**Support PTF**
FL508	UI62472	UI69740
FL507	UI62472	UI69740
FL506	UI62472	UI69740
FL505	UI62472	UI69740
FL504	UI62472	UI69740
FL503	UI53769	UI53769
FL502	UI53769	UI53769
FL501	UI42137	UI42137
FL500	UI42137	UI42137

Db2 13 function level support

When you activate new Db2 13 function levels in a Db2 subsystem or data sharing group, enhancements might become available that impact IBM Db2 Log Analysis Tool for z/OS.

The levels of function level support are defined as follows:

Tolerated
    The product works as it did on a previous release or function level of Db2 for z/OS, but it does not support the new features of this function level.

Supported
    The product supports most, but not necessarily all, of the new function-level features that IBM deems the most significant for this product.

The following function levels are tolerated or supported by Db2 Log Analysis Tool and are provided with the corresponding PTF, if any.

Table 2. Db2 Log Analysis Tool PTFs in support of Db2 13 function levels		
Db2 13 function level	**Toleration PTF**	**Support PTF**
FL501	"UI80632" PTF.	"UI80632" PTF.
FL500	"UI80632" PTF.	"UI80632" PTF.

What does Log Analysis Tool do?

Log Analysis Tool provides you with a powerful tool to ensure high availability and complete control over data integrity. It allows you to monitor data changes by automatically building reports of changes that are made to database tables.

You can specify reports by various database resource criteria, such as date, user, or table. You can isolate accidental or undesired changes to your database tables.

With Log Analysis Tool, you can avoid the processing overhead associated with data change monitoring. Benefits include:

- Heightened confidence in data integrity
- Db2 for z/OS operates at optimal efficiency
- Maximum uptime for e-business availability

You can view data changes by dates, users, tables, and other criteria. Using this filtering process, you can view data, then obtain summary and detail reports that show the original state of the data and its current state, plus other valuable information. From this process, you may institute tighter controls over the data to ensure that it can no longer be compromised.

You can use Log Analysis Tool to perform a redo or undo operation without shutting down the system. Log Analysis Tool generates the SQL, which is sent to a file and can be executed by the DBA's preferred method.

Log Analysis Tool features and benefits

Log Analysis Tool provides solutions to many different types of problems.

Log Analysis Tool helps you to ensure high availability and complete control over data integrity. It allows you to monitor data changes by automatically building reports of changes that are made to database tables. Log Analysis Tool performs the following tasks:

Reporting

- View data changes by dates, users, tables, and other criteria.
- Create summary and detail reports that show:
  - Original state of the data.
  - Current state.
  - When data values have changed and who made the changes.
  - Change activity and see a history of changes by time, by user, and by application.
- Use the robust reporting capabilities, including reports on rollbacks, the ability to identify changes related to referential integrity, and expanded column data filtering.
- Specify which reports to generate when processing the log.
- Select from many options when applying the results of analyses.
- View current- and post-row image differences to let you easily identify changes that have occurred after the DML report was run.
- Report on all log records that are associated with units of recovery as soon as the commit (or abort) point for the UR is found.

- Use the "auto-continue" mode, enabling you to set report parameters once and then run any number of times with each subsequent run beginning in the log where the previous run stopped.

From this reporting, you can institute tighter controls over the data to ensure that it can no longer be compromised.

Recovery

- Generate SQL to undo or redo changes recorded in the log.
- Provide support for dropped object recoveries.
- Report on and recover data for dropped objects using both old and new Db2 identifiers.
- Write data to an SQL file or to a LOAD-format file.
- After DDL is recreated, restore the data in the regenerated table back to its state prior to the table being dropped.

Auditing

- Monitor and audit table activity.
    - Update, insert, delete and much more.
    - Determine who changed the data.
    - Determine the sequence of the changes.
- Validate SQL activity without traces – Is my program doing what I expect?
- Load reports into audit tables for review.

Replication

- Replay changes on another system or object using LOAD or REDO SQL.
- Use replication for data warehousing and internal processes.
- Populate test systems with production data for authentic application testing.

Performance
This robust tool also benefits you through improved performance and efficient processing:

- Run log reading in parallel, even with non-data sharing groups.
- Use a high-speed processing option to improve performance for sites using DATA CAPTURE CHANGES on tables.
- Archive output by running a detail report without incurring the overhead of the original run.
- Improve performance in an environment where large numbers of SQL statements are generated by Log Analysis Tool by running the SQL in batch using the Fast Apply processors.

Efficient processing

- Exploit multi-level security that provides authorizations down to the row level within a table.
- Store information on quiet times in Db2 tables for future reference.
- Use the "automatic" mode for generating detail reports that automatically selects the most efficient way to run reports (forward or backward).
- Automatically determine table space names.
- Write data from the Detail Report to a file in Db2 Load format, which can be loaded into a Db2 table where the data behind the Detail Report can be further analyzed using Db2 Web Query Tool.
- Dynamically allocate many data sets that were previously hardcoded in the JCL.

Db2 Log Analysis Tool and other Db2 Tools

Many Db2 tools provide database management features that are not available in Db2 itself or provide enhancements to capabilities built into Db2.

Log Analysis Tool seamlessly integrates its processing with a number of other tools, increasing its capabilities:

IBM Db2 Recovery Expert
> Db2 Recovery Expert is a self-managing backup and recovery solution that protects mission-critical data. Db2 Recovery Expert helps you to avoid accidental data loss or corruption by providing the fastest, least costly method of recovery when time is of the essence.

IBM Db2 Object Restore
> Db2 Object Restore allows recovery of valuable data assets even if they no longer exist in the Db2 catalog. Db2 Object Restore is an affordable, flexible, easy-to-use tool that provides faster recovery of data assets and keep lower database maintenance costs.

Backup and recovery solutions

IBM solutions help IT organizations maximize their investment in DB2® and IMS databases while staying on top of some of today's toughest IT challenges. Backup and recovery solutions can protect your data and lessen the negative impact that data loss can have on your business.

Backup and recovery is one of the most complicated areas of database management. Having the right resources to do a recovery is critical, and unfortunately in many cases, is not addressed until after data is already lost.

Database backup and recovery solutions include recovering from a dropped object to bouncing back from a major disaster, and everything in between. Recoveries that are done manually can be error prone, time consuming, and resource intensive.

Some of the questions you might face as a database administrator who is responsible for backup and recovery tasks include:

- Can a transaction be reversed or does the entire database have to be recovered?
- How can you determine which objects have been impacted?
- Do you have the necessary resources to recover to a point in time?
- Are you prepared for a disaster?
- Can you recover your subsystem?
- How much data are you willing to lose?

IBM Db2 Log Analysis Tool for z/OS enables organizations to monitor database changes by automatically building reports. It helps you to minimize Db2 recoveries by determining what data has changed and identifying points of recovery.

- Directly extracts information from Db2 logs and Db2 pages to identify who, what, when, where, and how data was changed
- Enables you to avoid running expensive Db2 traces
- Does not require DATA CAPTURE for any tables, but DATA CAPTURE is supported
- Helps you to identify quiet times for running utilities and for determining potential points of recovery
- Quickly isolates accidental or undesired changes to your database tables
- Allows easy UNDO of these undesired changes from an intuitive, menu-driven interface
- Allows easy REDO of desired changes from an intuitive, menu-driven interface
- Locates changes that are made to one database and copies the changes to another database
- Keeps your Db2 for z/OS systems at work even when large-scale corrections to data are required
- Minimizes downtime from conventional recovery sessions

- Maximizes uptime for e-business availability

Performance

Log Analysis Tool can be highly resource intensive, depending upon the filters you use running the product.

- The worst-case scenario is to run Log Analysis Tool with no filters and wide time ranges. This can cause potentially large amounts of Db2 log data to be read, depending on your site's overall activity. There is marginal impact to Db2 for this type of run, but overall I/O can impact your z/OS system.
- The best-case scenario is to run Log Analysis Tool with multiple filters (especially table filters) and small time ranges, minimizing I/O against your Db2 logs.

A heavily used Db2 subsystem may generate hundreds of cylinders of data each day, so specifying short time ranges within a specific day will result in the least impact on your overall system. In general and whenever possible, narrow the search criteria against the Db2 logs with the provided filters, and keep your date/time ranges short.

Constraints

As you plan to use Log Analysis Tool, consider the potential effects of the following constraints.

Limited support of ALTER TABLESPACE MOVE TABLE

Db2 function level (FL) 508 has introduced support for moving tables from deprecated table spaces to universal table spaces (UTS) using the ALTER TABLESPACE statement with a new MOVE TABLE option. The following format is used for this option:

```
ALTER TABLESPACE dbname.tsname MOVE TABLE tbowner.tbname TO TABLESPACE dbname.new-tsname
```

where:

dbname
　　Identifies the name of the database.

tsname
　　Identifies the table space name.

tbowner
　　Identifies the owner of the table to be moved to the new table space.

tbname
　　Identifies the table name to be moved to the new table space.

new-tsname
　　Specifies a new table space name.

Log Analysis Tool processes moved tables as usual if a time period after the alteration is chosen. If you want to report such objects before using the ALTER TABLESPACE MOVE TABLE statement, the dropped object mode should be used.

Processing moved objects through the alteration is not supported by Log Analysis Tool.

Using Db2 utilities with Log Analysis Tool

You cannot run REORG, or any Db2 utilities, on objects while Log Analysis Tool is reporting on them. It is perfectly acceptable to run these utilities on objects when Log Analysis Tool is not currently reporting on them.

Log-backward processing after LOAD REPLACE

Log Backward processing will be allowed when a LOAD REPLACE utility is found after the starting LOGPOINT if all of these restrictions are met:

- Only full row images are present.
- No compression is in use or KEEPDICTIONARY was used by the utility.
- Current row images are not requested.
- If the utility was run during the reporting time frame, it must have been run using the LOG(NO) parameter. If the utility was run after the reporting time frame then the LOG(YES) parameter will still allow log-backward processing for the current job.

Viewing ASCII and UNICODE data tables

Log Analysis Tool can handle data tables in EBCDIC, ASCII, and UNICODE formats, all internally without any user involvement. To display that information in a format other than hexadecimal, however, requires the installation of another product from IBM, OS/390® Support for UNICODE, and in particular its callable API. Without this product's API installed, Log Analysis Tool can handle the ASCII or UNICODE data, it just displays it in hexadecimal format. For more information on OS/390 Support for UNICODE and Log Analysis Tool, see "Using Log Analysis Tool with ASCII or UNICODE data tables" on page 19.

64-bit mode log restriction

When using Log Analysis Tool in 64-bit mode (one of the options for using one-phase mode) you are limited to reading 500 logs. If that limit is exceeded, an error occurs.

Temporal data: Generation of SQL for Temporal Data using the SYSTEM PERIOD and HISTORY Tables

Log Analysis Tool cannot currently generate SQL that will execute successfully for the SYSTEM PERIOD table because of Db2 restrictions on GENERATED ALWAYS TIMESTAMP columns. See Chapter 21, "Temporal Data," on page 251 for more information.

Referential integrity (RI)

Log Analysis Tool does not determine referential constraints, it only generates the appropriate SQL if it can from the set of filtered tables. The only way that Log Analysis Tool can generate the appropriate SQL is to provide all of the tables involved in the referential relationship. All tables involved in the RI *must* be included in the filters for the log analysis run, and all actions *must* be included (INSERTs, UPDATEs, DELETEs) if Log Analysis Tool is to properly generate SQL involved in the referential constraints.

For example, in a simple referential setup:

There is a parent table named P1. A child table, C1 is dependent of P1, with CASCADE DELETE rule.

A row in P1 is deleted, causing three dependent rows in C1 to be deleted.

If only C1 is included as a table filter, and undo SQL is requested, Log Analysis Tool generates three INSERTs to undo the 3 original DELETEs. These INSERTs will fail, however, unless the originally deleted row in P1 is first re-INSERTed. Thus, the filters must include P1 as well as C1.

LOB/XML support

Certain actions related to LOB/XML tracking are not currently supported. See Chapter 20, "LOB/XML support in SQL and LOAD operations," on page 247 for more information.

Concurrent image copies

Log Analysis Tool does not support image copies taken with the CONCURRENT option. Image copy access is often required during detail reporting in a log-forward mode. If such an image copy is required, processing will stop and an error message will be issued.

Data encryption

Log Analysis Tool supports the Db2 Encryption Tool EDITPROC and supports data compression with a fixed dictionary (linear compression). Data encryption and decryption using ENCRYPT_TDES/DECRYPT_* functions are not supported.

GENERATED ALWAYS columns

Columns identified as GENERATED ALWAYS can cause SQL execution failures. For example, assume you've requested REDO SQL on a table where a column named ROWID_KEY is defined as ROWID GENERATED ALWAYS, and this column is part of the generated WHERE clause. If an INSERT statement precedes an UPDATE on the same row, the UPDATE statement will fail because the product cannot determine what the value of the ROWID_KEY column is after the original INSERT. This value is needed as part of the WHERE clause on the ensuing UPDATE statement. The product attempts to use a key without using such a column, but it is not always possible (that is, if the table only has one unique key which includes the GENERATED ALWAYS column).

EBCDIC SQL only

If any table or column name cannot be translated to EBCDIC, SQL cannot be properly generated for that object. This product only generates EBCDIC SQL.

Cannot mimic application program logic

Log Analysis Tool cannot mimic application program logic. For example, if an application program exists that does UPDATE statements that are WHERE CURRENT OF CURSOR, the Db2 log merely records the fact that the update occurred to a particular row. Any SQL generated by this product cannot be performed against such a particular row; the WHERE clause only determines the row(s) against which the UPDATE will be performed. This can cause problems against non-uniquely keyed tables. Such an UPDATE statement, when redone, may update many rows. The same holds true for DELETE operations.

Limited support for table versioning

Log Analysis Tool provides support for table versioning where possible, however, due to the nature of table versioning the support is limited and might require manual intervention.

TRUNCATE TABLE statements

TRUNCATE TABLE, a new SQL statement provided in Db2 9, has only limited support. Due to the nature of TRUNCATE TABLE statements, Log Analysis Tool cannot differentiate between TRUNCATE TABLE statements and mass delete statements.

Table cloning

Db2 9 added the concept of table cloning. Support for cloned tables is not currently available. Neither the base table nor the clone table can be tracked by this product once the base table has been cloned.

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 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.

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:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Tools_for_z~OS

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 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:

http://www.ibm.com/support/docview.wss?uid=swg27020910

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:

https://www.ibm.com/analytics/us/en/db2/db2-for-zos/

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. TheDb2 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 helps IBM to provide quality information. Send any comments that you have about this book or other Db2 Tools documentation to comments@us.ibm.com. Include the name and version number of the product and the title and number of the book. If you are commenting on specific text, provide the location of the text (for example, a chapter, topic, or section title).

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.

Chapter 2. Preparing to customize Log Analysis Tool

Before you start to customize Log Analysis Tool for the first time, determine all of the customization values that you need to specify during the customization process, and familiarize yourself with all of the customization tasks.

The following checklist lists and describes each significant customization step. Use this checklist to guide you through the entire customization process.

Tip: Print the following checklist and the data set names and parameter values worksheets. Use the worksheets to record your values, and refer to them during the customization process.

Task	Link to detailed instructions	Status
Tools Customizer basics		
Prior to beginning the customization process, familiarize yourself with Tools Customizer terminology and data sets, and other basic information about Tools Customizer.	"Tools Customizer terminology" on page 405	
Software requirements		
Verify that your environment meets the minimum software requirements. To install and use Log Analysis Tool, your environment must be running a supported version of the z/OS operating system and of Db2 for z/OS. Additionally, certain levels of maintenance must be applied.	"Verify that your environment meets software requirements" on page 17	
SMP/E installation		
Verify that Log Analysis Tool has been installed correctly. Log Analysis Tool is installed by using standard SMP/E processing.	"Verify that Log Analysis Tool has been installed successfully" on page 18	
Verify that Tools Customizer for z/OS has been installed correctly. Tools Customizer for z/OS is installed by using standard SMP/E processing.	"Set up your environment prior to customization" on page 17	
Security requirements		
Make sure that you have the required authorizations to use Log Analysis Tool.	"Verify that your environment meets security requirements" on page 18	
Ensuring optimal performance		
If your site has a large number of objects in your catalog, you should create these recommended indexes only IF performance appears slow.	"Ensuring optimal performance" on page 19	
Data sharing support		
Log Analysis Tool supports the data sharing environment.	"Data sharing support" on page 19	

Task	Link to detailed instructions	Status
Write access to SALACLST		
Verify that the USERID performing the customization jobs has write access on the ALA.SALACLST data set.	"Ensure write access to the SALACLST data set" on page 19	
Working with ASCII and UNICODE data tables		
Verify that the callable API from z/OS Support for Unicode is installed and configured. Log Analysis Tool uses the IBM z/OS Support for Unicode product to process character conversions.	"Using Log Analysis Tool with ASCII or UNICODE data tables" on page 19	
Migrating to a new version of Log Analysis Tool		
Verify that you understand the migration process to the new version of Log Analysis Tool.	"Migrating to Log Analysis Tool V3.5" on page 20	
Upgrading to new version of Db2		
Verify that you understand the upgrade process to the new version of Db2.	"Upgrading versions of Db2" on page 22	
Gather data set names		
During the customization process, you must specify data set names for the following things: • Tools Customizer • FEC (common code) • Db2 Log Analysis Tool • Other (these are the dsnames for TCz setup -- Discover, data store, and so on.)	"Worksheets: Gathering required data set names" on page 22	
APF authorization		
The following data sets must be APF authorized: • SALALOAD • SFECLOAD	"APF authorizing load libraries" on page 24	
Gather parameter values		
During the customization process, you must specify parameter values for Log Analysis Tool, for Db2, and for your LPAR.	"Worksheets: Gathering parameter values" on page 24	
Customize Log Analysis Tool		
Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.	"Starting Tools Customizer" on page 38	

Task	Link to detailed instructions	Status
Set up Tools Customizer user settings. If you are running Tools Customizer for the first time, you must modify several user settings to suit your environment. Otherwise, if the user settings that you have already established are still appropriate, skip this step.	"Modifying Tools Customizer user settings" on page 39	
Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing.		
Customizing Log Analysis Tool for the first time Follow this roadmap if you do not have a customized version of Log Analysis Tool, and you need to customize it for the first time.	"Roadmap: Customizing Db2 Log Analysis Tool for the first time" on page 45	
Customizing a different version of Log Analysis Tool Follow this roadmap if you have already customized a version of Log Analysis Tool and you want to use the same parameter values to customize a different version.	"Roadmap: Customizing a new version of Db2 Log Analysis Tool from a previous customization" on page 46	
Recustomizing Log Analysis Tool Follow this roadmap if you have a customized version of Log Analysis Tool but you want to change one or more parameter values.	"Roadmap: Recustomizing Db2 Log Analysis Tool" on page 47	

Set up your environment prior to customization

Prior to beginning the customization process, ensure that your environment meets all requirements, that you have installed all prerequisite software, and that you have considered how you want to customize optional features.

Verify that your environment meets software requirements

Ensure that you are using z/OS V1.12 (5694-A01) or later.

Ensure that you are using one of the following supported versions of Db2 for z/OS:

- Db2 V9.1 (5635-Db2)
- Db2 Value Unit Edition V9.1 (5697-P12)
- Db2 V10 (5605-Db2)
- Db2 Value Unit Edition V10.1 (5697-P31)
- Db2 V11 (5615-Db2)
- Db2 Value Unit Edition V11.1

Additionally, make sure that the following FEC maintenance has been applied:

- FEC APAR PK43912

- FEC APAR PK76167
- FEC APAR PM06651 (required for Db2 V10 and later)

Verify that Log Analysis Tool has been installed successfully

See the Program Directory for IBM Log Analysis Tool for z/OS, GI10-8772-03 for installation instructions.

The IBM Tools Base for z/OS (5655-V93), which is available free of charge, provides a standard approach to customizing IBM Db2 for z/OS Tools. See the Program Directory for IBM Tools Base for z/OS, GI10-8819 for installation instructions.

Verify that your environment meets security requirements

The primary user interface to Log Analysis Tool is through its ISPF application, which has multi-factor authentication (MFA) support through TSO logon processing. All authentication is done before Log Analysis Tool is invoked and no additional MFA processing is necessary.

All batch jobs created by Log Analysis Tool and submitted for execution inherit the ID that has already been authenticated by the submitter. The Db2 services that are used by Db2 Log Analysis Tool are invoked using the standard attachment facilities.

Log Analysis Tool requires no extra security measures outside of standard Db2 security. If a user does not have authority to view a table within a Db2 subsystem, Log Analysis Tool will not allow the user to see data changes made to that table. Similarly, undo and redo SQL that generated from the product can be run through products such as SPUFI or QMF, and therefore also adheres to normal Db2 security for the user who runs this SQL.

You must have authorization to run the SELECT statement on the following tables:

- SYSIBM.SYSAUXRELS
- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSCOPY
- SYSIBM.SYSFIELDS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSKEYS
- SYSIBM.SYSKEYTARGETS
- SYSIBM.SYSTABLEPART
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSXMLRELS
- SYSIBM.SYSXMLSTRINGS

By default, Log Analysis Tool can run the REPORT utility against filtered objects. This activity is transparent to the user and can be disabled by setting the Misc Flags value on the general report panel to a value of X. If you do not disable this feature for any given run, you must have one of the following authorizations to access the REPORT utility through Log Analysis Tool:

- RECOVERDB privilege for the database
- DBADM or DBCTRL authority for the database
- SYSCTRL or SYSADM authority

Ensuring optimal performance

If your site has a large number of objects in your catalog, your performance may be decreased because Db2 offers no index on these fields. You should create these recommended indexes only IF performance appears slow during certain times of Log Analysis Tool general report processing.

Create index on SYSIBM.SYSTABLESPACE:

```
CREATE INDEX ixname ON SYSIBM.SYSTABLESPACE (DBID,PSID);
```

Create index on SYSIBM.SYSFIELDS:

```
CREATE INDEX ixname ON SYSIBM.SYSFIELDS (TBCREATOR, TBNAME);
```

Data sharing support

Log Analysis Tool automatically and transparently reads through multiple member logs for the needed log information in support of data sharing. If you specify a group name or any member's name as the SSID, all members' logs are read when Log Analysis Tool runs.

Ensure write access to the SALACLST data set

The user running the customization jobs must have write access on the ALA.SALACLST data set to complete the customization process successfully.

Using Log Analysis Tool with ASCII or UNICODE data tables

Log Analysis Tool requires no special action to process ASCII or UNICODE Db2 data (that is, data contained in Db2 table rows). The data must be converted to EBCDIC, however, to correctly display that data within Log Analysis Tool. For example, the detail report must display the data in EBCDIC format.

OS/390 Support for Unicode

Log Analysis Tool uses the IBM z/OS Support for Unicode product to process character conversions. In particular, Log Analysis Tool uses the callable API from z/OS Support for Unicode, but the API component used by Log Analysis Tool may not be installed or configured. Additionally, you may be running a version of z/OS where the z/OS Support for Unicode product is not yet available. If any of these issues exist, Log Analysis Tool cannot successfully perform the necessary character conversions. Log Analysis Tool will not abnormally end, but it will handle the data without conversion.

If the character conversion fails and a detail report must display the data, this format is used: X'nnnn'. This represents the data in hexadecimal format. This format will also be used in any generated SQL and it will successfully execute.

For example, an ASCII character column contains this ASCII data: 'ABC'. If the conversion was unsuccessful, this data would display as X'414243'. Obviously this is not the preferred format and the goal is to convert the actual data to EBCDIC.

Note: Even with this API available, there are cases where character conversion still fails. In those cases, the X'nnnn' format will still be used, but efforts should be made to ensure that your API is properly configured to handle all necessary conversions.

To ensure proper installation and configuration

The OS/390 Support for Unicode product is separate from Log Analysis Tool, and therefore has its own manuals and documentation.

The most current information (and even a download of the product if you require it) can be found on the IBM Web site. To locate this information:

1. Navigate to www.ibm.com, the IBM Web site.

2. Click **Support & downloads**.

3. Type `unicode` in the technical support search function and click **Go**.

4. Locate this link: **Installation steps for OS/390 Unicode to support OS/390 Wide APIs used in Db2**, or a similar one.

Note: The manual for installing and configuring the OS/390 Support for Unicode product is *OS/390 Support For Unicode: Using Conversion Services* and is IBM publication number SC33-7060.

Migrating to Log Analysis Tool V3.5

JCL

A general rule is that any JCL generated by any previous release of this product that has been saved off for repeated execution (for example, JCL generated by the product and entered into a job scheduler), may need to be regenerated using the new (V3.5) panels and saved. This is true for general report JCL, detail report JCL, and all the 'backend' options (loading details into Db2, and so on). You should consult the User's Guide for general descriptions of migration issues regards installation and configuration, as well.

If you are migrating from V2.3 or any V3.X release, JCL regeneration via the V3.5 panels is not required. That is, if migrating from V2.3 or any V3.X release, regeneration may not be necessary if you do not want to use the new V3.5 options. Two JCL modification steps are still required, however:

1. The DD MODEFILE must be converted to V3.5 format. If you do want to use the new V3.5 options, then you must go through the panels to regenerate both a new MODEFILE and new JCL to reflect those new options. Conversion of the MODEFILE is done via the "Utilities" primary menu option, and see the option "Mode file operations". Ensure that the newly converted MODEFILE is used in your V3.5 JCL.

2. A new DD for SYSPRINT was added, so replace the current SYSPRINT DD with the following in steps 2 (EXEC PGM=ALAGEN1) and 3 (EXEC PGM=ALADTL1):

```
//SYSPRINT DD    DSN=&LGRTEMP,
// DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
// SPACE=(CYL,(100,100),RLSE),DCB=(LRECL=23472,BLKSIZE=23476,RECFM=VB)
```

If migrating from a release prior to V2.3, you will need to regenerate the MODEFILE and JCL using V3.5 panels. You can use the "Mode file operations" option of the "Utilities" panel to determine the settings within your old MODEFILE. This can be helpful in determining your old settings, and what you wish to specify for your new settings.

MODEFILE conversion must be performed by all sites to use V3.5 that do not regenerate the JCL from the V3.5 panels. If you generate JCL from the V3.5 panels, Log Analysis Tool will generate the proper MODEFILE for you and no conversion is necessary.

Important: Under no circumstances can you merely change your previously saved off JCL from a release prior to V3.1 to point to the new V3.5 load libraries. At a minimum, you must convert the MODEFILE, and make the SYSPRINT DD change as described previously.

ROWDATA VSAM file

Log Analysis Tool V3.4 required a change to the ROWDATA VSAM file, which has existed in every prior release with a key length of 18. That key length is now 32. This change requires you to recreate the ROWDATA VSAM file if you continue to use two phase mode and do not make use of the dynamic allocation process. If saved JCL exists for prior releases of this product that includes any hardcoded DD ROWDATA statements, those statements themselves do not need to be modified, but the files they reference must be recreated. If you do not recreate those referenced files, you will have problems when trying to write data to the ROWDATA VSAM file. You can recreate these files by performing the following tasks:

1. Access the "User settings" primary menu, option '0' from the main menu.

2. Specify **R** (to recreate the file with the same name) for the "Create/Re-create ROWDATA VSAM file" option.

3. Complete the subsequent panel with the proper file size attributes (these can be the same as the previous file).

4. Submit the JCL.

This process recreates the file with the proper key length for Log Analysis Tool V3.4 and later.

- If migrating from a prior version and you do not use either one phase mode nor dynamic allocation in this release, then you must recreate your ROWDATA VSAM file.

- If the DD ROWDATA does not appear in your saved JCL from prior releases, then no action is necessary because Log Analysis Tool internally handles this file for you during the dynamic allocation process.

- If you bypass this migration step but it was required in your situation, an error message alerts you that Log Analysis Tool detected the improper ROWDATA VSAM file attributes. You must then perform the steps previously documented.

LOADFILEs

The LRSN, RBA, URID fields have been expanded from 6 bytes to 10 bytes or from 12 bytes to 20 bytes when stored in character format. If you have downstream programs that use the LOADFILE and you manually modify it, you must adjust the header portion to support the expanded LRSN/RBA.

Quiet time tables

The LRSN, RBA, URID fields have been expanded from 6 bytes to 10 bytes or from 12 bytes to 20 bytes when stored in character format. If you use a quiet time table, you must ALTER it to support the expanded LRSN/RBA and run UPDATE statements to convert it to the proper format.

Two-phase jobs

It is highly recommended, though optional, that you modify your two-phase jobs to use QSAM processing that was added to Log Analysis Tool V3.4:

1. Access the "User settings" primary menu, option '0' from the main menu.

2. Specify **Y** for the "Use QSAM for ROWDATA" option.

This setting improves performance for your two-phase jobs by switching to QSAM processing instead of the slower VSAM processing of previous releases. If you are using saved JCL from a prior release, you must compensate for this as well. Any hardcoded references to DD ROWDATA must be replaced with a QSAM data set name after the setting above has been specified. If prior saved JCL does not reference DD ROWDATA, then no action is required. The new attributes of the QSAM for ROWDATA data set can be viewed by generating JCL in Log Analysis Tool V3.4 or later while bypassing dynamic allocation.

Catalog Audit Report

Log Analysis Tool v3.5 no longer requires the SQL to be generated before producing the Catalog Audit Report, so you may want to regenerate your JCL and turn off SQL generation for better performance.

If you want to continue using JCL from ALA V3.4 that produced the Catalog Audit Report, you must make the following JCL modifications or regenerate your JCL.

- If using 1 Phase 64 bit or 1 Phase Work File mode, no JCL modifications are required.

- The following JCL modifications are required for 2 Phase processing:

  - If using the option "Write reports to datasets................... ", then the following actions will need to be done:

    1. The step in the JCL that executes ALADTL3 will need to have the following DD added to it: //
       CATRPT DD SYSOUT=*

    2. The same DD will need to be removed from the step that executes ALADTL4.

  - If using option "Write reports to datasets................... ", then the following actions will also need to be done:

1. In the step before ALADTL1 add the following to the list of reports that should be deleted before creating them again. If there is not an IEFBR14 step, then add the whole step:

```
//STEP3A   EXEC PGM=IEFBR14,COND=(4,LT)
//CATRPT   DD   DSN=&SYSUID..CATRPT,
//
DCB=(LRECL=80,BLKSIZE=8000,RECFM=FBA),DISP=(MOD,DELETE,DELETE),
// SPACE=(TRK,(1)),UNIT=SYSDA
```

2. In the step that executes ALADTL3, add the following :

```
//CATRPT   DD   DSN=&SYSUID..CATRPT,
//
DCB=(LRECL=80,BLKSIZE=8000,RECFM=FBA),DISP=(NEW,CATLG,CATLG),
// SPACE=(CYL,(10,10),RLSE),UNIT=SYSDA
```

3. The same DD will need to be removed from the step that executes ALADTL4.

Upgrading versions of Db2

If you are upgrading to a higher release of Db2, both new installations and existing installations of Log Analysis Tool require a rebind of the packages and plans using the appropriate Db2 library for the release you are upgrading to. You must use the Tools Customizer to regenerate the BIND jobs.

If you are upgrading modes of Db2 (for example, from Db2 Vx CM to Db2 Vx NFM), simply rebind the Log Analysis Tool packages and plans after upgrading from one mode to the next.

Worksheets: Gathering required data set names

Identify and record the data set names that will be used during the customization process and make sure that requirements for certain data sets are met.

Data set names for Tools Customizer

Identify and record the following Tools Customizer data set names:

Data set name	Description	Special requirements	Your data set name
SCCQDENU	Metadata library for Tools Customizer		
SCCQLOAD	Executable load module library for Tools Customizer		
SCCQMENU	ISPF messages for Tools Customizer		
SCCQPENU	ISPF panels for Tools Customizer		
SCCQSAMP	Sample members for Tools Customizer		
SCCQTENU	Table library for Tools Customizer	You must have write access to this data set.	

Data set names of Log Analysis Tool

Identify and record the following Log Analysis Tool data set names (ALA as HLQ). During the customization process, you will enter the following values on panel CCQPPRD.

Data set name	Description	Special requirements	Your data set name
SALADBRM	DBRM library for Log Analysis Tool		

Data set name	Description	Special requirements	Your data set name
SALALOAD	Executable load module library for Log Analysis Tool	You must APF authorize this data set.	
SALAMENU	ISPF messages for Log Analysis Tool		
SALAPENU	ISPF panels for Log Analysis Tool		
SALASAMP	Sample members for Log Analysis Tool		
SALACLST	Table library for Tools Customizer	You must have write access to this data set.	
SALADENU	Metadata library for Log Analysis Tool		
SALASLIB	ISPF skeleton library for Log Analysis Tool		

Data set names of FEC (common code)

Identify and record the following FEC data set names (FEC as HLQ). During the customization process, you will enter the following values on panel CCQPPRD.

Data set name	Description	Special requirements	Your data set name
SFECMENU	FEC ISPF messages for Log Analysis Tool		
SFECLOAD	FEC executable load module library for Log Analysis Tool	You must APF authorize this data set.	

Data set names of other libraries

Identify and record the following data set names. During the customization process, you will enter the following values on the Setup panel.

Data set name	Description	Special requirements	Your data set name
Discover output data set	Contains the output that is generated when you run the Log Analysis Tool Discover EXEC. The Log Analysis Tool Discover EXEC retrieves the metadata and values for the parameters from a previous customization of Log Analysis Tool. 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.	You must have write access to this data set.	

Data set name	Description	Special requirements	Your data set name
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, and LPAR when you customize products on the local LPAR. The default name of the data set is DB2TOOL.CCQ110. DATASTOR. You can change the default value on the Tools Customizer Settings panel.	You must have write access to this data set.	
Product customization Library	Contains the customization jobs that Tools Customizer generates for Log Analysis Tool. To customize Log Analysis Tool, 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: `hlq.$LPAR-name$.xyzvrm` where: • *hlq* is the value of the Customization library qualifier field on the Tools Customizer Settings panel (CCQPSET) • *LPAR-name* is the four-character LPAR name • *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.	You must have write access to this data set.	

APF authorizing load libraries

Some of the programs in Log Analysis Tool load libraries must be APF-authorized to run.

Procedure

Include the following load libraries as part of your authorized list:

• SALALOAD
• SFECLOAD

Worksheets: Gathering parameter values

During the customization process, you will need to provide parameter values for Log Analysis Tool, for Db2, and for your LPAR.

Use the worksheets in this topic to record the appropriate parameter settings for your purposes, and then use these worksheets during the customization process. The worksheets are organized based on the order of the customization panels in the product.

Customization values for the Discover EXEC

Description

Use the following worksheet to identify and record the customization values for the Discover EXEC. The values in this worksheet are for extracting information from a product that has already been customized. During the customization process, you will enter these values on panel CCQPDSC.

Note: Complete this worksheet only if you are re-customizing a product that has previously been customized.

Parameter	Sample or default value	Your value
Discover EXEC for Extracting Information from an Already Customized product		
Discover EXEC library The fully qualified data set name that contains the product Discover EXEC.	The name of the Discover EXEC Library that you entered on the settings panel.	
Discover EXEC name The name of the Discover EXEC.	ALADISC	
Discover output data set The name of the data set for the output from the product Discover EXEC.	The name of the discover output library that you entered on the settings panel.	
New ALA load library The load library name for the new release of Log Analysis Tool. For example, if this release is V3.5, the library name might be ALA.SALALOAD or ALA.LOADLIB (using a generic message library name) or ALA.V350.SALALOAD or ALA.V350.LOADLIB (using a release specific library name).	ALA.V350.SALALOAD	
Information for Discover EXEC section		
New ALA ISPM library The message library name for the new release of Log Analysis Tool. For example, if this release is V3.5, the library name might be ALA.ALAMENU or ALA.ISPMLIB (using a generic message library name) or ALA.V350.ALAMENU or ALA.V350.ISPMLIB (using a release specific library name).	ALA.V350.SALAMENU	

Parameter	Sample or default value	Your value
Discover from previous ALA control file ALA control file name to discover from.	ALA.V330.CONTROL	
Discover from previous ALA CLIST library ALA CLIST library to discover from.	ALA.V330.SALACLST	
Discover from previous ALA SAMPLIB library ALA SAMPLIB library to discover from.	ALA.V330.SALASAMP (ALABIND)	

Product to Customize

Description

The parameters that are listed in the "Product to Customize" section are read-only. They contain information that was provided on other panels or by the Log Analysis Tool metadata data set.

Parameter	Discovered?	Source of this value
Product metadata library This value is the library that you specified on the **Specify the Product to Customize** panel. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.	No	This value is specified on the **Specify the Product to Customize** panel (CCQPHLQ)
LPAR The **LPAR** field displays the LPAR on which you are customizing Log Analysis Tool.	No	This value is supplied by Log Analysis Tool.
Product name This value displays the product that is being customized. In this example, IBM Log Analysis Tool should be displayed in this field. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.	No	This value is provided by the product metadata file.
Version The **Version** field displays the version, release and maintenance of the product you are customizing in the format V*n*.R*n*.*nn*.	No	This value is provided by the product metadata file.
Product customization library This value displays the name of the data set in which the generated library customization jobs will be stored.	No	This value is derived from the user-specified customization library qualifier on the **Tools Customizer Settings** panel (CCQPSET).

Required parameters section

Description

The parameters in this task are required for all customizations. During the customization process, you will enter these values on panel CCQPPRD.

Parameter	Required?	Discovered?	Default value	Your value
Control file name The name of the control file that will be used for this installation or release of Log Analysis Tool. The control file contains configuration information for each Db2 subsystem against which the product can run.	Yes	No	ALA.CONTROL	
Qualifier for unqualified tables The qualifier for unqualified tables. This qualifier is used in the bind of Log Analysis Tool SQL within the QUALIFIER parameter.	Yes	Yes	SYSTOOLS	
CLIST for ALA The CLIST library name for Log Analysis Tool.	Yes	No	ALA.SALACLST	
ALA load library The name of the Log Analysis Tool load library.	Yes	No	ALA.SALALOAD	
FEC load library The name of the FEC load library.	Yes	Yes	FEC.SFECLOAD	
ALA ISPMLIB The name of the Log Analysis Tool ISPMLIB.	Yes	Yes	ALA.SALAMENU	

Task: ALA control file

Description

This task creates the control file if it does not exist or if you are creating a new control file for a new release. The control file contains specific information about each Db2 subsystem where Log Analysis Tool might run. During the customization process, you will enter these values on panel CCQPPRD.

This task is *optional*.

Jobs generated

This task generates the A0CNTL1 job. This job is based on the ALACNTL1 template.

Required authority

Create authority on the ALA control file.

Step or parameter	Required?	Discovered?	Default value	Your value
ALA control file creation This step creates the control file if it does not exist or if you are creating a new control file for a new release. The control file contains specific information about each Db2 subsystem where Log Analysis Tool might run.	No	-	Selected	
Volume serial number for the control file The volume serial number (VOLSER) to be used for the control file creation. May be left as blanks for SMS shops.	No	No	blank	

Task: ALA Fast Apply DDL

Description

This task creates the objects that are required to run Fast Apply on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation. During the customization process, you will enter these values on panel CCQPPRD.

This task is *optional*.

Jobs generated

This task generates the A1FAS1*nn* job. This job is based on the ALAFAS1 template.

Required authority

The user ID that runs the A1FAS1*nn* job must have create authority on ALA Db2 objects.

Step or parameter	Required?	Discovered?	Default value	Your value
ALA Fast Apply DDL This step creates the objects that are required to run Fast Apply on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation.	No	-	Selected	
ALA database name This parameter indicates the database name that is used to store Log Analysis Tool objects.	Yes	No	ALADB01	
ALA table space name This parameter indicates the table space name that is used to store Log Analysis Tool objects.	Yes	No	ALATS01	

Task: ALA profiles DDL

Description

This task creates the objects that are required to use profiles on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation. Before this DDL can be run, the Log Analysis Tool Fast Apply DDL must have been run, in this installation or a previous installation. During the customization process, you will enter these values on panel CCQPPRD.

This task is *optional*.

Jobs generated
>This task generates the A2PROF*nn* job. This job is based on the ALAPROF1 template.

Required authority
>The user ID that runs the A2PROF*nn* job must have create authority on ALA Db2 objects.

Step or parameter	Required?	Discovered?	Default value	Your value
ALA Profiles DDL This step creates the objects that are required to use profiles on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation. Before this DDL can be run, the Db2 Log Analysis Tool Fast Apply DDL must have been run, in this installation or a previous installation.	No	-	Selected	
ALA database name This parameter indicates the database name that is used to store Log Analysis Tool objects.	Yes	No	ALADB01	
ALA profiles table space name This parameter indicates the table space name that is used to store the Log Analysis Tool profiles table.	Yes	No	ALATS02	
ALA profile attributes table space name This parameter indicates the table space name that is used to store the Log Analysis Tool profile attributes table.	Yes	No	ALATS03	

Task: ALA bind SQL

Description
>This task binds the ALA SQL. During the customization process, you will enter these values on panel CCQPPRD.

>This task is *required*.

Jobs generated
>This task generates the A3BIND*nn* job. This job is based on the ALABIND1 template.

Required authority
>The user ID that runs the A3BIND*nn* job must have bind authority on ALA Db2 plans.

Step or parameter	Required?	Discovered?	Default value	Your value
ALA bind This step binds the ALA SQL.	Yes	-	Selected	
DBRMLIB for ALA This parameter indicates the DBRM library name for Log Analysis Tool.	Yes	Yes	ALA.SALADBRM	

Step or parameter	Required?	Discovered?	Default value	Your value
User ID for ALA Bind This parameter indicates the user ID for the owner of the plan. The user ID value is used during the bind on the OWNER(userid) parameter.	Yes	Yes	USERID	
Collection name for ALA bind This parameter indicates the collection name to be used within the Log Analysis Tool bind.	Yes	Yes	ALACOLL	

Task: Edit startup CLIST

Description
This task binds the ALA SQL. During the customization process, you will enter these values on panel CCQPPRD.

This task is *required*.

Jobs generated
This task generates the A4CLST1 job. This job is based on the ALACLST1 template.

Required authority
The user ID that runs the A4CLST1 job must have update authority on the ALA CLIST library.

Step or parameter	Required?	Discovered ?	Default value	Your value
Edit startup CLIST This step edits the startup CLIST to provide proper dsnames for your site.	Yes	-	Selected	
ALA ISPPLIB This parameter indicates the name of the Log Analysis Tool ISPPLIB.	Yes	No	ALA.SALAPENU	
FEC ISPPLIB This parameter indicates the name of the FEC ISPPLIB.	Yes	No	FEC.SFECPENU	
FEC ISPMLIB This parameter indicates the name of the FEC ISPMLIB.	Yes	No	FEC.SFECMENU	
ALA ISPSLIB This parameter indicates the name of the Log Analysis Tool ISPSLIB.	Yes	No	ALA.SALASLIB	

Task: ALA configure Db2

Description
This task configures each Db2 subsystem within the control file.

This task is optional in the sense that you can either perform this task for all your Db2 subsystems (or any subset of them), or you can perform this same task using option #11 ("Setup") from the Log Analysis Tool main menu to configure each Db2 subsystem individually, as needed. A Db2 subsystem must be configured using one of these methods before it can be used with Log Analysis Tool.

During the customization process, you will enter these values on panel CCQPPRD.

This task is *required / optional*.

Jobs generated
> This task generates the A5LOAD*nn* job. This job is based on the ALALOAD1 template.

Required authority
> The user ID that runs the A5LOAD*nn* job must have update authority on the ALA control file.

Step or parameter	Required?	Discovered ?	Default value	Your value
ALA configure Db2 This step configures each Db2 subsystem within the control file. It is optional in the sense that you can either perform this task here for all your Db2 subsystems (or any subset thereof), or you can perform this same task using option #11 ("Setup") from the Log Analysis Tool main menu to configure each Db2 subsystem individually, as needed. A Db2 subsystem must be configured using one of these methods before it can be used within Log Analysis Tool.	No	-	Not selected	

Task: ALA ALTER DDL

Description
> This task alters tables that were created in previous releases as part of the 'Load data into Db2'. During the customization process, you will enter these values on panel CCQPPRD.

> This task is *optional*.

Jobs generated
> This task generates the A6LT1*nn* job. This job is based on the ALALT1 template.

Required authority
> The user ID that runs the A6LT1*nn* job must have alter authority on ALA Db2 objects.

Step	Required?	Discovered ?	Default value	Your value
ALA ALTER DDL This step alters tables that were created in previous releases as part of the 'Load data into Db2'.	No	-	Not selected	

Task: Launchpad

Description
> This two-part task adds Log Analysis Tool to the Db2 Administration Tool Launchpad. During the customization process, you will enter these values on panel CCQPPRD.

> This task is *optional*.

Jobs generated
> This task generates the A7ADBI job, which is based on the ALAADBI template, and the A8ADBI2 job, which is based on the ALAADBI2 template.

Required authority
The user ID that runs the A7ADBI job and the A8ADBI2 job must have update authority on the ALA CLIST library and Db2 Admin Tool Launchpad.

Step or parameter	Required?	Discovered ?	Default value	Your value
Launchpad Part 1 modifies the REXX EXEC to add Log Analysis Tool to the Db2 Administration Tool Launchpad. Part 2 runs that REXX EXEC and adds Log Analysis Tool to that Launchpad.	No	-	Not selected	

Db2 Parameters section

Description
This section contains Db2 parameters. All parameters are required. During the customization process, you will enter these values on panel CCQPDB2.

Parameter	Required?	Discovered?	Default value	Your value
Mode This parameter indicates the mode in which the Db2 subsystem is running. The following values are valid: • CM is compatibility mode on all listed Db2 versions except Db2 10. • CM8 is conversion mode from Db2 V8 on Db2 10. • CM9 is conversion mode from Db2 Version 9.1 on Db2 10. • NFM is new-function mode on all listed Db2 versions.	Yes	No	NFM	
Level number This parameter indicates the version, release, and modification level of the Db2 subsystem. The following values are valid: • 910 is valid only for CM or NFM. • 101 is valid only for CM8, CM9, or NFM. • 111 is valid only for CM or NFM. • 121 is valid only for CM or NFM.	Yes	No	blank	
Load library This parameter indicates the data set name of the Db2 load library.	Yes	Yes	DSN. SDSNLOAD	
Run library This parameter indicates the data set name of the Db2 run library.	Yes	Yes	DSN.RUNLIB. LOAD	

Parameter	Required?	Discovered?	Default value	Your value
Exit library This parameter indicates the data set name of the Db2 exit library.	Yes	Yes	DSN.SDSNEXIT	
Bootstrap data set This parameter indicates the name of the Db2 bootstrap data set.	Yes	Yes	DSN.SDSNBSDS	
Name of the 4 KB bufferpool This parameter indicates the name of the 4 KB bufferpool to be used for customization. The value must be 8 characters or less.	Yes	No	BP0	
Plan name for the DSNTEP2 utility This parameter indicates the plan name for the DSNTEP2 utility. The value must be 8 characters or less.	Yes	No	DSNTEP2	
Storage group name This parameter indicates the name of the storage group that will be used for creating Db2 objects for customization. The value must be 128 characters or less.	Yes	No	SYSTOOLS	
SQLID used while creating Db2 objects This ID is used in installation JCL while creating or altering Db2 objects used by Log Analysis Tool.	Yes	No	SESSION_USER	
Plan name for ALA bind This parameter indicates the plan name to be used on the Log Analysis Tool bind.	Yes	Yes	ALASQLP	
ZPARM module name This parameter indicates the name of the ZPARM module for this specific Db2 subsystem. This module typically resides in the Db2 SDSNEXIT library, and the name is typically in *ssid*PARM format. For example, if your Db2 SSID is D91A, the name of the ZPARM module might be D91APARM.	Yes	Yes	ZPARM	

Parameter	Required?	Discovered?	Default value	Your value

Use of archive logs 1 and 2
This parameter determines how you want Log Analysis Tool to process your archive logs. The field combines the settings for both ARCHLOG1 and ARCHLOG2. The first character is for ARCHLOG1, and the second character is for ARCHLOG2. The following values are permissible:
 	
YN
    Use ARCHLOG1 only; ARCHLOG2 is ignored.
 	
NY
    Use ARCHLOG2 only; ARCHLOG1 is ignored.
 	
PN
    ARCHLOG1 has priority over ARCHLOG2.
 	
NP
    ARCHLOG2 has priority over ARCHLOG1.
 	
NN
    No special considerations are made for either ARCHLOG1 or ARCHLOG2.
	Yes	Yes	NN	

Active log priority Specify Y to indicate that Log Analysis Tool will process the active log when data resides on both active and archive logs.	Yes	Yes	Y	

LPAR Parameters section

Description
This section contains LPAR parameters. All parameters are required. During the customization process, you will enter these values on panel CCQPLPR.

Parameter	Required?	Discovered?	Default value	Your value
Message library	Yes	No	ISP.SISPMENU	
ISPF table input library	Yes	No	ISP.SISPTENU	

Chapter 3. 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 Log Analysis Tool.

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 , 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`

- SCCQPENU
- 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:

- `DB2TOOL.PTF420.SINZDBRM.RSU1406 --> TCZ.R420.SINZDBRM`
- `DB2TOOL.PTF420.SINZDENU.RSU1406 --> TCZ.R420.SINZDENU`
- `DB2TOOL.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 Log Analysis Tool.

Modifying Tools Customizer user settings

Before you can customize Db2 Log Analysis Tool 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 Log Analysis Tool 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:

```
CCQPSET                    Tools Customizer Settings                   14:03:51
Command ===>
Enter the settings for customizing a product or press End to save and exit.

Commands: SAVE  OPTIONS

 Product Customization Settings

  *Customization library qualifier . . DB2TOOL.PRODUCT.CUST
     Volume serial  . . . . . . . . .
  *Use DB2 group attach  . . . . . . . YES (YES/NO)

 Tools Customizer Library Settings

  *Metadata library . . . . . DB2TOOL.CCQ110.SCCQDENU
  *Discover output data set . DB2TOOL.CCQ110.DISCOVER
     Volume serial  . . . . .
  *Data store data set  . . . DB2TOOL.CCQ110.DATASTO1
     Volume serial  . . . . .

 User Job Card Settings for Customization Jobs

  ===> //          JOB
(ACCOUNT),'NAME',
```

Figure 1. The **Tools Customizer Settings** panel (CCQPSET)

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 Log Analysis Tool 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
   Db2 Log Analysis Tool does not support Db2 group attach names. You must specify NO in 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 Log Analysis Tool 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 Log Analysis Tool or to change parameter settings.
Related concepts
Customizing Db2 Log Analysis Tool
Using Tools Customizer to customize Db2 Log Analysis Tool consists of identifying the product to customize; defining any required Db2 Log Analysis Tool, 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.

```
CCQPOP1                     Miscellaneous Options
Command ===>
--------------------------------------------------------

Enter a / to select options and press Enter. To cancel, press End.

Panel Display Options
  /   Show the panel instructions
  /   Show the Product to Customize section
  /   Show the Usage Notes section

User Profile Sharing Option
  _   Copy user profile to another data set
```

*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.

Chapter 4. Customizing Db2 Log Analysis Tool

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

Customization roadmaps describe the steps that you must complete to customize Db2 Log Analysis Tool. 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 3. Customization roadmaps

Environment description	Roadmap
You do not have a customized version of Db2 Log Analysis Tool, and you need to customize it for the first time.	"Roadmap: Customizing Db2 Log Analysis Tool for the first time" on page 45
You have already customized a version of Db2 Log Analysis Tool, and you want to use the same parameter values to customize a different version.	"Roadmap: Customizing a new version of Db2 Log Analysis Tool from a previous customization" on page 46
You have a customized version of Db2 Log Analysis Tool, but you want to change one or more parameter values.	"Roadmap: Recustomizing Db2 Log Analysis Tool" on page 47

Roadmap: Customizing Db2 Log Analysis Tool for the first time

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

If you are customizing a previous version of Db2 Log Analysis Tool, see "Roadmap: Customizing a new version of Db2 Log Analysis Tool from a previous customization" on page 46.

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 Log Analysis Tool for the first time.

Table 4. Steps for customizing Db2 Log Analysis Tool for the first time

Step	Description	Instructions
1	Specify the product metadata library for the product that you want to customize. The name of this library is *hlq*.SALADENU.	"Specifying the metadata library for the product to customize" on page 48
2	Create new Db2 entries and associate them with Db2 Log Analysis Tool.	"Creating and associating Db2 entries" on page 51
3	Define the required parameters.	"Defining parameters" on page 53
4	Generate the customization jobs for the product or for the Db2 entries on which Db2 Log Analysis Tool is ready to be customized.	"Generating customization jobs" on page 58

Table 4. Steps for customizing Db2 Log Analysis Tool for the first time (continued)

Step	Description	Instructions
5	Submit the generated customization jobs.	"Submitting customization jobs" on page 59

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

Table 5. Administrative tasks

Description	Instructions
Browse the different types of parameters.	"Browsing parameters" on page 61
Copy an existing Db2 entry to the list of Db2 entries on which Db2 Log Analysis Tool can be customized.	"Copying Db2 entries" on page 61
Remove one or more Db2 entries from the associated list.	"Removing Db2 entries" on page 62
Delete one or more Db2 entries from the master list.	"Deleting Db2 entries" on page 63
Display a list of customization jobs that have been previously generated.	"Displaying customization jobs" on page 63
Maintain the customization jobs in the customization library.	"Maintaining customization jobs" on page 64

Roadmap: Customizing a new version of Db2 Log Analysis Tool from a previous customization

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

Use this roadmap even if the previous version of Db2 Log Analysis Tool 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 Log Analysis Tool from a previous customization.

Table 6. Steps for customizing a new version of Db2 Log Analysis Tool from a previous customization

Step	Description	Instructions
1	Specify the product metadata library for the product that you want to customize. The name of this library is *hlq*.SALADENU.	"Specifying the metadata library for the product to customize" on page 48
2	Use the Db2 Log Analysis Tool Discover EXEC to discover information about the version of Db2 Log Analysis Tool that you previously customized manually.	"Discovering Db2 Log Analysis Tool information automatically" on page 49
3	Define the required parameters.	"Defining parameters" on page 53

Table 6. Steps for customizing a new version of Db2 Log Analysis Tool from a previous customization (continued)

Step	Description	Instructions
4	Generate the customization jobs for the product or for the Db2 entries on which Db2 Log Analysis Tool is ready to be customized.	"Generating customization jobs" on page 58
5	Submit the generated customization jobs.	"Submitting customization jobs" on page 59

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

Table 7. Administrative tasks

Description	Instructions
Browse the different types of parameters.	"Browsing parameters" on page 61
Copy an existing Db2 entry to the list of Db2 entries on which Db2 Log Analysis Tool can be customized.	"Copying Db2 entries" on page 61
Remove one or more Db2 entries from the associated list.	"Removing Db2 entries" on page 62
Delete one or more Db2 entries from the master list.	"Deleting Db2 entries" on page 63
Display a list of customization jobs that have been previously generated.	"Displaying customization jobs" on page 63
Maintain the customization jobs in the customization library.	"Maintaining customization jobs" on page 64

Roadmap: Recustomizing Db2 Log Analysis Tool

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for Db2 Log Analysis Tool 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 Log Analysis Tool.

Table 8. Required steps for recustomizing Db2 Log Analysis Tool

Step	Description	Instructions
1	Specify the product metadata library for the product that you want to recustomize. The name of this library is *hlq*.SALADENU.	"Specifying the metadata library for the product to customize" on page 48

Table 8. Required steps for recustomizing Db2 Log Analysis Tool (continued)

Step	Description	Instructions
2	Edit the specific tasks, steps, or parameters that need to be changed.	• "Defining Db2 Log Analysis Tool parameters" on page 53 • "Defining LPAR parameters" on page 55 • "Defining Db2 parameters" on page 57
3	Generate the customization jobs for the product or for the Db2 entries on which Db2 Log Analysis Tool is ready to be customized.	"Generating customization jobs" on page 58
4	Submit the new generated customization jobs.	"Submitting customization jobs" on page 59

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

Table 9. Administrative tasks

Description	Instructions
Browse the different types of parameters.	"Browsing parameters" on page 61
Copy an existing Db2 entry to the list of Db2 entries on which Db2 Log Analysis Tool can be customized.	"Copying Db2 entries" on page 61
Remove one or more Db2 entries from the associated list.	"Removing Db2 entries" on page 62
Delete one or more Db2 entries from the master list.	"Deleting Db2 entries" on page 63
Display a list of customization jobs that have been previously generated.	"Displaying customization jobs" on page 63
Maintain the customization jobs in the customization library.	"Maintaining customization jobs" on page 64

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 Log Analysis Tool. This information controls what is displayed on the **Product Parameters** panel, the **LPAR Parameters panel**, and the **DB2 Parameters** panel.

After Db2 Log Analysis Tool has been SMP/E installed, the default name of the product metadata library is *high_level_qualifier*.SALADENU, 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:

```
CCQPHLQ            Specify the Product or Pack Metadata Library         15:12:22
 Command ===>                                              Scroll ===> PAGE

 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 . ALA.PRD00.SALADENU

 Name                      Version  Metadata
Library

 =>
 =>
 =>
 =>
```

*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 Log Analysis Tool 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 Log Analysis Tool 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 Log Analysis Tool for the first time
   Do not run the Db2 Log Analysis Tool 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 Log Analysis Tool parameters.

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

Discovering Db2 Log Analysis Tool information automatically

You can use the Db2 Log Analysis Tool Discover EXEC to discover information from a previous or current customization of Db2 Log Analysis Tool.

About this task

Tip: Using the Db2 Log Analysis Tool 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 Log Analysis Tool provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on Db2 Log Analysis Tool.

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 Log Analysis Tool 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:

```
CCQPDSC              Discover Customized Product Information           16:38:37
 Command ===>                                              Scroll ===> PAGE

 For the product you are customizing, the Discover EXEC retrieves product
 information from an already customized product. Specify the required
 information, and press Enter to run the Discover EXEC. Press End to cancel.

 Product to Customize
   Product metadata library : ALA.PRD00.SALADENU      > LPAR. . . : RS22
   Product name . . . . . . : DB2 Log Analysis Tool   > Version . : .0

                                                          More:     +
Discover EXEC for Extracting Information from an Already Customized Product
 Discover EXEC library  . . . ALA.PRD0330.SALADENU
 Discover EXEC name . . . . . ALADISC
 Discover output data set . . ALA.TCZ.DISCOVER

 Information for Discover EXEC
  Current ALA LOAD library  . . . . . . . . ALA.SALALOAD                  >
  Current ALA ISPM library  . . . . . . . . ALA.SALAISPM                  >
  Discover from this ALA control file . . . ALA.CONTROL                   >
  Discover from this ALA CLIST library  . . ALA.SALACLST                  >
  Discover from this BIND JCL library and member
                                   ALA.SALASAMP(ALABIND)       >
```

*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 Log Analysis Tool Discover EXEC.

   Discover EXEC name
   > The name of the Db2 Log Analysis Tool Discover EXEC.

   Discover output data set
   > The fully qualified data set where output from the Db2 Log Analysis Tool 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 Log Analysis Tool. They show the information that is required to run the Db2 Log Analysis Tool Discover EXEC.

4. Issue the RUN command to run the Db2 Log Analysis Tool Discover EXEC.

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

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

Results

The discovered parameter values for Db2 Log Analysis Tool 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 Log Analysis Tool on new Db2 entries, create and associate the entries.
- If Db2 entries were discovered and you want to customize Db2 Log Analysis Tool 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 Log Analysis Tool.

Defining parameters
To customize Db2 Log Analysis Tool, you must define Db2 Log Analysis Tool 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 Log Analysis Tool.

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:

```
CCQPDAD                   Associate DB2 Entry for Product               16:36:42
Command ===>                                            Scroll ===> PAGE

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands:  CREATE - Create new DB2 entries

DB2 Entries
  Line commands: A - Associate  C - Copy  D - Delete
  Cmd SSID GrpAttch
      *    *
---------------------------- End of DB2 entries -----------------------------
```

   *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 master 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.

   The **Create DB2 Entries** panel is displayed, as shown in the following figure:

```
CCQPCDB                       Create DB2 Entries                     Row 1 of 1
 Command ===>                                                  Scroll ===> PAGE

 Specify the SSID, the group attach name, or both in the appropriate columns
 for each new DB2 entry and press Enter. To create additional entries, issue
 the Inn line command, where nn is the number of entries to be inserted, and
 press Enter. To cancel, press End.

 New DB2 Entries
   Line commands: I - Insert into list   R - Remove from list
   Cmd  SSID  GrpAttch  Message

 --------------------------- End of DB2 entries ---------------------------
```

*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 I*nn* 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 master list, as shown in the following figure:

```
CCQPDAD                  Associate DB2 Entry for Product        Row 1 to 3 of 3
Command ===>                                                   Scroll ===> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands:  CREATE - Create new DB2 entries

DB2 Entries
  Line commands: A - Associate  C - Copy  D - Delete
  Cmd SSID GrpAttch
      *    *
      DBAA  --
      DBAB  --
      DBAC  --
----------------------------- End of DB2 entries -----------------------------
```

Figure 7. Associate DB2 Entry for Product panel

c) Repeat steps b and c for each Db2 entry that you want to create.

d) When you have created all the Db2 entries, associate them with Db2 Log Analysis Tool, or press End to display the **Customizer Workplace** panel.

3. Associate Db2 entries.

a) Specify A against one or more Db2 entries in the master list, and press Enter to associate them with Db2 Log Analysis Tool.

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 Log Analysis Tool, you must define Db2 Log Analysis Tool parameters, LPAR parameters, and Db2 parameters, if your customization requires Db2 entries.

About this task

You must define the Db2 Log Analysis Tool parameters first for the following reasons:

- If you ran the Db2 Log Analysis Tool 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 Log Analysis Tool, 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 Log Analysis Tool parameters

Db2 Log Analysis Tool parameters are specific to Db2 Log Analysis Tool.

About this task

If you ran the Db2 Log Analysis Tool 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 Log Analysis Tool parameters, a note labeled **Important** will display the correct sequence on this panel.

```
CCQPPRD                        Product Parameters                      13:56:08
 Command ===>                                              Scroll ===> PAGE

 Complete the following tasks to customize the products. The required tasks,
 required steps within a required or selected task, and required parameters
 are preceded by an asterisk (*). Ensure that values are specified for the
 required parameters. Press End to save and exit.

 Commands: SAVE   VERIFYOFF
 Line Commands: / - Select

 Product to Customize
   Product metadata library : ALA.PRD00.SALADENU      >  LPAR. . . : SC01
   Product name . . . . . . : DB2 Log Analysis Tool    >  Version . : .0

 Product customization library : ALA.TCZ.$SC01$.ALA0
                                                              More:     +

  Required parameters
   Control file name  . . . . . . . . . . . ALA.CONTROL                 >
   QUALIFIER for unqualified tables . . . . . SYSTOOLS
   ALA database name  . . . . . . . . . . . ALADB01
   ALA tablespace name  . . . . . . . . . . AALAS01
   CLIST for ALA  . . . . . . . . . . . . . ALA.SALACLST                >
   ALA loadlib  . . . . . . . . . . . . . . ALA.SALALOAD                >
```

*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 Log Analysis Tool.

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 Log Analysis Tool.

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:

```
CCQPLPR                     LPAR Parameters                        15:13:29
Command ===>                                          Scroll ===> PAGE

Enter values for all of the LPAR parameters. Press End to save and exit.

Commands: SAVE - Save parameter values

Product to Customize
  Product metadata library : ALA.PRD00.SALADENU         >  LPAR. . . : MVS1
  Product name . . . . . . : DB2 Log Analysis Tool      >  Version . : .0




 ISPF Libraries
  Message library  . . . . . . . . . . . . ISP.SISPMENU        >     Add...
  ISPF table input library . . . . . . . . ISP.SISPTENU        >     Add...
```

*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 Log Analysis Tool parameters

Db2 Log Analysis Tool parameters are specific to Db2 Log Analysis Tool.

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 Log Analysis Tool 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 51.

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:

```
CCQPDB2                      DB2 Parameters                    13:36:01
 Enter values for all of the DB2 parameters. Press End to save and exit.

 Commands: SAVE - Save parameter values

 Product to Customize
   Product metadata library : ALA.PRD00.SALADENU      >  LPAR. . . : RS22
   Product name . . . . . . : DB2 Log Analysis Tool    >  Version . : .0

                                                         More:     +

 DB2 subsystem ID  . . . . . . . . . . . : D91A
 Group attach name . . . . . . . . . . . :

 General DB2 Information
   Mode . . . . . . . . . . . . . . . . . NFM    (CM,CM8,CM9,NFM)
   Level Number . . . . . . . . . . . . . 101 (910,101)

 DB2 Libraries
   Load Library . . . . . . . . . . . . . DSN.V910.SDSNLOAD   >    More...
   Run Library  . . . . . . . . . . . . . DSN.RUNLIB.LOAD     >    Add...
   Exit Library . . . . . . . . . . . . . A91A.SDSNEXIT       >    Add...
   Bootstrap data set . . . . . . . . . . A91A.BSDS01         >    More...
```

*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 Log Analysis Tool parameters
Db2 Log Analysis Tool parameters are specific to Db2 Log Analysis Tool.

Defining LPAR parameters
LPAR parameters are parameters on the local LPAR that are required to customize Db2 Log Analysis Tool.

Generating customization jobs

To generate customization jobs for Db2 Log Analysis Tool and any associated Db2 entries, issue the GENERATEALL command, or select one or more Db2 entries on which to customize Db2 Log Analysis Tool.

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 Log Analysis Tool 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 Log Analysis Tool.

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.

```
CCQPCST                   Finish Product Customization        Row 1 to 23 of 41

Submit the members in the order in which they apply to each DB2 entry. To
submit the job, edit the member and issue the TSO SUBMIT command, or edit
the customized library and submit the jobs from there.

Product to Customize
  Product metadata library : ALA.PRD00.SALADENU   >      LPAR  . . : RS22
  Product name . . . . . . : DB2 Log Analysis Tool  > Version . : .0

Line Commands: E - Edit  B - Browse

   Product customization library . : ALA.TCZ.$RS22$.ALA0                >
```

*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:

   `<job_sequence_number><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 Log Analysis Tool assigns the template name.

DB2_entry_ID
>   Two alphanumeric characters, AA - 99, that Tools Customizer assigns to a Db2 entry.

For example, the XYZBND*DB2_entry_ID_1* and XYZBND*DB2_entry_ID_2* jobs are generated from the XYZBNDGR template, and the XYZ4*DB2_entry_ID_1* and XYZ4*DB2_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 Log Analysis Tool 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 Log Analysis Tool is customized, and the **Customizer Workplace** panel is displayed. The status is Customized for the Db2 entries on which Db2 Log Analysis Tool 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 Log Analysis Tool.

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 I*nn* 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 master 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 master list.

About this task

When you delete Db2 entries from the master 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 Log Analysis Tool 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 Log Analysis Tool by completing the steps for recustomization.

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 recustomization.

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 Log Analysis Tool 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.

Chapter 5. Required post-customization tasks

The following tasks are required and should be completed after you have finished running with Tools Customizer.

Start Log Analysis Tool

Procedure

- Start Log Analysis Tool by running the SALACLST library member ALAFRONT.

  The main menu for Log Analysis Tool opens. (For more information on the Log Analysis Tool main menu, see "Starting to use Db2 Log Analysis Tool" on page 79.

  Important: Log Analysis Tool v3.5 opens to the quick start panels the first time you run it. Thereafter, Log Analysis Tool remembers whether you were in classic or quick start panel mode when you exited and starts in that same mode. Type 9 on the command line on either the Quick Start Menu or the Classic Menu to toggle back and forth between classic and quick start panel modes.

Set user settings

Before you begin

You must be on the Log Analysis Tool - Classic Menu before typing **0** to access User Settings as described in this section.

Important: Log Analysis Tool v3.5 opens to the quick start panels the first time you run it. Thereafter, Log Analysis Tool remembers whether you were in classic or quick start panel mode when you exited and starts in that same mode. Type 9 on the command line on either the Quick Start Menu or the Classic Menu to toggle back and forth between classic and quick start panel modes.

Procedure

- From the Log Analysis Tool - Classic Menu, select option **0** for "User Settings".

  The User Settings panel appears as shown:

```
V3.5.0 ------------------------ User Settings ------------------- SC01/SS1A
Command ===>
                                                          More:     +

*DB2 name (SSID)............................ SS1A    (Subsystem Identifier)

File Defaults:
*Override product dynamic allocation......... Y      (Y/N)
 Optional datasets prefix (VSAM).............
 Optional volume serial numbers (VSAM).......
 Optional datasets prefix (non-VSAM).........
 Disk unit name for dynamic allocations...... SYSDA
 Tape unit name for dynamic allocations......
*Create continuous mode dataset.............. N      (Y/N)
*Create/Re-create ROWDATA VSAM file.......... N      (C - Create, N - No,
                                                      R - Recreate)
*HSM recalls................................. N      (Y - Yes, N - No,
                                                      S - Specify)
*Write reports to datasets................... N      (Y/N)

 SMS Management class........................
 SMS Storage class...........................
 SMS Data class..............................
```

Figure 12. User Settings panel (screen 1)

```
V3.5.0 ----------------------- User Settings ------------------ SC01/SS1A
Command ===>
                                                     More:   - +

Miscellaneous Product Defaults:
*Continue processing after warnings.......... Y       (Y - Yes, N - No,
                                                       E - Yes and some errors)
*Parallelism maximum.......................... 01      (1-16)
*Send informational messages to file......... N       (Y/N)
*Skip formatting (when possible)............. N       (Y/N)
*Use DB2SORT (when possible)................. N       (Y/N)
*Level to send to zIIP (when possible)....... 00      (0-16)
*Wait time in minutes for contention........ 00      (0-99)
 One phase mode:
 *Use 64 bit storage......................... N       (Y/N/S)
 *Use work files............................. N       (Y/N)
 Two phase mode:
 *Use QSAM for ROWDATA....................... Y       (Y/N)
*Restart assist on continuous mode........... N       (Y/N)
*Use member name on reports.................. N       (Y/N)
*Ask to save generated jobs as profiles...... N       (Y/N)
 Support Reports output flags............... BCGIMST (B - BSDS, C - Control
                                                       File, G - Storage,
                                                       I - IDMAP, S - SYSCOPY,
                                                       M - Maintenance,
                                                       T - Table Versioning,
                                                       X - Exclude all reports)
*Use fast backup during continuous mode...... Y       (Y - Yes, N -No,
                                                       F - Flashcopy only,
                                                       I - IEBGENER only)
*Create JSON parameters file................. N       (Y - Yes, N - No,
                                                       S - Short list)
```

Figure 13. User Settings panel (screen 2)

```
V3.5.0 ----------------------- User Settings ------------------ SC01/SS1A
Command ===>
                                                     More:   - +
Temporary Image Copy Allocation:
*Device type.................................. I       (D - Disk, T - Tape,
                                                       I - Input)
 Unit name....................................
 Disk space units............................ C       (T - Tracks,
                                                       C - Cylinders)
 Disk primary space.......................... 0100    (1-99999)
 Disk secondary space........................ 0010    (0-99999)
 Minimum volume count........................ 001     (1-255)
 DB2 Log Analysis Tool Jobcard:
*==> //jobcard 1
===>
===>
===>
```

Figure 14. User Settings panel (screen 3)

- Specify the user settings as desired.

   Db2 name
   Enter a four-character subsystem ID (SSID) to identify your default subsystem. This default can be
   changed at any time.

   File defaults

   Override product dynamic allocation
   Type Y to have complete control over the data sets used by the product. This means files
   used by the product will not be dynamically allocated, but rather hardcoded into your JCL.
   This allows you to modify names, attributes, sizing, and so on. The default is N, which means
   the product will dynamically allocate files as necessary. You can still override any dynamically
   allocated files as you see fit with this setting (N), but the file names will not be hardcoded
   into your JCL. If Y is selected, be sure to also create the ROWDATA VSAM file as described
   following.

Optional datasets prefix (VSAM)
This option only applies if you do not override dynamic allocation. The VSAM data set used internally by the product is prefixed with the current TSO ID of the user as the default. Optionally, you may specify a data set prefix. It must be no longer than 21 characters in length and a valid data set prefix. For example, XYZ.JOHNDOE is valid, 12345 is not. Leave blank for the default behavior which means the VSAM data set is prefixed by your userid

The following keys are available:

– &&SYSUID: User ID (up to 8 symbols)
– &&DATE/&&YYMMDD: Current date (6 symbols, numeric)
– &&TIME/&&HHMMSS: Current time (6 symbols, numeric)

Note: Numeric keys are replaced with numbers, and must be treated as such while composing the dataset prefix.

Optional volume serial numbers (VSAM)
This option only applies if you do not override dynamic allocation. Specify the DASD volume(s) onto which the VSAM data set will be created. If more than one volume is desired, enter up to 3 comma-separated volume serial numbers for your site. Each volume serial number must be exactly 6 characters long, so you must pad it with spaces if it is shorter. If not required at your site, the field may be left blank. If specified, the format must be:

```
volser,volser,volser
```

where each volser is represents a DASD volume for the VSAM file.

For example,

VSM102 is valid

VSM102,VSM106 is valid

VSM102, is not valid (comma with no following value)

VSM102VSM103VSM104 is not valid (missing commas)

VSM102,VSM30 ,VSM104 is valid (and notice the padded space after VSM30)

Optional datasets prefix (non-VSAM)
This option applies regardless of overriding dynamic allocation. All data sets allocated for this product will be prefixed with the current TSO ID of the user as the default. Optionally, you may specify a data set prefix. It must be no longer than 27 characters in length and a valid data set prefix. For example, XYZ.JOHNDOE is valid, 12345 is not. Leave blank for the default behavior which means the data sets are prefixed by your userid

The following keys are available:

– &&SYSUID: User ID (up to 8 symbols)
– &&DATE/&&YYMMDD: Current date (6 symbols, numeric)
– &&TIME/&&HHMMSS: Current time (6 symbols, numeric)

Note: Numeric keys are replaced with numbers, and must be treated as such while composing the dataset prefix.

Disk unit name for dynamic allocations
This option only applies if you do not override dynamic allocation. Enter a unit name here to be used internally for data sets that are dynamically allocated. This is unrelated to the unit specification for temporary image copy allocations. This field can be left blank. For example, set field to "SYSDA".

Tape unit name for dynamic allocations
This option only applies if you do not override dynamic allocation. Enter a unit name here to be used internally for data sets that are dynamically allocated but fail because of size limitations. In other words, all data sets are initially allocated to DASD, but if not enough space is available

for the file, then a tape allocation will be attempted. This will only occur if you provide a non-blank value here. If left blank, you are telling the product that it is not allowed to allocate any data sets on tape. Please note that this field does not prevent x37 type abends; it is only utilized if the initial allocation request fails because more space is requested than is available.

Create continuous mode dataset
This option applies regardless of overriding dynamic allocation. Type Y to create the continuous mode data set. This data set is required if you wish to run in continuous mode, which means each subsequent run of the JCL will continue from the previous point-in-time.

Create/Re-create ROWDATA VSAM file
This option only applies if you override dynamic allocation. Type C to create the ROWDATA VSAM data set (needed to run any detail report). Type R to re-create (delete the current ROWDATA file and re-create a new one with different attributes as defined by you). This should only be done if you have not previously created such a file, or want to change the file from as it exists now, and only if you are overriding the product dynamic allocation feature. Otherwise, the product will create and manage this file for you.

Note: If using one-phase mode, there is no need to create a ROWDATA VSAM file regardless of dynamic allocation settings because no such file is used in one-phase mode. The ROWDATA VSAM is only used in two-phase mode.

HSM Recalls
This option allows you to request that Log Analysis Tool issue HSM recalls against migrated data sets that are needed. These recalls do not guarantee that the data sets will be available upon usage, they merely request the recall. These data sets that may be recalled are image copies and archive logs. As there is overhead involved in this process, set this field to Y only if there are expectations that such resources may have been migrated.

If you set this parameter to N, Log Analysis Tool does not explicitly issue HSM recalls, but they are issued implicitly when an open request is made for the data set. Because the recall is being done at the time of open, Log Analysis Tool waits until control is returned from z/OS for the open request.

Set this field to S to specify the number of seconds to wait for HSM recalls **against migrated data sets**.

General rules for this option are:

- If most/all of your image copies and archive logs are on tape, leave this option to its default setting of N.
- If DASD image copies or archive logs may be migrated to ML1 (that is, DASD), set the option to N as the overhead here will far outweigh any benefits.
- If DASD image copies or archive logs may be migrated to ML2 (that is, tape), set the option to Y as savings may occur in elapsed times.
- If HSM is not implemented at your site, leave the option set to N.
- If you are unsure about any of the above conditions, the general recommendation is: leave the option set to N and only change it to Y if noticeable waits occur during product runs due to HSM migrations while processing image copies or archive logs.

Important:

- Log Analysis Tool checks for HSM recalls until the recall is performed. If, for whatever reason, the recall cannot be resolved, the wait for recall stops after approximately 5 minutes. After that time, if a recall is not performed by HSM, an informational message is displayed identifying the data set (for example, `ALAA156I: HSMM-DA1A.ARCHLOG1.D12145.T1659248.A0013916` indicates that the HSM maximum wait time has expired), and normal processing continues. The ability of Log Analysis Tool to process the relevant data set after that point depends on native HSM processes. If HSM does not ultimately recall the data set, it is likely that the process will be terminated by z/OS, but there will be no hang or loop within Log Analysis Tool.

- Since overhead is involved in determining if an HSM recall is needed, you should not use this feature unless you expect the possibility of data set migration. For example, if archive logs and image copies are on tape, you should not use this feature.
- If you do not use HSM at all, do not use this feature.

Write reports to datasets

This option allows you to write all generated reports to data sets. Without this option, reports are written to SYSOUT. The data set name for each report is a concatenation of the "Optional data sets prefix (non-VSAM)" value, the report identifier, and the job identifier. If no data set prefix is specified, the user authid is used instead. The report identifier is equivalent to the DD name of the report. For example, the detail report identifier is DTLRPT and the general report identifier is GENRPT. For example, if you specify a prefix of "ABCDEF2.ALA" and a job identifier of "JOB1" while running a report that generates a detail report, the detail report will be written to "ABCDEF2.ALA.DTLRPT.JOB1".

SMS Management class

This option is available if you wish to specify an SMS Management class for files allocated by this product. Only data sets that are allocated during job runtime will use the SMS specifications. Leave blank if no such class is desired.

SMS Storage class

This option is available if you wish to specify an SMS Storage class for files allocated by this product. Only data sets that are allocated during job runtime will use the SMS specifications. Leave blank if no such class is desired.

SMS Data class

This option is available if you wish to specify an SMS Data class for files allocated by this product. Only data sets that are allocated during job runtime will use the SMS specifications. Leave blank if no such class is desired. If data class is specified, this product assumes the data class will provide space allocation information for that data class, and therefore no overriding space allocation is provided by the product. If no space allocation is provided within the specified data class, the allocation will fail. Therefore, only provide a data class name that includes space allocation information.

Important:

- For all SMS class specifications, the requested class may not actually be used if your site has set up automatic class assignment based on other rules (for example, class assignment by data set name).
- Although archived data sets are produced during job runtime, they do not use the SMS specifications.
- Data sets created during JCL generation do not use SMS specifications. For example, the data sets used to hold the generated JCL do not use SMS specifications.

Miscellaneous product defaults

Continue processing after warnings

Type Y to allow processing to continue after any warnings are detected in the general and detail report runs. Type N to halt the process completely for any warnings. Type E to allow processing to continue after any warnings are detected and when specific errors are encountered.

Setting this field to N can have additional implications while processing the detail report in automatic mode. Automatic mode means the product determines whether to run in log-forward or log-backward (as opposed to you specifying the method explicitly). If log-backward is determined as the best method to use by the product in automatic mode, it is still possible that warnings can occur using log-backward processing (specifically, message ALAA061W). In that case, and if this field is set to N, the product automatically switches to log-forward mode.

If you do not want that type of automatic switching, either set this flag to Y or explicitly set your detail report log processing mode to log-forward or log-backward. In any other case, if this flag is set to N and a warning occurs, processing halts and the return code is set to 4.

Setting this field to E allows Log Analysis Tool to continue running when an object in a filter is not found (ALAA036E error), or when an image copy is not available (ALAA107E error), or table versioning data is not available (ALAA486E error), or when a S0C7 is encountered during row expansion. Typically, these conditions lead to immediate termination of the job. However, with this option, the table that encountered the error is removed from the job and Log Analysis Tool continues processing other tables. Tables that are removed due to these errors are reported as a warning, and the return code for the step is set to 4.

Parallelism maximum
Type a value from 1 (default) to 16. This value represents the maximum number of concurrent subtasks that can run to process various inputs, such as Db2 logs or Db2 image copies.

This number should most likely represent the number of tape drives available if Db2 archive logs or image copies are on tape and will be accessed. For example, if you expect 10 Db2 logs to be accessed, and your site has a total of 4 tape drives available, a value of 2 might be appropriate to keep two drives available for other tasks. If all your Db2 logs are on DASD, a number of 5 might be appropriate. Keep in mind that your batch address space might exhaust all available storage if a high number is used. Also, you may not want to tie up all available tape drives. So fine tuning may be required.

If you specify 16 as a value, each group of 16 will be processed sequentially, up to the amount needed. That is, no more than 16 subtasks can be reading input at any given time, and again, that number is likely high in terms of batch address space resource availability.

Send informational messages to file
N is the default.

Type Y to send most informational messages to SYSOUT rather than to the job log output. This information must not be sent to a permanent file, only to SYSOUT (as the JCL will be generated by the product for you). This includes most WTO messages that would otherwise go to your job log and console, but does not include error messages. Some messages may still go to the job log even if this field is set to Y because errors may occur during error handling preventing use of this option. However, this will be the exception and generally all informational messages will follow the rules of this field setting. If set to N (No), the default behavior is taken and these messages continue to be issued as WTOs, going to your job log and console. If set to Y, and problems occur during execution, always send the contents of the file (identified by the DD statement 'INFOM' in your job log file output) containing these informational messages along with any other supporting documentation. These informational messages are useful in resolving most problems during job execution. This flag is provided for those sites that wish to suppress these messages from both their console and job log messages output.

Skip formatting
Type Y to skip formatting raw row data into displayable format. This process can be time consuming, and may not be needed. When this flag is set to Y, formatting of raw row data will be bypassed only if the following are all true:

1. no SQL is requested
2. detail reports are bypassed ("Output flags = B" for detail report)
3. no archiving is requested
4. no loading of data using audit mode

If any of the preceding are not true, formatting must occur, regardless of this flag setting. Additional notes to keep in mind here: if you plan on using the archive option to re-run against current data only, do not set this flag to Y, as the needed input to that process will not be available. Also, if you plan on using the "Audit Mode" option of the "Load data into DB2" feature, do not set this flag to Y, as the needed input to that process will also not be available. These two notes are the reason this is not automatically handled by the product; this flag allows you to control the behavior based on your expectations of future use.

Use DB2SORT

Type Y if you want the product to use DB2SORT in place of the standard sorting routines. If set to Y, the product will ascertain whether DB2SORT is available for use; if so, it will use it; if not, it will default to its standard sort calls (that is, it will not terminate if DB2SORT is requested but cannot be used, for any reason). There is some overhead involved in making those decisions internally, so if it is known that DB2SORT is not available at your site, set this field value to N. Even if requested, due to restrictions within DB2SORT's API, it will not be used exclusively within the product for every needed sort. Use of DB2SORT may result in better performance.

Level to send to zIIP

Enter a value of 00 through 16.

00

Do not send any process to the zIIP.

01 - 16

Log Analysis Tool should attempt to offload some processes to the zIIP. This is a tuning value. A value of 01 sends the minimal overhead loads to the zIIP. Higher values will have additional overhead. For instance, a value of 10 will cause a process that utilizes CPU to use the zIIP resulting in reduced CPU usage, but the overhead of sending this to the zIIP could cause the overall run times to increase by a small amount. This parameter can be set to tune the product as to what is important to you.

Restriction: You must have a zIIP processor to offload processes and save CPU time. If there is not a zIIP processor that can be used, set this value to '00' so that the setup and scheduling of the offload to the zIIP processor is bypassed.

Wait time in minutes for contention

Enter a value of 00 through 99. A value of 00 means do not wait for a data set to become available - issue an error message immediately. A value of 01 through 99 signifies that the product should wait the specified number of minutes to allow the other process or job that has an exclusive hold on the data set to terminate. If at the end of the number of minutes the data set is still being held, an error message is issued. If the other process or job releases the data set within the timeframe, the Log Analysis Tool will continue as normal. Log Analysis Tool checks periodically throughout the waiting period for the data set to become available.

One phase mode

The methods for one-phase mode are using 64 bit storage, or using work files. See the following descriptions for each. They are mutually exclusive. See the tutorial topic on "one-phase mode" for a complete description of the implications of using these methods of processing.

Use 64 bit storage

Y

Places certain data into 64 bit storage (that is, above the bar) rather than to files. This can result in much better performance as I/O will be averted for this data. However, keep in mind that your site has limits on the amount of available 64 bit storage, and this could affect the results of your run (that is, the job may exhaust all available storage and fail). Use this method for small to medium sized runs.

The files this affects include all of the following: EXTFILE, PLOGD, FRMTFILE, and ROWDATA.

If you use this option, both the general and detail reports behave differently. The generated JCL will be one step, rather than the many steps used when not using this option. The implications of this are more simplicity, but also, any job failure will require re-running from the start. You should experiment with this option to see how it compares to normal, file-based runs for your average expected job load.

S

The 'S' (spill) option tells the product that you want to use 64 bit storage, but you want it to switch into work file mode if the value of MEMLIMIT is exceeded. MEMLIMIT is the

64 bit memory limit that z/OS keeps track of as your job runs. If you are concerned that a job may take up too much 64 bit memory, you can control the value of MEMLIMIT through that JCL parameter which works like the REGION parameter, or you can just use your system default. In any case, Log Analysis Tool will examine the current allocated 64 bit memory and compare it to the MEMLIMIT value, and if it is about to be exceeded, the job will automatically switch to work file mode and continue.

This avoids potential job failures due to memory exhaustion in 64 bit mode. But it also requires work file mode allocations (which are automatically included if this field is set to 'S'). If the job switches to work file mode, the next run of the job will revert to 64 bit mode, assuming the mode file is not re-created via the panels. This allows you to place such a job in a scheduler and allow it to start to run in 64 bit mode each time, until and unless it needs to switch to work file mode again. This switching operation only occurs during 64 bit storage obtains for the major files, that is, for the EXTFILE, PLOGD, ROWDATA, or FRMTFILE as they exist in 64 bit mode. The checking does not occur for smaller internal allocations. If MEMLIMIT is exceeded on those allocation attempts, a normal IARV64 message will be issued and job processing must stop.

Use the 'S' for jobs that most likely will not exceed your MEMLIMIT value, but may occasionally. There is some overhead incurred during the switch to work file mode, so if a job consistently switches into work file mode from 64 bit mode, it may be beneficial to just use work file mode instead of the 'S' setting.

When you specify 'Y' or 'S', you are considered to be running in "one-phase" mode, as compared to the normal two-phase mode this product allows. Two-phase mode means you can run a general report, and later on, run a detail report against the output of the general report. One-phase mode means there is no possible separation of processes. In one-phase mode, you must run everything needed at the same time, as the files necessary for two-phase mode are not available.

Important:

1. When using Log Analysis Tool in 64-bit mode you are limited to reading 500 logs. If that limit is exceeded, an error occurs.

2. You cannot use 64-bit storage in conjunction with LOB/XML reporting. The inclusion of LOB/XML data types are mutually exclusive with using 64-bit storage for these files, and an error message will be generated if you attempt to include LOB/XML columns while running using 64-bit storage.

Use work files

Type Y to operate in one-phase mode, but use temporary work files rather than 64 bit storage for data. The files affected by this are identical to those described under "Using 64 bit storage", as is the description contained therein. The only difference is that 64 bit memory is not used, but rather temporary work files. LOB/XML columns are supported using work files. Use this method for very large report runs. This is the default configuration.

Two phase mode

Use QSAM for ROWDATA

Type Y to indicate that you want the ROWDATA file to use the QSAM access method. The default is N for conformity with prior releases. Log Analysis Tool will use a QSAM file everywhere ROWDATA is referenced in its JCL rather than a VSAM file.

Important: The use of this option is mutually exclusive with any one phase mode settings, so those must both be set to N.

Restart assist on continuous mode

Type Y if you want the product to assist in the restart capabilities while running in continuous mode. Manual intervention for a restart in continuous mode is required if this field is set to N.

Note: Restart assist in continuous mode does not depend on the existence of, or lack of, a "RESTART=STEPn" parameter on your job card. Once restart assist is set to Y, it tracks

these jobs automatically. But it is still mandatory that you restart on the proper job step. In one-phase mode, a RESTART parameter would only be needed if you have additional steps in the job, for example, if LOAD options are present, because a basic one-phase mode job typically only has one (primary) step. But if running in two-phase mode where there are several job steps, you must restart from the failing step. For example, if the job fails in some step after the step where program ALAGEN1 was executed, and you restart from the step where program ALAGEN1 executes, the files will not be properly copied because the product has determined that step already completed successfully. So it is imperative that you restart from the failing step. If you still wish to restart from the very beginning of the job for whatever reason in two-phase mode, you will have to manually do the copies yourself, as was always the case in prior releases.

See "Continuous mode processing" on page 113 for a full explanation of this feature.

Use member name on reports
Type Y if you want to use the Datasharing Member name for Datasharing Subsystems or the SSID when reporting on non-Datasharing subsystems instead of using just the ID.

Ask to save generated jobs as profiles
Type Y if you want to be given the option to create a profile, saving all your specified options, after a job has been generated and edited and/or submitted.

Support Reports output flags
Enter the report code to have the appropriate support report produced in the INFOM DD SYSOUT. For any report that should not be produced, make sure that its report code is not entered. If you blank out this field, then all the reports will be selected (which is the default). Valid values are.

- B: BSDS report is to be generated.
- C: Control File report is to be generated.
- G: Storage report is to be generated.
- I: IDMAP report is to be generated.
- S: SYSCOPY report is to be generated.
- M: Maintenance report is to be generated.
- T: Table versioning report is to be generated.
- X: Do not produce any support reports.

Use fast backup during continuous mode
When running in continuous mode, the EXTFILE is backed up in case a restart is needed. Options for backing up the file using a fast backup method are provided with this parameter. Valid values are:

- N: Do not use a fast backup method.
- Y: Use FlashCopy with failover to IEBGENER as needed.
- F: Only use FlashCopy.
- I: Only use IEBGENER.

Create JSON parameters file
This option enables support for JCL simplification and the generation of a JSON parameters file. Valid values are:

- N - No, do not generate this file. A standard JCL job will be generated with MODEPARM and MODE35 to store parameters.
- Y - Yes, generate a JSON parameters file as JCL input to store all of the available parameters.
- S - Short. Create a shortened version of the JSON file as JCL input. This version stores only the parameters that are used and omits fields with default values.

For more information on the JSON parameters file, refer to .

Temporary Image Copy allocation
The data requested for temporary image copies is required for log-forward applications. Supply the requested information for allocations of temporary data sets needed to process inline image copies, if any found during processing. (This temporary data set is ICDD.)

Device type
Valid values are:

– Type D if you want temporary image copy allocations to be forced to disk.

– Type T if you want temporary image copy allocations to be forced to tape.

– Type I if you want temporary image copy allocations to be derived from the device type of the originating image copy.

Unit name
If **Device Type** was specified as either **D** or **T**, supply a unit name for the requested type. For example, if T, the unit name might be 3390.

Disk space units
If **Device Type** was not specified as **T**, disk space units may be needed. Specify either **C** for cylinders, or **T** for tracks.

Disk primary space
If **Device Type** was not specified as **T**, disk primary space may be needed. Specify a five-digit number for primary space allocation.

Disk secondary space
If **Device Type** was not specified as **T**, disk secondary space may be needed. Specify a five-digit number for secondary space allocation.

Minimum volume count
Specify a 3-digit number between 1 and 255 to be used as the volcount during allocation. Most of the time this can be left as 1. If the product is processing a large image copy and needs to allocate the temporary data set over more volumes than the original image copy used, then increase this field to the number of volumes that should be used for the data set. The larger number is used in the allocation request, whether it is your specification or the volume count of the original image copy.

Db2 Log Analysis Tool Jobcard
Enter a valid job card that will be used in all generated JCL. Be sure to enter this information prior to creating any JCL within this product. At least one line of the jobcard input must be provided (denoted by the '*==>' prefix).

Important: Log Analysis Tool makes some use of 64 bit storage, regardless of whether 64 bit in one-phase mode is requested. This is strictly internal, and can help alleviate storage constraints from previous releases. Because of this, the product requires a MEMLIMIT value greater than 0. MEMLIMIT is the z/OS parameter that controls how much 'above the bar' storage (64 bit storage) the address space can use. Log Analysis Tool as shipped sets REGION=0M on the job card, which sets MEMLIMIT=NOLIMIT, but is still ultimately controlled by your IEFUSI exit MEMLIMIT setting, if any. If you change REGION to anything but 0M, you must ensure MEMLIMIT does not end up defaulting to 0, which will result in an error situation (MEMLIMIT=0 means no 'above the bar' storage is available to the product). You can explicitly set MEMLIMIT on the job card statement, for example:

```
//TEST  JOB  'D83,123456',MEMLIMIT=10000M,REGION=0M
```

MEMLIMIT specification is therefore completely in your control, or you can just use the default REGION=0M. See the *z/OS MVS JCL Reference* and the *z/OS MVS Programming Extended Addressability Guide* for complete MEMLIMIT information.

Create the continuous mode data set

About this task

If you want to run Log Analysis Tool in a continuous mode, meaning that each subsequent run of JCL continues from the end of the preceding run of JCL, you must create the continuous mode data set according to these steps.

1. Type Y in the **Create continuous mode dataset** field. The Create continuous mode file panel appears.

```
Create continuous mode file -----------------------------------------
COMMAND ===>

Action.................... ===> E       (E=Edit,S=Submit)

Continuous mode file name.. ===>

Disk space units........... ===>        (T=tracks,C=cylinders)
Primary allocation......... ===>        (0001-9999)
Secondary allocation....... ===>        (0001-9999)
```

Figure 15. Create continuous mode file panel

Action
Type E to edit the generated JCL or S to submit the JCL directly.

Continuous mode file name
Specify up to 44 characters to denote a valid file name.

Disk space units
The allocation unit for the continuous mode file. Type T to allocate space in tracks, or C to allocate space in cylinders.

Primary allocation
Specify up to five characters denoting the number of tracks or cylinders to use as a primary allocation for the continuous mode file.

Secondary allocation
Specify up to five characters denoting the number of tracks or cylinders to use as a secondary allocation for the continuous mode file.

2. Enter the continuous mode file information. Press Enter.

3. Log Analysis Tool creates the continuous mode file.

Note:

1. If the continuous mode data set you initially created must be recreated later (for example, to resize it), you can delete the current continuous mode file and recreate it from this panel.

2. If the continuous mode data set you created is not managed by SMS then you must open and close the file before using it in Log Analysis Tool.

Create the modeparm file

The modeparm file allows you to change certain parameters in the JCL without having to step through the ISPF panels. There are certain parameters that still must be updated using the panels, but those mainly affect the generation of the JCL.

The modifiable parameters are passed into the Log Analysis Tool job via the DD MODEPARM. The parameters are read and updated by the associated load module at the time the MODEFILE Report is being generated.

For existing jobs, you can populate a file with the modifiable parameters from the 'Utilities' panel of the Log Analysis Tool ISPF main menu. In the Utilities panel select 'Mode File Operations', there is an option 'Create modeparm file'. Type Y in the field and the MODEPARM file will be created using the same naming

convention as the parameters within the named MODEFILE DSN and populated with the values found in the MODEFILE. You can then add this file in each step of your existing JCL under the DD MODEPARM.

For all new jobs generated by Log Analysis Tool, the parameters are written automatically to a file at JCL creation time. The file will be referenced by the appropriate DD as well.

Important:

- Not all MODEFILE parameters are being exposed via the DD MODEPARM. Only the following are being exposed:
  - CHAR DATA IN HEX
  - COMMIT SCOPE
  - CONTINUE PROCESSING AFTER WARNINGS
  - CONTINUOUS MODE
  - CURRENT ROW BY KEY
  - DATA SHARING MODE
  - DETAIL REPORT BYPASS SYSLGRNX
  - DETAIL REPORT OUTPUT FLAGS
  - DISK UNIT NAME FOR DYNAMIC ALLOCATIONS
  - GENERAL REPORT OUTPUT FLAGS
  - GENERATE REDO SQL
  - GENERATE UNDO SQL
  - HSM RECALLS
  - LEVEL TO SEND TO ZIIP (WHEN POSSIBLE)
  - LOG READ PARALLELISM MAXIMUM
  - LOG READING MODE
  - MISC FLAGS
  - REPORT ROW LENGTH
  - RESOLVE STARTED UOWS
  - SEND INFORMATIONAL MESSAGES TO FILE
  - SET CURRENT SQLID
  - SHOW VARCHAR LENGTH
  - TAPE UNIT NAME FOR DYNAMIC ALLOCATIONS
  - TREAT MASS DELETES
  - USE DB2SORT (WHEN POSSIBLE)
  - WHERE CLAUSE BY KEY
- The parameter values have all of the same restrictions as their respective fields in the ISPF panels.

Chapter 6. Getting started with Log Analysis Tool

This topic contains information on launching and setting up Log Analysis Tool. It also describes basic components of Log Analysis Tool and provides a basic orientation for the product.

