CA Datacom®

Release Notes for z/VSE

Version 12.0, First Edition

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA Datacom® CICS Services
- CA Datacom® Datadictionary™ (Datadictionary)
- CA Dataquery™ for CA Datacom® (CA Dataquery)
- CA Datacom® DL1 Transparency
- CA Datacom® Server
- CA Datacom® SQL (SQL)
- CA Datacom® TOTAL Transparency
- CA Datacom® VSAM Transparency
- CA Dynam®/D Disk Management for z/VSE (CA Dynam/D for z/VSE)
- CA Ideal™ for CA Datacom® (CA Ideal)
- CA IPC

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction

This guide provides an overview of the changes in Version 12 of the CA Datacom products for z/VSE and information about upgrading to Version 12.

Intended Audience: This guide is for users who are upgrading from r11 (SP4 or later) to Version 12 on z/VSE.

Chapter 2: Discontinued at Version 12.0 for z/VSE

While each CA Datacom version delivers a number of new features and functions, it is also at the version boundary that CA will announce the discontinuance of features or functions that no longer delivers significant value.

The following feature has been discontinued for CA Datacom Version 12.0 for z/VSE.

This section contains the following topics:

CA Datacom Guest Machine Option

The Guest Machine Option (MGSO) provided the z/VSE user the ability to allow a user address space (program) on one z/VSE system to access the CA Datacom database address space (MUF) running on a different z/VSE system under a z/VM environment.

This usage of this functionality requires underlying CA technology that is known as CA IUCV. The usage of the MGSO functionality has declined significantly and is no longer in use at most z/VSE sites.

Chapter 3: New Features

CA Datacom Version 12 is a major release level that includes a number of new features and functions. Some key features are designed to lower your Total Cost of Ownership (TCO). Other features are designed to provide a more intuitive approach to managing a CA Datacom environment to assist you in your efforts to train new personnel.

CA Datacom/DB Enhancements

The following are new features and functions specifically for the CA Datacom/DB z/VSE environment.

Add and Delete Keys Faster

You can add or delete a key faster than in previous releases, because enhancements let you change table definitions in non-SQL applications more quickly. The ability to add and delete keys faster helps users meet 24x7 operational requirements. Adding and deleting keys faster substantially reduces outages due to time lost in adding another key to a table or removing a key from a table.

To use this feature, databases must be executing with the Directory (CXX) format indicator that is specified as FORMAT 2. The CXX format indicator is a new column that is printed in the database section of the CXX report. It indicates whether you want databases to remain compatible with r11 (FORMAT 1) or be fully compatible with Version 12 (FORMAT 2).

Note: For more information, see the Add and Delete Keys Faster topic in the *CA Datacom/DB Database and System Administration Guide*.

Multi-Dataset Index

The Multi-Dataset Index feature allows you to allocate unique index datasets for one or more key definitions (key IDs) in a given database. For applications with heavy index processing limited to a few key IDs, this allows you to optimize performance by using the Memory Resident Data Facility (MRDF) to selectively maintain the high-use key IDs in memory while allowing low-use key IDs to be serviced by the normal buffers and DASD. Also, if an index needs to be re-built, this system task can be focused on the affected index dataset which limits the downtime for production applications.

Note: For more information, see the Multi-Dataset Index chapter in the *CA Datacom/DB Database and System Administration Guide*.

On-Demand Dynamic Extend

It is preferable to be able to anticipate needed extends and to request them during off-hours instead of during a peak processing time in the MUF. It is also useful if you can override the following three current choices for dynamic extend:

- Whether dynamic allocation is to be requested
- Whether the type is track or cylinder
- The number of the type

The on-demand dynamic extend feature is designed to provide maximum flexibility. It is supported as a console-like command. Being console-like, the command can be issued as follows:

- By using the DBUTLTY COMM function with OPTION=CONSOLE
- By issuing it as a true console command
- By issuing it using any SQL tool that can insert a record (including non-mainframe)

The on-demand dynamic extend feature occurs in the MUF address space. It therefore must be enabled, and a task area must be available. The on-demand dynamic extend can occur with any other normal MUF processing, for example a user application, processing the area being extended.

Note: For more information, see the DYNAMIC-EXTEND console command in the *CA Datacom/DB Database and System Administration Guide*.

DBUTLTY Dynamic Extend

When executing the DBUTLTY LOAD function, if the input data does not fit within the existing extents of the data area being loaded, an enhancement to the LOAD function attempts to dynamically extend the existing extents, with restrictions. Details of the dynamic extend match those that would occur in the Multi-User Facility (MUF) during ADDIT processing.

Additionally, a dynamic extend is attempted when executing the DBUTLTY LOAD and RETIX functions, if the index that is needed does not fit within the existing extents of the Index Area (IXX) or Index Areas. The RECOVERY function supports dynamic extend for both the Index and data areas.

In the cases just described, the Datadictionary attributes for dynamic extend are honored, and the DBUTLTY functions involved do not have independent keywords or options. These attributes, as during ADDIT processing, are subject to the many rules of the operating system.

There are restrictions that, if not met, prevent dynamic allocation from being attempted. The restrictions specific to the DBUTLTY functions are as follows:

- For the RETIX and LOAD functions, the SORT= keyword must be specified, and with a value other than zero. Only the code path using a SORT= contains the option to dynamic extend. There is no message generated if this requirement is not met.
- Every table in the database must have been defined with the Datadictionary attribute RECOVER YES. There is no message generated if this requirement is not met.
- For the RETIX and RECOVERY functions, the area must have been loaded with YES specified for the URI option. For the LOAD function, YES must have been specified for the URI option or allowed to default to YES for URI. There is no message generated if this requirement is not met.

Reduce Compression Overhead

The Reduce Compression Overhead enhances performance by providing a new search algorithm for the benefit of multi-table areas and areas with tables that use compression. This feature provides an optional search algorithm named DATAFS.

DATAFS is chosen by using a new option in DBUTLTY CXXMAINT ALTER. This global option turns the feature on and off for all areas in a given CXX.

Note: For more information, see the CXXMAINT ALTER topic in the *CA Datacom/DB DBUTLTY Reference Guide*.

Performance Memory Manager

Memory management within the MUF is an internal process with low user visibility that was sometimes seen as a big item, particularly in SQL execution. It has been changed in the following ways:

- It executes faster.
- It is available to more types of activity in the MUF.
- Its information is available to user query by having two Dynamic System Tables.

MUF Startup Options

These MUF startup options are new for Version 12.

DYNAMIC_EXTEND_MSGS

Use the DYNAMIC_EXTEND_MSGS startup option and console-like command to select either r11-style dynamic extend messages (DB01701 and DB01703) or the new style messages for Version 12. If you have automated actions dependent on r11 message numbers and style, you can set those messages to print as in r11. The new message numbers and style in Version 12 provide more information. When using the new style, you can get the counts as cylinders (smaller numbers) or tracks. Any counts, however, that are not in full cylinders are always given in tracks.

LIMITED_INSTALL_MUF

The LIMITED_INSTALL_MUF startup option blocks options that can cause known conflicts. A normal MUF execution automatically performs many actions for efficiency and ease of use. However, there are times when some actions could be unwanted because of conflicts they could cause. For example, in an upgrade to a new release, LIMITED_INSTALL_MUF could be used to avoid conflicts and allow user actions that require a MUF to be enabled but not ready for production work. Normal startup activities include MUF requiring and automatically opening the temporary CBS index. However, if the CXX was renamed (INIT CXXNAME=) while MUF was down, a conflict could be caused if the CBS database is not VIRTUAL. LIMITED_INSTALL_MUF specified as YES could prevent that conflict by not automatically opening the CBS temporary index.

SQL_KATAKANA_CODE_PAGE

The SQL_KATAKANA_CODE_PAGE startup option and console-like command specifies which code-page CA Datacom should use to interpret any incoming Katakana characters in SQL statement syntax. This option is relevant only when Katakana language usage has been activated using the DBUTLTY CXXMAINT function and default code page 290 is incompatible with the terminals or software in use.

X_DUMP_SIP_ERROR

The X_DUMP_SIP_ERROR startup option is a debugging tool that is occasionally required to assist CA Support in the analysis of Multi-User startup option errors. Use X_DUMP_SIP_ERROR only if Technical Support requests it.

X_MSB_MSGS

Use the X_MSB_MSGS startup option and console-like command to monitor MUF activity for debugging purposes. You can set the option during the enabling of the MUF or at any time later using the console facility. Every MUF performs a check approximately every 2 minutes to validate users and set the Dynamic System Table MUF_RATES values for the previous 2 minutes. If the X_MSB_MSGS option is set to YES, additional processing issues zero to four messages, based on MUF activity.

X_RESTART_DETAIL

Use the X_RESTART_DETAIL startup option to change the way restart reprocessing works for this release as compared to the previous release. You can also use it to change what is reported in the restart reprocessing report.

For more information, see the Modifying Multi-User Facility Startup Options and the Maintenance using Console Commands topics in the *CA Datacom/DB Database and System Administration Guide*.

Console and Console-Like Commands

The two basic changes involving console commands as follows:

- For most of the console and console-like commands, the same command can be issued from all of the following:
  - The true console, using MODIFY in z/OS and MSG in z/VSE
  - SQL, using the SQL_CONSOLE (SQX) Dynamic System Table
  - MUF startup options, with some exceptions
  - DBUTLTY, using the function COMM OPTION=CONSOLE,OPTION2=
- The SQL_CONSOLE (SQX) Dynamic System Table has been created. This table provides a new method of issuing maintenance commands. You can now do inserts with a character string that is formatted by all the console commands. This method replaces SQL maintenance to all other tables where it previously existed.

The following new console commands are in addition to the MUF startup options that have console-like commands.

SQL_CODE_ERROR

SQL_CODE_ERROR modifies what is dumped to the PXXSQL file when the specified SQLCODE occurs. On certain occasions, CA Support may request that you use SQL_CODE_ERROR. SQL_CODE_ERROR modifies what is dumped to the PXXSQL file when the specified SQLCODE occurs.

SQL_LRU_STATEMENT_CACHE

SQL_LRU_STATEMENT_CACHE speeds processing by defining a cache to hold SQL statements in memory for reuse.

SQL_OPTIMIZATION_LEVEL

SQL_OPTIMIZATION_LEVEL specifies the level of SQL optimization that the SQL table join optimizer uses. It replaces DIAGOPTION 9,1,ON, which is no longer supported. We recommend a level of 0, the default, as the best level to give the broadest benefits during table join optimization.

SQL_SQUARE_BRACKETS

SQL_SQUARE_BRACKETS can be used to direct SQL to recognize alternative code point values for square bracket characters.

SQL_STATE_ERROR

SQL_STATE_ERROR modifies what is dumped to the PXXSQL file when the specified SQLSTATE occurs.

Note: For more information, see the Maintenance using Console Commands topic in the *CA Datacom/DB Database and System Administration Guide*.

DBUTLTY

The new DBUTLTY VERINDEX function ensures the integrity of the CA Datacom/DB data rows and their corresponding keys in the index. This function is designed to use when you question the integrity of the index with the data, but the table cannot be taken offline. VERINDEX executes while the table is open for processing.

Note: For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

DBUTLTY Functionality Provided for Online Reorganization Capabilities

The following are new DBUTLTY functions have been provided as part of the new online reorganization process. The online reorganization process is covered within the larger data reorganization discussion in the *CA Datacom/DB Data Reorganization Green Book* which can be found at:

www.ca.com/greenbooks
https://support.ca.com/irj/portal/anonymous/phpdocs?filePath=0/common/greenbooks.html

DATANE Native Efficiency Report

The main function of the Native Efficiency Report (NER) is to build a simple report that shows the sequential access efficiency for a given data table. You can use this report to analyze the sequential effectiveness (native key order) of a data table and whether data reorganization is needed.

The NER requires the MUF to be up and the selected data table to be available for processing. The NER uses highly efficient read-only index commands to limit the effect on other processing in the MUF.

For detailed instructions about executing the DBUTLTY DATANE function, see Appendix A: Implementing and Using DBUTLTY REPORT=DATANE in the *CA Datacom/DB Data Reorganization Green Book*.

DATASP Data Area Space Report

The main function of the Data Area Space Report (DATASP) is to build a simple report that shows the detailed space usage in a selected data area. This space usage provides insight into the number of rows per block, bytes per block, compression efficiency and so on. The DATASP report provides a simple data area map that can help you determine:

- available space in a data area
- amount of compression
- an estimate of rows that can be added (based on calculated average row size and available space)

The DATASP runs outside of the MUF and does not require that the MUF availabity. The DATASP report uses a highly efficient I/O process that has no effect on the MUF processing.

For detailed instructions about executing the DBUTLTY DATASP function, see Appendix B: Implementing and Using DBUTLTY REPORT=DATASP in the *CA Datacom/DB Data Reorganization Green Book*.

OLREORG Online Data Table Reorganization

The online reorganization (OLREORG) function was introduced as part of DBUTLTY in CA Datacom/DB. This function uses a patented process to reorder the table rows within the selected table into improved native key order. This improved native key order enhances the efficiency of applications that must access the data table in native key sequence.

OLREORG runs within the MUF while the data table is open for user access. This allows you to improve the sequential efficiency of the data without taking the data table offline.

Restart/recovery fully protects OLREORG within the MUF. Therefore, you can cancel the OLREORG process at any time and the amount of completed work is kept. Similarly, any MUF abend or LPAR failure terminates the OLREORG, but the reordering work that is done in the data table is kept.

You can implement the OLREORG process with three different modes:

- Reference group within existing data blocks (REFGROUP=2 or more). In this mode, the OLREORG process selects a set of blocks (the reference group) and moves rows between the blocks to improve the sequential order of the rows in those blocks. Then it moves onto the next set of blocks and repeats. The end goal of this reorganization is to improve native order efficiency within the existing data blocks. While moving rows to improve order, some blocks may be emptied and placed on the free block list.

- Reference group with additional data blocks (REFGROUP=2 or more). In this mode, the OLREORG process follows the same process as noted previously. However, if the order of the rows in the reference group cannot be improved because no free space is available in the selected blocks, the OLREORG process is allowed to acquire an empty block from the data area and use it to move the rows into sequential order and improve the efficiency. It then moves onto the next set of blocks and repeats. The end goal of this reorganization is to improve native order efficiency within the existing data blocks where possible and to use a limited number of new blocks to improve order for groups that cannot be improved otherwise. While moving rows to improve order, some blocks may be emptied and placed on the empty block list. In many cases, these newly emptied blocks make up for the empty blocks that are used during the move process. For OLREORG REFGROUP to use empty blocks for the move process, the DSOP option must be set to option 4 or option 5. For more information, see the following Data SPACE Option changes.
- Reference group with full reordering of data rows (REFGROUP=0). In this mode, the OLREORG process utilizes empty data blocks to copy the data rows into native sequence order. At the end of the REFGROUP=0 reorganization, the data rows are native key sequence within their data blocks providing a sequential efficiency similar to a full offline reorganization (BACKUP/LOAD). For OLREORG REFGROUP to use empty blocks for the move process, the DSOP option must be set to option 4 or option 5. For more information, see Data SPACE Option 4 - Random Reuse for OLREORG (see page 18) and Data Space Option 5 - Sequential Reuse for OLREORG (see page 18) changes.

For detailed instructions about executing the DBUTLTY OLREORG function, see the *CA Datacom/DB Data Reorganization Green Book*.

Data Space Option 4 - Random Reuse for OLREORG

To support the use of empty blocks during the OLREORG reference group reorganizations, the data area must have its data space option set to DSOP=4 (or DSOP=5).

DSOP=4 provides the same data space reuse functionality as DSOP=1. Additionally, DSOP=4 creates and maintains an empty data block map that is used for the OLREORG processing.

For detailed information about the DSOP=4 setting, see the *CA Datacom/DB Data Reorganization Green Book.*

Data Space Option 5 - Sequential Reuse for OLREORG

To support the use of empty blocks during the OLREORG reference group reorganizations, the data area must have its data space option set to DSOP=5 (or DSOP=4).

DSOP=5 provides the same data space reuse functionality as DSOP=2. Additionally, DSOP=5 creates and maintains an empty data block map that the OLREORG processing uses.

For more detailed information about the DSOP=5 setting, see the *CA Datacom/DB Data Reorganization Green Book*.

Dynamic System Tables

The new Dynamic System Tables are:

DIR_INDEX (DRI)

Provides Directory (CXX) information.

MUF_BUF_USE (MBU)

Provides information about buffer usage.

MUF_CBS (MCB)

Provides Compound Boolean Selection (CBS) information and statistics.

MUF_CBS_OLD_SETS (MCO)

Provides information about the oldest open CBS sets residing in the CBS buffer.

MUF_CONFIG (MCF)

Provides one row for nearly all MUF startup options and console API commands that are not represented in another Dynamic System Table. For example, since FLEXPOOL is listed in the MUF_OPTIONS table, the first or any changes to FLEXPOOL do not exist in the MUF_CONFIG table.

MUF_MEM_DETAIL (MMD)

Provides information about individual memory pools to help use the retrieved information using the ?MEM console command.

MUF_MEM_SUMMARY (MMS)

Provides summary information about current MUF memory use to help use the retrieved information using the ?MEM console command.

MUF_RC_DUMP_OPT (MDO)

Provides information about CA Datacom return codes to help you determine the PXX dump options in effect for a given return code.

SQL_CONSOLE (SQX)

Provides a new method of processing maintenance commands to the Dynamic System Tables.