Overview

IBM Db2 Log Analysis Tool for z/OS enables you to easily monitor changes that are made to data without significant system overhead. You can use automatically created reports to monitor changes using various resource criteria such as date, user, or table. Log Analysis Tool maintains the efficiency of your Db2 databases while monitoring data integrity.

Important: You should make no modifications to affected objects concurrent with a Log Analysis Tool run, including, but not limited to, REORGs, image copies, table alterations, table drops, and so on. Any modifications to affected objects concurrent with a run will lead to unpredictable and unsupported results.

Starting to use Db2 Log Analysis Tool

You can start to use Log Analysis Tool from the Classic Menu. The Classic Menu is the primary menu from which you navigate Log Analysis Tool. You can access all Log Analysis Tool functions from this menu.

Important: Log Analysis Tool v3.5 opens to the quick start panels the first time you run it. Thereafter, Log Analysis Tool remembers whether you were in classic or quick start panel mode when you exited and starts in that same mode. Type 9 on the command line on either the Quick Start Menu or the Classic Menu to toggle back and forth between classic and quick start panel modes.

```
-------------------- DB2 LOG ANALYSIS TOOL - Classic Menu --------------------
Option ===>

                                                  User ID: TWUSR
                                                  Date...: 2020/04/06
                                                  Time...: 16:36
                                                  Release: V3.5.0
                                                  System.: LPR1
------------------------------------------------------------------------------
  0. User settings
  1. Generate database activity report (general)
  2. Generate database activity report (details)
  3. Load data into DB2
  5. Execute dynamic SQL in batch
  6. Process archived output
  7. Utilities
  8. Profiles
  9. Quick start menu
 10. Message help
 11. Setup
 12. Tutorial
 13. Exit
------------------------------------------------------------------------------
```

Figure 16. Db2 Log Analysis Tool main menu

Classic Menu options

User settings

Select this option to enter user settings necessary for product utilization. These settings must be defined per user of the product.

Tip: Values of '0' or 'D' select this option. The 'D' value exists for compatibility with earlier releases, when this option was named "Defaults".

Generate database activity report (general)
Select this option to create a generalized report of Db2 log activity. Though its primary purpose is for a generalized report, a combination general and detail report may be selected from within this option.

Generate database activity report (details)
Select this option to create a detailed report of Db2 log activity. This option cannot be executed until after the general report is created, as it uses output from the general report run as input.

Load data into Db2
Select this option to load header or table data generated by this product into a Db2 table. Either a general or detail report must be run prior to invoking this option.

Execute dynamic SQL in batch
Select this option to dynamically execute SQL created during the detail report run. This option cannot be executed until a detail report has been run.

Process archived output
Select this option to run a detail level report using (as input) either the last detail report run output, or archived output. This option allows you to process previously generated output without the need for re-running from the beginning (that is, it bypasses log reads and Db2 resources such as image copies or table space data which were already accessed in a previous run).

Utilities
Select this option to view the utilities menu. The utilities are used to perform various functions within the product.

Profiles
Select this option to view the profiles panel. The profile panel is used to save and load general report, detail report, and LOAD settings.

Quick start menu
Select this option to access the quick start panels. The quick start panels are designed to generate a Log Analysis job more quickly and easily.

Message help
Select this option to get more detailed explanation of any ALAA- prefixed messages generated while navigating through the online panels or during generation of any reports.

Tip: Values of '10' or 'M' select this option. The 'M' value exists for compatibility with earlier releases.

Setup
Select this option to enter settings necessary for product utilization. These settings must be defined for each needed Db2 subsystem, and would likely be done by a product administrator or installer.

Tip: Values of '11' or 'S' select this option. The 'S' value exists for compatibility with earlier releases.

Tutorial
Select this option to enter the tutorial for this product. The tutorial contains much information about the product's operations and panels, though HELP (PF1) resides behind each panel as well.

Tip: Values of '12' or 'T' will both select this option The 'T' value exists for compatibility with earlier releases.

Exit
Select this option to exit the product.

Tip: Values of '13' or X will both select this option. The '13' value exists for compatibility with earlier releases.

If this message appears the first time you start Log Analysis Tool:

```
ALAA062E-No default SSID set; choose option 0 for User settings
```

you must define your default settings according to before you can use Log Analysis Tool.

Db2 Log Analysis Tool setup options

Customization of Db2 Log Analysis Tool is primarily accomplished with IBM Tools Customizer. However, you might want to change options after customization. The Setup option on the Db2 Log Analysis Tool main menu can be used to configure some features and functions.

On the Db2 Log Analysis Tool main menu, type 11 and press Enter to access the **Enter DB2 System Parameters** panel. The following provides a description of the parameters on the **Enter DB2 System Parameters** panel.

DB2 Control Dataset
This field displays the name of the Db2 control data set. You cannot change this field.

DB2 Subsystem ID
Enter the Db2 subsystem identifier (SSID) for the subsystem you want to configure.

Configuration options

On the **Enter DB2 System Parameters** panel, enter one of the following values in the **Command** field.

1
Configures options specific to a particular Db2 subsystem, including ZPARMs, BSDS, and load library information.

2
Configures the Db2 plan name, message library, custom DSNHDECP module, and BSDS options for Log Analysis Tool.

The topics that follow provide additional information about each of these options.

Entering ZPARM, BSDS, and load library information

When configuring a Db2 subsystem for use with Log Analysis Tool, you must provide ZPARM, BSDS, and load library information on the **Update DB2 Parameters** panel.

Procedure

1. On the **Enter DB2 System Parameters** panel, type 1 in the **Command** line and press Enter.
2. On the **Update Parameters for DB2 Subsystem** panel, enter the required subsystem parameters as follows:

   DB2 ZPARMs Member
   Enter the eight-character load module member name generated for this Db2 subsystem.

   DB2 Bootstrap DSN #01 and #02
   Enter the fully qualified data set names of the bootstrap data sets for this Db2 subsystem.

   Loadlib1 through Loadlib5
   Enter the names of the data sets that comprise the current load library concatenation for Db2. This concatenation is usually a subsystem-specific DSNEXIT library, the base DSNEXIT library for the current Db2 version, and the base DSNLOAD library for the current Db2 version.

   Note: If you use edit or field procedures in your tables, RUNLIB must be both authorized and defined as a non-DB2 load library. This will include the RUNLIB in generated JCL under the STEPLIB DD, which is required to internally access the edit or field procedures.

   Non-DB2 Loadlib6 through Non-DB2 Loadlib0
   Enter the names of the libraries (data sets) that you want to include in the STEPLIB. The data sets must already exist and can represent any load libraries that might be necessary during job execution, such as sort libraries or product libraries. All libraries must be APF authorized.
3. When you have entered all required parameters, press Enter, then PF3 to exit and save your changes.

Entering Db2 Log Analysis Tool specific information

You must enter the plan name and set other configuration settings for Db2 Log Analysis Tool on the **Update LAT Parameters for DB2 Subsystem** panel.

Procedure

1. On the **Enter DB2 System Parameters** panel, type 2 in the **Command** line and press Enter.
2. On the **Update LAT Parameters for DB2 Subsystem** panel, enter the parameters as follows:

   Plan Name
   Enter the plan name for Log Analysis Tool.

   Message Library
   Enter the data set name of the Log Analysis Tool message library.

   DSNHDECP module
   Enter the member name of the load module that retrieves Db2 default values. You can leave this field blank for the default value, or specify a different member name.

   Use only ARCHLOG1 and Use only ARCHLOG2
   Indicate whether you want to process only one of the defined sets of archive logs in your BSDS. If either of these fields is set to Y, the other set, whether defined or not, will not be available. You can also prioritize ARCHLOG1 or ARCHLOG2 by setting either field to P.

   Active Log priority
   Set this field to Y to indicate that the active log is to be used whenever an active log is available and it contains data that also resides on an archive log. The default behavior (field set to N) is that archive logs have priority over active logs.

Using message help

About this task

You can get help on warning, error, and informational messages in Log Analysis Tool by using the Message Help option. Select option **10**, **Message Help**, from the Db2 Log Analysis Tool menu.

Here is how to use message help:

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **10** (Message Help).

   The Message Help panel appears.

   ```
   V3.5.0 ---------------------- Message Help ---------------------- SC01/SS1A
   COMMAND ===>


   *Enter message ID... ===>          (must be prefixed ALAA*)

   ------------------------------------------------------------------------------

   ------------------------------------------------------------------------------
   ```

   Figure 17. Message help panel

2. In the **Enter message ID** field, type the seven-character ID that precedes the message for which you want information. (Do not type the letter.)

   An explanatory message appears.

   For example, a message similar to this one is returned when you type ALAA001 and press **Enter**:

```
V3.5.0 --------------------- Message Help --------------------- SC01/SS1A
COMMAND ===>


*Enter message ID... ===> ALAA001  (must be prefixed ALAA*)

--------------------------------------------------------------------------------
 'ALAA001E-FIELD MUST NOT BE LEFT BLANK'
--------------------------------------------------------------------------------
 EXPLANATION: THE SPECIFIED FIELD WAS LEFT BLANK. THE PANEL REQUIRES INPUT
 FOR THE FIELD.

 SYSTEM ACTION: PANEL PROCESSING HALTS AND AN ERROR MESSAGE IS ISSUED.

 USER RESPONSE: CORRECT THE INPUT AND RETRY.
```

Figure 18. Message help panel

Note: Log Analysis Tool Message Help does not explain SQL errors. If you require more information on an SQL error, consult your Db2 documentation.

Accessing the tutorial

About this task

You can access the tutorial in Log Analysis Tool by using the Tutorial option. Select option **12**, **Tutorial**, from the menu.

Here is how to use the tutorial:

Procedure

1. From the Db2 Log Analysis Tool menu, select option **12** (Tutorial).

   The main tutorial panel appears.

   Note: Log Analysis Tool uses scrollable panels. Near the upper right corner of the panel are indicators that identify if there is more information available by scrolling:

   ```
   More:   - +
   ```

   Pressing **Enter** scrolls the panel forward, as does the PF8 key. Use the PF7 key to scroll backward.

```
TUTORIAL ------------- DB2 LOG ANALYSIS TOOL FOR Z/OS ------------------------
Option ===>
                                                           More:     +

                            TABLE OF CONTENTS

  The following topics are presented in sequence, or may be selected by
  entering a selection code in the option field:

 Primary menu panels
  0  Quick start menu vs. classic menu
  1  User settings
  2  Generate reports
  3  Generate UNDO/REDO SQL
  4  Execute dynamic SQL in batch
  5  Generate LOADFILE
  6  Generate database activity report (general)
  7  Generate the database activity report (details)
  8  Load data into DB2
  9  Process archived output
 10  View LOB/XML data (Utilities menu)
 11  Table Versioning Support (Utilities menu)
 12  Profiles
```

Figure 19. Db2 Log Analysis Tool Tutorial panel

2. You can then enter a selection code to view a specific topic or use the navigation keys to read through from start to finish.

   Note: There are many more topics available than those shown here. This is a scrollable panel.

3. Press PF3 to return to the Db2 Log Analysis Tool - Classic Menu.

Chapter 7. Running reports

This topic provides information on using Log Analysis Tool to run general reports, details reports, and all of their associated reports.

Running General Reports

The first step in analyzing Db2 log activity using Log Analysis Tool is to run a general report. This must occur prior to running any details level reports. The output from the general report run is a combination of generalized reports and data sets. These data sets are used as input to details level report jobs. The general reports show the different types of Db2 activity that matched your filter criteria, but not the actual data changes made. Use these reports to determine if you require the actual data changes, and if so, generate the details level reports.

Important: If running in one-phase mode, the general report and detail report are always run in the same job step. You cannot decide to run a detail report after running a general report in one-phase mode; you either run only a general report, or both reports together in one-phase mode.

How to create a General Report

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **1**, **Generate database activity report (general)**.

   The Generate Database Activity Report (General) panel appears.

```
V3.5.0 ---------- Generate database activity report (general) ---- SC01/SS1A
COMMAND ===>
                                                         More:     +

*DB2 subsystem name..... SS1A     (SSID)
*Action................ E        (E - Edit, S - Submit)
 Job Identifier.........
*Generate details....... Y        (Y - Yes, N - No,
                                   X - No, do not populate EXTFILE)
*Data Sharing Mode...... Y        (Y/N)
*Specify logs........... N        (Y/N)
*LOAD options........... N        (Y/N)
 Misc flags.............          (X - Bypass SYSLGRNX,
                                   M - Bypass SYSLGRNX when Modify Recovery,
                                   P - Include partial recovery points,
                                   H - High speed mode,
                                   L - High speed mode with LOAD LOG YES)
*Output flags........... GS       (B - Bypass reports, G - General,
                                   S - Summary, X - Extended, T - Transaction,
                                   Q - Quiet time, I - Impact,
                                   J - Impact by row, F - Commit Frequency,
                                   C - Continuous mode file, M - DB2 Commands)

(continued)
```

Figure 20. Generate database activity report (general) panel (top half)

```
 V3.5.0 ---------- Generate database activity report (general) ---- SC01/SS1A
 COMMAND ===>

 Log range:                                                          More:
-
  Start/End Date-Time.... 2014/01/11 - 00:00:00 / 2014/09/01 - 00:00:00
  Start/End RBA (URID)... _____  / _____
  Start/End LRSNs........ _____  / _____
  Continuous mode file... _____
 *Resolve started UOWs... N      (Y/N)
  64 bit mem threshold... __      (00,40-99)
 *Override GMT offset ... N      (Y/N)
   with this GMT offset.. +00:00

 Filters for log data:
 *Show UPDATEs........... Y      (Y/N)
 *Show DELETEs........... Y      (Y - Yes, N - No,
                                  X - Yes, but exclude mass deletes)
 *Show INSERTs........... Y      (Y - Yes, N - No, X - Yes, but exclude loads)
 *Show rollbacks......... N      (Y - Yes, N - No, O - Only)
 *Compensation recs...... N      (Y/N)
 *Include LOB/XML data... N      (Y/N)
 *Show uncommitted....... N      (Y/N)
 *Include catalog data... N      (Y/N)
 *Misc filters........... Y      (Y/N)
 *Object filters......... A      (N - None, M - By Name,
                                  I - By IDs, A - Advanced)
 *Filter file usage...... N      (N - None, S - Save, E - Edit, U - Use)
  Filter file name.......
```

Figure 21. Generate database activity report (general) panel (bottom half)

2. Enter the required information and press Enter.

   Log Analysis Tool runs the General Report.

Specifying values in the General Report fields

The general report fields are described below. Any values that can be specified are also listed in this section.

Db2 Subsystem Name
Enter the four character subsystem identifier (SSID) for the Db2 subsystem. Db2 activity is analyzed using this subsystem. If you set up a default SSID, it is displayed here.

Important: After you complete the specifications for this job and generate the JCL, you can edit the SSID within the JCL and run this job on another subsystem. Log Analysis Tool will see the change and update the MODEFILE without causing an error condition. You still must define the SSID the control file for proper usage throughout the product.

Action
Type E to edit the generated JCL, or S to submit the JCL directly to generate the report.

Job Identifier
You can use this optional field (8 characters maximum) to identify data sets associated with a given run of the created JCL. It is used in the formation of the data set names used in the JCL. For example, if the Job Identifier field is set to TEST01, the REFILE is named USERID.REFILE.TEST01. Use this field when you need to execute multiple runs of the JCL and isolate each run from the other. If this field is left blank, each run of a General Report replaces the previous report's data.

Generate Details
Type Y to run a concatenated job of both the General and Detail reports at the same time. This creates a JCL stream that generates both reports in the same job stream. This option does not allow further filtering at the Detail report level (only General Report filters are used), however, other options on the Details report panel are used when processing in this mode (for example, **Log reading mode** and **Generate UNDO/REDO SQL**).

- Y – If set to Y, another pop-up panel is displayed requesting the required Detail report information to allow for greater ease of use when running a combined General/Detail report. See "Generate details" on page 98 for more information.

Note: You should use extreme caution when choosing this option. It is always advisable to examine the results of a General Report prior to invoking a potentially resource-intensive Detail report.

- N – If set to N, Log Analysis Tool will only create a General Report (you can create a Details Report later).
- X – If set to X, Log Analysis Tool will only run the General Report and will not populate the EXTFILE DD. By not populating the EXTFILE DD, DASD can be saved, but a subsequent Detail report cannot be run until the General Report is run again with option 'N' or 'Y'.

Data Sharing Mode

Type N if you are in a data sharing environment but you only want to process the logs for the specified subsystem. For any other case (even if not in a data sharing environment), leave as the default value (**Y**).

Important: You must not specify **N** if the data sharing group name is used; your results will be unpredictable.

Specify Logs

Type Y to hard code all the desired Db2 logs to be used for both the general and detail reports (if set to Y, the logs specified are always carried over to the detail report as well). See <u>"Specify logs" on page 112</u> for more information on specifying logs.

LOAD options

Type Y to directly include LOAD into the JCL stream. This option allows you to specify the LOAD options (the same as option 3 off the main menu) in one step, rather than running a general or detail report first, and then running the LOAD options separately. An N excludes LOAD options at this point in processing. If you specify Y, you will be directly placed into the 'Load data into DB2' panel where you will specify all your LOAD options. The JCL for LOAD options will be appended to your general/detail report JCL. If running in one-phase mode, you MUST use this option if you wish to use the LOAD option. LOAD options are not available as a separate step when using one-phase mode.

Misc flags

These values indicate certain processing modes to the product:

X

Bypass use of the SYSLGRNX table.

M

Bypass use of the SYSLGRNX table when using Modify Recovery.

P

Include partial recovery points.

H

Allow high-speed processing.

L

Use high-speed processing with LOAD LOG YES.

You can enter spaces to avoid these settings, allowing Log Analysis Tool to use its default processing. You can use a combination of these flags. For example, spaces are valid, X is valid and XP is valid. A detailed description of each processing mode follows:

Bypass use of the SYSLGRNX table

SYSLGRNX is a Db2-owned internal table that contains useful information about table space access as related to RBAs or LRSNs (timestamps). Using this table can reduce the number of logs that Log Analysis Tool must read. However, there is overhead associated with such access, so bypassing the use of SYSLGRNX may help in some situations and hurt in others.

Guidelines:

- If using dropped object mode, always bypass use of SYSLGRNX.
- If your requested time frame covers a period for which SYSLGRNX entries have been removed, as can be done with the Modify Recovery utility, then you must bypass SYSLGRNX or else the associated Db2 logs for that time period will not be read.

- If your SYSLGRNX table has been cleared of data going back as far as your start date and time, you must bypass the internal use of this table. In that case, type X.
- If filtering on SYSIBM.SYSCOPY, or it is likely that SYSIBM.SYSCOPY will be in your result set, also type X, because that table space is not recorded in SYSLGRNX.

For any other case, it is your choice.

There is a fair amount of overhead involved in using SYSLGRNX to determine a log's usefulness. If you have many object type filters, or wild carding will result in many object type filters (table spaces), and your time range is very narrow, avoid using SYSLGRNX. This is because the time it takes to process this information can take longer than any possible savings resulting from eliminating access to a few Db2 logs.

If your time frame is large, or you know many logs will be encountered for your requested time-of-day, then allow normal usage of SYSLGRNX information. If Log Analysis Tool encounters problems accessing SYSLGRNX and an error message is displayed, you can set this flag value to X to temporarily bypass its use until the problem is resolved.

For SYSLGRNX processing, the DD named SYSPRINT will contain a reference to a temporary data set, and this is hardcoded in the generated JCL. This allocation is set to CYL,(100,100). If you need more space, increase it in this JCL. However, if you do need more space, then the use of SYSLGRNX is most likely unwarranted. This is because that many entries in your SYSLGRNX table will almost certainly mean no logs can be avoided for reading, and that is the whole purpose of using this resource. So if the product gets a B37 abend while processing SYSLGRNX, you should most likely set this flag to X to bypass its use.

Bypass use of the SYSLGRNX table when using Modify Recovery
The "M" flag indicates that the product will automatically turn off SYSLGRNX processing if it detects MODIFY RECOVERY utility usage during the reported timeframe.

The product selects a log point of the highest deleted SYSLGRNX record among the filtered tables and compares it with the start log range specified by the user.

If the MODIFY RECOVERY utility was run on a tablespace that matches the filtering parameters, SYSLGRNX processing will be bypassed for all objects, even if this tablespace does not appear in the resulting report.

Thus if you have many object type filters or if using wildcard characters will result in a lot of object type filters ('object type' here means tables), it can be beneficial not to use option 'M' (bypass SYSLGRNX when Modify Recovery), but rather to skip SYSLGRNX usage totally (set Misc flags to 'X'). Miscellaneous flags 'X' (bypass SYSLGRNX) and 'M' (bypass SYSLGRNX when Modify Recovery) are mutually exclusive.

Include partial recovery points
This feature should only be used by Db2 experts; if you are unfamiliar with the concept of partial recovery, do not use this feature. With the default behavior, partial recovery points are determined and ignored, and message ALAA156I LOG RECS BYPASSED DUE TO PIT RECOVERY (GFL2) is generated in the job log. You can set this flag value to P to bypass this check and process those log records. A point-in-time (PIT) recovery can also limit the sources that Log Analysis Tool uses for a full row image formation process. This situation occurs only if you explicitly specify that Log Analysis Tool is to read log records within PIT ranges. In this case, the results are unpredictable when this value is set to P.

A PIT recovery is a partial recovery (performed with a Db2 RECOVER utility) that restores a set of Db2 objects to their state at a previous point in time. After a PIT recovery is performed on a set of objects, the Db2 log contains a range of log records for those objects that are no longer valid because the objects were recovered to a point before the log records were created. This range of invalid log records is called a PIT range.

It is expected that the only application of this setting is to allow Log Analysis Tool to report on log records within the partial recovery logpoints. For example, if you have already run a Db2 RECOVER, but still want to report on (or generate SQL for) log record activity within the recovery

logpoints, type P. If a Db2 RECOVER is not involved in the scope of your process, or it is but you have not run it yet, then it is unnecessary to set this field value to P.

Notes:

1. If you use this flag value, ensure that you remove the setting for subsequent runs.

2. If you use this flag value, a detail report run following this can only be run in log-forward mode; log-backward is not possible if partial recovery points are bypassed.

3. If you take actions or apply changes to your database based on the Log Analysis Tool output that combines PIT and normal log records, you might corrupt the data in your database. To successfully process log records within a PIT range, process the PIT log records in a separate run that covers only the PIT range.

Allow high-speed processing.
You can specify that Log Analysis Tool should use high-speed processing by typing H in the Misc flags field, but only the product can determine if it will use high-speed processing. It does not mean high-speed processing will be used, only that Log Analysis Tool will check if it can be used. Log Analysis Tool determines if it will use high-speed processing by using two factors:

- Misc flags ... H

  This flag setting stipulates that you have allowed high-speed processing.

- The Db2 log data processed

  All log records processed and selected for use are full row image log records. This means there are only log records in the result set from the general report that include the following:

  – UPDATEs with DATA CAPTURE CHANGES

  – INSERTs

  – DELETEs

  Any other type of log record means high speed processing cannot be used. For example, the inclusion of Mass Deletes, any type of LOAD log records, and UPDATEs without DATA CAPTURE CHANGES would all prevent the use of high speed processing.

The main benefit of high-speed processing is that it provides sites that set DATA CAPTURE CHANGES on their tables a significant performance improvement over normal processing.

High-speed processing also implies that current row images will not be presented under normal log-backward processing. You can elect to obtain current row images by key values (an option of details report processing only), but they will not be automatically provided as is normal using log-backward detail report processing. This is the primary reason high-speed processing is offered as an option, rather than strictly determined by the product. If current row images and log-backward detail report processing are required for any given run, do not use high-speed processing. There is no high-speed implication for log-forward processing in the detail report because log-forward does not normally provide current row values.

In general, your performance will benefit from high-speed processing. However, high-speed processing also means longer CPU times in your general report (as all the savings occur during the details report only in high-speed mode). Additionally, high-speed processing does not allow more filtering at the detail report level. Therefore, high-speed processing must be run as a combined general and details report run ("Generate Details=Y").

Use "L" setting to allow high-speed processing even with LOAD LOG YES activity.
This flag allows the use of high-speed processing as described above even when LOAD LOG YES activity is found during the reporting time period. All other restrictions have to be met before high-speed processing will be allowed.

When setting this parameter to 'L', be aware that some records may be reported by Log Analysis Tool as being inserted that actually existed before the LOAD LOG YES utility was executed. This parameter allows these LOAD LOG YES records to be processed in high speed by bypassing a check of a previous image copy for the prior existence of the row. This option will allow Log

Analysis Tool to process the LOAD LOG YES records faster by potentially reporting more rows as being inserted than actually were.

Mass deletes are still not eligible for high-speed processing, so the mass delete portion of a LOAD REPLACE LOG YES must be excluded by placing an 'X' in the "**Show DELETEs**" parameter.

The KEEP DICTIONARY setting must also be used for LOAD REPLACE LOG YES due to the dictionary not being processed from the log during high-speed processing.

Output flags

This field determines the type of output reports that will be generated from the general report run. The type of output report can also affect what you can do after the general report is run.

B

Bypass all reports. This means no general reports are created at all, and you are running solely for the purpose of creating inputs for a detail report run. This setting is mutually exclusive with all others.

Important: Any combination of G, S, X, and T values is valid. Omitting any of these values means that particular report will not be created. The default setting is GS.

C

Create the continuous mode file report. This report looks at what the contents of the continuous mode file will be at the completion of the job and provides some of the details. This can be used to identify activity that had occurred within the time frame of the run, but was not reported because the unit of work had not been committed.

F

Create the commit frequency report. This report displays information about the frequency of commits throughout the reporting time period. This report can be used to identify which plan, object, and / or activity type is taking too long between commits.

G

Create the general report.

I

Include impact reports for all the other specified reports (though there is no impact report component for a transaction ID report). See "Specifying an Impact report" on page 107 and the tutorial topic on "Impact Report" for more information on using this setting as it can have a major effect on the job in regards to performance. This type of impact report carries your filters forward.

J

Include impact report by row for all the other specified reports (though there is no impact report component for a transaction ID report). See "Specifying an Impact report" on page 107 and the tutorial topic on "Impact Report" for more information on using this setting as it can have a major effect on the job in regards to performance. This type of impact report does not carry your filters forward; it tracks only those rows reported on during the requested timeframe.

M

Write any Db2 commands found during the log scan to a command report.

If you only want to see Db2 commands, make sure all other filters are set to 'N' (i.e., Include INSERTs/DELETEs/UPDATEs all = N). Commands are not carried through to the detail report because there is no further information to be displayed beyond the general report level. If only commands are desired, do *not* specify 'Generate details=Y'.

Object type filters do not apply to commands. However, some miscellaneous filters do apply, such as authid, connection type, and the LUWID fields. The Db2 log records may not contain all of the information. For example, authid may not be available if the command is issued in a distributed environment (e.g., from a remote server). If a filter is used and it is possible that some are missing, retry with same timeframe but without the filter to ensure all Db2 commands are reported.

Considerations:

- If other reports are requested with commands, a return code of 2 may be issued if those other reports have no data, but there are Db2 commands reported. It is best to run Db2 Command reports separately to avoid confusion with other reporting actions.
- If other reports are requested with commands, and miscellaneous filters are provided, those miscellaneous filters are applied to both sets of reports. For example, assume you want to track TABLE X and see all of the Db2 commands performed by authid USERX. The USERX filter will be applied to both the Db2 Commands and any activity done against TABLE X. If that is not the intention, two separate reports must be run to accomplish this (one for TABLE X activity without an authid filter, and one for Db2 Commands with an authid filter).

Q

Create a quiet time report only. This flag setting behaves differently than any of the other flags. A detail report cannot follow a quiet time report. Quiet time is also mutually exclusive with all other Output flag settings. When you specify **Q** for an **Output flag**, another panel is displayed for more input. See "Specifying a Quiet time report" on page 109 and the online tutorial topic on quiet time processing for more information.

Important: All the filter criteria entered for a quiet time report is considered part of an AND condition. That is, if you specify two table spaces as filters, the quiet time reported on represents quiet time for both of those filters (not each one individually). Also, if object-level filters are used (for example, table space, table, or page), it is best to keep the filters at the table space level for quiet time reporting purposes. This is because Db2 CHECKPOINT records do not record inflight activity below the table space level. These Db2 CHECKPOINT records are used (from the beginning of a Db2 log) to determine inflight activity, but because the inflight activity is recorded at the table space level, quiet time reporting cannot check below the table space level, and therefore all object filters related to the affected table space must be considered inflight (that is, not quiet). This can result in a more limited range of reported quiet times at a table level, for example. For the most accurate results, if object filters are used, use table space filters only. Miscellaneous filters are also allowed for quiet time, enabling you to find quiet time for plans, for example.

S

Create the summary report. This report summarizes, by objects, all activity.

T

Create the transaction ID report. This report shows available distributed data information on two-phase commit activity, if any. This information is not always present on all two-phase commit activity, but if Db2 records these actions, Log Analysis Tool can report on those actions. See "Specifying a Transaction ID report" on page 106 for more information.

X

Create the extended report. This report breaks down every action into the page(s) and RID(s) it affected, and shows other information like LRSN and byte counts. This report can be extremely large in relation to the general report, so request only if absolutely necessary.

Additional report
BSDS Report

The BSDS report is automatically printed to the INFOM DD when running a General report, provided that the INFOM DD is in the JCL for the General Report step. If the INFOM DD is not in the JCL, then the report is not printed. The BSDS report provides two different types of information:

- The remaining time until the RBA and LRSN reach their limit.
- Job information that can aid in the support process.

Remaining time
The projections of when the RBA and LRSN reach their maximum values are created for all the logs contained in the BSDS and once again just for the logs that meet the timeframe of the Log Analysis Tool job. The accuracy of the projections depends upon how consistently the Db2 subsystem is being used. If the logging rates differ drastically from hour to hour and day to day, then the projections can vary widely for each run of Log Analysis Tool. For systems that

have a consistent logging rate, the projections should be similar from each run, especially the projection based upon all of the logs in the BSDS.

Support information

The BSDS report contains several different values found in the BSDS. Some of these values include highest written RBA, highest written LRSN, member number, and BSDS DSN, and the different log files that meet the analysis range before applying SYSLGRNX information. Requesting a DSNJU004 is a request that is made for some of our support issues and this will provide that information up front. Usually the DSNJU004 request is needed at the time the job was run, so requesting it later may be too late.

Log Range fields

These fields provide the range of log activity to be read. The ranges can be based on date/time, RBAs or LRSNs, and are all mutually exclusive. At least one value must be provided, and the others left blank.

Start/End Date-Time

Enter these fields in the format indicated on the panel. The Db2 logs are searched for any activity found between these date/time values. These fields are set to their default formats if blanked out. The default formats are considered equivalent to spaces internally, so they do not need to be blanked out if using RBA or LRSN ranges.

You can specify a relative End Date and End Time by entering a plus sign or a minus sign and three digits in the End Date field and blanking out the End Time field. The three-digit amount field represents the number of minutes to be used in the relative End Date / Time calculation.

- If you specify the plus sign, the amount is added to the Start Date / Time to obtain the End Date / Time.
- If you specify the minus sign, the amount is subtracted from the current date and time to obtain the End Date / Time.
- The amount must be greater than or equal to 000 and less than or equal to 1440.

Start/End RBA (URID)

These fields can be used in the following ways:

Start/End RBA filter

Enter a pair of starting and ending RBA values. You must enter these values in full hexadecimal display format with a length of 12, such as 0001ACCDEF02. If RBAs are desired, the date/time and LRSN log range fields must be blank. RBAs are relative byte addresses of log records.

URID filter

If you wish to filter on a single URID, the start RBA can be set equal to the URID, and the end RBA left blank. Only that specific URID will be reported on. In all other cases, the start and end RBA values are intended as a range of RBAs to process.

If you want to specify a set of URIDs, you can do so by including the first URID on the panel as described previously. Subsequently, you can edit the JCL and repeat those URID specifications up to 682 times (that is, there is a maximum of 682 URIDs you can specify for a single run).

For example, assume you specify the first URID = 001E668838B4 on a subsystem named SS1A. The JCL generated will have a section like this:

```
//DATAIN    DD   *
SSID          =SS1A
START RBA     =001E668838B4
END RBA       =000000000000
/*
```

If you want to include more URIDs, merely repeat the START/END RBA lines for the number of URIDs you want. For example, if you also want to specify 2 more URIDs of 001E66BBB000 and 001E66BC00EE, then the final edited JCL would look like this:

```
//DATAIN    DD   *
SSID          =SS1A
START RBA     =001E668838B4
```

```
END RBA          =000000000000
START RBA        =001E66BBB000
END RBA          =000000000000
START RBA        =001E66BC00EE
END RBA          =000000000000
/*
```

You must enter each URID as a group as shown previously. Each START RBA value must be non-zero; each END RBA value must be zero. Any deviation from this method is unsupported.

Important: If you use RBAs or URIDs as input, you cannot be using a data sharing group. RBAs have no uniqueness in a data sharing context, so if you wish to use them in such an environment, you can only do so going against a specific member of the group, not the group as a whole. The field 'Data Sharing Mode" should be set to N when using RBAs/URIDs as input and the Db2 subsystem should reflect a specific member, otherwise results will be unpredictable and unsupported.

Start/End LRSNs

Enter a pair of starting and ending LRSN values. These values must be entered in full hexadecimal display format with a length of 12, such as B877232DEDD0. If LRSNs are desired, the date/time and RBA log range fields must be blank. LRSNs are log recording timestamps associated with log records.

Continuous mode file

Type the name of the continuous mode file to specify continuous mode processing.

Continuous mode means that you want to run the reports (general with/without detail) continuously over time. See "Continuous mode processing" on page 113 for more information.

Resolve started UOWs

Y

Continue reading the Db2 logs until all uncommitted units of work have been committed or aborted.

N

Stop reading the Db2 logs at the end of the analysis timeframe regardless of uncommitted units of work.

You can specify that you want Log Analysis Tool to continue processing the log up to the current date and time in order to complete processing all units of work that began during the specified timeframe but ended after the timeframe. Prior to V3.2, Log Analysis Tool reported on units of work within a timeframe specified by the user, bypassing any units of work that had not been committed or aborted when the timeframe ended. Log Analysis Tool then provided other options for the uncommitted units of work.

This functionality allows Log Analysis Tool to continue reading the Db2 logs until all uncommitted units of work have been committed or aborted without reporting on any new units of work that start outside of the analysis timeframe. Performance may be negatively affected when running in Resolve Unit Of Work mode, as additional log records may need to be read, especially for long-running tasks. Log Analysis will continue reading log records until the unit of work is resolved or the time associated with the start of the job is reached, whichever comes first.

Restriction:

- Resolve started UOWs and showing uncommitted UOWs are mutually exclusive.
- Resolve started UOWs option is only available with the Date / Time ranges.
- Resolve started UOWs option is not available for the Quiet Time report.

A few examples will help you to better understand how Resolve started UOWs functions. Assume that a unit of work is started at 8:00 and is not committed until 8:10.

Example 1

Log Analysis JOB1 is run at 8:05 AM in non-continuous mode with a time range from 7:30 AM to 8:01 AM:

- Setting Resolve started UOWs to No in this scenario causes Log Analysis Tool to read the logs up to 8:01 AM, then issue warning message "ALAA070W-Uncommitted URID found and bypassed."
- Setting Resolve started UOWs to Yes in this scenario causes Log Analysis Tool to read the logs to 8:05 AM instead of 8:01 AM, then issue warning message "ALAA070W-Uncommitted URID found and bypassed."

Example 2

Log Analysis JOB2 is run at 8:05 AM in continuous mode with a time range from 7:30 AM to the current time:

- Setting Resolve started UOWs to No in this scenario causes Log Analysis Tool to read the logs up to 8:05 AM and then stores uncommitted/unaborted units of work in the Continuous Mode file.
- Setting Resolve started UOWs to Yes in this scenario causes Log Analysis Tool to read the logs up to 8:05 AM and then stores uncommitted/unaborted units of work in the Continuous Mode file.

Example 3

Log Analysis JOB3 is run at 8:20 AM in non-continuous mode with a time range from 7:30 AM to 8:01 AM:

- Setting Resolve started UOWs to No in this scenario causes Log Analysis Tool to read the logs to up to 8:01 AM, then issue warning message "ALAA070W-Uncommitted URID found and bypassed."
- Setting Resolve started UOWs to Yes in this scenario causes Log Analysis Tool to read the logs up to 8:10 AM, and report a whole unit of work.

As this example illustrates, when Log Analysis Tool is run with Resolve started UOWs set to Yes, the logs are processed beyond the requested timeframe as needed, up until current date and time in order to resolve any UOW that was not committed prior to the end date and time specified by the user.

64 bit men threshold

This field instructs Log Analysis Tool to stop processing the logs in the General Report after this percentage of the maximum amount of 64-bit memory has been exceeded. Values from 40 to 99 are allowed.

- A value of spaces or '00' turns this processing off.
- A value of 75 will instruct Log Analysis Tool to stop processing the archive logs in the General Report when more than 75 percent of the available 64 bit memory has been used.

A Detail Report is not allowed to be executed when the General Report uses this parameter.

Note: It is recommended to specify 'Y' in the 'Continue processing after warnings' parameter on the User Settings panel when using this parameter.

Override GMT offset

Y

Override the system defined GMT offset. When choosing to override the GMT offset, be sure to enter the proper GMT offset to use.

N

Continue using the system defined GMT offset.

with this GMT offset

Defines the GMT offset to use when overriding the system defined GMT offset. Specify the value with a leading '+' to indicate GMT plus some offset or with a leading '-' to indicate GMT minus some offset. The rest of the field should be specified in HH:MM format. If the desired GMT offset is GMT -5 hours, the value '-05:00' should be specified.

Restriction: "with this GMT offset" must be in the range of -12:00 TO +14:00

Important: Log Analysis Tool cannot report on log data that is still within the Db2 buffers and may not include any DML that was submitted just prior to running the report. You should, therefore, avoid using a

very narrow time frame near the current time. You should allow ample time for Db2 to unload its buffers of very recent activity or STOP/QUIESCE the object to cause Db2 to externalize the data to DASD.

Specifying filters for log data fields

The remaining fields on the "Generate database activity" panel act as filters to be applied to the log data. These filters are applied with an AND condition.

Only those log records matching all the filter criteria will be included in your report. Any conditions left blank indicate that the data should not be filtered out based on that filter. A single trailing wildcard (% only) may be used on all but the SHOW filters. For example, TABLE% is valid, but TAB%00% is not.

Show UPDATEs, INSERTs, DELETEs
Type Y in any of these fields to show the associated action (any type of UPDATE, INSERT (by LOAD or SQL), or DELETE) in the General Report, or N if you do not want a report on the associated action. INSERTs and DELETEs also allow you to type X to allow for unique exclusion situations:

INSERTs
X means exclude only INSERTs done via LOAD (that is, by utility, not by SQL). INSERTs done by a LOAD utility (LOAD/LOAD REPLACE) can be very expensive to process, and using this setting can alleviate that potential cost without affecting your need to see SQL INSERTs.

DELETEs
X means exclude only DELETEs done via mass delete.

Show Rollbacks
Rollback, in this context, means an aborted unit of work/recovery. It does not relate to the ROLLBACK TO SAVEPOINT SQL statement. Valid values are:

- Type Y to show rollbacks.
- Type N to exclude rollbacks.
- Type 0 to show only rollbacks.

Rollbacks will be indicated in both the general and detail reports (URID column heading will display as URID(R)), and the summary report has a separate section for rollbacks. All records that occur within the aborted URID will be reported on (that is, both original and compensation log records are reported on).

Compensation recs

- Type Y to show the compensation records in the reports when rollbacks are encountered.
- Type N to exclude compensation records when processing rollbacks.

This field is ignored when rollbacks are not selected.

Include LOB/XML data

Important: The term "LOB" is used in this section to refer to either LOB or XML data.

- Type N to exclude LOB data where LOB columns exist. This behavior will be identical to previous releases (pre-V3.1 of Log Analysis Tool). That is, any tables reported on that contain at least one LOB column cannot be processed for SQL, but can be processed for detail reporting. The LOB columns show as locator values only; the actual LOB data is not available.
- Type Y to include LOB data where LOB columns exist. Log Analysis Tool will now determine modified LOB data, and include it in generated SQL, if any is requested. The detail report will still show only the LOB locator values, but a separate report will be generated (if the detail report is requested) showing the LOB column data. It is very important to note that only modified LOB data is tracked, not unchanged LOB data. For example, if two rows are modified, but only one of them changes actual LOB data, then only that one row will have LOB data available. This is done to minimize the impact of tracking LOB data.

Due to the overhead involved in processing LOB data, it is highly recommended this field be set to N unless absolutely needed. Performance can be impacted greatly with a setting of Y.

Important: LOB/XML data can only be tracked in:

Two-phase mode
Two-phase mode means your "User Settings" panel has values of N (No) for both "Use 64 bit storage" and "Use work files" fields.

One-phase work file mode
One-phase work file mode means your "User Settings" panel has a value of N (No) for "Use 64 bit storage" field and a value of Y (Yes) for "Use work files" field.

Show uncommitted
If you set this field to N (No), Log Analysis Tool will exclude uncommitted units of work.

If you set this field to Y (Yes), Log Analysis Tool will include uncommitted units of work. All uncommitted units of work will be denoted as uncommitted, distinguishing them from committed or rolled back units of work.

Important: If you specify Y to **Show uncommitted** units of work, you cannot also specify a **Continuous mode file** nor can you specify Y to **Resolve started UOWs**. Continuous mode is mutually exclusive with these two settings because the showing of uncommitted units of work is in direct contrast with the purpose of continuous mode (which tracks uncommitted units of work across multiple runs of the product).

Uncommitted units of work are identified in various ways in the output of Log Analysis Tool:

General report
The URID heading is changed to URID(U).

Summary report
There is a section called "Uncommitted Activity" added to the report.

Extended report
The URID heading is changed to URID: *urid-value* (UNCOMMITTED).

Detail report
The URID heading is changed to URID(U).

SQL
The URID heading in the commented header is changed to UURID.

Loaded data
The loaded data can include an action field, if the header was included. This action field indicates actions, such as an update (U), insert (I), or delete (D).

If an update or delete was performed on behalf of a referential constraint, those actions are listed as (V) and (E), respectively.

If the actions were rolled back (aborted), the actions show a trailing "R". For example, UR is a rolled back update.

If the actions were not committed (uncommitted), the actions show a trailing "U". For example, UU is an uncommitted update.

The following table explains the values that can be found in the first byte of this two-byte field:

Table 10. Action values for first byte	
Value	**Type of Action**
D	Delete
E	Delete due to RI. (Delete performed due to a referential integrity constraint.)
F	Delete from trigger.
G	Delete from compensation log record (CLR).
I	Insert
J	Insert from trigger.

Table 10. Action values for first byte (continued)	
Value	**Type of Action**
L	Load utility
M	Mass delete
N	Insert from compensation log record (CLR).
P	Load replace utility
R	Reorg utility
U	Update
V	Update due to RI. (Update performed due to a referential integrity constraint.)
W	Update from trigger.
X	Update from compensation log record (CLR).

The following table explains the values that can be found in the second byte of this field, if needed:

Table 11. Possible values for second byte	
Value	**Description**
Blank	Committed Unit of Work
R	Rollback
U	Uncommitted

Include catalog data

If this field is set to Y (Yes), the report includes catalog data found in the Db2 logs. If this field is set to N (No), catalog data found in the Db2 logs is excluded from the report. The default is N.

If this field is set to Y, your reports will include all activity that was performed against the Db2 catalog tables. Set this flag to Y only if you want to specifically see these actions, as a considerable amount of data will be generated from this type of report run.

Even if this filter is turned on, other filters used for this report run may exclude catalog data. For example, if you specify a user-created table filter, this would eliminate all Db2 catalog tables from the report.

Misc filters

These miscellaneous filters allow to you filter on such things as AUTHID, plan name, job name, connection ID, and connection type. Type Y if you want to use such filters.

Object filters

These filters should be used when you want to filter on Db2 objects such as databases, table spaces, or tables.

N

No object filters.

Subsequent panels will display for all options other than N.

M

Specify objects by their names.

I

Specify objects by their IDs (DBID/PSID/OBID).

A
>
> A special type of filtering for objects which should be used only when you filter at the column level for a table. This advanced filter option means you will provide not only table level filters, but row and column level filters as well. Advanced filters require that you run a combined general and details report (that is, "Generate details = Y") because column and row level data is not truly available until the detail report is run. However, the table(s) specified here will be used at the general report level as filters.

Filter file usage
> Use this field if you wish to save your filters, edit previously saved filters, or just use previously saved filters.
>
> **N**
> > Do not use filter files.
>
> **S**
> > Save all filters generated from this invocation of the general report to the named file. This overwrites any previously saved filters with the new set of filters.
>
> **E**
> > Edit any previously saved filters within the file itself. This overrides all other panel settings for filters. In other words, you are going into EDIT mode to modify the contents of the filter file directly without the aid of any panel-driven processes. See the tutorial for instructions on editing the filter file properly
>
> **U**
> > Use the file as it is with no modifications. This also overrides all other panel settings for filters.

Filter file name
> This field must be non-blank if the **Filter file usage** field is set to anything other than N. This field identifies the name of the filter file you are currently using. If the file does not exist, the product will create it for you. If it does exist, it will be used. A PDS with a member name can be used here, but the product will create only the member, not the entire PDS. Do not use quotes around the specified file name; enter only a fully qualified data set name.

Generate details

Procedure

- If you type Y in the **Generate details** field to run a concatenated job of both the general and detail reports at the same time, the Generate details panel displays:

```
V3.5.0 --------------------- Generate details ---------------- SC01/SS1A
COMMAND.....
                                                        More:    +
Hit 'Enter' to accept detail report settings
Hit 'PF3' to bypass combined general/detail report

Detail report settings with combined general/detail report run:
*Log reading mode...... F     (A - Auto, B - Backward, F - Forward,
                               P - Fastpath, C - Combo,
                               D - Auto default backward)
*Commit scope.......... 000   (000-999, UOW)
*Generate Undo SQL..... N     (Y - Yes, N - No,
                               X - Yes, exclude triggered actions)
*Generate Redo SQL..... N     (Y - Yes, N - No,
                               X - Yes, exclude triggered actions)
 LOB PDS name..........
 Set Current SQLID.....
*Archive output........ N     (Y - Yes, N - No, S - Specify archive options)
*Current row by key.... N     (Y - Yes, N - No, X - Exclude current rows)
*WHERE clause by key... Y     (Y/N)
*Output flags.......... D     (B - Bypass reports, D - Detail,
                               T - Transaction, I - Impact,
                               C - Catalog Audit, S - Summary)
*Report row length..... 00080 (00080-32752)
*Treat mass deletes.... I     (I - As individual delete statements,
                               O - As one mass delete statement)
*CHAR data in hex...... N     (Y - Yes, N - No)
*Show varchar length... N     (Y - Yes, N - No)
 Only changed columns.. N     (Y - Yes, N - No)
```

Figure 22. Generate details panel

- **Generate details panel fields:**

  Log reading mode
  This field determines how the detail report will be generated.

  A

  Log Analysis Tool automatically determines the best way to generate the detail report. Use this setting if you are unsure of how best to proceed, or if you want the product to make the determination for you. This should be the normal setting.

  If you type A, and the product determines log-backward mode, it is still possible to get warnings due to message ALAA061W (log-RID mismatches). If you set the Defaults panel field **Continue processing after warnings** to N, the product will attempt to switch automatically to log-forward mode. If this is not the behavior you want, either change the Defaults panel field, or use the F or B selections for log processing mode.

  B

  Force a log-backward process (LB).

  F

  Force a log-forward process (LF).

  P

  FASTPATH type processing (FP). FASTPATH is a method in which no attempt is made to materialize full row images. FASTPATH is only allowed when running a combined general and detail report, and it does not allow the use of advanced object filters. See the tutorial for more information on FASTPATH processing.

  C

  Combo mode processing. Use a combination of log-forward and log-backward processes, if possible.

  This option is a request to run in combo mode; it does not guarantee it will do so. Log Analysis Tool determines if some objects can be run in log-backward mode and some in log-forward mode during the detail report run. If that is true, then and only then will combo mode actually be chosen. If the product finds all objects can run in log-backward mode, it will still chose log-backward mode. If it determines all objects must run in log-forward mode, it will choose log-forward mode. It is only when a combination of both modes is

possible that this method will be selected by the product, and only if you originally selected combo mode.

For example, assume two table spaces are in the result set after the general report component runs. If only one table space has done a subsequent REORG, that table space must run in log-forward mode; the other table space can likely run in log-backward mode. This now means only one of the table spaces' image copies needs to be accessed. Performance may (or may not) be positively affected when running in combo mode. It is impossible to tell ahead of time. Though image copy access can be avoided for some objects, additional Db2 logs may also be read. Therefore, combining both modes in one run may lead to extra Db2 log access which may not have occurred had only one of the modes (backward or forward) been chosen. The product cannot determine if this potential Db2 log access increase is offset by image copy avoidance. You must determine if this option provides you with a benefit.

Restriction:

- Fast path mode is mutually exclusive with Combo mode because FP does not materialize full row images.
- Hi speed mode, if selected by you AND the product, effectively ignores Combo mode for same reason (that is, full rows are not materialized as hi speed mode requires all rows to be full initially).
- Combo mode is not allowed with expert options
- The option **Specify logs** means LF only, a combination of modes is not possible (you can request it, but it will not be used).
- Dropped object mode must run in LF mode (you can request it, but it will not be used)
- Work file mode not allowed with Combo mode.

D

Auto default backward. Log Analysis Tool determines if log-backward mode is possible, that is, no reasons exist that force log-forward mode. If that is true, then Log Analysis Tool will choose log-backward mode.

Performance may or may not be positively affected when running in Auto Default log-backwards mode. With this option, if log-backward mode is possible, then it will be used regardless of how many log records must be applied to the current row image to get back to the analysis time frame.

For more information on log processing, see the tutorial and .

Commit Scope

There are two distinctly different types of values that you can enter for Commit scope:

000–999

A three-digit number from 000 to 999 indicating the number of SQL statements to execute prior to issuing a COMMIT statement. There is also an implicit COMMIT at the termination of the SQL.

If set to 000, no COMMIT statements are added to the SQL flat file. This value is ignored if no SQL flat file is requested (both Generate undo SQL = N and Generate redo SQL = N).

Note: Even though no COMMIT statements are added to the SQL file, an implicit COMMIT occurs at the termination of the SQL.

UOW

This feature allows the product-generated SQL to be committed and executed in unit-of-work order. The SQL is grouped by end unit of work time and sorted by the RBA or LRSN within the end unit of work. Additionally, the commits are issued as soon as the unit of work is finished. This is in contrast to the preceding method of committing by the number of SQL statements executed only.

For example, in Log Analysis Tool V3.2, if the updates are done similar to the sequence shown in Example 1, then the resultant SQL file will be created, as shown in Example 2 (this presumes a REDO operation). The term 'transaction' refers to a unit of work.

```
Example 1: Transaction sequence in DB2:
  Transaction A updates record X
  Transaction B updates record Y
  Transaction A updates record Z
  Transaction A ends (end unit of work)
  Transaction B updates record X
  Transaction B updates record W
  Transaction B ends (end unit of work)
```

```
Example 2: Resultant generated REDO SQL statements
    by Log Analysis Tool:
  Transaction A updates record X
  Transaction A updates record Z
  COMMIT A
  Transaction B updates record Y
  Transaction B updates record X
  Transaction B updates record W
  COMMIT B
```

Restriction: If you specify UOW for commit scope during an archive run (option #6 off the main menu), the timeframe must not include older archives that were created in releases prior to V3.2. This option is not available on archives created in prior releases, as those archives would not contain the necessary information. Only release V3.2 (and subsequent releases) will handle this option within the context of archiving.

Generate Undo SQL

Type Y to generate undo SQL as the final step in the Details Report process. This creates a flat file holding SQL that can be used to undo the log modifications reported on in the Details Report. The generated SQL exists in the SQLOUT DD after the job successfully runs.

Type N if you do not want the undo SQL step added to the job stream.

Type X if UNDO SQL is not needed to back out actions resulting from triggers firing.

This field is mutually-exclusive with generating redo SQL.

Generate Redo SQL

Type Y to generate a flat file containing SQL to redo the changes reported on. The generated SQL exists in the SQLOUT DD after the job successfully runs.

Type N if you do not want the redo SQL step added to the job stream.

Type X if REDO SQL is not needed for actions that originally resulted from triggers firing.

This field is mutually-exclusive with generating undo SQL.

LOB PDS name

The term 'LOB' as it is used on this panel refers to either LOB or XML column types. This field identifies the name of a PDS (library) that will be used by the product to contain LOB data. It only has effect if there is LOB data in your result set. If not, it is not used. If LOB data does exist, this PDS will be populated with one member per LOB column data per modified row. For example, if you have 3 INSERTs in your result set, and all INSERTs contain 2 LOB columns each, then there will be 6 members added to this library.

The naming convention for the members is Ln, where n = 0000001 through 9999999. If the member pre-exists, it will be overwritten. The attributes of the PDS must be this: LRECL=32756, BLKSIZE=32760, RECFM=VB,DSORG=PO. The number of directory blocks depends on the number of needed members. The PDS will be created for you if it does not pre-exist, with default values for space (that is, SPACE=(CYL,(100,100,50),RLSE)).

If you wish to have complete control of this PDS, create it outside of the product and the pre-existing PDS you create will be used instead. You must use the file attributes stated herein if you create it outside of this product.

The name used must be a valid PDS name. For example, ALA.LOBS1 is a valid name, whereas ALA.LOBS(L1) is not (that is, do not include member names here). Do not quote the PDS name, it will be used as entered.

If no name is given, and LOB columns exist in your tables, the SQL generation process will fail. This PDS is used as the 'file by reference' component when processing LOB data in SQL.

Set Current SQLID

Enter either spaces or a valid SQLID into this field. If this field contains anything other than spaces and UNDO SQL or REDO SQL has been requested, a SET CURRENT SQLID statement will be inserted into the SQLOUT file with this value. If this statement is not needed or wanted, then fill out this field with blank spaces.

Archive output

This field allows you to archive (save) the results of your detail report run. Type N to bypass this feature. Type Y to save your results to a specified data set. Type S to specify that data set, and other required information. Before you can use the Y setting, you must use the S setting at least once to specify this required information. For more information on archiving output, see "Specify archive information" on page 105.

Current row by key

Type Y to have the product attempt to retrieve the current row image of each reported log action (update, insert, delete) based on the PRIMARY key of the post-change row. Type N to bypass this process. If you type Y, keep in mind these points, especially the fact that this option can be EXTREMELY resource intensive (see #4):

1. This process cannot work for any table that is not uniquely keyed.

2. If any table is not uniquely keyed, the results are different depending on the log reading mode (log-forward or log-backward):

   - If it is log-forward, the current row values are not normally available, therefore, all current rows are displayed as empty (dashes in all column values).

   - If it is log-backward, the current row values are normally based on physical page and row id (RID), so those values will be maintained (as if this field was set to N for those tables).

   Warnings are written for each table that cannot resolve current row values based on keys.

3. If the key returns no row, the current row will display as empty.

4. This process can be extremely resource intensive. Do not request it unless absolutely necessary. Keep in mind that the current row values are valid only during the run; these row values may be different immediately after (or even during) the run if the affected tables are constantly being updated (in which case this option should not be used).

Type X to omit current rows altogether. This can be a saving in both performance and DASD space. It has no effect if log-forward mode is used as log-forward mode does not obtain current row images at all. If using log-backward mode, however, a value of "X" means current rows will not be captured into internal files nor reported on. This can mean a savings of 33 percent in some of these internal data sets. It can also mean that table space access is not required if all log records have full row images. If current rows are not mandated for your site, use of this setting is recommended. Current row images are only used in detail reporting, so no other back-end processes would be affected by omitting current row images altogether.

WHERE clause by key

Type Y to have the product use only the key columns of any tables in all WHERE clauses. This is the default.

Type N to have no key columns obtained and all columns are used in WHERE clauses.

It is strongly recommended that you use the default value Y only. If no key columns are available for any given table, the product must use all columns regardless of this setting. If neither UNDO nor REDO SQL is requested, this field setting is irrelevant as no WHERE clauses will be generated.

Output flags

This field determines the type of output reports that will be generated from the detail report run.

B

Bypass all reports. This means no detail reports are created at all, and you are running solely for the purpose of creating output such as SQL or loadable format data. This value is mutually exclusive with the other values.

Important: Any combination of D and T values is valid. Omitting any of these values means that particular report will not be created. D is the default setting.

D

Create the detail report.

T

Create the transaction ID report. This report shows available distributed data information on two-phase commit activity, if any. This information is not always present on all two-phase commit activity, but if Db2 records these actions, Log Analysis Tool can report on those actions.

I

Include impact reports for all the other specified reports (though there is no impact report component for a transaction ID report). This also means all activity associated with the impact report will be carried forward to the output from the detail report run. Impact report activity is not included in SQL (if so requested), but is included if the data is loaded into Db2 (that is, option "Load details into DB2" off the primary menu). If you requested an impact report in the general report run, you must explicitly include it on this panel if you want to see the detail report level of that activity. Including an impact report now if it was not included on the general report panel will have no effect (no impact report data is available).

See "Specifying an Impact report" on page 107 and the tutorial topic on "Impact Report" for more information on using this setting as it can have a major effect on the job in regards to performance.

C

Create the Catalog Audit report. This report looks at the catalog table data and uses this to display information about Db2 objects that were created, altered, and dropped. To generate a catalog audit report, catalog data must be included in the run. This is done by specifying Y for "Include catalog data" on the "Generate database activity report (general)" panel.

Restriction: All appropriate Db2 catalog tables must be included in the filter set.

S

Create the Details Summary report. The Details Summary Report is a summary of the activity in the Details Report. The counts include how many rows were inserted by the LOAD utility and how many rows were deleted by a mass delete operation. The counts are accurate because all of the filters will have been applied.

Report row length

Enter a 5 digit number from 00080-32572 which will indicate the maximum number of characters that will be displayed per line in the detail and impact report for the current, post-change, and pre-change rows. This allows very large rows to display all data on one line instead of spanning multiple lines. Specifying a value of "00080" will result in similar output to that of prior versions of this product.

Treat mass deletes

Type I (the default value) to have the product materialize the mass deletes individually. Prior versions of this product materialized mass deletes individually. Type 0 to have the product report only that a mass delete occurred and if using Redo SQL generate a single delete statement. Undo SQL requires this field to be set to "I" so each row that was deleted can be materialized into an insert statement. The materialization of mass deletes requires an image copy and larger quantities of storage because of the possible large number of statements that were deleted.

By having Log Analysis Tool report only that a mass delete occurred and creating a single delete statement for Redo SQL, resources can be conserved as image copies now may not be needed and individual rows do not have to be materialized. A Mass Delete statement still requires log-forward processing. If Mass Delete statements are not materialized, they will not be able to be loaded into the Audit Table. Also, the load back into Db2 option for non-materialized Mass Deletes will load a row with default or null values with a MD indicator in the action field. Be aware of the limitations of loading and auditing when selecting this feature.

For Mass Deletes on tables that are defined with Data Capture Changes, Db2 logs the individual delete statements and Log Analysis Tool will continue to treat them as individual deletes.

CHAR data in hex

Type Y only if you want all character type data displayed in hexadecimal format. CHAR data means fields defined as CHAR or VARCHAR in your table definitions. Hexadecimal means binary data.

For example, rather than 'ABC', an EBCDIC CHAR(3) field will display as X'C1C2C3'. This should not be used unless you plan on generating SQL and your site places hexadecimal data (with or without character data) into character type fields. The reason this option exists is that it is possible to enter hexadecimal data into character type fields through some method that is not supportable via SQL. In those cases, the SQL generated by this product may not run due to translation errors on that character data. The only way to circumvent such Db2 SQL errors is to use the hexadecimal format, bypassing further Db2 translations. Such SQL errors may be -404 (data too long for field). Should you receive such errors, the job must be re-run using a Y value in this field. The default is N, and should remain that way unless such SQL errors occur at your site.

Typically, if you populate/update tables without using the hexadecimal format via SQL, the hexadecimal format specification should not be necessary. It is only if you use the hexadecimal format, or some other loading method such as the LOAD utility, that the character format of these fields may not successfully be processed by this product. And in those cases only, set this field to Y and retry.

If set to Y, all output will contain character data in hex, including detail reports and SQL. Therefore, if SQL is not to be generated, do not set this field to Y.

Important: The hexadecimal values used will be as they would exist in Db2; there is no translation done. For example, if you set this field to Y and run against a Unicode table, the hexadecimal values will represent Unicode (not EBCDIC).

Show varchar length

Specify Y only if you want to display the length of each varchar data value in the Details Report. If you do not want to see the length of the varchar data fields displayed along with the varchar data, then leave this field with the defaulted value of N.

Examples

This row from the report shows the length of the VARCHAR data and the data:

```
ROW STATUS   ROW_ID ROW_VARCHAR ROW_CHAR ROW_VARCHAR2
-----------  ------ ----------- -------- ------------
POST-CHANGE  -      -           -        -
PRE-CHANGE   +1     (+5)AAAAA   A        (+5)ZZZZZ
```

This row shows how an empty string, not a null value, is displayed – notice the +0 or zero length:

```
ROW STATUS   ROW_ID ROW_VARCHAR ROW_CHAR ROW_VARCHAR2
-----------  ------ ----------- -------- ------------
POST-CHANGE  -      -           -        -
PRE-CHANGE   +2     (+5)BBBBB   B        (+0)
```

This row shows how a null value is displayed – notice the +0 or zero length plus the keyword null:

```
ROW STATUS   ROW_ID ROW_VARCHAR ROW_CHAR ROW_VARCHAR2
-----------  ------ ----------- -------- ------------
POST-CHANGE  -      -           -        -
PRE-CHANGE   +4     (+0)NULL    D        (+0)NULL
```

This row shows how VARCHAR data padded with spaces is displayed:

```
ROW STATUS   ROW_ID ROW_VARCHAR ROW_CHAR ROW_VARCHAR2
-----------  ------ ----------- -------- ------------
POST-CHANGE  -      -           -        -
PRE-CHANGE   +8     (+2)A       H        (+5)A
```

Only changed columns

By default, this option is "N" (No) and full row images are displayed in the report. Set this field to "Y" (Yes) if you want the detail report to be generated with only the changed columns printed in the report. In this case the output for INSERTs and DELETEs will not change, because such actions involve full row changes. Update actions will be formatted to display columns that changed along with primary key column(s) to allow identification of the table row. If the table doesn't have a primary key, the option to print "only changed columns" is ignored, the corresponding warning message is issued, and full row images will be displayed in the detail report output.

Specify archive information

This topic details the Specify archive information panel.

When you type S to specify the archive data set, the Specify archive information panel appears:

```
V3.5.0 ---------------- Specify archive information ------------- SC01/SS1A

Hit 'Enter' to accept archive settings and archive the output
Hit 'PF3' to bypass archiving output (will not save settings)

Archived output table specifications:
*Archive table creator..===> PDUSER
*Archive table name.....===> ALA_ARCHIVE_TABLE
*Archive database.......===>
*Archive tablespace.....===>

Archived output dataset specifications:
*Archive dataset prefix.===> ALA.ARCHIVE
*Archive dataset unit...===> SYSDA
*Archive retention......===> 0000




COMMAND ===>
```

Figure 23. Specify archive information panel

Use this panel to specify your archived output settings. You are specifying the needed archive table (very similar to Db2's BSDS; the table specifications are in a following section) which acts as a repository for Log Analysis Tool archived data sets, as opposed to Db2 archived logs. Also, you must identify where this table will exist (that is, its database and table space), the data set prefix (its suffix is fixed by the product), and its unit name (it must be disk, though you can certainly allow HSM migration to tape). See the tutorial and <u>Chapter 10, "Process archived output," on page 161</u> for more information on the archived output process.

Specify archive information panel fields

Archived table creator

Enter from 1-50 characters for a table creator name.

Archived table name
Enter from 1-50 characters for a table name.

Archived database
Enter from 1-8 characters for the database which will contain this archive table repository. This database must pre-exist.

Archived tablespace
Enter from 1-8 characters for the table space which will contain this archive table repository. This table space must pre-exist within the named database.

Archive dataset prefix
Enter up to 27 characters here to be used as the prefix for your archive data sets. The remaining characters of the data set are set by the product, and are used for data set uniqueness.

Archive dataset unit
Enter up to 8 characters here representing a valid unit specification for your archive data sets. For example, SYSDA or SYSALLDA. Do not specify a tape unit here, though the data set can be a candidate for HSM migration.

Archive retention
Enter a 4 digit number here representing a valid retention period. The archive data set will be created with this retention period. Valid values are 0000-9999. This has no effect on existing data sets, only a newly created archive will be given this retention period.

Examples of valid values are:

- 0010

- 0365

- 1000

Examples of invalid values are:

- 3

- 28

- spaces

The value must be 4 digits long and not spaces. The value 0365 indicates that you want the retention period to be 365 days. The value 0000 indicates that you want no retention period and is the default.

Archive table specifications

Only one table is used for archiving. The table that Log Analysis Tool creates has the following columns:

```
START_DATE CHAR(8) NOT NULL,
START_TIME CHAR(6) NOT NULL,
END_DATE CHAR(8) NOT NULL,
END_TIME CHAR(6) NOT NULL,
DSNAME CHAR(44) NOT NULL
```

One row for each run of Log Analysis Tool that requested the details to be archived are entered into this table, so space is based upon how many times you request that to occur. Use the Db2 defaults for primary and secondary sizes–SEGSIZE is not needed or can be 4.

Specifying a Transaction ID report

Log Analysis Tool can filter and report on transaction IDs when Db2 writes the log records providing the information about distributed transactions. When Db2 is performing two-phase commits, log records containing information about the end user, transaction ID, accounting string, transaction/application name, and workstation name may be written. It may be quite likely there is no data to report, however.

Important: Not all distributed applications write the distributed data to the log. If you expect to find distributed data written to the log but the distributed report does not show any data, run a DSN1LOGP to determine if the data exists.

The Transaction ID report can be generated from either the General or the Details report processes and the transaction ID filters are considered part of miscellaneous filters. These fields, called XID fields from here on, can also be saved into Db2 audit tables just like any other header information. Because the fields reported on here are all defined as VARCHAR(256), this information is not automatically included in your reports because of potential size issues.

To request an XID report, or Distributed Report, specify T in the **Output flags** field on either the general or detail report panels. To request filters on XID data, type Y in the **Misc filters** field on either panel as well, and include any XID filters on the subsequently displayed Miscellaneous Filters panel. See "Using miscellaneous filters" on page 117 for more information. Including XID data in LOAD detail (as part of your header) is also determined by panel settings; see "Customizing headers during LOAD" on page 149.

Specifying an Impact report

When you specify that you want to generate an impact report, you are essentially requesting that Log Analysis Tool extend your requested end date and time to the current date and time, but to report the information separately from your original timeframe. The purpose of this is to identify what happened after your requested timeframe and, therefore, its potential impact on the activity associated with your original timeframe. You could accomplish the same results by running two separate reports with two different timeframes, but Log Analysis Tool provides it as a one-step option to minimize the use of resources. Because an impact report request always means running to the current date and time, you should use it with extreme caution. If your requested end date and time is not near the current date and time, this could result in an extremely long running job. Use this option only when absolutely necessary.

Since only reports are generated from the extended time period, use this option only for relatively small expected result sets. For example, if thousands of rows are affected, going through such a report manually may be far outside the scope of any usefulness. This type of report is best when used for exception processing only, and only on a small scale.

There are 2 types of impact reports; they are mutually exclusive:

Impact (Output flags=I)
    This an extension of the timeframe including the original filters. The extended time period tracks all activity based on your original filters through to the current time.

Impact by row (Output flags=J)
    This an extension of the timeframe through to the current time, but without regard to your filters. It extends the time period by tracking only those rows that are in the result set of your normal timeframe. For example, assume your original filter was on a particular userid for a time period of 10-11 AM, and 10 rows in 2 different tables were modified during that time period by that userid. Also, assume the current time is now 11:30 AM. Running an impact report by row means that those 10 rows will subsequently be tracked for further activity regardless of who performed the modifications through to 11:30, the current time.

The two different types of impact reports are relevant only during the General Report run. During the Detail Report run, if you want the impact report results at the detail level, use the setting of "I" to request impact report data from a detail report panel. Both are treated identically during the detail report, so only a specification of "I" is needed if you want impact report details.

For impact report by row (Output flags="J"), several extra rules and restrictions apply due to the fact that this option requires filtering at the physical row level:

1. If a DELETE type action is found during the normal or extended time period, that physical row is considered terminated, and no further tracking on it is done. Therefore, if a subsequent INSERT occupies the same physical row, that is not considered the same as the deleted row and will not be tracked.

2. If a REORG or LOAD REPLACE is found in the extended time period, tracking can only be accomplished up until that point. Row correlation across such utilities is not possible within Db2.

3. If any LOAD is found in the normal time period, those rows are not tracked across the extended time period.

4. If any MASS DELETE type actions are found in the normal or extended time period, they will be treated as a single DELETE action and not materialized into many individual deletes. This treatment is identical to setting option "Treat mass deletes=O" on the detail report panels.

5. Advanced filters cannot be used with this option. Advanced filters effectively track rows by column values whereas impact report by row tracks rows by physical position; the two are incompatible.

An impact report is initiated by setting the general report panel field "Output flags" to include an "I" or a "J". You cannot initiate an impact report from the detail report panel, but you can omit it at that level. It may be necessary to just see, for example, the impact data at a summary level. Therefore, you could initiate the impact report on the general report panel by setting the "Output flags" field to 'SI', and then set the same field on the detail report panel to 'D' (detail report only with no impact report). The way it works is that if you want an impact report, it will be included with all other reports (except transaction ID {distributed data} reports). There is no stand-alone impact report. So requesting an impact report only with no other reports requested yields no reports at all.

For example, if you set the "Output flags" field to "GSI" on the general report panel, and impact data is found, both the general and summary reports will produce impact reports. If your JCL contains a DD statement to contain the impact report section for the requested report (that is, there is a GENIMP DD in the JCL when requesting the general report and impact report), then the impact report will be found in the associated DD. If your JCL does not contain the impact report DD statements, then the heading "IMPACT REPORT BEGINS" denotes where the original report ends and the impact report begins.

It may be that no impact data is found, and therefore, no such section would be found in your reports. This would likely only occur if the original requested end date/time was very close to the current date/time. This same method of reporting impact data applies to the detail report as well.

There is no difference in the formatting of impact data within any given report to reporting pre-impact data in that report. For example, summary data within the impact report section is formatted exactly the same as summary data prior to the impact report section in a summary report. This holds true for all of the reports.

As mentioned previously, using impact reports can cause an otherwise fast-running job to run very slowly to handle all the extra data. Therefore, if the general report level reports yield enough information to make a decision on subsequent processes, it is strongly recommended that the impact data be filtered out of detail level reporting.

Impact report data is carried through to detail level files (if you requested), but it is never included in SQL generation. If you want SQL generated from your original requested start date/time to the current date/time, that should be your request rather than using an impact report. Impact report data is meant for analysis to make certain decisions on how you want to proceed, but not for UNDOing or REDOing SQL beyond your original timeframe.

Impact report data does carry over to the backend load process if you want to examine the results via a Db2 table. The impact data is denoted by the row-status column. Normal row-status column values are marked as "POST" and "PRE" (meaning after the change and before the change, respectively). Impact data rows will have row-status column values of "POSI" and "PREI" (meaning after the change and before the change, respectively, for any activity occurring after your requested end date/time).

Impact report constraints

This section summarizes the implications and rules of requesting an impact report:

- Your end date and time on the general report is changed to the current date and time. All report headings continue to reflect your original requested date and times, if present on the report heading.
- Impact reports cannot be used with continuous mode.
- Impact reports cannot be used with RBA or LRSN log ranges; only date and time inputs are allowed.
- All requested reports have an impact report section contained within them if any impact data is found, except for the distributed/transaction ID reports.
- Impact reports cannot be initiated at the details report level. If you select this in the detail report, but did not select it on the general report panel field, it has no meaning and is ignored.

Impact report example

This is an example of how the impact report works:

Assume the following series of times and SQL statements:

```
12:00:00  UPDATE TABA SET COL1 = 'A';
```

```
13:00:00  UPDATE TABA SET COL1 = 'B';
```

```
14:00:00  UPDATE TABA SET COL1 = 'C';
```

The current time is 14:10:00.

If you want to see activity from 11:00:00 to 13:30:00, you can set the start time to 11:00:00, and the end time to 13:30:00 on the general report panel (ignoring dates for this simple example). You also want to know if anything occurred after 13:30:00, so you request an impact report. You specify "GSI" in "Output flags", and run the general report only job. The general report would show the updates at 12:00:00 and 13:00:00 as normal. The impact report would show the update at 14:00:00. The summary report would show two updates as normal and the impact report would then show a single update.

The impact report is written to the GENIMP DD (General report), DTLIMP DD (Details report), EXTIMP DD (Extended report), or the SUMIMP DD (Summary report) if those DDs are present in the JCL. If not, then the Impact report will be part of those reports (General, Details, Extended, or Summary) as a separate section.

Specifying a Quiet time report

Quiet time processing generates a report listing periods of inactivity found in your logs for all the filters you requested. The time must be a period of time greater than or equal to the quiet time threshold that you specify. For example, if you specify a quiet time threshold = 00:30:00 (30 minutes), this means you want to see any 30 minute periods (or longer) in the log where none of the filter criteria had any activity.

When you type Q in **Output flags** and press **Enter**, the Quiet Time Specifications panel is displayed.

```
V3.5.0 ---------------- Quiet Time Specifications --------------- SC01/SS1A
Command ===>

Hit 'ENTER' to process panel; 'PF3' to exit and bypass quiet time mode

*Quiet time threshold value...... 00:30:00 (HH:MM:SS, max = 23:59:59)

*Save quiet times................ N        (Y/N)
 Database........................ SYSTOOLS (must pre-exist)
 Group tablespace name........... ALATSQ1  (must pre-exist)
 Group table owner............... SYSTOOLS
 Group table name................ QTG_NAME
 Group index owner............... SYSTOOLS
 Group index name................ QTX_NAME
 Quiet times tablespace name..... ALATSQ2  (must pre-exist)
 Quiet times table owner......... SYSTOOLS
 Quiet times table name.......... QTQ_NAME
```

Figure 24. Quiet Time Specifications panel

Use this panel to specify your desired quiet time threshold value. Additionally, you can request to save the returned quiet time values into a set of Db2 tables by typing Y in **Save quiet times** and specifying the table names. (See the online tutorial on quiet times for a full description of these tables). If you do not want to save quiet time information in a Db2 table, you are only required to specify a threshold value.

Important: Pressing **PF3** or cancelling out of this panel turns off quiet time mode and returns you to normal general report mode.

Quiet Time Specifications fields

Quiet time threshold value
This value is formatted by hours, minutes, and seconds: HH:MM:SS. A maximum 24-hour timespan is allowed (from 00:00:00 to 23:59:59). Clearly, very small threshold values will not have much significance in displaying useful gaps of quiet times.

The threshold represents the minimum amount of time that no log activity was encountered for all the matching filters. For example, if you wish to know when table space A and B had no activity over a 2 hour contiguous time period, specify a value of 02:00:00.

Save quiet times
Type Y if you want to save the reported quiet times into a set of Db2 tables. If Y, you must specify all of the fields that follow this one. If N, all other fields that follow this one are ignored. Log Analysis Tool will create these tables and indexes if they do not already exist. The database/table space fields must reference a pre-existing non-partitioned database/table space into which these tables will be created (Log Analysis Tool does not create the database/table space).

Important: All owner and name fields can be a maximum of 45 characters.

Database
Type the name of a pre-existing database that will contain the identified tables.

Group tablespace name
Type the name of a pre-existing table space within the identified database that will contain the group table.

Group table owner
Type the owner (creator) value for the Db2 table that will hold the group values. This is the table which contains information about the filters used in creating the quiet time report.

Group table name
Type the name of the Db2 table that will hold the group values. This is the table that contains information about the filters used in creating the quiet time report.

Group index owner
Type the owner (creator) value for the Db2 index that will be used for the group table.

Group index name
Type the name of the Db2 index that will be used for the group table.

Quiet times tablespace name
Type the name of a pre-existing table space within the identified database that will contain the quiet times table.

Quiet times table owner
Type the owner (creator) value for the Db2 table that will hold the actual quiet time ranges (start/end RBA or LRSN).

Quiet times table name
Type the name of the Db2 table that will hold the actual quiet time ranges (start/end RBA or LRSN).

Press **Enter** to continue.

Quiet time tables

On this panel you specify the names of two tables used to hold the quiet time data. The first Db2 table contains all the filters used in resolving the quiet times. There is an internally generated key value associated with each filter. This table is called the group table. The GROUPID column is part of the primary key, and can be used to reference individual reported quiet times in the second table. This second table is called the quiet times table. The tables are formatted and described here:

GROUP TABLE

GROUPID SMALLINT NOT NULL PRIMARY KEY
References rows in quiet times table.

GROUPID-NUM SMALLINT NOT NULL PRIMARY KEY
>Key, filter counter.

FILTER-IDENTIFIER CHAR(1) NOT NULL
>Internal filter identifier which describes FILTER NAME type:

>**blank**
>>No filters

>**S**
>>Database/table space

>**T**
>>Table owner/name

>**A**
>>Authid

>**J**
>>Jobname/corrid

>**P**
>>Plan name

>**I**
>>Connection ID

>**Y**
>>Network name

>**L**
>>LU name

>**U**
>>Uniqueness vale in LUW

>**C**
>>Commit count in LUW

>**1**
>>XID

>**2**
>>XUser

>**3**
>>XWorkstation name

>**4**
>>XApplication name

>**5**
>>XAcctg string

>**Important:** Filters such as rollbacks, inserts, deletes, updates are not included in this table. Only object and miscellaneous filters are tracked. So two different runs with no object or miscellaneous filters will be considered identical. That is, they will both have FILTER-IDENTIFIER = ' '. All other fields are also spaces in this scenario, except the identifying GROUPID* fields.

FILTER-NAME VARCHAR(257) NOT NULL
>Filter name. If wild cards are used, they are shown here (that is, just the wild card, not each resolved object).

FILTER-PART CHAR(4) NOT NULL
>Partition, if any, or 0000.

FILTER-EXCLUDE CHAR(1) NOT NULL
>Y if the filter is exclude, otherwise it is N.

FILTER-COND CHAR(1) NOT NULL
>Conditional (A = AND, O = OR).

QUIET TIMES TABLE

GROUPID SMALLINT NOT NULL
> Reference to GROUP TABLE filters.

START-TIMESTAMP TIMESTAMP NOT NULL
> Start date/time of requested time frame.

END-TIMESTAMP TIMESTAMP NOT NULL
> End date/time of requested time frame.

THRESHOLD TIME NOT NULL
> Requested threshold value.

START-LOGPOINT CHAR(6) FOR BIT DATA NOT NULL
> Start quiet time RBA/LRSN.

END-LOGPOINT CHAR(6) FOR BIT DATA NOT NULL
> End quiet time RBA/LRSN.

START-LP-TIMESTAMP TIMESTAMP NOT NULL
> Start date/time of START-LOGPOINT.

END-LP-TIMESTAMP TIMESTAMP NOT NULL
> End date/time of END-LOGPOINT .

The quiet times table can be queried to determine actual quiet times for any set of filters. Those filters can be determined using the GROUPID field of the two tables. There will be one too many filters associated with each GROUPID value in the GROUP table (the actual number of filters is part of the primary key for the GROUP table, the column named GROUPID-NUM).

It is important to note that you can change the threshold value for a set of filters without creating a new GROUPID. That new threshold value is reported in the quiet times table only. However, any other change to the filters results in a new (set of) GROUPID(s).

After you have created the tables the first time, anytime a quiet time report is executed, the output of that run can be saved into these tables by specifying **Save quiet times = Y** on the Quiet Time Specifications panel. The use and maintenance of these tables is otherwise determined by your site.

If you request a quiet time report but no quiet times are found, an entry is still added to the table. The entry shows the time frame requested (START/END TIMESTAMP), but all logpoint values are zeros (for START/END LOGPOINT) or low values (for the START-LP/END-LP TIMESTAMP).

Specify logs

The **Specify Logs** option on the **Generate database activity report (general)** panel is an optional way to specify the log data sets that you want Log Analysis Tool to use. The BSDS will not be used to determine appropriate Db2 log data sets needed.

Notes:

1. This is an ADVANCED feature and must be used with extreme caution. Failure to properly enter the all the needed log data sets in the proper format and order will lead to unpredictable and unsupported results.

2. This option should only be used when

   - **Data Capture Changes** is on for all filtered tables, or
   - You are filtering on only INSERTs and/or DELETEs.

The **Specify Logs** panel is scrollable and enables adding hardcoded Db2 log data sets. In addition, you can edit the generated JCL and add them here. All log data set names entered are used by both the general and details reports exclusively. The BSDS is not used to determine appropriate Db2 log data sets needed. All filters, including your specified date and time filters, are still in effect with hardcoded logs. Therefore, ensure your date and time filters correspond closely with the date and time ranges covered by the specified logs.

The named logs MUST be cataloged and active logs are not permitted. If this option is used when Data Capture Changes is not on for all filtered tables and a partial row update is processed, then additional logs will most likely need to be specified in the JCL for the execution of the Detail Report. For example, assume you need to specify the following three logs to cover the timeframe being analyzed. Also assume DSNA is data sharing member 01 and DSNB is data sharing member 02.

```
DSNA.ARCHLOG1.D16215.T1506203.A0001041
```

```
DSNB.ARCHLOG1.D16215.T1459331.A0001301
```

```
DSNB.ARCHLOG1.D16215.T1508174.A0001302
```

Your input will look similar to this:

```
V3.5.0 --------------------- Specify Logs --------------------- RS28/QC1A

Hit 'Enter' to use these logs;'PF3' to bypass specification of logs
  Row 1 of 30
. DB2 Log Dataset name                        . Member ID (00-32)
-------------------------------------------. -----------------------
DSNA.ARCHLOG1.D16215.T1506203.A0001041          01
DSNB.ARCHLOG1.D16215.T1459331.A0001301          02
DSNB.ARCHLOG1.D16215.T1508174.A0001302          02

COMMAND ===>
```

Figure 25. Specify logs panel

If your DB2 subsystem is non data sharing, then use 00 as the Member ID. A value of spaces in any log dataset field ends the list, and therefore, any data after that line is ignored both for validation and input purposes.

Continuous mode processing

Continuous mode processing is an optional way to run Log Analysis Tool general reports. In continuous mode, you run reports continuously over time, each new report beginning where the previous report ended. In this manner, you can track updates to specific objects on a daily basis.

You indicate that you want to run in continuous mode by typing name of the continuous mode file in the Continuous mode file field on the general report options panel. Prior to running Log Analysis Tool in continuous mode, you must have created the continuous mode file using the Defaults option on the main menu. For more information on creating the continuous mode file, see "Create the continuous mode data set" on page 77.

Once you have created the continuous mode file, follow these steps to run Log Analysis Tool in continuous mode:

1. On the general report option panel, specify a start date and time. The first run of the JCL uses this specified date and time.

2. Specify an end date and time. The first run of the JCL uses this specified date and time. If you do not specify an end date and time, the end date and time is set to the current date and time when the job runs. The job then processes the log continuously over time.

3. Type the name of the continuous mode file in the Continuous mode file field.

4. Set all of the other options for the general report.

5. Generate the JCL.

6. Submit the job.

This initial run of the JCL logs the start date and time for the next run in the continuous mode file. Log Analysis Tool also logs uncommitted units of recovery, so that on a subsequent run, any URIDs

with an uncommitted status will likely be resolved (committed or rolled back). Without continuous mode processing, any uncommitted records are lost if the end date and time occurs prior to a commit.

You can put the JCL in a job scheduler for execution every night. Each run will find data in the continuous mode file and use it to determine the needed start date and time as well as handling uncommitted URIDs. If you put this JCL in a job scheduler, do not include an end date or time, because each subsequent run will always end at that specified end date and time. In this case, leave end date and time fields blank, or at their default values (for example, YYYY/MM/DD).

Note: For each scheduled job you want to run in continuous mode, you must have a unique continuous mode file. For example, if you want to run a continuous mode process against two distinct sets of filters, you must create two separate continuous mode files from the **Defaults** option and generate the continuous mode JCL for each process. The continuous mode file is associated with DD statement UNCRE in the general report JCL.

⚠️ **Attention:** When using continuous mode processing, it is possible that units of work (UOW) will be in the Continuous Mode File that do not match your filters. Most of the time these UOWs are uncommitted during the specified timeframe and must be tracked in case additional log records come through that match the filter criteria the next time the job runs.

In certain circumstances, SYSLGRNX may have eliminated a Db2 log file that contained the COMMIT or ABORT record so this UOW remains in a pending state. In this case, the UOW will be tracked for an internally specified period to make sure no subsequent activity comes in. After this internally specified period, with no activity that matches the filter, this UOW will go away.

If you use MISC FLAGS=X, all the logs for the time period are read and the pending UOW will not be included in the continuous mode file – just the ones that are truly uncommitted. This does not cause any problems other than it may appear that more activity is being tracked / reported on than was requested.

Using continuous mode

Continuous mode is an option for a general report run. A panel field on the general report options determines if running in continuous mode. If in continuous mode (herein referred to as CM), the start date and time is determined from the contents of the CM file. The CM file must first be created within the "Defaults" option off the main menu; failure to create the specified continuous mode file will result in a JCL error at run time.

CM means that you want to run reports continuously over time. For example, you want to track updates to a specific object every day. You can generate the CM JCL and place it in a job scheduler for execution every night. Each run of the job will track the needed start date and time for a subsequent run. It will also track uncommitted units of recovery, so that on a subsequent run, the uncommitted status will likely be resolved (commit or rollback). Without CM, uncommitted records are lost if the end date and time occurs prior to commit.

Note: Because continuous mode runs continuously over time, some files are used as communication mechanisms from one run to the next. For restart purposes, this product now saves a backup copy of these files before reading and modifying them for the current run. The files involved are associated with the DDs UNCRE and EXTFILE. The product now saves these automatically into backup files associated with the DDs UNCREB and EXTFILB. The product does not use these backup files in anyway; it just populates them to allow for easier restart capabilities.

To start the process, generate the JCL with a start date and time as desired. The first run will use this start date and time, as the CM file will currently be empty. That initial run will then log to the CM file any uncommitted URIDs and keep track of the next needed start date and time. Each subsequent run will find data in the CM file and use it to determine the needed start date and time (as well as handling uncommitted URIDs).

The end date and time can either be omitted (spaces), entered as normal end date/time, or use a relative end date/time. If omitted, the end date/time of the CM run is the current date and time. This way, the job processes the log continuously over time. If end date/time is included or a relative end date/time is used the continuous mode will stop at the specified or calculated end date/time, and this will be used as the

next start date/time. This allows you to run continuously, but at time increments deemed appropriate by you. Using the relative end date/time specified as -060, would allow you to run each job from the time the previous job ended to 1 hour before run time. If you place this JCL in a job scheduler, the end date/time can either be omitted to always run to current or the relative end date/time can be used. However, if a specified end date/time is used, the job will never run past the specified end date/time.

After an initial run of CM, subsequent runs will actually use internally stored LRSNs as the true starting time. This is done because data in Db2's log buffers may not be externalized at the actual end time, but could have timestamps (LRSNs) that are lower than the end time. Using the last returned LRSN (by member, in a data sharing environment) ensures that this data will not be lost. Please note, therefore, that it is possible to report on log activity that is earlier than the specified start date/time when running in continuous mode. The specified start date/time is reported in the general report heading, and based only on actual dates and times, not LRSNs.

Note that for each scheduled job you want to run as CM, a unique CM file is needed. For example, if you want to run a CM process against 2 distinct set of filters (for example, 1-tablespace and plan name; 2-table name), you need to create 2 separate CM files from the "Defaults" option and generate the CM JCL for each of these. The CM file is associated with DD statement "UNCRE" in the general report JCL.

Restarting the JCL in continuous mode

All references to the term 'copy' below mean 'copy with replace'. An ISPF 3.3 copy will copy with replace, but be sure the 'Sequential Disposition = Old'. You do not want to add the records from the backup files to the originals; you want to completely replace them. IEBGENER can also be used to copy with replace.

If the "User Settings" panel field entitled "Restart assist on continuous mode" is set to Y, this means you are asking the product to track continuous mode jobs over time, and the product will automatically determine whether a job is being restarted, or if it is a new job. A restarted job occurs when the product determines the previous run of the job did not complete successfully. This type of job tracking can only be accomplished if the job JCL is static, that is, the job is not continuously recreated using the product panels. In general, this tracking is intended for JCL setup in a job scheduler, so that a third party operator can more easily restart a failed job without manual intervention. See the conditions for restart in a section below. The manual intervention necessary when this flag is set to N is described next.

If your jobs fails in continuous mode, the following rules apply:

1. If the job fails after STEP2, restart from whatever step is appropriate after STEP2. Do not restart from the beginning of the job as the continuous mode communication files have already been modified. Also, restarting from the beginning in this case is not necessary, so doing so just causes more overhead in the process. If, for whatever reason, you do restart from the beginning of the job (that is, from STEP1), you must first copy the backup files (DDs UNCREB and EXTFILB) into their associated original files (DDs UNCRE and EXTFILE). If you fail to perform this copy, your restart job will overlay the contents of these backups with the previously updated file information. This means you've permanently lost the contents of these communication files as they existed at the time of start of the originally failing run. Restart becomes more difficult in this case, so be sure to follow this copy procedure whenever restarting from the beginning.

2. If the job fails during STEP2, the copy procedure from the backups to the original files outlined above must be followed. This presumes the job has failed during program execution, and not, for example, prior to program execution by a JCL error. If you're unsure of where in the process the job has failed. look at the informational messages generated in the job log by the product. If you see this: 'ALAA156I: CMODE FILES UPDATED', then you must copy the backup files (DDs UNCREB and EXTFILB) into their associated original files (DDs UNCRE and EXTFILE). If you fail to perform this copy, your restart job will overlay the contents of these backups with the previously updated file information. This means you've permanently lost the contents of these communication files as they existed at the time of start of the originally failing run. Restart becomes more difficult in this case, so be sure to follow this copy procedure whenever restarting from the beginning.

A final note on this backup process is that GDGs may be used to ensure that you never accidentally lose the ability to restart using continuous mode. For example, if you wish to use GDGs, you can remove the UNCREB and EXTFILB DDs from STEP1, and replace these DDs in STEP2 with the needed GDGs changing

disposition from SHARE to (NEW,CATLG,DELETE). In this way, an accidental restart without first copying the backups into the originals would mean a wasted generation, but you could still go back and copy the previous generation of the backup into the original. See IBM document "DFSMS/MVS Using Data Sets" for more information on GDGs.

Conditions for restart assist in continuous mode

If the "User Settings" panel field entitled "Restart assist on continuous mode" is set to Y, this means you are asking the product to track continuous mode jobs over time, and the product will automatically determine whether a job is being restarted, or if it a new job. It does this by tracking the status of the previously executed job within the MODEFILE. So, this process can only work if the MODEFILE is not re-generated between jobs. Re-generation of the MODEFILE occurs when JCL for a job is created via the product panels. Therefore, this process is intended for JCL in a job scheduler, or any saved off JCL for repeated execution.

It is important to understand the conditions that will cause the product to assume a restart, versus a new job run. Restart is considered in effect if the previous job did not end successfully. This would include such things as an abend, or a return code that ceases job execution. For example, assume a job is run that terminates with return code=8. The next submission of this job will be considered a restart operation by the product, and it will automatically restore the backup files into their original counterparts. (This is the purpose of restart assist; it removes the manual intervention of this file restore process allowing an operator to merely resubmit the job without other external actions). Obviously, the situation causing the return code=8 must be resolved first, or it will most likely recur. But the issue here is that the job can then be restarted without further actions.

Conditions that place the job in restart mode are as follows:

• any abend, such as a S0C4, or a User abend 55, etc.
• a return code which causes job termination in an unsuccessful state. Such return codes are: return code=8 or higher, or return code=4 which is a warning, but the "User Settings" panel field "Continue processing after warnings" is set to N, which causes the job to terminate.

Conditions that place the job in successful mode (that is, no restart) are as follows:

• a return code which causes job termination in a successful state. Such return codes are: return code=0,2, or return code=4 which is a warning, but the "User Settings" panel field "Continue processing after warnings" is set to Y, which allows the job to successfully complete with warnings.

Therefore, it is important to understand that although the product tracks these states, if your required/ desired actions are not in line with these conditions, manual intervention will still be required. For example, assume you get a return code=8, but do not wish to restart that particular job. The product automatically handles the internal restart mechanism and copies the backup files into their originals, and this isn't your intention. Therefore, you would have to take some action to handle that, such as re-generating the MODEFILE and JCL via the panels. Modifying the "User Settings" field which handles restart assist will not solely handle such a case as the MODEFILE will still need to be re-generated.

It is recommended that this restart assist be used with the full understanding of its implications and conditions, as stated above. If a continuous mode job is run after failure (as defined above) or success (as defined above), and you desire to take a different path than that assumed by the product (as defined above), then some sort of manual intervention will be required depending on your actions. If you do not wish the product to track your jobs this way, simply set the "User Settings" field for restart assist to N.

Note: Restart assist in continuous mode does not depend on the existence of, or lack of, a "RESTART=STEPn" parameter on your job card. Once restart assist is set to Y, it tracks these jobs automatically. But it is still mandatory that you restart on the proper job step. In one-phase mode, a RESTART parameter would only be needed if you have additional steps in the job, for example, if LOAD options are present, because a basic one-phase mode job typically only has one (primary) step. But if running in two-phase mode where there are several job steps, you must restart from the failing step. For example, if the job fails in some step after the step where program ALAGEN1 was executed, and you restart from the step where program ALAGEN1 executes, the files will not be properly copied because the product has determined that step already completed successfully. So it is imperative that you restart

from the failing step. If you still wish to restart from the very beginning of the job for whatever reason in two-phase mode, you will have to manually do the copies yourself, as was always the case in prior releases.

Continuous mode file report

The CM files store information about activity that occurred during a run's timeframe, but was not reported because the unit of work had not completed. This information may be of interest and can be report on by entering 'C' in the general report panel's "Output flags" field. This report examines the contents of the CM file at the end of a run and provides some of the details, such as information about the unit of work, the tables involved in the unit of work, the actions performed on each table, and the number of times each action was performed.

Using miscellaneous filters

When you type Y in the Misc filters field, the Miscellaneous Filters panel displays. This panel allows entries for any combination of AUTHIDs, job names, and plan names. You can use wild cards in the list, but only a single trailing %.

```
V3.5.0 -------------------- Miscellaneous Filters --------------- SC01/SS1A

Hit 'Enter' to use these filters; 'PF3' to bypass use of these filters

*Conditional for filters..... O (A - And, O - Or)

*Filter Type    *Authid/Corr ID (jobname)/Plan Name/Conn ID/
(A/J/P/I/T/N/   Conn Type/Network ID/LU Name/Uniqueness Value/
 L/U/C/XI/XU/   Commit/XID/XUser/XWorkstation Name/XAppl Name/     Exclude?
 XW/XP/XA)      XAcctg Strng (all values case-sensitive)          (Y/N/blank)
-----------     -------------------------------------------------- -----------
    A           TS6025                                                 N
    A           TSALAUSR                                               N
    P           DSNTEP2                                                N
    P           DSNTEP4                                                N

COMMAND ===>
```

Figure 26. Miscellaneous Filters panel

The Miscellaneous Filters panel fields

Conditionals for filters

Type A if these filters should be combined with each other in an AND condition. Type O if they should be combined with each other in an OR condition. These conditionals apply only to miscellaneous filters here, not to other filters on other panels (all miscellaneous filters are combined with object type filters using the AND condition). For example, if you included an object type filter for table T1, and also want to only see when user USERJOE modified table T1, enter an authid type filter here of USERJOE. If you want to see modifications done to table T1 by USERJOE or USERSUE, set this field to O and enter two authid type filters.

Note: If you enter two or more equivalent filter types on this panel, do not set this field to A, because it cannot be matched. For example, no row update (and subsequent log record) can ever be done by both USERJOE and USERSUE, it could only be done by either USERJOE or USERSUE. The A is a conditional for unlike filter types only.

Filter Type

- Type A if the filter name is an authid
- Type J if the filter name is a job name.
- Type P if the filter name is a plan name.
- Type I if the filter name is a connection ID.
- Type T if the filter name is a connection type.

- Type N if the filter name is a network ID (part of the LUWID).
- Type L if the filter name is an LU name (part of the LUWID).
- Type U if the filter name is a unique value (part of the LUWID).
- Type C if the filter name is a commit count (part of the LUWID).

Attention: All the X-prefixed field values here relate to log records written by a participant or coordinator in a two-phase commit process. These records are not available for all two-phase distributed transactions, but if present, can be filtered and reported on. Also, because these records are only available during commit, massive amounts of data may need to be stored internally if these filters are used exclusively. To avoid this possible negative impact on both performance and job completion, always use other filters in conjunction with these X-type filters. For example, a table filter or authid filter could eliminate these potential problems if used with these X-type filters.

- Type XI if the filter name is a transaction ID (DDF record only, 50 character maximum).
- Type XU if the filter name is an end user (DDF record only, 50 character maximum).
- Type XW if the filter name is a workstation name (DDF record only, 50 character maximum).
- Type XP if the filter name is an application name (DDF record only, 50 character maximum).
- Type XA if the filter name is an accounting string (DDF record only, 50 character maximum).