Note: For more information, see the *CA Datacom/DB System Tables Reference Guide.*

Messages and Return Codes

The following are new messages:

```
DB00122I - ACCESS TYPE access1,access2,access3 TASKS=nnn
DB00131W - REQUEST BACKOUT WITH TXNUNDO=NO a b c d
DB00133I - JOB OPEN a b c d e
DB00134I - TESTING !_ DOING LOGIT x
DB00135I - CONNECT TO a b c d e f g h i j
DB00136I - DISCONNECT a b c d e f g h i j
DB00137E - INVALID RELEASE DBSYSID MODULE - x
DB00138I - OPEN MULTUSE=NO ALLOWED
DB00139I - XCF REQUESTS n1 ERRORS n2 n3 n4 n5
DB00205E - MULTI-USER ... ERROR - text
           added new codes - 193, 1123, 1124, 1125, 125
DB00218E - Insufficient 64-bit memory
DB00235W - XCF MESSAGE BUFFER SHORTAGE, nnnnnnn TIMES
DB00236W - XCF SIGNALLING PATHS BUSY, nnnnnnn TIMES
DB00237W - XCF ENFIRONMENTAL FAILURE, nnnnnnn TIMES
DB00256W - SDI ISSUE FAILED TO BE OPENED: RC=n REASON=r
DB00258i - MULTI-USER ENDING XCF SUPPORT (grpname,mufname)
DB00271I - AREAEV status, DATAFS status, DATAHU status, SU status
DB00272E - IWM4ECRE ERROR R15(r)=
DB0027RW - SMPTASK nn ZIIP AT  PERCENT, NOT CONFIGURED
DB00275W - SMPTASK nn ZIIP NOT SET, MUF ERROR
DB00276W - SMPTASK nn ZIIP NOT SET, MAY RETRY
DB00278I - DATASPACE NAME X
DB00280I - MUF STARTUP X_CXX_ALLOW_SHARING=YES, mufs MUF FOUND IN SYSPLEX
DB00281I - MUF CLEANUP job-name run-unit task clock info
DB00282I - MUF EOJ FORCE TERMINATE jobname run-unit
DB00622I - COMM close/stats n IGNORED, reason
DB00813I - ELAPSED SECONDS n CPU SECONDS n.nn
DB00814I - RQ/E n RD/E n WR/E n IX/E n DX/E n DT/E n
DB00815I - RQ/C n RD/C n WR/C n IX/C n DX/C n DT/C n
DB00816I - RQ n RD n WR n IX n DX n DT n
DB00817E - I/O ERR error dd/dtf-name ccccccc:h:r type volser volume
DB00910I - DEBUG CXX BASE ...
DB00911I - DEBUG CXX LOCK ...
DB01223I - RESTART, STAR UNCERTAIN a b c d e f
DB01226I - RESTART,VXX FORCE TASK jobname run-unit mufplex-number SPILL
           and/or REDO and/or ACTIVE and/or RRSOK
DB01335I - CONSOLE SCHEDULED - x
DB01705I - DYNAMIC EXTEND START ddname btc n dsn
DB01706I - DYNAMIC EXTEND END ddname atc n volser VOL n VOLS n EXT n +tc n
DB01919E - CONFLICT WITH LIMITED_INSTALL_MUF
DB02336E - SHADOW cccccccc IGNORED, XES STRUCTURE UNAVAILABLE
DB02807I - OPEN BASE n WITH TABLE WITH RECOVERY NO
DB02808I - OPEN BASE n WITH AREA WITH URI NO
```

```
DB02809I - LAST DBUTLTY LOAD/RETIX SHORT= BASE n
DB02810I - BASE n ACCESSED BY PROGRAM USING CBLDBMS/DATACOM
DB08002I - CA DATACOM PRESSPACK R12 AT SERVICE PACK: SPnn
DB08101I - CA DATACOM FAST RESTORE R12 AT SERVICE PACK SPnn
DB10095E - DBUTLTY NOT RUNNING AUTHORIZED
DB10098I - STARTING FUNCTION - echo
DB10099I - USER WTO--x
DB13044I - RETIX VERIFY MISMATCH COUNT nnnnnnnn
DB13092E - x HISTORY ERROR BASE n TABLE t SPL RC r (i) MUF(PLEX) m
DB13099W - SPILLOPT HISTORY ERROR BASE n TABLE aaa RC xx (yy) MUF(PLEX) bbbbbbb
DB13128E - keyword NOT COMPATIBLE WITH MULTI-DATASET INDEX
DB13129E - RETIX MINIMAL NOT SUPPORTED WITH BASE FORMAT 1
DB13278E - BASE b INDEX AREAS HAVE BEEN DELETED, IXX REMAINS
DB13279E - MULTIPLE FILEOUT OR NOT TABLE=
DB13282E - BASE b TABLE t NOT ALLOWED, 94(135) DB+USER COMPRESSION
DB13283E - RETIX MINIMAL WITHOUT URI INDEX LOADED
```

The following are new return codes:

```
Return Code 43 - added 006, 007
Return Code 72 - added 004
Return Code 75 - added 041, 042
Return Code 86 - added 016, 197
Return Code 89 - added 064
Return Code 94 - added 098, 127, 128, 129, 134, 135, 142, 167, 168, 170, 171, 172
```

The SVC Codes have been reorganized. The codes from 1 through 45 are now for z/OS only. The following new codes are for z/VSE only:

```
70 - INVALID SYSTEM TYPE
71 - INTERNAL ERROR
72 - SVC NOT INSTALLED
74 - UNABLE TO LOAD SVC PROGRAM
76 - INVALID VERSION OF TRANSIENT
78 - SYSTEM GETVIS FAILURE
81-82 - INTERNAL ERROR
83 - NOT VALID WHEN MUF DOWN
84 - MULTI-USER FACILITY RELEASE DOES NOT MATCH SVC
85 - INTERNAL ERROR
86 - NO SVC FOR THIS VERSION OF PRODUCT
87-89 - INTERNAL ERROR
91-96 - INTERNAL ERROR
99 - INTERNAL ERROR
```

Note: For more information, see the *CA Datacom/DB Message Reference Guide.*

Datadictionary Features

The following are new features and functions specifically for CA Datacom Datadictionary.

Datadictionary Audit Facility

You can use the Datadictionary Audit Facility to audit changes made to a DATABASE or TABLE structure and applied to the CXX that can affect the operation of the database engine and other products, such as CA Dataquery and CA Ideal. This facility is populated by LOGDW commands that place informational entries describing the events occurring in Datadictionary in the CA Datacom/DB Log Area.

Note: For more information, see the *CA Datacom/DB Database and System Administration Guide.*

Datadictionary View Generator

The Datacom View Generator is a facility to assist in the creation of SQL views. The facility reads the TABLE sub-structure and generates a view template containing all of the columns in the table. This template can then be edited to provide the specific SQL views required. Though the template is always generated as a DATACOM VIEW, the generator can process any table defined as eligible for SQL processing.

The Datacom View Generator is part of the Source Language Generation facility of DDUTILTY.

Note: For more information, see the chapter on DDUTILTY in the *CA Datacom Datadictionary Batch Reference Guide*.

Datadictionary Report Comparing DATA-DICT

You can use the CA Datacom Datadictionary Structure Definition Comparator Program (DDSDCLM) to assure the DATA-DICT database structure at your site is consistent with the structure as it would be installed by a new customer. The DDSDCLM reads the DATA-DICT database structure in PROD status at your site and compares it to the structure that would be installed at a new customer site. Any inconsistencies between your definition and the expected DATA-DICT definition are reported.

The purpose of the DDSDCLM is to catch those inconsistencies that could be present before upgrading from one version of the CA Datacom Datadictionary to another. These errors are most often caused by some inconsistency in the definition of the DATA-DICT database. The benefits are as follows:

- Before beginning the upgrade process, the DDSDCLM module assists you in making the necessary corrections to bring your definition into alignment with the expected baseline for the upgrade.
- Using it after the upgrade completes, helps ensure that the resulting structure is consistent with the new version.

DDSDCLM Processing

The DDSDCLM performs two functions:

- Reads the DATA-DICT DATABASE Structure in PROD status at your site. The DDSDCLM compares selected information regarding the structure to the information provided in the module created and delivered by CA Technologies that reflects the minimum expected structure.
- Reads all PRODuction and TEST occurrences of certain entity-types and edits them from compliance with current requirements.

The processing performed by DDSDCLM is as follows:

1. Reads the SYSIN input file and processes the following records:
    a. The first record in the SYSIN file must be a standard USR transaction unless the CA Datacom Datadictionary System Resource Table (DDSYSTBL) indicates that the external security system User ID is used. This transaction is processed as described for all other CA Datacom Datadictionary batch utilities. The user-specified in the USR transaction must have DDADMIN authority or processing does not proceed.
    b. The records described in this section are optional. The records are SET transactions as documented in the *CA Datacom/DB System and Database Administration Guide*. One additional SET transaction is accepted for this utility which is formatted as follows:
        - Bytes 1-4 contain -SET.
        - Byte 5 is blank.

- Bytes 6-9 contain Warning Messages OFF (WMOF) specifies that messages indicating differences in the structures that are not incompatible with a valid structure but are not part of the standard installation will not be printed. These are messages that indicate such things as additional tables in the database, additional keys or elements not in the standard structure, and additional columns not in the standard structure. All of these could indicate that you have implemented user-defined extensions. The default is to print these messages.

c. The final record must be a DDSDCLM transaction that is formatted as follows:

Comparator Transaction

```
-SDC vvv,stat
```

- Bytes 1-4 contain –SDC
- Byte 5 is blank
- Bytes 6-8 contain the version identifier of the DATA-DICT structure being processed (for example, 140, 120, 121)
- Byte 9 is, optionally, a comma
- Bytes 10-13 are, optionally, the Status of the DATA-DICT Database Structure to be processed (for example, PROD, T001, and so forth). If not provided, the default status is PROD. History versions are not allowed.