Most of these filters are self-explanatory. The LUWID filters are provided primarily for the situation whereby an LUWID value is known ahead of time, and you wish to filter on that specific LUWID (the LUWID might be known by means of a Db2 PM report, for example).

An LUWID is made up of four components in this order:

1. A network ID (eight characters),
2. An LU name (eight characters),
3. A uniqueness value (six bytes binary),
4. A commit count (two bytes binary),

for a total of 24 characters. However, it is not necessary to enter all four components. It is most likely that all the network ID and LU name values associated with all LUWIDs on a specific subsystem will be identical. Therefore, entering just the unique value will likely get your desired results. Do not use wild cards in the Uniq or Commit names because these must be the exact hexadecimal values desired.

The X-prefixed filters represent filters for DDF (distributed) type log records only. Use these with caution, as this information is not available on non-distributed log activity, and each unit of work will have to be stored internally until such log record information is available for filter checking (X filters are generally not known until commit processing begins, so the unit of work log records must all be saved until the end of the unit of work). It may also be that the partner in the distributed unit of work is not providing this information (XID, XUSER, and so on), so these fields may all be empty. There are no predetermined set of possible values for these filters, they are provided for by the distributed application. For example, a CICS® application may (or may not) provide values such as end user or workstation name. Each value can be up to 50 characters long (This is the limit imposed by Log Analysis Tool. Future use may mandate longer maximum values because the theoretical values can be up to 255 bytes each).

Authid/Corr ID (job name)/Plan name/Conn ID/Conn Type/Network ID/LU name/Uniqueness value/ Commit/XID/XUser/XWorkstation Name/XAppl Name/XAcctg Strng
Enter the value as specified by the filter type setting. Db2 wildcards (that is, %, _) may be used in all specifications except the LUWID filters (as these would be primarily numeric in nature).

Most of these filters are a maximum of eight characters long, the exceptions are:

- Uniqueness value = 12 character uniqueness value associated with a known LUWID.
- Commit value = 4 character commit count associated with a known LUWID.

Both of these exceptions require input of exactly the lengths specified, and must be the actual hexadecimal values within the LUWID (that is, do not enter decimal equivalents of these hexadecimal

numbers). These values are passed to the batch job as actual binary values, so your generated JCL will have non-displayable values in it (if you do not save the filters to a file). For example, a valid Uniqueness value would be B91AAAC04523, and a valid Commit value would be 001A.

Exclude

Type Y to exclude the named filter. Type N or leave blank to indicate the filter should be included.

The Miscellaneous Filters panel commands

The Miscellaneous Filters panel in Log Analysis Tool requires you to enter such information as plan names and AUTHIDs. You can look up values for filters without having to leave Log Analysis by using these commands.

PLANS command

On the Miscellaneous Filters panel, enter the command "PLANS";. This displays the Plan list panel that can be used to find all the plans for the specified subsystem. You can filter the plans by name to assist with finding a specific plan. You can select multiple plans to add to the miscellaneous filter list without ever manually typing a plan name. While converting miscellaneous filters from the previous version of the ISPF, the filters might be displayed in a wrong position. To correct this, clear the panel and reenter filters.

```
V3.5.0 ------------------------ Plan list --------------------- SC01/SS1A

 Hit 'Enter' to save the selected authids; 'PF3' to exit and use saved authids;
 Type 'Cancel' to bypass use of all saved authids.

 Line Commands:  S - Select

 Conditional for filters..... O (A - And, O - Or)
*Exclude plans............... Y (Y/N)

 Plan Like ALA
                                  Row 1 of 275
 -----------------------------------------------------------------------------
    Plan Name
    ---------
    ALA31GF
    ALA31PV
    ALA32JT
    ALAV32K
    ALAV32R
    ALA33GF
    ALAV33GF

 Command ===>                                           Scroll ===> PAGE
```

Figure 27. Plan list panel

AUTHIDS command

On the Miscellaneous Filters panel, enter the command "AUTHIDS". This displays the Authid list panel that can be used to find AUTHIDs for the specified subsystem. You can filter the AUTHIDs by name to assist with finding a specific AUTHID. You can select multiple AUTHIDs to add to the miscellaneous filter list without ever manually typing an AUTHID.

```
V3.5.0  ---------------------- Authid list ------------------- SC01/SS1A

 Hit 'Enter' to save the selected authids; 'PF3' to exit and use saved authids;
 Type 'Cancel' to bypass use of all saved authids.

 Line Commands:  S - Select

  Conditional for filters..... O (A - And, O - Or)
 *Exclude authids............. N (Y/N)

  Authid Like PD
                                             Row 1 of 346
 -------------------------------------------------------------------------------
    Authid Name
    -----------
    PD17
    PDAJET
    PDCJOB
    PDCJK
    PDCJ5
    PDCLP
    PDFTRT
    PDSTT
 Command ===>                                            Scroll ===> PAGE
```

Figure 28. Authid list panel

Using object filters

There are three different types of object filters that you can use. You can filter objects by name, by ID, or by using advanced filters.

Filtering objects by name

When you type M in the **Object filters** field, the Object Name Filters panel displays. This panel allows you to specify a maximum of 15 fully qualified tables or table spaces to use as filters in the general report run.

```
V3.5.0 ------------------- Object Name Filters ------------------ SC01/SS1A

 Hit 'Enter' to use these filters; 'PF3' to bypass use of these filters

 *Object   *Table Creator/       *Table Name/                  Part   Exc?
 Type(T/S) Database Name          Tablespace Name               Number (Y/N)
 --------  -------------------   -----------------------------  ------ ------



 COMMAND ===>
```

Figure 29. Object Name Filters panel

Note: : If you require more than 15 object names, you can edit the JCL to add more. See the online tutorial section on "Filter file handling" for more information on editing your filters outside of the ISPF front-end. Do not attempt to create a filter file from scratch. You must use the panel to load up as much data as possible and edit the file only after it has been saved

Object type
    Type T to indicate the named object as a table, otherwise type S to indicate the object is a table space.

Table Creator/Database, Table Name/Tablespace
    Type a table creator and name if the Object Type is T. Type a database name and table space name if the Object Type is S. Blanks are not allowed in these fields if the Object Type is non-blank. Db2 wildcards (that is, %, _) may be used in all specifications.

    Note: If you wish to filter on a table, do not also enter the database and table space that contains that table as a filter. For example, assume table T1 exists in database/table space DB1.TS1. If you want to filter only on T1, include a single entry of type = T and T1. Do not also add an entry for type S on DB1.TS1. Doing so would include all tables in DB1.TS1, which is likely not the desired effect. In other words, if you need table filters, include entries for them as type T entries, but there is no need to also include their containing database/table space (and doing so would likely yield undesired results).

Part number

This field identifies a one-based partition number, if so desired. Enter any number from 1 to 4096 if partition-level filtering is desired. You can leave it blank when no partition filter is desired (or the table is not partitioned). Partition filters are not allowed whenever a LOAD REPLACE utility-type log action is found. If there is any chance of such a log record being encountered during your specified time range, do not include a partition filter as an error message will be generated.

Note: If it is likely that rows move among your partitions, *never* filter by partition if your goal is to REDO or UNDO SQL. Rows can move from one partition to another if, for example, the partition key columns are updated, forcing the row from one partition to another. Filtering out a partition involved in such an action will omit the related row activity within that partition, and therefore any executed REDO/UNDO SQL will not be able to properly update that omitted partition.

Exc?

(Exclude) Type Y to exclude the named object. Type N (or leave blank) to include the named object.

Using Db2 long names

About this task

You can use much longer names for the table owner and the table name (up to 128 characters) than the product panels can accommodate. Should you have a table name or owner greater than the field size on a product panel:

Procedure

1. Type as many characters as possible to fill the field on the panel.
2. Press Enter.

   A pop-up window appears that allows you to complete the long name.

```
-------------------- Table Owner/Name Extender --------------
|                                                            |
|   If needed, extend field up to 128 characters below:      |
|   PRODDEVELOPMENTUSERID64583235                            |
|                                                            |
|                                                            |
-------------------------------------------------------------
```
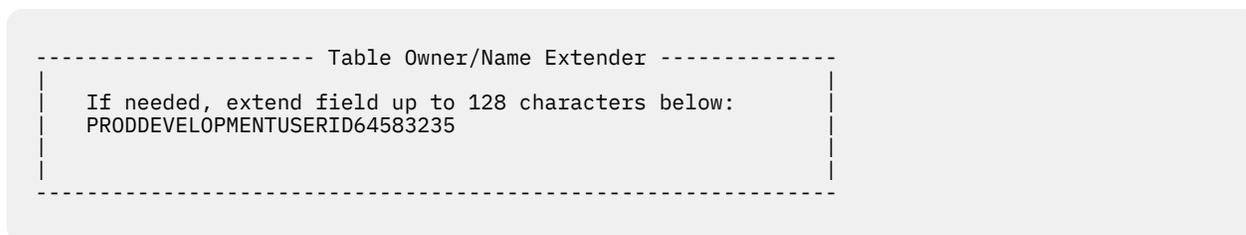
Figure 30. Table Owner/Name Extender panel

3. Finish typing the full name in the window.
4. Press Enter to return to the originating panel.

Object Name Filters panel commands

The Object Name Filters panel in Log Analysis Tool requires you to enter such information as table and table space names. You can look up values for these filters without having to leave Log Analysis Tool by using the following commands.

TABLES command

On the Object Name Filters panel, enter the command "TABLES". This displays the Table list panel that can be used to find all the tables for the specified subsystem. You can filter the tables database, table space, table name, and table owner to assist with finding a specific table. You can select multiple tables to add to the object name filter list without ever manually typing a table name.

```
V3.5.0 ---------------------- Table list -------------------- SC01/SS1A

 Hit 'Enter' to save the selected tables; 'PF3' to exit and use saved tables;
 Type 'Cancel' to bypass use of all saved tables.

 Line Commands:  S - Select  O - View table owner  N - View table name

 Database Like    %          Table Space Like %
 Table Owner Like PDUSER
 Table Name Like  %

                                  Row 1 of 189
 -------------------------------------------------------------------------------
    Table Owner                   Table Name                    Part  Exc?
 -  ----------------------------  ----------------------------  ----  ----
    PDUSER                        PMRTEST1
    PDUSER                        PMRTEST2
    PDUSER                        PMRTEST3
    PDUSER                        PMRTEST4
    PDUSER                        PMRTST1A
    PDUSER                        PMRTST1B
    PDUSER                        PMRTST1C
    PDUSER                        PMRTST1D
 Command ===>                                           Scroll ===> PAGE
```

Figure 31. Table list panel

TABLESPACES command

On the Object Name Filters panel, enter the command "TABLESPACES". This displays the Tablespace list panel that can be used to find table spaces for the specified subsystem. The table spaces can be filtered by database and table space name to assist with finding a specific table space. You can select multiple table spaces to add to the object name filter list without ever manually typing a table space name.

```
V3.5.0  --------------------- Tablespace list ------------------- SC01/SS1A

 Hit 'Enter' to save the selected table spaces; 'PF3' to exit and use saved
 table spaces; Type 'Cancel' to bypass use of all saved table spaces.

 Line Commands:  S - Select

 Database Like ABT       Table Space Like ABT

                          Row 1 of 214
 -------------------------------------------------------------------------------
    Database  Table Space  Part  Exc?
 -  --------  -----------  ----  ----
    ABT1556D  ABT1556S
    ABT1566D  ABT1566S
    ABT1568D  ABT1568S
    ABT1573D  ABT1573S
    ABT1574D  ABT1574S
    ABT1575D  ABT1575S
    ABT1582D  ABT1582S
    ABT1644D  ABT1644S
    ABT1659D  ABT1659S
    ABT1768D  ABT1768S
 Command ===>                                           Scroll ===> PAGE
```

Figure 32. Tablespace list panel

Filtering objects by ID

When you type I in the **Object filters** field, the Object IDs Filter panel displays.

```
V3.5.0 ---------------------- Object IDs Filter ------------------ RS22/DC1A
Command ===>

Hit 'Enter' to save current filters; 'PF3' to exit and use saved filters;
Type 'Cancel' to bypass use of all saved filters; Type 'Skip' to clear
the current screen.

 (Current or New)      Pages           (Optional Old)
*DBID   *PSID  *OBID    (Y/N)          DBID    PSID   OBID
------  -----  -----    -----          ------  -----  -----


  COMMAND ===>
```

Figure 33. Object IDs Filter panel

This panel allows for a maximum of 14 tables or table spaces represented by their internal Db2 identifiers. If you need more, follow the format as described in the tutorial in *Filter File Handling*. The numeric identifiers for PSID and OBID must be in decimal notation, and cannot exceed 5 digits. To allow a specification of negative DBID values, DBID contains 6 digits. The DBID and PSID must always be provided. If you want to filter the object at a table level, also supply an OBID. If old IDs are included, the OBID must be supplied.

Object IDs Filter panel fields

(Current® or New) DBID/PSID/OBID
> Enter up to 5 numeric decimal digits (or 6, for a negative DBID value), which represent the Db2 internal identifiers for the database, table space, and (optionally) table. If old IDs are included on the line, you must include the table (OBID).

Pages
> Type Y to request one or more page filters for the identified object. Type N or spaces to bypass page level filters. If requested, after all validation is performed for the current panel, another panel will be displayed for all page level filters. Page level filters are not used in dropped object mode, so they only apply to the current/new object IDs specified. Page level filters are not allowed when LOAD REPLACE type log records are found during the general report run (page level filters cannot be used in either the general or detail report runs in such cases). If there is any chance such a log record could be encountered during your time range, do not use page level filters as an error message will be generated. For more information on page filters, see "Using page filters" on page "Using page filters" on page 124.

(Optional Old) DBID/PSID/OBID
> Enter up to 5 numeric decimal digits (6 for a negative DBID) which represent the Db2 internal identifiers for the database, table space, and table as they existed prior to being dropped. If these fields are entered, you are considered to be running in dropped object mode. Special processing is required for dropped object mode, so do not include values here unless you specifically want to track activity against previously dropped objects.

This panel allows you to run in two different modes:

- Standard mode – If you specify only current or new object IDs, you are filtering against objects that exist now and existed at the timeframe requested in the general report. You are considered in standard mode if old IDs are not present.

- Dropped object mode – If you specify the optional old IDs, this is considered dropped object mode. In this mode, you are requesting information from the log about an object that has been dropped and subsequently restored. The log data obtained relates only to the old object IDs. You initiate dropped object mode by including old object IDs in an Object IDs Filter panel.

To use dropped object mode, these things must be true:

- The dropped object must have been restored (redefined to Db2 and recovered to an old image copy).
- A new full image copy must be taken of that restored object.

The new full image copy must come from the COPY utility only, not from a LOAD or REORG. These prerequisites are necessary because Db2 catalog information must be available on both the restored object's attributes (SYSCOLUMNS data) and its history (SYSCOPY data). If you deviate from these prerequisites, the results are unpredictable and unsupported.

Note: If a detail report is subsequently run after a general report that ran in dropped object mode, only a log-forward mode is acceptable therein. Though not forced, it is most probable that only redo-type SQL would be desired.

When running in dropped object mode, all IDs must be present. That is, the translation of old to new IDs requires that DBID, PSID, and OBID are all specified (OBID is not optional in dropped object mode). It is most likely that you specify only a single DBID/PSID group for the run as the start date and time will likely need to reflect a single image copy timestamp for that dropped object (table space/table).

Using page filters

If you specify Y in the Page field on the Object IDs Filter panel, the Page Filters panel displays:

```
V3.5.0 ------------------------- Page Filters ------------------ SC01/SS1A

Hit 'Enter' to use the list; 'PF3' to bypass using the list

 Enter pages below for DBID/PSID/OBID:  57673,76466,35767


  ------------------------------------------------------------------------------
  ------------------------------------------------------------------------------
  ------------------------------------------------------------------------------
  ------------------------------------------------------------------------------
  ------------------------------------------------------------------------------


 COMMAND ===>
```

Figure 34. Page Filters panel

Page level filters should only be used by highly knowledgeable Db2 personnel. Each panel displayed for page level filters will save all the pages and page ranges you request for the identified object. Pages are only relevant at the table space level, but if an OBID is provided, the log record must match on both the fully qualified object and page. These pages you enter are not saved in ISPF variables, so if you need to enter many values and wish to reuse them, request saving all the filters to a file prior to specifying them on this panel. See "Filter file usage" on general/detail report panels.

Typically, you would only use this option after identifying certain pages of interest from other sources. Without such, it is unlikely page level filters would be appropriate. Again, this option is intended for advanced users only.

The rules for page level filter input are as follows:

- Eight-character hexadecimal display format MUST be used for all pages.
- This format must be a valid Db2 page including partition values, if any. If you are unsure of how partition numbers are stored in a Db2 page number, it is strongly recommended that you do not use this option. That topic is outside the scope of this product.
- This product reports only on DML, so header pages, space map pages, and dictionary pages will not be reported on.
- Specific pages can specified, such 0000020B.
- Pages can specified as ranges, such 00000005-0000001A, which is inclusive.
- All pages and page ranges must be comma separated.
- Do not include any blanks or other values between pages unless shown here.
- All pages and page ranges must be fully contained within a single line. Pages and page ranges cannot span lines.

- You do not have to blank out the remainder of any unused lines or a portion of a line. Leaving the input field as is indicates end of input.

Page specification examples

You want to request pages 00000015, 00000025, and range 00000034-00000076. Your input looks like this:

```
00000015,00000025,00000034-00000076
```

You want to request pages 00100022,00100032, and ranges 00100034-00100036 and 001001BF-0010020C and 00100222-00100244 on 2 input lines. Your input looks like this:

```
00100022,00100032,00100034-00000036,001001BF-0010020C
```

```
00100222-00100244
```

Using advanced object filters

When you type A in the **Object filters** field, the Advanced filters panel displays. Advanced filters allow specification of filtering down to the column level.

Note: You must also have specified Y on the Generate details field to use advanced filters. This is because advanced filters are primarily applied at the details report level.

```
V3.5.0 ---------------------- Advanced filters ----------------- SC01/SS1A

Hit 'Enter' to save current filters; 'PF3' to exit and use saved filters;
Type 'Cancel' to bypass use of all saved filters

*Table Owner          *Table Name
-------------------    -----------------------------


Column Name                    Op  Column Value (case-sensitive)     Cond
----------------------------- --  ---------------------------------- ----



COMMAND ===>
```

Figure 35. Advanced Filters panel

Advanced filters panel fields

Table Owner
  Enter the table owner (creator) of the desired table. Enter as much of the value as necessary until the end of the field. If the table owner is longer than the allotted space, see "Using Db2 long names" on page 121 for information on how another panel displays so you can complete the value.

Table Name
  Enter the table name of the desired table. Enter as much of the value as necessary until the end of the field. If the table name is longer than the allotted space, see "Using Db2 long names" on page 121 for information on how another panel displays so you can complete the value.

Column Name
  Enter the name of the column to be filtered on. The action taken for this column filter depends on the value of the Op field. The first occurrence of spaces in any column name field indicates the end of data, and no further checking, validation, or saving of entries beyond that line on the panel is done. This allows for quick exclusion of unneeded entries already entered, if you want to do so.

Op
  Enter the operator. The operator determines the type of action to be taken against the column name and value. Valid operators are:

EQ
   equal
NE
   not equal
LT
   less than
LE
   less than or equal
GT
   greater than
GE
   greater than or equal
LK
   LIKE (value must have single trailing percent sign)
NL
   not LIKE (value must have single trailing percent sign)
CH
   column value has changed
NC
   column value has not changed
IC
   include the column
XC
   exclude the column

IC and XC:

IC and XC mean include or exclude this column from the detail report, any SQL, and any LOAD processes. This operator also ignores column values, if specified.

IC and XC are mutually exclusive at the table level. If one IC operator is found, you must explicitly include all desired columns. Any other column not listed will be excluded. For every XC operator found, that column will be excluded from the result set. These operators can be useful in examining only a subset of a table's columns. Be aware that exclusion of any columns can result in an UPDATE action with no apparent column update (that is, an excluded column was the only column updated).

Because the IC and XC operators result in only a subset of column data being reported on, be very careful if generating SQL with this option. It is actually most likely you would bypass SQL generation using these operators. For example, if you omitted any of the key columns for a table, unpredictable results would occur if running any SQL generated since the WHERE clause would not properly reflect full key values.

Column Value
   Enter the value against which the column data will be compared. This field may be irrelevant depending on the value of the "Op" field. If relevant, the value to be compared against will come from the column data value for either the pre- or post-row image (never the current image, if present). For example, it is considered a match on an EQ condition if the column value of either the pre- or post-row image matches on the supplied filter value here. When you need to match a column based upon a hexadecimal value, you can enter the value using the pattern X'HH..HH' (where H is a valid hexadecimal number).

   If the operator field is LK or NL, indicating some form of LIKE condition, only a single trailing percent sign (%) is allowed in the value. If omitted, the first blank will be considered the wildcard. If multiple percent signs are found, only the first is used and the rest of the data is considered all wild cards. Use single quotes only around character type data if and only the string contains embedded spaces, otherwise do not use quotes.

Cond

Enter the conditional value, if necessary. Conditionals are only necessary if multiple columns are present. Conditional values are A (AND), O (OR), with any set of parentheses as needed. (See the tutorial on this topic for complete description of these conditionals). If this field is left blank, it terminates all checking on subsequent lines of input and is considered an end-of-data marker. Therefore, do not leave this field blank if subsequent lines of input are needed and non-blank. When the Op field is either IC or XC, this field should always be set to O, unless it is the last line of input, in which case it can be left blank.

You can use column data filters when you want to limit your report size and the SQL that is generated to specific column data values. Try to enter values as close as possible to the format they would be shown as in the reports. For example, a DEC(7,2) field may look like '+00250.03' in the report, so using the same exact value would ensure a match.

Log Analysis Tool attempts further matching logic on such numeric fields if not exactly identical. For example, 250.03 will match, but in cases where decimal points are present, you must supply the full scale value or results are unpredictable. In this case of DEC(7,2), 100 is an invalid comparison value, but 100.00 is valid.

If any column data does not satisfy these conditions, it will be omitted from both the detail report and SQL generation (if so requested).

The Advanced filters panel commands

The Advanced filters panel in Log Analysis Tool requires you to enter such information as column names. You can look up values for this filter without having to leave Log Analysis Tool by using the following command.

COLUMNS command

On the Advanced filters panel, enter a table name and then type the command "COLUMNS". This displays every column name for the specified table and allows you to set filters for each column. When the "COLUMNS" command is used, the column names do not necessarily have to be typed in manually.

```
V3.5.0 ------------------------ Column list -------------------- SC01/SS1A

 Hit 'Enter' to save current columns; 'PF3' to exit and use saved columns;
Type 'Cancel' to bypass use of all saved columns for this table.

Table Owner            Table Name
------------------     -----------------------------
PDUSER                 PMRTEST1
                          Row 1 of 12
--------------------------------------------------------------------------------
Column Name                    Op  Column Value (case-sensitive)      Cond
---------------------------- --  --------------------------------  ----
  NAME1
  NAME10
  NAME11
  NAME12
  NAME2
  NAME3
  NAME4
  NAME5
  NAME6
  NAME7
  NAME8
  NAME9



  ****************************** Bottom of Data ******************************
  Command ===>                                          Scroll ===> PAGE
```

Figure 36. Column list panel

Column data filter examples

Listed here are three examples of the column data filters in the created JCL.

Example 1

The first example shows a single table, single column filter.

```
//FILTERS       DD   *
TABLE OWNER=JOHNQ
TABLE NAME=TABLEX
COLUMN NAME=GROUPNAME
OPERATOR=EQ
VALUE=WALDO GROUP
COND=
/*
```

Example 2

The second example adds another table to Example 1.

```
//FILTERS       DD   *
TABLE OWNER=JOHNQ
TABLE NAME=TABLEX
COLUMN NAME=GROUPNAME
OPERATOR=EQ
VALUE=WALDO GROUP
COND=
TABLE OWNER=BOBX
TABLE NAME=TABLEZ
COLUMN NAME=DEPT
OPERATOR=EQ
VALUE=100
COND=
/*
```

Example 3

This example shows how to use the COND field for a single table. In this example, ((GROUP_NAME = 'WALDO GROUP' AND CITY = 'NEW YORK') OR CITY = 'BOSTON').

```
//FILTERS  DD *
TABLE OWNER=JOSH1
TABLE NAME =GROUP_LOCATIONS
COLUMN NAME=GROUP_NAME
OPERATOR    =EQ
VALUE       =WALDO GROUP
COND        =(A
COLUMN NAME=CITY
OPERATOR    =EQ
VALUE       =NEW YORK
COND        =)O
COLUMN NAME=CITY
OPERATOR    =EQ
VALUE       =BOSTON
COND        =
```

Running Details Reports

Once you have run and analyzed a general report, you may determine that it is necessary to see the actual data changes that have occurred. The detail report uses the output from the general report, as well as any other filtering you select, to generate a report showing the actual data changes that occurred. Optionally, you can also generate SQL to undo or redo the data changes.

Important: If running in one-phase mode, the detail report cannot be run separately from the general report. You cannot decide to run a detail report after running a general report in one-phase mode; you either run only a general report, or both reports together in one-phase mode.

To run a Details Report

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **2**, **Generate database activity report (details)**.

   The Generate Database Activity Report (details) panel appears.

```
V3.5.0 ---------- Generate database activity report (details) ---- SC01/SS1A
COMMAND.....
                                                            More:     +

*Action............... E        (E - Edit,S - Submit)
 Job Identifier........          (8 character job identifier)
*Log reading mode...... F        (A - Auto, B - Backward, F - Forward,
                                  C - Combo, D - Auto default backward)
*Commit scope.......... 000      (000-999, UOW)
*Generate Undo SQL..... N        (Y - Yes, N - No,
                                  X - Yes, exclude triggered actions)
*Generate Redo SQL..... N        (Y - Yes, N - No,
                                  X - Yes, exclude triggered actions)
 LOB PDS name..........
 Set Current SQLID.....
*Archive output........ N        (Y - Yes, N - No,
                                  S - Specify archive options)
*Expert options........ N        (Y/N)
*Current row by key.... N        (Y - Yes, N - No,
                                  X - Exclude current rows)
*WHERE clause by key... Y        (Y/N)
*Output flags.......... D        (B - Bypass reports, D - Detail,
                                  T - Transaction, I - Impact,
                                  C - Catalog Audit, S - Summary)
```

Figure 37. Generate database activity report (details) panel (top half)

```
V3.5.0 ---------- Generate database activity report (details) ---- SC01/SS1A
COMMAND.....
                                                            More:     +

*Action............... E        (E - Edit,S - Submit)
 Job Identifier........          (8 character job identifier)
*Log reading mode...... F        (A - Auto, B - Backward, F - Forward,
                                  C - Combo, D - Auto default backward)
*Commit scope.......... 000      (000-999, UOW)
*Generate Undo SQL..... N        (Y - Yes, N - No,
                                  X - Yes, exclude triggered actions)
*Generate Redo SQL..... N        (Y - Yes, N - No,
                                  X - Yes, exclude triggered actions)
 LOB PDS name..........
 Set Current SQLID.....
*Archive output........ N        (Y - Yes, N - No,
                                  S - Specify archive options)
*Expert options........ N        (Y/N)
*Current row by key.... N        (Y - Yes, N - No,
                                  X - Exclude current rows)
*WHERE clause by key... Y        (Y/N)
*Output flags.......... D        (B - Bypass reports, D - Detail,
                                  T - Transaction, I - Impact,
                                  C - Catalog Audit, S - Summary)
*Report row length..... 32000    (00080-32752)
*Treat mass deletes.... I        I - As individual delete statement,
                                  O - As one mass delete statement)
*CHAR data in hex...... N        (Y - yes, N - No)
*Show varchar length... N        (Y - yes, N - No)
 Only changed columns.. N        (Y - yes, N - No)
```

Figure 38. Generate database activity report (details) panel (bottom half)

2. Enter the required information and press **Enter**.

   Log Analysis Tool runs the Details Report.

   Generate database activity report (details) panel fields:

Action

Type E to edit the generated JCL, or S to submit the JCL directly to generate the report.

Job Identifier

(Optional) Type an identifier for a particular job run (eight character maximum). This ID is used in the creation of data set names used in the batch JCL to generate a Details Report, for example, if set to TEST01, the REFILE will be named USERID.REFILE.TEST01. By utilizing this field, you can run multiple Details Reports without losing data from previous runs.

Additionally, this field correlates data sets used in General Reports with data sets used in Details Reports. If this field is left blank, each run of a Details Report replaces the previous report's data.

Log reading mode

This field determines how the detail report will be generated.

A

Log Analysis Tool automatically determines the best way to generate the detail report. Use this setting if you are unsure of how best to proceed, or if you want the product to make the determination for you. This should be the normal setting.

If you type A, and the product determines log-backward mode, it is still possible to get warnings due to message ALAA061W (log-RID mismatches). If you set the Defaults panel field **Continue processing after warnings** to N, the product will attempt to switch automatically to log-forward mode. If this is not the behavior you want, either change the Defaults panel field, or use the F or B selections for log processing mode.

B

Force a log-backward process (LB).

F

Force a log-forward process (LF).

C

Combo mode processing. Use a combination of log-forward and log-backward processes, if possible.

This option is a request to run in combo mode; it does not guarantee it will do so. Log Analysis Tool determines if some objects can be run in log-backward mode and some in log-forward mode during the detail report run. If that is true, then and only then will combo mode actually be chosen. If the product finds all objects can run in log-backward mode, it will still chose log-backward mode. If it determines all objects must run in log-forward mode, it will choose log-forward mode. It is only when a combination of both modes is possible that this method will be selected by the product, and only if you originally selected combo mode.

For example, assume two table spaces are in the result set after the general report component runs. If only one table space has done a subsequent REORG, that table space must run in log-forward mode; the other table space can likely run in log-backward mode. This now means only one of the table spaces' image copies needs to be accessed. Performance may (or may not) be positively affected when running in combo mode. It is impossible to tell ahead of time. Though image copy access can be avoided for some objects, additional Db2 logs may also be read. Therefore, combining both modes in one run may lead to extra Db2 log access which may not have occurred had only one of the modes (backward or forward) been chosen. The product cannot determine if this potential Db2 log access increase is offset by image copy avoidance. You must determine if this option provides you with a benefit.

Restriction:

- Fast path mode is mutually exclusive with Combo mode because FP does not materialize full row images.
- Hi speed mode, if selected by you AND the product, effectively ignores Combo mode for same reason (that is, full rows are not materialized as hi speed mode requires all rows to be full initially).
- Combo mode is not allowed with expert options

- The option **Specify logs** means LF only, a combination of modes is not possible (you can request it, but it will not be used).
- Dropped object mode must run in LF mode (you can request it, but it will not be used)
- Work file mode not allowed with Combo mode.

D

Auto default backward. Log Analysis Tool determines if log-backward mode is possible, that is, no reasons exist that force log-forward mode. If that is true, then Log Analysis Tool will choose log-backward mode.

Performance may or may not be positively affected when running in Auto Default log-backwards mode. With this option, if log-backward mode is possible, then it will be used regardless of how many log records must be applied to the current row image to get back to the analysis time frame.

For more information on log processing, see the tutorial and Chapter 16, "Log-forward versus log-backward processing," on page 233.

Commit Scope

There are two distinctly different types of values that you can enter for Commit scope:

000–999

A three-digit number from 000 to 999 indicating the number of SQL statements to execute prior to issuing a COMMIT statement. There is also an implicit COMMIT at the termination of the SQL.

If set to 000, no COMMIT statements are added to the SQL flat file. This value is ignored if no SQL flat file is requested (both Generate undo SQL = N and Generate redo SQL = N).

Note: Even though no COMMIT statements are added to the SQL file, an implicit COMMIT occurs at the termination of the SQL.

UOW

This feature allows the product-generated SQL to be committed and executed in unit-of-work order. The SQL is grouped by end unit of work time and sorted by the RBA or LRSN within the end unit of work. Additionally, the commits are issued as soon as the unit of work is finished. This is in contrast to the preceding method of committing by the number of SQL statements executed only.

For example, in Log Analysis Tool V3.2, if the updates are done similar to the sequence shown in Example 1, then the resultant SQL file will be created, as shown in Example 2 (this presumes a REDO operation). The term 'transaction' refers to a unit of work.

```
Example 1: Transaction sequence in DB2:
 Transaction A updates record X
 Transaction B updates record Y
 Transaction A updates record Z
 Transaction A ends (end unit of work)
 Transaction B updates record X
 Transaction B updates record W
 Transaction B ends (end unit of work)
```

```
Example 2: Resultant generated REDO SQL statements
   by Log Analysis Tool:
 Transaction A updates record X
 Transaction A updates record Z
 COMMIT A
 Transaction B updates record Y
 Transaction B updates record X
 Transaction B updates record W
 COMMIT B
```

Restriction: If you specify UOW for commit scope during an archive run (option #6 off the main menu), the timeframe must not include older archives that were created in releases prior to V3.2. This option is not available on archives created in prior releases, as those archives

would not contain the necessary information. Only release V3.2 (and subsequent releases) will handle this option within the context of archiving.

Generate Undo SQL

Type Y to generate undo SQL as the final step in the Details Report process. This creates a flat file holding SQL that can be used to undo the log modifications reported on in the Details Report. The generated SQL exists in the SQLOUT DD after the job successfully runs.

Type N if you do not want the undo SQL step added to the jobstream.

Type X if UNDO SQL is not needed to back out actions resulting from triggers firing.

This field is mutually-exclusive with generating redo SQL.

Generate Redo SQL

Type Y to generate a flat file containing SQL to redo the changes reported on. The generated SQL exists in the SQLOUT DD after the job successfully runs.

Type N if you do not want the redo SQL step added to the jobstream.

Type X if REDO SQL is not needed for actions that originally resulted from triggers firing.

This field is mutually-exclusive with generating undo SQL .

LOB PDS name

See "LOB PDS name" in "Generate details" on page 98.

Set Current SQLID

See "Set Current SQLID" in "Generate details" on page 98.

Archive output

This field allows you to archive (save) the results of your detail report run. Type N to bypass this feature. Type Y to save your results to a specified data set. Type S to specify that data set, and other required information. Before you can use the Y setting, you must use the S setting at least once to specify this required information. See the tutorial for more information on archiving output.

Expert options

Type Y to display another panel with expert option settings. Type N to bypass expert option settings. Expert options are appropriately named and should only be used in very exceptional cases by experts of Db2 and the local Db2 data and design.

Current row by key

See "Current row by key" in "Generate details" on page 98.

WHERE clause by key

See "WHERE clause by key" in "Generate details" on page 98.

Output flags

See "Output flags" in "Generate details" on page 98.

Report row length

See "Report row length" in "Generate details" on page 98.

Treat mass deletes

See "Treat mass deletes" in "Generate details" on page 98.

CHAR data in hex

See "CHAR data in hex" in "Generate details" on page 98.

Show varchar length

See "Show varchar length" in "Generate details" on page 98.

Only changed columns
> See "Only changed columns" in "Generate details" on page 98.

Bypass all filters
> Type Y to process all the data that is included in the output of the general report. This means all the other filters defined on this panel are ignored. This will likely be your desired setting, and therefore this flag is always initialized to **Y**, regardless of your past setting. This behavior is forced to avoid the overhead of unneeded filtering at the detail report level. If you wish to further narrow down the data contained in the general report output, type N, and the filters on this panel will be in effect for the detail report. If you set this field to **N** and do not modify your filters from the general report (a subset of the filters only is allowed, not a superset), then you are incurring added overhead that will have no effect.

Show UPDATEs, INSERTs, DELETEs
> Type Y in any of these fields to show the associated action (any type of UPDATE, INSERT (by LOAD or SQL), or DELETE) in the Details Report, or N if you do not want a report on the associated action. INSERTs and DELETEs also allow you to type X to allow for unique exclusion situations:

> **INSERTs**
>> X means exclude only INSERTs done via LOAD (that is, by utility, not by SQL). INSERTs done by a LOAD utility (LOAD/LOAD REPLACE) can be very expensive to process, and using this setting can alleviate that potential cost without affecting your need to see SQL INSERTs.

> **DELETEs**
>> X means exclude only DELETEs done via mass delete.

Show uncommitted
> See "Show uncommitted" in "How to create a General Report" on page 85.

URID
> Set this to a URID as found in the general report. The format must be either blank, to ignore this option, or a 12-character hexadecimal number. For example, 000012AA0567 is valid but 12AA0567 is not valid. If specified, only this URID will be processed.

Object filters
> See "Using object filters" on page 120.

Misc filters
> See "Using miscellaneous filters" on page 117.

Filter file usage
> See "Filter file usage" in "How to create a General Report" on page 85.

Filter file name
> See "Filter file name" in "How to create a General Report" on page 85.

Expert Options

Expert options are meant for Db2 experts only who require very specialized control of the detail report run. This option has no effect on the general report process and uses that output as input to the detail report, just as in normal processing modes. If any of the terminology used here is unfamiliar to you, do not use this option.

Important: The use of this option can cause serious problems, abends, and incorrect output. The results of specifying any invalid input is unpredictable and unsupported. Therefore, only those with complete knowledge of Db2 workings and their specific data should use this option.

Overview

This option allows you to specify resources rather than have the product determine them. The term "resources" in this instance means two things:

1. Where starting full row images are resolved from, and
2. What type of log-reading mode will follow

Normally, Log Analysis Tool determines where starting full row images are resolved from, and this is determined in conjunction with the log-reading mode.

- If image copies are used as the resource for starting full row images, log-forward is the log-reading mode.
- If current table spaces are used, log-backward is the log-reading mode.

The expert options allow you to specify and mix both. Because you are in complete control of these input values, Log Analysis Tool does not do any checking to determine the validity of this input. But all of the normal constraints still apply.

For example, if you use a current table space as the input source, and the general report time frame occurred before a recent REORG was done, that current table space is truly invalid as input and should not be used. However, the product is trusting that you (as an expert) know this, and would not use such a current table space as input. Additionally, if the process normally would require a log-forward mode, such as when mass deletes are present, forcing the product to use log-backward mode causes problems.

Caution: It cannot be stressed enough that only experts who are aware of all relevant factors should use this option, and only in very exceptional conditions. All results are unpredictable and unsupported if improper input is specified.

Only a single table space can be processed using the expert options. If you must process multiple table spaces, each table space must be run as its own separate detail report job (they can all be off of a single general report run, however, by specifying a single table space filter for each detail report job). Given the highly specialized nature of this process, it is not meant for system-level recovery, but rather a very focused recovery effort (for example, a single table or a single page).

Using expert options

These expert settings are only available from the detail report panel and by setting the **Expert options** field to **Y**. A panel is displayed for you to specify your expert-level parameters. Basically, you are telling the product which data set(s) to use as a starting point for resolving full row images, and where in the Db2 log the product should start and stop reading log records. The data set name provides the input resource which is similar to how image copies and current table space pages are normally used. For example, in a log-forward method, the product would normally use an image copy it determined to be appropriate as a starting point. Using expert options, you can tell the product the name of a data set to use instead of that image copy.

When specifying input data sets, you also are asked for information about the data set type, device, file sequence number, inline indicator and volsers. Not all of this information is required all the time. Usually, only the data set type field is required, and the other values are necessary only if Log Analysis Tool cannot determine them automatically. If the data set is cataloged or available in SYSIBM.SYSCOPY, most of this additional information is not required.

The possible types of data sets are:

- Db2 VSAM data set (type T)
- An image copy whose entry is recorded in SYSIBM.SYSCOPY (type I)
- An image copy which has no entry in SYSIBM.SYSCOPY (type O)

Type T must be a VSAM data set; types I and O cannot be VSAM data sets. Though multiple data sets can be provided, multiple data sets can be provided only if partitioning is used on the single table space. There is no other possible reason to specify more than one data set. If partitioning is used, the data sets must be provided in ascending partition order, and must all be the same type. That is, you cannot specify an input data set for partition 1 as type T (current table space) and partition 2 data set as type I (image copy data set). All data set types must be identical, and this is forced on the input panel.

Additionally, you must supply the log point to tell the product where to start or stop applying log records. These values (for example, Starting RBA) determine the method of log processing as well.

- If you provide a starting point (RBA or LRSN), you are telling Log Analysis Tool to use a log-forward method. Normally the product determines this point automatically via SYSIBM.SYSCOPY information, but this option overrides that functionality.
- If you provide an ending point (RBA or LRSN), you are telling Log Analysis Tool to use a log-backward method. Normally the product determines this as well using values from the current table space(s), but again this option overrides that functionality.

Important: These start and end values are extremely important and must be appropriate given the input data sets. These values also must be appropriate within the context of the general report timeframe.

Expert options examples

Before discussing the examples, a simple explanation of how log-forward and log-backward methods relate to general output is necessary. The general report will pass information to the detail report regarding the high and low values (RBA/LRSN) it processed. A log-forward detail report will read the Db2 logs from the image copy creation value forward to the low general report value. A log-backward method will read from the current table space values (assume current time, for simplicity) backward to the high general report value. In this way there is no overlapping.

It is important to understand the basic workings of the detail report, and how this relates to the general report. An example:

```
Time 10:00                  Time 11:00-11:30              Time 13:00
--------------------        ----------------------        -----------------
 Image copy X created        General report request        Current time with
 with Start RBA = 100        covers RBAs = 220-270          RBA = 350 recorded
                                                            in current data
                                                            (DB2 VSAM table-
                                                            space dsn = Y)
```

The example indicates that an image copy (X) was taken at 10:00. A general report was requested with a timeframe of 11:00 through 11:30. The current table space (data in the Db2 VSAM data set backing the table space) has an RBA = 350 and the current time is 13:00. To resolve full row images for the activity reported on from 11:00 to 11:30, the product has two possible methods: use image copy X as a starting point, apply all log records from RBA = 100 to RBA = 270 (log-forward), or use current table space Y, apply all log records from 350 back to 220 (log-backward).

Using an image copy has always meant a log-forward method; using a current table space has always meant a log-backward method. This expert option allows you to control the input-type and the processing method (forward/backward). An image copy can now be used in log-backward mode, and a current table space can now be used in log-forward mode.

Example 1

In the example, suppose an image copy (Z) was actually taken at time = 12:00, RBA = 310. You could tell the product to use Z and specify an ending RBA = 310. The ending RBA forces log-backward, which is what you would need as image copy Z occurred after the general report timeframe. It would not be possible to specify a starting RBA value using Z because RBA = 310 could not be read forward to RBA 270. This is an error. So you are in control of the input and processing method, and both must make sense in the context of the general report. Some limited checking of this type is done by Log Analysis Tool, but in expert mode you are essentially telling the product to bypass most validation and trust your input.

The rule in this instance is that a starting value (RBA/LRSN) must be lower than the lowest value processed by the general report, and this will result in a log-forward method in the detail report. The ending value (RBA/LRSN) must be higher than the highest value processed by the general report, and this will result in a log-backward method in the detail report. The starting and ending values are all mutually exclusive.

Example 2

Assume that a single table had several updates a month ago and those need to be resolved. That is, you need to see the full pre-update and post-update row images. However, many REORGs have been

performed since that time, so log-backward is not available (nor desirable as many, many logs must be read for the month period). An image copy is available, but it was taken two months prior to these updates. Once again, this is too many logs to read. As an expert, you know that image copy can be used because you know this table was not modified during those two months until these updates (though many other tables in the same table space were modified).

To use this expert option, you could specify a general report timeframe of those updates in question, let's say from 12:15 to 3:45 on a single day. After the general report is run, type Y in **Expert options** on the detail report panel. On the next panel displayed, specify the name of the usable image copy, and a starting RBA equal to some appropriate value. In this case, that value may be the lowest URID in the general report minus 1. That way, the image copy will be used as a starting point for the table, only the updates in question will then be applied, and full row images can all be materialized with a few log record reads rather than months of log data.

This example is just one possible use of this option. But it should be clear that the potential for error here is great. For example, if the table was actually updated during those two months, the results cannot be predicted, except to say they will be wrong. Also, assume you specified an ending RBA value rather than a starting RBA. This would tell the product to use the input data set (an older image copy, in this example) and then process all log records from the highest RBA reported on in the general report to that ending RBA value (because an ending RBA forces a log-backward method). Clearly, this would not yield good results.

Though it has been mentioned several times, it is imperative that you understand that using inputs outside of this product's ability to validate those inputs can result in many possible errors. The worst type of error could well be a return code of 0, meaning that the processing completed without errors, but the results are completely wrong. Abends of any type are also possible. Support for any problems encountered while using this option will be extremely limited as providing incorrect resources can lead to so many unpredictable problems because you are bypassing normal product validations. You must be sure that your input should work prior to notification of errors using this option. Issues such as data compression, REORGs, and other operations must all be taken into account while providing input resources.

Specifying expert options

When you type Y in **Expert options** and press **Enter**, the Expert options panel is displayed.

```
V3.5.0 ---------------------- Expert Options ------------------- SC01/SS1A
Command ===>

Hit 'Enter' to use these values; 'PF3' to bypass use of the expert option

 Starting RBA.........                    (forces log-forward method)
 Starting LRSN........                    (forces log-forward method)
 Ending RBA...........                    (forces log-backward method)
 Ending LRSN..........                    (forces log-backward method)
*Input dataset type...          (T - VSAM tablespace, I - Image copy,
                                  O - Other)

 Input Dataset Name                          Device   F Seq R Vol Ser Num(s)
 ------------------------------------------- -------- ----- - -------------
```

Figure 39. Expert Options panel

Starting RBA

This value is mutually exclusive with all of the other Starting and Ending fields on this panel. The RBA specified (12-character hexadecimal display format only) is used to force a log-forward method of processing. This start RBA value determines where in the Db2 log the detail report starts reading log records. It must coincide with the input override data set names. That is, the input override data sets are used as the sole resource for resolving starting full row images, and this start RBA value indicates the point in the Db2 logs at which to begin applying log records (changes) to those starting full row images. It stops applying changes at a normal point, as predetermined by the product. A

start RBA value must be used on a non-data sharing system. An LRSN value causes unpredictable and unsupported results on a non-data sharing system.

Starting LRSN
This value is mutually exclusive with all other Starting and Ending fields on this panel. The LRSN specified (12-character hexadecimal display format only) is used to force a log-forward method of processing. This start LRSN value determines where in the Db2 log the detail report starts reading log records. It must coincide with the input override data set names. That is, the input override data sets are used as the sole resource for resolving starting full row images, and this start LRSN value indicates the point in the Db2 logs at which to begin applying log records (changes) to those starting full row images. It stops applying changes at a normal point, as predetermined by the product. A start LRSN value must be used on a data sharing system. An RBA value causes unpredictable and unsupported results on a data sharing system.

Ending RBA
This value is mutually exclusive with all other Starting and Ending fields on this panel. The RBA specified (12-character hexadecimal display format only) is used to force a log-backward method of processing. This end RBA value determines where in the Db2 log the detail report stops reading log records. It must coincide with the input override data set names. That is, the input override data sets are used as the sole resource for resolving starting full row images, and this stop RBA value indicates the point in the Db2 logs at which to stop applying log records (changes) to those starting full row images. It starts applying changes at a normal point, as predetermined by the product. An end RBA value must be used on a non-data sharing system. An LRSN value causes unpredictable and unsupported results on a non-data sharing system.

Ending LRSN
This value is mutually exclusive with all other Starting and Ending fields on this panel. The LRSN specified (12-character hexadecimal display format only) is used to force a log-backward method of processing. This end LRSN value determines where in the Db2 log the detail report stops reading log records. It must coincide with the input override data set names. That is, the input override data sets are used as the sole resource for resolving starting full row images, and this stop LRSN value indicates the point in the Db2 logs at which to stop applying log records (changes) to those starting full row images. It starts applying changes at a normal point, as predetermined by the product. An end LRSN value must be used on a data sharing system. An RBA value causes unpredictable and unsupported results on a data sharing system.

Input dataset type
This field setting defines the type of data sets specified. The possible values are:

T
Table space. The data set is a Db2 VSAM data set containing Db2 table space data. It must be cataloged. No other fields on the line are valid (that is, the S field and Volume serial numbers are invalid).

I
An image copy that can be found in the current SYSIBM.SYSCOPY catalog table. All information about this image copy can be resolved via the current Db2 catalog (via table SYSIBM.SYSCOPY). No other fields on the line are valid (that is, the S field and Volume serial numbers are invalid).

O
Some other type of image copy data set. The information for this data set cannot be resolved from the Db2 catalog (via table SYSIBM.SYSCOPY) and it is not a Db2 VSAM data set. All information about the data set must come from this panel. If it is cataloged, no other fields on this line need be entered. If it is not cataloged, the volume serial numbers field must be non-blank. The S field is optional, depending on the file.

Input dataset name
At least one data set name must be provided, except in the case described in the next paragraph. This data set must contain an image copy, Db2 table space data component, or some other data set that exactly mirrors those type of resources. Any other type of formatted data set is not allowed, and the results are unpredictable and unsupported. Multiple table spaces cannot be processed here in a single run; only a single table space is allowed. That table space can be partitioned, and each partition needed must be included either in separate data sets, or in a single data set. This input

is considered a 'trust me' input. That is, you are solely responsible for ensuring that this input is valid and appropriate for the run and no other inputs are needed. If that rule is broken, results are unpredictable and unsupported. Multiple data sets reflecting different images of the same table space (at different times) is also not supported. A maximum of 8 data sets can be provided here. If more are needed, edit the JCL, adding them to the list identified by DD XOPTS. Partitioned data sets must be in partition ascending order, as needed.

You can leave this field blank and all the rest of the data on the line will be ignored, but only if the input data set type = 'T'. A blank field here means use the current table space only, and let the product determine the data set name(s). This can be useful if your table space has many partitions and you wish to use the active current Db2 table space only, because you don't have to type in all the different partitioned data set names (that is, ...A001 through A00n where 'n' is the number of partitions). If the field is blank, any other lines of input data are ignored.

Device
Enter up to eight characters to identify the device type (for example, TAPE). This value is valid only for an input data set of type O. Leave it blank otherwise.

F Seq
File sequence number. Enter a numeric value here (0-65535) only if the data set is not cataloged. This value is invalid for types T and I. Leave the field blank if no file sequence number is needed.

R
Indicates if the data set is an inline image copy data set. That is, whether it was created during a REORG utility. If it was, type Y, otherwise leave it blank or type N (they are treated identically). Specify this value only for data set type equal to O; it is invalid for any other data set type specification regardless of whether it is an inline image copy or not.

Vol Ser Num(s)
Enter up to three comma-separated volume serial numbers. Do not embed any spaces between the specified volume serial numbers. If more are needed, edit the JCL where DD XOPTS is found (the total maximum is eight volsers per data set even when edited from the JCL, so if more than that are needed, you must catalog the data set to use it). This value is invalid for types T and I. If the data set is cataloged, leave the field blank. An example of adding three volume serial numbers is shown here:

```
XT0123,XT0124,XT0233
```

Chapter 8. Loading data into Db2

About this task

The **Load data into DB2** option allows you to view the report data in a format you prefer. It also allows you to load data into Db2 tables for replication purposes. The data created in the report is written to a file in Db2 LOAD format. Log Analysis Tool also generates the necessary DDL and LOAD format JCL to view or replicate your data.

You can run this option as many times as you want against the same report data, specifying different options for each run. As long as the input (ROWDATA VSAM file) does not change, the data is available. If you run another report, then this will modify the contents of the ROWDATA input data set, and the load data option will only be available against that new set of data.

Important: Db2 imposes a maximum limit to the amount of information allowed in a row in the target table. This limit is 32k. If the amount of data that you have is near that limit, you must also take into account that Log Analysis Tool includes the table name and table owner of each row for identification purposes.

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **3**, **Load data into DB2**.

   The "Load data into DB2 panel" is displayed, as shown in the following figure.

```
V3.5.0 ------------------- Load Data into DB2 ------------------- SC01/SS1A
Command ===>
                                                       More:      +

*Action................... E   (E - Edit, S - Submit)
 Job Identifier........... PDUSR
*General report data...... N   (Y/N)
*Delimited output......... N   (Y - Yes, N - No, S - Specify options)
*Fixed external format.... N   (Y - Yes, N - No)
*Header................... N   (Y - Yes, N - No, S - Specify options)
*Change Flag.............. N   (Y/N)
*Data Only................ Y   (Y/N)
*Create target tables..... N   (Y/N)
 Database Name............
 Tablespace Name..........
*Log...................... N   (Y/N)
*Audit mode............... Y   (Y - Yes, N - No, S - Specify options)
*LOAD specification....... A   (A - All, B - Insert, C - Delete,
                                D - Update post, E - Update pre)
 LOB PDS name.............
*Split load files......... Y   (Y/N)
*Split load cards......... Y   (Y/N)
*Modify split LOADFILE DSN N   (Y/N)
*Edit column order........ N   (Y/N)
*Expert options........... N   (Y/N)
*Include load step in JCL. N   (Y/N)
 Source fields:                    Target fields:
 SSID...................: DB1A         *SSID.................. DB1A
 Table Owner/Name (LOAD from):        Table Owner/Name (LOAD to):
 _____ / _____  _____ / _____
 _____ / _____  _____ / _____
 _____ / _____  _____ / _____
 _____ / _____  _____ / _____
```

Figure 40. Load data into DB2 panel

2. Select the desired options and press PF3 to process. Log Analysis Tool generates the JCL to load the data into the target table and saves it in a file named LOADFILE.

3. Submit the generated JCL to load the data into the target table and query it using any standard methods.

- **Load data into DB2 panel fields:**

  Action
  Type E to edit the generated JCL, or S to submit the JCL directly to generate the report.

  Job Identifier
  This is the same job identifier as used in general/detail reports and is an eight-character maximum field with which data set names for different reports can be identified.

  General report data
  - Type N to indicate that the data is strictly detail report data. This is the default.
  - Type Y to indicate if the data you wish to load is strictly general report data. You are loading data into a Db2 table which can be generated from the general report only. No detail report run is necessary. Only header fields can therefore be loaded when selecting this option. No row data is available. You might want to load general report data if you only require audit type information, and do not need to track actual data changes to rows.

    There is no concept of pre- or post-row images using general report data. There is only one row per action indicating such things as who did the action, what the action was (update/delete/insert), on what table the action took place, and so on.

    The header fields loaded are those you specify using the "Header=S" field setting. When set to Y, you:

    - cannot use Audit Mode
    - must include the header
    - must not be data only
    - the Change Flag is ignored
    - load specifications are relevant, but not for pre- or post-update row images. There are no pre- or post-row images here, so including either will get you all updates. Excluding both will omit all updates.

  Delimited output
  Type a Y (yes), an N (no), or an S (specify options) to indicate you wish to create a load file in delimited format. When creating a delimited output file, the data is written using an external format which means the data must be formatted. If the "Skip formatting (when possible)" option is used in conjunction with this option, formatting is still performed. When using this option, you may customize the column delimiter, string delimiter (or character delimiter), and the field delimiter (the value used to "quote" the column names in the when clause of the load cards). These delimiters can be changed anytime by specifying S for this option. In this case, another panel is displayed for the delimiter settings. When set to Y, you:

  - cannot load General report data
  - cannot specify the Change Flag
  - cannot use Audit Mode
  - cannot use CHAR data in hex
  - must choose Table Creator and Table Name header fields, unless choosing to also split the load files

  Fixed external format
  - Type Y to indicate a fixed external format. This setting creates a fixed column length output file in external format with load cards so the Details Report data can be loaded into your other DBMS.
  - Type N to indicate that there is not a fixed external format.

  Header
  Type a Y (yes), an N (no), or an S (specify headers) for header inclusion in the target table. Includes data such as database, time, and so on. If you specify Y all specified headers are included in the target table. If you specify N and **Data Only** is also N, then two column headers

(Row action and status) are included to identify the row. This field cannot be set to Y if **Data Only** is also set to Y. Setting this field to N with **Data Only** set to Y means no column headers are included. When set to Y, the headers previously specified will be used. If you have not yet specified your headers, type **S** at least one time to do so. These specified headers can be changed anytime by specifying S. In this case, another panel is displayed for your header settings. See "Customizing headers during LOAD" on page 149 for more information.

Change Flag
Type a Y (yes) or an N (no) to indicate whether you want each column preceded by a change indicator in the target table. The change indicator tells whether or not the column value has changed between the pre-change and post-change row, and is represented by a C in the generated column (LAT_CI_nnnnn) preceding the column if a change occurred. If no change occurred, this column is left blank.

Data Only
Type a Y to indicate that you want to load only the column data into a target table. The LOAD cards are generated such that no Log Analysis Tool information is included in the loaded data. This means that there will be no indication of action types (update/insert/delete) or row status (pre or post row image). This field is mutually exclusive with the Header and Change flag fields. Setting this field to Y is useful if data propagation is the desired result. If set to Y, it is most likely that the LOAD specification field would be set to 1, to include only INSERT type records. See the help for the LOAD specification field for more information.

> ⚠️ **Attention:** Though not available from the panel, there are extra parameters available to further control the output/format of the load file contents. See "Advanced LOADFILE creation" in this document or the tutorial topic "Load data into DB2" for more information.

Create target tables
Type an N to bypass creating any target tables. The target tables must already exist at the target subsystem (SSID).

Type a Y to create the target tables at the target subsystem (SSID). The target tables must not exist at the target subsystem, or else an SQL error will occur. The database and table space name fields cannot be blank.

Database Name/Tablespace Name
Enter a single database and table space name into which the targeted table(s) will be created. These fields must be non-blank if the Create target tables field is set to Y. If that field is set to N, these fields are ignored. The database and table space named here must pre-exist, the product will not create them. They also must be appropriate for any created tables, including the audit table. For example, if you are requesting creation of the audit table, and you have defined an audit table where the row size can be greater than 4K in length, then the database/table space into which you are creating this table must also handle page sizes > 4K. If not, SQL errors will occur during the table creation process. The pre-existing table space must not be partitioned.

Log
This field determines the setting for the LOG parameter in the LOAD utility. Valid values are:

Y
LOG YES. Db2 logs the loaded records and the table space is not placed in COPY PENDING status.

N
LOG NO. No logging is done, and the table space is placed in COPY PENDING status.

Audit Mode
This field determines if you are running in a special mode called audit mode.

– Type Y to indicate that you will be running in audit mode.
– Type N to indicate that you will not be running in audit mode.
– Type S to indicate that you want to specify the audit mode parameters.

If you specify **S**, more panels will be displayed for your audit mode specifications. After you finish your specifications, you will be returned to this panel with the Audit Mode field set to **Y**, and you should reset this field to either **Y** (use audit mode) or **N** (bypass audit mode).

Important:

– If you request audit mode (field set to **Y**), fields Header, Change Flag, and Data Only are ignored. Your audit mode specifications determine the layout of the audit table. See the tutorial for more information about running in audit mode.

– If you request audit mode (field set to **Y**), and there is LOB/XML data present, only the locator values will be saved, which are of little use. The actual LOB/XML data cannot be audited this way. See the tutorial for more information about LOB/XML data.

Load Specification
This field indicates which records from the detail report process will be used in the LOAD utility. The letters indicate the records that will be included:

A
- All records are included.

B
- INSERT records (post-change) are included.

C
- DELETE records (pre-change) are included.

D
- UPDATE post-change records are included.

E
- UPDATE pre-change records are included.

Any combination of these record types can be requested. Simply include the letters as needed in the field provided. For example, BC means include INSERTs and DELETEs only.

LOB PDS name
See "LOB PDS name" in "Generate details" on page 98.

Split load files
This field indicates whether the LOAD files will be split into one file for each table or if there will be one LOAD file for all the tables. The LOAD card information will be in one file regardless of whether the LOAD files are split. The DD LOADFILE specifies the LOAD file if there is no splitting and dynamic allocation has been overridden.

The DD L*dddtttt* specifies the LOAD file for each of the tables if the tables are to be split and dynamic allocation has been overridden. The 'ddd' of the DD name is replaced with the last three hexadecimal digits of the table's database ID. The 'tttt' will be replaced with the hexadecimal digits of the table's ID. For example, a table with a DBID of '4A20' and an OBID of '12BA' would be specified with the DD LA2012BA. When dynamic allocation is overridden and the LOAD files are to be split, it is up to the user to provide the DD for all tables requiring a LOAD file. An error occurs if this is not done.

In the case where dynamic allocation is not overridden, the data set for the split LOAD files is renamed using the same L*dddtttt* format. For example, if no data set prefix is specified and the job identifier is TEST01, the data set name will be USERID.L*dddtttt*.TEST01 for each table.

Important: When LOAD files are not split, the generated JCL has a step to LOAD the data into the specified tables using the single LOAD file and the LOAD card. When the LOAD files are split, the generated JCL does not attempt to run a LOAD and instead just creates the LOAD card and the various LOAD files.

Split load cards
This field indicates whether the LOAD cards will be split into one file for each table or if there will be one LOAD card file for all the tables. The DD LOADCARD specifies the LOAD cards if there is no splitting and dynamic allocation has been overridden.

The DD C*dddtttt* will specify the LOAD file for each of the tables if the tables are to be split and dynamic allocation has been overridden. The *ddd* of the DD name will be replaced with the last three hexadecimal digits of the table's database ID. The *tttt* will be replaced with the hexadecimal digits of the table's ID. For example, a table with a DBID of '4A20' and an OBID of '12BA' would be specified with the DD CA2012BA. When dynamic allocation is overridden and the LOAD cards are to be split, it is up to the user to provide the DD for all tables requiring a LOAD card. An error will occur if this is not done.

In the case where dynamic allocation is not overridden, the dataset for the split LOAD cards will be named using the same L*dddtttt* format. For example, if no data set prefix is specified and the job identifier is TEST01, the dataset name will be USERID.C*dddtttt*.TEST01 for each table. When LOAD files are not split, the generated JCL will have a step to LOAD the data into the specified tables using the single LOAD file and the LOAD card. You cannot choose to split LOAD cards without choosing to split the LOAD files.

Modify split LOADFILE DSN
Enter Y in this field to modify data set node names for the split load files and the load cards. This option is only valid if **Split load files** is set to Y. When you press Enter, a panel will display that allows you to modify the data set node names. For more information, see "Modifying split LOADFILE data set names" on page 146.

Edit column order
This field indicates whether the column ordering for one or more tables in the LOAD should be changed. By default, the columns in the LOAD are ordered the same as the ordering of the table that the data represents. A value of Y indicates that columns should be reordered or excluded for at least one of the tables. A panel will display allowing for the reordering to be specified.

Expert options
This field indicates that you want greater control over how the LOAD file is created. These parameters are provided in case the default generation of the LOADFILE does not meet your specific needs. To use this option, expert knowledge of your data is required. A value of Y indicates the 'LOAD expert options' panel should be displayed.

Include load step in JCL
Specify Y or N to indicate whether you want to add the load step (LOADFILE) in the JCL that is generated. The default is N. Depending on other parameters chosen, you may not be able to specify a Y.

SSID
Specify the subsystem ID (SSID) for the Source fields and for the Target fields. The default will be listed in both fields.

Source fields
The Source fields information relates directly to the Target field information. The information under the Source fields heading represents which Db2 subsystem and tables the data originates from, and the Target fields information represents which Db2 subsystem and tables the data will be loaded into.

Each entry line for table owner and name can contain a source table owner and name (think of this as the object from which you are LOADing) and a target table owner and name (think of this as the object into which you are LOADing). If the table owner or name is longer than the allotted space, see "Using Db2 long names" on page 121 for information on how a pop-up panel displays so you can complete the value. The source SSID is a display-only field showing which Db2 subsystem the detail report ran against.

Source tables can be omitted, and this infers that all tables reported on in the detail report are to be included. It also infers that the target tables are identical to the source tables. That is, a target table can only be specified on a line where a source table was specified, so if no source tables are specified, no target tables can be specified.

In this case where no source tables are provided, and therefore the target tables will have the same names as the source tables, two possible scenarios are being requested:

  a. You want to LOAD data into the originating tables.

b. You want to LOAD data into identically named tables on another subsystem.

The first scenario will be true if the source and target SSIDs are equal. The second scenario will be true if these SSIDs are not equal. Be very careful when omitting source table specifications as you may impact production tables, especially in the first case scenarios.

There are several entries possible on this panel. If more are needed, edit the JCL to specify as many tables as needed. The DD named DATAIN contains the control cards and the line(s) starting "SOURCE/TARGET TAB=s1s2/t1t2" show the current source and target tables requested, where `s1,t1 = source, target table owner`; `s2,t2 = source, target table  name`. Follow this format exactly or results are unpredictable. Source table, if left blank on the panel, terminates the list.

Target fields

All the fields under this subheading reflect information regarding the target for the LOAD utility process:

– The SSID field must be entered to direct the LOAD utility to the execute on the proper Db2 subsystem.

– The table owner/name fields indicate the target tables into which the data will be loaded. If a target owner/name field is specified, the corresponding source entry on the same line must also be specified. If the target owner/name entry is left blank on any line which has a source table owner/name, it is implied that the source owner/name is the same as the target owner/name. In that case, if the target SSID is identical to the source SSID, the LOAD process will be loading data into the same table where the log events originally occurred.

  Note: If the table owner or name is longer than the allotted space, see "Using Db2 long names" on page 121 for information on how a pop-up panel displays so you can complete the value.

If no source tables are specified, no target tables can be specified.

Specifying delimiters

The Specify Delimiters panel allows you to specify delimiters for columns, string, and field parameters.

```
V3.5.0 ------------------ Specify Delimiters ------------------ RS22/DC1A
Command ===>
Hit 'ENTER' to process panel;
'PF3' to cancel updates to parameters Enter the hexadecimal value of the delimiter to be used.
*Column delimiter................. 6B
*String delimiter................. 7D
*Field delimiter................. 7F
```

Column delimiter

Enter up to three characters to use as a column delimiter when creating a delimited output file. If a value is not entered, the default value of 6B will be used which is the EBCDIC equivalent of ','. This value cannot be the same value as any of the other delimiters or as the default decimal point in use.

String delimiter

Enter a character value to override the default string delimiter when creating a delimited output file. If a value is not entered, the default value of 7D will be used which is the EBCDIC equivalent of "'". This value cannot be the same value as any of the other delimiters or as the default decimal point in use.

Field delimiter

Enter a character value to override the default field delimiter when creating a delimited output file. If a value is not entered, the default value of 7F will be used which is the EBCDIC equivalent of '"'. This value cannot be the same value as any of the other delimiters or as the default decimal point in use.

LOAD expert options

In the LOAD step of a Log Analysis Tool job, there are expert parameters that can be used to control how the LOAD file is created. These parameters are provided in case the default generation of the LOADFILE does not meet your specific needs.

When you specify **Y** in the "Expert options" field on the "Load data into DB2" panel, you can identify which expert options you want to use:

```
V3.5.0 ------------------ LOAD expert options ------------------ SC01/SS01

Hit 'ENTER' to process panel; 'PF3' to bypass use of the expert options

*Bypass DEFAULTIF................. N   (Y/N)
*Pack LRSN field.................. Y   (Y/N)
*Fixed length table names......... N   (Y/N)
*Max table owner field length..... 008 (008-128)
*Max table NAME field length...... 008 (008-128)
*Pad VARCHARs with spaces......... N   (Y/N)
*Bypass NULLIF................... N   (Y/N)
*Header fields in EBCDIC only..... N   (Y/N)
*Use CONTINUEIF in DB2 LOADFILE... N   (Y/N)

COMMAND.....
```

Figure 41. LOAD expert options panel

LOAD expert options panel fields

Bypass DEFAULTIF
Specify either **Y** or **N** to identify if you want to include any logic for determining DEFAULTIF specifications.

If you use this option, it is imperative that no default values are needed in your output. For example, assume a table was recently altered with this definition: ALTER TABLE TABX ADD COL100 INTEGER NOT NULL WITH DEFAULT. If a delete occurs against a row in this table that still has no value for the new column, there will be no data for that new column. This means a DEFAULTIF indicator is required for proper LOADFILE generation. Results are unpredictable if this option is used in such a case.

Pack LRSN field
Specify either **Y** or **N** to identify if you want to make the LRSN header field a packed field in the LOADFILE.

The LRSN header field is normally a hexadecimal display field, which does not allow for sorting. Making it a packed field allows for sorting on this field. It does not change the definition of the target header column (if used), which is still a CHAR(12) field, but would contain only the packed format for 6 bytes. You must include the header for this to have any effect.

Fixed length table names
Specify either **Y** or **N** to identify if you want to make the table name fields fixed length.

Normally these fields, which identify the table in the LOADFILE row, are variable length. Using this setting makes them fixed for lengths of 8 (table owner) and 18 (for table name). Do not use this setting if using long table names. Use of this option can be helpful if wanting header fields lined up in the same column for each row. For example, if you are going to pack the LRSN field and sort on it, fixing the table names allows you to sort on this field as it will be in the same column for each row in the LOADFILE.

Max table owner fixed length
Specify a value between 008 and 128 for the maximum length of the table owner.

Max table name fixed length
Specify a value between 008 and 128 for the maximum length of the table name.

Pad VARCHARs with spaces
Specify either **Y** or **N** to identify if you want to pad VARCHAR column data fields with spaces.

For example, if a VARCHAR(10) field has a current length of 4, the remaining 6 characters are padded with spaces. Normally, Log Analysis Tool does not pad the extra length with anything, and the contents of those unused bytes are unknown. This has no effect on the Db2 LOAD utility, but may affect users' downstream applications that may process these fields.

Bypass NULLIF

Specify either **Y** or **N** to identify if you want to include any logic for determining NULLIF specifications.

Normally, columns not defined as NOT NULL have an indicator byte used to indicate whether or not the data for that column is null. In many cases a column like this never has null values. Bypassing the NULLIF removes this field. If a null is encountered, there is no longer a way for Log Analysis Tool to indicate it. Log Analysis Tool clears out the data with zeros and issues a warning. If this occurs, loading the data may not be possible. For example, if NULLIF is bypassed and Log Analysis Tool encounters a nulled timestamp, the data is set to all zeros. A LOAD of this data will not be possible because this is an invalid timestamp for the load utility. Also, it is possible for the NULLIF and the DEFAULTIF indicators to use the same byte. To remove the byte entirely, both NULLIF and DEFAULTIF should be bypassed.

⚠️ **CAUTION:** This is an advanced feature and must be used with care. The use of some of these parameters may lead to invalid results, so expert knowledge of your data is required.

Header fields in EBCDIC only

Specify "Y" or "N" to indicate whether to display the header fields in EBCDIC format, rather than ASCII.

Use CONTINUEIF in DB2 LOADFILE

Specify "Y" or "N" to indicate whether to allow CONTINUEIF statements in the LOADFILE. These statements allow the load file to continue processing if certain conditions are met.

Modifying split LOADFILE data set names

When you request the LOADFILE to be split into one data set per object, you can modify data set node names for the split load files and the load cards so that object IDs are not used in the data set names.

About this task

Log Analysis Tool uses an internal mapping table to map the node names to table names. The node names are suffixed with an "L" to identify the load file DSN and DDNAME, and are suffixed with a "C" to identify the load card DSN and DDNAME.

Procedure

1. On the **Load Data into DB2** panel, specify Y in the **Split load files** and **Modify split LOADFILE DSN** fields (and optionally in the **Split load cards** field) and press Enter.

   The **CUSTOMIZE LOADFILE/LOADCARD DSN NAME** panel is displayed, as shown in the following figure:

   ```
   V3.5.0 ------------ CUSTOMIZE LOADFILE/LOADCARD DSN NAME ----------- SC01/SS01
   Command ===>


   Hit 'Enter' to save current mapping; 'PF3' to exit and use saved mapping
   Type 'Cancel' to bypass use of all saved mappings

   *Table owner          *Table Name                      *Node name
   -------------------   -----------------------------     ----------
   ```

   *Figure 42. **CUSTOMIZE LOADFILE/LOADCARD DSN NAME panel***

2. On the **CUSTOMIZE LOADFILE/LOADCARD DSN NAME** panel, enter the table owner and corresponding table name in the fields. The initial **Table owner** field allows 20 characters and

the **Table Name** field allows 30 characters. If more characters are needed for either field, enter the maximum number of characters and press Enter. A window is displayed to allow you to enter additional characters.

3. In the **Node name** field, enter a unique node name, up to 7 characters. The 8th character is reserved; Log Analysis Tool will use that character to append an "L"(for load file) or "C" (for card file) to the node name. Each node name must be unique.

4. Continue until you have entered all of the desired DSN modifications. If more than 14 table names must be specified, press Enter after specifying the last table name and node. The entries that you entered are saved, and a new panel is displayed so that you can add additional entries.

Example

Given the following information:

- DSN prefix: ALA
- Job identifier: TESTLOAD
- Table owner: TWUSR
- Table name: TABLE1
- Node name: NODE1

For the table TWUSR.TABLE1, the DSN for the load job is generated as ALA.NODE1L.TESTLOAD and for the load cards is generated as ALA.NODE1C.TESTLOAD.

What to do next

If the **Override dynamic allocation** user setting is set to Y, you must create the DD name as it is specified in the **Node name** field to link it with the Log Analysis Tool LOADDSN mapping table.

For tables that are not found in the LOADDSN mapping table, the node name is generated with object IDs, and a warning message (ALAA724W) is generated for each of those tables.

Edit column order

Traditionally, the LOAD file stores the data in the order of the columns of the table that the data represents. You have the option to rearrange the data or exclude columns from the LOAD file.

On the **Load data into DB2** panel, if you specify **Y** for the **Edit column order** field, the LOAD column orderings panel appears.

```
V3.5.0 ------------------- LOAD Column Orderings ---------------- Top of data
Command ===>
                                                           More:     +

Hit 'Enter' to save current translations; 'PF3' to exit and use saved
translations; Type 'Cancel' to bypass use of all saved translations;

Commands:  SKIP  COLUMNS

*Table Owner          *Table Name
-------------------    -----------------------------


*Column Name                  *Position (1-750, XCL)
----------------------------   ---
```

Figure 43. LOAD column orderings panel

LOAD column orderings panel fields

Table Owner
Type the table owner (creator) of the desired table. Type as much of the value as necessary until you reach the end of the field. If the table owner is longer than the allotted space, another panel will display where you can complete the value.

Table Name
Type the table name of the desired table. Type as much of the value as necessary until you reach the end of the field. If the table name is longer than the allotted space, another panel will display where you can complete the value.

Column Name
Type the name of the column to be positioned. The ordered position for this column depends on the value of the **Position** field. The first occurrence of spaces in any column name field indicates the end of data, and no further checking, validation, or saving of entries beyond that line on the panel is done. This allows for quick exclusion of unneeded entries already entered, if so desired. At least one column name must be entered for a table.

You can display all of the columns for a table by using the COLUMNS command. For this command to be used, the **Table Owner** and **Table Name** fields must be populated, and all other fields must be left blank. Enter COLUMNS and press Enter. A **Column List** panel is displayed, from which you can see all of the columns in the table, set the column positions, or exclude columns. For more information, see "Using the COLUMNS command" on page 148.

Position
Specify the position for the column in the reordered LOAD file, LOAD card, and, if requested, the target table DDL. The valid entries are any integer between 1 and 750 and "XCL".

- A value of 1 indicates that the column represented by the corresponding **Column Name** should be the first column in the generated LOAD files. A value of 2 indicates the column should be second in the LOAD files. This pattern continues for all valid integers.

- A value of "XCL" indicates that the column and its data will be excluded from any of the LOAD files.

- If a column is not explicitly entered into the column reordering panel, it will still be included in the LOAD output. All columns that are not entered are put into a queue in the order of their original column ordering in the table. Then, one by one, each one is placed in the first position that was not assigned to a column in the LOAD column orderings panel.

When you have finished entering all table and column information that you want to reorder, press PF3 to exit and use the saved reordering. You can type CANCEL at any time to bypass the use of the reordering.

Using the COLUMNS command

Use the **COLUMNS** command to view all of the columns for a table. Instead of manually entering column names, you can order and exclude columns from a list of column names. This command can be used for Db2 LOAD, fixed output, and delimited output formats.

About this task

When you use the COLUMNS command from the **LOAD Column Orderings** panel, the **Column List** panel displays all of the column names for the table, according to the order defined in SYSIBM.SYSCOLUMNS.

Refer to the help system for detailed information about column ordering.

Procedure

1. On the **LOAD Column Orderings** panel, enter the **Table Owner** and **Table Name** fields, and ensure that all other fields are blank.
2. Enter **COLUMNS** in the Command line and press Enter.
3. On the **Column List** panel, enter the position that you want a column to be in the LOAD file in the **Position** field. To exclude a column, enter XCL.

Not all columns need to have a position entered; only the columns that must be in a specific position or that you want to excluded must have their Position field specified. Columns that do not have their position entered will still be in the LOAD. Those columns will be located in the first available unused position, based on their order in the table.

Conversion of load file header data to proper CCSID

Log Analysis Tool converts log record header data in its load file to the same CCSID of the source table in normal load mode. This only affects option 3 off the main menu named "Load data into Db2" and does not affect the audit mode option allowed therein. This is done to make all data consistent within the load file and to allow better use of WHERE clauses when querying the loaded table.

The data is converted internally and (most) header character columns are no longer defined in the Db2 target table as FOR BIT DATA. The header columns now behave identically to the actual table column data itself in the load file. Some columns are still defined with FOR BIT DATA, because they may contain binary data as well, such as the LUW header column. This data cannot be safely converted, and generally should not be used in a WHERE clause.

This behavior is handled internally; you are not required to do anything. This behavior applies only to new tables that the product creates. If a pre-existing load table was created by a release of Log Analysis Tool prior to V3.2, the header columns will continue to behave as they did in that prior release. Because this process requires different DDL for the target load table (that is, no FOR BIT DATA definitions on most header columns), previously defined tables are not compatible with the load file format that was introduced in V3.2. Therefore, to take advantage of the header data CCSID conversion, the table must be created using Log Analysis Tool V3.2 or later.

Customizing headers during LOAD

This feature provides the capability to customize your standard column headers. You can select which columns you want included and customize their header information rather than accept the full set.

Remember: It is important to keep some of these column headers included, though all are fully selectable. For example, if you included just AUTHID as a column header, you would not know the action type (update/delete/insert) or row status (pre-, or post-row) of each row in the table.

When you specify **S** in **Header**, the "Customize standard header" panel is displayed:

```
V3.5.0 --------------- Customize standard header ---------------- SC01/SS1A
COMMAND ===>
                                                          More:     +
Hit 'ENTER' to process panel; 'PF3' to exit and bypass saving customization

Column ID       *Column Name                   *Include?(Y/N)
-----------     -----------------------------  --------------
DBID            LAT_DBID                       Y
PSID            LAT_PSID                       Y
OBID            LAT_OBID                       Y
DBNAME          LAT_DBNAME                     Y
TSNAME          LAT_TSNAME                     Y
DATE            LAT_DATE                       Y
TIME            LAT_TIME                       Y
TIMESTAMP       LAT_TIMESTAMP                  N
LRSN            LAT_LRSN                       Y
URID            LAT_URID                       Y
AUTHID          LAT_AUTH                       Y
PLAN NAME       LAT_PLAN                       Y
JOB NAME        LAT_JOBN                       Y
CORR ID         LAT_CORRID                     N
MEMBER ID       LAT_MEMBER_ID                  Y
CONN TYPE       LAT_CONN_TYPE                  Y
CONN ID         LAT_CONN_ID                    Y
```

Figure 44. Customize standard header panel

Important: This is a scrollable panel, because not all column headers fit onto a single displayable page. Though not all column headers are displayed, they are all listed in "Column identifiers" in "Column identifiers" on page 150.

On this panel, you include the column headers relevant to you. You can also change the column names, if so desired, but they should make sense relative to the Column ID field, which identifies the contents of each column. Changing the names or the included columns always requires creation of a new table. You cannot change the name or the inclusion of columns in a pre-existing table.

The LOAD process will only include the columns you have selected, and the names of those columns you have specified. Even if you do not want to change the column settings at all, you are required to use the specification process once prior to using header columns. The settings in the first time will reflect the forced settings of previous releases, for backward compatibility.

Column identifiers

DBID
Db2 object identifier of the database containing the audited table.

PSID
Db2 object identifier of the table space containing the audited table.

OBID
Db2 object identifier of the audited table.

DBNAME
Database name containing the audited table.

TSNAME
Table space name containing the audited table.

DATE
Date of the log activity (when the change was made).

TIME
Time of the log activity (when the change was made).

TIMESTAMP
Timestamp of the log activity (when the change was made).

LRSN
Db2 log record timestamp.

URID
Db2 RBA associated with the beginning of the unit of work.

AUTHID
Authorization ID of the user making the change.

PLAN NAME
Plan name associated with the change.

JOB NAME
MVS™ job name associated with the change (subset of CORR ID).

CORR ID
Db2 correlation ID, is actually same as JOB NAME plus 4 more bytes.

   Attention: CORR ID defaults to N in the Include field for backward compatibility of previous releases. See the tutorial topic on this panel for more information.

MEMBER ID
Member ID of the subsystem where the change was made.

CONN TYPE
Connection type associated with the change.

CONN ID
Connection ID associated with the change.

LUW
Logical unit of work ID.

PAGE
Page on which the row was modified.

RID
Row ID of the row which was modified.

ACTION
Change action, such as INSERT/DELETE/UPDATE.

ROW STATUS
Pre-row (before change) or post-row (after change) image.

Important: The X-prefixed fields listed here are VARCHAR(256) fields, and therefore can consume much space in your target table. Do not use these fields unless absolutely necessary. If used, expect that a target table space of >4K page size will be required.

XID
Distributed data only; transaction ID.

XUSER
Distributed data only; end user.

XAPPL
Distributed data only; application or transaction name.

XWRKS
Distributed data only; workstation name.

XACCTGS
Distributed data only; accounting string.

Other fields

Column Name
This field allows you to change the default name of the columns. Specify a column name up to 30 characters in length.

Include?
Specify Y to include the column in the audit table, and N to exclude the column from the audit table.

Using Audit mode

Audit mode provides the capability to use the current LOAD process to load detail level report data into a single audit table. This audit table can be a central resource for querying against, and can contain every table's modification activity. All table modifications can be loaded into the audit table, or only selected tables can be loaded. Also, in conjunction with advanced filters, a subset of a table's columns for a table can be loaded as well.

Using an audit table
The basic procedure for using an audit table is similar to other data tables. You must:

1. Create the audit table using the Audit table specifications panel.

2. Set up the JCL to populate the audit table.

3. Populate the audit table

4. Maintain the audit table:

   • Perform backups

   • Allow for table recovery

   • Archive the table

   • Clean up obsolete data in the table.

Creating an audit table

When you type S on the Audit mode field, the "Audit table specifications" panel displays. This panel allows you to define the layout of the audit table.

```
V3.5.0 ---------------- Audit table specifications -------------- SC01/SS1A

Hit 'ENTER' to process panel; 'PF3' to exit and bypass saving specifications

*Audit Table Creator.................. ===>
*Audit Table Name..................... ===> AUDIT_TABLE
*Customize audit table header columns. ===> N    (Y/N)
*Max number of audited table columns.. ===> 50  (1-750)
*Max length of audited table columns.. ===> 50  (1-255)

Audited table column names:
  *Column name prefix................ ===> LAT_COLNAME_     (max 15 chars)
  *Column value prefix............... ===> LAT_COLVALUE_    (max 15 chars)



COMMAND ===>
```

Figure 45. Audit table specifications panel

Audit table specifications panel fields

Audit Table Creator
This is the creator of the audit table and is a maximum of 35 characters in length.

Audit Table Name
This is the name of the audit table and is a maximum of 35 characters long. Though a default name is provided, set this value to any name appropriate for your site.

Customize audit table header columns
Type Y to view and/or change the default settings for the header columns. You can change which header columns to include, and the actual column names for those header columns. Type N to use the header column default settings. If set to Y, another panel is displayed showing the current settings for the header-type columns. After you finish customizing the header columns, you will be returned to this panel.

Max number of audited table columns
This is the number of audited table columns that will exist in your audit table (in addition to the header columns). Use a number between 1 and 750. For example, assume your largest audited table has 100 columns, and you will need all those columns tracked in the audit table. Therefore, a value of 100 would be appropriate.

Max length of audited table columns
This is the maximum length of the audited table columns. Use a number between 1 and 255. If any audited table column is longer than 255, it will be truncated. Every audited table column will be defined as VARCHAR with the length you specify. That is, all column data in the audit table is saved in its displayable format, as varying length character columns, not as its original column type. This is how the audit table can save all different column types from different tables in the same table. Note that you cannot use these audit columns in anyway other than as displayable values. If the original column was an integer, you cannot now perform arithmetic operations on it in the audit table.

Note: Keep in mind that the audit table row length may require that this table be created in a table space with greater than 4K page sizes. For example, if you specify 100 columns with a length of 255, you would need a table space with 32K page sizes to hold that table.

Column name prefix
This is prefix of the column names in the audit table that contain the column names of the audited tables. It is a maximum of 15 characters so that a number from 1 to 750 can be appended to the column name. It can be changed to any valid column name prefix.

Column value prefix
> This is prefix of the column names in the audit table that contain the column values of the audited tables. It is a maximum of 15 characters so that a number from 1 to 750 can be appended to the column name. It can be changed to any valid column name prefix.

The audit table is created with columns of two distinct types:

1. Header type columns. These columns hold information about the table modification, such as what table was modified, who did it, what type of action they performed (update/insert/delete). These header columns help identify the rest of the rows of audit table data.

2. Audited table columns. These columns hold the actual row data for the table modification, and identify the column names. These generic columns come in groups of two. Each audited table column must have a column to hold the name of the table column, as well as a column to hold the data itself. All column data is stored as character data, not as the type of data it was in the table. The column data comes directly from the detail report, and is always character based.

This is an example of an audit table layout with a requested maximum column length = 40 and maximum number of audited columns = 50:

```
Name: LOCAL.AUDIT_TABLE

Column definitions:
(start header columns)
LAT_TABLE_CREATOR            CHAR(8)
LAT_TABLE_NAME               CHAR(18)
LAT_DATE                     CHAR(10)
LAT_TIME                     CHAR(8)
LAT_AUTHID                   CHAR(8)
LAT_ACTION                   CHAR(2)
LAT-ROW_STATUS               CHAR(4)
(end of header columns)
(start audited table columns)
LAT_COLN_001                 VARCHAR(18)     (holds column name)
LAT_COLV_001                 VARCHAR(40)     (holds column value)
LAT_COLN_002                 VARCHAR(18)
LAT_COLV_002                 VARCHAR(40)
                .
                .
                .
                .
                .
LAT_COLN_050                 VARCHAR(18)
LAT_COLV_050                 VARCHAR(40)
```

To customize the column headers:

When you type Y in the Customize audit table header columns field, the "Customize audit table" panel appears:

```
V3.5.0 ----------------- Customize audit table ----------------- SC01/SS1A
COMMAND ===>
                                                         More:    +
Hit 'ENTER' to process panel; 'PF3' to exit and bypass saving customization

Column ID       *Column Name                 *Include?(Y/N)
-----------     -----------------------------  --------------
DBID            LAT_DBID                     Y
PSID            LAT_PSID                     Y
OBID            LAT_OBID                     Y
DBNAME          LAT_DBNAME                   Y
TSNAME          LAT_TSNAME                   Y
TABLE CREATOR   LAT_TABLE_CREATOR            Y
TABLE NAME      LAT_TABLE_NAME               Y
DATE            LAT_DATE                     Y
TIME            LAT_TIME                     Y
TIMESTAMP       LAT_TIMESTAMP                Y
LRSN            LAT_LRSN                     Y
URID            LAT_URID                     Y
AUTHID          LAT_AUTH                     Y
JOB NAME        LAT_PLAN                     Y
PLAN NAME       LAT_JOBN                     Y
CONN ID         LAT_CONNID                   Y
CONN TYPE       LAT_CONNTY                   Y
MEMBER ID       LAT_MEMID                    Y
```

Figure 46. Customize audit table panel

Important: This is a scrollable panel, because not all column headers fit onto a single displayable page. Though not all column headers are displayed, they are all listed in "Column identifiers" on page 150.

This scrollable panel allows for both specification of desired header type columns in the audit table and definition of those column names. Though all columns can be excluded, the data in the resulting audit table would be very difficult to identify completely without these heading columns.

Customize audit table panel fields

Column Name
    This field allows you to change the default name of the columns. Specify a column name up to 30 characters in length.

Include
    Type Y to include the column in the audit table. Type N to exclude the column from the audit table.

Chapter 9. Executing dynamic SQL in batch

You should use **Execute dynamic SQL in batch** if you want to execute the generated SQL in batch mode, as opposed to submitting it through the Db2 SPUFI utility.

The SQLOUT DD file generated during the detail report run is used as input to this batch process. The SQL is executed dynamically.

- If you requested redo SQL during the detail report run, the execution of the generated SQL will redo those changes.
- If you requested undo SQL during the detail report run, the execution of the generated SQL will undo those changes.

This process is very similar to running your SQL through SPUFI, except it executes in batch mode. The SQL should not be modified in any way prior to execution without complete knowledge of the results of such actions. As with any SQL execution, errors can occur. Restart capabilities are built into this process.

From the Db2 Log Analysis Tool Selection menu, select option **5**, **Execute dynamic SQL in batch**.

The Execute dynamic SQL in batch: Fast Apply panel appears.

```
V3.5.0 -------- Execute dynamic SQL in batch: Fast Apply --------- SC01/SS1A
COMMAND ===>
                                                            More:     +

*Action.............  E           (E - Edit, S - Submit)
 Job Identifier......              (8 character maximum)
*DB2 subsystem name.. SS1A         (DB2 SSID)
 LE runtime loadlibs.

*Restart............  I           (A - Auto, I - Ignore, S - Startover,
                                       N - New, M - Number)
 Restart Name........              (not used if Restart=I)
 Restart SQL number..              (only used if Restart=M)
*Commit Frequency....  1
*Commit Type.........  T           (T - Transaction, S - Statement)
*Loglevel............  1           (1-9 only)
*Bypass Multi-Row Inserts.            (Y - Off, N - On)
*Table filters............ N          (Y/N)

*ON SQLERROR.............. S           (S - Stop, W - Warn,
                                        I - Ignore, R - Rollback)
*ON SQLWARN..........  I          (S - Stop, W - Warn,
                                        I - Ignore, R - Rollback)
*ON Multiple rows....  I          (S - Stop, W - Warn,
                                        I - Ignore, R - Rollback)
```

Figure 47. Execute dynamic SQL in batch: Fast Apply panel (top half of panel)

```
V3.5.0 -------- Execute dynamic SQL in batch: Fast Apply --------- SC01/SS1A
COMMAND ===>
                                                            More:    -
 ON SQLCODEs (Code,S/W/I/R;Code,S/W/I/R;etc):


Source tables:                      Target tables:
Table Owner/Name (Convert from);     Table Owner/Name (Convert to):
          /                                  /
          /                                  /
          /                                  /
          /                                  /
          /                                  /
```

Figure 48. Execute dynamic SQL in batch: Fast Apply panel (bottom half of panel)

Execute dynamic SQL in batch: Fast Apply panel fields

Action
Type E or S.

E

Indicates that the temporary JCL created as a result of the data entered on the panel will be displayed in edit mode.

S

Indicates the temporary JCL will be submitted automatically.

Job identifier
This is the same job identifier as used in general and detail reports. It is an eight-character maximum field that is used to identify the data set names for the different runs of reports. It must be a valid node in a data set name.

Db2 subsystem name
Enter a valid subsystem ID (SSID) identifying the Db2 subsystem against which the batch SQL will run. The Log Analysis Tool plans must be bound at this SSID. It does not have to be the same SSID against which the SQL was generated. In other words, you can be performing data replication, but the tables named in the SQL must exist at this SSID.

LE runtime loadlibs
(*Optional*) Enter the fully qualified name (no quotes) of your Language Environment (LE) runtime load libraries. Put a single load library on each line. These libraries will likely be named CEE, or something similar, though your site has complete control over the naming convention of these libraries. These libraries are used by the C program handling the fast apply SQL process. Contact your systems programming staff if you do not know the names.

Restart
Enter one of the values:

A

AUTO mode. The restart program determines if there is a row in the restart table for this process and proceeds as follows:

- If there is not a row in the restart table, it creates one and starts processing SQLIN from the start of the file.
- If there is a row in the restart table and the row indicates that the last run was a success, the restart program starts processing SQLIN from the start of the file.
- If there is a row in the restart table and the row indicates that the last run failed, the restart program starts processing SQLIN at the statement following the last successful commit.

I

IGNORE mode. No restart is possible, and the restart table and process name are ignored.

S

STARTOVER mode. The restart program determines if there is a row in the restart table for this process name and proceeds as follows:

- If there is not a row, the restart program creates one and starts processing SQLIN from the start of the file.
- If there is a row, the restart program resets the row and starts processing SQLIN from the start of the file.

N

NEW mode. The restart program assumes that this is a totally new process, and creates a new entry for the process name in the restart table. If the process name already exists, the program terminates.

M

NUMBER mode. The restart program starts processing SQL at the specified SQL statement number.

Restart name

This is the process name used to associate different runs of the restart program. It must be provided if restart is desired. Use an eight- character unique name to identify the run. If you re-use this name, be sure that the previous run (with the same restart name) ran successfully, or else the SQL will be started at the point of failure from the previous run. If you are not sure of the results of the previous run, either change the restart name or set **Restart field** to **S**, so it is treated as a new run from the beginning of the SQL.

Restart SQL number

This is the SQL statement number to start processing from if you set Restart=M. It is otherwise ignored.

Commit Frequency

Enter a value from 1 to 99999. This controls how frequently commits are performed. It is used in conjunction with the Commit Type parameter.

Commit Type

Enter one of the values described following:

T

Transaction. This refers to the COMMIT statements in the SQL file.

- If the Commit Frequency is 1 and the Commit Type is Transaction, a commit is performed for every commit statement found in the SQL file.
- If the Commit Frequency is 10 and the Commit Type is Transaction, a commit is performed for every 10 commit statements found in the SQL file.

S

Statement. This refers to the executed Fast Apply SQL statements, not the actual SQL statements in the SQL file.

- If the Commit Frequency is 1 and the Commit Type is Statement, a commit is performed after each executed Fast Apply statement which might include multiple, combined SQL statements from the SQL file.
- If the Commit Frequency is 10 and the Commit Type is Statement, a commit is performed after every 10 executed Fast Apply statements, each of which might include multiple, combined SQL statements from the SQL file.

With this option, the actual commit statements in the SQL file are ignored.

When there are more than two inserts per commit in one statement, Fast Apply switches on multiple-row insert with the attributes `NOT ATOMIC CONTINUE ON SQLEXCEPTION` by default. To switch off the multi-row insert, add `,MRIOFF` to the PARM=SSID parameter in the JCL EXEC statement.

Note: Adding the MRIOFF parameter helps prevent issues that might be caused by multi-row insert, but requires more CPU usage.

Loglevel

Enter a value from 1 to 9. This controls the level of messages written to the ALAPRINT DD data set. Leave as the default (1) unless directed otherwise for problem analysis.

Bypass Multi-Row Inserts

The multi-row Inserts are used by Fast Apply to improve performance. This option prevents the use of multi-row inserts when Fast Apply runs. Enter 'Y' to bypass Multi-Row Inserts processing.

Table filters

Table filters allow for tables found in the SQL to be excluded and ignored or for execution of SQL only for specific tables. Enter a Y if you want to use table filters.

Restriction: If you are converting a table name from its value in the SQL file to something different and specify the table on the filter panel, you must specify the target name.

On xxxxx fields

For all of the ON xxxxx fields listed here, these values and rules apply:

S
STOP after an event. It will rollback to the previous commit point, and stop all processing.

W
Issue a WARNING after an event, but continue processing.

I
IGNORE an event, and continue processing.

R
ROLLBACK after an event. It will rollback to a previous commit point, but then continue processing with the next SQL statement.

The precedence rules for the ON xxxxx fields are as follows:

1. ON SQLCODEs setting
2. ON SQLERROR setting
3. ON Multiple rows setting
4. ON SQLWARN setting

ON SQLERROR
Enter one of the described values. The event in this case is any SQL error return code.

ON SQLWARN
Enter one of the described values. The event in this case is any SQL warning return code.

ON Multiple rows
Enter one of the described values. The event in this case is any SQL statement which causes modifications to more than one row.

ON SQLCODEs
Use this line to enter up to 10 different SQL codes for which you want to control the specific action to take. For example, if you want SQL code +100 to be ignored, you can specify that here. The format of this input is shown here, and must be copied exactly. If you need more than 10 SQL codes to handle specifically, edit the JCL in DD ALACFG ensuring that you follow that format.

- The format for each entry is: `<sign><code>,<action>;`
- The `<sign>` can be + or -. It must be present.
- The `<code>` can be any 3 digit SQL condition code (warning or error).
- The `<action>` is as described (S/W/I/R), and the event is the listed SQL condition code.

Examples:

```
+100,I;
```

Ignore SQL code = +100

```
+100,I;-511,R;-904,I;
```

```
Ignore SQL code = +100, rollback on SQL code =-511, ignore SQL
 code = -904
```

Important: These codes are not intended to have any meaning or usefulness for your specifications; they are examples only.

Execute dynamic SQL in batch: Fast Apply table filters
When you specify Y in the Table filters option on the "Execute dynamic SQL in batch: Fast Apply" panel, you are identifying that you want either the SQL to be excluded and ignored for specific tables or for the SQL to be run only for specific tables.

When you type Y in the Table filters field and press **Enter**, the "Fast Apply Table Filters" panel is displayed:

```
V3.5.0 ---------------- Fast Apply Table Filters ---------------- SC01/SS1A
COMMAND ===>

Hit 'Enter' to save current filters; 'PF3' to bypass use of these filters

*Exclude tables ..... N (Y/N)

*Table Owner                          *Table Name
----------------------------------    ---------------------------------------
----------------------------------    ---------------------------------------
```

Figure 49. Execute dynamic SQL in batch: Fast Apply Table Filters panel (top half of panel)

Fast Apply Table Filters fields

Exclude Tables
Identify whether you want to exclude the tables that you are specifying on the panel. If you specify **Y** to exclude tables, any table specified is ignored when processing the SQL file. All other tables are included and processed. If you specify **N** to not exclude tables, only the specified table is processed. All other tables are excluded and ignored.

Table Owner
Identifies the table owner.

Table Name
Identifies the table name.

Source and Target tables

If you want to run REDO and UNDO SQL on a different set of tables from those referenced in the original SQL, you can do so using the "Fast Apply SQL" option. On the panel titled "Execute dynamic SQL in batch: Fast Apply" a set of fields allow you to specify those different tables.

To specify new values for the table names in the SQL:

1. Specify the table owner and name that appear in the generated SQL that need to be converted. You type this information under the column titled "Source Tables".

2. Specify the replacement values for table owner and name. You type this information under the column titled "Target Tables" on the same row as the corresponding source table.

Entering no tables will lead to the SQL being executed as is.

In Db2, a table owner and name can be up to 128 characters long. On this panel the table owner fields are ten characters long. The table name fields are 25 characters long. If any field is full when you press PF3 a dialog will pop allowing for the entry of up to 128 total characters.

The ISPF panel allows for five table conversions to be specified at one time. If more table conversions are needed, you must manually edit the JCL. Within the Log Analysis-generated JCL ALACFG DD statement, you must insert the following syntax for each additional table conversion:

```
CONVERT
  SOURCE TABLE OWNER owner
  SOURCE TABLE NAME  name
  TARGET TABLE OWNER owner
  TARGET TABLE NAME  name
```

The Fast Apply process is unlike the rest of Log Analysis Tool in that it uses C code (rather than Assembler code), and acts as a separate, distinct function. The messages and codes originating from this process are generated by C function code and LE runtime code. These messages generated from this process are not included in the Message Help Facility. See the DD statement ALAPRINT for informative messages about the success (or failure) of the execution of Fast Apply SQL. The C runtime libraries (LE) may also generate multiple job logs (of similar names) when failures occur. For example, if a job named USERIDX1 was run,

using SDSF (or a similar product to view job log output) may reveal multiple USERIDX1 job log outputs. All of these are needed in handling diagnostics for failed runs.

Memory use in Fast Apply

There are two factors that determine the amount of memory used by program ALA@FSP, the Fast Apply SQL processor:

The first factor is the minimum memory that program ALA@FSP uses for every batch job. Things that affect the minimum amount of memory are values such as system settings, operating system settings, Db2 settings, and ALA@FSP configuration variables. To view the minimum amount of memory needed for program ALA@FSP, run a batch job with an empty SQL file. Values being used to calculate the minimum amount of memory will be displayed in the job output. The following example illustrates memory requirements. These values are examples only; your values will differ:

```
--------Below 16Meg--------          --------Above 16Meg--------     -
- Private Area        9192K          Private Area       1485824K     -
- Max Allocated       1268K          Max Allocated        17568K     -
- LSQA And SWA         352K          LSQA And SWA         10276K     -
```

The second factor is the total column lengths for each unique SQL statement being executed. When an SQL statement is executed, all of the column values are stored in blocks of memory that are equal to the size of each column length. For example, a column defined as INTEGER would need 4 bytes of memory and a column defined as CHAR(8) would need 8 bytes of memory. Once an SQL statement is executed, it is cached and future executions of the same statement do not need extra memory, even if the values change. By adding up the memory each column would need for each unique SQL statement and adding it to the minimum memory requirement for program ALA@FSP, you can calculate the memory needed when executing program ALA@FSP.

Requirements for the SQL Input

Log Analysis Tool is used to execute the SQL generated by the detail report run. Input created or updated outside the tool is not supported and can cause program errors. You can use a standard format for a SQL comment. The input must be a fixed length 80 byte dataset. For multiline inputs or comments use a continuation symbol on the column 73.

List of supported SQL statements:

- **INSERT**: adds a new row in a table.
- **UPDATE**: edits the existing data in a table.
- **DELETE**: deletes one or more rows from a table.
- **COMMIT**: ends the unit of recovery in which it is executed.
- **SET CURRENT SQLID**: assigns a value to the CURRENT SQLID special register.
- **SET SYSIBMADM.REPLICATION_OVERRIDE**: indicates if Db2 allows an application to specify values for generated expression columns, including temporal table GENERATED ALWAYS columns.

List of supported SQL special register:

- **DEFAULT**: shows the default value of a column.
- **CURRENT_DATE**: shows a date based on the time-of-day clock.
- **CURRENT_TIME**: shows a time based on the time-of-day clock.
- **CURRENT_TIMESTAMP**: shows a time stamp based on the time-of-day clock.

To process the above special registers successfully, ensure you use underscore(_). For example: CURRENT_DATE. Any other operations on the listed special registers, for example CURRENT_DATE+1 are not supported and can cause a program error.

Note: Make sure you do not use any other statement or special register in the SQLIN file to avoid parsing error or program failure.

Chapter 10. Process archived output

About this task

Processing archived output allows for running detail level reports using either previously archived output, or the last detail report output, as input to another detail report run. The primary purpose of this option is to let you re-run a previously run detail report without incurring any of the overhead of that run. You can now save detail report output into an archive and generate both detail level reports and/or SQL from it at a later date. It also lets you add more filters to further narrow down the previously saved result set.

The layout of the archive output data set is described in the topic "Archive output data set description" on page 411.

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **6**, **Process archived output**.

   The Process archived output panel appears.

```
V3.5.0 ----------------- Process archived output ---------------- SC01/SS1A
COMMAND.....
                                                        More:     +

*DB2 subsystem name..... SS01   (SSID)
*Action................. E      (E - Edit, S - Submit)
*Generate Undo SQL...... N      (Y - Yes, N - No,
                                 X - Yes, exclude triggered actions)
*Generate Redo SQL...... N      (Y - Yes, N - No,
                                 X - Yes, exclude triggered actions)
*Commit scope........... 000    (000-999, UOW)
*WHERE clause by key.... Y      (Y/N)
 Set Current SQLID......
 Job Identifier......... PDUSER
*Output flags........... D      (B - Bypass reports, D - Detail,
                                 T - Transaction, C - Catalog Audit)
 Report row length...... 00080 (00080-32752)
*Show varchar length.... N      (Y - Yes, N - No)
 Only changed columns... N      (Y - Yes, N - No)

Archive input and range:
 Archive table creator=>
 Archive table name...=>
 Start/End Date-Time.... 2007/01/11 - 00:00:00 / 2007/09/01 - 00:00:00
*Override GMT offset.... N       (Y/N)
  with this GMT offset.. +00:00
*Current data only...... N      (Y/N)
```

Figure 50. Process archived output panel

2. Enter the required information and press **Enter**.

   Log Analysis Tool runs the report.

   - **Process archived output panel fields:**

      Most of the fields on the Process archived output panel are the same as the fields on the general and details report panels. The different fields are described here. See Chapter 7, "Running reports ," on page 85 for those fields not described here.

      Archive table creator/Archive table name
      These fields identify the archive table (a Log Analysis Tool table which is very similar in purpose to the Db2 supplied BSDS) to be used to determine the needed archive data sets. This table must have been previously specified in a detail report run. The field is available here primarily if you have multiple archive tables, but will always be pre-filled with your most recently used table during an archive output detail report run. This product currently allows a maximum of 50 characters for both table creator and table name.

Start Date/Start Time/End Date/End Time

Enter these fields in the format indicated on the panel. The archives will be searched for any activity found between these date/time values. These fields will be set to their default formats if blanked out. The default formats are considered equivalent to spaces internally, so they do not need to be blanked out if using current data only. If the Current data only field is set to Y, that overrides these values, and they are ignored.

Current data only

Type Y to use input from your last detail report run only. If set to Y, it is not necessary to have previously archived the output from your last detail report run. Type N to use archived output only, and if so, the archive input and range fields must be non-blank. Setting this field to Y is a quick way to re-run your last detail report job without restarting from scratch (no Db2 log reads are performed in this re-run). When set to Y, the start/end date-time fields are ignored (you do not have to type spaces to blank them out).

Chapter 11. Using utilities

This topic provides information on using Log Analysis Tool utilities.

Utilities panel

```
-------------------- DB2 LOG ANALYSIS TOOL - Utilities ----------------------
 Option ===>

   0. View LOB/XML data
   1. Convert Timestamp to LRSN
   2. Convert LRSN to Timestamp
   3. Table Versioning Support
   4. Mode File Operations
   5. APAR report
   6. Application-level Relationship
Discovery
```

Viewing LOB/XML data

About this task

LOB and XML data types are supported for viewing with limited support for other operations. LOB data can be viewed, included in SQL operations, and included in LOAD operations if using Db2 V9 or later. XML data can be viewed and used in SQL, and in LOAD operations. XML data issues are only relevant for Db2 V9 and later. See Chapter 20, "LOB/XML support in SQL and LOAD operations," on page 247 for information on operations other than viewing.

Important: In the following text, the term "LOB" is used to refer to either a LOB or XML column. There are certainly many differences in these two data types, however, they are both tracked by this product in a similar way, and both are effectively auxiliary tables to the base table containing their column definitions. Therefore, the use of the term "LOB" is just this product's denotation of these types of objects and how they are processed. When differences between LOB and XML data types become relevant, they are noted.

Before you can view any LOB column data, you must run a general/detail report which captures the relevant LOB data. The general/detail report runs must have included LOB data, or else no LOB column data is available afterward. You can only view one column of LOB data at a time. LOB column data is not materialized ahead of time, to prevent unnecessary processing. LOB data can be very large, and therefore cannot be treated in the same manner as non-LOB column data. View LOB column data only as needed; it is highly unlikely you need to view all LOB column data, but individual LOB column(s) of certain log actions may warrant such a view.

The detail report itself contains information that is required in order to view LOB data. Therefore, you must create a detail report in order to subsequently view such data. If you omit the detail report itself, you will not be able to identify the particular action containing the LOB data. So always include the detail report output in the detail report run if you think you may want to view the LOB data.

Viewing LOB data requires that you extract certain information from the detail report to be able to pinpoint the specific LOB data you wish to view. You must first materialize a LOB column's data before you can browse it. The materialization process is a batch job that will write the needed LOB column data to a file. That file can subsequently be browsed. If you wish to view this data over and over, the file is not deleted or overwritten until you request a different LOB column instance for materialization. The browse process uses standard ISPF browse mode.

The file containing the LOB column data has an initial record which identifies the name of the LOB column to which the rest of the data belongs. This can be helpful if you erroneously attempt to browse a new LOB column before materializing it.

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **7**, **Utilities**, and then option **0**, **View LOB/XML data**.

   The View LOB/XML data panel is displayed.

   ```
   V3.5.0 ------------------ View LOB/XML data -------------------- SC01/SS1A
   COMMAND ===>


   *LOB/XML Action...........===> M        (M - Materialize, B - Browse)
   *Action..................===> E        (E - Edit, S - Submit)
    Job Identifier...........===>

   Identify LOB/XML column to report on:
   *Record identifier........===> 1
   *Column name..............===>
   *Update-type data.........===>         (P - Pre-update, O - Post-update)
   ```

   Figure 51. View LOB/XML data panel

2. Enter the required information and press **Enter**.

   Log Analysis Tool performs the requested task.

   - **View LOB/XML data fields:**

     LOB/XML Action
       Specify whether you want to materialize or browse the LOB column data.

       – Type M to materialize a LOB column. LOB column data must be materialized prior to browsing.

         It is possible to receive a return code = 8 when materializing the LOB data if your input is not correct. If you receive a bad return code from the materialization job, check that your input is precise before reporting the problem. Several issues can cause a return code = 8. If the LOB column value is NULL, you will get a return code = 8. You can identify this if the LOB locator value in the detail report starts with an '8', which indicates a NULL LOB value. Do not attempt to materialize such a LOB. Also, if the LOB column did not change on an update operation, its value will not be available. Only changed LOB values are tracked, because Db2 does not log non-modified LOB values on an update.

         If a return code = 8 (or higher) is generated from the materialization job, the generated LOBC file (to contain LOB contents) would also contain the error message from the failing job. So you can browse the LOB data even after such a failure just to see the error message.

       – Type B to browse a previously materialized LOB column. When you request a browse, all other panel fields are ignored and you are placed into browse mode on the most recently materialized LOB column. An example of browse output:

         ```
         *************************** Top of Data**********************
         LOB COLUMN NAME=C4
         TEST LOB DATA
         ************************** Bottom of Data********************
         ```

         The first line identifies the LOB column, and the second line contains the LOB data. In this example the LOB data is "TEST LOB DATA".

     Action
       Type E to edit the generated JCL, or S to submit the JCL directly. This field is only relevant if "Action = M", as only that setting generates JCL.

     Job Identifier
       This optional field is an 8 character maximum field that can be used to identify data sets associated with a given run of the created JCL. It will be used in the formation of the data set names used in the JCL. For example, if set to TEST01, the MODEFILE will be named USERID.MODEFILE.TEST01. This value must correspond to a value previously used in a detail

report run. If left blank on your associated detail report, it must be left blank here. The LOB column data will be written to a file using this field. For example, if the PLOGD file generated in your detail report had this naming convention: USERID.PLOGD.TEST, then the LOB file will be created as USERID.LOBC.TEST. This means you used a job identifier of TEST. If your detail report created a PLOGD file named DBAX.ABC.PLOGD.JAN05, then your job identifier is JAN05, and must be used here as well. This also indicates a data set prefix of DBAX.ABC (as seen in your "User settings" panel option). The product will create the LOB file as DBAX.ABC.LOBC.JAN05. Whatever file name is used by this product in generating the LOB file can be seen once in browse mode (or by examining the output of the batch job to materialize the LOB file).

- **Identify LOB column to report on:**

  These fields identify the specific LOB column you want to view. You must use the detail report output to gather this information.

  Record identifier
  Specify the value of the record identifier from the detail report. This record identifier pinpoints the exact log activity reported on for which you wish to materialize the LOB column data. This implies that you must create the detail report in order to view LOB data. If you omit the detail report, there will be no way to know the record identifier value.

  Column name
  Specify the name of the LOB column containing the data you want to view. This is required because a table can contain more than one LOB column.

  Update-type data
  If the action associated with this record number is an update, you must identify to the product whether you want to see the column data before the update occurred, or after the update occurred.

  - Type P to get the pre-update image of the LOB column data.
  - Type 0 to get the post-update image of the LOB column.

  If the record identifier represents a non-update (that is, an insert or a delete), this field is ignored. If the record identifier represents an update, but this field is left blank, the post-update image is the default.

Converting a timestamp to an LRSN

If you need to determine the LRSN that corresponds to a timestamp, you can provide the timestamp and Log Analysis Tool converts it to an LRSN.

About this task

The Convert Timestamp to LRSN panel also can be invoked from the Generate Database Activity Report (General) panel. In this case, converted values are passed to the **Start/End LRSN** fields on Generate Database Activity Report panel.

For details about the fields on the Convert Timestamp to LRSN panel, refer to the online help or online tutorial.

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **7**, **Utilities**, and then option **2**, **Convert Timestamp to LRSN**.

   The Convert Timestamp to LRSN panel is displayed.

```
V3.5.0 ---------------- CONVERT TIMESTAMP TO LRSN ---------------- LPR1/SS01
Command ===>

Hit 'Enter' after entering Timestamp to convert to LRSN
Hit 'PF3' to EXIT


Input parameters:
 Timestamp.............. YYYY/MM/DD-HH:MM:SS.MMMMMM

 STCK to LRSN Delta..... 00000000000000000000

*Override GMT offset.... N        (Y/N)
  with this GMT offset.. +00:00

Converted LRSN:
 LRSN..................

.--------------------------------------------------------------------.
| ALAA736I - Timestamp to LRSN translation is available from         |
| "Generate Database Activity Report (General)" ISPF panel with pressing |
| PF06 key or entering XTSTAMP command                               |
'--------------------------------------------------------------------'
```

Figure 52. Convert Timestamp to LRSN panel

2. Enter the timestamp that you want to convert in the **Timestamp** field.

   Enter the timestamp in the format YYYY/MM/DD-HH:MM:SS.MMMMMM.

3. Optional: If an LRSN delta value has been applied, enter the delta value in the **STCK to LRSN Delta value** field.

   If you enter a value in this field, it will be added to the timestamp value before conversion to the LRSN.

4. Optional: If you want to adjust the timestamp for a particular time zone, enter Y in the **Override GMT offset** field and enter the offset value in the **with this GMT offset** field.

   The default is the current GMT offset that is in use by the associated LPAR.

5. Press Enter.

   The LRSN value is calculated and is displayed in the **LRSN** field.

Converting an LRSN to a timestamp

If you provide an LRSN, Log Analysis Tool can convert it to a timestamp.

About this task

For details about the fields on the Convert LRSN to Timestamp panel, refer to the online help or online tutorial.

Procedure

1. From the Db2 Log Analysis Tool - Classic Menu, select option **7**, **Utilities**, and then option **2**, **Convert LRSN to timestamp**.

   The Convert LRSN to Timestamp panel is displayed.

```
V3.5.0 ---------------- CONVERT LRSN TO TIMESTAMP ---------------- LPR1/SS01
Command ===>

Hit 'Enter' after entering LRSN to convert to Timestamp
Hit 'PF3' to EXIT


Input parameters:
  LRSN..................

  STCK to LRSN Delta..... 00000000000000000000

*Override GMT offset.... N       (Y/N)
   with this GMT offset.. +00:00

Converted timestamp:
  Timestamp..............
```

Figure 53. Convert LRSN to Timestamp panel

2. Enter the LRSN that you want to convert in the **LRSN** field.

   Enter the LRSN in 10-byte extended format using only hexadecimal characters

3. Optional: If an LRSN delta value has been applied, enter the delta value in the **STCK to LRSN Delta value** field.

   If you enter a value in this field, it will be subtracted from the timestamp value before conversion to the LRSN.

4. Optional: If you want to adjust the timestamp value for a particular time zone, enter Y in the **Override GMT offset** field and enter the offset value in the **with this GMT offset** field.

   The default is the current GMT offset that is in use by the associated LPAR.

5. Press Enter.

   The timestamp value is calculated and is entered in the **Timestamp** field.

Table versioning support

This option involves two related concepts: table versioning and dropped columns. Both of these features alter a tables mapping based on a point in time. The Db2 table ALA_TABLE_VERSIONING is an internal repository, that is supplied to help support these concepts while the product runs.

First discussed is table versioning, and subsequently, dropped column support is discussed below.

DB2 V8 introduced the concept of altering a column's data type. There are only a limited number of column types that may actually be altered (see IBM DB2 V8 or higher SQL Reference). Each alteration of a table's column in this way creates a new 'version' of the table. DB2 provides very limited support for handling of this option, and in many cases this product will not be able to automatically handle this type of activity. This is not a limitation of this product, but rather a limitation imposed by DB2.

The best way to describe these limitations is by example. Suppose TABLEX currently has this format and has never been altered previously:

EMP_NUM INTEGER,

EMP_NAME VARCHAR(50),

EMP_SALARY DEC(5,2)

Suppose further that the table is now altered changing column EMP_SALARY to DEC(7,2). If no modifications (updates, inserts or deletes) are subsequently done to this table, DB2 does not provide any mechanism to determine the state of the table prior to the above alter. Therefore, this product would have no way of determining the actual format of rows that were modified prior to this alter action.

If modifications are subsequently done to this table, DB2 will write SYSOBD information into the current tablespace. The SYSOBD information allows this product to determine different table versions, or row formats. However, if a REORG occurs, all this versioning information (SYSOBDs) is wiped out by DB2,

leaving only the current version as SYSOBD information. Again, this product may not be able to resolve row formats for activity prior to this REORG. The safest way to proceed here is to REORG your tablespace immediately after such an ALTER, and then do NOT expect to track any data prior to the ALTER.

These are just a few cases where this product will fail in resolving proper row formats given the existence of table versioning. It attempts to resolve the SYSOBD information where available, but if any given table row cannot be formatted, this will result in an error and the program will stop.

Given this, it is strongly recommended that table versioning not be done. However, if it is necessary, always keep track of each version, including the original version of your tables. Without this knowledge, this product may be completely incapable of processing data against such tables.

Note that the ability of the product to automatically retrieve versioning data from DB2 is dependent on the whether log-backward or log-forward is used. If log-backward is used, the versioning data must be found in the current tablespace; if log-forward is used, the versioning data must be found in the image copy(s) accessed. If the versioning data cannot be resolved in these ways, then an error message will be displayed and user intervention will be required.

Given the example above, you would track that TABLEX had an original version whereby column EMP_SALARY was defined as DEC(5,2), and it's first version change was to DEC(7,2). This would be considered as two versions: version 0 (the original format) is DEC(5,2), version 1 is DEC(7,2). The important information to track is the version number, the column number that changed, the column type it changed from, and the change in length and/or scale. With this information available, you will always be able to provide this product with the data it needs to resolve all row formats. Manual intervention will be needed if the product cannot determine this information from available SYSOBD data. See below for a complete description on how to input this information if needed by the product.

Table versioning can mean that changes to a row result in DB2 logging of two different row formats. For example, if an update is done to a row immediately after the alter, the DB2 log record for this update will show a pre-row image reflecting the older format of the row and the post-row image reflecting the new format of the row. Given that the versioning information is available, the product will convert each row format appropriate to its version.

If you are loading output of this product back into DB2, only one format can be generated into the loadfile. That format will always be the current format. Therefore, this product does convert the old formats into the current formats where necessary. If you changed the format of the source table, the target table must also be modified the same exact way or the load will fail.

DB2 V11 introduced the concept of dropping a column. Though the net result of dropping a column is very similar to table versioning, it requires some unique information. DB2 does not provide any trail of a dropped column action, so it must be manually entered in this file if any job run incurs table activity that spans the old and new formats of the table. Without this data, the product cannot successfully map both formats of the table. Though the DB2 catalog table SYSCOPY provides a record of when the drop column was materialized for the table (example after a REORG), it does not record such information as column number, type, length, etc. Most of the above information regards table versioning is relevant for dropped column data as well. The difference is that field 'Table Version' is not utilized on a dropped column entry, and fields 'Alter Table Logpoint' and 'Nulls' are used only for a dropped column entry. The 'Alter Table Logpoint' field will likely originate from the START RBA field on the appropriate record in SYSIBM.SYSCOPY (though it must be entered here in hex display format, i.e., as 20 character bytes rather than 10 packed bytes).

Managing table versioning entries

Using the Manage Table Versioning Entries panel, you can add, delete, import, and export the table versioning entries. You can also view the versioning entries using different filters.

```
V3.5.0 -------- Manage Table Versioning Entries -------- RS22/DC1A
Command ===>                                            Scroll ===> CSR

Commands: NEW  DELETE  AUTODEL  IMPORT  EXPORT  TRACK

Line Commands: D - Delete    V - View full row

Database like: %              Tablespace like: %
Table Owner like: %
Table Name like:  %
Date-Time range:  YYYY/MM/DD - HH:MM:SS / YYYY/MM/DD - HH:MM:SS

                              Row 1 of 1      +
-----------------------------------------------------------------------------
  Table Owner  Table Name   Ver Drop Col Column   Column Timestamp
                                    Num Type     Length
- -----------> -----------> --- ---- --- -------- ------ --------------------
```

Commands:

- NEW: Add a new entry of versioning or dropped column information. If the entry with the same DBID, PSID, OBID, VERSION, COLUMN_NUMBER is already available, it is updated with the specified data.
- DELETE: Delete all entries displayed using the specified filters. To undo this action, generate the UNDO SQL for the versioning table.
- AUTODEL: Use this command to delete versioning entries for the following tables:
  - tables that don't exist anymore.
  - tables where versioning happened before the recreation of the table.
- IMPORT: Use this command to fetch records from a TVFILE. In case the entry with the same DBID, PSID, OBID, VERSION, COLUMN_NUMBER already exists, it is updated with the specified data.
- EXPORT: Save all the entries displayed using the specified filters to a TVFILE.
- REFRESH: Refresh the displayed entries using the specified filters.
- BROWSE: View all the table versioning entries in full rows using horizontal and vertical scrolling.
- TRACK: Start the Track Schema Changes panel to enable automatic table versioning support.

Line commands:

- D: Delete versioning record.
- V: View the full row of the versioning record.

Filters:

- Database like: Using Database like, you can filter the list of table versioning entries by the database name. You can enter a full or partial database name. Use a wildcard character (%) to specify a part of the name.
- Tablespace like: Enter a full or partial tablespace name to filter the list of table versioning entries by the tablespace name. You can use the wildcard character (%) to specify part of the name.
- Table Owner like: Enter a full or partial table owner name to filter the list of table versioning entries by the table owner. You can use the wildcard character (%) to specify part of the name.
- Table Name like: Enter a full or partial table name to filter the list of table versioning entries by the table name. You can use the wildcard character (%) to specify part of the name.
- Date-Time range: Enter the date and time range to filter the list of table versioning entries by timestamp.

Displayed columns:

- Table Owner: The OWNER of the ALTERed table (from SYSIBM.SYSTABLES).

- Table Name: The NAME of the ALTERed table (from SYSIBM.SYSTABLES).
- Ver: The version of the data row format for the table.
- Drop: The dropped flag indicates whether the column has been dropped.
- Col Num: The number of column that has been versioned or dropped.
- Column type: The data type of the column that has been versioned or dropped.
- Column length: The length attribute of the column that has been versioned or dropped. If the column type is DECIMAL, then represents precision.
- Timestamp: The timestamp when the ALTER action has occurred. This value can be greater than the date and time when the row was inserted into the SYSCOPY table, but it is never lower.

Resolving 'ALAA486E - Versioning data not available for table'

An error message ALAA486E is displayed when the versioning information is not resolved automatically. To resolve this, add the corresponding table versioning entry into the product repository. The message contains object identifiers of the versioned table and a version number that cannot be resolved. Use the 'Table Versioning Support' panel from the 'Utilities' menu to manage table versioning entries. There are following three ways to specify new entries when using the product's versioning feature.

- Manually add a new entry by using the **NEW** command. This way you can enter the details of the new entry directly into the product repository.
- Use the **TRACK** command to generate a 'Track Schema Changes' job. This job automatically identifies any schema changes that has occurred within a specified time range and updates the product repository with the new entries. This is a convenient and time saving approach that eliminates the need for manual entry.
- Use the **IMPORT** command to import all versioning records from an existing TVFILE into an internal table.

Adding new table versioning entries

The Add New Table Versioning Entries panel is displayed when the **NEW** command is specified in the 'Manage Table Versioning Entries' panel. Using this panel, you can manually add a new entry of versioning or dropped column information.

```
V3.5.0 ------- Add New Table Versioning Entries ------- RS22/DC1A
Command ===>

Hit 'Enter' to save the information; 'PF3' to exit without saving.

All IDs should be entered with their decimal value.

*DBID   *PSID  *OBID  *Table Version  Drop Flag (Y/N)  Alter Table Logpoint
------  -----  -----  -------------   --------------   -----------------

*Column Nbr   *Column Type   *Column Len   Scale (if needed) Nulls (Y/N)
-----------   ------------   -----------   ----------------- -----------
```

Minimum one row is required for an entry. To add an entry, press the Enter key after all information is entered. An entry must be either a table versioning or a dropped column entry. It cannot be a combination of both. After you are done editing, press PF3 to exit.

Panel fields:

- **DBID**: Db2 object identifier of the database that contains the table that has been modified. Enter the associated decimal value. This information is displayed in the error message ALAA486E. This is a mandatory field.
- **PSID**: Db2 object identifier of the tablespace that contains the table that has been modified. Enter the associated decimal value. This information is displayed in the error message ALAA486E. This is a mandatory field.

- **OBID**: Db2 object identifier of the table that has been modified. Enter the associated decimal value. This information is displayed in the error message ALAA486E. This is a mandatory field.

- **Table Version**: The version of the table required to successfully perform detail report. Enter the associated decimal value. This information is displayed in the error message ALAA486E. This is a mandatory field.

- **Drop Flag**: Indicates if the entry represents DROP COLUMN versioning. This is an optional field. Default value is N.

- **Alter Table Logpoint**: Select this value from the SYSCOPY table where the ICTYPE field = 'A' on the tablespace. The value is available in the field START RBA, and represents either the RBA or LRSN, when a Db2 utility materialized the column drop or alteration. Specify this value as 20 character hexadecimal (display format) of the START RBA field. Logpoint specification is required only when the 'Drop flag' is set to 'Y' else it can be entered and saved for the associated ALTER COLUMN versioning.

To add an entry, minimum one row with the following column information is required:

- **Column Nbr**: The column number of the column that has been versioned or dropped. You can find it in SYSIBM.SYSCOLUMNS (COLNO), assuming no columns were dropped. By this restriction, the current value there would be the correct value which you can include. Enter the column number in the decimal form. If any columns of the table were also dropped, or if this is a drop column entry itself, the column number entered must be the column number at the time of the action.

    Note: A dropped column is not effective until a subsequent REORG so relative column numbers remain intact until that time. For example, a table has 4 columns. Column 2 is dropped, and column 3 is altered. And then a REORG is performed, making the drop column action effective for the table. The column number to enter for the drop action would be 2, and the column number to enter for the versioning action would be 3 and not 2, since the drop was not effective at the time of the alter or versioning action.

- **Column Type**: The data type of the column that has been versioned or dropped. For versioning, this information is available in SYSIBM.SYSCOLUMNS (COLTYPE). However, the current value there may not be the correct value. For example, if you changed a DEC(5,2) column to DEC(9,2), the COLTYPE will still be DECIMAL. But if you change an INTEGER column to a DEC(15,0) column, the current value in SYSCOLUMNS will be DECIMAL, but the value required for versioning is INTEGER. You must include the COLTYPE of the missing versioned information, which may or may not equal the current column type. For dropped columns, you must know the column type. It will not be available in the catalog after the drop is effective.

- **Column Len**: The length of the column, except the decimal data type. For decimal data type, enter the decimal digits. You can find this information in the SYSIBM.SYSCOLUMNS (LENGTH). The current value there may not be the correct value. For example, if you change a DEC(5,2) column to DEC(9,2), the LENGTH currently in SYSCOLUMNS will not be the correct length to include. You must include the LENGTH value of the missing versioned information, which may or may not equal the current length value. This length value must be equal to the value that was there in SYSCOLUMNS during the particular version of the column data type. For dropped columns, you must know the column length. It will not be available in the catalog after the drop is effective. To provide information for an ALTER ADD COLUMN scenario (when you are using Db2 V9 or later and a reordered row format table), set the column length field to zero for any version that did not contain the subsequently added column. See the tutorial topic on 'Table versioning Support' for more information.

- **Column Scale**: The column scale when the data type is decimal, Db2 VX timestamp, or timestamp with time zone, else leave the field blank. This is an optional field. Use decimal digits to enter a scale. For versioning, this information is available in SYSIBM.SYSCOLUMNS (SCALE). The current value there may not be the correct value to include here. For example, if you changed a DEC(5,2) column to DEC(9,4), the SCALE currently in SYSCOLUMNS will not be the correct scale to include. You must include the SCALE value of the missing versioned information, which may or may not be equal to the current scale value. For dropped columns, you must know the column scale. It will not be available in the catalog after the drop is effective.

- **Nulls**: This field is required only if you want to enter the DROP COLUMN information, else leave it blank. If dropped column was null, enter Y, else enter N. If the entry is not a DROP COLUMN entry, the value entered is ignored.

Tracking schema changes for automatic table versioning

When one or more schema changes are made to a table by using ALTER TABLE, Db2 creates a version of the table space (and the corresponding table). Table versioning changes a table's mapping based on a point in time, and can cause Log Analysis Tool to fail if Log Analysis Tool encounters a version change when processing logs. You can manually enter schema changes into the product repository. Or, Log Analysis Tool can generate a job to track schema changes over a period of time, and can update the product repository. Running this job regularly enables automatic table versioning support.

About this task

The job that tracks schema changes analyzes activity in several catalog tables (SYSIBM.SYSTABLES and SYSIBM.SYSCOLUMNS), and detects table version changes during a specified period of time.

Procedure

1. In the **DB2 subsystem name** field, enter the Db2 SSID.
2. In the **Action** field, enter E to edit the job before submission, or S to automatically submit the job.
3. In the **Job Identifier** field, enter a job identifier. You should use a job ID that differs from the usual Log Analysis Tool run, so that no data sets are overwritten during the run of automatic table version support job.
4. Enter one of the following sets of fields to provide the range of log activity to be read. At least one range must be provided, and the ranges are mutually exclusive. Refer to the help panel for specific formats and options for specifying these fields.

   - **Start/End Date-Time**
   - **Start/End RBA (URID)**
   - **Start/End LRSNs**

5. Optional: To run the job to track schema changes in continuous mode, enter the name of an existing continuous mode file.
6. Optional: If you want to adjust the timestamp value for a particular time zone, enter Y in the **Override GMT offset** field and enter the offset value in the **with this GMT offset** field.

   The default is the current GMT offset that is in use by the associated LPAR.
7. Press Enter.
8. On the Report Specifications Complete panel, enter Y and press Enter.

   Depending on the **Action** field setting, the job is either generated and displayed in an ISPF edit session, or is submitted.

What to do next

After the job to track schema changes successfully completes, if any table versioning was detected, the product repository will be updated with new entries. The 'Manage Table Versioning Entries' panel should be used to view those entries.

Mode file operations

The Modefile operators panel can be invoked from the "7. Utilities" menu item "4. Mode File Operations".

Mode file operations

```
V3.5.0 ----------------------- MODE FILE OPERATIONS --------------------- RS22/DC1A
Option ===>

 Hit 'Enter' after setting needed field to 'Y'
 Hit 'PF3' to EXIT

 Mode file DSN........... USER.MODEFILE
 Mode file report........................... N        (Y/N)
 Populate panels from mode file............. N        (Y/N)
 Convert mode file.......................... N        (Y/N)
 Create modeparm file....................... N        (Y/N)
```

These operations are to assist in the migration from a previous version of Log Analysis Tool or to assist in re-running a previous job where the details are not known.

Mode file DSN
Type the DSN of the mode file from a previous version of the Log Analysis Tool or from a previous run. The DSN must already exist. This field must be non-blank to use any of the following mode file operations.

Mode file report
Type Y to create a report of the contents from the given mode file DSN.

Populate panels from mode file
Type Y to read the given mode file DSN and populate the appropriate panels with the values contained in the mode file. If the mode file DSN entered is a version prior to release V3.3 then default values will be used for fields that have been added since the entered mode file DSN was created.

Convert mode file
Type Y to read the given mode file DSN and convert it to the version needed in the current release of Log Analysis Tool. This conversion will use default values for all new fields that have been added since the given mode file was created. The DSN of the converted file will be based upon the values found in the mode file being converted. The converted mode file DSN must not exist prior to the conversion. The DSN of the new mode file will be displayed after the conversion.

Create modeparm file
Type Y to read the given mode file DSN and create the modeparm file. This feature is only needed when migrating from a previous release. The modeparm file allows for easy modification of certain parameters without having to go back through all the ISPF panels. For additional information, see "Create the modeparm file" on page 77.

APAR report

The APAR report panel can be invoked from the "7. Utilities" menu item "5. APAR report".

APAR report

```
Maintenance information:
  APAR report ............................... N        (Y/N)
```

Select this option to create a report listing all the APARs that have been applied to the product. This information is displayed in the INFOM DD during batch product execution.

Chapter 12. Using profiles

This topic provides information on using Log Analysis Tool profiles to save and easily restore all options so that a report run can be recreated without having to remember what options had been specified. Without profiles, fields in the Log Analysis panels could be overwritten by unrelated jobs. Profiles make it easier to restore this data.

Important: To use profiles on a specific subsystem, the Db2 tables used to store the data must be setup. ALAPRDDL in the SALASAMP library must be edited and then run for each subsystem where you want to use profiles.

Accessing the main profile panel

This topic explains how to access the main profile panel.

When you type **8** to specify **Profiles** on the Log Analysis Tool main panel, a dialog box appears and asks you to specify a subsystem ID:

```
----------- Log Analysis Profiles -----------


   DB2 Subsystem ID?...... SS1A
```

Figure 54. Subsystem ID dialog box

Type in the subsystem ID on which you want to create the profile and press **Enter**. The Manage Log Analysis Profiles panel appears:

```
V3.5.0 -------------- Manage Log Analysis Profiles ------------- SC01/SS1A
COMMAND ===>                                          Scroll ===> PAGE


Line Commands: N=New     D=Delete    J=Generate JCL   P=Populate JSON
               E=Edit    R=Rename    M=Generate mode file
               C=Copy    X=Export    I=Import
               Z=Convert to classic

Profile Like  %                              Last Updated By Like  %
Creator Like  %
                              No rows to display
-----------------------------------------------------------------------------
                                         <------ Last Updated ------>
Cmd Profile Name                  Q/C Creator  Userid   Timestamp
    AAPROF1                        C   PDUSR2   PDUSR2   2011-07-08-12.58.24
    ADVANCE                        C   PDUSR2   PDUSR2   2011-06-12-16.42.50
    ADVSHOW                        C   PDUSR2   PDUSR2   2011-05-26-11.15.54
    ADVSHOW2                       C   PDUSR2   PDUSR2   2011-05-26-11.19.09
    ADV1                           C   PDUSR2   PDUSR2   2011-02-03-11.02.32
    ADV1A                          C   PDUSR2   PDUSR2   2011-03-10-13.52.46
    ADV1B                          C   PDUSR2   PDUSR2   2011-03-10-16.17.53
```

Figure 55. Manage Log Analysis Profiles panel

Manage Log Analysis Profiles line commands

The fields under the **Cmd** label are used to enter processing commands on a specific profile. Except for the delete (D) command, only one command can be entered at a time. The commands are defined following:

N

Creates a new profile. The N command can be entered on any of the profile command lines and whichever is used has no influence on the profile being created. If no profiles are displayed, a single

command line will be available for entering the N command. Another option for creating a new profile is to enter NEW on the command line.

E

Edits an existing profile. This option can be used to edit the profile name or any of the job settings.

C

Copies a profile when creating a new profile. Unlike with the N command, the C command will copy all the settings from the profile that is being referenced into the panels for the profile being created.

D

Deletes a profile. This is the one command that allows multiple command entries at one time.

R

Renames a profile. No other settings can be changed.

X

Exports a profile. The profile can be exported to a different subsystem on the current system or it can be exported to a file for use at a later time or on a different system. If a profile is exported to a file, it must be imported to be used.

J

Generates the JCL for the profile. The mode file is also generated with this command. This command also sets all panels and fields used in the profile to the profile's values throughout Log Analysis Tool.

M

Generates the mode file. Along with the mode file, the modeparm file will also be generated. After the generation is successful, the option to edit/submit related JCL will be available. The JCL must have been generated from the profile previously.

I

Imports a profile from a file. This is used to import a profile that was previously exported to a file.

Z

Converts to classic panel. Converts a quick start panel profile to a classic panel profile. This will open up all the options available within Log Analysis. Once this conversion is made, you cannot reverse it.

P

Populates the JSON parameters file with values saved in the profile and creates parameterized JCL. The parameters file contains the list of settings in a special format and can be easily edited, so the job can be re-run with new parameters without ISPF interaction.

Manage Log Analysis Profiles panel fields

Profile Like

Type a full or partial profile name to filter the list of profiles by name. The wildcard (%) can be used explicitly, but this is unnecessary. Anything that you type has an implied wildcard appended to the end.

Last Updated By Like

Type a full or partial authid to filter the list of profiles by the user who last edited the profile. The wildcard (%) can be used explicitly, but this is unnecessary. Anything that you type has an implied wildcard appended to the end.

Creator Like

Type a full or partial authid to filter the list of profiles by the profile creator. The wildcard (%) can be used explicitly, but this is unnecessary. Anything that you type has an implied wildcard appended to the end.

Display columns

The display columns are self-explanatory except for one:

Q/C

Identifies if a profile is a quick start profile or a classic profile. This value determines which panels are used when editing or copying a profile.

Creating a new profile

This topic explains the process for creating a new profile.

When you type N in the **Cmd** column or NEW on the command line, the "Create new profile" panel appears:

```
V3.5.0 ------------------- Create new profile ------------------- SC01/SS1A


*Profile type ===>      (1 - Start with general report,
                         2 - Start with detail report,
                         3 - Start with LOAD)
*Profile name ===>



COMMAND ===>
```

Figure 56. Create new profile panel

On this panel, you specify what type of profile you want to create and name the profile.

Create new profile panel fields

Profile type
Specify the profile type. This field determines the type of profile that you want to create. You can create a profile that handles general reporting, detail reporting, and loading of the data all in one profile or the steps can be broken up into individual profiles. The values are defined following:

1

The profile starts with the general report. Starting with a general report run allows detail reports and LOADS to be run in the same job, depending on which settings you select. Alternatively, the general report can be the only step for this profile. This allows any required detail reporting or LOADs to be in separate profiles.

2

The profile starts with the detail report. In order for a profile to be used for just a detail report, a corresponding general report profile must be created just prior or a mode file for a corresponding general report profile must be generated just prior to creating the detail report profile. This corresponding profile must share the same Job Identifier as well. The detail report profile uses output from the general report profile as input. Failure to create the general report profile first may lead to errors or unexpected behavior.

3

The profile starts with the loading of data. In order for a profile to be used for just loading data, a corresponding general report and/or detail report profile must be created just prior or a mode file for a corresponding general report and/or detail report must be generated just prior to creating the load profile. The corresponding profile must share the same Job Identifier as well. Failure to do this will lead to errors or unexpected behavior.

Profile name
You must specify a name for the profile. This field is a 30-character maximum field that is used to uniquely identify the profile to be created. The name can contain letters, numbers, and spaces.

When the new profile information is complete, press **Enter**.

The User Settings panel (option 0 on the main Log Analysis Tool panel, "Set user settings" on page 67) appears to allow you to create specific user settings for the profile. When you have finished the user settings and press **Enter**, the panel that corresponds to your "Profile type" (general report, detail report, or LOAD) appears.

When you finish entering all the information on the appropriate panels, you will see the JCL created by the profile.

You are then prompted to save the profile.

Editing a profile

This topic explains the process for editing a profile.

1. When you type E in the **Cmd** column, the "Create new profile" panel appears with the profile type and name already filled in, because you want to edit an existing profile.
2. When you finish entering all the information on the appropriate panels, you will see the JCL that is created by the profile.
3. You are then prompted to save the profile.

Copying a profile

This topic explains the process for copying a profile.

1. When you type C in the **Cmd** column, all of the Log Analysis Tool panels are populated using the selected profile's options and then the "Create new profile" panel appears so that you can specify the profile type and name.
2. When you finish entering all the information on the appropriate panels, you will see the JCL created by the profile.
3. You are then prompted to save the profile.

Deleting a profile

This topic explains the process for deleting a profile.

1. When you type **D** in the **Cmd** column and press **Enter**, the "Profile delete confirmation" dialog appears.

```
---------------- Profile delete confirmation -----------------

   Profile to be deleted: ADVSHOW2

   Delete profile now?... Y  (Y/N)
```

Figure 57. Profile delete confirmation dialog

Profile delete confirmation fields

Profile to be deleted
This is the name of the profile that you selected to be deleted.

Delete profile now

Y
Confirms the deletion of the profile specified.

N
Cancels the deletion. You may also use PF3 to exit the panel without deleting the specified profile.

2. When you press **Enter**, the profile is deleted or not, depending upon your specification. Once confirmed, the delete cannot be undone.

You can specify Delete for multiple profile names. You will get a "Profile delete confirmation" for each profile that you selected.

Renaming a profile

This topic explains the process for renaming a profile.

1. When you type **R** in the **Cmd** column and press **Enter**, the "Rename profile" dialog appears.

```
----------------- Rename profile ------------------

   Old Name... ADVSHOW2

   New Name...
```

Figure 58. Rename profile dialog

Rename profile fields

Old Name
This is the name of the profile that you selected to be renamed.

New Name
Type a 30-character maximum new profile name. The name must not previously exist.

To cancel the rename, exit the popup window.

2. When you press **Enter**, the profile is renamed.

Generating JCL from a profile

This topic explains the process for generating JCL from a profile.

1. When you type J in the **Cmd** column and press **Enter**, Log Analysis Tool generates the JCL for the selected profile. The mode file is also generated with this command. This command also sets all panels and fields used in the profile to the profile's values throughout Log Analysis.

2. You can make changes to the JCL, but nothing will be saved.

Generating the mode file for a profile

This topic explains the process for generating the mode file for a profile.

This is used in cases where you save the JCL on your own and want to run it. You must first ensure that the mode file data is correct.

1. When you type M in the **Cmd** column and press **Enter**, Log Analysis Tool generates the mode file for the selected profile and displays the "Mode File Successfully Generated" dialog.

```
-------------------- Mode File Successfully Generated ----------------------

   Selected profile: ADVSHOW2

  *Process related JCL... N   (Y/N)
   JCL PDS name..........
   Action...............
```

Figure 59. Mode File Successfully Generated dialog

Mode File Successfully Generated fields

Selected profile
This is the name of the profile that the mode file was generated for. It cannot be changed on this panel.

Process related JCL
Use this field to specify the desire to edit/submit JCL that has been previously saved. It is not uncommon to generate JCL in Log Analysis Tool and then to edit it to add any user specific pre-processing steps or post-processing steps. Entering Y in this field will edit/submit this JCL. It is important that the specified JCL was generated from the same profile that was used to generate the mode file. If the JCL is not related to the profile, then the resulting behavior is undefined.

JCL PDS name

This is the file name for the JCL that is to be edited or submitted. If "Y" is used for the processing of related JCL, this file name must be filled out. The JCL must already exist. A PDS with a member name can be supplied. The JCL must have been generated from the same profile that was used to generate the mode file or undefined behavior will occur.

Action

Type either an E or an S.

E

Display the JCL being referenced.

S

Submit the JCL being referenced.

Along with the mode file, the modeparm file is also generated.

2. When you press **Enter**, the JCL is either displayed or submitted, depending on your specification.

Exporting a profile

This topic explains the process for exporting a profile.

1. When you type X in the **Cmd** column and press **Enter**, the "Export Profile" dialog appears.

```
------------------------ Export Profile --------------------------

   Profile to be exported: ADVSHOW2

   Export type... S  (S - SSID, F - File)
   Target SSID...
   File name.....
```

Figure 60. Export Profile dialog

Export Profile fields

Profile to be exported

This is the name of the profile that the user selected to be exported. It cannot be changed on this panel.

Export type

Specify what type of export to use.

S

Export to a Db2 subsystem that is running on the current system. This option connects to the specified subsystem and copies the selected profile information over to the specified subsystem. A subsequent dialog asks you to name the exported profile on the new subsystem.

F

Export the profile to a file for later use or because a subsystem is on a different system. The selected profile's information is written out to the specified file.

Target SSID

This field must be populated when you specify option S in the Export type field. The SSID must identify a subsystem on the current system that has been defined in the Log Analysis Tool control file (option 11 on the main Log Analysis Tool panel).

File name

This field must be populated when you specify option F in the Export type field. This field identifies the name of the file that will be used. If the file does not exist, Log Analysis Tool will create it for you. If it does exist, it will be used. You can use a PDS with a member name, but Log Analysis Tool will create only the member, not the entire PDS. The attributes of the PDS must be: LRECL=140, BLKSIZE=14000, RECFM=FB. Do not use quotes around the specified file name; enter only a fully qualified data set name.

2. When you press **Enter**, the selected profile is exported.

Importing a profile

This topic explains the process for importing a profile.

1. When you type I in the **Cmd** column or IMPORT on the Command line and press **Enter**, the "Import Profile" dialog appears.

```
-------------------------- Import Profile --------------------------

New profile name...
File name..........
```

Figure 61. Import Profile dialog

Import Profile fields

New profile name
Type a name for the profile you are importing. The name can be a maximum of 30 characters and the name must not already exist.

File name
Type the name of the exported profile that you want to import. This file must exist and be a valid exported profile. You can use a PDS with a member name. Do not use quotes around the specified file name; enter only a fully qualified data set name.

2. When you press **Enter**, the specified profile is imported.

Chapter 13. Using the quick start panels

The quick start panels are designed for quicker and easier job generation. The quick start panels are organized so you can choose what type of Log Analysis Tool activity you want and only see the most important options that are relevant to that activity.

The activity types are the following:

- auditing
- recovering
- replicating

User settings are available regardless of whether you are using classic panels or quick start panels. In quick start mode, however, the user settings have been simplified to make setting up a job quicker and easier. In both panel modes, the profile, message help, setup, and tutorial options are available. This gives you quick access without having to think about the panel mode.

When you save a profile, it is either saved as a classic profile or a quick start profile. This value determines whether you see classic panels or quick start panels when editing or copying the profile:

- In classic mode, any new profile is created as a classic profile.
- In quick start mode, any new profile is created as a quick start profile.

You can type the new line command, `Convert to classic` that converts a quick start profile into a classic profile. This change cannot be undone and there is no way to convert in the opposite direction.

Important:

Quick start panels automatically select certain user options to allow you to more easily generate a job. This can lead to error conditions based on the information that is found at run time. In these scenarios, you must switch to classic mode, set the appropriate options manually, and run log analysis again.

- One example of where a switch to classic mode would be required is if an "ALAA411E MASS DELETE record found during log-backward processing:" message occurs. In this scenario, Log Analysis Tool may have determined log-backward processing was most efficient based on the time frame of the run. However, during execution, a MASS DELETE was encountered which can only be handled via log-forward processing. You must switch to classic mode to set the log reading mode to 'F' and generate the JCL.
- Another example is if a job ends with the error "ALAA628E Unsupportable drop column activity on tablespace." This message indicates that table versioning was found and processing is unable to continue. To move past this issue, you must switch to classic mode to input table versioning information. Once the versioning has been set it would be possible to return to the quick start panels if desired.

Please note that these examples are only a subset of the possible scenarios that would necessitate switching to classic mode.

Selecting quick start activity

You use the quick start menu to select to select the activity you want to perform.

Important: Log Analysis Tool v3.5 opens to the quick start panels the first time you run it. Thereafter, Log Analysis Tool remembers whether you were in classic or quick start panel mode when you exited and starts in that same mode. Type 9 on the command line on either the Quick Start Menu or the Classic Menu to toggle back and forth between classic and quick start panel modes.

From the Db2 Log Analysis Tool - Classic Menu, select option **9**, **Quick start menu** The "Quick Start Menu" appears.

```
------------------ DB2 LOG ANALYSIS TOOL - Quick Start Menu -----------------
Option ===>
                                                    User ID: TWUSR
                                                    Date...: 2020/04/07
                                                    Time...: 11:54
                                                    Release: V3.5.0
                                                    System.: LPR1
 -----------------------------------------------------------------------------
   0. User settings                         Other Options:
                                             7. Utilities
   Audit Activity:                           8. Profiles
   1. Generate reports                       9. Classic menu
                                            10. Message help
   Recover Activity:                        11. Setup
   2. Generate UNDO/REDO SQL                12. Tutorial
   3. Execute dynamic SQL in batch          13. Exit

   Replicate Activity:
   4. Generate LOADFILE
 -----------------------------------------------------------------------------
```

Figure 62. Quick Start Menu

Quick Start Menu options

User settings
Select this option to enter user settings necessary for product utilization. These settings must be defined per user of the product.

Tip: Values of '0' or 'D' select this option. The 'D' value exists for compatibility with earlier releases, when this option was named "Defaults".

Generate reports
Select this option to create reports of Db2 log activity. This option can be used to create both generalized reports and detailed reports.

Generate UNDO/REDO SQL
Select this option to generate SQL that will either UNDO Db2 log activity or REDO Db2 log activity.

Execute dynamic SQL in batch
Select this option to dynamically execute SQL created during the generate UNDO/REDO SQL run. This option cannot be used until SQL has been generated.

Generate LOADFILE
Select this option to generate LOADFILE data for the purpose of replicating Db2 log activity.

Utilities
Select this option to view the utilities menu. The utilities are used to perform various functions within the product.

Profiles
Select this option to view the profiles panel. The profile panel is used to save and load general report, detail report, and LOAD settings.

Classic menu
The classic panels are the panels that have appeared in previous versions of Log Analysis Tool. They are designed to provide more options for Log Analysis jobs.

Message help
Select this option to get more detailed explanation of any ALAA- prefixed messages generated while navigating through the online panels or during generation of any reports.

Tip: Values of 10 or M select this option. The M value exists for compatibility with earlier releases.

Setup
Select this option to enter settings necessary for product utilization. These settings must be defined for each needed Db2 subsystem, and would likely be done by a product administrator or installer.

Tip: Values of 11 or S select this option. The S value exists for compatibility with earlier releases.

Tutorial

Select this option to enter the tutorial for this product. The tutorial contains much information about the product's operations and panels, though HELP (PF1) resides behind each panel as well.

Tip: Values of 12 or T will both select this option The T value exists for compatibility with earlier releases.

Exit

Select this option to exit the product.

Tip: Values of 13 or X will both select this option. The 13 value exists for compatibility with earlier releases.

Quick start-specifying user settings

You use the Quick Start User Settings panel to specify your user settings

```
V3.5.0 ------------------ Quick Start User Settings -------------- SC01/SS1A
Command ===>


File Defaults:
 Optional datasets prefix....................
*Disk unit name for dynamic allocations......
 SMS Management class........................
 SMS Storage class...........................
 SMS Data class..............................

Miscellaneous Product Defaults:
*Continue processing after warnings.......... Y   (Y - Yes, N - No
                                                   E - Yes and some errors)
*Create JSON parameters file................. Y   (Y - Yes, N - No)
*Log read parallelism maximum................ 01  (1-16)
*Use DB2SORT (when possible)................. N   (Y/N)
*Offload to zIIP (when possible)............. Y   (Y/N)
*Ask to save generated jobs as profiles...... N   (Y/N)

DB2 Log Analysis Tool Jobcard:
*==>
 ==>
 ==>
```

Figure 63. Quick Start User Settings

Menu options

File Defaults

Optional datasets prefix

All datasets allocated for this product will be prefixed with the current TSO ID of the user as the default. Optionally, you may specify a data set prefix. It must be no longer than 26 characters in length and a valid data set prefix. For example, XYZ.JOHNDOE is valid, 12345 is not. Leave blank for the default behavior which means datasets are prefixed by your userid.

Important: Do not use ampersands here, only hardcoded values.

Disk unit name for dynamic allocations

This option only applies if you use dynamic allocation. Enter a unit name here to be used internally for data sets that are dynamically allocated. This is unrelated to the unit specification for temporary image copy allocations. For example, set field to "SYSDA". This field can be left blank.

SMS Management class

This option is available if you wish to specify an SMS Management class for files allocated by this product. Only data sets that are allocated during job run time will use the SMS specifications. Leave blank if no such class is desired.

SMS Storage class
This option is available if you wish to specify an SMS Storage class for files allocated by this product. Only data sets that are allocated during job run time will use the SMS specifications. Leave blank if no such class is desired.

SMS Data class
This option is available if you wish to specify an SMS Data class for files allocated by this product. Only data sets that are allocated during job run time will use the SMS specifications. Leave blank if no such class is desired. If data class is specified, this product assumes the data class will provide space allocation information for that data class, and therefore no overriding space allocation is provided by the product. If no space allocation is provided within the specified data class, the allocation will fail. Therefore, only provide a data class name that includes space allocation information.

Important:

- For all SMS class specifications, the requested class may not actually be used if your site has set up automatic class assignment based on other rules (for example, class assignment by data set name).

- Although archived data sets are produced during job run time, they do not use the SMS specifications.

- Data sets created during JCL generation do not use SMS specifications. For example, the data sets used to hold the generated JCL do not use SMS specifications.

Miscellaneous product defaults

Continue processing after warnings
You can specify the following values:

- Type Y to allow processing to continue after any warnings are detected in the general and detail report runs.

- Type N to halt the process completely for any warnings.

- Type E to continue even with some errors.

Create JSON parameters file
This option enables support for JCL simplification and the creation of a JSON parameters file for use as input. Valid values are:

- N - No, do not create this file. A standard JCL job will be generated with MODEPARM and MODE35 to store parameters.

- Y - Yes, create a JSON parameters file as JCL input to store all of the available and actively used parameters. You can also choose to hide the default and unchanged parameters. This allows a job to use valid parameters easily, including values for processing mode and product activity, such as audit, recover, and replication.

For more information on the JSON parameters file, refer to Chapter 23, "Modifying a JSON parameters file," on page 259.

Log read parallelism maximum
Type a value from 1 (default) to 16. This value represents the maximum number of concurrent subtasks that can run to process (in parallel) Db2 logs.

This number should most likely represent the number of tape drives available if Db2 archive logs are on tape and will be accessed. For example, if you expect 10 Db2 logs to be accessed, and your site has a total of 4 tape drives available, a value of 2 might be appropriate to keep two drives available for other tasks. If all your Db2 logs are on DASD, a value of 5 might be appropriate. Keep in mind that your batch address space might exhaust all available storage if a high number is used. Also, you may not want to tie up all available tape drives. So fine tuning may be required.

If you specify 16 as a value, each group of 16 will be processed sequentially, up to the amount needed. That is, no more than 16 subtasks can be reading logs at any given time, and again, that number is likely high in terms of batch address space resource availability.

Use DB2SORT

Type Y if you want the product to use DB2SORT in place of the standard sorting routines. If set to Y, the product will ascertain whether DB2SORT is available for use; if so, it will use it; if not, it will default to its standard sort calls (that is, it will not terminate if DB2SORT is requested but cannot be used, for any reason). There is some overhead involved in making those decisions internally, so if it is known that DB2SORT is not available at your site, set this field value to N. Even if requested, due to restrictions within DB2SORT's API, it will not be used exclusively within the product for every needed sort. Use of DB2SORT may result in better performance.

Offload to zIIP (when possible)

Type Y if you want the product to attempt to offload some processes to the zIIP. Using zIIP may result in reduced CPU usage, but the overheard of sending a process to the zIIP could cause overall run time to increase by a small amount. Type N to not attempt to offload to zIIP.

Ask to save generated jobs as profiles

Type Y if you want to be given the option to create a profile, saving all your specified options, after a job has been generated and edited and/or submitted.

Db2 Log Analysis Tool Jobcard

Enter a valid jobcard here which will be used in all generated JCL. Be sure to enter this information prior to creating any JCL within this product. At least one line of jobcard input must be provided (denoted by the '*==>' prefix).

Quick start-generating reports

You use the Quick Start Report Activity panel to create reports auditing activity.

```
V3.5.0 ------------------ Quick Start Report Activity ------------ SC01/SS1A
Command ===>
                                                              More:     +

DB2 Parameters:
*Subsystem name......... D91A (SSID)
*Log range Start/End.... 2007/01/11 - 00:00:00 / 2007/09/01 - 00:00:00

Filters for Log Data:
*Misc filters........... Y (Y/N)
*Object filters......... M (N - None, M - By Name)
*Show uncommitted....... (Y/N)
*Include catalog data... (Y/N)

High Level Reports:            Detail Level Reports:
 General Report......... Y (Y/N)    Detail Report........... Y (Y/N)
 Summary Report......... Y (Y/N)    Detail Summary Report... N (Y/N)
 Transaction ID Report.. N (Y/N)    Catalog Audit Report.... N (Y/N)
 Commit Frequency Report N (Y/N)

Additional Reports:
 Impact Report.......... N (Y/N) / Requires General or Summary Report and
 Impact Report by Row... N (Y/N) \ consumes large amounts of resources
 Detail Impact Report... N (Y/N) Requires Impact or Impact by Row
 Quiet Time Report...... N (Y/N) Cannot be combined with other reports

LOADFILE Creation (Can Be Used to Create Custom Reports):
*Display LOAD options... N (Y/N)

Other Setting Quick Overrides:
 Job Identifier......... PDUSER
*Processing mode........ M (F - Fewer files with less flexible restart,
                            M - More files with more flexible restart)
```

Figure 64. Quick Start Report Activity

Menu options

Db2 Parameters

Use these fields to specify which Db2 logs to examine.

Db2 subsystem name
> Enter the four character subsystem identifier for the Db2 subsystem. Db2 log activity will be analyzed using this subsystem.

Log Range Start Date/Start Time/End Date/End Time
> Enter these fields in the format indicated on the panel. The Db2 logs will be searched for any activity found between these date/time values. These fields will be set to their default formats if blanked out. A relative End Date and End Time can be requested by entering a plus sign or a minus sign and three digits in the End Date field and blanking out the End Time field. The three digit amount field represents the number of minutes to be used in the relative End Date / Time calculation. If the plus sign is used, then the amount will be added to the Start Date / Time to obtain the End Date / Time. If the minus sign is used then, the amount will be subtracted from the current date and time to obtain the End Date / Time. The three digit amount must be greater than or equal to 000 and less than or equal to 999.

Filters for log data
> Use these fields as filters to be used when processing the Db2 logs. These fields define the type of log records you want to report on. Only those log records matching all the filter criteria are included in your report. All filters are used in conjunction with each other, unless otherwise noted (that is, report log activity only when table = X AND plan name = Y).

Misc filters
> These miscellaneous filters allow you to filter on such things as AUTHID, plan name, job name, connection ID, and connection type. Enter a Y if you wish to use such filters.

Object filters
> These filters should be used when you want to filter on Db2 objects such as databases, table spaces, or tables. A subsequent panel will be displayed if this field is not set to N (none) for your Db2 object level filter input. M indicates you will provide these objects by their names.

Show uncommitted
> Set the field to N (No) to exclude uncommitted units of work, or to Y (Yes) to include them. All uncommitted units of work will be denoted as such.

Include catalog data
> Enter a 'Y' to include any catalog data found in the logs. This flag is used in conjunction with other filters, so the specification of a user table filter would result in no catalog data regardless of this flag setting.

High level reports
> These reports are fairly non-detailed, showing what happened, if anything, to tables during a specific time period. Filters are used to narrow your search during this time period. Making good use of these filters means a much less expensive run of the batch job. The better the filtering is, the quicker the job will run and the less expensive the job will be in CPU and I/O costs.

General report
> Set the field to Y (Yes) to generate a general report. This report shows the number of rows that were modified by an INSERT, UPDATE, or DELETE statement as logged by Db2. The information is summarized by Unit of Work and displays the date and time of the event along with the AUTHID and other information that is contained in the log records.

Summary report
> Set the field to Y (Yes) to generate a summary report. This report shows the number of INSERTs, UPDATEs, and DELETEs per table owner and table name during the specified time period. These counts are accumulated at the TABLESPACE and DATABASE level as well.

Transaction report
> Set the field to Y (Yes) to generate a transaction id report. This report shows the transaction ID, end user, transaction or application name, workstation name, and accounting string per Unit of Work. Db2 does not always log this information. If this information is not logged then it cannot be reported.

Commit frequency report
Set the field to Y (Yes) to generate a commit frequency report. This report provides statistical analysis dealing with the frequency of commits by plan broken down by activity type.

Low level reports
These reports are more detailed than the high level reports. Generating low level reports requires full row images to be generated. This will require more CPU and longer run times. Using the filters to narrow the range of reported data may be vital.

Detail report
Set the field to Y (Yes) to generate a detail report. This report shows the before and after image of the modified rows with column headers. The report also displays information from the log about the unit of work such as the date and time of the event along with the AUTHID.

Detail summary report
Set the field to Y (Yes) to generate a detail summary report. This report shows the number of INSERTs, UPDATEs, and DELETEs per table owner and table name during the specified time period similar to the High Level Summary Report. In addition, this report will display the number of INSERTs that were performed by a load utility and the number of DELETEs that were done by a mass delete operation. These counts are accumulated at the TABLESPACE and DATABASE level as well.

Catalog audit report
Set the field to Y (Yes) to generate a catalog audit report. This report takes the system catalog table INSERTs, UPDATEs, and DELETEs and describes what happened to the Db2 objects in terms of CREATE, ALTER, DROP, BIND, REBIND, FREE, GRANT, and REVOKE.

Additional reports
These reports can be selected to gather additional information when analyzing the data for a desired time period.

Impact report
Set the field to Y (Yes) to generate an impact report. An impact report displays additional activity in the general report and summary report formats. The activity that occurred after the specified timeframe up to the current date and time will be shown if it meets the original filter criteria. This option can cause the product to consume large amounts of resources as additional log records will be read and processed. The general report and/or summary report must be selected if an impact report is desired.

Impact report by row
Set the field to Y (Yes) to generate an impact report by row. This report displays additional activity in the general report and summary report formats. The activity that occurred after the specified timeframe up to the current date and time will be shown only if it occurred on rows that were reported in general report. This option can cause the product to consume large amounts of resources as additional log records will be read and processed. The general report and/or summary report must be selected if an impact report by row is desired.

Detail impact report
Set the field to Y (Yes) to generate a detail impact report. This report shows the before and after images of the rows that were reported with the impact report or impact report by row. This option is not available if neither the impact report nor impact report by row were selected.

Quiet time report
Set the field to Y (Yes) to generate a quiet time report. This report shows periods of time where no activity was logged that met filter criteria. The threshold for determining a quiet period is a configurable option for the report. This report cannot be combined with other reports due to the way the information is obtained from the log.

Display load options
Set the field to Y (Yes) to display the LOAD options panel. The LOAD options panel can be used to load all the activity into Db2 for the purpose of creating customizable reports.

Job identifier
This optional field is an 8 character maximum field that can be used to identify data sets associated with a given run of the created JCL. It will be used in the formation of the dataset names used in the JCL. For example, if set to TEST01, the REFILE will be named USERID.REFILE.TEST01.

Processing mode
Set the field to 'F' to generate fewer files during a log analysis run. Using this setting will force a job to be restarted from the beginning if a problem occurs. If the restarting at the last step of a job is important, choose the 'M' option. This will require more files to be created and utilized during a run.

Quick start-generating UNDO/REDO SQL

You use the Quick Start Generate UNDO/REDO SQL panel to generate SQL.

```
V3.5.0 -------------- Quick Start Generate UNDO/REDO SQL --------- SC01/SS1A
Command ===>
                                                        More:     +

DB2 Parameters:
*Subsystem name......... D91A (SSID)
*Log range Start/End.... 2007/01/11 - 00:00:00 / 2007/09/01 - 00:00:00
*Bypass SYSLGRNX........ N    (Y/N)

Filters for Log Data:
*Misc filters........... Y (Y/N)
*Object filters......... M (N - None, M - By Name, I - by IDs)
*Include LOB/XML data... N (Y/N)
*Include catalog data... (Y/N)
 LOB PDS name...........

High Level Reports:             Detail Level Reports:
 General Report......... Y (Y/N)    Detail Report........... Y (Y/N)
 Summary Report......... Y (Y/N)    Detail Summary Report... N (Y/N)

Additional Reports:
 Impact Report.......... N (Y/N) / Requires General or Summary Report and
 Impact Report by Row... N (Y/N) \ consumes large amounts of resources
 Detail Impact Report... N (Y/N) Requires Impact or Impact by Row

SQL Parameters:
*Commit scope.......... 000 (000-999, UOW)
*Generate SQL.......... R   (R - Redo, U - Undo)
 Set Current SQLID......

 Other Setting Quick Overrides:
 Job Identifier......... PDUSER
*Processing mode........ M (F - Fewer files with less flexible restart,
                           M - More files with more flexible restart)
```

Figure 65. Quick Start Generate UNDO/REDO SQL

Menu options

Db2 Parameters
Use these fields to specify which Db2 logs to examine.

Db2 subsystem name
Enter the four character subsystem identifier for the Db2 subsystem. Db2 log activity will be analyzed using this subsystem.

Log Range Start Date/Start Time/End Date/End Time
Enter these fields in the format indicated on the panel. The Db2 logs will be searched for any activity found between these date/time values. These fields will be set to their default formats if blanked out.

A relative End Date and End Time can be requested by entering a plus sign or a minus sign and three digits in the End Date field and blanking out the End Time field. The three-digit amount field represents the number of minutes to be used in the relative End Date / Time calculation, and must be greater than or equal to 000 and less than or equal to 999.

- If the **plus** sign is used, the amount will be added to the Start Date / Time to obtain the End Date / Time.
- If the **minus** sign is used, the amount will be subtracted from the current date and time to obtain the End Date / Time.

Bypass SYSLGRNX

SYSLGRNX is a Db2-owned internal table that contains useful information about tablespace access as related to RBAs or LRSNs (timestamps). The use of this table can reduce the number of logs this product may have to read. However, there is overhead associated with such access, so bypassing the use of this table may help in some situations, and hurt in others.

If using dropped object mode, always bypass use of SYSLGRNX. Set this field to a value of Y whenever using dropped object mode.

If your SYSLGRNX table has been cleared of data going back as far as your start date/time, it will be necessary to bypass the internal use of this table. In that case, set this field to Y. Also, if filtering on SYSIBM.SYSCOPY, or its likely that this table will be in your result set, also set this field to Y, as that tablespace is not recorded in SYSLGRNX. For any other case, it is your choice.

There is a fair amount of overhead involved in using SYSLGRNX to determine a logs usefulness. If you have many object type filters, or if wildcarding will result in many object type filters ('object type' here means tablespaces), and your time range is very narrow, it can be beneficial to skip SYSLGRNX usage (set to Y). This is because the time it takes to process this information can take longer than any possible savings resulting from eliminating access to a few Db2 logs.

If your timeframe is large, or you know many logs will be encountered for your requested timeframe, then allow the product normal usage of SYSLGRNX information (set to N). If the product encounters problems accessing SYSLGRNX and an error message indicating such is presented, you can set this field to Y to temporarily bypass its use until the problem is resolved.

For SYSLGRNX processing, the DD named SYSPRINT will contain a reference to a temporary data set, and this is hardcoded in the generated JCL. This allocation is set to CYL,(100,100). If you need more space, increase it in this JCL. However, if you do need more space, then the use of SYSLGRNX is most likely unwarranted. This is because that many entries in your SYSLGRNX table will almost certainly mean no logs can be avoided for reading, and that is the whole purpose of using this resource. So if the product gets a B37 abend while processing SYSLGRNX, you should most likely set this field to Y to bypass its use.

Filters for log data

Use these fields as filters to be used when processing the Db2 logs. These fields define the type of log records you want to report on. Only those log records matching all the filter criteria will be included in your report. All filters are used in conjunction with each other, unless otherwise noted (that is, report log activity only when table = X AND plan name = Y).

Misc filters

These miscellaneous filters allow you to filter on such things as authid, plan name, job name, connection ID, and connection type. Enter a Y if you wish to use such filters.

Object filters

These filters should be used when you want to filter on Db2 objects such as databases, table spaces, or tables. A subsequent panel will be displayed if this field is not set to N (none) for your Db2 object level filter input. 'M' indicates you will provide these objects by their names. 'I' means you will provide these objects by their IDs (DBID/PSID/OBID).

Include LOB/XML data

The term "LOB" is used to refer to either LOB or XML data. Set the field to N (No) if the data does not have LOB columns, or Y if there are LOB columns. If set to N when LOB columns are present, SQL cannot be generated. Processing of LOB data will greatly affect performance.

Include catalog data

Enter a 'Y' to include any catalog data found in the logs. This flag is used in conjunction with other filters, so the specification of a user table filter would result in no catalog data regardless of this flag setting.

LOB PDS name

This field identifies the name of a PDS (library) that will be used by the product to contain LOB data. It only has effect if there is LOB data in your result set. If not, it is not used. If LOB data does exist, this PDS will be populated with one member per LOB column data per modified row. For example, if you have 3 INSERTs in your result set, and all INSERTs contain 2 LOB columns each, then there will be 6 members added to this library. The naming convention for the members is Ln, where n = 0000001 thru 9999999. If the member pre-exists, it will be overwritten. The attributes of the PDS must be this:

```
LRECL=32756, BLKSIZE=32760, RECFM=VB,DSORG=PO
```

The number of directory blocks depends on the number of needed members. The PDS will be created for you if it does not pre-exist, with default values for space (that is, SPACE=(CYL, (100,100,50),RLSE)). If you wish to have complete control of this PDS, create it outside of the product and the pre-existing PDS you create will be used instead. You must use the file attributes stated herein if you create it outside of this product. The name given must be a valid PDS name. For example, ALA.LOBS1 is a valid name, whereas ALA.LOBS(L1) is not (i.e., do not include member names here). Do not quote the PDS name, it will be used as entered. If no name is given, and LOB columns exist in your tables, the SQL generation process will fail. This PDS is used as the 'file by reference' component when processing LOB data in SQL.

High level reports

These reports are fairly non-detailed, showing what happened, if anything, to tables during a specific time period. Filters are used to narrow your search during this time period. Making good use of these filters means a much less expensive run of the batch job. The better the filtering is, the quicker the job will run and the less expensive the job will be in CPU and I/O costs.

General report

Set the field to Y (Yes) to generate a general report. This report shows the number of rows that were modified by an insert, update, or delete statement as logged by Db2. The information is summarized by Unit of Work and displays the date and time of the event along with the AUTHID and other information that is contained in the log records.

Summary report

Set the field to Y (Yes) to generate a summary report. This report shows the number of inserts, updates, and delete statements per table owner and table name during the specified time period. These counts are accumulated at the TABLESPACE and DATABASE level as well.

Low level reports

These reports are more detailed than the high level reports. Generating low level reports requires full row images to be generated. This will require more CPU and longer run times. Using the filters to narrow the range of reported data may be vital.

Detail report

Set the field to Y (Yes) to generate a detail report. This report shows the before and after image of the modified rows with column headers. The report also displays information from the log about the unit of work such as the date and time of the event along with the AUTHID.

Detail summary report

Set the field to Y (Yes) to generate a detail summary report. This report shows the number of inserts, updates, and delete statements per table owner and table name during the specified time period similar to the High Level Summary Report. In addition, this report will display the number of inserts that were performed by a load utility and the number of deletes that were done by a mass delete operation. These counts are accumulated at the TABLESPACE and DATABASE level as well.

Additional reports

These reports can be selected to gather additional information when analyzing the data for a desired time period.

Impact report

Set the field to Y (Yes) to generate an impact report. An impact report displays additional activity in the general report and summary report formats. The activity that occurred after the specified

timeframe up to the current date and time will be shown if it meets the original filter criteria. This option can cause the product to consume large amounts of resources as additional log records will be read and processed. The general report and/or summary report must be selected if an impact report is desired.

Impact report by row

Set the field to Y (Yes) to generate an impact report by row. This report displays additional activity in the general report and summary report formats. The activity that occurred after the specified timeframe up to the current date and time will be shown only if it occurred on rows that were reported in general report. This option can cause the product to consume large amounts of resources as additional log records will be read and processed. The general report and/or summary report must be selected if an impact report by row is desired.

Detail impact report

Set the field to Y (Yes) to generate a detail impact report. This report shows the before and after images of the rows that were reported with the impact report or impact report by row. This option is not available if neither the impact report nor impact report by row were selected.

SQL parameters

These parameters are specific to the generated SQL.

Commit Scope

Enter a three-digit number from 000-999 which will indicate how many SQL statements will occur in the optional SQL file before a COMMIT is issued. 000 means no COMMIT's are issued till all statements are executed. You may also enter a value of 'UOW', meaning commit by unit of work. Each unit of work will be executed on its own, and committed immediately after it is completed. If this is desired, it is important to understand that the generated SQL will be ordered by unit of work as well. For example, assume the following DML activity in the Db2 logs:

```
SQL INSERT by UOW 1 at 12:00:01
SQL INSERT by UOW 2 at 12:00:02
SQL INSERT by UOW 2 at 12:00:03
SQL INSERT by UOW 1 at 12:00:04
SQL INSERT by UOW 2 at 12:00:05
UOW 1 ends at 12:00:06
UOW 2 ends at 12:00:07
```

If commit scope = 'UOW', the SQL will be generated this way and in this order:

```
SQL INSERT by UOW 1 at 12:00:01
SQL INSERT by UOW 1 at 12:00:04
SQL COMMIT
SQL INSERT by UOW 2 at 12:00:02
SQL INSERT by UOW 2 at 12:00:03
SQL INSERT by UOW 2 at 12:00:05
SQL COMMIT
```

Generate SQL

Set the flag to 'R' to for REDO SQL. Set the flag to 'U' for UNDO SQL. SQL will be generated in a flat file by the JCL which can be used to either to undo or redo the reported table changes. The flat file is associated with the DD named SQLOUT in the JCL.

Set Current SQLID

Enter either spaces or a valid SQLID into this field. If this field contains anything other than spaces and UNDO SQL or REDO SQL has been requested, a SET CURRENT SQLID statement will be inserted into the SQLOUT file with this value. If this statement is not needed or wanted, then blank out this field

Job identifier

This optional field is an 8 character maximum field that can be used to identify data sets associated with a given run of the created JCL. It will be used in the formation of the dataset names used in the JCL. For example, if set to TEST01, the REFILE will be named USERID.REFILE.TEST01.

Processing mode

Set the field to 'F' to generate fewer files during a log analysis run. Using this setting will force a job to be restarted from the beginning if a problem occurs. If the restarting at the last step of a job is important, choose the 'M' option. This will require more files to be created and utilized during a run.

Using dynamic SQL in batch mode

You should use the **Execute dynamic SQL in batch** option if you want to execute the generated SQL in batch mode, as opposed to submitting it through the Db2 SPUFI utility.

See Chapter 9, "Executing dynamic SQL in batch," on page 155 for the full details.

Quick start-generating a LOADFILE

You use the Quick Start Replicate Activity panel to generate a LOADFILE.

```
V3.5.0 ---------------- Quick Start Replicate Activity ----------- SC01/SS1A
Command ===>


DB2 Parameters:
*Subsystem name......... D91A (SSID)
*Log range Start/End.... 2007/01/11 - 00:00:00 / 2007/09/01 - 00:00:00
*Run in continuous mode. N    (Y/N)

Filters for Log Data:
*Misc filters........... Y (Y/N)
*Object filters......... M (N - None, M - By Name)
*Include LOB/XML data... N (Y/N)
 LOB PDS name...........

High Level Reports:                Detail Level Reports:
 General Report......... Y (Y/N)    Detail Report........... Y (Y/N)
 Summary Report......... Y (Y/N)

Other Setting Quick Overrides:
 Job Identifier......... PDUSER
*Processing mode........ M (F - Fewer files with less flexible restart,
                           M - More files with more flexible restart)
```

Figure 66. Quick Start Replicate Activity

Menu options

Db2 subsystem name

Enter the four character subsystem identifier for the Db2 subsystem. Db2 log activity will be analyzed using this subsystem.

Start Date/Start Time/End Date/End Time

Enter these fields in the format indicated on the panel. The Db2 logs will be searched for any activity found between these date/time values. These fields will be set to their default formats if blanked out. A relative End Date and End Time can be requested by entering a plus sign or a minus sign and three digits in the End Date field and blanking out the End Time field. The three digit amount field represents the number of minutes to be used in the relative End Date / Time calculation. If the plus sign is used, then the amount will be added to the Start Date / Time to obtain the End Date / Time. If the minus sign is used then, the amount will be subtracted from the current date and time to obtain the End Date / Time. The three digit amount must be greater than or equal to 000 and less than or equal to 999.

Run in continuous mode

Enter a Y (Yes) in order to run in continuous mode. If in continuous mode (herein referred to as CM), the start date and time is determined from the contents of the CM file. The CM file will be created based on the optional dataset prefix, the job identifier, and the Log Analysis naming convention. In quick start mode, to use a CM file again, the dataset prefix and the job identifier must be set correctly.

CM means that you want to run reports continuously over time. For example, you want to track updates to a specific object every day. You can generate the CM JCL and place it in a job scheduler for execution every night. Each run of the job will track the needed start date and time for a subsequent run. It will also track uncommitted units of recovery, so that on a subsequent run, the uncommitted status will likely be resolved (commit or rollback). Without CM, uncommitted records are lost if the end date and time occurs prior to commit.

Filters for log data

Use these fields as filters to be used when processing the Db2 logs. These fields define the type of log records you want to report on. Only those log records matching all the filter criteria will be included in your report. All filters are used in conjunction with each other, unless otherwise noted (that is, report log activity only when table = X and plan name = Y).

Misc filters

These miscellaneous filters allow you to filter on such things as AUTHID, plan name, job name, connection ID, and connection type. Enter a Y if you wish to use such filters.

Object filters

These filters should be used when you want to filter on Db2 objects such as databases, table spaces, or tables. A subsequent panel will be displayed if this field is not set to N (none) for your Db2 object level filter input. 'M' indicates you will provide these objects by their names.

Include LOB/XML data

The term "LOB" is used to refer to either LOB or XML data. Set the field to N (No) if the data does not have LOB columns, or Y if there are LOB columns. If set to N when LOB columns are present, SQL cannot be generated. Processing of LOB data will negatively affect performance.

LOB PDS name

This field identifies the name of a PDS (library) that will be used by the product to contain LOB data. It only has effect if there is LOB data in your result set. If not, it is not used. If LOB data does exist, this PDS will be populated with one member per LOB column data per modified row. For example, if you have 3 INSERTs in your result set, and all INSERTs contain 2 LOB columns each, then there will be 6 members added to this library. The naming convention for the members is Ln, where n = 0000001 thru 9999999. If the member pre-exists, it will be overwritten. The attributes of the PDS must be this: LRECL=32756, BLKSIZE=32760, RECFM=VB,DSORG=PO. The number of directory blocks depends on the number of needed members. The PDS will be created for you if it does not pre-exist, with default values for space (that is, SPACE=(CYL,(100,100,50),RLSE)). If you wish to have complete control of this PDS, create it outside of the product and the pre-existing PDS you create will be used instead. You must use the file attributes stated herein if you create it outside of this product. The name given must be a valid PDS name. For example, ALA.LOBS1 is a valid name, whereas ALA.LOBS(L1) is not (i.e., do not include member names here). Do not quote the PDS name, it will be used as entered. If no name is given, and LOB columns exist in your tables, the SQL generation process will fail. This PDS is used as the 'file by reference' component when processing LOB data in SQL.

High level reports

These reports are fairly non-detailed, showing what happened, if anything, to tables during a specific time period. Filters are used to narrow your search during this time period. Making good use of these filters means a much less expensive run of the batch job. The better the filtering is, the quicker the job will run and the less expensive the job will be in CPU and I/O costs.

General report

Set the field to Y (Yes) to generate a general report. This report shows the number of rows that were modified by an INSERT, UPDATE, or DELETE statement as logged by Db2. The information is summarized by Unit of Work and displays the date and time of the event along with the AUTHID and other information that is contained in the log records.

Summary report

Set the field to Y (Yes) to generate a summary report. This report shows the number of INSERTS, UPDATEs, and DELETEs per table owner and table name during the specified time period. These counts are accumulated at the TABLESPACE and DATABASE level as well.

Low level reports

These reports are more detailed than the high level reports. Generating low level reports requires full row images to be generated. This will require more CPU and longer run times. Using the filters to narrow the range of reported data may be vital.

Detail report

Set the field to Y (Yes) to generate a detail report. This report shows the before and after image of the modified rows with column headers. The report also displays information from the log about the unit of work such as the date and time of the event along with the AUTHID.

Job identifier

This optional field is an 8 character maximum field that can be used to identify data sets associated with a given run of the created JCL. It will be used in the formation of the dataset names used in the JCL. For example, if set to TEST01, the REFILE will be named USERID.REFILE.TEST01.

Processing mode

Set the field to F to generate fewer files during a log analysis run. Using this setting will force a job to be restarted from the beginning if a problem occurs. If the restarting at the last step of a job is important, choose the **M** option. This will require more files to be created and utilized during a run.

Chapter 14. Db2 Log Analysis Tool reports

Db2 Log Analysis Tool creates a number of different reports. There are quick reports that provide you with a brief overview of your log activity and there are processing intensive reports that provide you with the smallest details of log transactions. The different reports are:

- BSDS Report
- Catalog Audit Report
- Commit Frequency Report
- Continuous Mode File Report
- Details Report
- Details Summary Report
- Extended Report
- General Report
- IDMAP Report
- Impact Report
- Mode File Report
- Quiet Time Report
- Summary Report
- Transaction ID Report

Each of these reports provides you with the information that you require, depending on your needs.

BSDS Report

The BSDS report is automatically printed to the INFOM DD when running a General report, provided that the INFOM DD is in the JCL for the General Report step. If the INFOM DD is not in the JCL, then the report is not printed. The BSDS report provides two different types of information:

- The remaining time until the RBA and LRSN reach their limit.
- Job information that can aid in the support process.

This is an example of a BSDS report:

```
                      ******************************
                      DB2 LOG ANALYSIS - BSDS REPORT
                      ******************************

 Type Begin Log Range       End Log Range
 ---- -------------------- --------------------
 GMT  2013.262 19:35:00     2013.262 19:35:59
 RBA  00000000000000000000 00000000000000000000
 LRSN 00CBFD1EF09E10000000 00CBFD1FE7B350B00000

 MBR  MBR Name BSDS DSN                        SYS TS (local time)  Flag
    / High RBA Written     High LRSN Written     LRSN Delta
    / Calculation Type
    /      RBA Log Range        LRSN Range          RBAs per minute
    /      Approximate remaining time
    / DB2 Archive/Active Log DSN                              Flag
    /      Type Low Value on Log     High Value on Log
 ---------------------------------------------------------------------------
 0000         D91A.BSDS01                      2013.262 15:31:58.53 02
      00000000011C0282C323 00CBFD1E438E4E000000  00000000000000000000
      Maximum value for LRSN will be reached on 2042-09-17
      Based upon all log files in BSDS
          00515377F323        026B3EE55E7D        0000000783CA
          Time left to WARNING RBA value  YEARS:01048 WEEKS:36 DAYS:003
          Time left to CRITICAL RBA value YEARS:01082 WEEKS:34 DAYS:004
```

```
      Based upon log files used in this job
          000001369323            00002B09027A            000000019E19
          Time left to WARNING RBA value  YEARS:04871 WEEKS:51 DAYS:003
          Time left to CRITICAL RBA value YEARS:05029 WEEKS:40 DAYS:004

      D91A.LOGCOPY1.DS03                                                  84
          RBA   00000000011C014C3000 00000000011C03682FFF
          LRSN 00CBFCF33A8BD4000000 00FFFFFFFFFFFF000000
```

Catalog Audit Report

A Catalog Audit report uses information obtained from the Db2 catalog tables to determine which Db2 commands were executed over the requested timeframe. The following Db2 catalog tables are analyzed during Catalog Audit report generation, thus must be included in the filter set:

- SYSDATABASE
- SYSROLES
- SYSROUTINES
- SYSINDEXES
- SYSSTOGROUP
- SYSTABLES
- SYSTABLESPACE
- SYSTRIGGERS
- SYSCONTROLS
- SYSCOLUMNS
- SYSPACKAGE
- SYSPLAN
- SYSCOLAUTH
- SYSUSERAUTH
- SYSPLANAUTH
- SYSDBAUTH
- SYSSCHEMAAUTH
- SYSPACKAUTH
- SYSRESAUTH
- SYSOUTINEAUTH
- SYSSEQUENCEAUTH
- SYSVARIABLEAUTH
- SYSTABAUTH

The following table describes which catalog tables are used to report CREATE, ALTER and DROP commands of the objects:

Catalog table name	Reported objects
SYSIBM.SYSDATABASE	Databases
SYSIBM.SYSROLES	Roles
SYSIBM.SYSROUTINES	Procedures and functions
SYSIBM.SYSINDEXES	Indexes
SYSIBM.SYSSTOGROUP	Storage groups

Catalog table name	Reported objects
SYSIBM.SYSTABLES	Tables, aliases and views
SYSIBM.SYSTABLESPACE	Tablespaces
SYSIBM.SYSTRIGGERS	Triggers
SYSIBM.SYSCONTROLS	Masks

SYSIBM.SYSCOLUMNS catalog table is also used to describe ALTER TABLE command and specify ADD, DROP, ALTER or RENAME actions for column.

Log records of SYSIBM.SYSPACKAGE and SYSIBM.SYSPLAN tables are used to report BIND, REBIND and FREE commands executed for packages and plans respectively.

Catalog tables, used to report GRANT and REVOKE commands, are described in the following table:

Catalog table name	Reported objects
SYSIBM.SYSCOLAUTH	Columns
SYSIBM.SYSUSERAUTH	Users
SYSIBM.SYSPLANAUTH	Plans
SYSIBM.SYSDBAUTH	Databases
SYSIBM.SYSSCHEMAAUTH	Schemas
SYSIBM.SYSPACKAUTH	Packages
SYSIBM.SYSRESAUTH	Tablespaces, distinct types, JAR files, collections and resources
SYSIBM.SYSROUTINEAUTH	Routines
SYSIBM.SYSSEQUENCEAUTH	Sequences
SYSIBM.SYSVARIABLEAUTH	Variables
SSYSIBM.SYSVARIABLEAUTH	Tableviews

This is an example of a Catalog Audit report:

```
                    ***********************************************
                    DB2 LOG ANALYSIS - CATALOG AUDIT REPORT: D91A
                    ***********************************************


==============================================================================
URID                  TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
--------------------  -----------------  -------- -------- -------- --------
00000000013BFCA27828 2022-07-08-11.13.23 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- ------------------------------------ -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEE31A13/0001 00CD92FEE33EB6000000 CATAUD01

ACTION OBJ TYPE   OBJECT NAME
------ ---------- --------------------------------------------------------------
CREATE DATABASE   CATAUDDB


==============================================================================
URID                  TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
--------------------  -----------------  -------- -------- -------- --------
00000000013BFCA2B3D7 2022-07-08-11.13.23 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- ------------------------------------ -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEE31A13/0001 00CD92FEE363FD000000 CATAUD01
```

```
ACTION OBJ TYPE   OBJECT NAME
------ ---------- ----------------------------------------------------------
CREATE TABLESPACE CATAUDDB.CATAUDTS

==============================================================================
URID                TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
------------------- ------------------ -------- -------- -------- --------
00000000013BFCA2B3D7 2022-07-08-11.13.23 PDALAH  DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ------------------------------------ ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEE31A13/0001 00CD92FEE364AE000000 CATAUD01

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ----------------------------------------------------------
CREATE TABLE      CATAUD_LONG_SCHEMA_NAME_FOR_TESTING_LINE_WRAP.CAT_AUD_LONG_TA
                  BLE_NAME_FOR_TESTING_LINE_WRAP

==============================================================================
URID                TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
------------------- ------------------ -------- -------- -------- --------
00000000013BFCA497C4 2022-07-08-11.13.43 PDALAH  DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ------------------------------------ ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEF623D9/0001 00CD92FEF63870000000 CATAUD03

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ----------------------------------------------------------
DROP   INDEX      PDALAH.D35CTAUX

==============================================================================
URID                TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
------------------- ------------------ -------- -------- -------- --------
00000000013BFCA497C4 2022-07-08-11.13.43 PDALAH  DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ------------------------------------ ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEF623D9/0001 00CD92FEF64BFF000000 CATAUD03

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ----------------------------------------------------------
DROP   TABLE      CATAUD_SCHEMA.CAT_AUD_TABL2

==============================================================================
URID                TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
------------------- ------------------ -------- -------- -------- --------
00000000013BFCA497C4 2022-07-08-11.13.43 PDALAH  DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ------------------------------------ ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEF623D9/0001 00CD92FEF64CCB000000 CATAUD03

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ----------------------------------------------------------
DROP   TABLESPACE CATAUDDB.CATAUDTS

==============================================================================
URID                TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
------------------- ------------------ -------- -------- -------- --------
00000000013BFCA65768 2022-07-08-11.13.43 PDALAH  DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ------------------------------------ ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEF623D9/0001 00CD92FEF69774000000 CATAUD03

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ----------------------------------------------------------
ALTER  VIEW       PDALAH.CAT_AUD_TABL3
       ALTER COLUMN  ROW_DECIMAL                       COLNO(+2)
       POST-CHANGE   DECIMAL(+15,+3)
       PRE-CHANGE    DECIMAL(+8,+0)

==============================================================================
URID                TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
------------------- ------------------ -------- -------- -------- --------
00000000013BFCA65768 2022-07-08-11.13.44 PDALAH  DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ------------------------------------ ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEF623D9/0001 00CD92FEF6B9C6000000 CATAUD03
```

```
ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
ALTER  TABLE      CATAUD_SCHEMA.CAT_AUD_TABL3
       ADD COLUMN   ROW_ADD_COL                         COLNO(+4)
       POST-CHANGE  BIGINT                              VERSION(+1)

================================================================================
URID                 TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
-------------------- ------------------ -------- -------- -------- --------
00000000013BFCA69BFD 2022-07-08-11.13.44 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ---------------------------------- ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FEF623D9/0001 00CD92FEF6BE31000000 CATAUD03

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
CREATE STOGROUP   PDALAH.ALASTO03

================================================================================
URID                 TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
-------------------- ------------------ -------- -------- -------- --------
00000000013BFCA6B787 2022-07-08-11.13.56 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ---------------------------------- ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF02BCB3/0001 00CD92FF02E83E000000 CATAUD04

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
CREATE FUNCTION   AUDTST01

================================================================================
URID                 TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
-------------------- ------------------ -------- -------- -------- --------
00000000013BFCA6E3F7 2022-07-08-11.13.57 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ---------------------------------- ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF02BCB3/0001 00CD92FF03197D000000 CATAUD04

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
CREATE FUNCTION   AUDTST04_LONG1234567890_NAME_TEST012345

================================================================================
URID                 TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
-------------------- ------------------ -------- -------- -------- --------
00000000013BFCA6F9D4 2022-07-08-11.13.57 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ---------------------------------- ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF02BCB3/0001 00CD92FF03677B000000 CATAUD04

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
DROP   FUNCTION   AUDTST01

================================================================================
URID                 TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
-------------------- ------------------ -------- -------- -------- --------
00000000013BFCA6F9D4 2022-07-08-11.13.57 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ---------------------------------- ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF02BCB3/0001 00CD92FF038D73000000 CATAUD04

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
DROP   FUNCTION   AUDTST04_LONG1234567890_NAME_TEST012345

================================================================================
URID                 TIMEFRAME          AUTHID   PLAN     CONNTYPE CONNID
-------------------- ------------------ -------- -------- -------- --------
00000000013BFCA71EDA 2022-07-08-11.14.04 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
--------- ---------------------------------- ------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF0A6C30/0001 00CD92FF0A6D51000000 CATAUD05

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ------------------------------------------------------------
```

```
GRANT  COLLECTION CATAUD_COLLID
       GRANTEE   PDALAHA
       PRIVILEGE(S) WITH GRANT OPTION: USE

===============================================================================
URID                 TIMEFRAME          AUTHID   PLAN      CONNTYPE CONNID
-------------------- ------------------ -------- --------- -------- --------
00000000013BFCA7226B 2022-07-08-11.14.04 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- --------------------------------- -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF0A6C30/0001 00CD92FF0A6DBF000000 CATAUD05

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ---------------------------------------------------------------
GRANT  COLLECTION PACKADM.CATAUD_COLLID
       GRANTEE   PDALAHA
       PRIVILEGE(S) WITH GRANT OPTION: USE

===============================================================================
URID                 TIMEFRAME          AUTHID   PLAN      CONNTYPE CONNID
-------------------- ------------------ -------- --------- -------- --------
00000000013BFCA72912 2022-07-08-11.14.04 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- --------------------------------- -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF0A6C30/0001 00CD92FF0A77BA000000 CATAUD05

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ---------------------------------------------------------------
GRANT  PACKAGE    CATAUD_COLLID.*
       GRANTEE   PDALAHA
       PRIVILEGE(S): BIND COPY EXECUTE

===============================================================================
URID                 TIMEFRAME          AUTHID   PLAN      CONNTYPE CONNID
-------------------- ------------------ -------- --------- -------- --------
00000000013BFCA7326B 2022-07-08-11.14.04 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- --------------------------------- -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF0A6C30/0001 00CD92FF0A8560000000 CATAUD05

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ---------------------------------------------------------------
REVOKE COLLECTION PACKADM.CATAUD_COLLID
       GRANTEE   PDALAHA
       PRIVILEGE(S) WITH GRANT OPTION: USE

===============================================================================
URID                 TIMEFRAME          AUTHID   PLAN      CONNTYPE CONNID
-------------------- ------------------ -------- --------- -------- --------
00000000013BFCA735AA 2022-07-08-11.14.05 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- --------------------------------- -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF0A6C30/0001 00CD92FF0B5306000000 CATAUD05

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ---------------------------------------------------------------
REVOKE PACKAGE    CATAUD_COLLID.*
       GRANTEE   PDALAHA
       PRIVILEGE(S): BIND

===============================================================================
URID                 TIMEFRAME          AUTHID   PLAN      CONNTYPE CONNID
-------------------- ------------------ -------- --------- -------- --------
00000000013BFCA735AA 2022-07-08-11.14.05 PDALAH   DSNTEP2  BATCH    BATCH

MEMBER NM LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- --------------------------------- -------------------- ------------
D91A      ROCKNET1/D91ADB2 /CD92FF0A6C30/0001 00CD92FF0B530A000000 CATAUD05

ACTION OBJ TYPE   OBJECT NAME
------ ---------- ---------------------------------------------------------------
REVOKE PACKAGE    CATAUD_COLLID.*
       GRANTEE   PDALAHA
       PRIVILEGE(S): COPY
```

Commit Frequency Report

The Commit Frequency Report provides some statistical analysis dealing with frequency of commits by plan.

- The Commit Frequency report can be combined with the General, Summary, Transaction, Extended and Continuous Mode File reports.
- The Commit Frequency report cannot be combined with the Quiet Time Report.