Verifier Transaction

```
-SDC VERIFY
```

- Bytes 1-4 contain -SDC
- Byte 5 is blank
- Bytes 6-11 contain VERIFY

2. For the Comparator Transaction, the following processing is performed:

a. Loads the Basis module corresponding to the version identifier specified in the DDSDCLM transaction. For example, if the DDSDCLM transaction contained 140, then DDSDCLM attempts to load module DD140SD. If the load fails, an error message is printed describing the problem. For more information, see DDSDC messages in the *CA Datacom/DB Message Reference Guide*.

b. Reads the DATA-DICT database structure in PRODuction status in the MUF accessed based on the DBSIDPR module used. DDSDCLM reads through the entire DATA-DICT structure and reports on differences between your DATA-DICT structure and the expected structure as defined by the Basis module.

3. For the Verifier Transaction, the following processing is performed:
   a. Reads all occurrences of the following entity-types:
      - AREA
      - DATABASE
      - CONSTRAINT
      - ELEMENT
      - FILE
      - FIELD
      - KEY
      - PARTITION-COLUMN-VALUE
      - SYNONYM
      - TABLE
      - VIEW
   b. Verifies that each occurrence passes the edit requirements for the current release of CA Datacom Datadictionary.

SDC Message Identifier Format

The DDSDCLM generates messages. The Verifier Transaction generated the same messages used in the DDUPDATE Batch Processor.

Message Identifier Format

Messages generated by DDSDCLM have the following identifier format:

DDSDCtnnnl

- DDSDC – Constant for all messages from DDSDCLM.
- t – Entity-type associated with the message or other generic messages:
  - D – Database
  - A – Area
  - T – Table
  - C – Column
  - E – Element
  - K – Key
  - N – No specific entity-type associated
  - G – General processing messages
- *nnn* – Numeric value unique within the set of messages defined by the first six characters.
- l – Error level:
  - **I – Informational messages.** No action required. Sets condition code to 00 if no other message levels are generated.
  - **W – Warning messages.** These are messages generated unless the –SET WMOF is used. If WMOF is set, then these messages do not affect the condition code. If WMON is in effect, then the condition code is set to 04 if no higher message level is generated.
  - **A – Attention messages.** These are messages indicating there is an inconsistency in the structure that requires user attention to correct. The condition code is set to 08 if no higher message level is generated.
  - **P – Processing messages.** These messages indicate a failure that typically ends processing immediately. For example, when an unexpected database return code is encountered during processing. The condition code is set to 16 and a snapshot dump may be taken.

Message Occurrence Names

Most messages have an entity occurrence associated with them. For FIELD, ELEMENT and KEY occurrences, the name displayed consists of the occurrence name of the owning TABLE occurrence followed by a period (.) followed by the occurrence name of the particular entity occurrence.

Message Values

Many messages are followed by two values separated by a slash (/) and contained within parentheses. The first value is the local value and the second value is contained in the Basis.

Finalization

If any messages indicate that corrective action is required, restore the DATA-DICT production structure to a test status and make the necessary changes. Once the DATA-DICT structure is determined as correct, copy the structure to PRODuction status and use the DDCFBLD batch program –CXX BUILD function and the CA Datacom/DB DBUTLTY batch program CXXMAINT DDPROD function to update the Directory (CXX).

KEY-INDEX-NAME Attribute

To implement the Multi-Dataset Index feature, the KEY entity-occurrence definition in Datadictionary has been enhanced by the addition of the KEY-INDEX-NAME attribute, a 3-character attribute that defaults to IXX. It can be set to IXX, or it can be set to I*nn*, where *nn* is a number from from 00 through 99. The name is used as the first 3 positions of the DDNAME for z/OS or DTFNAME for z/VSE.

AUDIT1 DDSYSTBL Macro Parameter

The AUDIT1 parameter in the DDSYSTBL macro specifies the level of detail recorded for the Datadictionary Audit Facility. The parameter refers to the CATALOG and BUILD functions

DATCOMIN Alias

DATCOMIN is an alias that can be used as a short form of DATACOM-INSTALL USER. If your site has a user ID length restriction of eight (8) characters or less, it can be used in place of the longer name. DATACOM-INSTALL is the user name for the Datadictionary ++USR statement.

FUNCTION Table Name Reserved

In CA Datacom Datadictionary Version 12.0, FUNCTION is a CA Technologies reserved name for a table. If you have previously named a user-defined table FUNCTION, rename it before upgrading to CA Datacom Version 12.0.

SQL Enhancements

The following enhancements have been added to CA Datacom SQL.

Note: For more information, see the *CA Datacom/DB SQL User Guide*.

SQL DROP PLAN (DBSQLPR)

The new DROP PLAN statement removes plans from CA Datacom SQL. The DROP PLAN statement is executable in DBSQLPR only.

Reserved Words

The MUF_NAME and TSN reserved words have been added to the list.

Special Registers

The following have been added to the special registers:

CURRENT DATACOM MUF_NAME

Produces a CHAR(8) containing the name of the MUF that is processing the request. This name can be specified in the MUF startup option, or it defaults to the job name of the MUF.

CURRENT DATACOM TSN

Produces an INTEGER containing the current value of the Transaction Sequence Number (TSN) of the current unit of work. If no maintenance (INSERT/UPDATE/DELETE) has been performed in this unit of work, the value returned is zero

Messages and Codes

A new SQL return code is -537 with a message string that contains information depending upon the error that has occurred. The message is truncated if it exceeds the 80-byte length of the SQLCA error message area.

Added to the SQL States Table, 2FS04 equates to the -537 SQL return code.

Messages and Codes

The following are new Datadictionary numbered messages:

```
DDFSMG0122 - INVALID FROM ENTITY-TYPE
DDFSMG0123 - INVALID TO ENTITY-TYPE
DDTRS0001E - FILE SYSPRINT COULD NOT BE OPENED
DDTRS0002W - PROGRAM FINISHED WITH WARNINGS
DDTRS0002E - PROGRAM FINISHED WITH ERRORS
DDTRS0010E - FILE filename COULD NOT BE OPENED, RC=n
DDTRS0011E - FILE filename COULD NOT BE CLOSED, RC=n
DDTRS0012E - NO SPACE FOR structure SSTRUCTURE, RC=n
DDTRS0013E - ERROR FREEING structure STRUCTURE, RC=n
DDTRS0014E - function FAILED DB COMMAND command, RC: nn(iii)
DDTRS0015E - function FAILED DSF COMMAND command, RC: ccc
DDTRS0016E - function FAILED SQL COMMAND command, RC: nnn SQL ERROR MESSAGE
DDTRS0017E - ERROR WRITING TO FILE 'filename', RC=n
DDTRS0018E - function FAILED $CSP COMMAND, RC=n
DDTRS0020I - DDTRSLM IS TRYING TO ROLL-BACK THE DATABASE DUE TO ERROR(S)
DDTRS0021I - BACKOUT SUCCESSFUL
DDTRS0022I - ROLLBACK SUCCESSFUL
DDTRS0099E - INTERNAL ERROR: text
DDTRS0101E - ERROR READING FROM FILE 'filename,' READ-RC=n
DDTRS0102E - FIRST COMMAND MUST BE 'SET USER'
DDTRS0103E - COMMAND LENGTH EXCEEDED MAXIMUM (n)
DDTRS0104E - INCORRECT COMMAND 'command'
DDTRS0015E - INCORRECT PARAMETER 'parameter'
DDTRS0106E - MISSING SEMICOLON (;) AFTER LAST COMMAND
DDTRS0107E - CAN'T MIX IMPORTS AND EXPORTS
DDTRS0108E - CAN'T MIX TIRAL IMPORT, IMPORT AND IMPORT ALL
DDTRS0109E - END OF FILE FOUND 'TRANSF,' RC=n
DDTRS0202E - authid_sqlname NOT FOUND IN DICTIONARY
DDTRS0203W - CNS SRC ROW(S) MISSING
DDTRS0401E -entity ddname NOT FOUND
DDTRS0406W - TABLE authid.sqlname OUT OF SYNC, SKIPPING
DDTRS0601E - IMPORT ENTITY NOT FOUND: entity authid.sqlname
DDTRS0602E - INVALID TRANSPORT ROOT ENTITY TYPE: entity
DDTRS0603E - IMPORT DATA NOT FOUND: text, READ-RC=nn
DDTRS0606I - SUCCSSFULLY IMPORTED
DDTRS0606E - entity FAILED TO IMPORT SUCCESSFULLY
DDTRS0606W - entity NOT IMPORTED
DDTRS0607E - UNEXPECTED END OF TRANSF DATASET
```

```
DDTRS0609E - INVALID TRANSF FORMAT
DDTRS0610I - TRIAL IMPORT REPORT:
DDTRS0701I - IMPORT PROCESSING:
DDTRS0710E - UNKNOWN ENTITY TYPE 'type'
DDTRS0711E - UNKNOWN PLAN ATTRIBUTE 'attribute'
DDTRS0712E - COUND NOT DETERMINE SQL NAME FOR entity type name
DDTRS0801I - CHOOSING OLD AUTHID FOR PLAN plan
DDTRS0802I - AUTHID FOR PLAN authid.name
DDTRS0803I - CHOOSING NEW AUTHID FOR PLAN plan
DDTRS0901E - entity authid-sqlname TRIAL DSF ERROR: COULD NOT PROCESS
              OCCURRENCE DUE TO DSF ERROR
DDTRS0902E - entity authid.sqlname TRIAL ERROR: DUPLICATE EXISTS IN DICTIONARY
DDTRS0903I - entity authid_sqlname TRIAL PROCESSED: REQUIRED PRESENT IN DICTIONARY
DDTRS0904E- entity authid.sqlname TRIAL ERROR: EXPECTED BUT NOT IN
             DICTIONARY OR TRANSPORT FILE
DDTRS0905E - entity authid.sqlname TRIAL ERROR: NOT IN DICTIONARY;
             IN TRANSPORT FILE; NOT IMPORTED
DDTRS0906E - entity authid.sqlname TRIAL ERROR: REPLACE REQUESTED
              BUT NOT FOUND IN TRANSPORT FILE
DDTRS0907E - entity authid.sqlname TRIAL ERROR: EXISTING OCCURRENCE
              NOT MARKED SQL ACCESSIBLE
DDTRS0908E - entity authid.sqlname TRIAL ERROR: EXISTING OCCURRENCE
              NOT IN SYNCH WITH DDD
DDTRS0909W - entity authid.sqlname TRIAL WARNING: DUPLICATE WILL BE REPLACED
DDTRS0910I - entity authid.sqlname TRIAL PROCESSED: WILL BE IMPORTED,
              NOT ALREADY IN DICTIONARY
DDTRS911E - entity authid.sqlname TRIAL ERROR: TARGET TABLE FOR
             CONSTRAINTS DOESN'T EXIST
DDTRS914E - entity authid.sqlname TRIAL ERROR: CONSTRAINTS ALREADY
             EXIST ON TARGET TABLE
DDTRS915E - entity authid.sqlname TRIAL ERROR: REFERENCED (PARENT)
             TABLE MISSING AND NOT IMPORTED
DDTRS0000E - INTERNAL ERROR - MESSAGE NOT FOUND(n)
DDWARA0012 - SQLNAME OR AUTHID BLANK, NOT BOTH
DDWBAS0007 - SQLNAME OR AUTHID BLANK, NOT BOTH
```

The following are new DSF return codes:

```
AHT - AREA HAS TABLE
ANF - AREA NOT FOUND
BHA - BASE HAS AREA
COD - CONSTRAINT OUTSIDE DATABASE
DK1 - DROP ATTEMPTED ON THE MASTER KEY
DK2 - DROP ATTEMPTED ON THE NATIVE SEQUENCE KEY
DNF - DATABASE NOT FOUND
IIN - INVALID INDEX NAME
NAP - NOT ALLOWED FOR PARTITION TABLES
SNL - SRT NOT LOADED
```

Note: For more information, see the *CA Datacom/DB Message Reference Guide.*

Chapter 4: Changes to Existing Features

General Updates

The following topics are changes for the CA Datacom core products.

SAMPJCL Changes

The samples that were in SAMPJCL in previous releases can now be found through the Support By Product link for the CA Datacom products home page at support.ca.com. When you are on the CA Datacom product page, click the Recommended Reading link and then the Use and Disclosure of Sample Programs link.

There are no sample JCLs provided for z/VSE.

Changes for z/VSE

The Multi-Guest Sharing Option (MGSO) and the use of the IBM Inter-User Communication Vehicle (IUCV) are not available starting with this release. Therefore, accept the default of 0 for the *vmnumber* parameter in the TASKS= MUF Startup Option and do not specify values for the VMID= and VMSUBID= parameters in the DBSYSID macro.

CA Datacom/DB Changes

The following changes have been made to CA Datacom/DB.

Common Memory Usage Information

All common memory for modules or GETMAIN is now 31-bit memory, instead of the 24-bit in common memory used in prior releases.

Note: For more information, see the *CA Datacom/DB Database and System Administration Guide.*

MUF Startup Option Changes

Startup option input statements that have been flagged in error are printed as output from the MUF. They are also printed on the console regardless of the specified value for the ECHO Multi-User startup option. You to see a single error or a few errors on the console more quickly than by searching the MUF output.

The updated MUF startup options are:

ACCESS

The OPEN value for the ACCESS option is used to attempt automatically to open the specified databases immediately after the MUF has been enabled.

MESSAGE

The MESSAGE startup option and console-like command allows you to specify the position of the values specified with the MUFMSG startup option as either a prefix or a suffix.

STATBFR

The STATBFR option is obsolete and has been removed. All statistics are now provided and buffer size is not an issue.

SYSTEMDBID

The SYSTEMDBID startup option is required to activate Dynamic System Table access from the MUF.

Other MUF execution changes include a change in step-down detection in the MUF RESTART function from using external date and time to using internal store clock time.

Note: RXX sequence-checking has already been using internal store clock time.

Because of the speed of the current generation of processors, a switch was made in internal step-down processing from using the external form of century, year, month, day, hour, minute, and second to using the store clock value instead. This prevents issues involving users bringing down a MUF and changing the external time to an earlier time, then enabling the MUF to an external time less than that of the previous MUF, such as happens in the change from daylight saving time to standard time.

Note: For more information, see the Modifying Multi-User Facility Startup Options and the Maintenance using Console Commands topics in the *CA Datacom/DB Database and System Administration Guide.*

Console Command Changes

The following are no longer supported and are removed:

- AGENT
- CBS AGE
- ESATASK
- LOGSP1 and LOGSP2
- MAXSTEN and MAXSTIO
- For Dynamic System Tables, the INSERT and DELETE functions for Accounting and covered/virtual
- For Dynamic System Tables, the UPDATE function for Accounting, covered/virtual, SYSOUT options

CBS Performance

Compound Boolean Selection (CBS) performance has been enhanced as follows:

- The Compound Boolean Selection buffer has been changed from a structure needing a lock when adding or deleting entries, to a structure that can use comparing and swapping to get or delete an entry.
- Each set, once it is built, now remains in its expanded form in MUF memory, unless it must be spilled to the CBS index due to the size of the CBS buffer. This eliminates the overhead of reconstituting the set for each SELNR (select-next-record) Compound Boolean Selection Facility command.

The Compound Boolean Selection (CBS) buffer architecture was changed, but most of the changes were internal in nature. From a user perspective, however, the major change of which to be aware is that the CBS buffer is no longer one large, contiguous piece of memory that is acquired at MUF startup. The CBS buffer memory is now acquired and freed as needed for each individual set. This has several ramifications as follows:

- At any given time, the CBS buffer memory is no greater than the amount needed to store open sets.
- CBS buffer architecture can be changed to more efficiently save and restore sets.
- Less memory is therefore used to store any given set, and sets in memory are stored in a simpler, more natural form, so that they do not have to be reconstituted.

Two New Memory Pools Added

Two new memory pools have been added as follows:

CBSS

The CBSS memory pool is the current Compound Boolean Selection buffer memory being used to store open sets.

CBSW

The CBSW memory pool is the Compound Boolean Selection working memory.

DBSYSID Macro Parameters

The following DBSYSID macro parameters are not allowed and have been removed:

- CCIAPPL=
- MUFNAME=

Note: For more information, see the *CA Datacom/DB Database and System Administration Guide.*

DBUTLTY Functions

The following are changes to the DBUTLTY functions:

ACCESS

The ACCESS function has a new keyword IGN68= which makes it possible to ignore return code 68. IGN68= makes it easier to set up and run JCL incorporating a stack of DBUTLTY functions without requiring that the MUF be enabled. For example, IGN68= would be useful if you wanted to load a database, perhaps to allow restart to properly occur, without first enabling the MUF.

BACKUP

The BACKUP AREA=CXX function has a new optional keyword DBID= which identifies the database to be backed up. If not specified, the function backs up the entire CXX as would have occurred in the previous release. With DBID= specified, the only output is the designated database. Only one DBID can be specified and only once, that is, you cannot specify a range of databases.

COMM

The COMM function has a new keyword IGN68= which makes it possible to ignore return code 68. IGN68= makes it easier to set up and run JCL incorporating a stack of DBUTLTY functions without requiring that the MUF be enabled. For example, IGN68= would be useful if you wanted to load a database, perhaps to allow restart to properly occur, without first enabling the MUF.

COMM OPTION=CANCEL

COMM OPTION=CANCEL has been removed.

COMM STATS

The COMM STATS function flushes the pipeline for the database as part of pushing the STATS to the CXX, index, and data sets.

COMM STATUS

COMM STATUS has had changes to the COMM STATUS report.

CXXMAINT ALTER DBCS

The ALTER DBCS option of the DBUTLTY function CXXMAINT allows you to specify what kind of double-byte character set (DBCS) support you would like for your Directory (CXX). Information on how to use this option has been updated to indicate that, if using SQL, the MUF must be restarted after running this command.

CXXMAINT ALTER=OPTION2

The CXXMAINT OPTION=ALTER function uses the OPTION2=keyword to activate other features:

- *datafs* - Reduces compression overhead and provides an optional search algorithm named DATA FAST SEARCH (DATAFS). The DATAFS algorithm provides possible performance-enhanced searching of multi-table areas and areas with tables that use compression.
- *datahu* - Specifies the high used mark option (OPTION2=Y_DATAHU or N_DATAHU). This option replaces DIAGOPTION CXX,08,ON/OFF from the rpevious release.
- *areaev* - Specifies the data area dataset extent validation option (OPTION2=Y_AREAEV or N_AREAEV). This option replaces DIAGOPTION CXX,32,ON/OFF from the previous release.

CXXMAINT Single User Mode Keyword

By default, when you convert an r11 CXX to a version 12 CXX, use of Single User mode is blocked for user programs, even if secured Single User is allowed. If you want to run Single User, you can use a new keyword, OPTION2=, when executing CXXMAINT.