This is an example of a Commit Frequency Report:

```
                        ************************************************
1                       DB2 LOG ANALYSIS- COMMIT FREQUENCY REPORT: SS1A
                        ************************************************

 LOG RANGE
 ---------
 START DATE    : 2011/04/14
 START TIME    : 00:30:00
 END DATE      : 2011/04/14
 END TIME      : 01:30:00

 FILTERS
 -------
 SHOW UPDATES   : Y
 SHOW INSERTS   : Y
 SHOW DELETES   : Y
 SHOW ROLLBACKS : Y
 CATALOG DATA   : N

              Time is displayed in microseconds.

 ================================================================================
 Plan name:      AUOASHE3 Total times invoked: 23•

                 Total       Total    Average   Minimum   Maximum  Standard
 Summary        Counts        Time       Time      Time      Time Deviation
 --------------- -------- ------------ --------- --------- --------- ---------
 URID               31     5151080     166163        24   4372256    770920
  Commit            31     5151080     166163        24   4372256    770920

                 Total       Total    Average   Minimum   Maximum  Standard
 Details        Counts        Time       Time      Time      Time Deviation
 --------------- -------- ------------ --------- --------- --------- ---------
 All Activity    18088     5044312        278         0     78936      1856
  Commit         18088     5044312        278         0     78936      1856
   Update           49       58632       1196         0     17136      3391
   Insert         9039     1831944        202         0     68984      1683
   Delete         9000     3153736        350         0     78936      1999

 --------------------------------------------------------------------------------
 DBName.TSName:   AUOVRD.AUOCOP•

 Owner.Table:     AUOVR.SYSCOPY

                 Total       Total    Average   Minimum   Maximum  Standard
 Details        Counts        Time       Time      Time      Time Deviation
 --------------- -------- ------------ --------- --------- --------- ---------
 All Activity        4       51456      12864         0     38240     15068
  Commit             4       51456      12864         0     38240     15068
   Insert            4       51456      12864         0     38240     15068

 --------------------------------------------------------------------------------
```

Continuous Mode File Report

The Continuous Mode File Report contains information about the contents of the continuous mode file as it exists at the very end of the Log Analysis job.

You can run a report on the contents of the continuous mode file in order to examine the URIDs that occurred within the Log Analysis job's timeframe, but that were not committed by the ending timeframe. This report is run during the General Report step.

An example of a Continuous Mode File Report is shown below. For information on the various columns and values, see .

```
          *************************************************
             DB2 LOG ANALYSIS-CONTINUOUS MODE FILE REPORT: DA1A
          *************************************************

NEXT START DATE : 2013/09/20
NEXT START TIME : 09:06:59


===============================================================================
URID                    DATE       TIME     AUTHID    PLAN     CONNTYPE CONNID
--------------------    ---------- -------- --------  -------- -------- --------
0000000000C803BD8CB8 2013-09-20 09:06:32 PDMCWH    DSNESPCS BATCH    TSO

MEMBER ID LUWID= NETID/LUNAME/UNIQUE/COMMIT    LRSN                  CORRID
--------- ------------------------------------ -------------------- ------------
00000      ROCKNET1/DA1ADB2 /CBFE09F9B4AB/0003 00CBFE09FA0271000000 PDMCWH


-------------------------------------------------------------------------------
DBNAME   DBID  TSNAME   PSID  TABLE OWNER TABLE NAME         OBID  ACTION/COUNT
-------- ----- -------- ----- ----------- ------------------ ----- ------------
SWMDB    00387 TSPMR002 00061 PDMCWH      LAT_TEST_02        00064 I/1
```

⚠️ **Attention:** When using continuous mode processing, it is possible that units of work (UOW) will be in the Continuous Mode File that do not match your filters. Most of the time these UOWs are uncommitted during the specified timeframe and must be tracked in case additional log records come through that match the filter criteria the next time the job runs.

In certain circumstances, SYSLGRNX may have eliminated a Db2 log file that contained the COMMIT or ABORT record so this UOW remains in a pending state. In this case, the UOW will be tracked for an internally specified period to make sure no subsequent activity comes in. After this internally specified period, with no activity that matches the filter, this UOW will go away.

If you use MISC FLAGS=X, all the logs for the time period are read and the pending UOW will not be included in the continuous mode file – just the ones that are truly uncommitted. This does not cause any problems other than it may appear that more activity is being tracked / reported on than was requested.

Control File Report

The Control File Report is automatically printed to the INFOM DD for the given SSID.

This is an example of a Control File Report:

```
          ****************************************
             DB2 LOG ANALYSIS - CONTROL FILE REPORT
          ****************************************

SSID...................... D91A
DB2 ZPARMS MEMBER......... D91APARM
      SITE TYPE.......... LOCALSITE
      CATALOG ALIAS...... D91A
DB2 Bootstrap DSN #01..... D91A.BSDS01
DB2 Bootstrap DSN #02..... D91A.BSDS02

DB2 Loadlib1.............. D91A.SDSNEXIT
DB2 Loadlib2.............. DSN.V910.SDSNLOAD
DB2 Loadlib3.............. FEC.MNT0130.LOADLIB
DB2 Loadlib4.............. FEC.PRD0130.LOADLIB
DB2 Loadlib5.............. DSN.V910.RUNLIB.LOAD

Other Loadlib1...........
Other Loadlib2...........
Other Loadlib3...........
Other Loadlib4...........
Other Loadlib5...........

Plan Name................. ALAV35W
Message Library........... ALA.TST0350.ISPMLIB
DSNHDECP module...........MYDECP
Use only ARCHLOG1......... N
```

```
Use only ARCHLOG2......... P
Active Log priority....... N
```

Note: The **DSNHDECP module** detail line is included in the report only if an alternate DSNHDECP module name was specified using the Setup option from the Log Analysis Tool main menu.

Details Report

Each Details Report "group" of information, from RECORD IDENTIFIER to the last column reported on, represents a row modification to a table, as in the General Report. The heading shows information similar to the General Report, followed by the three row images at different points in time. Any report differences are noted:

RECORD IDENTIFIER
This field identifies the specific line of output on which you might want to see LOB (or XML) data. This identifier will be present on all detail report records regardless of the inclusion of LOB data, and is merely a non-ordered number associated with a particular reported action.

ACTION
This field shows the type of activity that affected the named table. The ACTION indicators are:

- **DELETE**
- **INSERT**
- **UPDATE**
- **DEL-CR** - Delete due to a compensation log record.
- **DEL-RI** - Delete due to referential integrity.
- **DEL-TR** - Delete due to a trigger.
- **INS-CR** - Insert due to a compensation log record.
- **INS-RI** - Insert due to Referential Integrity (may display during rollback processing).
- **INS-TR** - Insert due to a trigger.
- **UPD-CR** - Update due to a compensation log record.
- **UPD-RI** - Update due to referential integrity.
- **UPD-TR** - Update due to a trigger.

CURRENT
The row image is what the row looks like at the time the report is run.

Note: The CURRENT row image will not exist in a log-forward detail report.

POST-CHANGE
The row image is a snapshot of what the row looks like immediately after the modification of this log record action.

PRE-CHANGE
The row image is a snapshot of what the row looks like immediately before the log record action.

In the column data display component of the Details Report, several indicators are provided to help you understand and interpret the data.

- If the pre-change row image differs from the post-change row image, meaning that the action being reported on is an UPDATE, any columns changed are marked with a pound sign (#) following the column name.
- If the current row image (when available) differs from the post-change row image, the changed column is displayed with an exclamation point (!) following the column name.
- If the column value changed both as a result of the UPDATE, and the current row image is also different, both these signs are displayed in that order (#!). This allows the user to more easily identify where changes have occurred since the DML reported on occurred

- If the action is a DELETE, the data displayed in the POST-CHANGE and CURRENT columns is marked with a dash (-) indicating that no row currently exists. This can also occur in the CURRENT row image of an INSERT or UPDATE action if the row was deleted at a later time. Null values in columns are always listed as NULL.

Note: The LRSN column in the report contains a hexadecimal display value of the log record timestamp. This field represents the actual time of the log record creation, and is used mostly in data sharing environments to synchronize log records across members of a data sharing group.

If an impact report has been requested and there is impact report data, the impact report is written to the DTLIMP DD, if it is present in the JCL. Otherwise, a separate section will be included in the Details Report denoted by the heading IMPACT REPORT BEGINS. The data after this heading is identical in format to other data contained in the report, except that it is associated with the Impact Report data only (the data that occurred after your requested end date and time to the current date and time).

This is an example of a Details Report:

```
1                       ****************************************
                        DB2 LOG ANALYSIS - DETAILS REPORT: D91A
                        ****************************************


 RECORD IDENTIFIER: 1 ---------------------------------------------------------------

 ACTION DATE       TIME     TABLE OWNER  TABLE NAME
 ------ ---------- -------- ------------ -------------------------
 INSERT 2013-09-19 15.35.27 PDMCWH       LOG_ANALYSIS_TOOL

 URID                DATABASE TABLESPACE DBID  PSID  OBID  AUTHID
 ------------------- -------- ---------- ----- ----- ----- --------
 00000000011C028633AF SWMDB91S TSRPRT01   00268 00182 00200 PDMCWH

 PLAN     CONNTYPE LRSN                 MEMID CORRID       CONNID
 -------- -------- -------------------- ----- ------------ --------
 DSNTEP2  BATCH    00CBFD1F0ADC56000000 00000 RPTVARIN     BATCH

 LUW=NETID/LUNAME/UNIQUE/COMMIT     PAGE/RID
 ---------------------------------- -----------
 ROCKNET1/D91ADB2 /CBFD1F0ADB10/0001 00000002/01
 ROW STATUS  ROW_ID ROW_VARCHAR            ROW_CHAR ROW_AMOUNT
 ----------- ------ ---------------------- -------- ----------
 POST-CHANGE +1     Log Analysis Tool Insert A         +03.141
 PRE-CHANGE  -      -                      -        -


 RECORD IDENTIFIER: 2 ---------------------------------------------------------------

 ACTION DATE       TIME     TABLE OWNER  TABLE NAME
 ------ ---------- -------- ------------ -------------------------
 UPDATE 2013-09-19 15.35.27 PDMCWH       LOG_ANALYSIS_TOOL

 URID                DATABASE TABLESPACE DBID  PSID  OBID  AUTHID
 ------------------- -------- ---------- ----- ----- ----- --------
 00000000011C02863688 SWMDB91S TSRPRT01   00268 00182 00200 PDMCWH

 PLAN     CONNTYPE LRSN                 MEMID CORRID       CONNID
 -------- -------- -------------------- ----- ------------ --------
 DSNTEP2  BATCH    00CBFD1F0ADD22000000 00000 RPTVARIN     BATCH

 LUW=NETID/LUNAME/UNIQUE/COMMIT     PAGE/RID
 ---------------------------------- -----------
 ROCKNET1/D91ADB2 /CBFD1F0ADB10/0001 00000002/01

 ROW STATUS  ROW_ID ROW_VARCHAR#           ROW_CHAR ROW_AMOUNT
 ----------- ------ ---------------------- -------- ----------
 POST-CHANGE +1     Log Analysis Tool Update A         +03.141
 PRE-CHANGE  +1     Log Analysis Tool Insert A         +03.141


 RECORD IDENTIFIER: 3 ---------------------------------------------------------------

 ACTION DATE       TIME     TABLE OWNER  TABLE NAME
 ------ ---------- -------- ------------ -------------------------
 DELETE 2013-09-19 15.35.27 PDMCWH       LOG_ANALYSIS_TOOL

 URID                DATABASE TABLESPACE DBID  PSID  OBID  AUTHID
 ------------------- -------- ---------- ----- ----- ----- --------
```

```
    00000000011C02863867 SWMDB91S TSRPRT01   00268 00182 00200 PDMCWH

    PLAN    CONNTYPE LRSN                MEMID CORRID      CONNID
    PLAN    CONNTYPE LRSN                MEMID CORRID      CONNID
    -------- -------- -------------------- ----- ------------ --------
    DSNTEP2  BATCH    00CBFD1F0ADDDC000000 00000 RPTVARIN     BATCH

    LUW=NETID/LUNAME/UNIQUE/COMMIT     PAGE/RID
    ----------------------------------- -----------
    ROCKNET1/D91ADB2 /CBFD1F0ADB10/0001 00000002/01

    ROW STATUS   ROW_ID ROW_VARCHAR            ROW_CHAR ROW_AMOUNT
    ----------- ------ ---------------------- -------- ----------
    POST-CHANGE -      -                      -        -
    PRE-CHANGE  +1     Log Analysis Tool Update A       +03.141
```

Details Summary Report

The Details Summary report is a summary of the activity in the Details Report. The counts include how many rows were inserted by the LOAD utility and how many rows were deleted by a mass delete operation.

This is an example of a Details Summary report:

```
                         ****************************************
                         DB2 LOG ANALYSIS - DETAILS SUMMARY: DS2
                         ****************************************

*****************************
* COMMITTED ACTIVITY        *
*****************************

OBJECT TYPE/NAME                       UPDATES     INSERTS     INSERTS(LD) DELETES     DELETES(MD)
-------------------------------------- ----------- ----------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_ACTOR           4           0           0           0           0
TABLESPACE. TSSWM001                   4           0           0           0           0
TABLE...... PDMCWH.SWM_GENRE           2           0           0           1           0
TABLESPACE. TSSWM002                   2           0           0           1           0
TABLE...... PDMCWH.SWM_MOVIE           1           0           0           1           0
TABLESPACE. TSSWM003                   1           0           0           1           0
TABLE...... PDMCWH.SWM_MOVIE_ACTORS    0           0           0           4           0
TABLESPACE. TSSWM004                   0           0           0           4           0
TABLE...... PDMCWH.SWM_ACTOR_ROLES     4           0           0           0           0
TABLESPACE. TSSWM005                   4           0           0           0           0
DATABASE... SWMDB81S                   11          0           0           6           0


OBJECT TYPE/NAME (RI ACTIONS ONLY)     UPDATES     INSERTS     INSERTS(LD) DELETES     DELETES(MD)
-------------------------------------- ----------- ----------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_MOVIE_ACTORS    0           0           0           4           0
TABLESPACE. TSSWM004                   0           0           0           4           0
DATABASE... SWMDB81S                   0           0           0           4           0

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY) UPDATES    INSERTS     INSERTS(LD) DELETES     DELETES(MD)
-------------------------------------- ----------- ----------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_ACTOR           4           0           0           0           0
TABLESPACE. TSSWM001                   4           0           0           0           0
TABLE...... PDMCWH.SWM_GENRE           2           0           0           0           0
TABLESPACE. TSSWM002                   2           0           0           0           0
TABLE...... PDMCWH.SWM_MOVIE           0           0           0           1           0
TABLESPACE. TSSWM003                   0           0           0           1           0
TABLE...... PDMCWH.SWM_ACTOR_ROLES     4           0           0           0           0
TABLESPACE. TSSWM005                   4           0           0           0           0
DATABASE... SWMDB81S                   10          0           0           1           0


TOTAL SUMMARY REPORT
--------------------------------
TOTAL UPDATES      : 11
TOTAL INSERTS      : 0
TOTAL INSERTS(LOADS): 0
TOTAL DELETES      : 6
TOTAL DELETES(MD)  : 0

*****************************
* UNCOMMITTED ACTIVITY      *
```

```
******************************
OBJECT TYPE/NAME                        UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------

OBJECT TYPE/NAME (COMPENSATION RECORDS)  UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------

OBJECT TYPE/NAME (RI ACTIONS ONLY)       UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY)  UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------

TOTAL SUMMARY REPORT
---------------------------------
TOTAL UPDATES      : 0
TOTAL INSERTS      : 0
TOTAL INSERTS(LOADS): 0
TOTAL DELETES      : 0
TOTAL DELETES(MD)  : 0

******************************
* ROLLBACK (ABORTED) ACTIVITY *
******************************

OBJECT TYPE/NAME                        UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_ACTOR            4            0          0           0           0
TABLESPACE. TSSWM001                    4            0          0           0           0
TABLE...... PDMCWH.SWM_GENRE            2            0          0           1           0
TABLESPACE. TSSWM002                    2            0          0           1           0
TABLE...... PDMCWH.SWM_MOVIE            1            0          0           1           0
TABLESPACE. TSSWM003                    1            0          0           1           0
TABLE...... PDMCWH.SWM_MOVIE_ACTORS     0            0          0           4           0
TABLESPACE. TSSWM004                    0            0          0           4           0
TABLE...... PDMCWH.SWM_ACTOR_ROLES      4            0          0           0           0
TABLESPACE. TSSWM005                    4            0          0           0           0
DATABASE... SWMDB81S                    11           0          0           6           0


OBJECT TYPE/NAME (RI ACTIONS ONLY)      UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_MOVIE_ACTORS     0            0          0           4           0
TABLESPACE. TSSWM004                    0            0          0           4           0
DATABASE... SWMDB81S                    0            0          0           4           0

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY) UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_ACTOR            4            0          0           0           0
TABLESPACE. TSSWM001                    4            0          0           0           0
TABLE...... PDMCWH.SWM_GENRE            2            0          0           0           0
TABLESPACE. TSSWM002                    2            0          0           0           0
TABLE...... PDMCWH.SWM_MOVIE            0            0          0           1           0
TABLESPACE. TSSWM003                    0            0          0           1           0
TABLE...... PDMCWH.SWM_ACTOR_ROLES      4            0          0           0           0
TABLESPACE. TSSWM005                    4            0          0           0           0
DATABASE... SWMDB81S                    10           0          0           1           0


TOTAL SUMMARY REPORT
---------------------------------
TOTAL UPDATES      : 11
TOTAL INSERTS      : 0
TOTAL INSERTS(LOADS): 0
TOTAL DELETES      : 6
TOTAL DELETES(MD)  :
0


                         ************************************************
                         DB2 LOG ANALYSIS - DETAILS SUMMARY(IMPACT): DS2
                         ************************************************

******************************
* COMMITTED ACTIVITY          *
******************************

OBJECT TYPE/NAME                        UPDATES      INSERTS    INSERTS(LD) DELETES     DELETES(MD)
------------------------------------    -----------  ---------- ----------- ----------- -----------
TABLE...... PDMCWH.SWM_ACTOR            4            0          0           0           0
```

```
TABLESPACE. TSSWM001                      4           0           0           0           0
TABLE...... PDMCWH.SWM_GENRE             19           0           0           4           0
TABLESPACE. TSSWM002                     19           0           0           4           0
TABLE...... PDMCWH.SWM_MOVIE             1            8           0           9           0
TABLESPACE. TSSWM003                     1            8           0           9           0
TABLE...... PDMCWH.SWM_MOVIE_ACTORS      0            0           0           4           0
TABLESPACE. TSSWM004                     0            0           0           4           0
TABLE...... PDMCWH.SWM_ACTOR_ROLES       4            0           0           1           0
TABLESPACE. TSSWM005                     4            0           0           1           0
TABLE...... PDMCWH.SWM_ACTOR_TRIGRI      1            0           0           1           0
TABLESPACE. TSSWM006                     1            0           0           1           0
DATABASE... SWMDB81S                     29           8           0           19          0
```

OBJECT TYPE/NAME (RI ACTIONS ONLY)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)
TABLE...... PDMCWH.SWM_MOVIE_ACTORS	0	0	0	4	0
TABLESPACE. TSSWM004	0	0	0	4	0
TABLE...... PDMCWH.SWM_ACTOR_TRIGRI	1	0	0	0	0
TABLESPACE. TSSWM006	1	0	0	0	0
DATABASE... SWMDB81S	1	0	0	4	0

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)
TABLE...... PDMCWH.SWM_ACTOR	4	0	0	0	0
TABLESPACE. TSSWM001	4	0	0	0	0
TABLE...... PDMCWH.SWM_GENRE	14	0	0	0	0
TABLESPACE. TSSWM002	14	0	0	0	0
TABLE...... PDMCWH.SWM_MOVIE	0	0	0	5	0
TABLESPACE. TSSWM003	0	0	0	5	0
TABLE...... PDMCWH.SWM_ACTOR_ROLES	4	0	0	1	0
TABLESPACE. TSSWM005	4	0	0	1	0
TABLE...... PDMCWH.SWM_ACTOR_TRIGRI	0	0	0	1	0
TABLESPACE. TSSWM006	0	0	0	1	0
DATABASE... SWMDB81S	22	0	0	7	0

```
TOTAL SUMMARY REPORT
-----------------------------------
TOTAL UPDATES      : 29
TOTAL INSERTS      : 8
TOTAL INSERTS(LOADS): 0
TOTAL DELETES      : 19
TOTAL DELETES(MD)  : 0


*****************************
* UNCOMMITTED ACTIVITY      *
*****************************
```

OBJECT TYPE/NAME	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

OBJECT TYPE/NAME (COMPENSATION RECORDS)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

OBJECT TYPE/NAME (RI ACTIONS ONLY)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

```
TOTAL SUMMARY REPORT
-----------------------------------
TOTAL UPDATES      : 0
TOTAL INSERTS      : 0
TOTAL INSERTS(LOADS): 0
TOTAL DELETES      : 0
TOTAL DELETES(MD)  : 0


*****************************
* ROLLBACK (ABORTED) ACTIVITY *
*****************************
```

OBJECT TYPE/NAME	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

OBJECT TYPE/NAME (COMPENSATION RECORDS)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

OBJECT TYPE/NAME (RI ACTIONS ONLY)	UPDATES	INSERTS	INSERTS(LD)	DELETES	DELETES(MD)

```
-------------------------------------  ----------- ----------- ----------- ----------- -----------
OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY)  UPDATES     INSERTS     INSERTS(LD) DELETES     DELETES(MD)
-------------------------------------  ----------- ----------- ----------- ----------- -----------

TOTAL SUMMARY REPORT
---------------------------------
TOTAL UPDATES        : 0
TOTAL INSERTS        : 0
TOTAL INSERTS(LOADS): 0
TOTAL DELETES        : 0
TOTAL DELETES(MD)    : 0
```

Extended Report

The Extended Report is an optional extension of the General Report. It allows you to view the General Report output by transaction, rather than grouped by like-transaction per table, as in the General Report. It also provides a more technical view of the transaction. Also grouped by URID, as in the General Report, it shows:

URID: nnnnnnnn (UNCOMMITTED)
> This denotes that the URID was uncommitted at the time of the report.

ACT
> The action taken:
> - **D** -- Delete
> - **I** -- Insert
> - **U** -- Update
> - **DCR** -- Delete due to a compensation log record
> - **DRI** -- Delete due to referential integrity
> - **DTR** -- Delete due to a trigger
> - **ICR** -- Insert due to compensation log record
> - **IRI** -- Insert due to referential integrity (may display during rollback processing)
> - **ITR** -- Insert due to a trigger
> - **UCR** -- Update due to a compensation log record
> - **URI** -- Update due to referential integrity
> - **UTR** -- Update due to a trigger

DBID
> Database Db2 object identifiers

PSID
> Table space Db2 object identifiers

OBID
> Table Db2 object identifiers

PAGE
> The hexadecimal page where the affected row resides (at the time of the modification).

RID
> The hexadecimal row identifier of that row on the page.

RBA
> The relative byte address of the log record.

LRSN
> The log recording sequence number of the log record (basically, a Db2 timestamp).

Byte count
> The total number of bytes contained within the log record. These byte counts are totaled at the end of the extended report.

The byte count fields can be useful in comparing table activity as logged with and without DATA CAPTURE CHANGES. This Db2 option determines if full row images are logged, or just the changed portions of the row are logged. Using this report, you can easily identify the extra amount of logging that can occur with this option in effect.

If an impact report has been requested and there is impact report data, the impact report is written to the EXTIMP DD, if it is present in the JCL. Otherwise, a separate section will be included in the Extended Report denoted by the heading IMPACT REPORT BEGINS. The data after this heading is identical in format to other data contained in the report, except that it is associated with the Impact Report data only (the data that occurred after your requested end date and time to the current date and time).

This is an example of an Extended Report:

```
                      **********************************************
                        DB2 LOG ANALYSIS-EXTENDED REPORT: D91A
                      **********************************************

LOG RANGE
---------
START DATE     : 2013/09/19
START TIME     : 15:35:00
END DATE       : 2013/09/19
END TIME       : 15:35:59

FILTERS
-------
SHOW UPDATES   : Y
SHOW INSERTS   : Y
SHOW DELETES   : Y
SHOW ROLLBACKS : Y
CATALOG DATA   : N
   INCLUDE-TABLE...... PDMCWH.%


URID: 00000000011C028633AF------------------------------------------------------
================================================================================
ACT DBID  PSID  OBID  PAGE      RID RBA         LRSN        BYTES
--- ----- ----- ----- -------- --- ---------- ---------- -----
I   00268 00182 00200 00000002 01  0000000001+ 00CBFD1F0A+   153
                                   1C02863510  DC56000000


URID: 00000000011C02863688------------------------------------------------------
================================================================================
ACT DBID  PSID  OBID  PAGE      RID RBA         LRSN        BYTES
--- ----- ----- ----- -------- --- ---------- ---------- -----
U   00268 00182 00200 00000002 01  0000000001+ 00CBFD1F0A+   126
                                   1C02863747  DD22000000


URID: 00000000011C02863867------------------------------------------------------
================================================================================
ACT DBID  PSID  OBID  PAGE      RID RBA         LRSN        BYTES
--- ----- ----- ----- -------- --- ---------- ---------- -----
D   00268 00182 00200 00000002 01  0000000001+ 00CBFD1F0A+   153
                                   1C02863926  DDDC000000


URID: 00000000011C02863A9E------------------------------------------------------
================================================================================
ACT DBID  PSID  OBID  PAGE      RID RBA         LRSN        BYTES
--- ----- ----- ----- -------- --- ---------- ---------- -----
I   00268 00182 00200 00000002 01  0000000001+ 00CBFD1F11+   153
                                   1C02863B2E  9789000000


URID: 00000000011C02863CA6------------------------------------------------------
================================================================================
ACT DBID  PSID  OBID  PAGE      RID RBA         LRSN        BYTES
--- ----- ----- ----- -------- --- ---------- ---------- -----
U   00268 00182 00200 00000002 01  0000000001+ 00CBFD1F11+   126
                                   1C02863D65  9800000000


URID: 00000000011C02863E85------------------------------------------------------
================================================================================
ACT DBID  PSID  OBID  PAGE      RID RBA         LRSN        BYTES
```

```
--- ----- ----- ----- -------- --- ---------- ---------- -----
D   00268 00182 00200 00000002 01  0000000001+ 00CBFD1F11+   153
                                   1C02863F44  985D000000


TOTAL BYTE COUNTS BY TABLE ID───────────────────────────────────────────────
============================

DBID  PSID  OBID  BYTES
----- ----- ----- -----------
00268 00182 00200          864
```

General Report

Each General Report "group" of information, from URID to ACTION/COUNT, represents a row modification to a table. Most data in the report is self-explanatory and comes directly from the Db2 log record. Those that do not, however, are explained here:

URID, URID(R), or URID(U)
> **URID** - The unit of recovery identifier.
>
> **URID(R)** - The unit of recovery identifier that was rolled back.
>
> **URID(U)** - The unit of recovery identifier that was uncommitted.

MEMBER ID
> Identifies which member in a data sharing group did the modification. The member ID is the Db2 internally defined numeric representation of the member. If not part of a data sharing group, it will be all zeros.

LUWID
> Logical unit of work ID.

ACTION/COUNT
> This field shows the type of activity and number of log records found that affected the named table within the URID group. For example, U/35 means 35 updates were found. The ACTION indicators are:
>
> - **D** - Delete
> - **I** - Insert
> - **L** - LOAD LOG(YES)
> - **M** - Mass delete
> - **P** - LOAD REPLACE LOG(YES)
> - **R** - REORG
>
>   **Important:** Load or REORG activity that is found in the Db2 logs generates warnings if the details level report runs in the log-backward mode. No warnings are issued if the details level report runs in log-forward mode.
>
> - **U** - Update
> - **DCR** - Delete due to a compensation log record.
> - **DRI** - Delete due to referential integrity.
> - **DTR** - Delete due to a trigger.
> - **ICR** - Insert due to a compensation log record.
> - **IRI** - Insert due to referential integrity (may display during rollback processing).
> - **ITR** - Insert due to a trigger.
> - **UCR** - Update due to a compensation log record.
> - **URI** - Update due to referential integrity.
> - **UTR** - Update due to a trigger.

If an impact report has been requested and there is impact report data, the impact report is written to the GENIMP DD, if it is present in the JCL. Otherwise, a separate section will be included in the General

Report denoted by the heading IMPACT REPORT BEGINS. The data after this heading is identical in format to other data contained in the report, except that it is associated with the impact report data only (the data that occurred after your requested end date and time to the current date and time).

This is an example of a General Report:

```
                    ****************************************
                    DB2 LOG ANALYSIS- GENERAL REPORT: D91A
                    ****************************************

 LOG RANGE
 ---------
 START DATE    : 2013/09/19
 START TIME    : 15:35:00
 END DATE      : 2013/09/19
 END TIME      : 15:35:59

 FILTERS
 -------
 SHOW UPDATES   : Y
 SHOW INSERTS   : Y
 SHOW DELETES   : Y
 SHOW ROLLBACKS : Y
 CATALOG DATA   : N
    INCLUDE-TABLE...... PDMCWH.%

================================================================================

URID                   DATE        TIME      AUTHID   PLAN      CONNTYPE CONNID
-------------------- ---------- -------- -------- -------- -------- --------
00000000011C028633AF 2013-09-19 15:35:27 PDMCWH   DSNTEP2  BATCH    BATCH

MEMBER ID LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- ------------------------------------ -------------------- ------------
00000     ROCKNET1/D91ADB2 /CBFD1F0ADB10/0001 00CBFD1F0ADC55000000 RPTVARIN
--------------------------------------------------------------------------------

DBNAME   DBID  TSNAME   PSID  TABLE OWNER TABLE NAME         OBID  ACTION/COUNT
-------- ----- -------- ----- ----------- ----------------- ----- ------------
SWMDB91S 00268 TSRPRT01 00182 PDMCWH      LOG_ANALYSIS_TOOL  00200 I/1

================================================================================
URID                   DATE        TIME      AUTHID   PLAN      CONNTYPE CONNID
-------------------- ---------- -------- -------- -------- -------- --------
00000000011C02863688 2013-09-19 15:35:27 PDMCWH   DSNTEP2  BATCH    BATCH

MEMBER ID LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- ------------------------------------ -------------------- ------------
00000     ROCKNET1/D91ADB2 /CBFD1F0ADB10/0001 00CBFD1F0ADD22000000 RPTVARIN
 --------------------------------------------------------------------------------

 DBNAME   DBID  TSNAME   PSID  TABLE OWNER TABLE NAME         OBID  ACTION/COUNT
 -------- ----- -------- ----- ----------- ----------------- ----- ------------
 SWMDB91S 00268 TSRPRT01 00182 PDMCWH      LOG_ANALYSIS_TOOL  00200 U/1
```

IDMAP Report

The IDMAP Report is used primarily for support purposes, providing information about each of the tables in the General Report. When requesting technical support and supplying the IDMAP report with the job log, you help Software Support to make better decisions on the remaining documentation needs.

Important: This report will not print if you remove the INFOM DD from the JCL.

This is an example of an IDMAP report:

```
***********************************************
DB2 LOG ANALYSIS ALAGEN1 - IDMAP REPORT: LB1A
***********************************************

LOG RANGE
---------
START DATE    : 2021/08/03
START TIME    : 03:16:15.000000
END DATE      : 2021/08/03
END TIME      : 03:16:21.000000
```

```
FILTERS
-------
SHOW UPDATES   : Y
SHOW INSERTS   : Y
SHOW DELETES   : Y
SHOW ROLLBACKS : N
CATALOG DATA   : N
   INCLUDE-TABLESPACE. DB61225.T561225


==============================================================================
DBNAME TSNAME: DB61225.TS61225

DBID   PSID   PARTS PGSZ SGSZ  TYPE  DSSZ XML MAX PARTS
------ ------ ----- ---- ----- ----- ---- --- ---------
 0x1f8    0x2     0   4k   32k BASIC   0G NO          0

------------------------------------------------------------------------------
OWNER TABLE NAME: RSTEST.TB1225

OBID    PART TP EDITPROC FLD CSSID  VERS CUR CREATED RBA/LRSN DCC LF
------- ---- -- -------- --- ------ ---- --- ---------------- --- ---
   0x3    0 T             NO EBCDIC NO      0     0x558446e68d OFF OFF
                                                  0xda1b2c06f2a8
```

Impact Report

An impact report is really an extension to all other previously generated reports. The purpose of an impact report is to show log activity that occurred AFTER your requested end date and time range. The headings in an impact report mirror those headings in the non-impact section of the report. The impact report is written to the GENIMP DD (General report), DTLIMP DD (Details report), EXTIMP DD (Extended report), or the SUMIMP DD (Summary report) if those DDs are present in the JCL. If not, then the Impact report will be part of those reports (General, Details, Extended, or Summary) as a separate section.

This is an example of an Impact report from a General report:

```
                    ************************************************
                    DB2 LOG ANALYSIS- GENERAL REPORT (IMPACT): D91A
                    ************************************************

LOG RANGE
---------
START DATE     : 2013/09/19
START TIME     : 15:35:00
END DATE       : 2013/09/19
END TIME       : 15:35:59

FILTERS
-------
SHOW UPDATES   : Y
SHOW INSERTS   : Y
SHOW DELETES   : Y
SHOW ROLLBACKS : Y
CATALOG DATA   : N
   INCLUDE-TABLE...... PDMCWH.%


==============================================================================
URID                   DATE       TIME     AUTHID   PLAN     CONNTYPE CONNID
------------------- ---------- -------- -------- -------- -------- --------
00000000011C028650D2 2013-09-19 15:37:49 PDMCWH   DSNTEP2  BATCH    BATCH

MEMBER ID LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                 CORRID
--------- ------------------------------------ -------------------- ------------
00000     ROCKNET1/D91ADB2 /CBFD1F924809/0001 00CBFD1F928210000000 RPTVARIN
------------------------------------------------------------------------------
DBNAME   DBID  TSNAME   PSID  TABLE OWNER TABLE NAME        OBID  ACTION/COUNT
-------- ----- -------- ----- ----------- ----------------- ----- ------------
SWMDB91S 00268 TSRPRT01 00182 PDMCWH      LOG_ANALYSIS_TOOL 00200 I/1

==============================================================================
URID                   DATE       TIME     AUTHID   PLAN     CONNTYPE CONNID
------------------- ---------- -------- -------- -------- -------- --------
00000000011C028652DA 2013-09-19 15:37:49 PDMCWH   DSNTEP2  BATCH    BATCH
```

```
MEMBER ID LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
---------  -----------------------------------  --------------------  ------------
00000      ROCKNET1/D91ADB2 /CBFD1F924809/0001 00CBFD1F9284FA000000 RPTVARIN
--------------------------------------------------------------------------------

DBNAME   DBID  TSNAME   PSID  TABLE OWNER TABLE NAME       OBID  ACTION/COUNT
--------  -----  --------  -----  ----------- -----------------  -----  ------------
SWMDB91S 00268 TSRPRT01 00182 PDMCWH     LOG_ANALYSIS_TOOL  00200 U/1

================================================================================
URID                 DATE       TIME     AUTHID   PLAN     CONNTYPE CONNID
--------------------  ----------  --------  --------  --------  --------  --------
00000000011C028654B9 2013-09-19 15:37:49 PDMCWH    DSNTEP2  BATCH    BATCH

MEMBER ID LUWID= NETID/LUNAME/UNIQUE/COMMIT   LRSN                CORRID
---------  -----------------------------------  --------------------  ------------
00000      ROCKNET1/D91ADB2 /CBFD1F924809/0001 00CBFD1F928798000000 RPTVARIN
--------------------------------------------------------------------------------

DBNAME   DBID  TSNAME   PSID  TABLE OWNER TABLE NAME       OBID  ACTION/COUNT
--------  -----  --------  -----  ----------- -----------------  -----  ------------
SWMDB91S 00268 TSRPRT01 00182 PDMCWH     LOG_ANALYSIS_TOOL  00200 D/1
```

Mode File Report

The Mode File Report contains information about the current state of the mode file at various points of processing during the General and Detail Report runs.

The contents of the Mode File Report can change during this process as internal program modifications may occur. This is why the report is generated at different points of processing. The report is primarily to be used as a diagnostic tool for the vendor. You can also examine the report, however, to confirm that your settings are as expected. Much of the information in this report comes directly from panel values as you set them, and that information may be of interest to you. The Mode file output is written to the MODEFILE DD.

The mode file contains information passed from step-to-step of the General and Detail Report processes, and is the main mechanism for communication between these steps as to how the process is to execute. The report is broken down into two primary sections: user-defined values and product-determined values. Descriptions of the fields and their values can be found within this User's Guide.

This is an example of a Mode File Report:

```
                  **********************************
                  DB2 LOG ANALYSIS - MODE FILE REPORT
                       Generated From ALAMGR
                       Mode file version: 3.5
                  **********************************

*************
User Settings
*************
OVERRIDE PRODUCT DYNAMIC ALLOCATION... = N
OPTIONAL DATASETS PREFIX (VSAM)....... = PDMCWH.ALA.V350
OPTIONAL VOLUME SERIAL NUMBERS (VSAM). =
OPTIONAL DATASETS PREFIX (NON-VSAM)... = PDMCWH.ALA.V350
DISK UNIT NAME FOR DYNAMIC ALLOCATIONS = SYSDA
TAPE UNIT NAME FOR DYNAMIC ALLOCATIONS =
CONTINUE PROCESSING AFTER WARNINGS.... = Y
LOG READ PARALLELISM MAXIMUM.......... = 04
SEND INFORMATIONAL MESSAGES TO FILE... = N
HSM RECALLS........................... = Y
WRITE REPORTS TO DATASETS............. = N
SMS MANAGEMENT CLASS.................. =
SMS STORAGE CLASS..................... =
SMS DATA CLASS........................ =
SKIP FORMATTING (WHEN POSSIBLE)....... = N
USE DB2SORT (WHEN POSSIBLE)........... = Y
LEVEL TO SEND TO ZIIP (WHEN POSSIBLE). = 01
WAIT TIME IN MINUTES FOR CONTENTION... = 00
USE MEMBER NAME ON REPORTS............ = N
SUPPORT REPORTS OUTPUT FLAGS.......... = BCGIMST
USE FAST BACKUP DURING CONTINUOUS MODE =
ONE PHASE MODE - USE 64 BIT STORAGE... = N
 ONE PHASE MODE - USE WORK FILES....... = Y
```

```
TWO PHASE MODE - QSAM FOR ROWDATA..... = N
RESTART ASSIST ON CONTINUOUS MODE..... = N


**********************************************
Generate database activity report (general)
**********************************************
DB2 SUBSYSTEM NAME.................... = D91A
JOB IDENTIFIER........................ = V350
DATA SHARING MODE..................... = Y
SPECIFY LOGS.......................... = N
MISC FLAGS............................ = X
GENERAL REPORT OUTPUT FLAGS........... = Q
CONTINUOUS MODE....................... = N
RESOLVE STARTED UOWS.................. = N
POPULATE EXTFILE...................... = Y
64 BIT MEMORY THRESHOLD............... =
OVERRIDE SYSTEM GMT OFFSET............ = N
     WITH GMT OFFSET.................. = +00:00
INCLUDE LOB/XML DATA.................. = N
TRACK SCHEMA CHANGES.................. = N


**********************************************
Generate database activity report (details)
**********************************************
LOG READING MODE...................... =
COMMIT SCOPE.......................... =
GENERATE UNDO SQL..................... = N
GENERATE REDO SQL..................... = N
SET CURRENT SQLID..................... =
LOB PDS NAME.......................... =
ARCHIVE OUTPUT........................ = N
EXPERT OPTIONS........................ = N
CURRENT ROW BY KEY.................... =
WHERE CLAUSE BY KEY................... =
DETAIL REPORT OUTPUT FLAGS............ = B
REPORT ROW LENGTH..................... =
TREAT MASS DELETES.................... = I
CHAR DATA IN HEX...................... = N
SHOW VARCHAR LENGTH................... = N
DETAIL REPORT BYPASS SYSLGRNX......... = N


**************************
Quiet Time Specifications
**************************
QUIET TIME THRESHOLD VALUE............ = 00:02:00
SAVE QUIET TIMES...................... =


                   ***********************************
                   DB2 LOG ANALYSIS - MODE FILE REPORT
                          Generated From ALAGEN1
                          Mode file version: 3.5
                   ***********************************


*************
User Settings
*************
OVERRIDE PRODUCT DYNAMIC ALLOCATION... = N
OPTIONAL DATASETS PREFIX (VSAM)....... = RSTEST.ALA.V350
OPTIONAL VOLUME SERIAL NUMBERS (VSAM). =
OPTIONAL DATASETS PREFIX (NON-VSAM)... = RSTEST.ALA.V350
DISK UNIT NAME FOR DYNAMIC ALLOCATIONS = SYSDA
TAPE UNIT NAME FOR DYNAMIC ALLOCATIONS =
CONTINUE PROCESSING AFTER WARNINGS.... = Y
LOG READ PARALLELISM MAXIMUM.......... = 04
SEND INFORMATIONAL MESSAGES TO FILE... = N
HSM RECALLS........................... = N
WRITE REPORTS TO DATASETS............. = N
SMS MANAGEMENT CLASS.................. =
SMS STORAGE CLASS..................... =
SMS DATA CLASS........................ =
SKIP FORMATTING (WHEN POSSIBLE)....... = N
USE DB2SORT (WHEN POSSIBLE)........... = Y
LEVEL TO SEND TO ZIIP (WHEN POSSIBLE). = 01
WAIT TIME IN MINUTES FOR CONTENTION... = 05
USE MEMBER NAME ON REPORTS............ = N
SUPPORT REPORTS OUTPUT FLAGS.......... = BCGIMST
USE FAST BACKUP DURING CONTINUOUS MODE =
ONE PHASE MODE - USE 64 BIT STORAGE... = N
ONE PHASE MODE - USE WORK FILES....... = N
TWO PHASE MODE - QSAM FOR ROWDATA..... = N
RESTART ASSIST ON CONTINUOUS MODE..... = N
```

```
*********************************************
Generate database activity report (general)
*********************************************
DB2 SUBSYSTEM NAME..................... = TST1
JOB IDENTIFIER......................... = V350
DATA SHARING MODE..................... = Y
SPECIFY LOGS.......................... = N
MISC FLAGS............................ = X
GENERAL REPORT OUTPUT FLAGS........... = SF
CONTINUOUS MODE....................... = N
RESOLVE STARTED UOWS.................. = N
POPULATE EXTFILE...................... = Y
64 BIT MEMORY THRESHOLD............... =
OVERRIDE SYSTEM GMT OFFSET............ = N
     WITH GMT OFFSET.................. = +00:00
INCLUDE LOB/XML DATA.................. = N
TRACK SCHEMA CHANGES.................. = N




*********************************************
Generate database activity report (details)
*********************************************
LOG READING MODE...................... = A
COMMIT SCOPE.......................... = UOW
GENERATE UNDO SQL..................... = N
GENERATE REDO SQL..................... = Y
SET CURRENT SQLID..................... =
LOB PDS NAME.......................... =
ARCHIVE OUTPUT........................ = N
EXPERT OPTIONS........................ = N
CURRENT ROW BY KEY.................... =
WHERE CLAUSE BY KEY................... =
DETAIL REPORT OUTPUT FLAGS............ = D
REPORT ROW LENGTH..................... = 32752
TREAT MASS DELETES.................... = I
CHAR DATA IN HEX...................... = N
SHOW VARCHAR LENGTH................... = N
DETAIL REPORT BYPASS SYSLGRNX......... = N

*************************
Quiet Time Specifications
*************************
QUIET TIME THRESHOLD VALUE............ = 00:02:00
SAVE QUIET TIMES...................... = N

               ***********************************
               DB2 LOG ANALYSIS - MODE FILE REPORT
                     Generated From ALAGEN1
                     Mode file version: 3.5
               ***********************************

*************************************************
DB2 Log Analysis Tool - Fields Used Internally
*************************************************
LOG PROCESSING MODE.(ACTUAL).......... = B
LOG PROCESSING MODE (FORCED).......... = N
STOP PROCESSING....................... =
STOP MODULE........................... =
FAILED MODULE INDICATOR............... =
CONTINUOUS MODE RESTART NEEDED........ = N
IN DATA SHARING ENVIRONMENT........... = Y
USE EXTRACT FILE ALONE IN DTL1........ = N
ROW MOVEMENT FOUND.................... = N
REC3/LF3 AGAIN FOR CIK................ = N
PHASE 1 MERGE NEEDED.................. = N
DATA DECOMPRESSION NEEDED............. = N
UTILITY RECORDS PRESENT............... =
MERGE NEEDED.......................... = Y
MISC FILTERS, NO ID FILTERS........... = N
TABLE VERSIONING PRESENT.............. = N
NUMBER OF TRACKS FOR REFILE........... = 00001
TRACKS CALCULATED FOR FRMTFILE........ = 00001
TRACKS CALCULATED FOR SQLOUT.......... = 00001
COMMUNICATION FILE.................... =
STCK TO LRSN DELTA.................... = 00000000000000000000
AVERAGE EXTRACT FILE LENGTH........... = 00262
REFILE RECORD COUNTER................. = 0000000000013
SKIP FORMATTING IN USE................ = N
ENFM+ SET FROM BSDS................... = Y
ONE PHASE MODE - W/ DETAIL REPORT .... = N
```

```
                ONE PHASE MODE - LOAD................. = N
                ONE PHASE MODE - AUDIT MODE........... = N
                ONE PHASE MODE - CREATE ARCHIVE....... = N
```

Quiet Time Report

This report has a heading common to all output from the general report process. The log ranges and any filters are displayed. Additionally, part of the header will include this field:

- REQUESTED QUIET TIME THRESHOLD: HH:MM:SS

The requested quiet time threshold is your input, indicating the minimum amount of time that no activity was found to qualify for quiet time. Thereafter, either a single record indicating no quiet time was found will be displayed, or one to many quiet times will be shown. Each group of start and end quiet time records indicate a gap of 'quiet time', greater than or equal to your quiet time threshold, as shown:

```
    START QUIET TIME (YYYY-MM-DD HH:MM:SS) ... END QUIET TIME (YYYY-MM-DD HH:MM:SS)
            RBA/LRSN (nnnnnnnnnnnn)          ...          RBA/LRSN (nnnnnnnnnnnn)
```

The start quiet time value shows the beginning of a quiet time period, and the end quiet time value shows the end of that period. The RBA/LRSN record shows the start/end RBA or LRSN. If the subsystem is part of a data sharing group, the value will be an LRSN. Otherwise, it is an RBA. These RBA/LRSN values relate closely to the time period identified, but should be used primarily as identifiable boundaries of the quiet time. For example, it might be safe to recover to an RBA point somewhere in between the start/end RBAs, but not to either value exactly. Using a longer threshold value should yield a wide enough range to ensure safe recovery points.

Note that a subsequent record can appear to begin at the same time the previous record ended. This indicates that a unit of work affected your stated filters, but ran for a very short time period (less than a second). Nonetheless, two such records cannot be thought of as a contiguous period of quiet time as activity did occur (albeit briefly) between the two time periods. For example:

```
    START QUIET TIME (2014-09-19 15:35:34) ... END QUIET TIME (2014-09-19 15:37:49)
            RBA/LRSN (00000000011C0286409F) ...      RBA/LRSN (00000000011C028650D2)
    START QUIET TIME (2014-09-19 15:37:49) ... END QUIET TIME (2014-09-19 15:59:59)
            RBA/LRSN (00000000011C028656BD) ...      RBA/LRSN (00000000011C03682CFA)
```

The time period between 15:35:34 and 15:59:59 cannot be considered a single quiet time, but rather two separate periods of quiet time, with at least one intervening unit of work occurring at 15:37:49.

This is an example of a Quiet Time Report:

```
                    ********************************************
                      DB2 LOG ANALYSIS-QUIET TIME REPORT: R01A
                    ********************************************

    LOG RANGE
    ---------
    START DATE      : 2014/09/19
    START TIME      : 15:00:00
    END DATE        : 2014/09/19
    END TIME        : 15:59:59

    FILTERS
    -------
    SHOW UPDATES    : Y
    SHOW INSERTS    : Y
    SHOW DELETES    : Y
    SHOW ROLLBACKS  : Y
    CATALOG DATA    : N
      INCLUDE-TABLE...... PDUSR.%


    REQUESTED QUIET TIME THRESHOLD: 00:02:00
    START QUIET TIME (2014-09-19 15:00:00) ... END QUIET TIME (2014-09-19 15:35:27)
            RBA/LRSN (00000000011C036837D0) ...      RBA/LRSN (00000000011C028633AF)
    START QUIET TIME (2014-09-19 15:35:34) ... END QUIET TIME (2014-09-19 15:37:49)
            RBA/LRSN (00000000011C0286409F) ...      RBA/LRSN (00000000011C028650D2)
```

```
      START QUIET TIME (2014-09-19 15:37:49) ... END QUIET TIME (2014-09-19 15:59:59)
           RBA/LRSN (00000000011C028656BD) ...      RBA/LRSN (00000000011C03682CFA)
```

Storage Report

The Storage Report is automatically printed to the INFOM DD when running a General or Detail report, provided that the INFOM DD is in the JCL for the report step. If the INFOM DD is not in the JCL, then the report is not printed. The Storage Report shows storage activity by 24, 31, and 64 bit storage.

This is an example of a Storage Report:

```
                    *********************************
                       DB2 LOG ANALYSIS - STORAGE REPORT
                          Generated From ALAGEN1
                    *********************************
24 BIT GETMAIN HIGH WATER MARK: 000000000001112 (decimal)
24 BIT MEMORY LIMIT           : 000000009412608 (decimal)
31 BIT GETMAIN HIGH WATER MARK: 000000008098032 (decimal)
31 BIT MEMORY LIMIT           : 000001352663040 (decimal)
64 BIT GETMAIN HIGH WATER MARK: 000000000001665 (decimal, 1 = 1 meg segment)
64 BIT SYSTEM HIGH WATER MARK : 000000000001673 (decimal, 1 = 1 meg segment)
64 BIT MEMORY LIMIT           : 000000000010000 (decimal, 1 = 1 meg segment)
```

The report fields are:

24 BIT GETMAIN HIGH WATER MARK
   This is the accumulated total for the named load module (ALAGEN1 in the example above) for all GETMAINs done by the product for 24 bit storage, that is, below the 31 bit line. It does not include system generated GETMAINs, for example, buffers obtained automatically.

24 BIT MEMORY LIMIT
   This value comes from the z/OS control block named 'LDA' and is the limit for 24 bit storage.

31 BIT GETMAIN HIGH WATER MARK
   This is the accumulated total for the named load module (ALAGEN1 in the example above) for all GETMAINs done by the product for 31 bit storage, i.e., above the 31 bit line. It does not include system generated GETMAINs, for example, buffers obtained automatically.

31 BIT MEMORY LIMIT
   This value comes from the z/OS control block named 'LDA' and is the limit for 31 bit storage.

Note: All 64 bit storage fields are designated in increments of 1 megabyte segments, so a value of 1 means 1,048,576 bytes. All values come from the z/OS control block named 'RAX'.

64 BIT GETMAIN HIGH WATER MARK
   This is the accumulated total for the named load module (ALAGEN1 in the example above) for all IARV64 requests done by the product for 64 bit storage, that is, above the bar. It does not include system-generated above the bar requests.

64 BIT SYSTEM HIGH WATER MARK
   This is the accumulated total for the named load module (ALAGEN1 in the example above) for all IARV64 requests done by both the product and the system for 64 bit storage, that is, above the bar. The difference between this field and the one prior is the total number of system requests. For example, the value shown on this report is 7, and the prior value is 5, meaning system requests were for 2 megabyte segments.

64 BIT MEMORY LIMIT
   This value comes from the z/OS control block named 'RAX' and is the limit for 64 bit storage.

Summary Report

When you generate a General Report, Log Analysis Tool also generates a separate Summary Report. The Summary Report presents summary information about the table space, database, and number of

UPDATEs, INSERTs, and DELETEs for a table. The UPDATEs, INSERTs, and DELETEs fields are 11-digit fields. Final totals appear at the bottom of the report.

There is also a column (MD) that identifies if any mass deletes occurred. If at least one mass delete occurred, a "Y" appears in the MD column in the report. If no mass deletes occurred, the column is blank.

If an impact report has been requested and there is impact report data, the impact report is written to the SUMIMP DD, if it is present in the JCL. Otherwise, a separate section will be included in the Summary Report denoted by the heading IMPACT REPORT BEGINS. The data after this heading is identical in format to other data contained in the report, except that it is associated with the impact report data only (the data that occurred after your requested end date and time to the current date and time).

The example shows how multiple tables can belong to a table space and its database. The Summary Report provides subtotals by database and table space. Final totals at the bottom of the report represent the accumulated totals for all UPDATEs, INSERTs, and DELETEs of all databases being reported.

You can use this report in conjunction with the General Report to trace back who made changes to a table and determine the date and time at which these changes were made.

This is an example of a Summary Report:

```
                    *********************************************
                     DB2 LOG ANALYSIS- SUMMARY REPORT: DS2
                    *********************************************

 LOG RANGE
 ---------
 START DATE    : 2018/10/20
 START TIME    : 14:04:00
 END DATE      : 2018/10/20
 END TIME      : 14:07:00

 FILTERS
 -------
 SHOW UPDATES   : Y
 SHOW INSERTS   : Y
 SHOW DELETES   : Y
 SHOW ROLLBACKS : N
 CATALOG DATA   : N
   INCLUDE-TABLE...... PDUSER.JKTSLT


 ******************************
 * COMMITTED ACTIVITY        *
 ******************************

 OBJECT TYPE/NAME                        UPDATES     INSERTS     DELETES     MD
 --------------------------------------- ----------- ----------- ----------- --
 TABLE...... PDUSER.JKTSLT               50000       0           0
 TABLESPACE. JKTSL                       50000       0           0
 DATABASE... JKDB1                       50000       0           0

 OBJECT TYPE/NAME (COMPENSATION RECORDS) UPDATES     INSERTS     DELETES     MD
 --------------------------------------- ----------- ----------- ----------- --

 OBJECT TYPE/NAME (RI ACTIONS ONLY)      UPDATES     INSERTS     DELETES     MD
 --------------------------------------- ----------- ----------- ----------- --

 OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY) UPDATES     INSERTS     DELETES     MD
 --------------------------------------- ----------- ----------- ----------- --

 TOTAL SUMMARY REPORT
 ------------------------
 TOTAL UPDATES: 50000
 TOTAL INSERTS: 0
 TOTAL DELETES: 0


 ******************************
 * UNCOMMITTED ACTIVITY       *
 ******************************

 OBJECT TYPE/NAME                        UPDATES     INSERTS     DELETES     MD
 --------------------------------------- ----------- ----------- ----------- --
 TABLE...... PDUSER.JKTSLT               1           0           0
 TABLESPACE. JKTSL                       1           0           0
 DATABASE... JKDB1                       1           0           0
```

```
OBJECT TYPE/NAME (COMPENSATION RECORDS)  UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

OBJECT TYPE/NAME (RI ACTIONS ONLY)       UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY)  UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

TOTAL SUMMARY REPORT
--------------------------
TOTAL UPDATES: 1
TOTAL INSERTS: 0
TOTAL DELETES: 0

1*******************************
* ROLLBACK (ABORTED) ACTIVITY *
*******************************

OBJECT TYPE/NAME                         UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

OBJECT TYPE/NAME (COMPENSATION RECORDS)  UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

OBJECT TYPE/NAME (RI ACTIONS ONLY)       UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

OBJECT TYPE/NAME (TRIGGER ACTIONS ONLY)  UPDATES     INSERTS     DELETES     MD
---------------------------------------  ----------- ----------- ----------- --

TOTAL SUMMARY REPORT
--------------------------
TOTAL UPDATES: 0
TOTAL INSERTS: 0
TOTAL DELETES: 0
```

SYSCOPY Query

The SYSCOPY Query is automatically issued and printed to the INFOM DD when one of the following error conditions is encountered: ALAA440E, ALA447E, ALAA073, ALAA107, provided that the INFOM DD is in the JCL for the report step. If the INFOM DD is not in the JCL, then the query is not printed.

This is an example of a SYSCOPY Query:

```
                        *******************************
                        DB2 LOG ANALYSIS - SYSCOPY QUERY
                        *******************************

DBNAME    TSNAME   PARTS TYPE   ERROR ID
ALADB     TSLAT001 0000         ALAA447

DATE    TIME    ICTYPE STYPE STARTRBA            PITRBA                 DSNUM ICBK
                / DSN                                            SHRLVL OTYPE
                / OLDEST VERSION TTYPE   LOGGED RELCREATED
--------------------------------------------------------------------------------
140807 100333 F           00CD91AD6990F0000000 0000000000000000000 0000
                PDALAH.DB2.IMAGE.COPY.ANYTABLE.G2215V00        R      T
                +000                       Y        O

140807 100326 W           00CD91AD62D1C6000000 0000000000000000000 0000
                ALADB.TSLAT001                                        T
                +000                       Y        O

140807 100230 F           00CD91AD2DFF0F000000 0000000000000000000 0000
                PDALAH.DB2.IMAGE.COPY.ANYTABLE.G2214V00        R      T
                +000                       Y        O

140807 095924 F           00CD91AC7CA6F6000000 0000000000000000000 0000
                PDALAH.DB2.IMAGE.COPY.ANYTABLE.G2213V00        R      T
                +000                       Y        O

140807 095817 C     L     00CD91AC3CE150000000 0000000000000000000 0000
                ALADB.TSLAT001                                        T
                +000                       Y        O
```

SYSCOPY Report

The SYSCOPY Report is automatically generated within load module ALAGEN1 in a Log Analysis Tool general report run showing SYSCOPY entries for Db2 objects in the result set. Though this report is primarily intended for internal/diagnostic usage, you may find it beneficial.

This is an example of a SYSCOPY Report:

```
                   DB2 LOG ANALYSIS - SYSCOPY REPORT
                   ********************************

                   DBNAME: JKDB1    TSNAME: S18

ICTYPE STYPE TTYPE    OTYPE DSNUM LDSNUM HDSNUM LOGGED INSTANCE
------ ----- -------- ----- ----- ------ ------ ------ --------
    W                 T 00000  00000  00000     Y    00001

START_RBA           PIT_RBA             TIMESTAMP
------------------- ------------------- --------------------------
0000000001284C551D7A 00000000000000000000 2015-02-05-09.42.40.736162


***************************************************************
```

The report is grouped by database name and table space name. The fields listed are a subset of those contained within SYSIBM.SYSCOPY. Each entry begins with field ICTYPE and ends with field TIMESTAMP. There will be one to many entries per DBNAME/TSNAME, depending on what is in your current Db2 catalog. These entries exclude any ICTYPEs of 'I', 'F', or 'Q'. The purpose of the report is to show utilities or other activity that may have had an effect on the results of the run, or that may affect the detail report.

Transaction ID Report

This is an example of a Transaction ID Report (also called a Distributed Report), based on the URID of the log activity. Note that many of the fields in the report have no values. This is completely normal. Db2 logs whatever the application (client/server) fills in for these values (transaction ID, end user, transaction or application name, workstation name, and accounting string) during a distributed unit of work. Again, this product can only report on the information Db2 has made available in these DDF log records.

```
               *******************************************
               DB2 LOG ANALYSIS- DISTRIBUTED REPORT: DS1
               *******************************************

 LOG RANGE
 ---------
 START DATE   : 2014/10/20
 START TIME   : 14:24:00
 END DATE     : 2014/10/20
 END TIME     : 14:27:00

 FILTERS
 -------
 SHOW UPDATES  : Y
 SHOW INSERTS  : Y
 SHOW DELETES  : Y
 SHOW ROLLBACKS : N
 CATALOG DATA  : N
   INCLUDE-TABLE...... USER.TABLE

 --------------------------------------------------------------------------------
 URID: 0000000000C803BD8CB8     MEMBER ID: 00003
  TRANS ID   :
  END USER   :
  APPL NAME  :
  WRKSTN NAME:
  ACCTG STRG :DSN07010RS22SS1A        ROCKNET1SS1ADB2 DB2CALL BATCH    PDUSER
               PDUSER  QMF720  ACCT#
```

Table Versioning Report

The Table Versioning Report is automatically printed to the INFOM DD when Table Versioning has been detected.

This is an example of a Table Versioning Report:

```
                         *****************************************
                         DB2 LOG ANALYSIS - TABLE VERSIONING REPORT
                         *****************************************

DBID: 00268 PSID: 00093 OBID: 00094

DBNAME.TSNAME: DBTEST.TSTEST_1
OWNER.TBNAME: RSTEST.TEST_TABLE_1

COL# VER DATATYPE LEN   SCALE NULL DROP LOGPOINT            CURRENT VERSION
---- --- -------- ----- ----- ---- ---- --------------------
------------------
 003 001 INTEGER  00004 00000 Y    N                        in Db2 Catalog
 003 000 SMALLINT 00002 00000      N    00000000000000000000
 003 001 INTEGER  00004 00000      N    00DCF0745158F0000000 in repository


===============================================================================

DBID: 00284 PSID: 00349 OBID: 00350


DBNAME.TSNAME: DBTEST.TSTEST_2
OWNER.TBNAME:
RSTEST.TEST_TABLE_2


COL# VER DATATYPE LEN   SCALE NULL DROP LOGPOINT            CURRENT VERSION
---- --- -------- ----- ----- ---- ---- ------------------- ------------------
 No table versioning entries found


===============================================================================
```

Chapter 15. Log Analysis Tool files and sizes

The dynamic allocation of files is discussed in the following paragraphs. Dynamic allocation can be bypassed, however, and these data sets can then be hardcoded into your JCL. If you have not bypassed dynamic allocation, then you should read the following discussion completely. You can bypass dynamic allocation through settings on the "User Settings" panel of Log Analysis Tool.

Many data sets that were previously hard-coded in the product-generated JCL are now dynamically allocated by Log Analysis Tool. This means you no longer need be concerned about file sizing for particular runs of Log Analysis Tool for most (though not all) files. The DD EXTFILE data set still needs user specification, though you can allocate it one time with a large enough specification to handle all cases. The reason for this is that there is no way for Log Analysis Tool to determine ahead of time the number of records that will match your filters within the Db2 logs. This number of unknown records determines the needed size of the DD EXTFILE data set.

File names are generated based on settings on both your general and detail report panels and the "Defaults" panel. Part of each file name is predetermined by Log Analysis Tool. For example, an internal file identified by the DD REFILE is named as follows:

```
<optional non-vsam data set prefix or current user ID>.REFILE.<job identifier>
```

If the optional non-VSAM data set prefix is blank (from the "Defaults" panel), the current user ID is used instead. If the job identifier is blank (from the general/detail report generation panels), there is no suffix. Assuming the optional non-VSAM data set prefix is blank, current user is USERX01, and job identifier is D2005215, the file name would be USERX01.REFILE.D2005215. Assuming the optional non-VSAM data set prefix is GROUP.TEST1, and the job identifier is blank, the file name would be GROUP.TEST1.REFILE.

All files that are allocated as NEW are written to the SYSOUT DD named CFILES. You can browse that DD to view these allocations. If overrides become necessary due to B37 or D37 errors, this DD can be useful in allocating larger files.

Because many files are now dynamically allocated, they are also no longer readily apparent within the JCL. Almost all of these files are for internal use only, and are not documented or supported for external usage. The generated file names are based on values that you specify on the 'Defaults' panel. See the online help on that panel for more information. Only files documented in the tutorial topic: "External-use file mappings", are documented or supported for external use. If UNDO or REDO SQL is requested from a run of Log Analysis Tool, that SQL is placed into such a dynamically allocated file. That file's name is identified in the job log of such a run (as an informational message) so that you can use this file in other processes (such as SPUFI).

These new dynamic allocations are based on internal array sizing when possible. This means that Log Analysis Tool attempts to determine the proper size of these data sets based on internal arrays which are to be externalized to the files. In many cases, it is not possible to determine the exact size of dynamically allocated data sets. This can cause two possible outcomes: a B37-type abend on the file or re-allocation to tape (if so permitted). A re-allocation to tape will only occur if a tape unit name is provided on the 'Defaults' panel and the initial product-determined allocation could not be satisfied. A B37 abend (used here to mean any n37 type abend; could also be D37 or the like) can occur if the initial file size allocation can be satisfied by your system, but more space is actually required for all records. Again, not all dynamically allocated file sizes can be exactly ascertained. This addition of dynamically allocated files alleviates much of the guesswork involved in previous releases when attempting to determine proper file sizes.

If a B37 abend occurs on a dynamically allocated data set, the system will put a message indicating which file caused the failure. This message is included in your job log and is not controlled by Log Analysis Tool, but rather by the operating system. If these B37s become frequent, contact your product administrator to notify IBM as the calculations may need to be modified to prevent recurrences.

You can attempt to re-run the job by hard coding a proper JCL DD statement for the failing data set in the failing job step. Because the file names are user-modifiable, it is not possible to document what

they will be. The dsnames provided in the examples that follow merely identify a component of the internally built data set name. The definitive dsname must come from the B37 abend message. The name identified there must be the same dsname used on any hard-coded JCL DD allocation unless the data set is a temporary data set. In that case, any unique temporary name will do (for example, &&TEMPF1). The DDNAMEs are fixed in the JCL, and therefore, these can be used in conjunction with the B37 job log output to correctly add a hard-coded JCL DD statement. These DDNAMEs are identified in the next section

> **Attention:** Log Analysis Tool output data sets have hard-coded DCB attributes that cannot be overridden in JCL.

B37 abend errors

Follow this list according to failing step number and program execution. Again, this list is intended only for an extremely sporadic occurrence of B37s. If they become frequent, contact IBM customer support.

General report error in: //STEP2 EXEC PGM=ALAGEN1,...

```
//REFILE   DD   DSN=REFILE,
//   DCB=(LRECL=100,BLKSIZE=23500,RECFM=FB),
//   DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//CATFILE  DD   DSN=CATFILE,
//   DCB=(LRECL=400,BLKSIZE=23404,RECFM=VB),
//   DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//TEMPEXTF DD   DSN=&&TEMPEXTF,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//URIDF    DD   DSN=URIDF,
//   DCB=(LRECL=1300,BLKSIZE=13004,RECFM=VB),
//   DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
```

Detail report errors in: //STEP3 EXEC PGM=ALADTL1,...

```
//REIFILES DD   DSN=&&REIFILS,
//   DCB=(LRECL=100,BLKSIZE=23500,RECFM=FB),
//   DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//PLOGD    DD   DSN=&&PLOGD,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//PLOGDS   DD   DSN=PLOGD,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//EDICT    DD   DSN=&&EDICT,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//EDICTS   DD   DSN=EDICT,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//TEMPROWD DD   DSN=&&TEMPROWD,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
//TEMPROWS DD   DSN=&&TEMPROWS,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//   SPACE=(TRK,(?????,?????),RLSE)
```

Detail report errors in: //STEP4 EXEC PGM=ALADTL2,...

```
//FRMTFILU DD   DSN=FRMTFILU,
//   DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//   DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
```

```
//     SPACE=(TRK,(?????,?????),RLSE)
//TFRMTFIL DD   DSN=&&TFRMTFIL,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//SORTIN   DD   DSN=&&SORTIN,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//SORTOUT  DD   DSN=&&SORTOUT,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
```

Detail report errors in: //STEP5 EXEC PGM=ALADTL3,...

```
//FRMTFILE DD   DSN=FRMTFILE,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
```

Detail report errors in: //STEP6 EXEC PGM=ALADTL4,...

```
//SORTOUT  DD   DSN=FRMTFILS,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//SQLOUT   DD   DSN=SQLOUT,
//     DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
```

Load details into Db2 errors in: //STEP1 EXEC PGM=ALAGLOAD..., or //STEP1 EXEC PGM=ALALAUD..., (audit mode only)

```
//LOADFILE  DD  DSN=LOADFILE,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//LOADCARD DD   DSN=LOADCARD,
//     DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//DDLFILE   DD  DSN=DDLFILE,
//     DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
```

Process archived output errors in: //STEP2 EXEC PGM=ALAARC2,...

```
//CATFILE  DD   DSN=CATFILE,
//     DCB=(LRECL=400,BLKSIZE=23404,RECFM=VB),
//     DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//TFRMTFIL DD   DSN=&&TFRMTFIL,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
//FRMA     DD   DSN=FRMAFILE,
//     DCB=(LRECL=32752,BLKSIZE=32756,RECFM=VB),
//     DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//     SPACE=(TRK,(?????,?????),RLSE)
```

Correcting B37 abend errors

Procedure

- Use the B37 abend error information from the previous section to correct the error, according to the following example:

  Suppose you got a D37 abend on file SQLOUT in STEP6 of the detail report process. The D37 abend pointed to dsname = USERXX.SQLOUT.JOB1.

  1. You look up the current file allocation for data set 'USERXX.SQLOUT.JOB1' and see it was allocated with 100 tracks primary and secondary.

  2. You could delete that file and then re-run that step of the job (and any subsequent steps) by adding this DD statement to the JCL step (STEP 6) with a larger allocation amount as such:

```
//SQLOUT   DD   DSN=USERXX.SQLOUT.JOB1,
//    DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB),
//    DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//    SPACE=(TRK,(200,200),RLSE)
```

This hard-coded JCL DD statement will prevent any use of the dynamic allocation process (by Log Analysis Tool) for that data set only. In general, B37s should not occur on temporary data sets as they should be based on permanent data set allocations. However, if they do occur, these data sets will not exist after the abend, so base your larger file allocations on similarly allocated permanent data sets. You can also search out the DD CFILES for information on all newly allocated (DISP=NEW...) data sets in the job log.

The VSAM file (DD ROWDATA) can also be overwritten this way by creating it outside of Log Analysis Tool on your own, and hard coding this DD in the JCL as DISP=SHR only. This file is created during either the general report (STEP1) or during the detail report (STEP3), depending on whether the high speed option is actually utilized (that is, not just requested, but used per the determination of Log Analysis Tool). IDCAMS can be used to create this file outside of Log Analysis Tool, but must be done so using these exact specifications:

```
DEFINE CLUSTER (NAME(vsam.file.name)        -
               REUSE                         -
               VOLUMES(volume1,volume2,volume3) -  (optional)
               SHAREOPTIONS(1,3))            -
       DATA    (NAME(vsam.file.name.DAT)    -
               TRACKS(primary secondary)    -
               KEYS(32 0)                    -
               BUFFERSPACE(36864)            -
               FREESPACE(20 10)              -
               SHR(1,3)                      -
               CONTROLINTERVALSIZE(32768)    -
               SPANNED                       -
               RECORDSIZE(200 36864)         -
       INDEX   (NAME(vsam.filename.IDX)      -
               TRACKS(primary secondary)     -
               SHR(1,3))
```

The DD CFILES in your job log will contain information on the original allocation size of this file. If created outside of Log Analysis Tool, the file names must adhere to the naming conventions described previously.

Notes:

1. All files created in the general report are used in the detail report.

2. Some permanent files have associated temporary files for sorting or other purposes. Those files are identified here. Where temporary files are used, they are identified with the permanent files:

   - REFILE (temporary file is DD REIFILES)

   - ROWDATA (temporary files are DD TEMPROWD, TEMPROWS, SORTIN, SORTOUT)

   - EDICTS (temporary file DD EDICT)

- PLOGDS (temporary file DD PLOGD)
- FRMTFILx (temporary file DD TFRMFIL)

3. Rounding assumptions: 1 track of DASD = 47,000 bytes, 15 track per cylinder.

General report files

DD CATFILE

This file holds Db2 catalog information for each table reported on. CATFILE is VB, LRECL=400. It holds one record per table reported on, therefore, 5 tracks is usually sufficient. This file cannot be edited, however, you can delete it at the end of the job as it is regenerated the next time you run the job.

DD EXTFILE

This file holds log records that are read in the general report and that can be used in the detail report. It is effectively a log extract of matching Db2 log records, based on your filter criteria. This file cannot be edited, however, you can delete it at the end of the job unless this is a continuous mode job. The only time this DD is used in a one-phase job is when it is a continuous mode job.

Example 1: If 1,000 transactions are expected to be found, and a single table of fixed length 100 bytes is being filtered on, this file will need to be allocated to handle the maximum amount of all these transactions: (1,000 * (100 + 80)) = 180,000 bytes.

Attention: 80 is an approximate length of log record header information.

Therefore, a 4 track space allocation is sufficient.

Example 2: There are many tables are filtered on, and the expected number of transactions returned is 500,000, and the largest table row length is 200 bytes. To handle the maximum case, this is the formula: (500,000 * (200 + 80)) = 140,000,000 bytes, so approximately a 3,000 track space allocation would be needed, or 200 cylinders.

DD EXTFILB

This file is the backup of the extract file, it should be pre-allocated and retained from job to job. This file is for backup purposes in a continuous mode run only. You copy it into EXTFILE if, and only if, necessary in a restart procedure. See the online tutorial on "Continuous Mode" for more information on this file and restart using continuous mode. Log Analysis Tool maintains this file.

DD INFOM

This file holds the BSDS report, Control File report, IDMAP report, Maintenance report, Storage report, SYSCOPY query, SYSCOPY report, Table Versioning Report, and informational messages that are useful in resolving most problems during job execution.

BSDS report
   The BSDS report is automatically printed to the INFOM DD when running a General report, provided that the INFOM DD is in the JCL for the General Report step. If the INFOM DD is not in the JCL, then the report is not printed. The BSDS report provides two different types of information:

   • The remaining time until the RBA and LRSN reach their limit.

   • Job information that can aid in the support process.

Control File report
   The Control File report displays the information entered in the Control File for the specified Db2 subsystem. When requesting technical support and supplying the Control File report with the job log, you help Software Support to make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

IDMAP report

The IDMAP Report is used primarily for support purposes, providing information about each of the tables in the General Report. When requesting technical support and supplying the IDMAP report with the job log, you help Software Support to make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

Maintenance report

The Maintenance report lists the latest APAR and assembly date per module of all maintenance that has been applied. When requesting technical support and supplying the Maintenance report with the job log, you help Software Support in determining if any outstanding APARs need to be applied and make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

Storage report

The Storage report displays the information based on GETMAINS for 24, 31, and 64 bit storage. This report displays product usage of these areas and system limits. When requesting technical support and supplying the Storage report with the job log, you help Software Support to make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

SYSCOPY query

The SYCOPY query reports TABLESPACE activity from SYSCOPY when certain error conditions are encountered to INFOM DD. When requesting technical support and supplying the SYSCOPY query with the job log, you help Software Support to make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

SYSCOPY report

The SYCOPY report displays TABLESPACE activity from SYSCOPY for different events that may or may not have been logged such as REORG and LOAD utilities. When requesting technical support and supplying the SYSCOPY report with the job log, you help Software Support to make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

Table Versioning report

The Table Versioning Report displays the information from the table versioning repository when table versioning has been detected. When requesting technical support and supplying the Table Versioning Report with the job log, you help Software Support to make better decisions on the remaining documentation needs. If the INFOM DD is not in the JCL, then the report is not printed.

Informational messages

Informational messages are sent to this file when you have specified Y for 'Send informational messages to file' on the User settings panel.

DD MODEFILE

This is an internal file used for communication between the general and detail reports. You should never modify or delete the MODEFILE; Log Analysis Tool is solely responsible for creating and sizing the MODEFILE when you use the ISPF interface. If you encounter a MODEFILE error, regenerate the JCL using the ISPF interface.

DD MODEPARM

This file allows you to change certain parameters in the JCL without having to step through the ISPF panels. Not all parameters can be changed this way, only they ones found in this file. See "Create the modeparm file" on page 77.

DD REFILE

This file holds an internal snapshot of log records. One record per log record matching filter criteria. Length = fixed, 100 bytes. Example: If 1,000 transactions are expected to be found in the log, 1,000 REFILE records will be needed. 1,000 * 100 bytes = 100,000 bytes of space required. Therefore, 3 tracks are sufficient. You can delete this file at the end of the job as it is regenerated the next time you run the job.

DD UNCRE

This file is the continuous mode file containing information about uncommitted/unaborted units of work that may be reported the next time the job runs. This file also contains the information needed to start reading the Db2 log at the point the last job stopped reading the log.

DD UNCREB

This file is the backup of the continuous mode file. It should be pre-allocated and retained from job to job. This file is for backup purposes in a continuous mode run only. You copy it into UNCRE if, and only if, necessary in a restart procedure. See the online tutorial on "Continuous Mode" for more information on this file and restart using continuous mode. Log Analysis Tool maintains this file.

DD WORK*

These files are internal repositories used to assimilate the log records into full row before and after images for reporting. These files contain different types of data depending on how far along the job is and what options have been requested. You can delete this file at the end of the job as it is regenerated the next time you run the job.

Detail report files

DD EDICT

This file holds compression dictionary information in a log-forward process only. It is not present in log-backward mode. This file is used only if compressed data is found in the logs. Assume a maximum of 65,000 bytes per table space or partition of compressed data. Example: If 10 table spaces (all non-partitioned) are in your general report result set, a maximum of 650,000 bytes would be needed (65,000 * 10). A space allocation of 14 tracks is sufficient.

DD FRMTFILx

This file contains the displayable column data for each row affected by each log record reported on in the general report. (Where x represents different letters depending on the current sort of the file. This file should be sized the same for each value of x.) This is another unpredictable file size for any given run of Log Analysis Tool. A general rule here is to make the space allocation for this file at least as large as the DD PLOGD file. You can delete this file at the end of the job as it is regenerated the next time you run the job.

DD PLOGD

This file holds all the log records needed to materialize full row images for every log record reported on by the general report. A general rule is that this file must be allocated at least as large as the DD ROWDATA VSAM file (the data component), though it is not a VSAM file itself. Beyond that, it is impossible to determine in advance how many log records will be used in materializing full row images. Another useful guideline here is that if after the general report is run, the DD EXTFILE has more than one record in it, the DD PLOGD file can be allocated with the same space allocation as the DD EXTFILE. You can delete this file at the end of the job as it is regenerated the next time you run the job.

DD ROWDATA

This is the VSAM or QSAM file created in the **Defaults** option of Log Analysis Tool. This file holds multiple row images of each row reported on in the general report. The multiple row images reflect either the row's current image, pre-change image, or post-change image. Current images are not present in log-forward mode, however, for simplicity in this example, assume three row images exist for every log record. This file has 32 bytes of header information per record. To best size this file, or to know if resizing should occur prior to running the detail report, do this:

1. Run the general report only (**Generate details = N** from the general report panel)
2. View the summary report to determine the number of log records returned (this will be the total number of UPDATEs, INSERTs, and DELETEs from both committed and rollback activity). Assume total number = 5,000.
3. Determine the largest maximum row length of any tables in the result set. Assume maximum row length = 200 bytes.
4. Apply this formula:

```
(5,000 * (200 +94)) * 3 = 4,410,000 bytes
```

   Therefore, 80 tracks is sufficient.
5. If the current VSAM file is not allocated with this amount, use the **Defaults** option to re-create the VSAM file with larger space allocations, and then run the detail report.

You can delete this file at the end of the job as it is regenerated the next time you run the job.

DD SQLOUT

This file holds the generated undo/redo SQL, if any was requested. This is a fixed length, 80 byte file which can be used as input to IBM's Db2 SPUFI application. It should be sized at least as large as the DD PLOGD file.

DD URIDF

This file holds one record per unit of work reported on. Its size is totally dependent on the log data, like EXTFILE. If your site typically has few transactions per unit of work, this file's size could be large, like the EXTFILE. If your site typically has many transactions per unit of work, the size of this file could be smaller than the EXTFILE. URIDF is VB, LRECL=1300. This file cannot be edited, however, you can delete it at the end of the job as it is regenerated the next time you run the job.

Chapter 16. Log-forward versus log-backward processing

The details report of Log Analysis Tool can process the log in two distinct ways. It can process the log going forwards or backwards. The results from either process are very similar, but there are vast differences in how the processes arrive at the results. You can let Log Analysis Tool determine the mode it determines is best between log-forward and log-backward by using the automatic setting. The information that follows, however, provides background on how the type of action is determined, and provides a basis for why you might want to set a hard-coded value of either log-forward or log-backward. In V3.2, a new method was introduced allowing both methods of operation in a single run. This is explained fully in this section, and is named combo mode (that is, a combination of both methods in a single run).

Background

The general report is not associated with log-forward or log-backward processing. The general report, however, must be run prior to any details report, and the output from the general report is the input to either a log-forward or log-backward details report run.

A details report, regardless of mode: log-forward or log-backward, is run to display the details of a log record action. While the general report will show that an UPDATE, INSERT, or DELETE occurred, it will not show what data actually changed (that is, the data in the row of the Db2 table). You must run the details report to show the actual changes that occurred, and to generate any requested undo or redo SQL. The results in the details report from log-forward or log-backward processing must be the same. For example, if COL1 was changed from a value of 'A' to a value of 'B', both modes must show the pre-change column value as 'A' and the post-change column value as 'B'. The difference between these modes lies in how these values are determined.

The differences

Log-backward

Log-backward processing means that the starting point for resolving full row images comes from the current Db2 data pages. For every update reported on in the general report, a full row image may need to be materialized. The starting point is therefore the current row value from the appropriate Db2 data page. From that point, all log events affecting that row are applied backwards, resulting in the current, post, and pre-row images of the row at the time of the update. Log-backward processing can be thought of as running from the current time (backward) to the requested time range of the general report.

Log-forward

Log-forward processing means that the starting point for resolving full row images comes from image copies. From that point, the logs are read forward and all log events are applied to the starting points for each row. This results in post- and pre-row images only; there is no current row data available from this process.

Selecting a method

You can choose which process seems most appropriate for any given run, or you can request that the program chose the best method for any given run. There are certain situations where log-forward must be used, such as when REORGs have occurred on any table spaces involved and full row images are not present for any log activity involved therein. If while running a log-backward process, warnings are

generated indicating such situations, you can run a log-forward process against the same set of data (as generated from the general report). Factors contributing to a log-forward selection are:

- REORGs (unless full row images exist for all log records within the table space and the compression dictionary, if any, has not been rebuilt since any such log activity)
- mass deletes (if expanded)
- LOADs
- specifying logs
- user-forced log-forward
- dropped object mode
- LOB handling of updates or deletes
- span values, meaning both log-backward and log-forward are possible, but the timeframe from needed image copies to requested start time is shorter than the timeframe from requested end time to current time, with a weighting factor in favor of log-backward mode.

There are also situations where log-forward cannot be used. Log-forward requires, under most circumstances, an entry in SYSIBM.SYSCOPY for an image copy. If you have no such entry for a table space, that table space cannot be a candidate for log-forward processing. For example, table SYSIBM.SYSCOPY itself cannot be filtered on and processed in a log-forward manner because this Db2 catalog table does not have its image copies recorded in its own table (image copies for SYSCOPY are recorded in the Db2 DBET log).

If the requested log events are very current, a log-backward process will most likely be the most cost-efficient. If the log events are less current and it is likely that either REORGs have occurred or image copies were taken shortly before the log events, a log-forward process will be more efficient. If either method can work under the circumstances, the log-backward process will generally be more efficient. Keep in mind that table/table space filters should always be used in a log-forward process. Image copy processing can be very I/O-intensive, especially inline image copies. To minimize this impact, specify non-wild carded table or table space filters.

Remember: Keep in mind that the Db2 logs may need to be read from the created time of the image copy to the timeframe requested in the general report. If image copies are not frequently taken, this could represent a prohibitive amount of log reading.

The **Log reading mode** flag on the **Generate database activity report (details)** panel indicates to Log Analysis Tool which process to use. When set to A, Log Analysis Tool determines which method to use; when set to F, you are forcing Log Analysis Tool to use the log-forward process; when set to B, you are forcing the log-backward process; when set to D, Log Analysis Tool will choose log-backward unless the only way to successfully process the requested data and timeframe is through a log-forward process. When you run a combined general and detail report, you have the additional option of specifying a P, meaning FASTPATH processing. See the tutorial section on FASTPATH processing for more information.

If you have requested that Log Analysis Tool is to determine the best method, it is possible that it will chose log-backward, but return warnings of either ALAA061W or ALAA073W. When these specific warnings occur, and you have told the program to stop for any warnings (see Defaults panel), the program will automatically switch to log-forward mode. This is because log-forward will not be affected by the data still residing in Db2 buffers. If you do not wish this type of switching to occur, either change the setting on the Defaults panel for your warnings handling, or force the log reading mode to backwards (Log reading mode = B).

The only difference in the resulting reports and SQL between these two processes is the CURRENT row values. There are no CURRENT row values in a log-forward process. The Db2 data pages are never read. However, you can request current row values (by key only) in a log-forward process. The detail report itself shows only pre- and post- row images if no current row values are available.

A final word on log-forward processes: they require image copies as a resource in order to resolve full row images. If the image copies are not available, the product attempts to identify this situation and cease processing. However, there are circumstances that can make it very difficult for the product to ascertain that not all resources are available.

The primary reason for this would be the use of Db2 commands or utilities which basically tell Db2 that no recovery is available on the object. For example, the use of `REPAIR...NOCOPYPEND` or `START(db)...ACC(FORCE)` can get an object out of COPY PENDING status, but it will leave the object unrecoverable as well. The use of such commands or utilities can have a serious impact on Log Analysis Tool and the results will often be unpredictable. The use of these Db2 options bypasses the necessary resources (image copies, specifically) which allow this product to function in log-forward mode, and therefore any use of those options renders the affected Db2 objects unusable and unsupported by this product.

Combo Mode

When **Log reading mode=C**, you are requesting combo mode, a combination of log-backward and log-forward. This setting is similar to a setting of A, in that Log Analysis Tool makes the determination for you automatically. However, it may determine that some objects must be run in log-forward mode, but some objects can be run in log-backward mode. Therefore, it might run both modes internally. If all objects can be run in log-backward, it always chooses log-backward mode; if all objects must run in log-forward, it always chooses log-forward mode. It is only when a combination of both modes is possible does it choose to run both. The primary purpose of this option is for customers that wish to avoid running in log-forward mode as much as possible. Log-forward generally requires image copy access, and many customers want to avoid such access. So if some objects in the result set can be run log-backward, those objects do not incur the overhead of image copy access.

Important: Though the C option (combo mode) may reduce run times for some runs, it may actually increase it for others. There is a trade-off based on image copy access savings and extra Db2 log reads. A combination of log-backward and log-forward requires extra Db2 log reads because all the needed logs for both methods is required here. This trade-off between image copy access and more log reads affects whether the job runs faster or slower. For example, assume a case where 20 table spaces are in your result set, but only 2 of these table spaces need to run in log-forward mode. You save the overhead of 18 image copy accesses, minimally (that is, partitioning may introduce even more image copy access savings). But the overhead involved in now reading more Db2 logs to satisfy both a log-backward and log-forward run may result in more I/O, and it is unknown whether this extra log I/O may offset the I/O associated with the image copy access savings. Therefore, it is best to experiment with this setting first to see if your runs generally improve when using it.

Also note that the C option produces a result set similar to running in log-forward mode only in that current rows, normally available in log-backward mode, are not available. Log Analysis Tool does not provide current rows when this field is set to C, and a combo mode is actually used. If the option **Current row by key** is requested, current rows may still be included if unique keys are available on the tables. If log-backward mode only is chosen with this setting, then current rows are still available.

Remember that combo mode is only a request for the product to make the proper determination of modes; it does not guarantee any particular mode will be selected for any particular run, be it log-backward, log-forward, or a combination of both.

Combo mode has an added benefit if requested and used by Log Analysis Tool. In these cases only, if a table space encounters a warning message of ALAA061W indicating the log buffers have not yet been flushed, the table space is immediately transferred to log-forward processing. If combo mode is not being used, this condition results either in Log Analysis Tool completely stopping, or a complete transfer to log-forward mode for all objects. The warning message is still output, but the job does not result in condition code 4 and no data is lost.

Combo mode performs at the table space level. For example, if one table in a segmented table space has a mass delete performed against it, this requires a log-forward run for every table in that table space. Since the image copy must be accessed in this case for one table, it only makes sense to use that image copy for access to all tables in the table space. Log Analysis Tool identifies partition-level REORGs and can make some decisions based on this information. For example, it is possible that if a two-partitioned table space has one partition REORGed, and the other not REORGed during the relevant timeframe, Log Analysis Tool may determine it can use log-backward on the non-REORGed partition and log-forward on the REORGed partition.

Combo mode has certain restrictions:

- Expert options are not allowed in conjunction with combo mode.
- If specifying your own logs, log-forward is used regardless of combo mode requests.
- Dropped object mode uses only log-forward mode, regardless of combo mode requests.
- One-phase work file mode is mutually exclusive with combo mode. The panels do not allow such a combination. The reason for this is that combo mode would require more than the allowed 2 work files associated with such operations. Adding more work files would effectively make the work file method no different than standard two-phase mode with multiple files supporting the different repositories. One-phase 64–bit mode is allowed in conjunction with combo mode.
- Combo mode is never selected by Log Analysis Tool if LOB/XML activity is present.

If you specify combo mode and Log Analysis Tool uses combo mode, the objects selected for log-forward or log-backward modes are identified in the mode file report. See the subheading "Combo mode results" in the mode file report generated during detail report processing.

Suggestions

Minimize the number of Db2 logs needed

The most expensive and time-consuming part of the process is reading the Db2 logs. The fewer logs that must be read, the quicker the job will run. The critical factors are the requested time range in the general report, the current time, and the time of your usable image copies. For example:

```
General report time range : 05/06/14 09:00 - 17:00
```

```
Current time : 05/10/14 10:00
```

```
Useable image copy creation time: 04/15/14 14:00
```

From the example, 4 days span the general report time range to the current time. 26 days span the general report time frame to the image copy time. You should choose log-backward processing because 4 days' worth of log data is going to result in many fewer logs to read than 26 days' worth. The amount of processing time saved by making the correct decision can be substantial, as it would be in this example.

Where the number of logs required is similar for either process

Log-backward processing is almost always faster and therefore, less expensive. Log-backward processing uses current Db2 data pages to determine starting row images for every log record action. This means it is reading DASD, since all Db2 data must reside on DASD. Log-forward processing uses image copies for starting row images, and these image copies are most likely on tape. Tape is always slower to read than DASD. This is especially true if your site has compressed data. Compressed data requires the use of compression dictionaries. Again, these reside on DASD for log-backward processing, and most likely on tape (in your image copies) for log-forward processing.

Run Generate Details = Y reports with caution

This option is on the **Generate database activity report (general)** panel. This option allows you to concatenate the general and detail report JCL together in one job stream. The problem is that it does not allow you to size files after viewing the general report, because the detail report is immediately run after the general report finishes. It also uses the current settings from your detail report panel, and therefore, you may be running in a mode that is not efficient/appropriate for your current general report output. Certainly, you can and should use this option when you are comfortable with your detail report settings and the concepts of log-forward and log-backward processing.

Chapter 17. Generating SQL

Log Analysis Tool can generate SQL for REDO and UNDO purposes, though you should be aware of some specific scenarios.

SQL for UNDO/REDO purposes can be generated from the details level report. It will be included as a JCL step only if requested specifically from the details level report panel, or from the 'Generate Details' panel if generating a combination general and detail level report.

Important:

- It is VERY important that you understand the SQL generated within this product is not capable of reproducing application program scenarios. For example, suppose you have a table with 50 rows, all with the same values. If a program deletes one of those rows using a DELETE...WHERE CURRENT OF CURSOR statement, the log will record a single DELETE action. The SQL this product generates will be that single DELETE, but if run as a REDO, it would delete all 50 rows in the table. Because of this, it is imperative that you are aware of the tables involved and these tables should all be uniquely keyed. Without unique keys, the preceding scenario is possible and unexpected results and/or SQL failures are likely.

- It is VERY important that you understand that SQL cannot always be generated using this product. Some cases prevent generating valid SQL. For example, suppose you have a table with a single key column defined as AS IDENTITY GENERATED ALWAYS. That column must be used for the WHERE clause of any UPDATE or DELETE statement. However, in a REDO situation, any new INSERTs will generate different values for the key column. A subsequent UPDATE based on that INSERTed row cannot match the value of the key column as it exists in the log records to the newly generated value, causing the WHERE clause to fail. This is just one of many possible scenarios whereby SQL cannot be generated. Do not expect this product to be able to handle every possible recovery situation; it is not a stand-alone all-purpose recovery tool, it is a log analysis tool. Again, always examine the SQL prior to execution and remember that a complete knowledge of your data and table design is required to maintain your data integrity.

- For the SQL that Log Analysis Tool generates, there is a 32698 literal string byte limit. In the SQL WHERE clause, if more bytes are needed for the column, SUBSTR is used.

- VARBINARY strings will require a 16349 byte SUBSTR for hexadecimal binary strings longer than 16349 bytes. Two characters in the literal string of the SQL's WHERE clause will represent one byte of the binary string when BX is used.

- VARGRAPHIC string may require a SUBSTR of 16349 for VARGRAPHIC strings longer than 16349 double-byte characters because each VARGRAPHIC character may require two bytes in the literal string of the SQL. Additionally, VARGRAPHIC strings may require a SUBSTR of 8174 because each GRAPHIC character is represented by four hexadecimal characters when UX is needed in the SQL.

- Keep in mind that this product must be run PRIOR to any Db2 RECOVER operations. Db2 RECOVER will change the environment, and therefore, it may not be possible to generate SQL (or even show full row images) after this utility is run. For example, suppose you have a single table in a table space and a good full image copy at time 12:00. Bad updates occur after 12:00, and you realize this at time 13:00. If you run a Db2 RECOVER back to the image copy of 12:00, those log records from 12:00 to 13:00 for this object are now marked unavailable. This is due to the need to recognize partial recovery points in the log. The proper way to handle this is to run this product prior to the Db2 RECOVER. The environment is still proper to process the activity between 12:00 and 13:00. Once any reports and/or SQL are generated, then the Db2 RECOVER can be performed and actions associated with REDO SQL, for example, can be re-processed to any necessary points. The 'Misc Flag' on the general report panel does allow you to ignore these partial recovery points if Db2 RECOVER is run first, but usage of such requires expert-level understanding of Db2 and your local data/design. See the HELP on that field for more information.

Each SQL statement is numbered within comments, and the original associated URID and date/time fields are included in the comments as well. This SQL can be used as input to Db2's SPUFI, or cut and paste into other applications (for example, QMF) for execution. The SQL is written to the file associated with the DD named SQLOUT in the last step of the detail report JCL.

REDO SQL is, by definition, SQL that will redo an original action. In other words, REDO SQL re-applies changes made by an original action. UNDO SQL is, by definition, SQL that will undo an original action. In other words, UNDO SQL backs out changes made by an original action.

For REDO SQL, the SQL generated is identical to the original SQL. That is, for REDO SQL, an original UPDATE, INSERT, or DELETE action will be generated as an identical UPDATE, INSERT, or DELETE action, respectively.

UNDO SQL is slightly more complex in that the SQL generated is the opposite of the original SQL. That is, for UNDO SQL, an original UPDATE, INSERT, or DELETE action will be generated as an UPDATE, DELETE, or INSERT, respectively. For example, to undo an INSERTed row, you need to DELETE the row. To undo a DELETEd row, you need to INSERT the row. To undo an UPDATEd row, you also need to UPDATE the row, but the originally changed values need to be backed out.

Both REDO and UNDO SQL require that your affected tables are in the proper state in order to run successfully. Running either type of SQL requires that your tables looked exactly as they did at the time of their original execution. That is, the table data must be identical.

For REDO SQL, this means that some type of recovery has been performed. For example, the RECOVER utility may have been run to restore your data to a previous image copy. The REDO SQL could then be used to re-apply changes to a certain point-in-time.

For UNDO SQL, this means (most likely) that the data must not have changed at all since the reported changes here. That is, the actions reported on by this run of Log Analysis Tool are the most recent actions against the affected tables. Therefore, the table data is already in its proper state, without any needed modifications.

If the preceding conditions are not fully met, execution of the generated SQL will likely fail. Keep the following in mind as a general rule: it is most likely that you will require UNDO SQL generation for log activity that occurred very recently, and REDO SQL for that which occurred longer ago. It makes sense that you would want to UNDO actions that happened recently in that the current state of your tables involved likely matches the post-change row images for the log activity. Similarly, REDO SQL would be desired for events that occurred longer ago because it will likely be easier to recover to a point-in-time prior to the log activity and apply the REDO SQL.

The general idea here is that you need a point of consistency so that the WHERE clauses for both UNDO and REDO will be met when this SQL is run. The fastest way to get to that point should be the method chosen.

You should analyze each SQL statement and the order of execution of these statements to fully understand how running this SQL affects your Db2 tables. Modification of the SQL may be desired prior to execution. It is unlikely you'd need to modify the contents of any given SQL statement, but the omission/ inclusion of any given SQL statement should be scrutinized to attain the desired results, especially on those tables where unique keys do not exist.

WHERE clauses are built using unique index keys, unless otherwise directed (the user may override this behavior using detail report specifications). If there exists more than one unique index on the table, the product will use only one set of unique index keys. It determines which index will provide a path of 'least resistance' for SQL execution. For example, suppose TABLEX has two unique set of keys; the first on column EMP-ID, an INTEGER field, and the other index on EMP-NO and EMP-ROWID, defined as INTEGER and ROWID GENERATED ALWAYS, respectively. The product will only use EMP-ID in the WHERE clause, to prevent possible problems with the GENERATED ALWAYS aspect of the other key. If only a single unique index is available, that set of keys must be used regardless of any column attributes. If no unique index is available, all columns must be used in the WHERE clause. It is strongly not recommended to generate SQL for any such table without a unique index.

A COMMIT statement will be added every x number of SQL statements, depending on the value assigned to COMMIT SCOPE field on the details level report panel. For example, if x=100, after 100 SQL statements, a COMMIT is performed. If x=000, no COMMITs are added to the SQL.

Rolled back units of recovery are never included in SQL generation. Since all the work was already backed out, there would be nothing to UNDO or REDO.