To remove support for Single User, OPTION2=N_SU can be specified. This new option is provided to the full CXX report and has been added to the DB00271I MUF startup message

Running in Single User mode instead of Multi-User mode has been supported, but long recommended against. The Single User environment allows no logging and therefore has no restart or recovery capability. In addition, running in Single User bypasses external security. With these things in mind, the only uses of Single User mode that CA Datacom needs to continue to support are the DBUTLTY functions that run Single User and the DBFLSUB feature that loads an area with user program input. These exceptions of DBUTLTY and DBFLSUB are not affected by the N_SU option.

If a Single User open that is allowed occurs, message DB00138I is issued, confirming that fact and documenting the Single User execution.

EXTBKUP

The EXTBKUP function has a new option FORMAT=. If you do not specify FORMAT=, or if you specify FORMAT=EXTRACT, output is one or more extracted files, as was previously the case. If you specify FORMAT=BACKUP, the output is a single file in backup format with all records compressed by any option to be expanded. When FORMAT=BACKUP, you are required to have a previous single FILEOUT function to define the single output file, and in that FILEOUT function the keyword TABLE= must be set to *** (three asterisks) to specify that all tables are to be written to the output file.

LOAD

THE LOAD function option AREA=CXX using DBID= has been moved to MUF. This move prevents you from running it with the database open. It also ensures that MUF is always in control and has knowledge of all information in the CXX, a normal case being a single MUF or MUFplex to a CXX.

If a user has a LOAD or RETIX function coded in their JCL with SORT=0 specified or defaulted, that is, SORT= not specified at all, this information is added to the history report, and the first time in each MUF execution that an area is opened in a database that has the LOAD or RETIX function coded without the sort option, an informational message is generated. The message alerts you to the condition and can therefore help you know to add the sort option, and then see the results.

Following are two new features that are also blocked during LOAD or RETIX executions when the sort option is not being used:

- Dynamic extend of any data area or Index Area.
- Execution against a database with multiple index areas (partitions).

RECOVERY OPTION=FORWARD

The RECOVERY OPTION=FORWARD function has a new optional URINUM= keyword to speed support when MULTUSE=YES is not specified. The URINUM= keyword is used at the direction of Technical Support to gather documentation about the processing of a specific URI number.

Impact Report

The DBUTLTY Impact Report (TYPE=I,AREA=CXX) has been enhanced to provide information about cases in which the disposition of a table is INDEX NOT LOADED after a catalog. In those cases, further information is provided regarding key-level status. That is, whether each individual key in the table is LOADED or NOT LOADED.

REPORT AREA=PXX Changes

There are changes associated with certain messages related to the AREA=PXX report, and there are also changes with regard to the DUMPS=FULL and DUMPS=REQUEST options.

RETIX

The count of records in a table, as known to the CXX, can become wrong if the MUF is restarted after a MUF outage occurs. The DBUTLTY RETIX function, with the KEYNAME=*SETR option, lets you recount the records and set the new value in MUF.

If a user has a LOAD or RETIX function coded in their JCL with SORT=0 specified or defaulted, that is, SORT= not specified at all, this information is added to the history report, and the first time in each MUF execution that an area is opened in a database that has the LOAD or RETIX function coded without the sort option, an informational message is generated. The message alerts you to the condition and can therefore help you know to add the sort option, and then see the results.

Following are two new features that are also blocked during LOAD or RETIX executions when the sort option is not being used:

- Dynamic extend of any data area or Index Area.
- Execution against a database with multiple index areas (partitions).

RXX Report MOVER Command Changes

The RXX report command summary that prints the occurrences of some commands now specifies the following for the MOVER command:

MOVERU

Specified for MOVER log records that were done for a record moved in support of the UPDAT command for compressed tables.

MOVERO

Specified for MOVER log records that were done for a record moved in support of the DBUTLTY OLREORG function.

SET

The SET function has the following enhancements

- SET can be used to override the Master List (DBMSTLST) settings.

  You have a DBMSTLST comprised of numbers relating to the index and data buffers. Those numbers are often very small. In many DBUTLTY functions, the index and data buffers are not used at all. In other functions, they are used very little. But they are used a great deal in a few functions.

  Because the functions that would benefit from more index and data buffers are few but can be very long running, the DBUTLTY SET function OPTION1= keyword makes it easier for you to have a general Master List as in previous releases, but to also have the ability to increase the buffers for needed functions in databases with a large number of records. Consider the following, for example:

  - The DBUTLTY RECOVER FORWARD function would benefit from many index and data buffers when there is a lot of input to process.
  - The DBUTLTY BACKUP NATIVE function would benefit from having many data buffers when there are a lot of records unordered by the native sequence key.

  The override values must be set prior to the first DBUTLTY function that opens as Single User. The following are the Master List-related DBUTLTY SET function OPTION1= enhancements:

  - SET OPTION1=DATANO=9999 (3-9999)
  - SET OPTION1=DATALN=4096 (4096-32767)
  - SET OPTION1=IXXNO=9999 (3-9999)
  - SET OPTION1=DXXNO=9999 (3-9999)
  - SET OPTION1=IXXLN=8192 (4096-8192)

- Option OPTION1=ECHO_FUNCTIONS echoes statements read by DBUTLTY to the console as they are starting to be processed.
- Option OPTION1=WTO--*x* allows you to specify a set of characters to form a message you want written to the console. You supply the text after the two dashes.
- SET has a new stacking option to provide a maximum of SET options with a minimum number of input statements. To use this special stacking, use a semicolon (;) as a special character to separate individual set values.

SPILLOPT

The SPILLOPT function for z/VSE has the new keyword DEVICE= that specifies the device type for the output file.

Note: For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

Dynamic System Tables

The following Dynamic System Tables have been modified with new or modified columns:

- DIR_AREA
- DIR_DATABASE
- DIR_DIRECTORY
- DIR_KEY
- DIR_TABLE
- MUF_ACTIVE_TASKS
- MUF_AREA_STATS
- MUF_COVEREDVIRTUAL
- MUF_DATA_SHARING
- MUF_INTERNAL_STATS
- MUF_LOGGING
- MUF_SMP_STATS
- MUF_TABLE_STATS

Note: For more information, see the *CA Datacom/DB System Tables Reference Guide.*

Encouraging the Use of DBNTRY Entry Point

The DBNTRY entry point supports the caller in key 8 in problem mode with all work areas in key 8. We recommend using programs that use the DBNTRY entry point.

Recovery Recommendations

Address Space Changes

Because CA Datacom Version 12 is built for speed, it uses additional memory compared to previous releases in the following:

- MUF address space
- Applications using the MUF address spaces
- DBUTLTY address space

In z/VSE environments, you may need to enlarge the address space for the MUF and DBUTLTY (to at least 8M).

Monitoring Restart REDO Reprocessing

Message DB01203I announces the beginning of the restart REDO process. DB01203I also gives information about the number of log tracks (or blocks) that need to be reprocessed.

The STATUS console command has been enhanced to help you monitor the progress of restart reprocessing without having to use the restart report.

Encouraging the Use of Tables with RECOVER YES

We recommend using tables with RECOVER YES. Message DB02807I has been added to alert you to the existence of tables specified with RECOVER NO.

Many features are not compatible with RECOVER NO. Some features not available with RECOVER NO are as follows:

- Logging maintenance to the LXX
- Transaction backout
- Restart to ensure all index-data links are complete
- Ability to have multiple Index Areas for database
- Ability to dynamically extend during DBUTLTY LOAD and DBUTLTY RETIX

Encouraging the Use of Data Areas with URI YES

We recommend using data areas with URI YES. Message DB02808I was added to alert you to the existence of data areas specified with URI NO.

Many features are only compatible with URI YES data areas. Features only available with URI YES include the following:

- Full protection of transaction boundaries
- Ensured ability to roll back transactions
- Ability to have multiple Index Areas for database
- Ability to dynamically extend during DBUTLTY LOAD and DBUTLTY RETIX
- Ability to use SQL to process tables in the area

No COVERED and VIRTUAL in LXX or FXX

Neither COVERED nor VIRTUAL are allowed to be assigned to the Log Area (LXX) or the Force Area (FXX).

DBUTLTY STATUS Report Changes

The report generated by the DBUTLTY COMM OPTION=STATUS function has changed. The always present third line per task with the JOBNAME, R-UNIT, and TASK fields is now followed by status information fields BREF, R-TIME, C-TIME, D-TIME, and USER-ID.

Additional Recommendations

We recommend the following settings because many new features require them:

- Define all tables with the RECOVERY attribute set to Y.
- Load all areas as URI.
- Set all databases in the MUF to OPTIMIZE.

Messages and Codes

The following error messages have been removed:

```
DB00203I - MULTI USER CANCELLED
DB10044E - ACCT EDIT SUPPORTS DATABASE OR TABLE
DB13015W - DATA LOADED WITHOUT REORGANIZATION
DB13052I - INSUFFICIENT SPACE FOR AREA STATISTICS
DB13053I - JOB STATISTICS OVERFLOW: nnnnn BYTES
DB13055E - BASE nnnn ALREADY DEFINED
DB13956E - BASE nnnn NOT DELETED, CONTAINS ACTIVE TABLE(S)
DB13241I - INDEX TASK-n COMPLETED, CPU-HH:MM:SS, ENTRIES-n
DB20254W - FETCH SUPPLIES FEWER HOST VARIABLES THAN ITEMS SELECTED
```

The Return Code 20 - DIRECTORY KEY/ELEMENT BUFFER TOO SMALL has been removed.

The following SVC Codes have been removed:

```
12 - SMF ERROR
13 - ENABLE FOR MORE THAN MAXCORE
18 - GETMAIN ECSA/CSA/SVA FAILURE
19 - INVALID SVC TYPE
20 - INVALID SVC NUMBER
21-26 - INTERNAL ERROR
27 - SMF REC LT 128
28 - SMF REC NOT - MUF SMF REC
29-32 - INTERNAL ERROR
34 - COMM/ACCT REQUEST NOT IN MUF PRIVATE ADDRESS SPACE
39 - VSE/ESA OPTION REQUIRED
46 - INVALID VERSION OF TRANSIENT
```

Datadictionary Enhancements

The following changes are relevant to CA Datacom Datadictionary.

Address Space Size Change for DDTRSLM

The minimum execution size parameter for the execution of the DDTRSLM transport utility is 640K.

Minimum Block Size for Data Area (DBUTLTY Backup of DD1

The minimum block size parameter for the backup data set for CA Datacom Datadictionary data area is 8K – “BLKSIZE=8192”.

Datadictionary Catalog Opens CXX Database

From the point of view of the CXX, the catalog of a database causes the database to be opened.

Note: No data set is opened because of a catalog.

We strongly recommend that you use the MUF ACCESS OPTIMIZE option. If you follow our recommendation and run with that option and if Single User DBUTLTY functions are to be executed (for example, INIT AREA=IXX), it is possible that a COMM CLOSE might be needed after a catalog.

1000 ALTER Transaction

You can use the 1000 ALTER transaction to apply changes to the CA Datacom/DB CXX and to relevant attributes of certain entity-types without closing the database. The database is opened if the 1000 ALTER transaction finds that it is closed.

You can use the 1000 ALTER transaction for system-defined databases, such as DATA-DICT, DDD-DATABASE, and also for user-defined databases.

The 1000 ALTER transaction is only valid for the DATABASE and AREA entity-types. Processing of the 1000 ALTER only affects the CXX entry for the specified entity-occurrence in the header transaction.

The 1000 ALTER transaction is valid for [set the dad variable for your book].

When using 1000 ALTER, consider the following restrictions:

- The database must have been set or defaulted as ACCESS OPTIMIZE during the enabling of the MUF.
- The database must not be defined as VIRTUAL.
- The MUF must currently be a single instance of the MUF and not part of a MUFplex with two or more MUFs currently enabled. It is acceptable to have a Shadow MUF that is not enabled.
- The occurrence to be altered must have been previously cataloged and must match the identity information that is stored in the CXX.
- If you modify a database, the base must be FORMAT 2 in the CXX, indicating that a fallback to r11 is not likely to occur. Also, the SQL SECURITY data cannot be different from that currently in the CXX.
- If you modify a data area, the DATACOM NAME must match, and the area must be in URI format.

The DATABASE entity-type and AREA entity-type attributes are passed from Datadictionary to CA Datacom/DB for comparison to the CXX. Processing either does or does not take place depending on the results of that comparison.

Note: For more information, see the *CA Datacom Datadictionary Batch Reference Guide.*

DDTRSLM Messages

The Datadictionary DDTRSLM messages have been rewritten from the messages in the previous releases.

Note: For information, see the *CA Datacom/DB Message Reference Guide*.

Programming Change

The INQIN command, if there is an open URT, has been enhanced.

Note: For information, see the *CA Datacom/DB Programming" Guide*.

SQL Changes

The following is a change to CA Datacom SQL.

IBM LE C Requirement

If you want to use utilize SQL, you must have the IBM LE C runtime libraries available in the MUF.

Note: For more information, see the *CA Datacom/DB SQL User Guide*.

CREATE TRIGGER Statement

As a convenience to users of CREATE TRIGGER who need to pass large base table rows to a triggered procedure, DATACOM_WHOLE_ROW column has been added. DATACOM_WHOLE_ROW is visible only during execution of the CREATE TRIGGER statement. When you pass the special DATACOM_WHOLE_ROW column name to your procedure, the procedure receives a CHAR parameter containing the entire row in CA Datacom/DB format.

Note: For more information, see the *CA Datacom/DB SQL User Guide*.

SQL COBOL Preprocessor

The SQL COBOL Preprocessor now puts the translated SQL statements in SRCOUT. The translated statements were formerly put in SYSPCH. Modify your existing JCL to take this change into account.

SQL COBOL Precompiler

The SQL COBOL Precompiler DBXMMPR now writes the translated statements to SRCOUT and not to IJSYSPH as was the case with Version 11.0.

SQL Additional Memory Requirement

With Version 12.0, the CA Datacom Multi-User Facility (MUF) requires a minimum recommended partition size of 16M for systems with the SQL option enabled. The partition size requirement may be slightly larger for sites with large buffer pool allocations.

As a rule of thumb, a CA Datacom MUF running the SQL Option requires an additional 300K per task as defined in the "number" option of the TASKS Multi-User Startup Option. Failure to provide adequate memory can lead to the following abend error message:

CEE3322C EXECUTION ABNORMALLY TERMINATED WITH USER-ABEND CODE 4093

VSAM Transparency

The following change is for CA Datacom VSAM Transparency users running against a CA Datacom/DB Version 12 environment.

CODE CSP=NO in Online VSAM Interface Table (VIT)

To use VSAM Transparency to access tables that are defined in a CA Datacom/DB V12 environment, code the CSP=NO parameter in the Online VSAM Interface Table. If the parameter is not coded, CSP=NO is the default.

Note: For more information, see the *CA Datacom VSAM Transparency Guide*.

Chapter 5: CXX Upgrade

The process of upgrading from r11 to Version 12 differs from the process of upgrading from r10 to r11. The ability to leave the CXX at the previous release for a time is not allowed. Using the upgrade process requires the conversion of CXX to Version 12 and compatible solutions applied to r11 to provide protected access to its features.

During the upgrade to Version 12, the CXX is assigned a CXX level of LEVEL 1. The level is reported but there is no other immediate visibility. Additionally, a new database level format of FORMAT 1 is assigned to each database.

FORMAT 1 specifies that the database is compatible between Version 12 and r11 and features in Version 12 that are not compatible with r11 are disabled in Version 12.

The following are the disabled features of FORMAT 1 Version 12:

- The ability to define the dynamic extend value in cylinders instead of tracks
- The ability to have key level loaded status. One use of this status is having a loaded table with *n* keys, adding a new key, and to RETIX the new key instead of all keys.
- The ability to have multiple index data sets within one database.

You can enable the Version 12 features that are disabled upon the decision that no fallback is needed from Version 12 to r11. Enable the Version 12 features with a DBUTLTY function that alters the format for all databases from FORMAT 1 to FORMAT 2.

If you need to fallback from Version 12, use Version 12 to backup its CXX, use r11 to initialize the CXX as r11, and execute Version 12 to convert the Version 12 backup into the r11 CXX. This restores the complete CXX r11 specifications and blocks Version 12 from access.

Important! It is presumed there is no need for fallback to r11 if you catalog a database with a new DBID to a Version 12 CXX, and the database is therefore set as FORMAT 2. If the Version 12 FORMAT 2 is not the desired format, the format should be set with DBUTLTY to FORMAT 1 after the catalog.

Upgrade an r11 CXX to Version 12

You should only upgrade an r11 CXX to a version 12 CXX when the CXX is not open.

To upgrade an r11 CXX to a version 12 CXX:

1. Backup the CXX using r11.
2. Initialize the CXX using Version 12.
3. Issue the DBUTLTY CXXMAINT OPTION=CONVERT1112 command to accept the r11 backup as input and build the Version 12 CXX with the definitions.

The r11 CXX is upgraded to Version 12 CXX.

Note: The DBUTLTY CXXMAINT OPTION=CONVERT1112 command sets the CXX level to LEVEL 1 and each database format to FORMAT 1. The command must follow a DBUTLTY INIT AREA=CXX function in the same step or a following step. The r11 MUF diagnostic options, DIAGOPTION CXX, are not preserved in an r11 CXX backup and are not pushed into Version 12.

Actions and Requirements at the Full CXX Level

For Version 12, the general execution actions and requirements at the full CXX level, the following applies:

- The Version 12 code allows the open of a CXX set to r11 in the single condition of the DBUTLTY function CXXMAINT with OPTION=CONVERT1211.
- The Version 12 code executes in all other cases with a CXX of Version 12 and at CXX LEVEL 1.
- The DBUTLTY function REPORT with AREA=CXX and no TYPE=A and no DBID includes the CXX level on page one of the report.

Actions and Requirements at r11 when CXX Is Version 12

If fallback is needed, convert the Version 12 CXX back to r11 format and style. With the CXX at r11, all r11 features are permitted. The ability to open a version 12 CXX requires solutions applied to r11. Without solutions applied, the r11 code does not recognize the CXX and generates a return code 47(024).