Uncommitted units of work will be included in all SQL if you requested its inclusion in the general or detail reports. Execution of such SQL requires extra consideration and evaluation if requested as applying such could leave your data in an inconsistent state. All SQL included as part of an uncommitted unit of work can be identified by doing a find on 'UURID' (meaning 'uncommitted unit of recovery ID') located in the commented header of each SQL statement.

In cases of very long column data (> 254 bytes), it is necessary to generate multiple SQL statements from one action. For example, assume a row with a long varchar column is deleted, and you wish to UNDO this action. This will result in an INSERT statement which will recreate the row. Since only 254 bytes can be included in the INSERT statement for the long varchar column, it is initially set to a subset of its true data value. Immediately after the INSERT, subsequent UPDATE statement(s) will concatenate the long varchar column with its current contents and the remaining data. As many UPDATE statements are generated as needed to completely reproduce the columns value.

For similar situations occurring with an original UPDATE statement, the product will attempt to minimize these extra SQL statements by determining if a long data column has actually been modified. If the long data has not changed, the column itself will be omitted from the SQL statement avoiding the need for these extra SQL statements.

Some log records are generated due to referential integrity definitions. These log records are no different from any other, but are considered the results of referential constraints (see your Db2 manuals for full descriptions of referential integrity/constraints). The SQL generated from these referential constraints must be treated differently when applying UNDO or REDO SQL.

No user action is necessary to handle the issues of referential integrity, the product automatically takes care of it. The explanation here is just to aid the user in understanding the proper handling of this type of SQL. Basically, for REDO purposes, all referential-constraint-type records will be omitted. That is, only the originating SQL statement will be included, and this will activate all other referential-constraint-type activity (exactly as it originally occurred). For UNDO purposes, the order of the SQL is slightly modified to ensure any parent key rows are restored before issuing any SQL against dependent key rows.

Referential constraints are noted in the generated SQL output as being either UPD-RI or DEL-RI, meaning an UPDATE or a DELETE due to referential integrity definitions, respectively. INSERTs cannot be the result of referential constraints.

Important: If referential constraints exist on any tables within the result set, the SQL should never be executed unless all actions are included (that is, UPDATEs, INSERTs, and DELETEs must not be filtered out). Also, all tables involved in the referential table set should be included.

As previously mentioned, for REDO purposes, no records marked UPD-RI or DEL-RI will be included in the SQL. This is because the originating action (the original SQL which caused the referential constraint SQL) is to be re-done, which will once again activate those referential constraints, so no extra work is needed. For UNDO purposes, those records marked as UPD-RI and DEL-RI will be included, but only after the originating SQL is undone. This is necessary because referential integrity would otherwise prohibit the activated actions without the existence of the associated parental key values.

Log records are also generated due to triggers firing. These log records are no different than any other, but are considered the results of triggered SQL statements (see your Db2 manuals for full descriptions of triggers). The SQL generated from these triggers firing must be treated differently when applying UNDO or REDO SQL.

Some user action is necessary to handle the issues of triggers firing, as the product cannot interpret the logic behind the triggers or if there is a trigger in place that will act as the opposite of the originating trigger. The explanation here is to aid the user in understanding the proper handling of this type of SQL. Db2 allows for two different types of triggers. The user will need to know which type is present in the tables that REDO / UNDO SQL is being generated for. If Traditional Triggers are present and REDO SQL is requested then the user will probably want to choose X to exclude the actions that originated from the trigger. This is because when the original statement is redone, the trigger will fire again doing all the necessary work. If Instead Of Triggers are present and REDO SQL is requested then the user will probably want to choose Y to include the statements that were caused by triggers because the original statement was against a view and Db2 transformed the original statement into the triggered statement(s). In this case the original statement will no longer be available. For UNDO purposes, the user has the option of

excluding the triggered SQL statements or having them treated the same as the other SQL statements that are candidates for UNDO.

Triggered SQL statements are noted in the generated SQL output as being UPD-TR, INS-TR or DEL-TR, meaning an UPDATE, INSERT, or DELETE due to a trigger firing, respectively.

Important: If triggers exist on any tables within the result set, the SQL should never be executed unless all actions are included (that is, UPDATEs, INSERTs, and DELETEs must not be filtered out). Also, all tables involved in the trigger definition should be included.

As previously mentioned, the user can opt to include or exclude the records marked UPD-TR, INS-TR, or DEL-TR from the generated SQL by choosing Y to include these actions or X to exclude these actions.

If impact report was requested at the detail report level, SQL is not included pertaining to any impact data. An impact report is for analyzing the impact of what occurred after your requested time range, not for generating extra SQL. If the SQL is desired after your requested time range, the proper method to handle that is to increase your time range, not via an impact report. See the tutorial topic on "Impact Report" for more information.

Chapter 18. Recovering dropped objects

Log Analysis Tool can recover dropped objects when you use dropped object mode.

Steps for dropped object recovery

1. Create the objects that you are recovering.
2. Restore the dropped objects using DSN1COPY.
3. Take an image copy of the recreated objects.
4. Create REDO SQL using the old and new object IDs.
5. Apply the REDO SQL to the recreated objects.

Using dropped object mode

Dropped object mode is initiated by including old object IDs in an object ID list from the "Object Filters by IDs" panel from the general report. You are considered in standard mode if old IDs are not present.

Important: If you want LOB data, you must also include the LOB table IDs along with the base table IDs. The LOB table IDs cannot be determined automatically by the product in dropped object mode.

There are two types of dropped object mode:

1. The entire table space has been dropped which results in deleted SYSCOPY records.
2. One or more tables in a table space are dropped, which has no effect on SYSCOPY records.

For the latter case where the entire table space is not dropped, Log Analysis Tool can use current SYSCOPY information to gather the needed resources (that is, image copies). For the former case where the entire table space is dropped, extra steps are required to allow Log Analysis Tool to gather the needed resources, and those are described in the following section. In either case, the object that was dropped must be re-created exactly as it previously existed prior to running in dropped object mode.

Important: If you included LOB data, dropping only the table also drops the associated LOB table space(s). In that case, you must follow the rules for a dropped base table, but you must also follow the rules for a dropped table space regarding the LOB data. In other words, while the base table space does not require a new image copy, the LOB table space does require a new image copy, and must follow the rules for dropping an entire table space.

To use dropped object mode in the case where an entire table space was dropped, the following must be true:

1. The dropped object must have been restored (redefined to Db2 and recovered to an old image copy), and
2. A new full image copy must be taken of that restored object. The new full image copy must come from the COPY utility only, not from a LOAD or REORG.

These prerequisites are necessary because Db2 catalog information must be available on both the restored object's attributes (SYSCOLUMNS data) and its history (SYSCOPY data). If any deviation from the above is attempted, the results will be unpredictable and unsupported.

Important:

1. If you run a detail report after you ran a general report in dropped object mode, only log-forward mode is acceptable. Though not forced, it is most likely that you want only REDO SQL.
2. When running in dropped object mode, all IDs must be present. That is, the translation of old to new IDs requires that you specify the DBID, PSID, and OBID (that is, OBID is not optional in dropped object mode).
3. It is most likely that you specify only a single DBID/PSID group for the run as the start date and time will likely need to reflect a single image copy timestamp for that dropped object (table space or table).

4. If LOB/XML data types are present in your dropped objects, the old and new IDs of the associated LOB/XML 'auxiliary' table(s) must also be defined.

If a partitioned table space is dropped, the recovery should be done at the partition level if image copies are taken at the partition level. If image copies are taken for the entire table space (that is, all partitions are image copied together into a single dataset), then recovery can proceed as with a non-partitioned table space. The reason for this is that image copies taken at the partition level are likely done at different times, and each partition would require its own timeframe, and therefore, its own recovery job.

Partition level specification can be done using ID filters by making use of page level filters. For example, if you have 4 partitions, these pages may denote each partition:

- for partition 1: pages 00000002-000FFFFF
- for partition 2: pages 00100002-001FFFFF
- for partition 3: pages 00200002-002FFFFF
- for partition 4: pages 00300002-003FFFFF

The actual partition specification within a page number depends on many table space-defining factors. See the Db2 Diagnosis Guide for more information on how partitions are identified within page numbers.

Important: If running a case where the entire table space was dropped, the following apply:

1. Set 'Misc flags=X' on the general report panel to bypass use of SYSLGRNX.
2. If certain log records are found, the process may not work because needed resources are no longer available. For example, if during the entire relevant timeframe, a LOAD LOG YES was performed, Log Analysis Tool will encounter those log records, but will not be able to reconcile what types of records they are (because Db2 does not log that information). A REORG LOG YES can generate log records that are identical to a LOAD LOG YES, and without the needed information in SYSCOPY, which was deleted when the table space was dropped, Log Analysis Tool cannot determine if it was a REORG or a LOAD. An error message will be generated, and further processing will not be possible.
3. It is imperative that you do not include log activity prior to the last full image copy in this process, or results are unpredictable and unsupported. In cases of a dropped table space, the last full image copy will be used as the starting point for resolving all subsequent log activity. This 'last full image copy' will actually be the result of a DSN1COPY (see explanation above of required dropped table space actions). Because of this, the START RBA in SYSCOPY will not properly reflect the actual time of the original last full image copy. This prevents Log Analysis Tool from being able to make determinations about any actions that occurred prior to that image copy. This should not present any problems when using this scenario; it is completely unnecessary to include any log activity prior to that last full image copy in recovering the table space and to do so is considered a user error. Since the drop deleted such information from SYSCOPY, it can be difficult to determine the time the original last full image copy was taken. Products like Db2 Recovery Expert can assist in such efforts (though these products must be in use prior to the drop, not after the fact).

Chapter 19. One-phase mode

You can choose to operate Log Analysis Tool in one-phase mode, using 64 bit storage or work files to store data, as opposed to the normal two-phase mode where that same data would be saved in permanent files.

On the "User settings" panel, you can request that the product make use of 64 bit storage, or that it make use of work files. With either, you are running in one-phase mode (as opposed to two-phase mode, which is the standard mode for the product). The description and reasons for this are outlined following.

Using 64 bit memory/work files means the product will place certain data in to 64 bit memory/work files, rather than using standard I/O. This can improve performance greatly. When electing this option, however, certain changes occur related to how the product operates. Since the product uses memory or work files in this case, there is no permanence of the data contained within that memory or work files. Once the job stops, the data is lost. Therefore, you cannot run a separate general report, followed by a detail report.

You must run a combined general/detail report only (or just a general report with no detail report). When using the standard method of file I/O, you can run separate general and detail reports because the files generated in the general report are permanent, and can subsequently be used as input to a separate detail report. The same holds true if you want to LOAD output data from the general or detail report back into Db2. In one-phase mode, you must do this during the one-phase job; you cannot first run a general and/or detail report job, and then subsequently run the LOAD job. All these steps must be performed during the single step one-phase mode job because the required input to the LOAD process will be lost in one-phase mode as soon as the job completes.

Hence, the concept of one-phase mode versus two-phase mode. Using 64 bit memory or work files defines a one-phase approach, otherwise you are in two-phase mode. One-phase mode does not incorporate all the features of the product. It includes only the following processes:

- general report run
- detail report run
- LOAD data into Db2 run

SQL and archiving operations can still be used after a one-phase mode job runs. SQL is always externalized, so this data remains permanent. Archiving, by definition, creates permanent files for future use, so it too falls outside the restraints of one-phase mode. You still create SQL and/or archives during the one-phase job, but use of their outputs at a later time is still permitted.

Certain product features are limited or eliminated when using one-phase mode:

- Re-running against current data only from the "Process archived output" option is not possible because there is no current data after you run in one-phase mode, therefore, you cannot run this option.
- Running while including LOB/XML data is supported when running in work file mode but not in 64 bit mode. This is true for several reasons, primarily because LOB/XML is likely to completely exhaust all storage, and its usage, such as viewing it after reports are generated, requires data permanence.
- Expert options are also unavailable in one-phase mode. Again, this is due to the fact that expert options require a separate general and detail run, and this is not possible in one-phase mode. The expert options are specified on the detail report panel, and cannot be specified before the general report runs.

In general, use 64 bit memory when the amount of data to be reported on is considered low to medium in volume. If the amount of data to be reported on is excessive (millions of log records), use work files instead. Memory will quickly be exhausted if you attempt to use 64 bit storage in a high volume scenario.

Using work files in a high volume scenario requires that you estimate the size of these work files. Note that they cannot be allocated to tape due to sorting issues; these files must be on DASD, but can span multiple volumes (note that one-phase mode is meant for performance, and using tape data sets is contradictory to that goal, so tape data sets can only be used in two-phase). Sizing for these files depends on the amount of activity expected. Row sizes also play a role here. Though work files can contain several different files during any given run of the product, only the sizing of the largest of these files need be

considered. For example, the work files may contain Db2 log record extracts, and then the product's internal ROWDATA file data, and finally the product's internal FRMTFILE data. To properly size the work files, it is clear they must be large enough to hold the largest of any of these files.

In general, the FRMTFILE file will be the largest repository. So that is the file to focus on for sizing. It is best to think of sizing in terms of maximum file sizes, rather than average file sizes to avoid B37 abends.

Note: The FRMTFILE is only generated if detailed-level output is requested, and a detail report, archiving, or loading into Db2 using audit mode is requested. Otherwise, the FRMTFILE is not created. In such cases, skip the following section and see the section on using the ROWDATA output for sizing.

Using the FRMTFILE for sizing:

Assume your largest row sizes are 1000 bytes long. Also assume you expect to report on a maximum of 10 million log records (that is, 10 million updates were done during your requested time frame). The formula is as such for file sizing:

```
A = 10 million records (from example, number of log records)
B = 168 bytes (length of a FRMTFILE header record)
C = 44 bytes (header length of a FRMTFILE data record)
D = 1000 (from example, length of a row)
E = number of columns per table (this is an average amount here, assuming 50
    columns for this example. That's 50 columns making up the full 1000 byte
    row length, from the example only)

X = (A*B)...total bytes needed for FRMTFILE headers
Y = Y = A*E*(C + (D/E))...total bytes needed for FRMTFILE data records
Z = 3*Y if current rows are included (pre-row image, post-row image,
    current row image)
Z = 2*Y if current rows are not included (pre-row image, post-row image)

Cylinder allocation for this example (assume 750,000 bytes per cylinder)
Cyls = (X plus Z)/750,000

So for our preceding example:

X = (10,000,000 * 168) = 1,680,000,000
Y = (10,000,000 * 50 * (44 + (1000 / 50)) = 32,000,000,000
Z = 32,000,000,000 * 2 = 64,000,000,000 (assume no current row images)
CYLS = (1,680,000,000 plus 64,000,000,000)/750,000 = 87,574
```

Note: It is never recommended running against this amount of data; the preceding figures are used only for calculation purposes and to show how much space is needed for such a run.

Using the ROWDATA for sizing:

Assume your largest row sizes are 300 bytes long. Also assume you expect to report on a maximum of 10 million log records (that is, 10 million updates were done during your requested time frame). The formula is as such for file sizing:

```
A = 10 million records (from example, number of log records)
B = 94 bytes (length of a ROWDATA header record)
C = 300 (from example, length of a row)

X = 3*(A*(B plus C)) total size of ROWDATA if current rows are included
    (pre-row image, post-row image, current row image)
X = 2*(A*(B plus C)) if current rows are not included
    (pre-row image, post-row image only).

Cylinder allocation for preceding example (assume 750,000 bytes per cylinder)
Cyls = X/750,000

So for our preceding example, assuming no current rows:

X = 2*(10,000,000 * (94 plus 300)) = 7,880,000,000
CYLS = 7,880,000,000 / 750,000 = 10,507
```

Note: It is never recommended running against this amount of data; the preceding figures are used only for calculation purposes and to show how much space is needed for such a run.

Here are a few examples for one-phase mode usage:

If you want to run a combined general/detail report, and LOAD the results of that into Db2 for auditing purposes, you can run in one-phase mode, and specify all your general/detail/LOAD options at one time. This will generate a jobstream with all needed processes to complete the task in one step. You could not, however, run a combined general/detail report, and at a later time, LOAD the results of that job into Db2. It must be done all in one step.

Note: The loadfiles generated by the product are still permanent, so once created, you could process them separately, if so desired. This allows you to LOAD this data outside of the product, or process the product generated load files by downstream applications.

If you want to run a combined general/detail report, and generate REDO SQL within that process, you can run in one-phase mode. The SQL will still be written to a permanent file, and you can process that SQL at a later time.

If you want to run a combined general/detail report, and generate REDO SQL within that process, you can run in one-phase mode. However, you cannot now change your mind and re-run against current data to generate UNDO SQL using the "Process archived output" option. You would need to re-run the entire job from the beginning, requesting UNDO SQL.

If you want to run a general report only, you can run in one-phase mode. However, you cannot subsequently request a detail report to further process the output of that general report. No files exist for input to the separate detail report.

Chapter 20. LOB/XML support in SQL and LOAD operations

LOB and XML data types are now supported for some operations. LOB data can be included in SQL and LOAD operations if using Db2 V9 or later. XML data can be used in SQL, and in LOAD operations. XML data issues are only relevant for Db2 V9 and later. Both of these potentially large objects require the Db2 "file-by-reference" feature for support in SQL and LOAD operations, and these version restrictions reflect Db2's current support level in these areas.

Important: In the following text, the term "LOB" is used to refer to either a LOB or XML column. There are certainly many differences in these two data types, however, they are both tracked by this product in a similar way, and both are effectively auxiliary tables to the base table containing their column definitions. Therefore, the use of the term "LOB" is just this product's denotation of these types of objects and how they are processed. When differences between LOB and XML data types become relevant, they are noted.

Restrictions

- LOB types cannot be used in audit mode operations within **LOAD data into DB2**, option 3 off the primary menu. If you want to audit LOB data using the LOAD option, you must first create your own audit/backup table containing all the same attributes of the original table. The Db2 LOAD utility can then LOAD both the table base columns and LOB column(s) into that audit/backup copy. This copy must include the auxiliary LOB table itself.

- LOB data cannot be archived using this product's archiving facility. The actual data associated with the LOB column is not archived; only the Db2 internal locator values are, which are of little value.

- LOB data used in SQL must be executed using the supplied Fast Apply SQL (FAS) option, **Execute dynamic SQL in batch**, primary menu option 5. LOBs are only supported in SQL via the use of file-by-reference, and the generated SQL contains a proprietary format that can only be used by FAS. This format is required to allow dynamic SQL execution using file-by-reference. Db2 options like SPUFI and DSNTEP2 cannot handle file-by-reference, and will result in an SQL error if attempted.

- When LOBs are requested for inclusion, you cannot use the "Current row by key" option. LOB data is not a candidate for current row by key, and these settings are mutually exclusive.

- High speed mode is mutually exclusive with LOB data inclusion.

- One-phase 64 bit mode is mutually exclusive with LOB data inclusion, however, one-phase work file mode can be used with LOB data.

- Any type of the following actions are not supported: LOAD, LOAD REPLACE, and MASS DELETEs (possible against XML data only). An error message will be generated if any of those conditions are found in conjunction with LOB/XML data.

- LOB/XML columns cannot be specified in advanced filters operations.

LOB data can be tracked via Log Analysis Tool only if LOG=YES has been specified for the LOB table space. If LOG=NO has been specified, there will be no LOB data to track in the Db2 log records, and therefore, requesting such tracking will not yield any LOB data. For XML data, the XML table space will have the same logging attribute as its base table's table space.

LOB data can be tracked via Log Analysis Tool by setting field **Include LOB/XML data** on the general report panel to Y. Due to the overhead involved in tracking LOB data, it should be requested only when ABSOLUTELY necessary. Performance will be impacted by LOB data tracking. High speed mode cannot be used with LOB data inclusion. Archiving processes, both with actual archives and current data only, cannot be used with LOBs. Re-running any detail report involving LOBs must be performed using primary menu Option 2 only; it must not be done via primary menu Option 6. LOB tracking can only be accomplished in two phase mode or in one-phase mode with work file mode; it requires multiple files such as EXTFILE, PLOGDS, and so on, and the permanence of those files. Use of one-phase 64-bit mode with LOBs is prohibited.

Though mentioned previously, it cannot be stressed enough that LOB processing is an extremely resource intensive option, and should only be performed on a very limited set of activity. Requesting system-wide LOB activity reporting will almost certainly fail for various reasons (that is, DASD and memory constraints). In fact, Db2 recommends that you not log activity against LOBs at all. If you do log such activity, and wish to use LOB options now available in Log Analysis Tool, keep in mind and realize the negative performance implications associated with LOB reporting. Generating SQL within the context of LOBs will also negatively affect performance, and in a very large way. Do not generate or execute SQL using LOB data without full knowledge and consideration of the amount of data involved.

LOB data tracking, in the context of Log Analysis Tool, means that LOB data that has changed is tracked. It does not track LOB data that does not change, as that information is not available in the logs. For example, assume an UPDATE is done to a table with a LOB column, but the LOB column itself is not modified. Therefore, there is no LOB column data to track. However, if the UPDATE done actually modifies the contents of the LOB column, then there is LOB data to track in the Db2 logs. Anytime an INSERT or DELETE is performed, there is LOB data to track as those actions always affect the entire row. Again, LOB data tracking can be very resource intensive, so not tracking unchanged LOB columns is done to reduce overall resource consumption.

LOB data involved in rollbacks may or may not be available. Rollbacks would only be available if so requested, but are never included in SQL generation regardless of LOB inclusion/exclusion. So if both LOBs and rollbacks are requested, do not plan to either view or LOAD any reported rolled back LOB data (again, nor should you expect it in SQL generation, being consistent with how Log Analysis Tool deals with any rollbacks and SQL).

The inclusion of LOB column data can force the method of detail report processing. If only inserts are found affecting LOB data, then detail report processing can use either log-backward or log-forward methods. However, if updates or deletes are also present, then log-forward processing must be used. This is due to the way that Db2 logs these types of actions. The implications of this are that, in many cases, including LOB data means you will have to run log-forward. Therefore, LOB table space image copies must be taken as well as image copies for your base table(s) table space(s). Without these resources, Log Analysis Tool cannot track LOB data (except the INSERT-only case).

Important: DATA CAPTURE CHANGES on the base table containing the LOB column(s) has no effect on LOB logging. Also, if the LOB table space is defined as LOG(NO), LOB data support is not possible.

Due to the potential size of LOBs, every effort is made not to replicate the LOB data. Therefore, files such as the EXTFILE and PLOGDS files, which hold Db2 log data in various formats, do not both include similar LOB column data. Though this results in a savings regarding DASD/TAPE sizes, it also means that LOB data can be found in different places throughout Log Analysis Tool. While Log Analysis Tool resolves the appropriate origination of LOB data on request, it does not treat LOB column data in a similar fashion as non-LOB column data.

For example, when LOB data is included, the detail report does not include that data. If it is desired to see any particular LOB column data, that can be requested via Log Analysis Tool (see primary menu option **View LOB data**). The detail report will still just contain the LOB locator value, as it has in the past. Again, this method is used to prevent unnecessarily large reports that may contain LOB data.

If LOB data is included and SQL is requested, the LOB data may be included in the SQL.

- For generated DELETE SQL statements, LOB data is never included in the WHERE clause.
- For generated INSERT SQL statements, the LOB data is always included in the values list.
- For generated UPDATE statements, the LOB data is included only if the LOB column itself changed. If it did not, there is no tracking of the LOB data for the row (see the preceding description). Therefore, updates may or may not include LOB data depending on the situation.

LOB column data is omitted whenever possible for DASD/TAPE space savings. The term 'LOB data' means only that the data is included using file-by-reference.

SQL generated by Log Analysis Tool which contains LOB data will only contain LOB data by file reference. The LOB data itself is not included in the SQL (such inclusion would likely exceed SQL maximums). Therefore, SQL generated that includes LOB data cannot be executed by SPUFI or DSNTEP2. Fast Apply

SQL is required to run this SQL (see Chapter 9, "Executing dynamic SQL in batch," on page 155, primary menu option 5).

LOB data is not archived by this product, regardless of LOB data inclusion.

For loading data back into Db2, it is again important to note that any updates that do not modify the LOB data itself will not have LOB data available. If an update does have LOB column changes, the LOB data itself will use the Db2 LOAD utility file-reference method. That is, where the LOB data should exist in the load file is replaced by a file reference. The file reference will contain the name of the file containing the LOB data. Keep in mind this can be a very expensive option given that each LOB column requiring such a file reference must reside in its own file. Log Analysis Tool will create those files, but there may be many more than your site can accommodate.

For example, if the output contains just 1000 rows, and each has a single LOB column that has been updated, that means 2000 files need to be created in order to LOAD the data back into Db2. That is because a pre-change and post-change update image is needed for each modified LOB column (1000*2). So though this is supported, in general, such an action should not be performed without understanding the implications and resources involved.

Both LOB and XML column types require other columns in a table. A LOB column requires a ROWID column. An XML column requires a DOCID column. In the case of a LOB, the ROWID column may be generated by Db2 on your behalf, or you may explicitly create it. A DOCID is always generated by Db2. When generated by Db2, these columns are considered 'hidden'. For LOAD purposes, Db2–generated hidden columns are ignored when DATA ONLY=Y. If you create the target tables outside of the product, the target tables must also have the hidden columns. If the product creates the target table for you, it will be created with hidden columns. This allows DATA ONLY=Y type LOADs to generate new values for these hidden columns during the LOAD. If DATA ONLY=N and a ROWID column is not Db2 generated, it will be treated as a VARCHAR field, so you see the value as it originally occurred. If DATA ONLY=Y and the ROWID column is not Db2 generated, you must create the target table outside of Log Analysis Tool, as it will require an index, and Log Analysis Tool does not generate an index on a target table. Always make your target tables ROWID values as GENERATED BY DEFAULT if not Db2 generated.

Important: Though mentioned many times, it cannot be stressed enough that LOB support will have a large, negative impact on performance. Do not treat such inclusion as normal processing. It is very likely that LOB inclusion will mean your EXTFILE, and possibly others, need to be allocated to tape, rather than DASD. Minimally, your DASD allocation for the EXTFILE will need to be greatly enlarged. LOB inclusion should be used only for exception processing, not for general processing of your Db2 logs.

Chapter 21. Temporal Data

Log Analysis Tool cannot currently generate SQL that will execute successfully for the SYSTEM PERIOD table because of Db2 restrictions on GENERATED ALWAYS TIMESTAMP columns.

Generation of SQL for Temporal Data using the SYSTEM PERIOD and HISTORY Tables

The SQL generated for the SYSTEM PERIOD table does contain the actual values for the columns as they appear in the Db2 log, however:

- The SQL for UPDATE and INSERT statements does not execute because of the above restriction.
- The SQL for DELETE statements does work unless they are deleting a row that did not get inserted due to the above restriction.

The SQL for the HISTORY table does execute successfully because the GENERATED ALWAYS clause is not used in the HISTORY table.

Scenario 1-Insert

Insert a row into the SYSTEM PERIOD table. The original INSERT causes at least two columns to be updated with the current date and time due to the GENERATED ALWAYS clause. The row end column is set to Midnight December 31, 9999.

When Log Analysis Tool is run against the SYSTEM PERIOD table and the HISTORY table, Log Analysis Tool reports on the insert to the SYSTEM PERIOD table and create SQL that contains the actual values for the GENERATED ALWAYS columns.

When executed, the INSERT SQL encounters: `SQLCODE = -798, Error: a value cannot be specified for column COL_SYS_START which is defined as generated always.`

Scenario 2-Update

Update a row in the SYSTEM PERIOD table. The UPDATE causes at least two columns in the SYSTEM PERIOD table to be updated with the current data and time due to the GENERATED ALWAYS clause. The row end column remains unchanged. The UPDATE also inserts a row into the HISTORY table with the original values from the previous insert but the row end column is set to the update date and time signifying the end of the current row.

When Log Analysis Tool is run against the SYSTEM PERIOD table and the HISTORY table, Log Analysis Tool reports on the update to the SYSTEM PERIOD table and the insert into the HISTORY table. Log Analysis Tool creates the update SQL with the actual values for the GENERATED ALWAYS columns as found in the Db2 log and the insert SQL for the HISTORY table.

- The UPDATE SQL statement does not execute correctly because of Db2 restrictions on GENERATED ALWAYS TIMESTAMP columns.
- The INSERT SQL statement for the HISTORY table does work, but understand that the SYSTEM PERIOD table is now out of synch with the HISTORY table.

When executed:

The UPDATE receive: `SQLCODE = -151, Error: the update operation is invalid because the catalog description of column COL_SYS_START indicates that it cannot be updated.`

The INSERT receives: `SQLCODE = 000, SUCCESSFUL EXECUTION`

This creates an out of synch condition between the SYSTEM PERIOD table and the HISTORY table.

Scenario 3-Delete

Delete a row from the SYSTEM PERIOD table. The DELETE deletes the row from the SYSTEM PERIOD table while inserting a row into the HISTORY table. Log Analysis Tool creates the DELETE statement for the SYSTEM PERIOD table and the INSERT statement for the HISTORY table. Both SQL statements work with the understanding that two INSERTs are actually made into the HISTORY table.

The DELETE receives: SQLCODE = 000, SUCCESSFUL EXECUTION

The INSERT receives: SQLCODE = 000, SUCCESSFUL EXECUTION

Remember, however, there will be two INSERTs are actually made into the HISTORY table:

1. An INSERT from Db2 after deleting from the SYSTEM PERIOD table.
2. An INSERT from the INSERT SQL generated by Log Analysis Tool.

Summary

The SQL failures in the above scenarios are due to the restrictions Db2 sets forth with GENERATED ALWAYS TIMESTAMPS. Similar situations will also occur with UNDO SQL as well.

Chapter 22. After the Details report: accessing the data

Once you have run a details report, you may wish to use the results of that report in various ways. You may have requested undo or redo SQL, and you can process that SQL as necessary. However, there are other options available to you for using the results of the detail report output. The options **Load data into Db2** and **Execute dynamic SQL in batch** provide capabilities for such things as data replication, user-tailored report viewing, and running SQL more efficiently.

Load data into Db2

This process allows you to load data from the reports into Db2 tables. The key here is that this process uses the LOAD facility, therefore, it can do one and only one thing: insert data into a table. It cannot update or delete currently existing rows in a table. No product using the LOAD facility can do anything but insert rows into a table. If you need the capability to update or delete currently existing rows in Db2 tables, see Chapter 9, "Executing dynamic SQL in batch," on page 155.

This option provides a great deal of flexibility in terms of what data from the detail report can be loaded into a Db2 table. You can choose to LOAD only INSERT-type records, DELETE-type records, and UPDATE-type records (either pre-change row images or post-change row images). Keep in mind, you are not updating or deleting currently existing rows, you are inserting recorded row images from the detail report. For example, when requesting DELETE-type records, you would be inserting the pre-row image of a DELETE (a DELETE has no post-row image). Similarly, requesting UPDATE-type post-change row images means you are inserting recorded row images as they looked after the update occurred. Whatever it is you wish to LOAD (insert) into a table, you can do through this option.

You can also request to LOAD only the table data portion of the detail report, or include Log Analysis Tool information along with the data. If you chose to include this Log Analysis Tool information, you are basically using the LOAD option to view the detail report in a Db2 table. That is, the Log Analysis Tool information will allow you to view the detail report results in a Db2 table and query it as needed. The Log Analysis Tool information includes things like row status (pre-,post-row images) and action type (update/insert/delete). Without this information, it would be impossible to relate the data loaded into the Db2 table with actual log events. With this information, you can do just that.

If you are not interested in viewing the report in a Db2 table, but want to load data only for replication or similar type purposes, you can request that no Log Analysis Tool information be loaded along with the data. This can be useful if you wish to replicate INSERTs from one table to another, or if you wish to maintain a table of modifications against another table. The reasons for doing this type of LOAD activity are limitless, and this product option provides the flexibility to handle any type of LOAD request needed.

You can choose to LOAD one table's activity into another table, or many tables' activity into many tables. You can specify each table desired, or process them all. You can create the target tables on the fly, or they can pre-exist. You can LOAD a table's activity back into itself. The point is that most, if not all, possible scenarios are covered.

Executing dynamic SQL in batch

You should use **Execute dynamic SQL in batch** if you want to execute the generated SQL in batch mode, as opposed to submitting it through the Db2 SPUFI utility.

The SQLOUT DD file generated during the detail report run is used as input to this batch process. The SQL is executed dynamically.

- If you requested redo SQL during the detail report run, the execution of the generated SQL will redo those changes.

- If you requested undo SQL during the detail report run, the execution of the generated SQL will undo those changes.

This process is very similar to running your SQL through SPUFI, except it executes in batch mode. The SQL should not be modified in any way prior to execution without complete knowledge of the results of such actions. As with any SQL execution, errors can occur. Restart capabilities are built into this process.

From the Db2 Log Analysis Tool Selection menu, select option **5**, **Execute dynamic SQL in batch**.

The Execute dynamic SQL in batch: Fast Apply panel appears.

```
V3.5.0 -------- Execute dynamic SQL in batch: Fast Apply --------- SC01/SS1A
COMMAND ===>
                                                           More:     +

*Action..............  E          (E - Edit, S - Submit)
 Job Identifier......             (8 character maximum)
*DB2 subsystem name..  SS1A       (DB2 SSID)
 LE runtime loadlibs.

*Restart.............  I          (A - Auto, I - Ignore, S - Startover,
                                      N - New, M - Number)
 Restart Name........             (not used if Restart=I)
 Restart SQL number..             (only used if Restart=M)
*Commit Frequency....  1
*Commit Type.........  T          (T - Transaction, S - Statement)
*Loglevel............  1          (1-9 only)
*Bypass Multi-Row Inserts.        (Y - Off, N - On)
*Table filters............ N      (Y/N)

*ON SQLERROR..............  S        (S - Stop, W - Warn,
                                      I - Ignore, R - Rollback)
*ON SQLWARN..........  I          (S - Stop, W - Warn,
                                      I - Ignore, R - Rollback)
*ON Multiple rows....  I          (S - Stop, W - Warn,
                                      I - Ignore, R - Rollback)
```

Figure 67. Execute dynamic SQL in batch: Fast Apply panel (top half of panel)

```
V3.5.0 -------- Execute dynamic SQL in batch: Fast Apply --------- SC01/SS1A
COMMAND ===>
                                                           More:    -
 ON SQLCODEs (Code,S/W/I/R;Code,S/W/I/R;etc):


Source tables:                     Target tables:
Table Owner/Name (Convert from);   Table Owner/Name (Convert to):
          /                                  /
          /                                  /
          /                                  /
          /                                  /
          /                                  /
```

Figure 68. Execute dynamic SQL in batch: Fast Apply panel (bottom half of panel)

Execute dynamic SQL in batch: Fast Apply panel fields

Action

Type E or S.

E

Indicates that the temporary JCL created as a result of the data entered on the panel will be displayed in edit mode.

S

Indicates the temporary JCL will be submitted automatically.

Job identifier

This is the same job identifier as used in general and detail reports. It is an eight-character maximum field that is used to identify the data set names for the different runs of reports. It must be a valid node in a data set name.

Db2 subsystem name

Enter a valid subsystem ID (SSID) identifying the Db2 subsystem against which the batch SQL will run. The Log Analysis Tool plans must be bound at this SSID. It does not have to be the same SSID against which the SQL was generated. In other words, you can be performing data replication, but the tables named in the SQL must exist at this SSID.

LE runtime loadlibs

(*Optional*) Enter the fully qualified name (no quotes) of your Language Environment (LE) runtime load libraries. Put a single load library on each line. These libraries will likely be named CEE, or something similar, though your site has complete control over the naming convention of these libraries. These libraries are used by the C program handling the fast apply SQL process. Contact your systems programming staff if you do not know the names.

Restart

Enter one of the values:

A

AUTO mode. The restart program determines if there is a row in the restart table for this process and proceeds as follows:

- If there is not a row in the restart table, it creates one and starts processing SQLIN from the start of the file.
- If there is a row in the restart table and the row indicates that the last run was a success, the restart program starts processing SQLIN from the start of the file.
- If there is a row in the restart table and the row indicates that the last run failed, the restart program starts processing SQLIN at the statement following the last successful commit.

I

IGNORE mode. No restart is possible, and the restart table and process name are ignored.

S

STARTOVER mode. The restart program determines if there is a row in the restart table for this process name and proceeds as follows:

- If there is not a row, the restart program creates one and starts processing SQLIN from the start of the file.
- If there is a row, the restart program resets the row and starts processing SQLIN from the start of the file.

N

NEW mode. The restart program assumes that this is a totally new process, and creates a new entry for the process name in the restart table. If the process name already exists, the program terminates.

M

NUMBER mode. The restart program starts processing SQL at the specified SQL statement number.

Restart name

This is the process name used to associate different runs of the restart program. It must be provided if restart is desired. Use an eight- character unique name to identify the run. If you re-use this name, be sure that the previous run (with the same restart name) ran successfully, or else the SQL will be started at the point of failure from the previous run. If you are not sure of the results of the previous run, either change the restart name or set **Restart field** to **S**, so it is treated as a new run from the beginning of the SQL.

Restart SQL number

This is the SQL statement number to start processing from if you set Restart=M. It is otherwise ignored.

Commit Frequency

Enter a value from 1 to 99999. This controls how frequently commits are performed. It is used in conjunction with the Commit Type parameter.

Commit Type

Enter one of the values described following:

T

Transaction. This refers to the COMMIT statements in the SQL file.

- If the Commit Frequency is 1 and the Commit Type is Transaction, a commit is performed for every commit statement found in the SQL file.
- If the Commit Frequency is 10 and the Commit Type is Transaction, a commit is performed for every 10 commit statements found in the SQL file.

S

Statement. This refers to the executed Fast Apply SQL statements, not the actual SQL statements in the SQL file.

- If the Commit Frequency is 1 and the Commit Type is Statement, a commit is performed after each executed Fast Apply statement which might include multiple, combined SQL statements from the SQL file.
- If the Commit Frequency is 10 and the Commit Type is Statement, a commit is performed after every 10 executed Fast Apply statements, each of which might include multiple, combined SQL statements from the SQL file.

With this option, the actual commit statements in the SQL file are ignored.

When there are more than two inserts per commit in one statement, Fast Apply switches on multiple-row insert with the attributes `NOT ATOMIC CONTINUE ON SQLEXCEPTION` by default. To switch off the multi-row insert, add `,MRIOFF` to the PARM=SSID parameter in the JCL EXEC statement.

Note: Adding the MRIOFF parameter helps prevent issues that might be caused by multi-row insert, but requires more CPU usage.

Loglevel

Enter a value from 1 to 9. This controls the level of messages written to the ALAPRINT DD data set. Leave as the default (1) unless directed otherwise for problem analysis.

Bypass Multi-Row Inserts

The multi-row Inserts are used by Fast Apply to improve performance. This option prevents the use of multi-row inserts when Fast Apply runs. Enter 'Y' to bypass Multi-Row Inserts processing.

Table filters

Table filters allow for tables found in the SQL to be excluded and ignored or for execution of SQL only for specific tables. Enter a Y if you want to use table filters.

Restriction: If you are converting a table name from its value in the SQL file to something different and specify the table on the filter panel, you must specify the target name.

On xxxxx fields

For all of the ON xxxxx fields listed here, these values and rules apply:

S

STOP after an event. It will rollback to the previous commit point, and stop all processing.

W

Issue a WARNING after an event, but continue processing.

I

IGNORE an event, and continue processing.

R

ROLLBACK after an event. It will rollback to a previous commit point, but then continue processing with the next SQL statement.

The precedence rules for the ON xxxxx fields are as follows:

1. ON SQLCODEs setting
2. ON SQLERROR setting
3. ON Multiple rows setting
4. ON SQLWARN setting

ON SQLERROR
Enter one of the described values. The event in this case is any SQL error return code.

ON SQLWARN
Enter one of the described values. The event in this case is any SQL warning return code.

ON Multiple rows
Enter one of the described values. The event in this case is any SQL statement which causes modifications to more than one row.

ON SQLCODEs
Use this line to enter up to 10 different SQL codes for which you want to control the specific action to take. For example, if you want SQL code +100 to be ignored, you can specify that here. The format of this input is shown here, and must be copied exactly. If you need more than 10 SQL codes to handle specifically, edit the JCL in DD ALACFG ensuring that you follow that format.

- The format for each entry is: `<sign><code>,<action>;`
- The `<sign>` can be + or -. It must be present.
- The `<code>` can be any 3 digit SQL condition code (warning or error).
- The `<action>` is as described (S/W/I/R), and the event is the listed SQL condition code.

Examples:

```
+100,I;
```

Ignore SQL code = +100

```
+100,I;-511,R;-904,I;
```

```
Ignore SQL code = +100, rollback on SQL code =-511, ignore SQL
 code = -904
```

Important: These codes are not intended to have any meaning or usefulness for your specifications; they are examples only.

Execute dynamic SQL in batch: Fast Apply table filters

When you specify Y in the Table filters option on the "Execute dynamic SQL in batch: Fast Apply" panel, you are identifying that you want either the SQL to be excluded and ignored for specific tables or for the SQL to be run only for specific tables.

When you type Y in the Table filters field and press **Enter**, the "Fast Apply Table Filters" panel is displayed:

```
V3.5.0 --------------- Fast Apply Table Filters ----------------- SC01/SS1A
COMMAND ===>

Hit 'Enter' to save current filters; 'PF3' to bypass use of these filters

*Exclude tables ..... N (Y/N)

*Table Owner                            *Table Name
----------------------------------    ---------------------------------------
_____    _____
```

Figure 69. Execute dynamic SQL in batch: Fast Apply Table Filters panel (top half of panel)

Fast Apply Table Filters fields

Exclude Tables

Identify whether you want to exclude the tables that you are specifying on the panel. If you specify **Y** to exclude tables, any table specified is ignored when processing the SQL file. All other tables are included and processed. If you specify **N** to not exclude tables, only the specified table is processed. All other tables are excluded and ignored.

Table Owner

Identifies the table owner.

Table Name

Identifies the table name.

Choosing between the options

These options should handle all your back-end processing needs, but it may not be clear which one is appropriate for which set of circumstances. The simple rule is to use LOAD processing whenever possible. LOAD processing is much faster than SQL execution. Your goals must be considered, however, to determine the correct choice. For example:

- If your goal is to replicate all changes against one table in another table, only SQL execution can be used because the LOAD process cannot update or delete pre-existing rows. In that case, you should choose **Execute dynamic SQL in batch**.

- If your goal is to audit changes to a table, you can use the LOAD process to insert all changes into an audit table. You can also use the LOAD process to mirror all INSERTs made to a filtered table.

The possible scenarios are certainly limitless, and through the use of these options, all scenarios can be handled by Log Analysis Tool.

Chapter 23. Modifying a JSON parameters file

The JSON parameters file provides parameters that you can change to modify the functionality of Log Analysis Tool. Using this file provides you full control over the job without JCL re-generation and ISPF interaction. This section provides more detailed information on the JSON parameter file.

Details of JSON parameter file

The JSON parameters file consists of five high-level parts or "blocks" that provide a specific function.

- **Audit**. The Audit block provides report generation (both general and details reports).
- **Recover**. The Recover block provides the UNDO/REDO SQL generation and dynamic SQL execution.
- **Replicate**. The Replicate block provides the LOADFILE generation.
- **Settings**. The Settings block describes common settings, such as processing mode and the dataset prefix.
- **Dataset**. The Dataset definitions block describes datasets and parameters for allocation.

While the JSON parameter file does not contain filtering data or start/end log ranges, it does provide the following functionality:

- The FILTERS DD as inline data or as a filter file will be used to store all the filters for the run in a standard format, generated from the ISPF panels.
- The DATAIN DD is used to describe the range of log activity to be read, which can be based on date/time or RBAs/LRSNs, and are all mutually exclusive.
- The JSONIN DD represents all parameters listed in a standard data-interchange format, JSON.

Any line that starts with '//' is a comment and is ignored.

You can also generate this JSON file from a profile, which will enable you to export all of the settings and parameters from a Db2 table to the JSON parameter file.

Setting parameter values

There are two methods for specifying parameter values, which are:

- A sample JSON file is provided that you can edit and submit. This sample file contains comments that describe the parameters and possible values.
- On the "User Settings" panel, enter "Y" for the **Create JSON parameters file** option to generate a job with the JSON parameters file as input. You can then proceed through the ISPF panels, specifying all the needed parameters and filters.

Any value can be changed as you prefer. Any key-value pair, with the exception of "Processing Mode" from the "Settings" section, can be deleted from the file and a default parameter will be used in its place.

If there is an error during the parsing or validation process, a message will be displayed in MESSAGES DD, which describes the issue. If a parameters file contains a custom key-value pair, then an informational message will be included to the ALAJSLOG DD and the key will be omitted as unrecognized.

Restarting a failed job

Both 1-phase and 2-phase processing modes can be used with the JSON parameters file. However, the "restart-ability" was implemented for 2-phase mode, so the job can be continued from a load module that was previously unsuccessfully executed.

To support the restart-ability, a MODEFILE, which contains a failed load module name, is saved if the job fails. If a job completes successfully, the MODEFILE is automatically deleted.

To restart a job, the RESTART parameter should be specified in the PARM statement of the ALAJSON step. If the MODEFILE is empty or does not contain a failed load module name, an information message will be displayed and a job will be executed from the start.

Dataset Definitions

The "Dataset Definitions" section describes all of the datasets (if full JSON file is generated) that can be used during the run. However, it does not mean that all of the described datasets will be allocated during processing. Log Analysis Tool defines the list of needed files for the load module and allocates them before module invocation. Only needed datasets will be allocated during the run, based on the parameters of the job.

If the short JSON file generation is selected (by specifying "S" in the "Create JSON parameter file" field), a list of datasets used for this run is included.

Dataset allocation parameters can be overridden in the "Dataset Definitions" section of the JSON parameters file to manage dynamic allocation process. If any dataset parameter or the whole dataset description is deleted from the parameters file, Log Analysis Tool will use the default settings during the corresponding dataset allocation process.

Note: When using the JSON parameters file, the dataset space parameter is not dynamically calculated during the run, like it is in a standard job run, and default allocation parameters are used.

New Parser module

With the use of the JSON parameters file, there is no need to proceed through Log Analysis Tool ISPF panels to change a few options. Both a generated JCL file and a JSON parameters file are reusable and do not need to be recreated.

The ALAJSON step is generated with PARM statements which contain the Db2 subsystem name (SSID) as a first parameter and LOGLEVEL as a second. If the recover execution step is specified in the JSON parameters file, then Fast Apply will be executed and all parameters, previously specified for ALA@FSP, can be entered as PARM statements of the ALAJSON step.

The ALAJSON load module uses JSONIN DD as an input to define processing parameters. During the execution, JSON input is parsed and all the parameters are validated and saved to be passed to the proper load modules.

An ALAJSLOG DD is used as an output for the ALAJSON module. It contains all of the generated informational messages and a brief report to describe the selected processing mode.

Restrictions

A few restrictions should be considered when trying to use the JSON parameters file. These restrictions are listed below.

1. The setting for the **Create JSON parameters file** field cannot be used with the "2-phase" **VSAM** processing mode. The following error message will be issued:

   ```
   ALAA749E-JSON parameters file cannot be used in 2 phase mode VSAM
   ```

2. The "Specify logs" option of the "Generate Database Activity Report (General)" panel and the "Expert options" parameter of the "Generate Database Activity Report (Details)" panel cannot be used along with JSON parameters file generation. A corresponding error message is issued during panel validation:

   ```
   ALAA163E-Specify logs option is incompatible with JSON parameters file
   ```

   OR

   ```
   ALAA164E-Expert options are incompatible with JSON parameters file.
   ```

3. On the Main Menu, a header reordering for the "Load data into DB2" option is ignored when creating the JSON parameters file. In this case, headers have their default positions and a warning is displayed:

```
ALAA750W-JSON parameters file ignores header reordering.
```

Any validation issue is described within an error message. You can also use the JSON sample to look-up default values for parameters.

Chapter 24. Troubleshooting

Use these topics to diagnose and correct problems that you experience with Log Analysis Tool.

Return Codes

Table 12. Description of return codes

Code	Meaning	Description
0	No errors.	Successful completion of all tasks. **Note:** Even though processing ended with no errors, there may not be any data in the reports due to your filters.
2	No data to process.	• Details report The details report is being run, but there is no data to process. This is because either: 1. There was no data to report on in the general report. 2. The filters you specified for the details report discarded all of the data from the general report. • Dynamic SQL in batch You requested a single table, but no SQL affecting that table was found.
4	Warning. Processing continued, but data results may be invalid.	Check the WARNINGS DD for more information and review JCL to determine if the data results set is valid. Processing continues only if **Defaults** value for handling warnings indicates it should.
8	Error. Processing stopped.	Check the MESSAGES DD for more information.
16	Severe error. Job failed during error processing.	Check the job log for diagnostic information in the form of WTO messages.

While running batch JCL, Log Analysis Tool often attempts to open data sets that do not exist. These data sets may or may not exist due to certain filtering and processing methods. This is not a problem, and these messages are informational only. An example of such a message is:

```
IEC130I OBLIST   DD STATEMENT MISSING
```

These are not messages generated by Log Analysis Tool; they are operating system messages that Log Analysis Tool has no control over. All Log Analysis Tool generated messages are prefixed with ALA. Log Analysis Tool will write out informational messages, but these are also prefixed with ALA and are not for external use. Disregard all informational messages (generated from the operating system or Log Analysis Tool), unless their use is requested by support during diagnostic procedures.

Db2 Log Analysis Tool messages

All messages are available online through the Message Help Facility. The Message Help Facility should be considered the primary and most current source for all message text since messages can be updated throughout the life cycle of a release through PTFs. The hardcopy messages listed here reflect the state of all messages at the time the product is initially released.

All messages generated by Log Analysis Tool have a severity code printed as the last character of the message ID. The severity codes are described in this table:

Table 13. Error message severity codes

Severity Code	Description
I	Information only. No user action required.
W	Warning message. Results may not be as expected.
E	Error message. Some may be user-correctable, read the User Response to determine the course of action.

ALAA001E   **Field must not be left blank**

Explanation:
The specified field was left blank. The panel requires input for this field.

User response:
Enter valid input for this field.

ALAA002E   **Unsuccessful OPEN of ISPSLIB member ALAIN1**

Explanation:
An OPEN request failed for the specified member.

User response:
Verify that the member exists. Logging off TSO may resolve this error.

ALAA003E   **Invalid LOB action specification; M or B only allowed**

Explanation:
Only values of "M" (materialize) or "B" (browse) are allowed.

User response:
Correct your input and retry.

ALAA004E   **No jobcard data available**

Explanation:
A jobcard is required to process this request.

User response:
Enter jobcard data using the Defaults option from the main menu for DB2 Log Analysis Tool.

ALAA005E   **ISPSLIB member ALAIN1 bad record starting *recid***

Explanation:
A bad record was found in the specified member.

User response:
Use the record identifier to locate the bad record and correct the entry.

ALAA006E   **Bad date format or value**

Explanation:
A date format or value supplied on the panel was incorrect.

User response:
Use the specified formats and/or verify the correctness of YYYYMMDD values.

ALAA007E   **Bad time format or value**

Explanation:
A time format or value supplied on the panel was incorrect.

User response:
Use the specified formats and/or verify the correctness of HHMMSS values.

ALAA008E   **End date cannot be less than start date**

Explanation:
The end date must be greater than the start date for log analysis.

User response:
Check and correct your date values.

ALAA009E   **End time cannot be less than start time when dates are equal**

Explanation:
The end time must be greater than the start time for log analysis, given that the start and end dates are equal.

User response:
Check and correct your time values.

ALAA011E   **No RBAs found in BSDS for the requested log range**

Explanation:
The log range specified was outside the range of the contents of the bootstrap data set (BSDS).

User response

Verify the specified log ranges. If the log ranges look correct, possible scenarios for this are either:

1. The ranges specified were greater than the highest ranges for the currently active log, or

2. Date gaps may exist in the BSDS for the DB2 logs

If number 1, archive the log and retry. Otherwise, check the BSDS log ranges. The DB2 supplied utility DSNJU004 may be run against the BSDS to determine the log ranges and logs known to DB2.

| ALAA012E | The BSDS start date for logs was > the requested end date |

Explanation:
The BSDS contains no information for the DB2 logs in the date range specified. The BSDS keeps only a certain amount of data for the logs based on installation settings. Once the BSDS reaches its logging maximum, the most recent archive data wraps around, thereby overlaying older log data.

User response:
The DB2 supplied utility DSNJU004 may be run against the BSDS to determine the dates and logs known to DB2.

| ALAA013E | Time conversion routine found date lower than 1990/01/01 |

Explanation:
A date was found while processing the log analysis request that was earlier than 1990/01/01. This is the low date limit of the DB2 Batch Undo. This is only a limit on DB2 log dates, not on DB2 data dates.

User response:
If it seems unlikely that this date would be encountered in the logs, an internal error may have occurred; contact your product administrator.

| ALAA014E | Invalid value specified for DBID,PSID, or OBID. |

Explanation
The ID fields must be entered as valid decimal fields. For example:

- Valid: 00055, 55

- Invalid: 5 5, 5A5

User response:
Correct the input and retry.

| ALAA015E | Invalid member ID specified. |

Explanation:
A member ID represents the member of a data sharing group, or 0, if not data sharing. A valid number is from 00 to 32. Use the value as displayed from the detail report, but only the rightmost 2 digits.

User response:
Correct the input and retry.

| ALAA016E | Filter input bad record: *recid*. |

Explanation:
The filter input (DD DATAIN) contained a bad record. It is most likely this would only occur if the user edited

the input records. The offending record is shown (the first 20 characters of it).

User response:
Correct the input and retry. For help on valid records, see tutorial section on filter file handling.

| ALAA017E | Unsuccessful OPEN of ISPF skeleton member ALAIN3 |

Explanation:
An OPEN request failed for the specified member.

User response:
Verify that the member exists in your ALA skeleton library (as shipped, suffixed "SALASLIB"). Logging off TSO may resolve this error.

| ALAA018E | OPEN failure on DATAIN input dataset |

Explanation:
An OPEN request failed for the specified ddname.

User response:
Verify the ddname existence in the JCL.

| ALAA019E | DATAIN input file has bad record starting with: *recid* |

Explanation:
A bad record was encountered while processing the specified data set.

User response:
Use the record identifier to locate and correct the record.

| ALAA020E | Command not recognized |

Explanation:
The command was not recognized.

User response:
Enter one of the specified commands, options, or a valid TSO command. The value as entered was not recognized. Correct your input and retry.

| ALAA021E | Error occurred during error processing: Ret: *code* |

Explanation
An error occurred while error processing was in effect. The return code values are listed following:

- 14 -missing a quote in the ISPMLIB member message.

- 16/28 -OPEN failed/abended.

  For 28, a bad message ID may have been entered.

- 20 -dynamic allocation failure for ISPMLIB.

- 24 -dynamic de-allocation failure for ISPMLIB.

User response:

If you cannot resolve this issue, contact IBM Customer Support.

ALAA022E　　**OPEN failure for GENRPT output file**

Explanation:
An OPEN request for the specified data set failed.

User response:
Verify the existence of the ddname in the JCL.

ALAA023E　　**BSDS VSAM OPEN error: Ret: *code*, Res: *code***

Explanation:
An OPEN failure occurred while attempting to access the bootstrap data set.

User response:
The return and reason codes can be found in the IBM document "DFSMS Macro Instructions For Data Sets" in chapter "VSAM Macro Instructions".

ALAA024E　　**BSDS VSAM GET error: Ret: *code*, Res: *code***

Explanation:
A GET failure occurred while attempting to access the bootstrap data set.

User response:
The return and reason codes can be found in the IBM document "DFSMS Macro Instructions For Data Sets" in chapter "VSAM Macro Instructions".

ALAA025E　　**"Misc Flags = H" requires "Generate Details = Y".**

Explanation:
The high speed option requires a combined general and detail report run as the high speed processing does not allow further filtering at the detail report level. It also adds overhead during the general report run (the savings occur during the detail report only).

Explanation:
Either set "Generate Details = Y" or remove the "H" value from your "Misc Flags" field.

ALAA026E　　**Table owner: *id*, Table: *tablename* ;not found in DB2**

Explanation:
The specified table owner and name was not found in DB2. Either the table has been dropped, or it never existed.

User response:
Change the table owner and name and retry.

ALAA027E　　**DBname: *dbname*, TSname: *tsname*, not found in DB2**

Explanation:

The specified database/tablespace was not found in DB2. Either the table space has been dropped, or it never existed.

User response:
Change the database/ table space name and retry.

ALAA028E　　**Any set of log range values requires both start and end values.**

Explanation:
To request either an RBA or LRSN log range, both the start and end values must both be present.

User response:
Correct the input and retry.

ALAA029E　　**Start log range not less than end log range value.**

Explanation:
To request either an RBA or LRSN log range, the start value must be less than the end value.

User response:
Correct the input and retry.

ALAA030E　　**Field must be Y (yes) or N (no) only**

Explanation:
You entered an incorrect value.

User response:
Enter one of the specified values for the field on the panel.

ALAA031E　　**The URID specified must be a valid hexadecimal, length 12**

Explanation:
The URID specified was invalid. It must be a character representation of a 12 byte hexadecimal number. For example, 000012F8AFFF

User response:
Specify a valid URID.

ALAA032E　　**Field must be Y (Yes), N (No), or X (Yes, Exclude triggers) only**

Explanation:
The value as specified is incorrect. Only a value of "Y", "N", or "X" is acceptable.

User response:
Enter a "Y", "N", or "X" only for the identified field on the panel.

ALAA033E　　**No SSID found for details report**

Explanation:
No SSID was found which may mean that no database activity report (general) was run prior to running the detail report.

User response:

Verify a SSID exists on the panel to generate a database activity report (general), and that the general report was run.

| ALAA034E | Invalid hexadecimal LRSN supplied. |

Explanation

Only a 12 character hexadecimal display value is allowed here. For example:

- Valid values: B7AAA0125421, BC000000FF34

- Invalid values: BC, 00000MMM234

Explanation:
Correct your input and retry. Use the values only as shown in the detail report.

| ALAA035E | The field is invalid |

Explanation:
Character fields must start with a valid character. Wildcards are not currently supported in any fields.

User response:
Make sure the field conforms to the specified format, if one is provided. Change the field and try again.

| ALAA036E | The requested filter object *object* was not found. |

Explanation:
The object was not found in Db2 and may have been either typed incorrectly or dropped from Db2.

User response:
Verify the correctness of the object name and its existence in the specified Db2 subsystem. Dropped objects cannot be used as filters for Log Analysis Tool.

| ALAA037E | Process halted, memory exhausted for *storagename*. |

Explanation:
An internal storage area was filled beyond capacity and the processing was halted at that point.

User response:
Narrow your filter ranges and retry. If this occurs frequently, contact your product administrator to alert vendor.

| ALAA038E | Unsuccessful OPEN of the REFILE. |

Explanation:
The REFILE could not be opened for input or output.

System action:
Program processing halts and an error message is issued.

User response:
Verify the existence of the DD named REFILE in your JCL. Also, verify the correctness of the associated data set name. The LRECL parameter must be 100.

| ALAA039E | Unsuccessful OPEN for DD ROWDATA, reason code: *code*. |

Explanation:
The VSAM file associated with the ROWDATA DD statement could not be opened.

User response:
Verify the existence of the ROWDATA DD statement in your JCL and the associated data set name. The reason code is a standard OS/390 VSAM error code.

| ALAA040E | Unsuccessful PUT for ROWDATA DD, reason code: *code*. |

Explanation:
The VSAM file associated with the ROWDATA DD statement could not be written to.

User response:
Check the reason code included with the message in the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "VSAM Macro Instructions" for explanations of VSAM PUT errors.'

| ALAA041E | CATFILE lookup error: *varname* |

Explanation:
The CATFILE holds DB2 catalog information generated during the initial report creation for DB2 Log Analysis. This CATFILE can only be used with the REFILE generated in the same batch run. This message may indicate a mismatch between the files. The variable information may be either a table name, database and table space name, DBID/PSID/OBID, or DBID/OBID, depending on the format.

User response:
If you cannot resolve this issue, contact IBM Customer Support.

| ALAA042E | An unknown type of log record was found. It is dumped below. |

Explanation:
A log record was read but the program could not decipher its type based on known log record mappings. The first 70 bytes of the offending log record are dumped out with this message.

User response:
Notify your product administrator.

| ALAA043E | GENCB for ACB failed in ALAPGR: return code: *code*. |

Explanation:
GENCB for ACB failed in ALAPGR.

User response:
Check the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "VSAM Macro Instructions" for return code explanations.

ALAA044E   **GENCB for RPL failed in ALAPGR: return code: *code*.**

Explanation:
GENCB for RPL failed in ALAPGR.

User response:
Check the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "VSAM Macro Instructions" for return code explanations.

ALAA045E   **OPEN err DSN: dataset, res cde: *code*.**

Explanation:
The OPEN request failed for the named data set.

User response:
Verify validity of the data set name. Check the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "Non-VSAM Macro Instructions" for reason code explanations.

ALAA046E   **GET err DSN: *dataset*, res cde: *code*.**

Explanation:
The GET request failed for the named data set.

User response:
Check the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "Non-VSAM Macro Instructions" for reason code explanations.

ALAA047E   **CAF error: Unable to CONNECT to SSID: *ssid*.**

Explanation:
The Call Attach Facility was unable to CONNECT to the named subsystem.

User response:
Verify that the subsystem name is correct and that is up and running.

ALAA048E   **CAF error: Unable to OPEN plan: *planname*.**

Explanation:
The Call Attach Facility was unable to OPEN the named PLAN.

User response:
Verify that the PLAN name is correct, that it has been bound, and that the USERID executing the job has execute authority on the plan.

ALAA049E   **OPEN failed for PLOGD data set.**

Explanation:
The OPEN request for PLOGD from program ALAREC2 failed.

User response:

Check for the existence of the PLOGD DD statement in your JCL. Also, verify that the data set exists and contains data from previous step (program ALAREC).

ALAA050E   **OPEN err for ROWDATA, res cde: *code*.**

Explanation:
The VSAM file associated with the ROWDATA DD statement could not be opened.

User response:
Check the reason code included with the message in the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "VSAM Macro Instructions" for descriptions of VSAM OPEN errors.

ALAA051E   **GET err for ROWDATA, res cde: *code*.**

Explanation:
The VSAM file associated with the ROWDATA DD statement could not be written to.

User response:
Check the reason code included with the message in the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "VSAM Macro Instructions" for descriptions of VSAM GET errors.

ALAA052E   **PUT err for ROWDATA, res cde: *code*.**

Explanation:
The VSAM file associated with the ROWDATA DD statement could not be written to.

User response:
Check the reason code included with the message in the IBM document "DFSMS Macro Instructions For Data Sets" in the chapter "VSAM Macro Instructions" for descriptions of VSAM PUT errors.

ALAA053E   **OPEN error for DD FRMTFILE.**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA054E   **OPEN error for DD CATFILE.**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA056E   **Invalid update image request; must be P or O only.**

Explanation:
An invalid update image was requested. The only acceptable values are P (pre-update image) and 0 (post-update image).

User response:
Correct the input and retry.

ALAA055E	OPEN error for DD SQLOUT.

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set, in this case should be SYSOUT=*.

ALAA057I	There is no data to process for the details report. Job stops.

Explanation:
This message will occur if a detail report is being run but the general report generated no data. It will also occur if the filters for the detail report filtered out all the data so there is nothing to process. This is not an error condition but rather an indication of end of processing as no detail report can be generated.

User response:
None required.

ALAA058W	Not authorized to view this table: *tablename*

Explanation:
This warning message occurs if a table is found while processing the details report which the user is not authorized to SELECT from. The table will not be included in the details report and no more processing is done for this table.

User response:
None required.

ALAA059W	The DB2 dataset is not found: *dataset*

Explanation:
This warning message occurs if the named DB2 VSAM data set cannot be allocated because it cannot be found. This would occur if the VSAM data set was deleted outside of DB2s knowledge but the object was not dropped in DB2. The table will not be included in the details report and no more processing is done for this table.

User response:
None required.

ALAA060E	No records written to ROWDATA DD. Job stops.

Explanation:

The VSAM file associated with the ROWDATA DD statement had no records written to it. This can occur only if there were also warnings generated for the detail report run. Check the job for messages in the WARNINGS DD. Processing stops here because there is no data to further process and hence, no detail report can be generated.

User response:
None required.

ALAA061W	Log RID mismatch with page RID; *variable1-variable5*

Explanation:
When reading the data page associated with a log record modification, it was determined that the data page row ID did not match the log page row ID. This causes processing to shutdown for this particular table. The table is identified by the variables in the message, listed as such: page number, row ID, DBID, PSID, and OBID. These values are all from the log record.

User response:
This can occur if a table was REORGed, was recreated, or data is still in the buffer pool (this is very likely during testing of product where actions were performed very recently). If this latter case is likely, retry after a QUIESCE on the database. Otherwise, use a log-forward mode process.

ALAA062E	No default SSID set; please choose option 0 for User settings

Explanation:
A default SSID must be set before other actions can be performed. This message will be generated for each user the first time they enter the product, or the first time they enter a new version/release if the application ID changes. You must provide information for all required fields on the User settings panel. After you complete, press **Enter** to save the values.

User response:
After you complete the User settings panel and you press **Enter**, the values are saved and this message will not appear again.

ALAA063W	OPEN failed for auth: *dataset*

Explanation:
While attempting to open the DB2 VSAM data set, an authorization failure occurred. You are not authorized to OPEN this data set. This table/table space will not be included in the final details report.

User response:
None required.

ALAA064E	DYN ALLOC ERR: DDname: *ddname*, RET: *code*, RES: *code*, DSN: *dsname*

Explanation

An internal dynamic allocation attempt failed for the specified ddname. The return code and reason codes are SVC 99 standard codes and can be found in the IBM manual "Authorized Assembler Services Guide" in the chapter "Requesting Dynamic Allocation Functions". The most common codes likely to be encountered are:

- 0210/0410 - ddname unavailable or in use by another user.
- 0420 - ddname associated with an already opened data set.
- 1708 - data set associated with ddname not found.

User response:
If the return and reason codes provide enough information, correct the error and retry. If you cannot resolve this issue, contact IBM Customer Support.

ALAA065E	Unsuccessful OPEN of *ddname*

Explanation:
An unsuccessful OPEN was issued for the specified ddname. The data set may not be allocated or already open.

User response:
Logging off TSO may resolve this problem.

ALAA066E	ALAPARMS file bad for *ssid*, missing configuration information.

Explanation:
For the SSID specified on the message, there is missing common configuration information.

User response:
Verify with your product administrator that all the necessary configuration information has been supplied for the SSID.

ALAA067E	ALAPARMS file does not have requested SSID: *ssid*

Explanation:
The control file has no entry for the specified subsystem.

User response:
Edit the control file ("Setup" option) to add the specified subsystem.

ALAA068E	DYN:DSNAME: *dsname*, RET: *code*, RES: *code*, MORE INFO: *text*

Explanation

An internal dynamic allocation attempt failed for the specified dsname. The return code and reason codes follow after the DSName field. These codes are SVC 99 standard codes and can be found in the IBM manual "Authorized Assembler Services Guide" in the chapter

"Requesting Dynamic Allocation Functions". The most common codes likely to be encountered are:

- 0210/0410 - ddname unavailable or in use by another user.
- 0420 - ddname associated with an already opened data set.
- 1708 - data set associated with ddname not found.

User response:
If the return and reason codes provide enough information, correct the error and retry. If you cannot resolve this issue, contact IBM Customer Support.

ALAA069E	ALAPARMS file bad for *ssid*, missing DB2 Log Analysis parm data.

Explanation:
For the SSID specified on the message, there is missing configuration data.

User response:
Verify that DB2 Log Analysis has been properly configured by your product administrator and that all needed parameters were provided for the failing subsystem.

ALAA070W	Uncommitted URID found and bypassed; *urid, dbid, psid, obid*

Explanation:
An uncommitted URID was found that matched the filter criteria. This URID is not included in the report, as it cannot be determined at this time if the work will be committed or aborted. The variables listed are for the URID and the DBID, PSID, and OBID of the table affected, respectively.

User response:
Re-run the job at a later time to resolve the warning as the URID will eventually be either committed or rolled back.

ALAA071E	LOB materialization requested, but LOB data not available.

Explanation:
LOB materialization was requested, but LOB data is not available because it was not included in the general/detail reports. You must include LOB data if you wish to materialize and browse LOB column data. The modefile indicates such an action is not present. You must set the "Include LOB Data" field on General Report panel to Y and re-run both the General and Detail reports. Another possibility is that the job identifier does not identify a previous General/Detail report run which did include the LOB data.

User response:
Either change the job identifier field, or re-run General/Detail report jobs.

ALAA072E	**Edit proc error: name:** *name*; **retcde:***code*; **rescde:***code*

Explanation:
The named edit procedure returned an error while attempting to decode the row. The named edit procedure returned the listed return and reason codes. Processing halts.

User response:
If you cannot resolve this issue, contact IBM Customer Support.

ALAA073E	**Dictionary mismatch:** *dbid*, *psid*, *pgno*, *rlogpoint*, *dlogpoint*

Explanation:
A row was found that needed to be decompressed, but the current dictionary was created after the row was written to the log. The dictionary cannot be used to decompress the row. The variables listed show the DBID, PSID, page number, and logpoint for the row, followed by the dictionary logpoint.

User response:
This job should be re-run using log-forward mode.

ALAA074E	**OPEN error for DD WARNINGS**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA075E	**Field proc error: name:***name*; **retcde:***code*; **rescde:***code*

Explanation:
The named field procedure returned an error while attempting to decode the column. The named field procedure returned the listed return and reason codes. Processing halts.

User response:
If you cannot resolve this issue, contact IBM Customer Support.

ALAA076I	**Panel values successfully updated from mode file.**

Explanation:
The panel variables have been updated from the entered mode file.

User response:
None

ALAA077E	**MODEFILE DSN not entered. MODEFILE operation cannot be done**

Explanation:

You did not specify a modefile data set name.

User response:
Correct and retry.

ALAA078E	**Incremental IC merge err:** *icname*

Explanation:
While attempting to merge the named incremental image copy into the full image copy, a page mismatch occurred. The full image copy data set (TEMPIC) hit an end-of-file condition while trying to match the current incremental page number.

User response:
Verify the incremental image copy name is a valid and expected data set for the current process.

ALAA079I	**Mode file successfully converted. V3.1 DSN:** *ddname*

Explanation:
The given mode file was successfully converted to the version needed for Log Analysis Tool version 3.1.

User response:
None.

ALAA080E	**OPEN failure on full image copy:** *dsname*

Explanation:
The OPEN failed for the dsname.

User response:
Verify that the data set name is valid and that you are authorized to browse this data set.

ALAA081E	**OPEN failure on incremental image copy:** *dsname*

Explanation:
The OPEN failed for the dsname.

User response:
Verify that the data set name is valid and that you are authorized to browse this data set.

ALAA082W	**OPEN failed for auth:** *dataset*

Explanation:
While attempting to open the image copy data set, an authorization failure occurred. You are not authorized to OPEN this data set. The associated table space will not be included in the final details report.

User response:
None required.

ALAA083E	**Open failure on dsname:** *dsname* **during mode file conversion**

Explanation:
The open failed for the dsname. The dsname should not exist before converting and the user should have permission to create the data set. The DSNAME is built

internally based upon the values found in the mode file that is being converted.

User response:
Verify that the data set name does not exist and that you are authorized to create this data set

---

ALAA084E **No full image copy found:** *dbname, tsname, dsnum, rba*

Explanation:
A full image copy was not found with a RBA less than or equal to the RBA shown in the message. The product cannot continue processing. These values are all from ties for this table space. The variables shown are DBNAME, TSNAME, DSNUM, and RBA of log record.

User response:
Verify the validity of the shown RBA and then check your SYSIBM.SYSCOPY table to verify the state of your image copy, database name, table space name.

ALAA085E **UNDO and REDO SQL mutually exclusive. Set only one to Y.**

Explanation:
A request was made to generate a details report generating both UNDO and REDO SQL. This is not permitted.

User response:
Select only one of these options.

ALAA086E **Bad commit scope value. Enter value between 000 to 999 only.**

Explanation:
The value entered for commit scope must be between 000 and 999 only.

User response:
Correct the value and retry.

ALAA088W **SQL bypassed, LOB/XML data found on table:** *&v1, &v2*

Explanation:
A LOB or XML column was found in a table modified during the specified log range. SQL can only be generated on such column types if panel field "Include LOB/XML data=Y" on the "Generate Database Activity Report (General)" panel. &v1 represents the table owner, and &v2 is the table name. The identified table activity will not be included in the SQL.

User response:
If you wish to include LOB/XML data types in SQL, you must re-run the job setting panel field "Include LOB/XML data=Y", or do not request SQL.

ALAA089W **SQL bypassed, catalog table(s) present.**

Explanation:

SQL cannot update catalog tables, and therefore, SQL generation is bypassed for all the catalog tables reported on by the detail report. Non-catalog tables will still be included in the generated SQL, if any.

User response:
None required.

ALAA090E **Loadlib not APF authorized.**

Explanation:
The product load libraries need to be APF authorized before trying to execute any of the modules. Some of the programs in the load libraries must be APF-authorized to run.

User response:
Include the load libraries (SALALOAD and SFECLOAD) as part of your authorized list.

ALAA091E **Error in reading SYSLGRNX, return code:** *code*

Explanation:
An error occurred while attempting to read the SYSLGRNX directory table.

User response:
See your job log output for additional diagnostics relating to this error (look for possible eye-catcher of "FEC#REPO"). Report this error to your product administrator, but as a temporary workaround, you can set the "Misc flags" field to "X" on the General Report panel to bypass usage of SYSLGRNX and retry.

ALAA093E **LOB PDS name field is blank, but LOBs found in data.**

Explanation:
A LOB type column was found in the table while generating load cards, but no LOB PDS name was specified on the panel.

User response:
Enter a LOB PDS name value on the panel and regenerate the JCL.

ALAA094E **LOADFILE OPEN failure.**

Explanation:
An error occurred while attempting to open the DD LOADFILE.

User response:
Verify the existence of the DD statement in the JCL.

ALAA095E **DDLFILE OPEN failure.**

Explanation:
An error occurred while attempting to open the DD DDLFILE.

User response:
Verify the existence of the DD statement in the JCL.

ALAA096E **LOADCARD OPEN failure.**

Explanation:
An error occurred while attempting to open the DD LOADCARD.

User response:
Verify the existence of the DD statement in the JCL.

ALAA097E   **Job run identifier must start with a letter.**

Explanation:
The job run identifier will be appended to a data set name as a low level qualifier. It must start with a character between A-Z.

User response:
Correct and retry.

ALAA098E   **Object list filter value too long:** *filter*.

Explanation:
An object list filter value was found to be too long. The bad filter value is shown in the error message for a length long enough to indicate the error.

User response:
Correct and retry.

ALAA099E   **Split LOADFILE open failure:** *v1*.

Explanation:
An error occurred while attempting to open LOADFILE DD. The bad DD value is shown in the error message.

User response:
If dynamic allocation is overridden, verify that the DD statement shown in the error message is hardcoded in your JCL.

ALAA101E   **EDICT DD OPEN failure**

Explanation:
The EDICT DD failed to open.

User response:
Verify its existence in the job step.

ALAA102W   **Log-forward required for detail report**

Explanation:
Data was found during the general report process that requires a log-forward process for generating the detail report. Such data could be reorg, load, or mass-delete log record(s).

User response:
Re-run as a log-forward process.

ALAA103E   **Page column value on line indicated is invalid.**

Explanation:
The page field value is invalid. Enter either a space, a Y (yes), or N only.

User response:
Correct the input and retry.

ALAA104E   **Error during sort. See SORTMSGS DD for more information.**

Explanation:
An error occurred while sorting the inline image copy. More information is available by examining the hardcoded SORTMSGS DD statement in the JCL output.

User response:
If you cannot resolve this issue, contact IBM Customer Support.

ALAA105E   **ISPSLIB member ALAIN4 bad record starting** *var1*

Explanation:
A bad record was found in the specified member.

User response:
Use the record identified to locate the bad record and correct the entry.

ALAA106E   **TICSPECS DD OPEN failure.**

Explanation:
An inline image copy was found, but while attempting to read the TICSPECS DD, an OPEN failure occurred.

User response:
Verify the existence of this DD in the JCL.

ALAA107E   **No usable image copy found for tablespace** *tsname*

Explanation:
A full image copy was not found for the specified table space. An image copy must exist prior to the time of your earliest log records. If none can be found earlier than that time, no usable image copy exists. For example, if you insert many records and subsequently take an image copy of the table space, that image copy is not usable for the inserts because it occurred after them.

User response:
Verify that a full image copy is available prior to the time of your earliest log record actions for the table space or remove this table space from the original search criteria. This error can only happen in log-forward mode; for the table that generated the error, you can attempt to run Log Analysis Tool in log-backward mode

---

ALAA108E   **"Column Value" field cannot be spaces.**

Explanation:
The "Column Value" field cannot be left as spaces. If only spaces are desired as the value, they must be enclosed in single quotes. Otherwise, a non-blank value must be provided.

User response:
Correct and retry.

ALAA109E   **Disk primary and secondary space values must be numeric.**

Explanation:
The primary and secondary space values must be between 0000 and 9999 only, without any non-numeric characters. This field is only validated if "Device Type" is not equal to "T".

User response:
Enter numeric values.

ALAA110E   **No temporary image copy data found; go to Defaults.**

Explanation:
An attempt was made to run a detail report in log-forward mode. However, it was found that required data was missing. It is likely the default values for log-forward mode have not been properly set up yet.

User response:
Go to option "D" off the primary menu and be sure proper values have been supplied under the "Temporary Image Copy allocation:" heading. These fields are required for proper execution while in log-forward mode.

ALAA111E   **UNCRE DD or UNCREB DD failure on OPEN.**

Explanation:
Continuous mode was requested, but no UNCRE DD or UNCREB DD was found in the JCL or the OPEN failed for other reasons.

User response:
Verify the existence of this DD statement in the JCL.

ALAA113E   **Object type can be "T" for table, or "S" for tablespace only.**

Explanation:
The specified object type value was incorrect. If one of the values is not used, the entire line must be left blank.

User response:
Correct the input and retry.

ALAA114E   **OPEN error for DD SFMTFILE**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA115E   **OPEN error for DD CHFILE**

Explanation:

An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA116E   **OPEN error for DD HVFILE**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA117E   **Table owner: *id*, Table: *tbname* ;not found in DB2**

Explanation:
The specified table owner and name was not found in DB2. The table may have been dropped.

User response:
None required.

ALAA118E   **OPEN error for DD SQLIN**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA119E   **OPEN error for DD SQLTIN**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA120E   **OPEN error for DD SQLOUT.**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA121E   **Processing stops due to bad table list.**

Explanation:
A table was requested for the static program generation, but the table was not found in the input. Therefore, there is no data to further process.

User response:

Verify the correctness of the table specification in the TABLE DD statement and retry.

ALAA122E **DBID is blank, but PSID and/or OBID are not; invalid.**

Explanation:
DBID must be entered if PSID and/or OBID are entered.

User response:
Either blank out the entire line, or fill in the correct DBID value.

ALAA123E **DBID is nonblank, but PSID is blank; invalid.**

Explanation:
If DBID is nonblank, there must minimally be a value entered for PSID. OBID does not have to be specified if only a table space is being filtered on. If OBID is nonblank, PSID must also be nonblank.

User response:
Correct the entry and retry.

ALAA125E **Bad continuous mode file name.**

Explanation:
You requested the use of a continuous mode file. The name as entered is invalid. This must be a valid file name.

User response:
Correct and retry.

ALAA126E **Contiguous group values must be either all blank or all numeric.**

Explanation:
You specified a non-blank contiguous group value for at least one log entry therefore all entries must have a value. That is, the logs are either all contiguous, or they are not. If they are not, each log entry must have a non-blank value.

User response:
Correct and retry.

ALAA127E **No log datasets were specified, you must enter at least one.**

Explanation:
You pressed "ENTER" to accept the list of specified log data sets, but none were found. If you wish to bypass use of the list, hit "PF3", otherwise enter at least one log data set name. A line of spaces (from top to bottom in the log data set column stops the list, so be sure the first line is not spaces.

User response:
Correct and retry.

ALAA130E **IDCAMS job failure. See WTO messages in job output.**

Explanation:
The job failed while processing IDCAMS for the active log. The output from SYSPRINT has been written to your job log in the form of WTO messages. See these messages for help on the cause of the failure.

User response:
If you cannot resolve this issue, contact IBM Customer Support.

ALAA131E **Existence of old IDs requires both old and current/new OBID.**

Explanation:
You requested old IDs but omitted the old OBID and/or the current/new OBID This is not permitted. When requesting old IDs, all objects must be fully qualified with the current/new full set of DBID/PSID/OBID and old DBID/PSID/OBID.

User response:
Correct the input for the invalid line of IDs and retry.

ALAA132E **Object name fields cannot be spaces.**

Explanation:
The line has an object type = "T" or "S", and therefore object names representing either a table or a database/table space must not be left blank.

User response:
Set the entire line to spaces, or fill in these names.

ALAA133E **Object type is blank, therefore entire line must be blank.**

Explanation:
The line has an object type which is blank (or spaces), and therefore the entire line must be left blank.

User response:
Leave the entire line blank and retry.

ALAA134W **Warning: At least one utility record found.**

Explanation:
The detail report found the existence of a utility record(s) and/or a mass delete record(s). These records are identified by an "X" in the general report. They have been bypassed because only a log-forward process can handle these types of records and you have requested a log-backward process. If the "X" type record was the first record shown for a specific table/table space, running log-backward can yield good results.

User response:
Evaluate and retry if necessary.

ALAA135W **Warning: A BSDS member could not be found.**

Explanation:

While processing the named BSDS for the group, another BSDS was found which could not be allocated. This data set does not exist as the allocation attempt failed on a "1708" reason code. You likely have old and outdated BSDS names in your configured BSDS. This is treated as a warning only and the name of any such BSDS is written to your job log in the form of a WTO message.

User response:
Confirm that the named BSDS is not needed for the run.

ALAA136E	Unsupported column type found: &V1

Explanation:
The current function found a column type that is not currently supported.

User response:
Remove the columns owning table from your object list, or add it as an excluded-type filter.

ALAA137E	Bad partition value. Enter a valid number between 0-4096.

Explanation:
The value entered for the partition must be between 0 and 4096 only. Spaces are also permissible and equate to a value of 0.

User response:
Correct the value and retry.

ALAA138E	A database/tablespace filter is required with a partition filter.

Explanation:
A non-zero value was specified for the partition filter, but no database and table space filter was specified and/or a table filter was also specified.

User response:
Either request a single, non-wildcarded database/ table space filter (with no table filter), or set the partition filter to zeros.

ALAA139E	Table filters are not permitted with a partition filter

Explanation:
A non-zero value was specified for the partition filter, but a table filter was also requested. This is not allowed.

User response:
When using a partition filter, specify only a single database/table space filter to be associated with the requested partition filter.

ALAA140E	Continuous mode and logs specification mutually exclusive.

Explanation:

Continuous mode has been requested along with a setting of Y for the **Specify Logs** flag. This is not permitted. Continuous mode requires the product to determine all needed log data sets.

User response:
Reset either of these flags and retry

ALAA141E	Bad contiguous group specification. Must be 00-99 only.

Explanation:
You specified a bad value for a contiguous group field. The only allowable values are from 00 to 99.

User response:
Correct the input and retry.

ALAA142E	Valid operator values are EQ, NE, GT, GE, LT, LE, LK, NL, IC, XC, CH

Explanation:
You specified an invalid operator field. The field can be only one of those listed.

User response:
Change the field and retry.

ALAA143E	"Cond" field can only contain A, O,), (, or blank.

Explanation:
You specified an invalid "Cond" field. The field can either be "A" (AND) "O" (OR), or spaces to indicate there are no more conditions. If the field is not spaces, then there must be another conditional on the next line. Additionally, you may choose to nest your conditionals using up to 3 levels of parentheses. That is, any single line can contain from 1 to 3 left parentheses or right parentheses, and the total number of left and right parentheses must match.

User response:
Correct according to the explanation and retry.

ALAA144E	Bad advanced filter record: *string*

Explanation:
You specified an invalid advanced filter record. Examine your JCL and verify that any edited records have identical format to program generated messages.

User response:
Correct the listed bad entry and retry from the failing step.

ALAA146E	Use of ARCHLOG1 only and ARCHLOG2 only are mutually exclusive.

Explanation:
You requested use of both ARCHLOG1 and ARCHLOG2 only, which is not permitted. Only one of these can be requested exclusively.

User response:

If you wish to use either, that is the default behavior of the product and both should be set to N.

ALAA147E   **Unsuccessful OPEN of dsname**

Explanation:
An unsuccessful OPEN was issued for the specified dsname. Processing halts. This message will occur only if all logs (active and archive) with the same RBA range failed.

User response:
None required.

ALAA148E   **Unsuccessful LOCATE of dsname**

Explanation:
An unsuccessful LOCATE occurred while trying to resolve the dsname. This message will occur while the product attempts to resolve the DATA component name of an active log.

User response:
Verify the correctness of the dsname in the message.

ALAA150E   **Valid values are Y, N, or "O" only for field.**

Explanation:
For the **Show Rollbacks** field, specify only the following values: Y to include rollbacks in the output; N to exclude rollbacks from the output; "O" to show only rollbacks in the output.

User response:
Correct and retry.

ALAA151E   **Optional old IDs must have related new IDs.**

Explanation:
You specified a group of optional old IDs in the object list, but there were no IDs on the same line for the new IDs. This is not acceptable input. Old IDs must always correspond directly to new IDs.

User response:
Correct the input and retry.

ALAA152E   **Mismatch found between old and new IDs.**

Explanation:
You specified a line of both old and new IDs, but the IDs do not directly match. When specifying the optional new IDs, all IDs must be specified for both old and new. That is, the old DBID, PSID, and OBID must all be stated, as well as the new DBID, PSID, and OBID.

User response:
Correct the input and retry.

ALAA153E   **Object IDs list requested but no entries found.**

Explanation:
You requested use of object IDs, and hit ENTER to use the list, but no object IDs were found. If you wish to bypass the list, hit PF3, or enter at least one valid line of IDs.

User response:
Correct the input and retry.

ALAA154E   **Old object IDs detected in REFILE; must use log-forward mode.**

Explanation:
While running in log-backward mode, old object IDs were detected in the REFILE. These occur only when old object IDs were included in the general report.

User response:
You must run this detail report in log-forward mode, as log-backward mode does not support handling of previously dropped objects.

ALAA155E   **Old object IDs on one line require them on all lines.**

Explanation:
You specified the optional old IDs on at least one line, so they must be present on all lines. Running with old IDs is considered running in dropped-object mode. Running without old IDs is considered standard mode. You cannot mix these two modes.

User response:
Correct the input and retry.

ALAA156I   *message_text*

Explanation:
This informational message is generated during Log Analysis Tool processing. The *message_text* is intended to assist IBM Software Support in diagnosing product-related issues.

User response
No action is required; however, several specific *message_text* values and user responses follow:

- If the Log Analysis Tool job fails with S0C7 and message ALAA156I: S0C7 Object: *dbid/psid/obid* is issued, it is likely that the correct compression dictionary was not found. This is a common issue if KEEPDICTIONARY was not specified for a REORG LOG NO utility. If you receive this message, contact IBM Software Support.

- Message ALAA156I: LOG RECS BYPASSED DUE TO PIT RECOVERY (*<module name>*) indicates that partial recovery logpoints were found within the time frame and have been ignored. To bypass this check and process those log records, set the MISC FLAG value to P (Include partial recovery points).

ALAA157E   **Internal array in ALABSDS exhausted, processing halts.**

Explanation:
You have exceeded the maximum number of stored archive and active logs allowed. The maximum allowable number is approximately 2788, and is deemed excessive. This will likely only occur in a detail report run where one of the following situations has occurred: log-backward mode and the associated general report timeframe was very long ago or log-forward mode and the earliest image copy needed was created long before the associated general report timeframe.

User response:
Analyze your situation and take appropriate actions such as changing timeframes or detail report modes (e.g., change log-backward mode to log-forward).

ALAA158E   **Partition filter not allowed with wildcards.**

Explanation:
You have requested a partition filter, but also used a wildcard in the database or table space name field. This is not permitted. A partition filter can only be applied to a single table space.

User response:
Remove either the partition filter or the wildcard and retry.

ALAA159E   **An invalid ID was found on the line. Enter only decimal numbers.**

Explanation:
You have specified an invalid ID value. Only decimal values are permitted and these can only be between 0 and 99999. No alphanumeric or hexadecimal values are allowed.

User response:
Correct the input and retry.

ALAA160E   **Required fields from the general report panel have not been set.**

Explanation:
You are attempting to generate detail report JCL, but some required information is missing, and this information comes from your general report settings. You may be attempting to generate detail report JCL without ever having created general report JCL. This is not allowed. The detail report JCL uses output from the general report, so you cannot run a detail report without first running a general report.

User response:
Return to the general report panel (from main menu: **Generate database activity report (general)**) and start again.

ALAA161E   **A source and target table must both be specified, or both must be left blank.**

Explanation:
Either a source table is blank and the corresponding target table is specified, or a source table is specified and the corresponding target table is blank. You must specify both the source and target table, or leave both blank.

User response:
Correct the input and retry.

ALAA162E   **Invalid row length value. Enter a valid value between 00080 to 32752 only.**

Explanation:
The value entered for report row length must be between 00080 and 32752 only.

User response:
Correct the input and retry.

ALAA163E   **Specify logs option is incompatible with JSON parameters file.**

Explanation:
The process of creating a JSON parameters file does not allow specification of log files.

System action:
Panel processing halts and an error message is issued.

User response:
Correct the value and retry.

ALAA164E   **Expert options are incompatible with JSON parameters file.**

Explanation:
The "Create JSON parameters file" option does not allow specification of expert options.

System action:
Panel processing halts and an error message is issued.

User response:
Correct the value and retry.

ALAA165E   **The specified dataset could not be found in the MVS catalog.**

Explanation:
The specified dataset could not be found in the MVS catalog.

System action:
Panel processing halts and an error message is issued.

User response:
Verify the correctness of the dataset name. Enter a valid name.

ALAA166I　　　　**The control file record for DB2 subsystem and SYS have been successfully updated.**

Explanation:

System action:
None.

User response:
No action required.

ALAA167E　　　　**Specified value is out of permitted bounds.**

Explanation:
The specified value is incorrect and out of the permitted bounds.

System action:
Panel processing is stopped and an error message is displayed.

User response:
Enter the correct value and try again.

ALAA170E　　　　**Both table owner and name must be filled in, or both left spaces.**

Explanation:
You requested a table owner or name, but not both. Either leave these fields both blank, or fill them both in. The table name must be fully qualified.

User response:
Correct the input and retry.

ALAA171E　　　　**Invalid number of hex digits in DD filters input:** *hex_string*

Explanation:
When using hexadecimal values in the advanced filters, there must be at least two digits. If there are more than two digits, the total number of digits must be even.

User response:
Correct the input in the JCL and retry.

ALAA172E　　　　**SQL stmt not valid due to: WHERE clause could not be generated.**

Explanation:
The WHERE clause could not be generated, and therefore no SQL statement could be produced. This is an internal error.

User response:
Contact IBM customer support.

ALAA173E　　　　**Invalid or unbalanced quotes found in DD filters input:** *hex_string*

Explanation:

When using hexadecimal values in the advanced filters, the string that is entered must start with X, followed by a single quote and valid hexadecimal digits, and must end with a single quote. If you are attempting to use the LK or NL operator, put the % after the final single quote, not inside the quotes.

User response:
Correct the input in your JCL and retry.

ALAA174E　　　　**"ALTER TABLE" condition found on table:** *table*

Explanation:
A condition was found whereby there was no data for a currently existing column. This means the table was altered, and the static SQL program cannot be generated because the log data reflects two different states of the table (before the ALTER, and after the ALTER). Processing halts. You cannot use the **Generate Static SQL Program** option against log data that spans the before and after image of the altered table.

User response:
Run only dynamic SQL that is generated by the product.

ALAA175E　　　　**Unable to LOAD DB2 module DSNHDECP.**

Explanation:
A LOAD was issued against module DSNHDECP which was unsuccessful.

User response:
Verify that the "Setup" options for this subsystem correctly identify a DB2 load library containing this module. Typically this module would reside in *.SDSNLOAD or *.SDSNEXIT type libraries.

ALAA176E　　　　**OPEN error for DD DECLFILE**

Explanation:
An OPEN attempt failed for this DD statement.

User response:
Check for the inclusion of this DD statement in your JCL step and for correct specification of the associated data set.

ALAA177E　　　　**Table owner/name fields must both be specified, or both left blank.**

Explanation:
Either a table owner field is spaces and the name is not spaces, or a table owner is not spaces and the name is spaces. These are invalid specifications. Either specify both the owner and name, or leave both blank.

User response:
Correct the input and retry.

ALAA178E **Invalid hexadecimal character found in DD filters input: *hex_string***

Explanation:
When using hexadecimal values in the advanced filters, the string entered must start with X, followed by a single quote, valid hexadecimal digits (012345678ABCDEF), and end with a single quote. If you are attempting to use the LK or NL operator, put the % after the final single quote, not inside the quotes.

User response:
Correct the input in your JCL and retry.

ALAA179E **"Data Only= Y" mutually exclusive with "Header" and "Change Flag".**

Explanation:
A request to process data only precludes the possibility of including any DB2 Log Analysis Tool information. Therefore, the "Header" and "Change Flag" fields must both be set to N if the "Data only" field is set to Y.

User response:
Correct the input and retry.

ALAA180E **Database and tablespace names must not be left blank.**

Explanation:
You specified "Create target tables = Y", but have omitted values for the database and table space names. These names are required when creating the target tables. These values specify where the target tables will be created.

User response:
Either specify values for database and table space names, or change the setting for creating target tables to N.

ALAA181E **"Change Flag = Y" requested without UPDATE pre and post images.**

Explanation:
You specified "Change Flag = Y", but the "LOAD specification" field does not include both pre and post row images for UPDATEs. The change flag is used to indicate a change in a column value before and after an UPDATE only, so excluding both of these from the LOAD specification value is conflicting.

User response:
Either set the change flag to N, or include pre and post row images in the LOAD specification.

ALAA182E **No rows qualified for LOAD processing; job stops.**

Explanation:

Based on the source tables requested and/or the LOAD specification value, all rows examined from the detail level report were bypassed. This results in no data to process, and the LOAD job stops. For example, if there are only INSERTs in your detail report, and you entered a LOAD specification value = 8 (UPDATE PRE), no records will be processed. Any ROLLBACK records are also excluded from processing.

User response:
Verify that any source tables requested are actually in the result set from your detail report. Also verify that the LOAD specification value would not exclude all detail report records.

ALAA183E **Error: ROWID column type found in source table: table**

Explanation:
Target tables were requested to be created, but the source table has a ROWID column. DB2 imposes a restriction on the LOAD utility such that a ROWID column must be defined as "GENERATED BY DEFAULT", which requires an index. This product does not create indexes on target tables.

User response:
You must create any necessary target tables and their indexes outside the realm of this product. Set the field **Create target tables** to N after creating the target tables and indexes outside of the product.

ALAA184E **Source table not found in DB2 catalog: table**

Explanation:
The source table was not found in the DB2 catalog. It may have been dropped since the detail report was run. Processing cannot continue as the needed column information from that table is no longer available.

User response:
None required.

ALAA185E **Source table not found in CATFILE: table**

Explanation:
The source table was not found in the CATFILE, which is produced during the general report run. Either a bad source table was specified, or the CATFILE has been re-written by a subsequent run of another general report. The values shown for the source table reflect the expected DBID, PSID, and OBID which could not be found in the CATFILE

User response:
Ensure that your source table information is accurate.

ALAA186E **Group number of failing record: value**

Explanation:

The static program execution failed while processing a record. The number shown is the binary value of the SQLRGRP# field in the SFMTFILE DD file.

User response:
Issue this find command while browsing that file: FIND X"n" 13 ... where n = the value shown in the message and 13 is the relative column position of the field SQLRGRP#. Also, use single quotes rather than the double quotes shown in the FIND command. This information, along with the diagnostics provided in the MESSAGES DD file, will allow you to diagnose the problem.

ALAA187E	ASCII or UNICODE table data found with pre-DB2 V7 subsystem.

Explanation:
The static program execution halted because either ASCII or UNICODE data was found, and the subsystem targeted for SQL execution is pre-DB2 version 7. There is only support for non-EBCDIC type table data while using the static SQL program against DB2 subsystems at (or greater than) version 7. You can only use the generated UNDO or REDO SQL in an application such as SPUFI in such cases.

User response:
None required.

ALAA188E	Error message must be prefixed ALAA

Explanation:
You entered an invalid error message. All error messages must be prefixed with ALAA, and be of the full format ALAAnnn where nnn is the message number.

User response:
Correct the input and retry.

ALAA189E	Error message entered was not found.

Explanation:
The error message entered is formatted properly, but not found in the message library.

User response:
Verify the correctness of the message you entered. If correct, contact your product administrator to verify the correctness of the message library allocated in the product startup clist (ALAFRONT).

ALAA190E	Dictionary data missing in: value

Explanation:
Either some or all of the needed dictionary was not found for the identified object (format is DBID/ PSID/partition number). This should only occur if a "REPAIR SET TABLESPACE .. NOCOPYPEND" or "START(db) ACC(FORCE)" was run. Running such an action invalidates all types of DB2 recovery scenarios, as well as using this tool to report on the object.

User response:
If you have not run either DB2 action, contact your product administrator. A log-backward detail report can be attempted to bypass this situation, but in some cases, only a log-forward detail report can be used (and log-forward is only case where this situation can occur). Filtering out the identified object is also a remedy.