With r11 solutions, the following applies:

- The DB00905 message indicating the Directory (CXX) version reflects 12 when it occurs.
- The DBUTLTY function BACKUP with AREA=CXX fails with message: DB13149 - WRONG CXX RELEASE FOR THIS FUNCTION. With the solution, r11 does not have the ability to understand all the Version 12 changes or be able to successfully back them up.
- The DBUTLTY function LOAD with AREA=CXX fails with message: DB13149 - WRONG CXX RELEASE FOR THIS FUNCTION.
- The existing message: DB13060E - CXX BACKUP IS VERSION 12, CANNOT LOAD TO CXX VERSION 11, in Version 12 fails. Users who require the ability to move a CXX database definition from a version 12 CXX to an r11 CXX should add a temporary r11 CXX and downgrade the backup using the Version 12 command, CXXMAINT OPTION=CONVERT1211, then use r11 to BACKUP the temporary r11 CXX.
- The MUF properly supports the CXX level Version 12 options of AREAEV and DATAHU known in r11 as DIAGOPTION_CXX 08 and 32. The MUF does not allow any use of 'DIAGOPTION CXX'.
- The r11 DBUTLTY function REPORT with AREA=CXX is not enhanced to print any additional Version 12 information. It prints all databases that are FORMAT 1. If a database is FORMAT 2, it skips the database after printing the message: DB13102 - WRONG DATABASE FORMAT FOR THIS FUNCTION.
- The DBUTLTY function CXXMAINT with OPTION=CONVERT fails with the message: DB13149 - WRONG CXX RELEASE FOR THIS FUNCTION.
- The Version 12 CXX must have a CXX level of LEVEL 1.
- The open of a database for any purpose that is defined as FORMAT 2 fails with a return code 94(094).

Note: This is intended for special case test systems only and not for user production. Users in production are expected to convert the CXX from Version 12 to r11, if a fallback is needed.

New Version 12 Database Format

Each database in Version 12 is set to a format number that allows databases to be treated the same or differently by different CA Datacom releases.

Change One or More Databases from FORMAT 1 to FORMAT 2

If you have no plans to fallback to r11, use the DBUTLTY function CXXMAINT OPTION=FORMAT0102 (optional DBID=value). For normal users, this is done once with the keyword set as DBID=1-5000. At this point, the disabled Version 12 features relating to the CXX become available.

For testing, the function can be used with individual databases or ranges of databases. For each DBID selected, the database format is set to FORMAT 2. If the index is currently set as loaded, each key defined is also set to loaded and if the index is set to not loaded, each key defined is set to not loaded.

Actions and Requirements at FORMAT 2

For Version 12, the general execution actions and requirements at the database level when the database format is FORMAT 2, the following applies:

- The database activates key level loaded status, allowing the DBUTLTY function RETIX new option MINIMAL to be able to efficiently add or drop one or more indexes without rebuilding unchanged keys.
- The database allows multiple index data sets instead of the single IXX area.

Change One or More Databases from FORMAT 2 to FORMAT 1

To change a database from FORMAT 2 to FORMAT 1, perform the following steps.

Note: This procedure is intended for special case test systems only.

1. Execute the function DBUTLTY CXXMAINT OPTION=FORMAT0201 (optional keyword, DBID=*value*).
2. If the index is set to dynamic extend in cylinders, converts to tracks.
3. If there are multiple index data sets, removes them with warning message (DB13278W) but RC 0.
4. If any data area is set to dynamic extend in cylinders, converts to tracks.
5. Sets the tables index loaded or not based on every key definition being loaded.
6. Keeps every key definition in CXX as a normal key losing any other status, such as being created or dropped (not visible to users).
7. Sets selected database as FORMAT 1.

Actions and Requirements at FORMAT 1

For Version 12, the general execution actions and requirements at the database level when the database format is FORMAT 1, the following applies:

- The database ignores the key status, for example, loaded.
- The database rejects catalog with cylinder set for dynamic extend with RC 94(094).
- The database rejects catalog with multiple index data sets with RC94(094).

Downgrade a Version 12 CXX to r11

You can only downgrade a version 12 CXX to an r11 CXX when the CXX is not open.

To downgrade a version 12 CXX to an r11 CXX:

1. Backup the CXX using Version 12.
2. Initialize the CXX using r11.
3. Issue the Version 12 DBUTLTY CXXMAINT OPTION=CONVERT1211 to accept the Version 12 backup as input and build the r11 CXX with the definitions.

The Version 12 CXX database is downgraded to an r11 CXX database.

Note: The DBUTLTY CXXMAINT OPTION=CONVERT1211 command must follow an INIT AREA=CXX function as a following step. This command does not restore MUF DIAGOPTION CXX values used by r11 (08 or 32).

FORMAT 2 Databases

You can downgrade Version 12 CXX databases set with FORMAT 2 to r11.

During the CONVERT1211, the following actions are taken for each database set to FORMAT 2.

Note: Other than special case test systems, users are not expected to have these conditions.

1. If the index is set to dynamic extend in cylinders, converts to tracks.
2. If there are multiple index data sets, removes them with warning message (DB13278W) but RC 0. The index is set to NOT LOADED.
3. If any data area is set to dynamic extend in cylinders, converts to tracks.
4. Sets the tables index loaded or not based on every key definition being loaded.
5. Keeps every key definition in CXX as a normal key losing any other status, such as being created or dropped (not visible to users).

Example Messages

The following new Version 12 messages are related to the CXX upgrade feature. In the messages that follow, DBID=789 is used as an example.

DB13278W

BASE n INDEX AREAS HAVE BEEN DELETED, IXX REMAINS

Reason:

While the DBUTLTY function CXXMAINT was being executed with either OPTION=FORMAT0201 or OPTION=CONVERT1211, a database was found with an alternate index area, not IXX. With FORMAT 1, the only allowed index area is IXX. Every table with a key definition using the alternate index area is set to the message: INDEX NOT LOADED. In the message text, the *n* is the database ID.

Action:

At the completion of the function, execute and review a REPORT AREA=CXX without TYPE=A to see if any table has an index that is not loaded, then handle accordingly. We recommend that you avoid getting into this situation by not using FORMAT 2 for a database that has a risk of being subject to a downgrade. With this warning message, DBUTLTY ends the function without recognizing this as an error and therefore continues other functions.

DB13280W

BASE n AREA aaa NOT INITIALIZED, DEFAULTING TRACKS PER CYLINDER TO 15

Reason:

While the DBUTLTY function CXXMAINT was being executed with either OPTION=FORMAT0201 or OPTION=CONVERT1211, the index or a data area was found to be using dynamic extend in cylinders and must be converted to tracks. The index or data area has not been initialized. Therefore, the device type is unknown, which means that CA Datacom cannot determine the correct number of tracks per cylinder to use in the calculation. The function presumes the number to be 15 and continues. In the message text, the *n* is the database ID, and the *aaa* is the area name.

Action:

If the presumption of the number being 15 is not correct, the database must be cataloged to get the correct number of tracks as currently defined. We recommend that you not get into this situation by not using FORMAT 2 for a database at risk of being subject to a downgrade. With this warning message, DBUTLTY ends the function without recognizing this as an error and therefore continues other functions.

DB13281W

BASE n AREA aaa DYNAMIC EXTEND CHANGED FROM n CYLINDERS TO n TRACKS

Reason:

While the DBUTLTY function CXXMAINT was being executed with either OPTION=FORMAT0201 or OPTION=CONVERT1211, the index or a data area was found using dynamic extend in cylinders and must be converted to tracks. In the message text, the *n* following FROM is the number of CYLINDERS, and the *n* following TO is the number of TRACKS. In the message text, the *n* following BASE is the database ID, and the *aaa* is the name of the area.

Action:

If the presumption regarding the number of tracks is not correct, the database must be cataloged to get the correct number of tracks as currently defined. We recommend that you not get into this situation by not using FORMAT 2 for a database at risk of being subject to a downgrade. With this warning message, DBUTLTY ends the function without recognizing this as an error and therefore continues other functions.

DB13282E

BASE n TABLE ttt NOT ALLOWED, 94(135) DB+USER COMPRESSION

Reason:

While the DBUTLTY function CXXMAINT was being executed with OPTION=CONVERT1112, a table was found that has both DB compression and also user compression defined. This combination was not valid in r11 for processing and is not allowed in the CXX in Version 12. In the message text, the *n* is the database ID, and the *ttt* is the name of the table.

Action:

Before the CXX can be converted to Version 12, the table must be deleted using r11 facilities.

Special Case Test Systems

Two abilities of special case test systems that are not intended for normal use are the following:

- The ability to execute r11 with CXX Version 12, with exceptions.
- The ability to change the format of individual databases so that some are FORMAT 1, which can be accessed by r11 and Version 12 and subject to easy fallback while other databases are FORMAT 2, which are intended for Version 12 only and cannot be accessed by r11 and have an exception during fallback.

New Message during CXXMAINT OPTION=CONVERT1112

Before Version 12, Datadictionary disallowed tables with both DB and user compression, and users were told to convert tables that were so defined. In Version 12, that ability is neither supported nor allowed as a valid definition in the CXX.

During the CXX conversion from r11 to Version 12, if any table is found with this option a message is issued and, when all of the input file has been processed, DBUTLTY is terminated. This allows all tables to be found in a single pass of DBUTLTY. The table definition must be converted in r11 before attempting to convert again.

An example of the associated error message follows:

BASE 2 TABLE AGR NOT PROCESSED, 94 (135) DB+USER COMPRESS

In this sample message, 2 is the database containing the table, and AGR is the table that contains the invalid combination.