ALAA191E	No data to process, job processing stops.

Explanation:
Either all of the data was filtered out because of column data filters, or there was no data to process from the results of the detail report.

User response:
Verify the possibility of each condition, and either change your column data filters, or re-run the general/ detail reports and verify the existence of reported data (at least one line of log record activity data).

ALAA192I	No SQL was executed; table filter/ restart file may be cause.

Explanation:
A table filter may have prevented any SQL from being executed. If the SQL did not include the named table, all SQL would have been bypassed. Another possible cause is that a restart was requested, and the SQL statement number to restart on is incorrect. Verify you actually needed to restart here, that the previous failing run generated the restart file you are using This is informational only, no errors occurred.

User response:
None required.

ALAA193E	Static SQL program maximum length exceeded; processing stops

Explanation:
It was determined that the generated static SQL would be too large to successfully assemble (compile).

User response

- Option 1: Request a single table to process each table individually. If there is only one table in your detail report output, then follow option 2 below. Filtering on a single table, may not guarantee to eliminate the problem. This is true if the single table has more than 45 columns, which are nullable. The SQL must include updates, deletes, and inserts.
- Option 2: Use option 5 of the main menu to process the SQL dynamically in batch.

ALAA194E **Static SQL program found nullable column in WHERE clause.**

Explanation:
It was determined that a nullable column would be included in a WHERE clause. The static SQL program cannot tolerate nullable columns in WHERE clauses.

User response:
You must either process this SQL via the dynamic SQL in batch option, or through any other facility, such as DB2 SPUFI. Static SQL is not an option for processing this SQL.

ALAA195E **A Mass Delete type record was found, must use log-forward mode only.**

Explanation:
A mass delete record was found in the REFILE (input file generated in the general report process).

User response:
The only way to resolve such a record is to use a log-forward detail report ("Apply log forward = Y" on detail report panel). Either run the detail report in log-forward mode, or eliminate the related segmented table from your detail report by filtering on other specific objects. For example, filter only on table X when table Y had the mass delete action.

ALAA196E **IC,XC column data operators are mutually exclusive.'**

Explanation:
While generating column data filters, you used both IC (include) and XC (exclude) on the same table. This is not permitted. Either name all columns you wish to include, or name all the columns you wish to exclude, but do not mix these operators. It is implicit that if you are naming columns to be included that any columns not specified will be excluded. It is also implicit that if you are naming columns to be excluded, any columns not specified will be included.

User response:
Correct the input and retry.

ALAA197E **Invalid or unbalanced quotes in DD filters input: *string***

Explanation:
When using column filters, if you have a beginning quote, you must end the string with an ending quote.

User response:
Correct the input in your JCL and retry.

ALAA198E **DSN system symbol error: &V1**

Explanation:
A prefix of the dataset name is incorrect. The '&V1' variable represents the reason of the error.

User response:
Correct the entry and try again.

ALAA200E **INTERFACE: ALAIPARM DD missing**

Explanation:
The required DD ALAIPARM has not been allocated. This DD holds the interface records which communicate between other products and DB2 Log Analysis Tool.

User response:
Allocate the required DD and retry.

ALAA201E **INTERFACE: Unknown record type in ALAIPARM DD: text**

Explanation:
The file associated with DD ALAIPARM has an invalid record. The first 13 characters of the record are shown in the error message.

User response:
Attempt to resolve this issue using the error messages. If you cannot, contact IBM customer support.

ALAA202E **INTERFACE: UNDO SQL indicator invalid.**

Explanation:
UNDO SQL record in DD ALAIPARM file is invalid.

User response:
Valid values are Y or N only.

ALAA203E **INTERFACE: REDO SQL indicator invalid.**

Explanation:
REDO SQL record in DD ALAIPARM file is invalid.

User response:
Valid values are Y or N only.

ALAA204E **INTERFACE: COMMIT SCOPE record invalid.**

Explanation:
COMMIT SCOPE record in DD ALAIPARM file invalid.

User response:
Valid values are 000 through 999 only.

ALAA205E **INTERFACE: CATALOG DATA record invalid.**

Explanation:
CATALOG DATA record in DD ALAIPARM file invalid.

User response:
Valid values are Y or N only.

ALAA206E **INTERFACE: START DATE record invalid.**

Explanation:
START DATE record in DD ALAIPARM file invalid.

User response:
Valid values are of format YYYY/MM/DD only.

ALAA207E   **INTERFACE: END DATE record invalid.**

Explanation:
END DATE record in DD ALAIPARM file invalid.

User response:
Valid values are of format YYYY/MM/DD only.

ALAA208E   **INTERFACE: START TIME record invalid.**

Explanation:
START TIME record in DD ALAIPARM file invalid.

User response:
Valid values are of format HH:MM:SS only.

ALAA209E   **INTERFACE: END TIME record invalid.**

Explanation:
END TIME record in DD ALAIPARM file invalid.

User response:
Valid values are of format HH:MM:SS only.

ALAA210E   **INTERFACE: END DATE < START DATE is invalid.**

Explanation:
The requested end date must be greater than or equal to the requested start date.

User response:
Correct and retry.

ALAA211E   **INTERFACE: END DATE/TIME <= START DATE/TIME.**

Explanation:
The requested end date and time must be greater than or equal to the start date and time.

User response:
Correct and retry.

ALAA212E   **INTERFACE: SSID is spaces.**

Explanation:
The requested SSID is spaces. SSID must be provided in a DD ALAIPARM record.

User response:
Correct and retry.

ALAA213E   **INTERFACE: No OBJECT_IDS record was found in group.**

Explanation:
Every GROUP_RECORD set must contain at least one OBJECT_IDS record.

User response:
Correct and retry.

ALAA214E   **INTERFACE: Mode=4 cannot have OBJECT IDS request.**

Explanation:
Mode = 4 was specified in the header, but OBJECT IDS were requested in a group record. This is not allowed. Mode = 4 implies that log records will be searched for any deletes associated with catalog tables only, so OBJECT IDS are irrelevant in this mode.

User response:
Remove any OBJECT IDS records.

ALAA215E   **INTERFACE: Partition invalid.**

Explanation:
A partition value can only be 0 through 254.

User response:
Correct and retry.

ALAA216E   **INTERFACE: Update type invalid.**

Explanation:
Update filter can only be Y or N.

User response:
Correct and retry.

ALAA217E   **INTERFACE: Insert type invalid.**

Explanation:
Insert filter can only be Y or N.

User response:
Correct and retry.

ALAA218E   **INTERFACE: Delete type invalid.**

Explanation:
Delete filter can only be Y or N.

User response:
Correct and retry.

ALAA219E   **INTERFACE: AUTO SUBMIT request invalid.**

Explanation:
AUTO SUBMIT can only be "E" (edit), "S" (submit), or N (just create the JCL, neither edit nor submit it).

User response:
Correct and retry.

ALAA220E   **INTERFACE: MODE invalid.**

Explanation:
The requested MODE is invalid. Valid values are "1" (general report only), "2" (detail report), "3" (general and detail report), "4" (generate DROP type DDL).

User response:
Correct and retry.

ALAA221E   **INTERFACE: OBJECT_IDS record was spaces, invalid.**

Explanation:
An OBJECT_IDS record was found without any data. This is an invalid record and must either be omitted or completed with valid IDs.

User response:
Correct and retry.

ALAA222E   INTERFACE: UNDO and REDO SQL both requested; invalid

Explanation:
You can request either UNDO SQL or REDO SQL, not both.

User response:
Correct and retry.

ALAA223E   INTERFACE: Partition request invalid w/multiple OBJECT_IDS records.

Explanation:
A partition request was found in a group that also contained multiple OBJECT_IDS records. Since a partitioned table space can only contain one table, only one OBJECTS_IDS record is allowed when partition is requested.

User response:
Correct and retry.

ALAA224E   INTERFACE: SQL DSN is too long; max 44 characters (non PDS).

Explanation:
A requested data set name was too long. The SQL DSN data set can only be a maximum of 44 characters long, and no member specification is allowed.

User response:
Correct the entry and retry.

ALAA225E   INTERFACE: ALA_INTERFACE is not first record in DD ALAIPARM.

Explanation:
The first record of a valid data set associated with DD ALAIPARM must be ALA_INTERFACE.

User response:
Correct the file and retry.

ALAA226E   INTERFACE: Two groups found while in Mode=4.

Explanation:
You requested Mode=4 in the header, but included multiple group records. This is not allowed. A Mode=4 run implies a single start/stop time range.

User response:
Remove all but one group record with start/stop time ranges.

ALAA227E   INTERFACE: SQL DSN cannot be a PDS, only flat files valid.

Explanation:
The requested SQL data set name contained a member name, indicating it was a PDS. Only sequential flat files are allowed here.

User response:
Specify the SQL DSN as a flat file and retry.

ALAA228E   INTERFACE: JCL DSN requested but not found.

Explanation:
The requested JCL data set name could not be found. The use of this data set requires that it exists.

User response:
Either remove the JCL DSN control card from the ALAIPARM data set, or create the data set prior to invoking the interface.

ALAA229E   INTERFACE: RBAs, LRSNs, and date/times are all mutually exclusive.

Explanation:
The interface file can request a time range represented by a set of starting and ending RBAs, LRSNs, or date/time values, but not a combination of any of these, Each is mutually exclusive with the other.

User response:
Correct and retry.

ALAA230E   INTERFACE: Start RBA must be less than End RBA.

Explanation:
The requested Start RBA was not less than the requested End RBA. Check the validity and the existence of the values for both input parameters. If one of these input parameters is included, both must be included with ascending values.

User response:
Correct and retry.

ALAA231E   INTERFACE: Start LRSN must be less than End LRSN.

Explanation:
The requested Start LRSN was not less than the requested End LRSN. Check the validity and the existence of the values for both input parameters. If one of these input parameters is included, both must be included with ascending values.

User response:
Correct and retry.

ALAA232E   INTERFACE: Parm record size exceeded. Must decrease size.

Explanation:
The maximum allowable number of records in the parm file is 1637. This size was exceeded, so processing stops.

User response:
Reduce the number of objects to be filtered on and retry.

ALAA233E	INTERFACE: No REDO/UNDO SQL requested, but "SQL DSN" requested.

Explanation:
The interface file requested no SQL generation, but specified a data set name to contain generated SQL.

User response:
Either remove the "SQL DSN" specification, or request generation of either REDO or UNDO SQL.

ALAA300E	Multiple executions of the ISPF interface in one TSO session is not allowed.

Explanation:
The current user may already execute the ISPF interface through the split-screen mode. It cannot be entered multiple times within the same TSO session.

System action:
Panel processing halts and an error message is issued.

User response:
Press PF3 to return.

ALAA301E	Invalid value, must be "A", "J", or "P" only.

Explanation:
The filter type specified was incorrect. Filter types can only be "A" for Authid, "J" for job name, or "P" for plan name.

User response:
Correct the entry and retry.

ALAA302E	Filter type and name must both be specified, or neither.

Explanation:
Enter both a filter type and a filter name as specified on the panel. It is invalid to enter only a filter type or only a filter name without the other value.

User response:
Correct the entry and retry.

ALAA303E	SYSCOLUMNS entry not found for table: table

Explanation:
An SQLCODE = +100 was found immediately when querying for column data for the named table. The log record data cannot be parsed out without this information, so processing stops.

User response:
Examine your catalog for possible reasons for this condition (SYSIBM.SYSCOLUMNS catalog table where TBCREATOR/TBNAME equals the named table).

ALAA304I	No DB2 logs were read.

Explanation:
The general report will yield no report data because no logs were read. This can occur for many reasons. The use of SYSLGRNX may have eliminated all logs from processing. This cause of the message should be considered for your information only, and is not a problem. Other causes may indicate a problem which you might rectify. For example, if your BSDS options on the "setup" panel are set to use ARCHLOG2 only, and none were available on the subsystem, no logs can be processed. Other BSDS problems can also cause this warning.

User response:
Evaluate your settings and BSDS for the DB2 subsystem. Check the BSDS data as it relates to the time range requested.

ALAA305W	Referential integrity records detected. SQL has been modified.

Explanation:
Log record activity contained actions resulting from referential constraints. For REDO, all such records are ignored because it is implied that the originating action (that which caused the RI actions) will also be re-done, triggering again the RI actions. For UNDO, only the order of the SQL is changed. IMPORTANT: if all tables involved in the RI constraints were not present in your results (i.e., they were filtered out), the generated SQL can omit any such RI records as proper execution requires all such tables. This can lead to an empty SQLOUT DD data set, which contains all the SQL.

User response:
Examine the SQL and your filters closely before executing any such generated SQL.

ALAA306I	The entries were successfully imported to the Db2 table from the TVFILE: &V1

Explanation:
The IMPORT command was successfully executed. All table versioning entries were imported from the specified TVFILE data set to the ALA_TABLE_VERSIONING table.

User response:
None.

ALAA307I	The entries were successfully exported from the Db2 table to a file: &V1

Explanation:
The EXPORT command was successfully executed. All table versioning entries were exported from the ALA_TABLE_VERSIONING table to the specified TVFILE dataset.

User response:
None.

ALAA308I	Invalid table versioning entries were successfully deleted.

Explanation:
The ALA_TABLE_VERSIONING table was successfully cleared from the entries of the non-existent objects.

User response:
None.

ALAA309I	No invalid entries found for deletion.

Explanation:
No entries of the non-existent objects were found for deletion.

User response:
None.

ALAA310E	Maximum number of columns can be in range of 1-750 only.

Explanation:
The maximum number of columns can be in the range of 1-750 only. This value is the number of auditable table columns, not including header columns such as LAT_DBID.

User response:
Correct the entry and retry.

ALAA311E	Maximum length of columns can be in range of 1-255 only.

Explanation:
The maximum length of each auditable column can be in the range of 1-255 only. Each auditable column will be defined with this VARCHAR length.

User response:
Correct the entry and retry.

ALAA312E	Audit mode not yet defined. Set "Audit Mode = S" for specifications.

Explanation:
Audit mode was requested, but the parameters of audit mode have not yet been defined/specified.

User response:
Set "Audit Mode = S" prior to attempting to run in audit mode. This is a one-time event; future requests for audit mode will not need this specification, as the settings will be saved. Once the settings for audit

mode have been defined, you can then set "Audit Mode = Y" and proceed.

ALAA313E	Audit header column name included, but left blank is not valid.

Explanation:
An audit header column was requested for inclusion, but the column name was left as spaces. Either do not include the column, or make sure the column name is a valid, non-blank name.

User response:
Correct the input and retry.

ALAA314E	The length of the audit table would exceed 32714 bytes; invalid.

Explanation:
The audit table length is determined by multiplying the number of auditable columns by the length of those columns. The values you supplied for these fields generates a table length greater than the maximum allowable. You must also account for the length of the included header columns when determining the row size of the audit table. Lower these values as supplied and retry. Also remember that if the row size is greater than approximately 4K, the audit table must be created in a table space which will accommodate greater than 4K page sizes, up to the 32K maximum

User response:
Correct and retry.

ALAA315E	Audit mode does not permit source tables with target tables.

Explanation:
When "Audit mode = Y", target fields table owner and name must all be blank. Source table specifications are allowed, but not associated target tables. This is because audit mode implicitly uses the single audit mode table as the target table. The only valid target field value in audit mode is the SSID field, which will determine where the audit table resides (or where it will be created, if so requested).

User response:
Either blank out the target fields table owner and name values, or set "Audit mode = N" and retry.

ALAA316E	"Object filters = A" requires "Generate Details = Y".

Explanation:
When using advanced filters, the detail level report must be generated at the same time as the general report. This is because advanced filters must operate on both general and detail level data. The advanced filters specified will automatically be carried through to both levels of reporting.

User response:

Correct the input and retry.

ALAA317E   **Using current filters is mutually exclusive with saving filters.**

Explanation:
You can either use the current advanced filters that have been previously saved to the named file, or you edit and save a new set of advanced filters, but not both.

User response:
Set one of these flags to N and retry.

ALAA318E   **If using or saving filters within a file, file name cannot be blank.**

Explanation:
If you set either flag on this panel to Y to use advanced filters from a file, or save advanced filters to a file, a file name must therefore be provided.

User response:
Correct the input and retry.

ALAA319E   **Unable to OPEN file for DD ALAAFN for output.**

Explanation:
This represents an internal error and should not occur.

User response:
Contact IBM customer support.

ALAA320E   **Bad input sequence on column data filters: *record_id***

Explanation:
The identified record represents the beginning on an input record that is likely out of sequence. This should only occur if you have manually edited the DD CDF input records in the generated JCL. Use the identified record to pinpoint the location of the failing record and determine why it is considered out of sequence.

User response:
See the HELP for column data filters for correct input record sequences. Correct the input and retry.

ALAA321E   **Bad conditional value in DD CDF input: *conditional_value***

Explanation:
The identified conditional is incorrect. The conditional value may be incorrect, or it could be an "AND" condition as the last condition. An "AND" condition cannot logically be the last condition. The only acceptable values for conditionals are spaces, "A", "O", or one of those values preceded by left or right parentheses. For example, "))A" or "(O" are correct, but "()A" or "A((" or "()(" are all invalid.

User response:
Correct the input and retry.

ALAA322E   **Bad operator value in DD CDF input: *operator***

Explanation:
The identified operator is incorrect. See the HELP on column data filters for correct operator values.

User response:
Correct the input and retry.

ALAA323E   **Both INCLUDE (IC) and EXCLUDE (XC) operators found for table.**

Explanation:
For any given table, there can only be INCLUDE or EXCLUDE type column specifications, not both. They are mutually exclusive. Either specify all the columns to include, or specify all the columns to exclude, but do not mix these operators. .

User response:
Correct the DD CDF input and retry

ALAA324I   **Previous filters have been saved; add more or exit**

Explanation:
The filters on the previous screen have been saved.

User response:
You can add more filters or hit PF3 to exit and use the saved filters, or type CANCEL to exit and bypass usage of any previously saved filters.

ALAA325E   **SQL stmt not valid due to: SET clause could not be generated.**

Explanation:
The SET clause could not be generated, and therefore no SQL was produced. "IC" or "XC" has been specified). In this case, an UPDATE may have occurred but none of the included columns were changed, and therefore no SET clause can be generated. In effect, the image of the table you have defined has not been modified, and therefore no update for that image is possible.

User response:
This is an internal error. Contact IBM customer support.

ALAA326E   **"Advanced filters" mutually exclusive with object list/IDs.**

Explanation:
Specifying Y to "Advanced filters" means all objects (table space/table) will be specified there only. No other object lists or IDs are allowed when using this option. Either set this field to N, or set object IDs and object lists flags to N.

User response:
Correct the input and retry.

ALAA327E   **Exclude flag can be Y (yes) or N (no) or spaces (no).**

Explanation:
The exclude flag can be left blank, set to Y to exclude the object, or set to N, which is the same as being left blank.

User response:
Correct the input and retry.

ALAA328E   **All columns excluded for table:** *table*

Explanation:
Column data filters excluded all columns of a single table. This is not allowed, as no data can be generated for that table.

User response:
Correct the exclude filters to leave at least one column of the table included.

ALAA329I   **Previous column orderings have been saved; add more or exit**

Explanation:
The column orderings on the previous screen have been saved. You can add more, or press PF3 to exit and use the saved orderings. Type CANCEL to save and bypass usage of any previously saved orderings.

User response:
No action is required.

ALAA330E   **Unable to OPEN file for DD ALAAFN for input.**

Explanation:
This represents an internal error and should not occur.

User response:
Contact IBM customer support.

ALAA331W   **Hitting PF3 now will delete some/all data in the file.**

Explanation:
This warning indicates that you are using advanced filters with the file option, but have not processed all (or any) of the previously saved data in that file. This may be your intended action, and if so, hit PF3 again. If not, "CANCEL" out of the action, or continue processing the data. This message will appear anytime you are using the file option and have not hit "ENTER" for every pre-filled panel. The end of pre-filled panels will display as an empty panel. Hitting PF3 prior to getting the first empty panel will always result in this warning message because you risk losing some or all of the data in the file.

User response:
No response required.

ALAA332E   **Log range values date-time, RBA, and LRSN are mutually exclusive.**

Explanation:
Log range values can be specified as date-time, RBAs, or LRSNs, but not a combination of any of these. Each is mutually exclusive with the other.

User response

Specify a single set of log range start/end values and retry.

Note: The default format value for start/end date-time fields are considered to be equivalent to spaces internally, so there is no need to blank out these fields if using RBAs or LRSNs.

ALAA333E   **At least one set of log range values must be provided.**

Explanation:
Log range values can be specified as date-time, RBAs, or LRSNs, and one of these range values must be set. It is invalid to omit all of these ranges, and it is invalid to provide more than one as well.

User response

Specify a single set of start/end values and retry.

Note: The default format value for start/end date-time fields are considered to be equivalent to spaces internally, so if date-time ranges are desired, you must type over the default field formats with valid dates and times.

ALAA334E   **Continuous mode is available only with date-time log ranges.**

Explanation:
A continuous mode file was specified, but either RBAs or LRSNs were provided as the log range. Only date-time specifications are allowed with continuous mode.

User response:
Correct the input and retry.

ALAA335E   **Conditional value can be "A" (AND) or "O" (OR) only.**

Explanation:
The conditional value specified was neither "A" nor "O". These are the only allowable values.

User response:
Correct the input and retry.

ALAA336E   **OPEN failure on DD MODEFILE.**

Explanation:
The OPEN failed for DD MODEFILE. Verify its existence in the JCL, and for correct attributes.

User response:

If all looks correct, contact IBM customer support.

ALAA337E	No LOB column data found. Unavailable data or null.

Explanation:
A request was made to materialize LOB column data, but based on the input, no LOB column data could be found. This might be caused by bad input that does not match actual detail report output. Other causes could be that the LOB data is either null (the detail report shows locator value with high order bit on (X"80")), or not available (if the LOB data did not change on update).

User response:
Verify the correctness of your input fields. If the action is an update, verify that the LOB data changed (i.e., the LOB column name has a trailing "#" in the detail report). Verify that the column is not null and has a proper record identifier.

ALAA338E	Requested End date/time less than start date/time in CMODE.

Explanation:
When using continuous mode (CMODE), you requested an end date and time that was less than the stored start date and time in the CMODE file. This will result in no activity, and is considered an error.

User response:
Either omit the end date and time from the general report panel, or do not use CMODE. Correct the input and retry.

ALAA339E	Open failure for MODERPT output file

Explanation:
An open request for the specified data set failed.

User response:
Verify the existence of the DD name in the JCL.

ALAA340E	Filter file name must exist if usage is not = N.

Explanation:
The filter file field was left blank, but the filter file usage field was not = N. This is an invalid combination. The filter file field identifies the name of the file used in filter file saving, and must not be left blank unless the filter file usage field is set to N.

User response:
Either change the usage field to N, or specify a filter file name.

ALAA341I	IBM* Rocket** Licensed materials - Property of IBM 5655-E56 (c) Copyright IBM Corp. 2001, 2003 All Rights Reserved. (c) Copyright Rocket Software, Inc.

2001 - 2003 All Rights Reserved. *Trademark of International Business Machines **Trademark of Rocket Software, Inc.

Explanation:
Informational message.

User response:
No response required.

ALAA342E	Fastpath option not allowed with Object filters-Advanced.

Explanation:
The fastpath method of a combined general and detail report is mutually exclusive with advanced object filters ("Object Filters = A"). This is due to the nature of fastpath processing, which will likely result in non-full row images, and therefore, column level data is unobtainable.

User response:
Either set this field to a value other than "P", or change your "Object Filters" setting to something other than "A".

ALAA343E	"Uniq" value of LUWID filter must be exactly 12 characters.

Explanation:
The "Uniq" value (uniqueness value) of the LUWID must be entered as a 12 character hexadecimal display value, such as B91FF0CA0034. No characters can be omitted.

User response:
Correct the input and retry.

ALAA344E	Commit value of LUWID filter must be exactly 4 characters.

Explanation:
The Commit value (commit count) of the LUWID must be entered as a 4 character hexadecimal display value, such as 000A. No characters can be omitted.

User response:
Correct the input and retry.

ALAA345E	Seclabels found during FASTPATH mode. DBID:*dbid* PSID:*psid* OBID:*obid*

Explanation:
The FASTPATH method should not be used when tables that have security labels are being processed. This is due to the nature of FASTPATH processing, which will likely result in non-full row images, and therefore, the security label is unobtainable. The activity on this table cannot be reported. If **Continue processing after warnings** is set to Yes, then the tables with security labels are skipped; otherwise, processing halts.

User response:
Either change the **Log reading mode** to a value other than **P**, or change your filters so that tables with security labels are not included.

ALAA346W   Row(s) removed during MLS PROCESSING DBID:*dbid* PSID:*psid* OBID:*obid*

Explanation:
Multi-level security is in place and prevented row(s) from being processed. Batch processing is complete. If **Continue processing after warnings** is set to Yes, then the rows will be removed and processing continues; otherwise, processing halts.

User response:
No action is required.

ALAA347E   RBA log range request in data sharing system prohibited.

Explanation:
The general report run specified RBAs as the log range type, but it was determine the requested subsystem is part of a data sharing group. This combination is invalid. Only date/time or LRSN log ranges are allowable in a data sharing system.

User response:
Change the log range input as necessary.

ALAA348E   Filter file is PDS, but no member specified.

Explanation:
You specified a PDS (library) to be used as the filter file, but did not provide a member name. This is not valid.

User response:
Either specify a sequential file, or a PDS with a member name.

ALAA349E   Filter file usage is "Edit" or "Use", but file/member does not exist.

Explanation:
All references to "file/member" refer to either a sequential file name or a member within a partitioned data set (PDS or library). You specified a usage type of "Edit" or "Use", but no filter file/member was found. "Edit" infers that you are editing a previously saved file/member. "Use" infers that you are using a previously saved file/member. Either of these options requires the file/member to pre-exist and have saved filters in it.

User response:
You must first save filter data off to the named file/member before you can edit or use it. See help on filter file usage.

ALAA350E   Bad page listing at cursor position.

Explanation:
The page filter as entered was invalid. The format described in the HELP for the panel input must be followed exactly. Pages can be specified either as single pages in 8 displayable hexadecimal numbers, or as a range of pages only. For example, 00000236 is a valid page; 00005001-00005003 is a valid range of pages. 00005 is an invalid page. 100-2225 is an invalid range. All pages and ranges must be delimited by commas. Spaces or any other characters not listed are considered to end a line of input. Pages and ranges cannot span 2 lines of input.

User response:
Correct the input and retry.

ALAA351E   Page filter IDs have no matching object IDs filter.

Explanation:
The page filter is associated with a DBID/PSID/OBID. However, when attempting to match this set of IDs to an object filter set of IDs, no match was found. This could only happen if you edited the filter file incorrectly, as the panel-driven input would not allow such a mismatch. Page filters IDs are not permitted as standalone filters; they must be associated with a defined object ID filter. In other words, 2 object ID filter lines must be present to request page level filters: one for the object itself (dbid,psid,obid), and the second for the associated page level filters (dbid,psid,obid=pages).

User response:
Correct the input and retry, or use only the panel-driven process.

ALAA352E   General SORT failure...see SYSOUT output for step.

Explanation:
A program invoked SORT failed. The SYSOUT output for the failing job step will contain needed diagnostics.

User response:
Attempt to resolve and retry.

ALAA353E   CONCURRENT image copies not supported.

Explanation:
An image copy was found and determined needed, but it is a CONCURRENT image copy, which is not supported (STYPE="C" from SYSIBM.SYSCOPY). Processing stops.

User response:
Attempt to resolve and retry.

ALAA354E   Input invalid for field.

Explanation:
A field value was entered that is not valid.

User response:
Enter only one of listed acceptable field values.

ALAA355E	Quiet time report invalid with detail report.

Explanation:
A quiet time report cannot be run with a detail report. The quiet time report does not generate the proper input for a detail report.

User response:
Either set the "Generate Details" flag to N, or change your output mode.

ALAA356E	Quiet time report must have start date/time input only.

Explanation:
A quiet time report cannot be run with any log range values except start date and time.

User response:
Correct the input and retry.

ALAA357E	Quiet time threshold value bad.

Explanation
The specified value is incorrect. The quiet time threshold must be in the format specified, with the following restrictions:

- HH >= 00 and HH < 24

- MM >= 00 and MM <= 59

- SS >= 00 and SS <= 59

User response:
Correct and retry.

ALAA358E	ALTER COLUMN DATA TYPE found for object ID: *dbidpsidobid*

Explanation:
An unsupportable condition was found for the object (DBID PSID OBID). DB2 provides very limited support for tracking column data type alterations, and therefore most attempts to materialize full row images across such an alteration cannot currently be provided. This product can only process such an altered table if the following conditions are true: a REORG has been done after the last alter, and no needed log activity occurred prior to that REORG.

User response:
If possible, remove the identified table from the result set and rerun your general report creating new input for this detail report without the named table.

ALAA359E	Filter file member specified, but dataset is not a PDS.

Explanation:

A filter file was specified for use, but it included a member name and is not a PDS (library). Either remove the member name or specify a different file entirely.

User response:
Correct the input and retry.

ALAA360E	Quiet time report invalid with LOAD.

Explanation:
A quiet time report cannot be run with a LOAD. The quiet time report does not generate the proper input for a LOAD.

User response:
Either set the **LOAD options** flag to N, or change your output mode.

ALAA361E	Quiet time report invalid with Collect object data.

Explanation:
A quiet time report cannot be run with Collect object data. It does not generate the proper input for application-level relationship discovery.

User response:
Either set the **Collect object data** flag to **N**, or change your output mode.

ALAA362E	Field must be empty when using any command.

Explanation:
Panel command requires all line command fields to be empty.

User response:
Correct the input and retry.

ALAA363E	Too many line commands specified

Explanation:
More than one line command was found. Only one line command can be entered at a time.

User response:
Correct the input and retry.

ALAA400E	Storage obtain failure for MMTAB.

Explanation:
A request for storage failed for area MMTAB.

User response:
If possible, increase your region size, otherwise contact IBM customer support.

ALAA401E	Storage obtain failure for MMSP.

Explanation:
A request for storage failed for area MMSP.

User response:
If possible, increase your region size, otherwise contact IBM customer support.

ALAA402E   Storage obtain failure for MMIB.

Explanation:
A request for storage failed for area MMIB.

User response:
If possible, increase your region size, otherwise contact IBM customer support.

ALAA403E   Storage obtain failure for MRE.

Explanation:
A request for storage failed for area MRE.

User response:
If possible, increase your region size, otherwise contact IBM customer support.

ALAA404E   Storage obtain failure for MM buffers.

Explanation:
A request for storage failed for area MM buffers.

User response:
If possible, increase your region size, otherwise contact IBM customer support.

ALAA405W   No SQL generated, may be due to filters/rollbacks/other options.

Explanation:
No SQL was generated for this run. This can occur because of your filters (for example, advanced filters at the column-level filtered out all rows), rollbacks, only catalog tables present in your output (no SQL is generated for catalog tables), or the fastpath option did not materialize full rows and therefore no SQL could be generated.

User response:
Check for these possibilities as well as any possible warnings that may have been generated during the run.

ALAA406E   Bad input, field must start with value between A-Z.

Explanation:
The archive fields for table creator, name, and data set suffix must all start with characters from A-Z, with no other special characters.

User response:
Correct your input and retry.

ALAA407E   Archive fields have improper values; must specify archive values.

Explanation:
The archive fields did not have proper values, but you requested use of the archive process.

User response:

Before you can use the process of archived output you must first specify the proper values for archiving. This requires setting the "Archive output" field on the detail report panel to "S", and setting the proper fields therein. Once you have completed this step, then you can both create and subsequently use archived output.

ALAA408E   Catalog data for LOB object not found.

Explanation:
You requested data for a LOB column, but either the base table IDs are incorrectly specified, or the LOB column name is not valid. This can mean that the LOB column name is typed incorrectly, or is a valid column of the table, but is not a LOB column.

User response:
If materializing LOB column data, verify that the input is correct. If this message occurs during SQL generation, contact the vendor.

ALAA409E   ARCHDD DD file could not be opened.

Explanation:
The data set associated with DD ARCHDD could not be opened.

User response:
Verify the existence of this DD in your JCL and existing SYSIN-type records.

ALAA410E   Current data only = N requires start/end date-times.

Explanation:
You must request either current data only, or a date/time range. The current values specify neither.

User response:
Either set "Current data only" to Y, or specify a valid date/time range.

ALAA411E   MASS DELETE record found during log-backward processing: *value***.**

Explanation:
A MASS DELETE matching record was found during log-backward processing. This condition can only be determined while reading the logs (if the MASS DELETE record was not part of the general report output). Processing must stop, and only a log-forward approach is allowed. Whenever running against segmented tables, there is a possibility of mass delete records (that is, DELETE FROM table with no WHERE clause). The value shown is the DBID/PSID/OBID identifying the table where the MASS DELETE occurred.

User response:
If this possibility exists at your site, always run in log-forward mode to prevent unnecessary job stoppage such as this. DATA CAPTURE CHANGES will also

prevent mass delete type records, but adds overhead to DB2 logging for the table.

ALAA412E **Partition or page filters not allowed w/LOAD REPLACE log action.**

Explanation:
A "Load Replace" type utility was found during the general report run. The action-type for this record is "P", indicating "Load Replace". This type of action is not currently permitted with filters below the table level. That is, partition or page level filters are not permitted. Use of only table space or table level filters is permitted when processing a "Load Replace" type log record. These filters are not allowed in either the general or detail report runs when LOAD REPLACE actions exist.

User response:
Change your filters (remove any partition or page level filters for the table space identified in the "P" action-type record in your general report output) and retry.

ALAA413E **Archive specifications must be set prior to usage.**

Explanation:
You requested use of archived output, but have not yet set up your archive specifications.

User response:
Before you can use this option, you must run at least one detail report, and in so doing, specify your archive settings. Go to the detail report panel and set this field as such: "Archive output.... => S". This will allow you to specify you archive settings, and is a one-time setting per archive table (typically, only one archive table need be used unless otherwise requested). Once those values are set and at least one detail report has been run with these settings, you can then continue with this option.

ALAA414E **RBA or LRSN value specified is not proper hexadecimal value.**

Explanation:
You requested use of either RBAs or LRSNs, but the value(s) entered are not valid hexadecimal specifications. An RBA or LRSN must only contain hexadecimal values, which can be 0-9 and A-F only. For example, a value of 00012FE560BBC is valid, but a value of 0012WXXX0345 is not.

User response:
Correct your input and retry.

ALAA415E **DB2 log with needed starting logpoint unavailable. Processing stops.**

Explanation:

The general or detail report run required a starting logpoint (RBA/LRSN) which was lower than any available DB2 log logpoint in the BSDS. Therefore, processing must stop. This would likely only occur in situations where the date or time specified was so old that the relevant archive logs are no longer in the BSDS. It could also occur on test systems where archive logs are not available after a very short time, or at all. Log Analysis Tool cannot function without these needed resources (logs).

User response:
Correct your input using a more recent date and time and retry.

ALAA419I **Char conversions ended: Ret cde: *return code*, Res cde: *reason code* CCSID/SBCS CCSID/MBCS CCSID**

Explanation:
This message shows the status of the character conversion process. The "Support For Unicode" product was called to do character conversion from Unicode or ASCII to EBCDIC, but it failed with the shown return code and reason code. The last three variables displayed are the source CCSID, the target SBCS CCSID, and the target MBCS CCSID used in the conversion call.

User response:
Verify the correctness of these CCSIDs and that these CCSIDs are defined to the "Support For Unicode" product. For more information, on the IBM Db2 Log Analysis Tool for z/OS Main Menu, choose option 12 - Tutorial, and select the tutorial topic "Handling non-EBCDIC DB2 data."

ALAA420E **Invalid starting/ending RBA/LRSN value(s) provided.**

Explanation
You must request one and only one of the following values:

- A starting RBA
- A starting LRSN
- An ending RBA
- An ending LRSN

You cannot provide more than one of these values (they are all mutually exclusive).

User response:
Correct the input and retry.

ALAA421E **No input dataset names provided.**

Explanation:
You must request at least one input data set name. This input data set is required as the alternate source

of input for the product. Without this data set name, the expert option cannot be processed.

User response:
Read the help and tutorial on this subject prior to any attempts at utilizing this option.

ALAA422E	Invalid input dataset type; must be "T", "I", "O" only.

Explanation:
You requested an invalid data set type. The value must be one of those listed in the error message. A space is invalid here.

User response:
Correct the input and retry.

ALAA423E	Bad dataset line; dataset type/file sequence number invalid.

Explanation:
You requested a file sequence number with a data set type of either "T" or "I". File sequence numbers can only be specified with type "O". Any other data set type value must specify a value here of spaces or "00".

User response:
Correct the input and retry.

ALAA424E	Bad dataset line; file sequence number must be 0-99 only.

Explanation:
You requested an invalid file sequence number. The value can only be numeric between 0 and 99.

User response:
Correct the input and retry.

ALAA425E	Bad dataset line; dataset type/ volume serial number invalid.

Explanation:
You requested volume serial number(s), but the data set type was not "O". The data set type can only be "O" when volume serial number(s) are requested.

User response:
Correct the input and retry.

ALAA426E	Bad dataset line; invalid volume serial specification.

Explanation:
You requested an invalid set of volume serial numbers. Each volume specified must be exactly 6 characters long, and separated only by commas. For example, "XT0123,XT0234,SPD001" is valid. Invalid specifications might be: "XT0,SPD001" or "XT0123XT0234" or "XT0123 XT0234 SPD001".

User response:
Correct the input and retry.

ALAA427E	XOPTS DD not found, but expert options requested.

Explanation:
You requested Expert Options=Y from the detail report panel, but the actual run of the detail report could not find the required XOPTS DD statement. This DD is built by the front-end process, and it may be that you incorrectly edited the generated JCL.

User response:
Re-create the detail report JCL using Expert Options=Y and verify the existence of the DD XOPTS, then retry the detail report job.

ALAA428E	Mismatch found: log-backward mode with log-forward RBA/LRSN values.

Explanation:
Using expert options, the front-end determined a log-backward method was required. However, the actual detail report run found log-forward RBA/LRSN value in the XOPTS DD (i.e., "Start RBA=" or "Start LRSN="). This mismatch cannot be resolved by the program. The detail report JCL may have been improperly edited.

User response:
Re-create the detail report JCL, and retry.

ALAA429E	XOPTS DD is missing starting/ ending RBA/LRSN record.

Explanation:
Using expert options, the XOPTS DD was found to be missing the required starting or ending RBA/ LRSN record. The detail report JCL may have been improperly edited.

User response:
Re-create the detail report JCL verifying the XOPTS DD statement contains a single "Start..." or "End..." record indicating the RBA or LRSN values needed, then retry.

ALAA430E	Expert options specified, but request is improper.

Explanation:
You requested Expert options=Y from the detail report panel, but the execution of that detail report was invalid. This can occur if multiple table spaces are in your detail report filter set. It can also occur if partitions are in your filter set, but not all partitioned data sets were provided as input.

User response:
Read the tutorial and help provided on expert options. All needed input must come from the user in this mode; the product performs no other internal operations to resolve needed inputs.

ALAA431E	Expert options dataset name not found: dsname

Explanation:
The specified input data set name was not found in the DB2 catalog table SYSIBM.SYSCOPY. A data set type of "I" was indicated for this expert option input, but it could not be found. "I" tells the product to resolve all information about this image copy via SYSIBM.SYSCOPY.

User response:
Either change the data set type to "O" and supply all needed information manually from the "Expert options" panel, or verify correctness of the specified data set name. Please read the tutorial on expert options for more information.

ALAA432E	Invalid ending RBA/LRSN using expert options.

Explanation:
The value specified as the ending RBA/LRSN during expert option specifications was determined to be invalid during execution of the detail report run. This ending value (RBA or LRSN) cannot represent a point-in-time earlier than the ending point-in-time of the general report timeframe. For example, assume your timeframe on the general report run was 08:00-08:30, which read through RBA value 0006ABCD0000. Therefore, you could not specify an ending RBA value less than 0006ABCD0000. The ending RBA has to be greater than 0006ABCD0000.

User response:
Please read the tutorial on expert options and retry with an appropriate RBA/LRSN value.

ALAA433E	Bad dataset line; invalid "R" value, Y/N or space only.

Explanation:
The value specified for the "R" field (REORG/INLINE copy indicator) is bad. Valid values are Y (yes, it is an inline image copy taken during a REORG), N (it is not an inline image copy taken during a REORG), or space (treated same as N).

User response:
Correct the input and retry.

ALAA434E	Bad dataset line; invalid "R" value with dataset type.

Explanation:
The value specified for the "R" field (REORG/INLINE copy indicator) is bad because the data set type is not "O". Only a data set type of "O" can specify this value. It must be a space (or N) if data set type is not "O".

User response:
Correct the input and retry.

ALAA435E	Bad dataset line; invalid device specification with type.

Explanation:

The data set type must be "O" if a device is specified. Otherwise, the device field must be spaces.

User response:
Correct the input and retry.

ALAA436E	Multiple tablespaces found during expert option run; invalid.

Explanation:
You requested expert options during a detail report run, but also included multiple table spaces in the result set (i.e., there were multiple table spaces reported on in the general report output). This is not allowed. Expert options can process on a single table space only.

User response:
Either use the detail report panel to filter by a single table space, or rerun your general report to filter by a single table space. Please read the tutorial on expert options before attempting to use this mode of operation.

ALAA437E	Misc flags can only be spaces, "X", "M, "P", "H", or "L" values.

Explanation:
The possible values for miscellaneous flags are spaces, "X" (bypass SYSLGRNX), "P" (ignore partial recovery logpoints), or "H" (allow high speed option), "L" (allow LOAD LOG YES with high speed option), or "M" (bypass SYSLGRNX only when MODIFY RECOVERY utility removes entries.

User response:
Correct your input and retry.

ALAA438E	Extract file (DD EXTFILE or DD EXTFILEB) is missing; processing stops.

Explanation:
The DD named EXTFILE was not found in your report JCL. This file must be present. If you edited the product-generated JCL, you may have omitted this DD.

User response:
Re-generate the JCL using the panels and retry. If the file is still missing from the JCL, contact your product administrator.

ALAA439E	Temp extract file (DD TEMPEXTF) is missing; processing stops.

Explanation:
The DD named TEMPEXTF was not found in your report JCL. This file must be present. If you edited the product-generated JCL, you may have omitted this DD.

User response:
Re-generate the JCL using the panels and retry. If the file is still missing from the JCL, contact your product administrator.

ALAA440E   **ROWDATA validation error: value**

Explanation

An error was detected while validating the results of full row materialization. The value listed in the message identifies the first 19 bytes of the record in the ROWDATA DD file where the error was detected. This error can occur for the following reasons:

for an INSERT
> no post-insert row image found/pre-insert row image found

for a DELETE
> no pre-delete row image found/post-delete row image found

for an UPDATE
> no post-update row image found/no pre-update row image found

for an UPDATE
> no change in pre-update and post-update row images found

User response:
Contact your product administrator to report the problem.

ALAA441E   **Standard header not yet defined. Set "Include Header=S" one time.**

Explanation:
You requested inclusion of a header, but have not yet specified your header columns.

User response:
Please set the "Include Header" field to "S" for a minimal one-time setting of the desired header columns. Thereafter, any specification of "Include Header=Y" will utilize your preset values.

ALAA442W   **No keys found for table:** *table*

Explanation:
You requested current row images on the identified table (DBID,PSID,OBID) but no key columns were found for it. Only uniquely keyed tables are candidates for this type of current row materialization. An index on expression cannot be used as a unique key here. This identified table will not have current row images displayed based on its key. That is, if a log-backward method is being utilized, the current row images based on page/rid values will be kept. If a log-forward method is being utilized, no current row images are available for the table.

User response:
None.

ALAA443E   **Current row image process found LOB column in table: table**

Explanation:

You requested current row images on the identified table (DBID,PSID,OBID) but a LOB column exists on that table. This is not valid. Therefore, if a log-backward method is being utilized, the current row images based on page/rid values will be kept. If a log-forward method is being utilized, no current row images are available for the table.

User response:
You must specify a different table.

ALAA444E   **Cannot get current rows due to CCSID translation: table**

Explanation:
You requested current row images on the identified table (DBID,PSID,OBID) but it required CCSID translation which failed.

User response:
See job log output for additional diagnostics (ALAA156I messages). This table will not have current row images based on key values. If a log-forward method is being utilized , no current row images are available for the table.

ALAA445E   **Job Identifier invalid; not previously used or currently in-use.**

Explanation:
You specified a job identifier that was not previously used in a general report, or was used, but is currently in-use. For the first case, you must run a general report prior to running any detail report, and the detail report must reflect a job identifier that was previously specified for a general report run. For the second case, you are trying to use a job identifier that is currently in-use. This means a job is running using that same job identifier, and you cannot use it now. The job identifier distinguishes files between different runs of different jobs, and these files must exist and not be in-use while generating report JCL.

User response:
Correct your input.

ALAA446E   **Filter value too long for filter type; see HELP for panel.**

Explanation:
You specified a filter value which is too long for the filter type For example, you specified a filter type = 'A' (authid), but the filter value provided is greater than 8 characters. See the HELP for the panel on allowable values for each type.

User response:
Correct the input and retry.

ALAA447E   **ROWDATA update validation error: $V1 ; $V2 : $V3 ; $V4 : &V5**

Explanation:

An error was detected while validating during the merge phase. An update to a record was needed at a certain offset (>0) into the row, but there was no current data on that row. This is a fatal error. The values listed in the message identify the first 33 bytes of the record in the ROWDATA DD file where the error was detected, the first and the last log points of the row.

User response:
Contact IBM Customer Support to report the problem. The support team will need to see the contents of the PLOGDS DD file.

ALAA450E	Input validation error: value

Explanation:
An error was detected while processing input for the LOAD details into DB2 process. The value listed in the message identifies the start of the failing record. This could only happen if you incorrectly edited JCL generated by the ISPF front-end process.

User response:
Re-generate the JCL using the panels supplied and verify the correctness of this input record. If the product generates a bad input record from the front-end process, contact IBM Customer Support.

ALAA451E	SYSLGRNX storage failure

Explanation:
A failure occurred during storage obtains for the SYSLGRNX reading process. Too many table spaces were provided as input and not enough region is available for the request.

User response:
Either bypass SYSLGRNX processing, or increase your region size. If region is currently at the maximum setting for your batch jobs, you must either reduce the number of table spaces requested or bypass SYSLGRNX utilization.

ALAA452E	Invalid parallelism/concurrency value; must be between 01-16 only

Explanation:
The value entered for this field is incorrect. Enter only a value between 1 and 16.

User response:
Correct the input and retry.

ALAA453E	Quiet time report mutually exclusive with all others.

Explanation:
When requesting a quiet time report, only a value of "Q" can be entered for the "Output flags" field. This setting is mutually exclusive with all other output types.

User response:
Either request only "Q" for a quiet time report, or remove the "Q" from the list. Correct the input and retry.

ALAA454E	Quiet time report preceded detail report; invalid.

Explanation:
A quiet time report does not generate the appropriate input necessary to run a detail report. The MODEFILE indicates the last general report run was a quiet time report, and therefore, it is not possible to now run a detail report.

User response:
You must run a general report without the quiet time option prior to running a detail report.

ALAA455E	Bypassing all reports mutually exclusive with requesting reports.

Explanation:
You requested a bypass of all report types ("Output flags = B"), but added at least one other specification. This is not allowed. A value of "B" must be the only value in this field, or it must be removed to specify other report output types.

User response:
Correct the input and retry.

ALAA456E	Filters found to be in error; probable user-editing error.

Explanation:
The end-of-file was found while reading your filter list, but a filter group was still in-progress. Each group of filters must end with either an INC/EXC card, or a COND card. These are referred to as "terminating records" and must be present as the last record in every filter group.

User response:
See tutorial on filter file handling. Check your filters for such an omission and correct and retry. This could occur on object name filters, advanced filters, or miscellaneous filters only.

ALAA457E	Filter record invalid within grouping: record

Explanation:
The identified record is in error, and probably due to user-edit error. This would occur most likely from duplication of a record within a filter group or if you omitted a required record from the group. Regards the latter, if the terminating record in the group is found before all other required records, that terminating record will be displayed here (for example, COND= and INC/EXC= are terminating records). There are no optional records in any group; they must all be present, and in the specified order.

User response:
Verify your filters and read up on filter file editing in the tutorial. Correct your input and retry.

| ALAA458E | ATTACH failure during parallelism for log reads. |

Explanation:
An attempt was made to attach a subtask for parallelism during DB2 log reading. The ATTACH failed.

User response:
Check the job log for more error messages from the operating system. It may be that too many subtasks were requested so check your "Log read parallelism maximum" value on the "Defaults" panel. Try lowering the value and retry. If it is not a matter of excessive sub tasking, contact IBM Customer Support.

| ALAA459E | OPEN failure on a DD during subtask (parallelism) processing. |

Explanation:
An attempt was made to OPEN a DD during subtask processing for DB2 log reading parallelism. This OPEN failed.

User response:
Check the job log for system message (IEC130I) indicating a missing DD statement. It is most likely you have incorrectly edited the JCL. Re-generate the JCL from the ISPF front- end panels and retry. If the problem persists, contact IBM Customer Support.

| ALAA460E | Fatal error during subtask processing. Check job log for messages. |

Explanation:
An error was detected by the main calling task invoking subtasks for DB2 log read parallelism.

User response:
Check your job log for other system messages or a dump pertaining to this failure. Resolve the failure if possible and retry, or contact IBM Customer Support.

| ALAA461E | SORTIN DD statement missing. |

Explanation:
The DD for the SORTIN statement is missing from the JCL step. This could likely only happen if the JCL has been incorrectly edited or improper JCL used across different releases of the product.

User response:
Re-generate your JCL from the ISPF front-end.

| ALAA462E | SORTOUT DD statement missing. |

Explanation:
The DD for the SORTOUT statement is missing from the JCL step. This could likely only happen if the JCL has been incorrectly edited or improper JCL used across different releases of the product.

User response:
Re-generate your JCL from the ISPF front-end.

| ALAA463I | There is no data to report. |

Explanation:
The general report found no data. This could occur for many reasons, such as due to filters, time specifications, etc. This is an informational message only.

User response:
None required.

| ALAA464E | LOAD table limit exceeded, storage exhausted. |

Explanation:
You have exceeded the maximum limit (approximately 600 tables) allowable in a single pass.

User response:
Lower the number of tables to process and retry.

| ALAA465E | Restart value=M requires a restart SQL number. |

Explanation:
You requested to restart on a particular SQL number, but left the "Restart SQL number" field blank.

User response:
Either change the "Restart" field to a value other than "M", or include the restart SQL number.

| ALAA466E | Restart name field required unless restart value = "I". |

Explanation:
You requested a restart value of something other than "I", and this requires a restart name value.

User response:
Either change the "Restart" field to "I", or include a restart name value.

| ALAA467E | ON SQLCODEs field value incorrect. |

Explanation:
You requested certain actions for certain SQL codes, but your request is invalid.

User response:
See the HELP on the panel for a description of valid values for this field. These formats must be followed exactly. For example, "+100,I;-511,I" is valid, but "100, I; -511, I" is not.

| ALAA468E | Restart SQL number is invalid. |

Explanation:
The value you specified for this field is invalid. Only numeric values are allowed, do not include any non-numeric values.

User response:

Correct your input and retry.

ALAA469E   **Invalid starting RBA/LRSN using expert options.**

Explanation:
The value specified as the starting RBA/LRSN during expert option specifications was determined to be invalid during execution of the detail report run. This starting value (RBA or LRSN) cannot represent a point-in-time later than the starting point-in-time of the general report timeframe. For example, assume your timeframe on the general report run was 08:00-08:30, which returned a low RBA of 0006ABCD0000. Therefore, you could not specify a starting RBA value more than 0006ABCD0000. The start RBA has to be less than 0006ABCD0000.

User response:
Please read the tutorial on expert options and retry with an appropriate RBA/LRSN value.

ALAA470E   **Loadfile record too long; beyond DB2 32k pagesize limit.**

Explanation:
A loadfile record was encountered that was beyond the 32K pagesize limit imposed by DB2. You cannot create a row in your target table that will exceed this limit. If you are running against tables with row lengths close to or at this maximum size (32k) you must run with **Data Only=Y** set. No other combinations of settings will work. If you are running in Audit mode, you must reduce the size and/or number of your requested audited columns.

User response:
Correct your input and retry.

ALAA471E   **Showing uncommitted units of work not allowed with continuous mode.**

Explanation:
You cannot show uncommitted units of work in continuous mode. The purpose of continuous mode is to track uncommitted units of work across different runs of the product, and that is in direct contrast with showing uncommitted units of work in a single run.

User response:
Either set "Show uncommitted=N" or set the "Continuous mode file name" equal to spaces. Correct your input and retry.

ALAA472E   **Impact report requires date/time log range only.**

Explanation:
You cannot request an impact report ("Output Mode" includes value of "I") without also specifying a log range based on start and end date/time. Log ranges

of start and end RBA or LRSN are not allowed with an impact report.

User response:
Correct your input and retry.

ALAA473E   **Impact report mutually exclusive with continuous mode processing.**

Explanation:
You cannot request an impact report ("Output Mode" includes value of "I") with continuous mode processing. The two are mutually exclusive.

User response:
Either remove the impact report option or the continuous mode file name and retry.

ALAA474E   **OPEN error for DD:** *ddname*.

Explanation:
OPEN failed for the identified DD name. Either the DD is missing from the JCL, or dynamic allocation for the DD failed, but was not properly identified by the product.

User response:
If the identified DD name was hardcoded in the failing JCL step, verify the correctness of the statement and data set name. If the DD name was not provided in the JCL, contact your product administrator to notify the vendor.

ALAA475E   **Incorrect value for volume serial number.**

Explanation
VSAM volume serial number must be one of the following only:

- all spaces meaning no volume serial number specifically requested
- a single volume serial number (6 characters; e.g., VSM101)
- up to 3 comma-separated volume serial numbers (e.g., VSM101,VSM204,VSR341)

Any other format is not acceptable.

User response:
Correct your input and retry.

ALAA476E   **MODE file does not have all needed records.**

Explanation:
The MODE file acts as the primary resource of communications between all programs within the product. This file is uniquely identified by the value of your "Job Identifier" field on the product panels.

User response:

Re-generating JCL via the product panels will modify this MODE file and therefore could prevent execution of previously saved-off JCL with the same "Job Identifier" value. The only solution here is to re-run the general report JCL again and then retry the detail report JCL. Do not generate any new JCL via the panels in between those two separate reports, unless you first modify the "Job Identifier" field.

ALAA477I   **Error occurred during error processing.**

Explanation:
Some problem occurred that prevented the normal processing of an error condition. This may be an expected or unexpected situation. For example, some errors can occur before enough information about the environment is available to properly handle message processing (expected situation). Other errors can occur because the environment has not been properly setup (unexpected situation). In either case, the true error message is displayed following this message.

User response:
Examine that message in the "Message Help Facility" to resolve the problem or contact the product administrator/vendor.

ALAA478E   **Specify Logs=Y" mutually exclusive with impact report.**

Explanation:
You requested an impact report ("Output Flags" includes value of "I"), and also requested "Specify Logs = Y". This is not a valid combination. An impact report requires all logs to the current time, and this would require use of active logs. Active logs are not allowed within the context of specifying logs and, therefore, these two options are mutually exclusive.

User response:
Remove one of these options from your selections and retry.

ALAA479E   **MODE file is in an invalid state; must re-run general report.**

Explanation:
The start of the detail report determined the mode file was not in the proper state. The mode file is used as a means of communication between the general and detail report runs. You must always precede a detail report run with a general report run. You must also not re-generate any general report JCL in between running the general and detail reports. Such a re-generation changes the contents of the mode file.

User response:
At this point, you must run the general report JCL before running the detail report (either for the first time or again, depending on the actions that caused the mode file problem).

ALAA480E   **Including LOB data is mutually exclusive with High Speed Mode.**

Explanation:
LOB data cannot be accessed in high speed mode. The overhead involved in processing LOB data prevents such efficiencies

User response:
Remove either the inclusion of LOB data, or remove "H" from your miscellaneous flags setting.

ALAA481E   **LOB log record of type DELETE/UPDATE invalid in log-backward mode.**

Explanation:
You forced a log-backward mode for detail reporting, but LOB data was found of type DELETE or UPDATE, and these types of LOB log records cannot be processed in log-backward mode.

User response:
Change to log-forward processing ("Log reading mode = F") and retry. If you set "Log reading mode= A", the product will automatically select log-forward mode.

ALAA482E   **Invalid number of hex digits found in hex filter.**

Explanation:
When using hexadecimal values in the advanced filters, there must be at least two digits. If there are more than two digits, the total number of digits must be even.

User response:
Correct the input and retry.

ALAA483E   **Invalid or unbalanced quotes found in hex filter.**

Explanation:
When using hexadecimal values in the advanced filters, the string entered must start with X followed by a single quote, valid hexadecimal digits, and end with a single quote. If you are attempting to use the LK or NL operator then put the % after the final single quote not inside the quotes.

User response:
Correct the input and retry.

ALAA484E   **Invalid hexadecimal character found in hex filter.**

Explanation:
When using hexadecimal values in the advanced filters, the string entered must start with X followed by a single quote, valid hexadecimal digits (012345678ABCDEF), and end with a single quote. If you are attempting to use the LK or NL operator then put the % after the final single quote not inside the quotes.

User response:
Correct the input and retry.

| ALAA485E | Invalid or unbalanced quotes found in filter value. |

Explanation:
When using column filters if you have a beginning quote you need to end the string with an ending quote.

User response:
Correct the input and retry.

| ALAA486E | Versioning data not available for table: *tablename*. |

Explanation:
This message indicates versioning information was required to format row data, but was not found in the DD TVFILE. Versioning information is required because a table is needed to correlate different row formats. The message indicates the table ID (DBID,PSID,OBID) and the version number missing is the last value.

User response:
See the tutorial section on "Table Versioning support" for more information on how to proceed.

| ALAA487E | Error occurred writing informational messages to file. |

Explanation:
This message indicates a fatal error occurred while trying to write out informational messages to the DD INFOM. This can be avoided by setting the "Send informational messages to file" field on the "User settings" panel to N.

User response:
Try to determine the cause of the failure by looking in the job log for any system-generated messages related to this failure and file DD INFOM. If unable to detect and correct the error, contact your product administrator to notify the vendor.

| ALAA488E | General report data=Y" must also have "Header=Y" setting. |

Explanation:
This message indicates you requested the general report data to be loaded only, but you also did not request header information. General report data is only header data (no row data), and therefore you must select "Header=Y" when requesting a general report load.

User response:
Correct your input and retry.

| ALAA489E | "General report data=Y" mutually exclusive with Audit Mode. |

Explanation:
This message indicates you requested the general report data to be loaded only, but you also requested

"Audit Mode = Y" (or S). Loading using general report output only cannot be part of audit mode processing. Audit mode is for column-data-level auditing only.

User response:
Set "Audit Mode = N" and retry.

| ALAA490E | Error occurred during DSPSERV CREATE services: *ret_code*, *rsn_code* |

Explanation:
An error was generated during DSPSERV CREATE services while processing LOB data. The variables in the message reflect the return code and reason code associated with the error.

User response:
Check the return and reason codes in the IBM manual "Authorized Assembler Services Reference" for macro DSPSERV. If resolution cannot be determined, contact your product administrator to notify the vendor.

| ALAA491E | Error occurred during ALASERV ADD services: *ret_code*, *rsn_code* |

Explanation:
An error was generated during ALESERV ADD services while processing LOB data. The variables in the message reflect the return code and reason code associated with the error.

User response:
Check the return and reason codes in the IBM manual "Authorized Assembler Services Reference" for macro ALESERV. If resolution cannot be determined, contact your product administrator to notify the vendor.

| ALAA492E | LOB file does not exist or is currently in-use. |

Explanation:
You requested a browse of the current LOB file, but it is either in-use, or does not yet exist.

User response:
If you just submitted the batch job to materialize LOB data, please wait for that job to end. In this case, the file is still in-use. If you did not materialize any LOB data yet for viewing, then you must first do so before you can browse the LOB data. In this case, the file does not yet exist.

| ALAA493E | Conversion services failed for LOB materialization. |

Explanation:
The LOB contained non-EBCDIC data, and conversion services to EBCDIC failed for the LOB.

User response:
See your job log (or INFOM DD) messages for more information on why the conversion process failed. The LOB cannot be materialized due to this failure.

ALAA494E   **No point-in-time conversion could be performed.**

Explanation:
A conversion from one type of logpoint to another type was requested, but no conversion could be performed. For example, you requested that a LRSN be converted to a RBA, but the returned RBA value was all zeros. This service is only available via the product interface mechanism.

User response:
Confirm that your input is valid and relative. The input must not be beyond the current limits of any accepted values. For example, if a date/time was provided that is beyond the current date/time, this will occur. Correct your input and retry.

ALAA495E   **Number of log records tracked excessive; storage exhausted.**

Explanation:
An attempt was made to store over 60 million log records. This is considered excessive, and no more storage will be obtained.

User response:
Reduce the number of tracked log records in a single run by narrowing your time range or reducing the number of filtered objects. If you did not provide any filters, do so now, as running against all objects in your subsystem will likely lead to such an error.

ALAA496E   **DDLFILE > 32k SQL limit after whitespace compression.**

Explanation:
The DDLFILE created by this product contains DDL to create the target table(s) for the LOAD process. After whitespace was compressed, this DDL is still greater than 32K, a DB2 limit.

User response:
Reduce the number of tables to create, or the number of columns if in audit mode.

ALAA497E   **Requested column is neither a LOB nor an XML data type.**

Explanation:
You requested materialization of either a LOB or XML column, but the column is not marked as either a LOB or XML column in the FRMTFILE.

User response:
Verify the name of the column you entered is actually a LOB or an XML data type. Also verify the record identifier is pointing to a table containing this expected LOB or XML data type. Correct either the column name or the record identifier value. If both appear correct, contact the vendor.

ALAA498E   **"General report data=Y" mutually exclusive with split LOAD files.**

Explanation:
This message indicates that you requested the general report data to be loaded only, but you also requested "Split load files=Y". Loading using general report output only cannot be split into multiple LOAD files.

User response:
Set "Split load file=N" and retry.

ALAA499E   **"General report data=Y" does not allow editing the column order.**

Explanation:
This message indicates that you requested the general report data to be loaded only, but you also requested "Edit column order=Y". Loading using general report output only does not allow the editing of column order.

User response:
Set "Edit column order=N" and retry.

ALAA500E   **Getmain failure...Ret code: *ret_code*, Length: *length*.**

Explanation:
An error was generated during GETMAIN services. The return code and length of the failing GETMAIN are displayed in the message.

User response:
Check the return code in the IBM manual "Authorized Assembler Services Reference" for macro GETMAIN. It is most likely that storage has been completely exhausted, and a smaller run is required (i.e., shorter timeframe and/or more filters to further limit the result set).

ALAA501E   **IARV64 failure...Ret code: *ret_code*, Res cde: *rsn_code*.**

Explanation:
An error was generated during IARV64 services. The return code and reason code of the failing IARV64 service are displayed in the message.

User response:
Check the return/reason code in the IBM manual "Authorized Assembler Services Reference" for macro IARV64. It may be that storage has been completely exhausted, and a smaller run is required (i.e., shorter timeframe and/or more filters to further limit the result set).

ALAA502E   **Invalid use of detail report within one-phase mode.**

Explanation:
The "User settings" panel has one of the one-phase mode fields set to Y. The error occurs because one-phase mode is mutually exclusive with running a separate detail report.

User response:
Either set both one-phase mode fields to N on the "User settings" panel, or run a combined general/detail report. If you change the user settings, you have to re-run the general report as well, as it is not possible to generate any detail report as a separate process while in one-phase mode. See the tutorial topic on one-phase mode for more information.

ALAA503E   **"Current data only" invalid in one-phase mode.**

Explanation:
The "User settings" panel has one of the one-phase mode fields set to Y. The error occurs because one-phase mode is mutually exclusive with running against current data only.

User response:
Turn off the one-phase mode field, whichever is set, if you wish to run against current data within the archive process. You will have to re-run the general/detail reports as well, as just resetting this flag at this time will not generate the input necessary to run. See the tutorial topic on one-phase mode for more information.

ALAA504E   **"Include LOB/XML" mutually exclusive with one-phase mode.**

Explanation:
The "User settings" panel has field "Use 64 bit storage" set to Y, or field "Use work files" set to Y. These fields put you in one-phase mode. One-phase mode cannot be used when including LOB/XML data.

User response:
Either set "Include LOB/XML" to N, or go to the "User Settings" panel and set both one-phase mode fields to N, and then retry.

ALAA505E   **"Use 64 bit storage" mutually exclusive with "Use work files".**

Explanation:
The "User settings" panel has field "Use 64 bit storage" set to Y, and the "Use work files" field also set to Y. These fields are mutually exclusive.

User response:
Set at least one of these fields to N and retry. See the tutorial topic on one-phase mode for more information.

ALAA506E   **Base/Clone tables not supported:** *table1*, *table2*.

Explanation:
A table was found in the result set that is either a base table of a base/clone pair, or the clone table itself. Neither is supported.

User response:

Either remove the base/clone tables from your filter list, or explicitly exclude them from your filter list, and retry. The DBID and OBID of the failing table is included in the message (in decimal).

ALAA507E   **LOB PDS name must be specified.**

Explanation:
You requested to include LOB/XML data, and to generate either UNDO or REDO SQL. This combination requires specification of a LOB PDS name.

User response:
Specify a valid PDS name to contain your LOB/XML data, or do not include LOB/XML data, or do not request any type of SQL. See the tutorial topic on LOB/XML data for more information.

ALAA508E   **Cannot use LOAD option while in one-phase mode.**

Explanation:
You requested to go to the LOAD data into DB2 option, but are also currently using one-phase mode. These options are mutually exclusive. You cannot use the separate LOAD detail data into DB2 panel after running a one-phase job.

User response:
Either run a job including LOAD options in one-phase mode, or turn off one- phase mode (via User Settings panel) and run a job in two-phase mode. Once a two-phase mode job runs, you can then use the LOAD panel separately. See tutorial topic on one-phase mode for more information.

ALAA509E   **Cannot use detail report option while in one-phase mode.**

Explanation:
You requested to go to the detail report option, but are also currently using one-phase mode. These options are mutually exclusive. You cannot use the separate detail report option panel while in one-phase mode.

User response:
To use the separate detail report option panel you must not be in one-phase mode, so use the "User settings" panel to turn off one-phase mode. However, you must also first run a general report in two-phase mode prior to using the separate detail report option. See tutorial topic on "one-phase mode".

ALAA510E   **LOB data present and SQL requested using pre-V9 DB2:** *tableid*, *tableid*

Explanation:
LOB data was found in the result set, and SQL was requested, but the release of DB2 is not greater than or equal to V9. Support for LOB data in SQL requires DB2 V9 or higher. The owning LOB table is identified in the message.

User response:
Remove the identified table using the appropriate filters, or do not request SQL generation for this run.

ALAA511E   **XML data present and SQL requested: *tableid*, *tableid***

Explanation:
XML data was found in the result set, and SQL was requested, and XML is not supported for SQL at this time. The owning XML table is identified in the message.

User response:
Remove the identified table using the appropriate filters, or do not request SQL generation for this run.

ALAA512E   **LOB data present and using pre-V7 DB2: *tableid*, *tableid***

Explanation:
LOB data was found in the result set, but the release of DB2 is not greater than or equal to V7. Support for LOB data in SQL/LOAD requires DB2 V7 or higher. The owning LOB table ID identified in the message as DBID/OBID (in decimal).

User response:
Remove the identified table using the appropriate filters, or do not request LOAD processes for this run.

ALAA514E   **Warnings have forced LOAD processing to stop.**

Explanation:
The LOAD process must stop with a return code = 8 in order to prevent subsequent LOAD steps from processing erroneously. This is due to warnings generated in previous steps of the general or detail report in combination with the "Continue processing after warnings=N" setting in "User settings".

User response:
Identify the warnings in the previous steps and attempt to resolve them. Retry if the warnings can be circumvented.

ALAA515E   **Maximum of 500 DB2 logs to read has been exceeded.**

Explanation:
The requested timeframe results in more than 500 logs to be read. This is considered excessive, and the product cannot continue processing.

User response:
Shorten the timeframe requested and retry.

ALAA516E   **Required SYSCOPY records not found for DBID and PSID: *dbid*, *psid*.**

Explanation:
A page format log record was found, but no SYSCOPY records were found to resolve the original action causing the page format. The DBID and PSID are included in the message to identify the table space for which this occurred.

User response:
It is likely the database/table space was dropped for the identified object, and DB2 therefore deleted the rows from SYSCOPY. This object cannot be processed because the required resources (SYSCOPY data) have been eliminated.

ALAA517E   **Utility type record found, but log-backward was forced.**

Explanation:
While processing the output from the general report (REFILE records), a record was found indicating a utility type action, such as a LOAD record. Such records can only be processed in log-forward mode.

User response:
Most likely, log-backward mode was forced, but cannot be used here. Either change "Log reading mode" to a value of "F" (forward) or "A" (product will automatically determine proper mode, which will be log-forward here).

ALAA518E   **"Current row by key" and LOB inclusion mutually exclusive.**

Explanation:
You requested current row by key, and also have requested inclusion of LOB type columns. The two are mutually exclusive. Current row values cannot be accessed by key when LOB columns are present.

User response:
Either turn off inclusion of LOB columns, or turn off the current row by key fields (i.e., set either/both of them to N), and retry. The LOB inclusion field is on the general reports panel.

ALAA519E   **Editing column order is mutually exclusive with audit mode.**

Explanation:
This message indicates that you requested audit mode to be run along with editing the column order. Loading with audit mode does not allow the column order to be edited.

User response:
Set "Edit column order=N" and retry.

ALAA520E   **LOB/XML data present with LOAD or Mass Delete, not supported.**

Explanation:
LOB/XML data was found in conjunction with an action such as LOAD, LOAD REPLACE, or a Mass Delete. This combination is not currently supported.

User response:

Examine the general/summary reports to determine which objects had these types of actions, and remove them from the run using filters.

ALAA521E	Dyn alloc err: Dsname: *dsname*, Ret: *ret*, Res: *res*

Explanation
Dynamic allocation attempt failed for the specified ddname. The return code and reason codes are SVC 99 standard codes and can be found in the IBM manual "Authorized Assembler Services Guide" in the chapter "Requesting Dynamic Allocation Functions". The most common codes likely to be encountered are:

- 0210/0410 - ddname unavailable or in use by another user
- 0420 - ddname associated with an already opened data set
- 1708 - data set associated with ddname not found

User response:
If the return and reason codes provide enough information, correct the error and retry. Otherwise, contact the vendor.

ALAA522E	Override product dynamic allocation=N requires a disk unit name.

Explanation:
You set "Override product dynamic allocation=N", but left "Disk unit name for dynamic allocations" field blank. This is an invalid combination.

User response:
Set "Disk unit name for dynamic allocations" to a valid value if you wish to use the products dynamic allocation feature. Otherwise, set "Override product dynamic allocation=Y".

ALAA523E	Invalid number was entered. Enter only decimal numbers.

Explanation:
You have entered an invalid number. Only decimal values are permitted. No alphanumeric or hexadecimal values are allowed.

User response:
Correct the input and retry.

ALAA524E	At least one line of version information is needed.

Explanation:
You have not entered any information to be added to the table version file. Please enter the necessary fields for at least one line.

User response:
Enter the versioning information.

ALAA525E	OPEN failure on dsname: *dsname* during table version processing.

Explanation:
The OPEN failed for the dsname. The dsname should exist and must not be for performing the IMPORT or EXPORT command before starting the process.

System action:
Program processing stops and an error message is displayed.

User response:
Verify if the data set name exists and you are authorized to read the dataset (for IMPORT command) or edit the dataset (for EXPORT command).

ALAA526E	Versioning information has been added to *filename*. Enter more or exit.

Explanation:
The versioning information entered has been added to the TVFILE. You can either add more versioning information or exit the panel.

User response:
None.

ALAA527E	Optional VSAM fields must be spaces when overriding dyn allocation.

Explanation:
Overriding dynamic allocation means the ROWDATA VSAM data set must come from the standard naming convention only (i.e., name=userid.ROWDATA.CLUSTER) using only the "Create/Re-create ROWDATA VSAM file" panel. Optional VSAM fields here must be spaces.

User response:
Either set "Optional data sets prefix (VSAM)" to spaces, and "Optional volume serial numbers (VSAM)" to spaces, or set "Override product dynamic allocation" to N.

ALAA528E	Error occurred in time conversion routine.

Explanation:
A date was found/generated which was unable to be converted by the time conversion routine. This can occur when the date is out of range. The current maximum date allowed is GMT 2042/09/17 23:53:47.

User response:
If it seems unlikely that this date would be encountered, an internal error may have occurred; contact your product administrator. Otherwise, correct any time range input that may have generated this date.

ALAA530E   **Unsupported ALTER COLUMN data type found on column: *column*.**

Explanation:
An unsupported ALTER COLUMN data type was found during an attempt to create a static SQL program. Variable length to fixed length conversions, conversions to FLOAT, and conversions to DECIMAL are not currently supported.

User response:
Table versioning is supported via dynamic SQL processing only at this time.

ALAA531E   **Column Len field incorrect for data type.**

Explanation:
The value entered in the Column Len field does not match what the length should be for the data type. SMALLINT is 2. INTEGER is 4. BIGINT is 8. FLOAT is either 4 or 8. DECFLOAT is 8 or 16. CHAR less than 255 characters. VARCHAR less than 32704 characters. GRAPHIC less than 127 double-byte characters. VARGRAPHIC less than 16352 double-byte characters. BINARY less than 255 bytes. VARBINARY less than 32704 bytes.

User response:
Enter the proper value for the data type or change the data type.

ALAA532E   **Scale is only valid for Decimal or Timestamp data type.**

Explanation:
A value was entered in the Scale field but the data type is not Decimal or Timestamp.

User response:
Change the data type to either Decimal or Timestamp or remove the value in the Scale column.

ALAA533E   **Expert options not allowed with Log reading mode=C.**

Explanation:
This error only occurs if detail report panel field "Expert options=Y" is requested in conjunction with Combo mode, which is a product restriction.

User response:
Change the Expert options specification or the Log reading mode specification.

ALAA534E   **Work file mode not allowed with Log reading mode=C.**

Explanation:
This error only occurs if "User settings" panel field "Use work files=Y" is requested in conjunction with Combo mode, which is a product restriction.

User response:

Change the Use work files specification or the Log reading mode specification.

ALAA535E   **Invalid value entered for Mass Delete treatment field.**

Explanation:
Valid values are "I" for individual delete statements or "O" for one mass delete statement.

User response:
Enter a valid value.

ALAA536E   **Invalid combination for Mass Delete and UNDO SQL found.**

Explanation:
This only occurs if Undo SQL field is "Y" or "X" and Treat mass delete fields is "O". If you want to generate Undo SQL, then you must use "I" for Treat mass delete.

User response:
Change one of your values so that you do not have an invalid combination.

ALAA537E   **Showing uncommitted units of work not allowed when resolving UOWs.**

Explanation:
You specified in the "Filters for log data" section that you want to show uncommitted units of work while also specifying that you want to "Resolve started UOWs". These two options are mutually exclusive.

User response:
Change one of those two specifications and retry.

ALAA538E   **Resolve started UOWs is available only with date-time log ranges.**

Explanation:
You specified a log range using RBAs or LRSNs and not the "Start/End Date-Time" fields.

User response:
Specify a log range using the "Start/End Date-Time" fields and retry.

ALAA539E   **Resolve started UOWs is not an option for quiet time report.**

Explanation:
Resolve started UOWs was specified and quiet time report was specified. This is not a supported option for quiet time processing.

User response:
Correct the input and retry.

ALAA540E   **Invalid value for MODEPARM parameter *parmname*.**

Explanation:

An invalid value was entered for the indicated parameter in the MODEPARM DD.

User response:
Correct the value or remove the parameter.

ALAA541E   **OPEN failure on DD MODEPARM.**

Explanation:
An error occurred while attempting to open the DD MODEPARM.

User response:
Verify the existence of the DD statement in the JCL.

ALAA542E   **Unrecognized parameter found while parsing MODEPARM .**

Explanation:
An unrecognized parameter was found while parsing the modeparm file.

User response:
Verify that the parameters in the modeparm file are accurate.

ALAA543I   **Modeparm file successfully created. DSN: *dsname*.**

Explanation:
The modeparm file was successfully created.

User response:
None.

ALAA544E   **OPEN failure on dsname: *dsname* during modeparm creation.**

Explanation:
The OPEN failed for the named data set.

User response:
Verify its existence in the JCL, and for correct attributes.

ALAA545E   **Cannot use Commit Scope=UOW on this run.**

Explanation:
You specified Commit scope=UOW, but an archive file was found and needed in the generation of the SQL that is not compatible with this option. Only archive files generated in releases V3.2 and beyond are allowed, and the current request included an older archive file.

User response:
Change your commit scope value to a numerical value, or change the timeframe so as not to include older (pre-V3.2 generated) archives.

ALAA546E   **Must load general report when only running general report.**

Explanation:

When only running log analysis for the general report and choosing to load data, the option for loading the general report is the only option available.

User response:
Choose to load the general report or bypass loading data into DB2.

ALAA547E   **"Data only= Y" mutually exclusive with "Treat mass deletes=O".**

Explanation:
A request to process data only cannot be requested if mass deletes are not being materialized. Either Data only must be N or Treat mass deletes must be "I".

User response:
Correct the input and retry.

ALAA548E   **Wildcards not allowed in advanced filters.**

Explanation:
A wildcard was found as the ending character in either the table owner or name field. Wildcards are not permitted anywhere in advanced filters. Only object filters by name allow wildcarding. If you force a wildcard into the table owner or name field, results are unpredictable and unsupported.

User response:
Correct the input and retry.

ALAA549E   **Invalid RBA/URID filters.**

Explanation:
Multiple RBAs only allowed using multiple URID filters. It cannot be used as a group of start and end RBA values. This error means you edited the JCL improperly when attempting to use multiple URIDs as input. See the help for field "start/end RBA (URID)" on the "Generate database activity report (general)" panel.

User response:
Correct the input as specified in the help and retry.

ALAA550E   **Maximum of 682 individual URIDs exceeded.**

Explanation:
You included too many URIDs in the run. The maximum allowable is 682. See the help for field "Start/End RBA (URID)" on the "Generate database activity report (general)" panel.

User response:
Correct the JCL and retry.

ALAA551E   **Bad retention period specified.**

Explanation:
Retention period can be a number from 0-9999 only.

User response:
Correct the input and retry.

ALAA552E　　　**ARCHLOG1 and ARCHLOG2 cannot both be "P".**

Explanation:
The value of "P" cannot be specified for both ARCHLOG1 and ARCHLOG2.

User response:
Correct the input so either ARCHLOG1 or ARCHLOG2 have priority, or neither have priority. Setting them both as having priority is invalid.

ALAA553E　　　**ARCHLOG1 or ARCHLOG2 values can only be Y, N, or P.**

Explanation:
You specified a bad value for either ARCHLOG1 or ARCHLOG2.

User response:
Correct the input so ARCHLOG1 and/or ARCHLOG2 are set to the only allowable values of Y (Yes), N (No), or P (Priority).

ALAA554E　　　**"P" setting for either ARCHLOG requires N setting on other.**

Explanation:
You cannot request one ARCHLOG as having priority, and at the same time request to use only the other ARCHLOG. These settings are mutually exclusive. You cannot request both priority to use one of these logs at the same time you request exclusive use only of the other.

User response:
If ARCHLOG1 = "P", then set ARCHLOG2 to N. if ARCHLOG2 = "P", then set ARCHLOG1 to N. If either is set to Y, a value of "P" is not allowed.

ALAA555E　　　**General SORT failure in called program.**

Explanation:
A SORT failure occurred. See SYSOUT/job log for SORT messages related to the failure.

User response:
See SYSOUT/job log output for SORT messages related to the failure. Correct the problem if possible, or contact the product administrator.

ALAA556E　　　**Dictionary mismatch: *v1, v2, v3, v4, v5*.**

Explanation:
An XML value needed to be decompressed, but the current dictionary was created after the row was written to the log. The dictionary cannot be used to decompress the XML. The variables listed show the DBID, PSID, page number, and URID for the row, followed by the dictionary page PGLOGRBA.

User response:

This will not occur in log-forward mode, so the product can be re-run using log-forward mode.

ALAA557E　　　**Continuous mode file report requires continuous mode processing.**

Explanation:
You cannot request a continuous mode file report ("output mode" includes value of "C") without continuous mode processing.

User response:
Either remove the continuous mode file report option or specify a continuous mode file name and retry.

ALAA558E　　　**Wildcards not allowed in load translations.**

Explanation:
A wildcard was found as the ending character in either the table owner or name field. Wildcards are not permitted anywhere in load translations. If you force a wildcard into the table owner or name field, results are unpredictable and unsupported.

User response:
Correct the input and retry.

ALAA559E　　　**The table must have at least one column specified.**

Explanation:
No column was found for the specified table. For column reordering, each table entered must have at least one column that is being altered. If no columns for a table require reordering, the table does not need to be specified.

User response:
Correct the input and retry.

ALAA560E　　　**Relative end date/time incorrectly specified.**

Explanation:
The relative end date/time is specified as "+" or "-" and then an amount in the format 000 thru 999. A plus sign means after the starting date and time whereas a minus sign means before the current date and time. The amount is specified in minutes: 120 means 120 minutes or 2 hours.

User response:
Correct the end date field so the amount is a valid number greater than or equal to 000 and less than or equal to 999.

ALAA561E　　　**Relative end date resulted in end date being less than start date.**

Explanation:
The relative end date/time amount specified resulted with an end date being less than the start date after GMT conversions.

User response:
Correct the end date field so the amount specified results in calculating an end date that is greater than the start date.

| ALAA562E | Relative end date resulted in end time being before the start time. |

Explanation:
The relative end date/time amount specified resulted with an end timeframe being less than the starting timeframe after GMT conversions.

User response:
Correct the end date field so the amount specified results in calculating an end timeframe that is greater than the start timeframe.

| ALAA563E | Relative end date/time processing requires end time to be blank. |

Explanation:
The relative end date/time is specified and an end time was also specified.

User response:
Blank out the end time field or enter an actual end date.

| ALAA564E | Advanced filter parsing error. *v1*, rscd: *v2*. |

Explanation:
During the parsing of the advanced filters an error was found and identified by *v1*. The reason code supplies additional information the product vendor may need during support issues.

User response:
Review the field in question and correct the problem.

| ALAA565E | Impact report and Impact report by row mutually exclusive. |

Explanation:
You can request only one type of impact report, not both.

User response:
Specify either an "I" or a "J", but do not include both.

| ALAA566W | Invalid Detail report row length found in the mode file. |

Explanation:
An invalid detail report row length can indicate that the mode file was not properly updated to the current format or that the mode file was improperly edited. When this warning occurs, the default row length of 80 is used in the detail report.

User response:
Verify that the mode file was converted to the current format and that the mode file was not improperly edited.

| ALAA567E | The modefile is not the correct version for this operation. |

Explanation:
The modefile has not been converted to the proper version needed for this operation. Please use the modefile operations in the Utilities panel to convert this modefile to the proper version.

User response:
Convert the modefile to the proper format.

| ALAA568E | Impact report by row and advanced filters mutually exclusive. |

Explanation:
Advanced filters track rows by column value; impact report by row tracks rows by physical position; the two are deemed incompatible. Use only the standard impact report with advanced filters if necessary.

User response:
Either remove advanced filters or impact report by row.

| ALAA569E | Invalid column ordering specified. |

Explanation:
An error was detected while processing input for editing the load column ordering. This could only happen if you incorrectly edited JCL generated by the ISPF front-end process. Ensure that every table specified has a table owner followed by a table name and that every column name has a corresponding column position.

User response:
Re-generate the JCL using the panels supplied and verify the correctness of this input record. If the product generates a bad input record from the front-end process, contact your product administrator to report the error.

| ALAA570E | Invalid table column reordering detected: *v1*. |

Explanation:
An error was detected while handling the load column ordering for the specified table. This could happen if you incorrectly edited JCL generated by the ISPF front-end process. This indicates that the table specified may have an invalid table owner and name or at least one column has an invalid name or corresponding column position.

User response:
Verify the correctness of the load column ordering. If the ordering appears valid, contact your product administrator to report the error.

| ALAA571E | Advanced filter execution error. Type: *v1*, reason: *v2*. |

Explanation:
During the applying of the advanced filters an error was encountered. The type of error is displayed along with the reason code. The reason codes supplies additional information the product vendor may need during support issues.

User response:
Contact your product administrator to report the error.

ALAA572E	Invalid combination for mass delete and advance filters found.

Explanation:
Advanced filters requires that mass deletes be materialized individually (I).

User response:
Type an "I" in the mass delete treatment field when using advanced filters options.

ALAA573W	Start:end span record mismatch, start LRSN: *v1*.

Explanation:
A start span log record had no matching end span log record.

User response:
Re-running the program after more DB2 activity occurs may resolve this warning as may-be-needed data has not yet been flushed from DB2 log buffers. If such action does not resolve the warning and DB2 has incurred more activity, contact the vendor with contents of warning message.

ALAA574E	Open error for DD TVFILE.

Explanation:
An open attempt failed for this DD statement.

User response:
Check your DD CFILES output for the inclusion of this DD statement, or if hardcoded in your JCL step.

ALAA575E	PDS directory update failed for member: *membername* RC: *rc* Res: *reason*

Explanation:
The directory update for named member failed while using the STOW macro. The return code and reason code are listed in hexadecimal. Return code 12 and Reason Code 0 mean the directory blocks must be increased for the LOB PDS being used.

User response:
Check the return code and reason code included in the message in the IBM document "DFSMS Macro Instructions For Datasets" in the chapter "Non-VSAM Macro Descriptions" for explanations of STOW completion codes.

ALAA576E	Dyn Alloc Err: DDNAME: *ddname*, Ret: *returncode*, Res: *reasoncode*, More info: *text*

Explanation
Dynamic allocation attempt failed for the specified ddname. The return code and reason codes are SVC 99 standard codes and can be found in the IBM manual "Authorized Assembler Services Guide" in the chapter "Requesting Dynamic Allocation Functions". The most common codes likely to be encountered are:

021c
device name is undefined, check unit parameter

9700/970c
primary allocation amount cannot be fulfilled

User response:
Check the parameters in the TICSPECS DD, correct any problems, and retry. If that does not solve the problem, contact the vendor.

ALAA577E	Invalid value found in TICSPECS DD for device type

Explanation
The TICSPECS DD has an invalid value for device type. The valid values are:

D
Disk

T
Tape

I
Input

User response:
Correct the value in the TICSPECS DD and rerun.

ALAA578I	There were no quiet times found.

Explanation:
No quiet times were found for the combination of filters and threshold given. This is only an informational message.

User response:
None

ALAA580E	Invalid line command.

Explanation:
An invalid line command was entered for a profile.

User response:
Enter a correct line command.

ALAA581E	Too many line commands.

Explanation:

More than one line command was found. Only the delete line command can be entered more than one time.

User response:
Enter a single line command or remove all non-delete line commands.

ALAA582E   **Profile name already exists.**

Explanation:
A profile already exists with the specified name.

User response:
Choose a unique profile name.

ALAA583E   **General report load after Detail report in one-phase unavailable.**

Explanation:
You requested a combination of loading from the general report while running a detail report, and all of this in one-phase mode. This is not allowed as the detail report eliminates general report output in one-phase mode.

User response:
If loading general report output while running a complete detail report, set your "Defaults" panel one-phase mode fields to N.

ALAA584E   **CAF error: *v1*.**

Explanation:
An error occurred while running the Call Attach Facility.

User response:
Check the error message to determine what steps need to be taken.

ALAA585E   **The DB2 subsystem cannot be changed when editing profiles.**

Explanation:
The DB2 subsystem for the current profiles are specified when first launching the profile panel. The DB2 subsystem cannot be changed when editing a specific profile.

User response:
Do not edit the DB2 subsystem field. If you wish to use a certain profile on a different subsystem, you must use the export command.

ALAA586E   **Impact report by row requires tracking INSERTs, UPDATEs, DELETEs.**

Explanation:
A request was made for impact report by row, but one or more DML types were excluded, such as bypassing INSERTs, DELETEs, or UPDATEs. All DML types must be included to track activity by row.

User response:

Either remove the impact report by row option, or set all DML fields to Y (i.e., SHOW UPDATES, SHOW DELETES, and SHOW INSERTS ALL = Y).

ALAA587E   **The profile file to import is improperly formatted.**

Explanation:
The profile file cannot be read. It may not be a profile file or it may have been manually edited.

User response:
Verify that the specified file is an exported profile and that it has not been edited.

ALAA588E   **File is PDS, but no member specified.**

Explanation:
You specified a PDS (library) to be used as the file, but did not provide a member name. This is not valid.

User response:
Either specify a sequential file, or a PDS with a member name.

ALAA589E   **The file to be used does not exist.**

Explanation:
You specified a file that does not exist.

User response:
Specify a valid and existing file.

ALAA590E   **File member specified, but dataset is not a PDS.**

Explanation:
A file was specified for use, but it included a member name and is not a PDS (library). Either remove the member name or specify a different file entirely.

User response:
Correct the input and retry.

ALAA591E   **Profile job identifier invalid.**

Explanation:
The profile used a job identifier that was not previously used in a general report profile, or was used and is currently in-use. The job identifier distinguishes files between different runs of different jobs, and these files must exist and not be in-use while generating report JCL.

User response:
For the first case described above, you must first create a general report profile prior to creating any detail report or load profile. Also, the detail report and load profiles must use a job identifier that was specified in the general report profile. When generating the mode file or the JCL for a detail report or load profile, the corresponding mode file or JCL must first be generated for the general report profile.

ALAA592E   **Open failure for output DD *v1* , rc=*v2*.**

Explanation:
An open request for the specified data set failed.

User response:
Verify the information associated with the DD is correct.

ALAA593I   **DD *v1*not found. No report was produced.**

Explanation:
The DD listed was not present and prevented the report from being produced.

User response:
Verify the DD exists in the JCL.

ALAA594I   **Previous filters have been saved; add more or exit.**

Explanation:
The filters on the previous screen have been saved. You can add more filters or hit PF3 to exit and use the saved filters, or type CANCEL to exit and bypass usage of the previously saved filters.

User response:
None.

ALAA595I   **Previous filters have been removed; add more filters or exit.**

Explanation:
The filters from the previous screen have been removed. You can add more filters or hit PF3 to exit and use the saved filters, or type CANCEL to exit and bypass usage of any previously saved filters.

User response:
None.

ALAA596E   **The field must be empty when using the "COLUMN" command.**

Explanation:
The command to display the columns of a table only takes a table owner and table name as input. All other fields should be blank.

User response:
Correct the input and retry.

ALAA597E   **Not enough memory to store the data.**

Explanation:
The memory used to store the data would be exhausted with the current input. The cursor indicates the specific data where the memory overflow will occur.

User response:

Either remove the data that will cause the memory overflow or manually edit the JCL to add more data.

ALAA598E   **Apostrophes in the PDS name are not allowed.**

Explanation:
The inputted JCL PDS name contains one or more apostrophes. This is invalid.

User response:
Remove the apostrophes and enter a valid data set.

ALAA599E   **A detail report profile cannot use one-phase mode.**

Explanation:
The profile being editing is specified as a detail report profile. This is mutually exclusive with one-phase mode processing.

User response:
In the "User Settings" panel, turn off one-phase mode.

ALAA600E   **A load report profile cannot use one-phase mode.**

Explanation:
The profile being editing is specified as a load report profile. This is mutually exclusive with one-phase mode processing.

User response:
In the "User Settings" panel, turn off one-phase mode.

ALAA601E   **Duplicate column name entered.**

Explanation:
A column name for a table has been entered more than one time. A column may only be specified once.

User response:
Remove the duplicate column name and retry.

ALAA602E   **Duplicate column name found in column ordering.**

Explanation:
A column name for a table has been entered more than one time in DD TRANS. A column may only be specified once.

User response:
Remove the duplicate column name and retry.

ALAA603E   **Duplicate position entered.**

Explanation:
A position for a table has been entered more than one time. Each position may only be specified once.

User response:
Correct the duplicate position value and retry.

ALAA604E   **Duplicate position found in column ordering.**

Explanation:

A position for a table has been entered more than one time in the JCL DD. Each position may only be specified once.

User response:
Remove the duplicate position value and retry.

ALAA605E	Missing date field encountered while parsing DATAIN DD.

Explanation:
Start date and end date fields must be present in the DATAIN DD.

User response:
Add the necessary fields to the DATAIN DD and try again.

ALAA606E	Missing time field encountered while parsing DATAIN DD.

Explanation:
Start time and end time fields must be present in the DATAIN DD.

User response:
Add the necessary fields to the DATAIN DD and try again.

ALAA607E	Column name *V1* does not exist.

Explanation:
A column was specified for LOAD column ordering that does not exist.

User response:
'Correct the specified column name and try again.

ALAA608I	Export was successful.

Explanation:
Exporting the profile completed successfully.

User response:
None.

ALAA609I	Import was successful.

Explanation:
Importing the profile completed successfully.

User response:
None.

ALAA610E	QSAM for ROWDATA mutually exclusive with one phase mode

Explanation
You requested QSAM for ROWDATA, but also requested one of the following:

• One phase 64 bit mode
• Work file mode

When you request QSAM for ROWDATA you are implicitly requesting two phase mode, so one phase fields must = N.

User response:
In the "User Settings" panel, set one phase mode fields to N when QSAM for ROWDATA = Y, or set QSAM for ROWDATA = N.

ALAA611E	Field must be empty when using any filter command.

Explanation:
Object filter commands require all other fields to be left blank. Miscellaneous filter commands only require the conditional field to be filled out, but all other fields must be left blank.

User response:
Correct the input and retry.

ALAA612E	Unexpected error while loading ALA@DDLR, RC: *rc*.

Explanation:
An error occurred while loading the LOAD module ALA@DDLR.

User response:
Ensure that ALA@DDLR is present in the LOAD library and then contact the product administrator to notify the vendor.

ALAA613E	Unexpected error while unloading ALA@DDLR, RC: *rc*.

Explanation:
An error occurred while unloading the LOAD module ALA@DDLR.

User response:
Contact the product administrator to notify the vendor.

ALAA614E	An error occurred during the catalog audit report.

Explanation:
An error occurred during the catalog audit report step.

User response:
Check the DD ALAPRINT for more error information.

ALAA616W	A warning occurred during catalog audit report.

Explanation:
A warning occurred during the catalog audit report step.

User response:
Check the DD ALAPRINT for more error information.

ALAA617E	Catalog audit report requires REDO or UNDO SQL.

Explanation:

In order to create a catalog audit report, REDO SQL, including triggers, or UNDO SQL, including triggers, must also be generated.

User response:
Correct the entry and retry.

ALAA618W	SQL stmt not valid due to: set clause only had non-updatable cols.

Explanation:
The set to be produced only contained columns that are not updatable in an SQL statement. Some examples of this are generated always columns and generated always as row change timestamp columns. Because the SQL would be invalid, it is not generated.

User response:
See the tutorial topic "Generation of undo/redo sql" that explains that not all SQL can be generated by Log Analysis Tool.

ALAA619E	Invalid DB2 version format

Explanation:
An invalid DB2 version format was specified for the DB2VER field while attempting to bind the SQL for the Log Analysis Tool.

User response:
Enter the DB2 version in the correct format and run the bind again.

ALAA620E	Error occurred during attempt to offload to zIIP.

Explanation:
An error occurred while trying to offload a process to the zIIP. Check for other messages in the job log to determine what the issue may be.

User response:
Check for other messages in job log and correct the problem, or turn off zIIP processing in the user settings panel.

ALAA621I	Default value substituted for zIIP level.

Explanation:
When the field is blanked out, the default value is used.

User response:
Verify the default value and modify if necessary.

ALAA622E	Invalid value entered for zIIP level; must be between 00-16 only

Explanation:
The value entered for this field is incorrect. Enter a value between 0 and 16 only. 0 means do not send anything to the zIIP.

User response:

Correct the input and retry.

ALAA623E	Expert options are mutually exclusive with Audit Mode.

Explanation:
This message indicates you requested audit mode to be run along with expert options. Loading with audit mode does not allow expert options.

User response:
Correct the input and retry.

ALAA624W	A null value found during LOAD processing; *lrsn*, *col_num*

Explanation:
The "Bypass NULLIF" expert option was selected for the LOAD step. A null value was found during processing and this may prevent the LOAD utility from successfully running. The LRSN of the activity and the column number where the null was found are displayed.

User response:
Check the LOADFILE before attempting to run the LOAD utility.

ALAA625E	Table versioning and drop column data mutually exclusive

Explanation:
Values were supplied for both the table version number and the dropped column log point. Only one of those fields can be non-blank. If you wish to add an entry for table versioning, the DROP COLUMN LOGPOINT field must be blank, and alternately, an entry for a dropped column must have a blank TABLE VERSION field.

User response:
Correct the input and retry

ALAA626E	Bad LOGPOINT value. Must be 20 byte valid hex.

Explanation
The LOGPOINT value is bad. It must be a 20 byte hex display value. For example, this is correct:

```
00023CA345A2667800AB
```

and these are incorrect:

- `0000QW12345678901234--` (invalid characters QW)
- `000D123456789C--`(must be 20 bytes)

The value must come from a SYSCOPY START_RBA field (see panel help).

User response:
Correct the input and retry.

ALAA627E   **RBA or LRSN value has incorrect length.**

Explanation:
You requested use of either RBAs or LRSNs, but the lengths of the values entered are not correct. An RBA or LRSN must be either 12 characters long or 20 characters long. The same length must be used for all entries.

User response:
Correct the input and retry.

ALAA628E   **Unsupportable drop column activity on tablespace: *tbsp_name***

Explanation:
While processing the output of the general report, it was determined that the named tablespace had "drop column" activity, but no TVFILE records for that object were found.

User response:
Supply the necessary supporting "drop column" record(s). This requires going to "user settings" and selecting the "edit table versioning/drop column file" option to add any data about dropped columns. See the tutorial for more information on this topic.

ALAA629I   **Default value substituted for contention wait time.**

Explanation:
When the field is blanked out, the default value is used.

User response:
Verify the default value and modify if necessary.

ALAA630E   **Invalid value entered for Wait contention time - 00-99 only.**

Explanation:
The value entered for this field is incorrect. Enter only a value between 0 and 99. A value of 0 means do not wait for the data set to become available. A value from 01 - 99 means wait this amount of minutes for the data set to become available.

User response:
Correct the input and retry.

ALAA631E   **Too many line commands.**

Explanation:
More than one line command was found. Only the Select line command can be entered more than one time.

User response:
Enter a single line command or remove all non-select line commands.

ALAA632E   **Output flags cannot be changed when writing reports to datasets.**

Explanation:
The output flags cannot be changed from the MODEPARM file if the "Write reports to datasets" option is set to Y. To change the output flags while writing the reports to datasets, edit the ISPF panels and regenerate the JCL.

User response:
Set the output flags in the MODEPARM file to the original values or edit the output flags in the ISPF panels and regenerate the JCL.

ALAA633E   **Open failure on DD INFOM**

Explanation:
An open request failed for the specified DD name.

User response:
Verify the DD name (INFOM) existence in the JCL. If not there, the DD can be added or the JCL recreated. If the problem persists, you should contact your product administrator and/or the vendor.

ALAA634E   **Bad VSAM ROWDATA key length; recreate it in "User settings".**

Explanation:
The VSAM ROWDATA file is not correct for this release.

User response:
Re-create the VSAM ROWDATA file, or avoid VSAM usage completely by selecting either of the one-phase mode options or by using QSAM for ROWDATA. See the **User settings** panel option for details on all those methods.

ALAA635W   **Required SQL from *tablename* was not found.**

Explanation:
While creating the catalog audit report, partial information about a DDL action was found. However, information from the specified table was needed to fully analyze the DDL statement. The DDL action is skipped and will not appear in the report.

User response:
Check the tutorial topic "Report description (Audit catalog report)" and ensure that all filters for the desired DDL action are included in the filter list. Correct the input and retry.

ALAA636E   **Catalog audit report requires where clause by key to be set to n.**

Explanation:
The "Where clause by key" field must be set to N when requesting a Catalog audit report.

User response:
Correct the input and retry.

ALAA637E   **Minimum volume count field must be numeric and between 1 and 255.**

Explanation:
The minimum volume count field must be between 001 and 255 only, without any non-numeric characters.

User response:
Correct the input and retry.

ALAA638E   **Catalog audit report requires inclusion of catalog data.**

Explanation:
The catalog audit report requires information from the catalog tables. If a catalog audit report is desired, "Include catalog data" should be set to Y when running the general report. If not set, there will be no data to report.

User response:
Correct the input by setting "Include catalog data" to Yon the general report panel and retry.

ALAA639W   **Bad return code from ZPARM lookup, Reason CD: <reason code>**

Explanation
When trying to obtain information about the ZPARM member entered in the setup panels, an error condition was encountered. Please verify all setup information entered to be accurate. Common reason codes are listed below.

- 0012 - ZPARM member not found or the load attempt of the member failed.

- 0116 - The first Db2 LOADLIB in the subsystem concatenation is blank.

- 0120 - Bad Db2 LOADLIB found in the subsystem concatenation.

User response:
Validate all information entered in the setup panel.

ALAA640E   **"Delimited output=Y" mutually exclusive with audit mode.**

Explanation:
This message indicates that you requested the creation of a delimited output file, but you also requested "Audit mode=Y" (or S). The delimited output file cannot be used with audit mode processing.

User response:
Set "Audit mode=N" and retry.

ALAA641E   **"Delimited output=Y", "Header=N", "Split load files=N" not valid.**

Explanation:

This message indicates that you requested the creation of a delimited output file, but you did not request header information, or you did not request the LOAD files to be split. When the delimited output file is created for multiple objects in a combined file, there is no way to distinguish which record belongs to which object without the object owner and name.

User response:
Set "Header=Y" or "Split load files=Y" and retry.

ALAA642E   **"Delimited output=Y" mutually exclusive with "Change flag".**

Explanation:
This message indicates that you requested the creation of a delimited output file and also requested the change indicator to be included. These two options are not valid together. The **Change flag** field must be N if the **Delimited output** field is set to Y.

User response:
Correct the input and retry.

ALAA643I   **Please verify header fields when changing to/from delimited output**

Explanation:
The header fields for a delimited output file are handled differently than the header fields for a non-delimited output file.

User response:
Verify that the header fields are correctly set up for the output file type you have chosen and continue.

ALAA644E   **The values entered for the delimiters must be unique.**

Explanation:
One or more delimiters have the same value. The value that is entered for the delimiters must be unique.

User response:
Modify one or more delimiters and retry.

ALAA645E   **An invalid hexadecimal value was entered for a delimiter field.**

Explanation:
The value that was entered contains an invalid hexadecimal digit, is missing a digit, or cannot be used. There must be an even number of hexadecimal digits entered. Hex zeros is not a valid delimiter.

User response:
Enter a valid combination of hexadecimal digits and retry.

ALAA646E   **Header columns Table Creator and/or Table Name were not selected.**

Explanation:

Table Creator and Table Name are required header fields when creating delimited output files if the option to split the LOAD files was not chosen; however, one or both fields were not selected.

User response:
Select both Table Creator and Table Name, or specify LOAD files to be split.

ALAA647E	Split LOAD cards can only be used when split LOAD files is used.

Explanation:
The option to split LOAD cards is only allowed in conjunction with the option to split LOAD files.

User response:
Either select the option to split LOAD files or do not select the option to split LOAD cards.

ALAA648E	Option CHAR data in hex cannot be used with delimited output files.

Explanation:
The details report option CHAR data in hex is being used. This option is not valid for use with delimited output files.

User response:
Choose either CHAR data in hex or delimited output files, not both.

ALAA649E	Column ordering is not available with delimited output files.

Explanation:
The option to edit column order was selected along with the option to create a delimited output file. These two options are mutually exclusive.

User response:
Do not choose to edit column order when creating a delimited output file.

ALAA650E	LOB header page encountered.

Explanation:
LOBs are not supported. LOB page cannot be parsed.

User response:
Ensure that LOB data is not being included and retry.

ALAA651E	An unexpected abend occurred during zIIP processing.

Explanation:
An in-progress task was terminated on the zIIP processor and an SVC dump was captured.

User response:
Notify your product administrator to contact the vendor.

ALAA652E	Edit procedure found during zIIP processing for table: &V1

Explanation:

A table with an edit procedure requires special processing to get the data. This processing cannot be performed when the job is zIIP enabled. The table is identified by the DBID/PSID/OBID.

User response:
Remove the table from the filter list or turn off zIIP enabled processing.

ALAA653E	Table versioning found for table: &V1

Explanation:
Only the latest version of a table is supported. While processing the data, a row for the identified table was found that was in a previous versions format. The table is identified by DBID/PSID/OBID.

User response:
Remove the table from the filter list or turn off zIIP enabled processing.

ALAA654E	A CELL64 free request failed.

Explanation:
An attempt to free the memory failed.

User response:
Check the job log for further information.

ALAA655E	General report data is not available with delimited output files.

Explanation:
The option to load general report data was selected along with the option to create a delimited output file. These two options are mutually exclusive.

User response:
Do not select a delimited output file when loading general report data.

ALAA656E	Expert options are not available with delimited output files.

Explanation:
An expert option other than the 'Override Generated Always columns' option was selected along with the option to create a delimited output file. Only 'Override Generated Always columns' expert option is available with the delimited output files.

User response:
Do not change expert options values except for 'Override Generated Always columns' when creating a delimited output file.

ALAA657E	Mass deletes must be materialized with delimited output files.

Explanation:
The option to show a mass delete as one statement instead of individual delete statements is mutually exclusive with delimited output files.

User response:
Do not select a delimited output file when not materializing mass deletes.

| ALAA658E | Invalid record found in LOADOPT DD. |

Explanation:
While reading the LOADOPT DD for expert load options, an invalid record was found. The LOADOPT DD is correctly formatted from the ISPF menus.

User response:
Correct the invalid record or rebuild the JCL.

| ALAA659E | General report without EXTFILE preceded detail report; invalid. |

Explanation:
A general report without populating the EXTFILE does not generate the appropriate input necessary to run a detail report. The MODEFILE indicates that the last general report run did not populate the EXTFILE DD, and therefore, it is not possible to now run a detail report.

User response:
Run a general report with the EXTFILE DD option prior to running a detail report.

| ALAA660E | Table owner / name length must be between 008 and 128 only. |

Explanation:
The value that is entered for the table owner / name length must be a valid number between 008 and 128. It must be entered as three digits without spaces.

User response:
Correct the value and retry.

| ALAA661E | Invalid combination of entries for header "ACTION/COUNT". |

Explanation:
When using header **ACTION/COUNT**, the load expert parameters **General report data**, **Fixed length table names**, and **Header fields in EBCIDIC only** must all be specified Y.

User response:
Either deselect **ACTION/COUNT** or specify Y for **General report data**, **Fixed length table names**, and **Header fields in EBCIDIC only**.

| ALAA662E | General report without EXTFILE preceded hispeed process; invalid. |

Explanation:
A general report without populating the EXTFILE does not generate the appropriate input necessary to run in high speed. The modefile indicates the general report run did not populate the EXTFILE DD, and therefore, it is not possible to now run in high speed.

User response:
Run a general report with the EXTFILE DD option prior to processing in high speed.

| ALAA663E | WORKF1 or WORKF2 DD missing for 64 bit conversion |

Explanation:
The program was operating in 64 bit mode, but is in the process of being converted to work file mode. The DD of WORKF1 or WORKF2 was not found but it is needed.

User response:
The JCL must contain DDs for WORKF1 and WORKF2. It may be that the JCL was generated in a prior release. Regenerate the JCL.

| ALAA664E | Invalid MEMID/URID filter specified. |

Explanation:
The MEMID/URID filter is not in the correct format. The proper format is MEMID/URID =*mmmm/uuuuuuuuuuuuuuuuuuuu*. *mmmm* is the four-digit character representation of the member ID, which contains the desired URID. If this is a non-data sharing region, use 0000 for the member ID. *uu..uu* is the 20-character representation of the URID.

User response:
Correct the MEMID/URID filter and re-run the job.

| ALAA666W | LOADREC bypassed, LOB/XML data found on table: *table_owner . table_name* |

Explanation:
A LOB or XML column was found in a table modified during the specified log range. Load records can only be generated on such column types if panel field "Include LOB/XML data=Y" on the "Generate Database Activity Report (General)" panel. The identified table activity will not be included in the LOADFILE. Program processing halts only if warnings are deemed fatal.

User response:
If you want to include LOB/XML data types in the LOADFILE, you must re-run the job and set the panel field "Include LOB/XML data =Y", or do not request to create the LOADFILE.

| ALAA667E | LOB processing requires both pre and post rows for updates. |

Explanation:
LOB and XML data was included. The load step requires that both the update pre-row and the update post-row be included.

User response:
Either do not include LOB or XML data, or include both pre- and post-update rows on the load panel.

ALAA670E   **An invalid GMT offset was specified. Use format +HH:MM.**

Explanation:
An invalid value was entered for use as the GMT offset. The format is a plus or minus sign followed by HH:MM where HH is hours and MM is minutes. A valid entry would be -05:00 which represents GMT minus 5 hours.

User response:
Correct the input and retry.

ALAA671E   **The specified GMT offset is out of range.**

Explanation:
The GMT offset entered is out of range of known GMT offsets. The valid range is -12:00 to +14:00.

User response:
Correct the input and retry.

ALAA672E   **Open error for DD CATRPT.**

Explanation:
An open attempt failed for this DD statement.

User response:
Add the DD CATRPT to the JCL, or rebuild the JCL from the ISPF panels and resubmit the job.

ALAA673E   **Detail Impact Report needs Impact Report or Impact Report by Row**

Explanation:
The Detail Impact Report requires the information from an Impact Report or an Impact Report by Row.

User response:
Correct the input and retry.

ALAA674E   **Impact Reports require a General Report or Summary Report**

Explanation:
Impact Report and Impact Report by Row require a General Report or a Summary Report.

User response:
Correct the input and retry.

ALAA675E   **Cannot allocate continuous mode file with dsname *dsname***

Explanation:
The allocation failed for the dsname. The user needs the necessary permissions to create or access the dsname. The dsname is based on the data set prefix and the job identifier.

User response:
Verify that you are authorized to create or access the data set.

ALAA676E   **The profile is already a classic profile**

Explanation:
The profile is already in classic mode and cannot be converted.

User response:
Correct the input and retry.

ALAA677W   **Bootstrap DSN #01 does not match entry in control file.**

Explanation:
The Bootstrap DSN #01 entry in the control file does not match the Bootstrap DSN #01 entry in the BSDS. This can lead to reading the same archive log files multiple times.

User response:
Correct the entry in the control file and retry.

ALAA678W   **Rows removed during error processing for table: *dbid,psid,obid***

Explanation:
An error occurred on a table while the **Continue processing after warning** parameter was set to E. The specified table encountered an error; in order to continue processing, the table was removed from the job.

User response:
Check the job log to identify the error that caused the table to be removed.

ALAA679W   **S0C7 encountered during expansion of row in table: *dbid,psid,obid***

Explanation:
A S0C7 abend occurred on a table during row expansion while the **Continue processing after warning** parameter was set to E. This table is removed from the job and processing continues.

User response:
Check the joblog to identify any further information about the S0C7.

ALAA680E   **"Fixed external format =Y" is mutually exclusive with Audit Mode**

Explanation:
This message indicates you requested the creation of a fixed column external format file, but you also requested "Audit Mode=Y" (or S). The fixed column external format file cannot be used with audit mode processing.

User response:
Set "Audit Mode=N" and retry.

ALAA681E **Fixed external format file requires header or split load files.**

Explanation:
This message indicates you requested the creation of a fixed column external format output file, but you did not request header information or you did not request the load files to be split. When the fixed column external format output file is created for multiple objects in a combined file there is not a way to distinguish which record belongs to which object without the object owner and name included via the headers.

User response:
Set "Header=Y" or "Split load files=Y" and retry.

ALAA682E **"Fixed external format=Y" mutually exclusive with "Change Flag".**

Explanation:
This message indicates you requested the creation of a fixed column external format output file and also requested the change indicator to be included. These two options are not valid together. Therefore, the "Change Flag" field must be N if the "Fixed external format" field is set to Y.

User response:
Correct the input and retry.

ALAA683I **Please verify header fields when changing to/from external format file**

Explanation:
The header fields for a fixed column external format output file are handled differently than the header fields for other LOADFILE formats. Please verify that the header fields are setup properly for the output file type you have chosen.

User response:
Verify the header fields and / or continue.

ALAA684E **Option CHAR data in hex cannot be used with fixed external format files.**

Explanation:
The details report option CHAR data in hex is being used. This option is not valid for use with fixed external format files.

User response:
Change CHAR data in hex to 'N'.

ALAA685E **Column ordering is not available with fixed external format.**

Explanation:

The option to edit the column order was selected along with the option to create a fixed external format file. These two options are mutually exclusive.

User response:
Do not edit the column order while creating a fixed external format file.

ALAA686E **General report data is not available with fixed external format.**

Explanation:
The option to load general report data was selected along with the option to create a fixed external format output file. These two options are mutually exclusive.

User response:
Do not select fixed external format file when loading general report data.

ALAA687E **Mass Deletes must be materialized with fixed external format.**

Explanation:
The option to show a mass delete as one statement instead of individual delete statements is mutually exclusive with fixed external format output files.

User response:
Do not select fixed external format file when not materializing mass deletes.

ALAA688E **"Fixed external format =Y" is mutually exclusive with delimited output.**

Explanation:
This message indicates you requested the creation of a fixed column external format file, but you also requested "Delimited output=Y" (or S). The fixed column external format output and the delimited output file are mutually exclusive.

User response:
Choose one option or the other, but not both.

ALAA689E **Include LOAD step in JCL is in conflict with another chosen option.**

Explanation:
This message indicates that you requested to include the LOAD step in the JCL, but also chose a load format that is incompatible with this option. Delimited output, fixed external output, and split load files cannot contain the LOAD step in the JCL.

User response:
Set include LOAD step in JCL to N.

ALAA690E **The relative end date amount exceeds the maximum value of 1440.**

Explanation:
The maximum allowed value for the relative end date is 1440. Change the amount to a value between 0000 and 1440. This amount can be also entered in the format 000 through 999 for backward compatibility with prior releases.

User response:
Enter a value less than 1440.

ALAA700E	Parameter value too long. Maximum of 3 chars for each value.

Explanation

The value entered for the parameters is too long. The value must be maximum 3 characters long. For example:

- Correct: TIMERV=005, REBLDCT=010
- Incorrect: TIMERV=0005, REBLDCT=005

User response:
Correct the value and try again.

ALAA701E	CMDRPT DD statement not found in JCL.

Explanation:
Reporting on Db2 commands was requested, but the required DD statement CMDRPT was not present in the JCL.

User response:
The JCL may not have been created with the correct release of the product, which should include the DD CMDRPT. Verify that this DD statement exists.

ALAA702E	The specified 64 bit mem threshold is out of range.

Explanation:
The 64 bit memory threshold must be a valid number between 40 and 99. A value of 00 is also allowed, but it signifies that the feature is not being used.

User response:
Correct the input and retry.

ALAA703E	64 bit mem threshold needs continuous mode file specified.

Explanation:
The 64 bit memory threshold must be used in a continuous mode job. The continuous mode file name must be entered.

User response:
Correct the input and retry.

ALAA704E	64 bit mem threshold cannot be used with a detail report.

Explanation:

A detail report was requested while using the 64 bit memory threshold. These are mutually exclusive.

User response:
Correct the input and retry.

ALAA705W	Stopping log reads due to 64 bit mem threshold.

Explanation:
The maximum percentage of 64 bit memory usage has been exceeded. The current job stops without reading the additional archive logs for the time period. This job should be resubmitted so the additional archive logs can be read and processed.

User response:
Resubmit the continuous mode job to finish processing the time period.

ALAA706E	Open failure for MODEBKP output file

Explanation:
An open request for the specified data set failed.

User response:
Check your DD CFILES output for the inclusion of this DD statement, or if hard-coded, in your JCL step.

ALAA707E	Mismatch encountered between MODEFILE and JCL.

Explanation:
The MODEFILE and JCL are out of synch. One is set up for one-phase mode (1 Phase) while the other is set up for two-phase mode (2 Phase).

User response:
Regenerate or restore the MODEFILE and/or JCL so both are in synch.

ALAA708E	Misc flags values H and I are mutually exclusive.

Explanation:
Either choose H for high speed or I for high speed with LOAD LOG YES, but not both.

User response:
Correct the input and retry.

ALAA709W	Unknown utility found for DBID, PSID and LRSN: *dbid*, *psid*, *lrsn*.

Explanation:
A page format log record was found, but no SYSCOPY records were found to resolve the original action that causes the page format. The DBID, PSID and LRSN are included in the message to identify the table space and time for this activity. These records will be identified by a Z in the action column of general reports and general report loads.

User response:
Check for the Z action to identify the utility records.

ALAA710E   Initialization failure of Language Environment®.

Explanation:
During LOB decompression, the Language Environment could not be initialized. The return code indicates the failure.

User response:
Determine if a proper Language Environment is available. Contact your product administrator or the vendor if no resolution is available.

ALAA711E   zEDC environment not available, ret: *return_code* res: *reason_code*

Explanation:
LOB decompression requires an active zEDC environment, but none was found. zEDC return code and reason code are included. These codes are documented in z/OS MVS Programming: Callable Services.

User response:
Determine if a proper zEDC environment is available. Contact your product administrator or the vendor if no resolution is available.

ALAA712E   zEDC rendezvous token failure, ret: *return_code* res: *reason_code*

Explanation:
LOB decompression failed during zEDC rendezvous token call. zEDC return code and reason code are included. These codes are documented in z/OS MVS Programming: Callable Services.

User response:
Determine the problem based on the return and reason codes. Contact your product administrator or the vendor if no resolution is available.

ALAA713E   zEDC buffer registration failure, ret: *return_code* res: *reason_code*

Explanation:
LOB decompression failed during zEDC buffer registration. zEDC return code and reason code are included. These codes are documented in z/OS MVS Programming: Callable Services.

User response:
Determine the problem based on the return and reason codes. Contact your product administrator or the vendor if no resolution is available.

ALAA714E   zEDC decompression failure, ret: *return_code* res: *reason_code*

Explanation:
LOB decompression failed during decompression itself. zEDC return code and reason code are included. These codes are documented in z/OS MVS Programming: Callable Services.

User response:
Determine the problem based on the return and reason codes. Contact your product administrator or the vendor if no resolution is available.

ALAA715E   zEDC deregistration failure, ret: *return_code* res: *reason_code*

Explanation:
LOB decompression failed during zEDC buffer deregistration. zEDC return code and reason code are included. These codes are documented in z/OS MVS Programming: Callable Services.

User response:
Determine the problem based on the return and reason codes. Contact your product administrator or the vendor if no resolution is available.

ALAA716E   zEDC rendezvous token free up, ret: *return_code* res: *reason_code*

Explanation:
LOB decompression failed during zEDC rendezvous token free up. zEDC return code and reason code are included. These codes are documented in z/OS MVS Programming: Callable Services.

User response:
Determine the problem based on the return and reason codes. Contact your product administrator or the vendor if no resolution is available.

ALAA717E   Language Environment clean up failure

Explanation:
LOB decompression failed during Language Environment clean up.

User response:
Determine the problem based on the return code. Contact your product administrator or the vendor if no resolution is available.

ALAA718E   Invalid combination of fields entered.

Explanation:
A support report has been chosen along with the option to bypass all support reports. This combination is not valid.

User response:
Either remove all support reports from your selection, or remove the selection to bypass all support reports.

ALAA719E   LOADFILE record exceeds Db2 32K pagesize limit.

Explanation:
A record that is being generated for the LOADFILE exceeds the Db2 32K pagesize limit. The record cannot be processed by the Db2 LOAD utility. If discards are being captured, then Db2 will write the

record to the discard file. If discards are not being captured, then Db2 will stop processing the LOADFILE and issue an error message.

User response:
Either reduce the number of headers being selected so the LOADFILE record length will be shorter, or manual intervention with the record will be required.

ALAA720W**Table truncated due to length:** *table_owner.table_name*

Explanation:
During LOADCARD generation, it was determined that the target table for the load will exceed the 32k limit. The final column has been truncated.

User response:
This most likely occurred because headers have been included in the output. Those headers may push the length of the target table over the 32K boundary. This does not present a problem unless the actual length of any row in the LOADFILE exceeds 32K; variable length fields may likely keep it under that. If the data length for any row exceeds 32K, Db2 will not allow the load. Target tables with this message cannot be created within the product.

ALAA721E**CONTINUEIF parameter incompatible with another parameter**

Explanation:
The CONTINUEIF option cannot be used with general report data, delimited output files, fixed external format files, or audit mode files.

User response:
Either set the CONTINUEIF parameter to N, or turn off general report data, delimited output files, fixed external format files, or audit mode file selections.

ALAA722E**Modify split LOADFILE DSN parameter incompatible with another parm**

Explanation:
The modify split LOADFILE DSN option cannot be used when generating a single load file and load card.

User response:
Either set the modify split LOAD DSN parameter to N, or turn on the split load file or the split load cards option.

ALAA723E**Duplicate node name was specified**

Explanation:
The node name cannot have the same name as other default DD names in the JCL. The node name column cannot contain duplicated values.

User response:

Either remove duplicates from the table or modify repeated values.

ALAA724E**Duplicate tables were specified**

Explanation:
A table owner and table name pair should be unique per job.

User response:
Either remove duplicates from the table or modify repeated values.

ALAA725W**LOADFILE DSN is not specified for the table:** *table_owner.table_name*

Explanation:
The table was not specified in LOADFILE DSN mapping for split load files/load cards, so a default value is used for that table.

User response:
Specify a node name for the table, or a default node name will be used.

ALAA726E**The length of the node name is greater than 7:** *node_name*

Explanation:
The node name in the LOADDSN DD must be less than or equal to 7 characters, because it is used as a part of the data set name. If you enter a name with more than 7 characters, the product will use only the first 7 characters.

User response:
Make sure that you have no whitespace before the value of this field and the equal sign. Make sure that the length of the value of the node name is less or equal to 7 characters.

ALAA727E**Unknown field identifier in LOADDSN DD:** *field_identifier*

Explanation

There are three valid field identifiers for this DD:

- `TABLE OWNER =`
- `TABLE NAME =`
- `NODE NAME =`

If a line starts with another prefix, this error is issued because of undefined behavior of data processing.

User response:
Make sure that all your field identifiers are correct.

ALAA728I**Previous DSN customizations have been saved; add more or exit**

Explanation:
The DSN customizations on the previous screen have been saved. You can add more, or press PF3 to exit

and use the saved ones, or type CANCEL to exit and bypass usage of any previously saved customizations.

User response:
No action is required.

ALAA729E	Duplicate node name was specified: *node_name*

Explanation:
A node name cannot have the same name as other default DD names in the JCL. Node name fields cannot contain duplicated values.

User response:
Either remove duplicates from the LOADDSN DD or modify repeated values.

ALAA730E	Duplicate tables were specified *table_owner.table_name*

Explanation:
A table owner and table name pair should be unique per job.

User response:
Either remove duplicates from the LOADDSN DD or modify repeated values.

ALAA731W	Data sharing mode was not specified for data sharing environment.

Explanation:
Data sharing mode was set to N when in a data sharing environment. As a result, only the logs for the specified subsystem are processed.

User response:
Specify Data Sharing Mode = Y when in a data sharing environment, or leave this field set to N if you only want to process the logs for the specified subsystem.

ALAA732E	Invalid timestamp, specify using format YYYY/MM/DD-HH:MM:SS.MMMMMM.

Explanation:
Timestamp input is a string that should be represented in YYYY/MM/DD-HH:MM:SS.MMMMMM format: Year(4)/Month(2)/Day(2)-Hours(2):Minutes(2):seconds(2).Fractional seconds(6).

User response:
Correct the input and retry.

ALAA733E	Invalid hexadecimal STCK TO LRSN DELTA supplied.

Explanation:
Only a 20-character hexadecimal display value is allowed here. Valid values might be 0000000000000005BA10 or

0000000000000000FF34. Invalid values might be BC or 00000MMM234.

User response:
Correct the input and retry.

ALAA734E	Invalid hexadecimal LRSN supplied.

Explanation:
Only a 20-character hexadecimal display value is allowed here. Valid values might be 00C7AAA0125421050200 or 00D503A33D4E2B123456. Invalid values might be BC or 000000MMM23400000000.

User response:
Correct the input and retry.

ALAA735E	STCK TO LRSN DELTA value is too large.

Explanation:
The STCK TO LRSN DELTA value should be lower than the LRSN value, as it will be subtracted from the LRSN value prior to converting it to a timestamp field.

User response:
Correct the input and retry.

ALAA736I	Timestamp to LRSN translation is available from the Generate Database Activity Report (General) ISPF panel by pressing the PF06 key or entering the XTSTAMP command

Explanation:
When on the **Generate Database Activity Report (General)** ISPF panel, you can use the PF06 key or the XTSTAMP command to issue a timestamp to LRSN conversion. A panel will display with the new start and end LRSN values.

User response:
No action is required.

ALAA737E	Duplicate table space names were specified for Quiet Time report.

Explanation:
Only one table is allowed per universal table space. The table space name should be unique.

User response:
Use separate table spaces to store each table.

ALAA738E	Invalid value: An integer between 1 and 30 or blank is allowed

Explanation:
A valid entry is any integer between 1 and 30 that indicates the header position in the resulting DDL, or blank.

User response:

Correct the input and retry.

ALAA739E **The position field must be blank if the header is not included**

Explanation:
If the "Include?" field value is set to N, the header is not included. Therefore, the position field must be blank.

User response:
Correct the input and retry.

ALAA740I **Previous positions have been removed; add more positions or exit**

Explanation:
The positions from the previous screen have been removed. You can add more positions, press PF3 to exit and use the saved positions, or type CANCEL to exit and bypass usage of any previously saved positions.

User response:
No action is required.

ALAA742W **Clone table record found and bypassed:** *dbid obid*

Explanation:
A clone table activity was found in log records. Clone tables are not currently supported. The DBID and OBID of the affected table are listed in the message text. Program processing depends on the value that is specified for the **Continue processing after warnings** field on the **User Settings** panel.

User response:
Remove the identified clone table by using the appropriate filters.

ALAA743E **ROWDATA DD statement not found in JCL**

Explanation:
You requested to generate the detail report, but the required DD statement ROWDATA was not found in the JCL. Processing halts.

User response:
Verify that the ROWDATA DD statement exists. If the DD statement is missing, regenerate the JCL, or add the missing DD statement.

ALAA744I **Table versioning was detected, new entries were added to the table**

Explanation:
Table versioning was detected during the "Track schema changes" process. The corresponding records were added to the ALA_TABLE_VERSIONING table.

User response:

No action is required.

ALAA745I **No table versioning was detected for the specified time frame**

Explanation:
Table versioning was not detected during the "Track schema changes" process. No new records were added to the TVFILE.

User response:
No action is required.

ALAA746W **DBname:** *database_name*, **TSname:** *table_space_name*, **not found in Db2**

Explanation:
Table versioning was detected for the specified database/table space during the "Track schema changes" process. The database/table space was dropped and bypassed for table versioning processing. Program processing halts only if warnings are deemed fatal.

User response:
To continue processing, set the **Continue processing after warnings** parameter to Y.

ALAA747E **JSONIN parsing error:** *error text*

Explanation:
A syntax error was found during the JSONIN DD parsing process. The message includes a brief description and a line dump.

System action:
Program processing halts and an error message is issued.

User response:
Correct the input and retry.

ALAA748E **JSONIN validation error:** *error text*

Explanation:
An error occurred during the JSONIN DD validation process. The message includes a brief description.

System action:
Program processing halts and an error message is issued.

User response:
Correct the input and retry.

ALAA749E **JSON parameters file cannot be used in 2 phase mode VSAM**

Explanation:
The JSON parameters file cannot be generated when 2-phase mode VSAM is used.

System action:
Program processing halts and an error message is issued.

User response:
Correct the input and retry.

ALAA750W　　JSON parameters file ignores header reordering

Explanation:
When using the JSON parameters file, header reordering is ignored and the included headers have their default positions in a LOADCARD/LOADFILE.

System action:
None

User response:
No action is required.

ALAA751E　　JSON parameters file requires EXTFILE DD population

Explanation:
To use the JSON parameters file EXTFILE DD population is required.

User response:
Correct the input and try again.

ALAA752E　　Detail Transaction Report needs General Transaction Report.

Explanation:
The Detail Transaction Report requires information from the General Transaction report.

User response:
Correct the input and try again.

ALAA753E　　Compensation log record encountered that cannot be processed

Explanation:
The log record was split between multiple log datasets and could not be rolled back successfully. This occurs only if the split between log datasets occurs in the header section of the log record.

User response:
Rerun the JOB with the option Compensation recs = Y

ALAA754E　　LOB/XML data set allocation parameters inconsistency.

Explanation
The LOB/XML data set was allocated outside the product with an inconsistent set of attributes. The attributes of the PDS must be the following:

```
LRECL=32756, BLKSIZE=32760, RECFM=VB,DSORG=PO
```

System action:
Program processing halts and an error message is issued.

User response:

Please allocate LOB/XML PDS with the required allocation parameters and return the job.

ALAA755E　　Misc flags values X and M are mutually exclusive.

Explanation:
Either choose X to bypass SYSLGRNX or M to bypass SYSLGRNX only when MODIFY RECOVERY utility removed entries.

System action:
Panel processing halts and an error message is issued.

User response:
Correct the input and retry.

ALAA756E　　Old format record found in continuous mode file.

Explanation:
The product found an old format record in the continuous mode file. Such records cannot be processed. This situation occurs only when an uncommitted activity and a maintenance update are encountered between two continuous mode jobs.

System action:
Program processing halts and an error message is issued.

User response:
To proceed further, clear out the continuous mode file and rerun the previous iteration of continuous mode (to pick up the uncommitted activity in a current record format). Next, run the current iteration again and continue with regular runs.

ALAA757W　　No unique index found for table &V1.&V2

Explanation:
&V1 represents the table owner, and &V2 is the table name. The 'Print only changed columns' option requires a unique index for each reported table. Index columns will be printed along with the changed columns to provide information of the changed row in the table.

ALAA758E　　Error occurred during row formatting for table: &V1.

Explanation:
The product could not resolve row formatting. This situation can occur if the TVFILE populated incorrectly or some entries are missing. The message indicates the table IDs (DBID,PSID,OBID), the LRSN and the version number.

System action:
Program processing halts and an error message is issued.

User response:

To proceed further, add missing table versioning entries manually or run the "Track Schema Changes" utility to automatically populate TVFILE with missing records. After that, rerun to the job. If you cannot resolve this issue, contact IBM Customer Support.

ALAA759E	The data set for JCL generation is in use.

Explanation:
The data set where generated JCL would be placed is already in use and cannot be edited.

User response:
Make sure that data set is free and try again.

ALAA760I	Find string not found.

Explanation:
The requested find string was not found.

ALAA812E	The FIND command requires a match string.

Explanation:
FIND command was executed, but find string was not specified.

User response:
Specify find string and try again.

ALAA813E	The RFIND key can only be used after a FIND character string is entered.

Explanation:
RFIND key was used, but FIND character string was not entered.

User response:
Specify FIND character string and try again.

ALAA818E	Invalid command parameters.

Explanation:
Invalid command parameters were used with the specified command.

User response:
Correct command parameters and try again.

ALAA901E	The product default load library could not be located.

Explanation:
The product default load library could not be located.

System action:
Panel processing halts and an error message is issued.

User response:
Verify the startup clist has been defined with proper library settings.

ALAA902E	A Db2 subsystem ID has to be entered for processing.

Explanation:

A Db2 subsystem ID has to be entered for processing.

System action:
Panel processing halts and an error message is issued.

User response:
Enter a valid DB2 subsystem ID.

ALAA903E	A default GDG base dataset name could not be located.

Explanation:
The default GDG base database name could not be located.

System action:
Panel processing halts and an error message is issued.

User response:
Create model GDG and retry.

ALAA904E	The VASM dataset and V1 could not be opened for I/O.

Explanation:
The specified dataset (control file) could not be opened for I/O.

System action:
Panel processing halts and an error message is issued.

User response:
Contact your product administrator to verify the control file was properly defined.

ALAA905E	VSAM error occurred during the control file read.

Explanation:
An unexpected error from VSAM was encountered while doing a read of the control file.

System action:
Panel processing halts and an error message is issued.

User response:
Contact your product administrator to notify the vendor.

ALAA906I	The control file record for DB2 subsystem and SYS has been successfully updated.

Explanation:
The control file record for the named DB2 subsystem has been successfully updated.

System action:
None.

User response:
None.

ALAA907E	VSAM error occurred during the control file update.

Explanation:

An unexpected return code from VSAM was encountered while doing an update operation of the control file.

System action:
Panel processing halts and an error message is issued.

User response:
Contact your product administrator to notify the vendor.

ALAA908I	The control file record for DB2 subsystem and SYS have been successfully added.

Explanation:
The control file record for the named DB2 subsystem has been successfully added.

System action:
None.

User response:
None.

ALAA909E	The range of valid values is 1 - 3 and 5.

Explanation:
The product default load library could not be located.

System action:
Panel processing halts and an error message is issued.

User response:
Correct the input and retry.

ALAT001I	DISCOVER job ended successfully

Explanation:
The Tools Customizer DISCOVER job ended successfully.

User response:
None.

ALAT002E	DISCOVER job failure; BIND library/member name was not found

Explanation:
The Tools Customizer DISCOVER job failed because a BIND library/member name was not found.

User response:
Verify correctness of input and retry.

ALAT003E	DISCOVER job failure; Control file name was not found

Explanation:
The Tools Customizer DISCOVER job failed because the control file name was not found.

User response:
Verify correctness of input and retry.

ALAT004E	DISCOVER job failure; Fatal storage error while processing ALATCZD

Explanation:
The Tools Customizer DISCOVER job failed because of a fatal storage error while processing ALATCZD.

User response:
Contact IBM Software Support.

ALAT005E	DISCOVER job failure; DISCOVER output dsname allocation failure

Explanation:
The Tools Customizer DISCOVER job failed because Tools Customizer could not allocate the DISCOVER output data set.

User response:
Verify the correctness of input and the availability of the file in regards to z/OS. Retry.

ALAT006E	DISCOVER job failure; CLIST library allocation failure

Explanation:
The Tools Customizer DISCOVER job failed because Tools Customizer could not allocate the CLIST library.

User response:
Verify the correctness of the input and retry.

ALAT007E	DISCOVER job failure; Unable to OPEN control file

Explanation:
The Tools Customizer DISCOVER job failed because Tools Customizer was unable to OPEN the control file.

User response:
Verify the correctness of the input and the authority to open it. Retry.

ALAT008E	DISCOVER job failure; Unable to GET control file record

Explanation:
The Tools Customizer DISCOVER job failed because Tools Customizer was unable to GET the control file record.

User response:
Verify the correctness of the input and the authority to open it. If the field input appears correct, contact IBM Software Support.

ALAT009E	DISCOVER job failure; CLIST library or member name was not found

Explanation:
The Tools Customizer DISCOVER job failed because the CLIST library or member name was not found.

User response:

Verify the correctness of the CLIST library input and that the member name ALAFRONT resides within it. Retry.

FEC801E   **AN INVALID SQLCODE OF** *code* **WAS ENCOUNTERED.** *messages***.**

Explanation:
An invalid SQLCODE was encountered.

User response:
If you cannot resolve this issue using the messages displayed, contact IBM Customer Support.

FEC802E   **AN INVALID RETURN CODE OF** *code* **WAS ENCOUNTERED ON** *function* **FUNCTION. THE ERROR MESSAGE TEXT FOLLOWS:** *messages*

Explanation:
An invalid return code was encountered.

User response:
If you cannot resolve this issue using the messages displayed, contact IBM Customer Support.

FEC803E   **The first character of the command is not a dash. Correct the syntax of the DB2 command and resubmit.**

Explanation:
You entered a command without using a dash as the first character.

User response:
Correct the syntax of the DB2 command and resubmit.

FEC901E   **The Rocket Software default load library could not be located.**

Explanation:
Explanation in message.

User response:
The data set name entered for the DB2 Tools Load Library was not found. Enter a valid loadlib data set name and continue.

FEC902E   **A DB2 subsystem ID has to be entered for processing.**

Explanation:
Explanation in message.

User response:
Enter a valid DB2 subsystem name and continue.

FEC903E   **The default GDG base dataset name could not be located.**

Explanation:
Explanation in message.

User response:
Enter a valid model data set name and continue.

FEC904E   **The specified dataset could not be opened for I/O.**

Explanation:
Explanation in message.

User response:
Verify that the VSAM data set is accessible and continue.

FEC905E   **An unexpected return code from VSAM was encountered while doing a read of the control file. RC=code.**

Explanation:
A VSAM READ error occurred while attempting to access the data set specified for the DB2 Control Dataset.

User response:
The RC (VSAM return code) is provided for investigation. Refer to DB2 Messages and Codes documentation to resolve and then continue.

FEC906I   **The control file record for DB2 subsystem ssid has been successfully updated.**

Explanation:
Explanation in message.

User response:
None required.

FEC907E   **An unexpected return code from VSAM was encountered while doing an update operation of the control file. RC1=code RC2=code**

Explanation:
A VSAM update error occurred while attempting to update the data set specified for the DB2 Control Dataset.

User response:
The RC1 and RC2 (VSAM return codes) are provided for investigation. Refer to DB2 Messages and Codes documentation to resolve and then continue.

FEC908I   **The control file record for DB2 subsystem ssid has been successfully added.**

Explanation:
Explanation in message.

User response:
None required.

FEC909E   **The range of valid values is 1 to 3 and 5.**

Explanation:
Explanation in message.

User response:
Enter a valid value.

FEC910E   **An unexpected return code from VSAM was encountered while doing an add operation to the control file. RC1=code RC2=code**

Explanation:
A VSAM error occurred while attempting to perform an add operation to the specified DB2 Control Dataset.

User response:
The RC1 and RC2 (VSAM return codes) are provided for investigation. Refer to DB2 Messages and Codes documentation to resolve and then continue.

FEC911E   **No Find string specified**

Explanation:
The (F)IND command was entered but no parameters were specified.

User response:
Enter a FIND parameter.

FEC912I   **Find string not found**

Explanation:
The requested find string was not found.

User response:
None required.

FEC914E   **An unknown column was specified using the SORT command.**

Explanation:
Explanation in message.

User response:
Verify that you correctly typed the name of the column or select another column. Ensure that SORT criteria includes an underscore in place of any spaces when attempting to SORT by name.

FEC915E   **SORT is not supported for the specified column.**

Explanation:
Explanation in message.

User response:
You must select a different column.

FEC916E   **Sort column not entered. Column name or number must be specified.**

Explanation:
Explanation in message.

User response:
Specify the column name or number for SORT. Ensure that if column name is used, that spaces in the name are replaced with underscores.

FEC918I   **Bottom of data reached**

Explanation:
The search function has reached the bottom of the data.

User response:
Press PF5 to continue searching from the top.

FEC919I   **Top of data reached**

Explanation:
The search function has reached the top of the data.

User response:
Press PF5 to continue searching from the bottom.

FEC920E   **File tailoring open returned a file tailoring already in progress condition.**

Explanation:
An attempt to perform file tailoring for utility customization failed. There was file tailoring session already in progress. File tailoring sessions cannot be performed concurrently.

User response:
Contact IBM Customer Support.

FEC921E   **File tailoring open returned the output file already in use condition -- ENQ failed.**

Explanation:
An attempt to open the DB2 Control Data Set failed with an ENQ error. The data set is already open for output.

User response:
Verify that you are the only user attempting to access this file.

FEC922E   **File tailoring open returned the skeletal file or output file not allocated condition.**

Explanation:
An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response:
Verify that all required files are allocated prior to performing file tailoring.

FEC923E   **File tailoring open returned a severe error condition.**

Explanation:
An attempt to perform file tailoring failed because a severe error condition was encountered on open.

User response:
Verify that all required files are allocated and accessible prior to performing file tailoring.

FEC924E   **File tailoring open returned an unknown code -- severe error.**

Explanation:
An attempt to perform file tailoring failed because a severe error condition was encountered on open.

User response:
Verify that all required files are allocated and accessible prior to performing file tailoring.

FEC925E   **File tailoring close returned a file not open condition -- severe error.**

Explanation:
An attempt to perform file tailoring failed because a File-Not-Open condition was encountered on close.

User response:
Verify that all required files are allocated and accessible, and that there are no other tailoring sessions running concurrently with your session.

FEC926E   **File tailoring close returned an output file in use condition.**

Explanation:
An attempt to perform file tailoring failed because an Output-File-Inuse condition was encountered on close.

User response:
Verify that all required files are allocated and accessible, and that there are no other tailoring sessions running concurrently with your session.

FEC927E   **File tailoring close returned a skeletal file or output file not allocated condition.**

Explanation:
An attempt to close file tailoring failed because either a tailoring skeleton file or output file was not allocated.

User response:
Verify that all required files are allocated and accessible, and that there are no other tailoring sessions running concurrently with your session.

FEC928E   **File tailoring close returned a severe error.**

Explanation:
An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response:
Verify that all required files are allocated and accessible prior to performing file tailoring.

FEC929E   **File tailoring close returned an unknown code -- severe error.**

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response:
Verify that all required files are allocated and accessible prior to performing file tailoring.

FEC930E   **File tailoring close returned a output member exists in the output library and NOREPL was specified.**

Explanation:
An attempt to perform file tailoring failed because the close process could not replace the pre-existing tailored member in the output file.

User response:
Change the output member name to a new name or ensure that the output library allows for member replacement.

FEC931E   **File tailoring include returned a skeleton does not exist condition.**

Explanation:
An attempt to perform file tailoring failed because the tailoring process could not locate a required tailoring skeleton.

User response:
Ensure that all required files are allocated to perform file tailoring.

FEC932E   **File tailoring include returned a skeleton in use -- ENQ failed condition.**

Explanation:
An attempt to access a tailoring skeleton failed with an ENQ error (member-in-use).

User response:
Verify that all required tailoring files are allocated, and that there are no other tailoring sessions running concurrently.

FEC933E   **File tailoring include returned a data truncation or skeleton library or output file not allocated condition.**

Explanation:
An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response:
Verify that all required files are allocated prior to performing file tailoring.

FEC934E   **File tailoring include returned a severe error condition.**

Explanation:

An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response:
Verify that all required files are allocated and accessible prior to performing file tailoring.

FEC935E	File tailoring include returned an unknown condition -- severe error.

Explanation:
An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response:
Verify that all required files are allocated and accessible prior to performing file tailoring.

FEC936E	Allocation Error - The ISPFILE DD is already allocated and can not be deallocated - Process not completed.

Explanation:
The ISPFILE DD allocation failed. The DD is already allocated and cannot be deallocated for this TSO session. The process did not complete successfully.

User response:
Contact IBM Customer Support.

FEC937E	Allocation Error - An error was encountered allocating the ISPWRK1 or ISPWRK2 DD - Process not completed.

Explanation:
Explanation in message.

User response:
Verify TSO session parameters are set correctly for your site prior to allocation of these DD statements.

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.

Explanation:
Changes that were made to the product parameters were saved in the data store.

System action:
None.

User response:
No action is required.

CCQB001I	The DB2 parameter data was saved in the data store.

Explanation:

FEC938E	The dataset entered is a partitioned dataset and the member name is required.

Explanation:
A required field was not entered. The data set entered is a PDS (partitioned data set) and a member in this PDS must be referenced.

User response:
Enter a member name and retry.

FEC939E	The only valid values are "T" for tracks and "C" for cylinders.

Explanation:
Explanation in message.

User response:
Enter a valid value.

FEC940E	The specified dataset could not be found in the MVS catalog.

Explanation:
One or more of the specified data sets could not be found. The cursor is located on the line for the data set in question.

User response:
Ensure that the data set name is correct and that the data set exists in the catalog.

FEC941E	Enter a FIND command

Explanation:
A Repeat FIND was issued before a FIND command was issued. You must issue FIND before Repeat FIND will work.

User response:
Enter a FIND command before attempting to use RFIND.

Changes that were made to the Db2 parameters were saved in the data store.

System action:
None.

User response:
No action is required.

CCQB002I	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.

CCQB003E   **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.

CCQB004I   **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.

CCQB005E   **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.

CCQB006E   **One row must be selected.**

Explanation:
One row in the table must be selected.

System action:
Processing stops.

User response:
Select one row.

CCQB007E   **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.

CCQC000I   **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.

CCQC001W   **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.

CCQC002I   **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.

System action:
None.

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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 the default value for the programmer name 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 404. 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 404. 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.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQC018S   **The *template_name* template in the *library_name* metadata library is not valid because it does not contain any data.**

Explanation:
The specified template is missing required data.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQC019S   **The *template_name* template in the *library_name* metadata library is not valid because an)ENDDOT file-tailoring control statement is missing.**

Explanation:
A)ENDDOT file-tailoring control statement is required in the specified template.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQC021S   **The *template_name* template in the *library_name* metadata library is not valid because the template must start with the *parameter_name* job card parameter.**

Explanation:
The specified template must start with the specified job card parameter.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQC022S   **The parameters used in a)DOT file-tailoring control statement exceeded the number of allowed parameters in the *template_name* template. The template is in the *library_name* metadata library. The error occurs in)DOT section *section_number*.**

Explanation:
A)DOT file-tailoring control statement can contain only a limited number of parameters.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQC023S   **The)DOT file-tailoring control statement must include the *table-name* table name in the *template_name* template. The template is in the *library_name* metadata library. The error occurs in)DOT section *section_number*.**

Explanation:

The)DOT file-tailoring control statement is missing a required table name.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQC024S   **ISPF file tailoring failed for the *template_name* template in the *library_name* metadata library.**

Explanation:
An error occurred during ISPF file tailoring for the specified template.

System action:
Processing stops.

User response
Review the Tools Customizer-generated trace data set and the ISPF file tailoring trace data set. To create an ISPF file tailoring trace data set, complete the following steps:

1. Run Tools Customizer until the error is about to occur.

2. Specify the ISPFTTRC command, and press Enter.

3. Issue the Tools Customizer command that issues the error.

4. Specify the ISPFTTRC command, and press Enter. The ISPF file tailoring trace data set is created. It adheres the following naming convention: *TSO_ID*.ISPFT.TRACE, where *TSO_ID* is the TSO user ID that is being used.

If the problem persists, gather the following information and contact IBM Software Support.

- 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 404.
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 404.
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 404.
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 404.
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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQD008S   **The XML structure of the *member_name* environment index 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 environment index member.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 404. 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 404. 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 404. 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 404. 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 404. 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 404. Contact IBM Software Support.

CCQD108S	**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 404. Contact IBM Software Support.

CCQD109S   **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 404.
Contact IBM Software Support.

CCQD110S   **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 404.
Contact IBM Software Support.

CCQD111S   **The XML structure of the**
_member_name_ product 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 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 404.
Contact IBM Software Support.

CCQD112S   **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 404.
Contact IBM Software Support.

CCQD113S   **The XML structure of the**
_member_name_ product index
member is not valid.
The _attribute_name_ attribute in
the _element_name_ element is
unknown.

Explanation:
The specified attribute in the product index member is
unknown.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

CCQD118S   **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 404.
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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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:
None.

User response:
Specify another customization library qualifier on the **Tools Customizer Settings** panel.

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:
The specified product environment member contains an unknown element.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404.
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 404.
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 404.
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 404.
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 404.
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 404.
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 404.
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 404. 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 404. 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 404. 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 404. 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.

| CCQD353E | 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.

| CCQD354E | 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.

CCQD355E   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.

CCQD356S   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.

CCQD357I   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.

CCQD358I   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.

CCQD359I   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.

CCQD360S   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:
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 404. 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:
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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 entry 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 Db2 entry was not completely copied because a product can be associated with only 1200 Db2 entries.

System action:
Processing stops.

User response:
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 the 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:
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 Db2member 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

| 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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

CCQD604S   **The XML structure of the *member_name* product customization 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 404.
Contact IBM Software Support.

CCQD605S   **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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

CCQD606S   **The XML structure of the *member_name* product customization 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 404. Contact IBM Software Support.

| CCQD607S | The XML structure of the *member_name* product customization 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 404. Contact IBM Software Support.

| CCQD608S | The XML structure of the *member_name* product customization 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 404. Contact IBM Software Support.

| CCQD609S | The XML structure of the *member_name* product customization 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 404. Contact IBM Software Support.

| CCQD610S | The XML structure of the *member_name* product customization 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 404. Contact IBM Software Support.

| CCQD611S | The XML structure of the *member_name* product customization 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 does not contain required content.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

| 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

| 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.

User response:

See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 *group_attach_name* 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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

System action:
Processing stops.

User response:
Specify a valid sort field.

CCQF084E   **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.

CCQF085E   **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.

CCQF086E   **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.

CCQF087E   **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.

CCQF088E   **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.

CCQF089I   **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:
No action is required.

CCQF110I   **To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.**

System action:
None.

User response:
No is action required.

CCQF111I   **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 is action required.

CCQF112I   **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.

CCQF113I   **The specified values have been saved.**

System action:
None.

User response:
No action is required.

CCQF114I   **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.

CCQF116I   **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.

CCQF117I   **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.

CCQF110I   **To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.**

System action:
None.

User response:
No is action required.

CCQH001W   **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.

CCQH006W   **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.

CCQH007E   **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.

CCQH008W   **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.

CCQH009E   **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.

CCQI000W   **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 stops.

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 "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 404. 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 404. 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 404. 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 404. 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:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI008S　　**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.**

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI009S　　**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.**

Explanation:
The specified attribute did not occur enough times.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI010S　　**The XML structure of the *member_name* Db2 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 404. Contact IBM Software Support.

CCQI011S	**The XML structure of the *member_name* Db2 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 404. Contact IBM Software Support.

CCQI012S	**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 404. Contact IBM Software Support.

CCQI013S	**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.**

Explanation:
The specified attribute in the Db2 parameter metadata member is unknown.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI014S	**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*.**

Explanation:
The specified value of the element is not a valid value.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI015S	**The content of the 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 of the attribute is not a valid value.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI016S	**The content of the DB2 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 is not a valid data type.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI017S	**The content of the DB2 parameter 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 404. Contact IBM Software Support.

CCQI050S   **The *member_name* Db2 parameter metadata member was not found in the *data_set_name* data set.**

Explanation:
Tools Customizer could not find the specified Db2 parameter metadata member.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI051S   **The *parameter_name* LPAR parameter in the *template_name* template does not have associated metadata in the *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 404. Contact IBM Software Support.

CCQI052S   **The *parameter_name* product parameter in the *template_name* template does not have associated metadata in the *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 404. Contact IBM Software Support.

CCQI053E   **The following metadata data set was not found: *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.

CCQI054E   **The following metadata data set could not be opened: *data_set_name*.**

Explanation:
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.

CCQI055S   **The CCQ$$DB2 Db2 parameter metadata member was not found in the *data_set_name* Tools Customizer metadata data set.**

Explanation:
Tools Customizer could not find the Db2 parameter metadata member in the specified Tools Customizer metadata data set.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI056S   **The CCQ$$LPR LPAR parameter metadata member was not found in the *data_set_name* data set.**

Explanation:
Tools Customizer could not find the specified LPAR parameter metadata member.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI057S   **The *member_name* product parameter metadata member was not found in the *data_set_name* data set.**

Explanation:
The product parameter metadata member was not found in the specified data set.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI058I	*Product_name* does not have any Db2 parameters.

Explanation:
Db2 parameters are not required to customize the specified product.

System action:
Processing continues.

User response:
No action is required.

CCQI059I	*Product_name* does not have any LPAR parameters.

Explanation:
LPAR parameters are not required to customize the specified product.

System action:
Processing continues.

User response:
No action is required.

CCQI060S	The *parameter_name* Db2 parameter in the *task_description* task 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.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI061S	The *parameter_name* LPAR parameter in the *task_description* task 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.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI062S	The *parameter_name* product parameter in the *task_description* task 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.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 404. 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 404. 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* parameter metadata member.

Explanation:
Associated metadata is missing for the specified parameter in a task and step.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. Contact IBM Software Support.

CCQI073S	**The XML structure of the *member_name* Db2 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 already exists for an attribute.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI074S	**The XML structure of the *member_name* LPAR 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 already exists for an attribute.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI075S	**The XML structure of the *member_name* product 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 already exists for an attribute.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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:
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.

CCQI102S   **The XML structure of the** *member_name* **LPAR parameter metadata member is not valid. The** *element_name* **element is unknown.**

Explanation:
The specified element in the LPAR parameter metadata member is unknown.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI103S   **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.**

Explanation:
The specified element cannot contain content.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI104S   **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.**

Explanation:
The specified element requires content.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI105S   **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 404. Contact IBM Software Support.

CCQI106S   **The XML structure of the *member_name* LPAR 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 404. Contact IBM Software Support.

CCQI107S   **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.**

Explanation:
The specified element does not occur enough times.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI108S   **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.**

Explanation:
The specified attribute occurs too many times.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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:
See "Gathering diagnostic information" on page 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 stops.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI203S **The XML structure of the *member_name* information 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 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 404. 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 404. 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 404. 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 404. 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 404. 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 404. Contact IBM Software Support.

CCQI212S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI213S   **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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 exceeds *maximum_number* characters.**

Explanation:
The specified attribute contains too many characters.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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*.SALADENU.

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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 sequence metadata member in the metadata data set.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404. 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 404. Contact IBM Software Support.

CCQI360S	**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 404. Contact IBM Software Support.

CCQI361S	**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 404. 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 the 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.

CCQI401S	The XML structure of the *member_name* parameter metadata member is not valid. The PL/I XML parser issued the following exception error 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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.

User response:

See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

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 404.
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 404.
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 404. 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 404. Contact IBM Software Support.

CCQI416S   **The content of the *member_name* 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 parameter metadata member is not valid.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQI417S   **The content of the *member_name* 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 parameter metadata member is not valid.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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:

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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:

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404. 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 404.
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 404.
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 404.
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 404.
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 stops.

User response:
See "Gathering diagnostic information" on page 404.
Contact IBM Software Support.

CCQI603S   **The XML structure of the *member_name* product customization parameter 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 404. Contact IBM Software Support.

CCQI604S	The XML structure of the *member_name* product customization parameter 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 404. Contact IBM Software Support.

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 404. 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 404. 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 404. Contact IBM Software Support.

CCQI608S	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 404. 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 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 404. Contact IBM Software Support.

CCQI610S	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 404. Contact IBM Software Support.

CCQI611S **The XML structure of the *member_name* product customization 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 does not contain required content.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. 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 404. Contact IBM Software Support.

CCQI613S **The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in**

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 404. 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 404. 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 404. 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 404.
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 404.
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 404.
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 404.
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 404.
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* 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 404.
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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404.
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 404.
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 404.
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 404.
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 404.
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 404.
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 404.
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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. 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 404. Contact IBM Software Support.

CCQI755S   **The XML structure of the *member_name* 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.

CCQO000W   **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.

CCQO001S   **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. Contact IBM Software Support.

CCQO002S   **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 404. Contact IBM Software Support.

CCQO003S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO004S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO005S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO006S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO007S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO008S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO009S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO010S   **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.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO011S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO012S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO013S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO014S   **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:
A The specified value for an element in the discover parameter metadata member is not valid.

System action:
Processing stops.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO015S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO016S   **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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

CCQO017S   **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 404. 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.

CCQO058W	The *parameter_name* product 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 404. Contact IBM Software Support.

CCQO059W	The *parameter_name* product 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.

CCQO060W	The *record_name* 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 404. Contact IBM Software Support.

CCQO061I	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

Tip: Using the Discover EXEC saves time and reduces errors that can error when parameters are specified manually. If you want to use the Discover EXEC, specify the required information on the **Discover Customized Product Information** panel. Otherwise, press End to continue without discovering data from a previous customization of the product.

CCQO062W	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.

CCQO064W	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 404. Contact IBM Software Support.

CCQO065W	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 404. Contact IBM Software Support.

CCQO066W	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.

CCQO067W	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.

CCQO068W	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.

CCQO069W   **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.

CCQO070W   **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.

CCQO071W   **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.

CCQO072S   **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 404. Contact IBM Software Support.

CCQO073E   **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 404. 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.

User response:
See "Gathering diagnostic information" on page 404. Contact IBM Software Support.

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 404. 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 library in the correct format. 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*.SALADENU.

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.S*xxx*SAMP.**

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.S*xxx*SAMP, 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 (Formatted 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 *copied_configuration_ID* was successfully copied from *configuration_ID*.**

Explanation:
The specified configuration ID was copied.

System action:
None.

User response:
No action is required.

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.

CCQT006E	**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.

CCQT007E	**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.

CCQT008E	**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 Log Analysis Tool.

CCQT010I   **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.

CCQT011I   **The _configuration_ID_ configuration**
was not copied.

Explanation:
The specified configuration was not copied.

System action:
Processing stops.

User response:
No action is required.

CCQT012I   **The _configuration_ID_ configuration**
was not removed.

Explanation:
The specified configuration was not removed.

System action:
Processing stops.

User response:
No action is required.

CCQT013I   **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.

CCQT014E   **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.

CCQT015E   **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.

CCQT016I   **The _configuration_ID_ configuration**
was not created

Explanation:
The specified configuration was not created.

System action:
Processing stops.

User response:
No action is required.

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.

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*.SCCQTENU 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.

Chapter 25. Tools Customizer reference

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 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 Log Analysis Tool 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**
>> Db2 Log Analysis Tool does not support Db2 group attach names.

> **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 Log Analysis Tool. If the product to be customized requires Db2 entries, you can customize Db2 Log Analysis Tool only on Db2 entries that are in the associated list. When you customize Db2 Log Analysis Tool, 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 Log Analysis Tool.

> **Master list**

>> The list of all Db2 entries that are defined but are not associated with Db2 Log Analysis Tool. 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 master list. When you create a new Db2 entry, it is added to the master list, and when you associate the new entry with Db2 Log Analysis Tool, it is removed from the master list and added to the associated list. The master 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 Log Analysis Tool, you can associate existing entries from the master list to the associated list.

You can create new Db2 entries and copy existing entries to the master 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 Log Analysis Tool. These parameters are defined by Db2 Log Analysis Tool and are stored in a data member that is defined by Db2 Log Analysis Tool.

LPAR parameters
Parameters on the local LPAR that are required to customize Db2 Log Analysis Tool. 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 Log Analysis Tool, 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 Log Analysis Tool will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

Table 14. Status types for the product, the LPAR, and the Db2 entries

Status	Product	LPAR	Db2 entries
Incomplete	The required product parameters are not defined.	The required parameters are not defined.	The required parameters are not defined.
Discovered	The product parameter definitions were discovered by using the product Discover EXEC.	N/A	N/A
Ready to Customize	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.	The required LPAR parameters are defined or LPAR parameters are not required.	The required Db2 parameters are defined or Db2 parameters are not required.

Status	Product	LPAR	Db2 entries
Verify Values	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.	The required LPAR parameter values are defined, but they either have not been verified or verification is not enabled on the LPAR Parameters panel.	The required Db2 parameter values are defined, but they either have not been verified or verification is not enabled on the Db2 Parameters panel.
Customized	The jobs are customized on the local LPAR.	The jobs are customized for the product or for all of the associated Db2 entries on the local LPAR.	The jobs are customized for the Db2 entry.
Errors in Customization	N/A	N/A	Errors occurred while the customization jobs were being generated.
Not Required	N/A	LPAR parameters are not required.	Db2 parameters are not required.

Related tasks

Creating and associating Db2 entries
You can create new Db2 entries and associate them with Db2 Log Analysis Tool.

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 Log Analysis Tool with Tools Customizer. These data sets are supplied by Db2 Log Analysis Tool, supplied by Tools Customizer, or allocated by Tools Customizer.

Db2 Log Analysis Tool 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*.SALADENU, 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 Log Analysis Tool Discover EXEC. When you customize Db2 Log Analysis Tool, 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 Log Analysis Tool, the lowest-level qualifier is SALADENU. 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 Log Analysis Tool. 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, 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 Log Analysis Tool Discover EXEC. The Db2 Log Analysis Tool Discover EXEC retrieves the metadata and values for the parameters from a previous customization of Db2 Log Analysis Tool.

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, and LPAR when you customize products on the local LPAR.

The default name of the data set is DB2TOOL.CCQ110.DATASTOR. 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 Log Analysis Tool.

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 Log Analysis Tool, 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 *hlq*.$*LPAR_name*$.*xyzvrm*, where:

- *hlq* is the value of the **Customization library qualifier** field on the **Tools Customizer Settings** panel (CCQPSET)

- *LPAR_name* is the four-character LPAR name
- *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.$MVS1$.*XYZ*410.

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:

Table 15. Data set attributes for allocating the Discover output, data store, and customization library data sets

Data set	Organization	Record format	Record length	Block size	Data set name type
Discover output data set	PO	Variable block	16383	32760	LIBRARY
Data store data set	PO	Variable block	16383	32760	LIBRARY
Product customization library	PO	Fixed block	80	32720	LIBRARY

Restrictions:

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

Chapter 26. Reference

These reference topics are designed to provide you with quick access to information about Db2 Log Analysis Tool.

Archive output data set description

The results of a detail report run are archived (saved) in an output data set, which is then available for additional processing. The format of the archived output data set is described in this topic.

The archive output data set stores two different types of records. These records are not mixed in the file; the set of records of the first type comes prior to the set of records of the second type.

The first type of records are typically stored in the CATFILE and thus are called the CATFILE records. Similarly, the second type of records are usually stored in the FRMTFILE and thus are called FRMTFILE records.

Table 16. CATFILE record description

Offset (hexadecimal)	Offset (decimal)	Size (bytes)	Description
000000	0	4	Record descriptor word (RDW)
000004	4	2	Database identifier (old, if object was dropped)
000006	6	2	Table space identifier (old, the only ID available)
000008	8	2	Table identifier (old, the only ID available)
00000A	10	2	Database identifier (new/current, only if old present)
00000C	12	2	Table space identifier (new/current, only if old present)
00000E	14	2	Table identifier (new/current, only if old present)
000010	16	8	Database name
000018	24	8	Table space name
000020	32	2	Table owner name length
000022	34	128	Table owner name
0000A2	162	2	Table name length
0000A4	164	128	Table name
000124	292	8	Edit procedure name (spaces if none)
00012C	300	1	Does any field procedure exist? Possible values are: • X'00' - No • X'01' - Yes
00012D	301	2	Number of partitions in table space

Offset (hexadecimal)	Offset (decimal)	Size (bytes)	Description
00012F	303	2	Page size of table space
000131	305	1	Type of table space. Possible values are: • L - large (the table space can be greater than 64 gigabytes) • X'40' - not large
000132	306	1	Coded character set identifier (CCSID). Possible values are: • 'E' for EBCDIC • 'A' for ASCII • 'U' for UNICODE
000133	307	10	LOGPOINT when the table was created.
00013D	317	1	Data set size in flag format. Possible values are: • X'00' - 0 GB • X'01' - 1 GB • X'02' - 2 GB • X'04' - 4 GB • X'08' - 8 GB • X'10' - 16 GB • X'20' - 32 GB • X'40' - 64 GB • X'80' - 128 GB • X'88' - 256 GB
00013E	318	2	Segment size
000140	320	1	Miscellaneous flags. Possible values are: • X'01' - table versioning present on table • X'02' - if combo mode only, means object in log-forward mode only • X'04' - object is XML • X'08' - object has DATA CAPTURE CHANGES set to on • X'10' - table space versioning exists (current version) • X'20' - table space versioned entry (old version) • X'40' - table space versioned for page size (old version) • X'80' - table space versioned for data set size (old version)

Table 16. CATFILE record description (continued)

Table 16. CATFILE record description (continued)			
Offset (hexadecimal)	**Offset (decimal)**	**Size (bytes)**	**Description**
000141	321	1	Table type from SYSTABLES
000142	322	2	Specific partition number or 0 for all partitions
00144	324	2	Current version of table from SYSTABLES
00146	326	10	If non-data sharing environment, stores the created timestamp in LRSN format

FRMTFILE record description

Each FRMTFILE record consists of a format file row header and an array of column information items. Both structures have variable length fields. A field that is contains "C" in the Size column is paired with the field before it, with the physical length of the field stored in appropriate fixed length fields before the current one. So, after the first row that has a "C" in the Size column, the Offset column no longer describe the absolute disposition, but may be used as relative disposition.

Table 17. FRMTFILE record description			
Offset (hexadecimal)	**Offset (decimal)**	**Size (bytes)**	**Description**
Format file row header			
000000	0	2	Record length
000002	2	2	Restricted
000004	4	10	Extended log record sequence number (LRSN) of this UNDO/REDO record
00000E	14	10	Extended unit of recovery identifier (URID) of this UNDO/REDO record
000018	24	4	Group number of the record
00001C	28	1	Type of detail record. Possible values are: • X'10' - current format record • X'20' - new format record (post-change) • X'40' - old format record (pre-change) • X'01' - continue record (first part) • X'02' - continue record (second part)
00001D	29	4	Db2 subsystem identifier
000021	33	2	Database identifier
000023	35	2	Table space identifier
000025	37	2	Table identifier (or partition number if relative paging)
000027	39	4	Page identifier of record within table space
00002B	43	1	Record identifier
00002C	44	8	Authorization identifier

Offset (hexadecimal)	Offset (decimal)	Size (bytes)	Description
			Table 17. FRMTFILE record description (continued)
000034	52	8	Db2 package or plan name
00003C	60	8	Job name
000044	68	8	Connection identifier
00004C	76	8	Connection type
000054	84	24	Logical unit of work
00006C	108	2	Db2 subsystem member identifier within data sharing environment (always 0 if non-data sharing environment)
00006E	110	6	Action: UPDATE, INSERT, DELETE
000074	116	1	Flags for header. Possible values are: • X'80' - LOB present in table • X'40' - entry was rolled back (aborted) • X'04' - not committed (at original end date/time)
000075	117	1	Additional flags. Possible values are: • X'20' - This entry indicates that a delete statement without a WHERE clause occurred for this table.
000076	118	1	For internal use only
000077	119	2	For internal use only
000079	121	12	Db2 correlation identifier (job name plus 4 additional bytes)
000085	133	10	Extended end LRSN of URID
00008F	143	10	RBA
000099	153	14	For internal use only
0000A7	167	2	Distributed data only; transaction identifier length
0000A9	169	C	Distributed data only; transaction identifier
0000AA	170	2	Distributed data only; end user length
0000AC	172	C	Distributed data only; end user
0000AD	173	2	Distributed data only; application or transaction name length
0000AF	175	C	Distributed data only; application or transaction name
0000B0	176	2	Distributed data only; workstation name length
0000B2	178	C	Distributed data only; workstation name

Offset (hexadecimal)	Offset (decimal)	Size (bytes)	Description
0000B3	179	2	Distributed data only; accounting string length
0000B5	181	C	Distributed data only; accounting string
Column information			
0000B6	182	2	Length of this column entry
0000B8	184	2	Column count within group number
0000BA	186	1	Column flags. Possible values are: • X'20' - float column (has display and float values) • X'10' - column data is NULL (length is 4, data is empty) • X'08' - column data is LOB • X'01' - column data not available prior to altering the table
0000BB	187	1	Column flags. Possible values are: • X'20' - column encoding scheme EBCDIC • X'40' - column encoding scheme ASCII • X'80' - column encoding scheme UNICODE
0000BC	188	1	Column flags. Possible values are: • X'02' - column data is ROWID • X'04' - column is BLOB • X'08' - used defined data type (distinct) • X'20' - column is XML
0000BD	189	1	For internal use only
0000BE	190	2	Column name length
0000C0	192	C	Column name
0000C1	193	2	Column data length
0000C3	195	C	Column data
0000C4	196	2	Column data length for hex value or inline LOB
0000C6	198	C	Column data for hex value or inline LOB

Table 17. FRMTFILE record description (continued)

What's new in previous editions

This topic summarizes significant enhancements and changes to previous editions of Db2 Log Analysis Tool documentation.

Previous updates

SC27-6559-01 edition

General Reports
The "Loaded Data" information has been enhanced to explain more options. See "Loaded data" on page 96 for details.

New section in "Quick Start" chapter
A new section has been added that explains how to generate SQL for UNDO and REDO commands. See "Quick start-generating UNDO/REDO SQL" on page 190.

Updates to Reports
Additional descriptive information has been added to help you understand several reports:

- Continuous Mode File
- Details Report
- Details Summary Report
- Extended Report
- General Report
- Summary Report

See Chapter 14, "Db2 Log Analysis Tool reports," on page 197 for more information.

Updates to examples
A new example has been added to the "Column data filter examples" section to show how to specify values for the COND keyword. A change was also made to Example 2. See "Using advanced object filters" on page 125 for more information.

Message
A message, ALAA639W, was added to the documentation. See "ALAA639W" on page 316 for more information.

SC27-6559-00 edition

Quick Start Panels
Log Analysis Tool now has two operating modes. There are classic panels which show the panels in the same familiar format as in past versions of Log Analysis Tool. Classic panels provide access to every option within Log Analysis Tool. There are now quick start panels which are designed to for quicker and easier job generation. The quick start panels are organized so you can choose what type of Log Analysis Tool activity is desired and only see the most important options that are relevant to that activity. See Chapter 13, "Using the quick start panels," on page 183 for more information.

Continuous mode restart assist
Continuous mode restarts have been made more automatic, if so chosen. The new field "Restart assist on continuous mode" allows you to request that the product track where it failed in continuous mode. This allows the product to automatically restore continuous mode files on a restart, which eliminates the previous releases issues requiring manual intervention for such. See "Set user settings" on page 67 for more information.

Member name on reports instead of ID
In the past, the product has reported the member id of the Datasharing Member or 0000 for non-Datasharing subsystems when reporting information found in the header portion of the log record. This line item adds a user settings parameter to request the Datasharing Member name for Datasharing Subsystems or SSID when reporting on non-Datasharing subsystems instead of reporting the ID. See "Set user settings" on page 67 for more information.

Save Generated Job as a Profile
In previous versions of Log Analysis Tool, the only method to create a profile was to enter the profile panel and create a new profile or copy from an existing one. If you created a Log Analysis Tool job outside of the profile panel, there was no way to save the work. Now, after you have generated a job and edited and/or submitted it, you are asked if you want to save all the specified options as a profile. This default prompt can be turned off in the user settings. See "Set user settings" on page 67 for more information.

Storage activity report

A new report is automatically generated by each load module in a Log Analysis Tool general and/or detail report run showing storage activity by 24, 31, and 64 bit storage. Though this report is primarily intended for internal/diagnostic use, it may prove beneficial to some customers. The report is included in DD INFOM output. See "Storage Report" on page 219 for more information.

SYSCOPY report

A new report is automatically generated within load module ALAGEN1 in a Log Analysis Tool general report run showing SYSCOPY entries for DB2 objects in the result set. Though this report is primarily intended for internal/diagnostic usage, You may find it beneficial. See "SYSCOPY Report" on page 222 for more information.

64 bit mode switch to work file mode

Log Analysis Tool now checks on the value of MEMLIMIT (that is, the maximum allowed 64 bit storage value) and will switch over to work file mode in case it is about to be exceeded. This means the job can complete successfully without a restart, and no further user action is required. See "Set user settings" on page 67 for more information.

GMT Offset Override

In the past, if Log Analysis Tool used the current system GMT offset for log reading and reporting. In different places around the world, system programmers modify this value due to a time change when entering or exiting Daylight Saving Time. This caused confusion when specifying time ranges and reading reports that were for activity that occurred before a time change but were being reported after the time change. Using the current system GMT offset is still the default for Log Analysis Tool. However, a parameter has been added to the Classic ISPF menus to allow for overriding the GMT offset with a specified value instead of the current system value. See "How to create a General Report" on page 85 for more information.

Additional diagnostics for certain error conditions

If certain error conditions are encountered, the first piece of documentation that is requested other than the job log is the results of a SYSCOPY query. Whenever one of the following error conditions is encountered: ALAA440E, ALA447E, ALAA073, ALAA107, Log Analysis Tool will issue the query and print the results to INFOM DD before terminating with the error message. This will expedite the resolution process. See "SYSCOPY Query" on page 221 for more information.

Table Versioning Report

Table Versioning can lead to strange errors and/or incorrect results within Log Analysis Tool. Support for Table Versioning is on-going. Log Analysis Tool produces a report of the contents of the table versioning repository when Table Versioning is encountered. This report is mainly to aid in support of PMR resolution when Table Versioning is involved by getting the information earlier. You can also use the report to spot any anomalies or errors you may have introduced when manually entering information into the table versioning repository. See " Table Versioning Report" on page 223 for more information.

Improved Catalog Audit Report

The Catalog Audit Report no longer requires SQL to be generated in order to produce the report. Any jobs that you set up in Log Analysis Tool V3.4 to produce the Catalog Audit Report may need to be recreated due to the architectural changes. The report is also now produced from ALADTL3 instead of ALADTL4. See "Catalog Audit Report" on page 198 for more information.

Control File Report

There is a new report that displays the information from the Control File for the given SSID in the INFOM DD automatically. This report can be supplied to the support personnel with the complete job log during the first steps of collecting documentation to expedite resolving issues. See "Control File Report" on page 204 for more information.

4. Filters for DB2 objects now use complete DB2 wildcarding

When entering filter names for DB2 objects (that is, table names, tablespace names, database names, miscellaneous filters like authid, plan name, etc.), wildcarding has always been present, but not complete by DB2 standards. The use of '%' and '_' wildcards is now supported for filtering purposes.

23. Allow Log Analysis Jobs to Run on Different SSIDs

Currently, Log Analysis Tool jobs can only be run on the SSID that was specified when the JCL was generated because the SSID is stored in the MODEFILE. Users have tried changing the SSID in the JCL

and other minor modifications with the intent of running the job on another DB2 subsystem. This did not work and would result in an error condition. In the next release, the user will have the ability to edit the SSID within the JCL and the job will then run on the new subsystem. Log Analysis Tool will see the change and update the MODEFILE without causing an error condition. The SSID will still need to be defined in the control file for proper usage throughout the product.

24. Fixed External Format File

Within Log Analysis Tool, a LOADFILE can be generated in DB2 load format, one can be generated for loading into an AUDIT table, or one can be generated that has the data in external format with customizable delimiters. None of these formats meet the requirements for a customer that needs to load data into another DATABASE Management System. A new feature is being added to Log Analysis Tool to create a fixed column length output file in external format with load cards so the Details Report data can be loaded into their other DBMS.

Extended use of SYSIBM.SYSLGRNX

In previous releases of Log Analysis Tool, SYSLGRNX data was used to remove certain archive and/or active logs from processing, if no activity was found to be contained within them for filtered objects. Subsequently, that SYSLGRNX data was not used. In the 3.5 release, Log Analysis Tool can make more use of it during the detail report log reads. This data has now been carried into the log reading process, where ranges of log activity can be eliminated if no activity is found for any particular object. This is a potential performance improvement requiring no user actions. By bypassing such ranges of log activity, fewer control intervals need be parsed thereby reducing overall CPU. If you do not explicitly bypass SYSLGRNX, this new feature will be used if any given run internally determines it can use it.

TCz installation now discovers DB2 mode and level

During installation of Log Analysis Tool, TCz discovers the values of Db2 mode and level. These values can only be resolved during the DISCOVER process of TCz. In the past, these values were required as user input. No user interaction is required for this feature.

Extension of LOADLIB specifications

The control file now allows a maximum of 10 loadlibs to specify for any job run. These libraries are primarily utilized in STEPLIB for any batch jobs generated by the product. The libraries are specified either via the 'Setup' option within the product's panels or via TCz at installation time.

Expert Options Table increased

A table in the Expert Option modules was increased to handle 4096 partitions. No user interaction is required for this feature.

Message Library in JCL

Log Analysis Tool will now write out the message library to the JCL using ALAERR DD. This prevents you not getting descriptive messages if you entered the message library incorrectly or left it blank in the control file. The message library DSN comes from the CLIST that is invoked for using the product ISPF panels. If the ALAERR DD is not present in the JCL, then Log Analysis Tool will attempt to dynamically allocate the message library from the message library DSN in the control file.

BSDS Copy #1 Mismatch Detection

Log Analysis Tool requests that you input DATASET names for the Bootstrap DATASET #01 and #02. If you enter the DATASET name for #02 in the #01 field and the #01 in the #02 field, this could lead to problems during the log reading phase. Log Analysis Tool identifies this condition warning message that is generated when reading the BSDS for log file names.

Support for Implicitly Hidden Columns

In previous releases of Log Analysis Tool, there was a constraint that IMPLICITLY HIDDEN columns are not supported within SQL processes. This constraint has been removed. Log Analysis Tool automatically lists all column names in the SQL statement when needed for inserts.

Include the END LRSN before a COMMIT in UNDO/REDO SQL when "Commit Scope" is set to 'UOW'

When Log Analysis Tool generates UNDO or REDO SQL, you have the option to set the commit scope to 'UOW'. Committing by unit of work sorts all the SQL into unit of work blocks and commits each block immediately after it is executed. Now, when the commit scope is set to 'UOW', there will be a comment immediately before the unit of work COMMIT statement that identifies the END LRSN of that unit of work.

Chapter 27. Customization reference

This section provides additional information about customization of Db2 Log Analysis Tool.

Product information required to run Tools Customizer's Discover EXEC

You can use the Tools Customizer Discover EXEC to discover and store product information from a previous or current customization of Log Analysis Tool.

Discover customized product information

On the Discover Customized Product Information panel, either accept the default values for the following input fields that Tools Customizer generates, or replace the default values with your own values:

Table 18. Discover Customized Product Information panel parameters

Parameter	Description	Default value
Discover EXEC library	The fully qualified data set name that contains the product Discover EXEC.	The default value is the product metadata library that you specified on the Specify the Product to Customize panel.
Discover EXEC name	The name of the Discover EXEC.	ALADISC
Discover output data set	The name of the data set for the output from the product Discover EXEC.	The default value is the same data set you specified in option **0**, User Settings, from the Tools Customizer main menu.

On the Discover Customized Product Information panel, under the Information for Discover EXEC section, provide the following parameter values before running the Discover command:

Table 19. Information for Discover EXEC section parameters

Parameter	Description	Default value
New ALA Load library	The load library name for the new release of Log Analysis Tool. For example, if this release is V3.5, the library name might be ALA.SALALOAD (using a generic load library name) or ALA.V350.SALALOAD. Enter this value without quotes.	ALA.V350.SALALOAD
New ALA ISPM library.	The message library name for the new release of Log Analysis Tool. For example, if this release is V3.5, the library name might be ALA.SALAMENU (using a generic message library name) or ALA.V350.SALAMENU (using a release specific library name).	ALA.V350.SALAMENU

Table 19. Information for Discover EXEC section parameters (continued)

Parameter	Description	Default value
Control file	The name of the Log Analysis Tool control file to use as input to the discover process. This can represent the control file of any release, including the new release if it has been previously populated with data. However, most likely you will want to specify your previous release's control file. Enter this value without quotes.	ALA.V350.CONTROL
ALA CLIST library	The name of the Log Analysis Tool CLIST library to use as input to the discover process. This can represent the CLIST library of any release, including the new release if it has been previously configured. However, most likely you will want to specify your previous release's CLIST library.	ALA.V350.SALACLST
Previous BIND JCL library and member	The library and member name to use as input to the discover process, where JCL can be found that was used previously to bind the product. If this is the first release using TCz for configuration, then that library should come from the ALA SAMPLIB and the member name should be ALABIND. For example, ALA.ALASAMP(ALABIND). If you want to discover from a past bind using TCz, then specify any bind job member in that library. For example, it could be: XYZ.$LP01$.ALA340(A3BINDAA).	ALA.V350.SALASAMP(ALABIND)

Product parameters, steps and tasks for Log Analysis Tool

If you do not have a previous version of Log Analysis Tool installed on which to use the Discover EXEC, or if you wish to manually provide parameters for a new customization of Log Analysis Tool, you may skip running the Discover EXEC. Instead, you can manually provide the information required to customize Log Analysis Tool on the Product Parameters panel.

Required parameters, tasks and steps are preselected with a slash (/). Select the optional tasks and steps with a slash (/), and change or accept the default values that are displayed for those steps.

Parameter values that are discovered and parameter values that are specified manually are saved in the *hlq*.DATASTOR data set. If the product that you want to customize already exists in the data store data set, Tools Customizer issues a warning before existing values are replaced.

For a list of read-only product parameters, see "Worksheets: Gathering parameter values" on page 24. This worksheet contains information that was provided on other panels, by Tools Customizer, or by the Log Analysis Tool metadata data set.

⚠️ **Attention:** If you have edited the Db2 parameters before editing the product parameters, some of the product parameters might be moved from their original sections to the Required parameters section. To avoid this situation, create and edit Db2 entries first.

Required parameters for all customizations

These parameters are required for all customizations.

Parameter	Required?	Description	Discovered?	Default value
Table 20. Required parameters section				
Control file name	Yes	The name of the control file that will be used for this installation or release of Log Analysis Tool. The control file contains configuration information for each Db2 subsystem against which the product can run.	No	ALA.CONTROL
Qualifier for unqualified tables	Yes	The qualifier for unqualified tables. This qualifier is used in the bind of Log Analysis Tool SQL within the QUALIFIER parameter.	Yes	SYSTOOLS
CLIST for ALA	Yes	The CLIST library name for Log Analysis Tool.	No	ALA.SALACLST
ALA load library	Yes	The name of the Log Analysis Tool load library.	No	ALA.SALALOAD
FEC load library	Yes	The name of the FEC load library.	Yes	FEC.SALALOAD
ALA ISPMLIB	Yes	The name of the Log Analysis Tool ISPMLIB.	Yes	ALA.SALAMENU

ALA control file

This task creates the control file if it does not exist or if you are creating a new control file for a new release. The control file contains specific information about each Db2 subsystem where Log Analysis Tool might run.

Step or parameters	Required?	Description	Discovered?	Default value
Table 21. ALA control file task				
ALA control file creation	No	Creates the control file if it does not exist or if you are creating a new control file for a new release. The control file contains specific information about each Db2 subsystem where Log Analysis Tool might run.	–	Selected
Volume serial number for the control file	No	The volume serial number (VOLSER) to be used for the control file creation. May be left as blanks for SMS shops.	No	blank

ALA Fast Apply DDL

This task creates the objects that are required to run Fast Apply on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation.

Step or parameters	Required?	Description	Discovered?	Default value
Table 22. ALA Fast Apply DDL task				
ALA Fast Apply DDL	No	Creates the objects that are required to run Fast Apply on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation.	–	Selected
ALA database name	Yes	The database name used to store the Log Analysis Tool objects.	No	ALADB01
ALA table space name	Yes	The tablespace name used to store the Log Analysis Tool objects.	No	ALATS01

ALA profiles DDL

This task creates the objects that are required to use profiles on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation. Before this DDL can be run, the Log Analysis Tool Fast Apply DDL must have been run, in this installation or a previous installation.

Step or parameters	Required?	Description	Discovered?	Default value
Table 23. ALA profiles DDL task				
ALA profiles DDL	No	Creates the objects that are required to use profiles on a Db2 subsystem only if those objects were not previously created in this installation or a previous installation. Before this DDL can be run, the Log Analysis Tool Fast Apply DDL must have been run, in this installation or a previous installation.	–	Selected
ALA database name	Yes	The database name used to store the Log Analysis Tool objects.	No	ALADB01
ALA profiles table space name	Yes	The table space name that is used to store the Log Analysis Tool profiles table.	No	ALATS02
ALA profile attributes table space name	Yes	The table space name that is used to store the Log Analysis Tool profile attributes table.	No	ALATS03

ALA bind SQL

This task binds the ALA SQL.

Table 24. ALA bind SQL task

Step or parameters	Required?	Description	Discovered?	Default value
ALA bind	Yes	Binds the ALA SQL.	–	Selected
DBRMLIB for ALA	Yes	The DBRM library name for Log Analysis Tool.	Yes	ALA.SALADBRM
User ID for ALA bind	Yes	The user ID for the owner of the plan. The user ID value is used during the bind on the OWNER(userid) parameter.	Yes	USERID
Collection name for ALA bind	Yes	The collection name to be used within the Log Analysis Tool bind.	Yes	ALACOLL

Edit startup CLIST

This task edits the startup CLIST to provide proper dsnames for your site.

Table 25. Edit startup CLIST task

Step or parameters	Required?	Description	Discovered?	Default value
Edit startup CLIST	Yes	Edits the startup CLIST to provide proper dsnames for your site.	–	Selected
ALA ISPPLIB	Yes	The name of the Log Analysis Tool ISPPLIB.	No	ALA.SALAISPP
FEC ISPPLIB	Yes	The name of the FEC ISPPLIB.	No	FEC.SALAISPP
FEC ISPMLIB	Yes	The name of the FEC ISPMLIB.	No	FEC.SALAMENU
ALA ISPSLIB	Yes	The name of the Log Analysis Tool ISPSLIB.	No	ALA.SALAISPS

ALA configure Db2

This task configures each Db2 subsystem within the control file.

Table 26. ALA configure Db2 task				
Step or parameters	**Required?**	**Description**	**Discovered?**	**Default value**
ALA configure Db2	No	This step configures each Db2 subsystem within the control file. It is optional in the sense that you can either perform this task here for all your Db2 subsystems (or any subset thereof), or you can perform this same task using option #11 ("Setup") from the Log Analysis Tool main menu to configure each Db2 subsystem individually, as needed. A Db2 subsystem must be configured using one of these methods before it can be used within Log Analysis Tool.	–	Not selected

ALA ALTER DDL

This task alters tables created in previous releases as part of the 'Load data into Db2' process or quiet time loading process.

Table 27. ALA ALTER DDL task				
Step or parameters	**Required?**	**Description**	**Discovered?**	**Default value**
ALA ALTER DDL	No	Alters tables created in previous releases as part of the 'Load data into Db2' process or quiet time loading process.	–	Not selected

Launchpad

This two-part task adds Log Analysis Tool to the Db2 Administration Tool Launchpad.

Table 28. Launchpad task				
Step or parameters	**Required?**	**Description**	**Discovered?**	**Default value**
Launchpad	No	Part 1 modifies the REXX EXEC to add Log Analysis Tool to the Db2 Admin Launchpad. Part 2 runs that REXX EXEC and adds Log Analysis Tool to that Launchpad.	–	Not selected

Db2 Parameters

Use this panel to specify or change values for the Db2 parameters. All parameters are required.

Table 29. Db2 Parameters section				
Step or parameters	**Required?**	**Description**	**Discovered?**	**Default value**
Mode	Yes	The mode in which the Db2 subsystem is running. The following values are valid: • CM is compatibility mode on all listed Db2 versions except Db2 10. • CM8 is conversion mode from Db2 V8 on Db2 10. • CM9 is conversion mode from Db2 Version 9.1 on Db2 10. • NFM is new-function mode on all listed Db2 versions.	No	NFM
Level number	Yes	The version, release, and modification level of the Db2 subsystem. The following values are valid: • 910 is valid only for CM or NFM. • 101 is valid only for CM8, CM9 or NFM. • 111 is valid only for CM or NFM. • 121 is valid only for CM or NFM.	No	*blank*
Load library	Yes	The data set name of the Db2 load library.	Yes	DSN. SDSNLOAD
Run library	Yes	The data set name of the Db2 run library.	Yes	DSN.RUNLIB. LOAD
Exit library	Yes	The data set name of the Db2 exit library.	Yes	DSN.SDSNEXIT
Bootstrap data set	Yes	The name of the Db2 bootstrap data set.	Yes	DSN.SDSNBSDS
Name of the 4 KB bufferpool	Yes	The name of the 4 KB bufferpool to be used for customization. The value must be 8 characters or less.	No	BP0
Plan name for the DSNTEP2 utility	Yes	The plan name for the DSNTEP2 utility. The value must be 8 characters or less.	No	DSNTEP2

Table 29. Db2 Parameters section (continued)				
Step or parameters	**Required?**	**Description**	**Discovered?**	**Default value**
Storage group name	Yes	The name of the storage group that will be used for creating Db2 objects for customization. The value must be 128 characters or less.	No	SYSTOOLS
Plan name for ALA bind	Yes	The plan name to be used on the Log Analysis Tool bind.	Yes	ALASQLP
ZPARM module name	Yes	The name of the ZPARM module for this specific Db2 subsystem. This module typically resides in the Db2 SDSNEXIT library, and the name is typically in ssidPARM format. For example, if your Db2 SSID is D91A, the name of the ZPARM module might be D91APARM.	Yes	ZPARM
Use of archive logs 1 and 2	Yes	This field determines how you want Log Analysis Tool to process your archive logs. The field combines the settings for both ARCHLOG1 and ARCHLOG2. The first character is for ARCHLOG1, the second for ARCHLOG2. Only the following values are permissible: **YN** Use ARCHLOG1 only, ARCHLOG2 is ignored. **NY** Use ARCHLOG2 only, ARCHLOG 1 is ignored. **PN** ARCHLOG1 has priority over ARCHLOG2. **NP** ARCHLOG2 has priority over ARCHLOG1. **NN** There are no special considerations for either ARCHLOG1 or ARCHLOG2.	Yes	NN

Table 29. Db2 Parameters section (continued)				
Step or parameters	**Required?**	**Description**	**Discovered?**	**Default value**
Active log priority	Yes	Specify Y to indicate that Log Analysis Tool will process the active log when data resides on both active and archive logs.	Yes	Y

Jobs generated for Log Analysis Tool customization

This topic lists the customization jobs generated by Tools Customizer for Log Analysis Tool.

The customization jobs are grouped by the job sequence number. Run the jobs in the sequence in which they are displayed for all Db2 entries.

The jobs use an 8-character maximum member naming convention that follows the format of ssXXXXdd:

where:

- ss - The job sequence number, which is an alphabetic character (A - Z) followed by a numeric character (0 - 9). For example, a job sequence number is A0, A1, ..., Z9.
- XXXX - Static characters identifying the job name and its purpose.
- dd - one to two alphanumeric characters (AA - 99) that Tools Customizer assigns to identify a Db2 entry.

Table 30. Jobs generated for Db2 Log Analysis Tool customization		
Jobname	**Template**	**Description**
A0CNTL1	ALACNTL1	Log Analysis Tool control file JCL - This JCL allocates the control file for Log Analysis Tool. It should be run just once per installation of the product, not per Db2 subsystem. A new control file should be created for a new release of Log Analysis Tool, but if you want to use a previous release's control file this step can be eliminated completely. Using a previous release's control file, however, prevents proper functioning of both releases concurrently as certain information in a single control file cannot be valid for two releases per Db2 subsystem, for example, the plan name used for both releases cannot be the same under a single Db2 subsystem.
A1FAS1dd	ALAFAS1	Log Analysis Tool FAS DDL - This JCL creates the necessary Db2 objects for the Log Analysis Tool feature for Fast Apply SQL. Run this job for every subsystem you plan to bind against. These objects can be accessed by all users of this product, but the granting of execute privileges on the ALA plan will allow that access.
		If you have run this job for a previous release of Log Analysis Tool, you must not run it again, the required objects will already have been created. However, this job must be run for any Db2 subsystem where these objects have not yet been created.

Jobname	Template	Description
		Table 30. Jobs generated for Db2 Log Analysis Tool customization (continued)
A2PROFdd	ALAPROF1	Log Analysis Tool Profiles DDL - This JCL creates the necessary Db2 objects for the Log Analysis Tool feature for profiles. Run this job for every subsystem you plan to bind against. These objects can be accessed by all users of this product, but the granting of execute privileges on the ALA plan will allow that access.
		If you have run this job for a previous release of Log Analysis Tool, you must not run it again, the required objects will already have been created. However, this job must be run for any Db2 subsystem where these objects have not yet been created.
		This job assumes that the Log Analysis Tool Fast Apply SQL job has already been run and the database/tablespace names used therein will also be used here. If the database/tablespace specifications were changed in the Log Analysis Tool Fast Apply SQL job, they should be changed here as well. The Log Analysis Tool Fast Apply SQL job does not have to have been run for this release, as long as it was previously run for any release of this product. There should be no reason to create a separate database or table space for profiles (for the ones specified for Fast Apply DDL in the Log Analysis Tool Fast Apply SQL job), but if that is what you want to do, you must create them outside of this product and reflect their names herein.
A3BINDdd	ALABIND1	Log Analysis Tool Bind SQL - This JCL binds the necessary dataset for Log Analysis Tool. This job should be run for every Db2 subsystem where Log Analysis Tool will be run.
A4CLST1	ALACLST1	This JCL creates the necessary start-up CLIST for Log Analysis Tool. This job should run just once per installation of Log Analysis Tool, not per Db2 subsystem.
A5LOADdd	ALALOAD1	Log Analysis Tool configure Db2 - This JCL loads the control file for Log Analysis Tool. It should be run once per Db2 subsystem that Log Analysis Tool will operate on.

Jobname	Template	Description
		Table 30. Jobs generated for Db2 Log Analysis Tool customization (continued)
A6LT1dd	ALALT1	Log Analysis Tool ALTER - This JCL allows ALTERs against previously created tables used within the Log Analysis Tool 'Load details into Db2' process. These ALTERs compensate for the new extended RBA/LRSN formats introduced by Db2 V11. It is not necessary to use this JCL when creating new tables in this release of Log Analysis Tool; it is only necessary if the tables being loaded into were created in a previous release, and those tables use header fields LAT_URID or LAT_LRSN. Those headers fields have been extended to the 10-byte format in this release, and therefore require these ALTERs. Additionally, this JCL can update the rows in those tables to reflect the new extended RBA/LRSN formats. It will effectively pad the fields with zeros as appropriate. **Important:** This JCL is optional. It need only be run under the conditions explained previously. This JCL is a template for any tables affected per those conditions below, and must be run for any such table. It may be appropriate for end users to run this JCL rather than the installer as the installer may not be aware of such affected tables. It is only important that the end users have access to this template so that they may run it prior to attempts to update such tables from this release.
A7ADBI	ALAADBI	This job creates the Db2 Administration Launchpad EXEC in your CLIST library.
A8ADB2	ALAADB2	This job executes the Db2 Administration Launchpad EXEC, which adds Log Analysis Tool to the Launchpad.

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

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

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.

Programming interface information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Db2 Log Analysis Tool.

This publication documents information that is NOT intended to be used as Programming Interfaces of Db2 Log Analysis Tool.

This publication primarily documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Db2 Log Analysis Tool.

This publication also documents information that is NOT intended to be used as Programming Interfaces of Db2 Log Analysis Tool. This information is identified where it occurs by an introductory statement to a topic or section.

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of Db2 Log Analysis Tool.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Db2 Log Analysis Tool. This information is identified where it occurs by an introductory statement to a topic or section.

Trademarks

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

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

X

XC 125
XML
    one-phase 247
    two phase 247
XML data, viewing 163

Z

ZPARM 81

IBM ®

Product Number: 5655-T56