CA Datacom®/DB

System Tables Reference Guide

Version 15.00

technologies

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA Datacom® CICS Services
- CA Datacom® Datadictionary™
- CA Datacom® DL1 Transparency
- CA Datacom® Presspack
- CA Datacom® Server
- CA Datacom® SQL (SQL)
- CA Datacom® STAR
- CA Dataquery™ for CA Datacom® (CA Dataquery)
- CA Dynam®/D Disk Management for z/VSE (CA Dynam/D for z/VSE)
- CA SYSVIEW® Performance Management (CA SYSVIEW)

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.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- DIR_AREA (DRA) (see page 17) - includes new columns for GENERATION and URI_REUSE
- DIR_DATABASE (DRB) (see page 25) - the BASE_FORMAT column was updated
- DIR_DATASET (DRZ) (see page 27) - includes new columns for GENERATION, and SECONDARY_CYLS
- DIR_DIRECTORY (DRD) (see page 31) - includes new columns for GENERATION and CXX_VERSION
- DIR_KEY (DRK) (see page 35) - includes new columns for GENERATION and KEY_USE; changes were made to the CBS_IGNORE, KEY_STATUS, UNIQUE_SEARCHED and UNIQUE_TIME columns
- DIR_TABLE (DRT) (see page 46) - includes new columns for GENERATION and SQL_INTENT
- MUF_LOGGING (MFL) (see page 94) - includes new columns for GENERATION, LOG_BLOCK_SEQ, and LOG_RECORD_SEQ_G0. The format of the column LOG_RECORD_SEQ was changed.
- MUF_COVEREDVIRTUAL (MFC) (see page 81)
- SQL_STATUS (SQS) (see page 147)
- SQL Cache Tables (see page 161) - updated for the CA Datacom 15.0 SQL Cache feature
- SRV_CLNTINFO(SRV) - new table that allows tracking Java application connections with CA Datacom Server 15.0.

Contents

Chapter 1: Introduction

This guide is distributed with Version 14.0 of CA Datacom/DB to provide information about CA Datacom/DB system tables.

Intended Audience

This guide is intended for those responsible for one or more of the following:

- Supporting CA Datacom system software
- Administering the operations of the CA Datacom system
- Maintaining data integrity
- Ensuring data accessibility
- Maintaining system performance

Disclaimer

The sample code, JCL, and reports provided in this guide are intended for use as reference aids only. No warranty of any kind is made as to the completeness or correctness of the exact samples in your specific installation environment. If you are planning to use any of the samples provided in this guide, be sure to adjust them for your site standards and use.

Chapter 2: Dynamic System Tables

Dynamic System Tables (located in the SYS area, base 1000) exist to provide real-time access to CA Datacom/DB system information. All Dynamic System Tables are accessible through SQL, CA SYSVIEW, and Unicenter CA-Explore Performance Management for z/VSE. The information includes data typically produced through the CA Datacom/DB utility (DBUTLTY) reports and the MUF end-of-job reports. They also provide data on the current MUF configuration and status.

The data is normalized and built so that various tables can be easily joined. From the SQL perspective, these are normal tables with the full power of SQL, except that these tables do not accept maintenance or perform row locking.

The SQL_CONSOLE (SQX) Dynamic System Table is provided to change various Multi-User settings dynamically. It can also be used to perform console-like actions using an SQL query. For more information, see SQL_CONSOLE (SQX) (see page 143).

The tables are defined through CA Datacom Datadictionary and cataloged to the CXX as other tables would be. All data is built dynamically from other sources at the time it is requested. This helps ensure that the data is always correct at the point of need. However, the system does move on and by the time the application sees the information, it may have aged. It is as accurate and timely as it is possible to be.

Setting Up the Environment

The Dynamic System Table definitions are automatically installed to the CXX and CA Datacom Datadictionary as part of the Version 14.0 upgrade or install process. If you use the recommended DBID of 1000, you do not need to deal with the BTG member or catalog the database. Member BTG1000 in the macro library provides the Dynamic System Table definitions. Process these definitions through CA Datacom Datadictionary and catalog the base. The database selected is 1000, but it can be changed to any valid DBID. The database ID selected must be provided in the Multi-User startup option SYSTEMDBID *n* for the Dynamic System Tables to have data accessible.

Important! Specify the SYSTEMDBID Multi-User startup option to activate Dynamic System Table access from the MUF.

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

Do not modify other information. The SQL processing is based upon the CA Datacom Datadictionary definitions as they are cataloged. The actual data is built based upon the database ID matching the SYSTEMDBID, the CA Datacom table names, and the definitions provided. Changing field definitions would cause unpredictable results.

Place all the Dynamic System Tables in the same database. The DBID of the database in which you place all Dynamic System Tables must match the DBID specified with the SYSTEMDBID Multi-User startup option. We recommend that you use a DBID of 1000 for the Dynamic System Tables database, but it is possible to specify any DBID of your choice for the SYSTEMDBID. However, Dynamic System Tables cannot share a database with any other tables. The database containing Dynamic System Tables do not need to be initialized or loaded, and it should *not* be defined as virtual.

For security purposes, these tables are like all others. You can use any of the security features to protect or restrict the data.

Accessing Through CA Datacom STAR

The SYSTEMDBID Multi-User startup option represents a local database. All user jobs running against this base represent the MUF that they are connected to.

To access Dynamic System Tables at a remote MUF through CA Datacom STAR, the definitions can be added to CA Datacom Datadictionary in another database that is defined to be remote. The new remote database should have different occurrence names for DATABASE, AREA, and TABLE. Each table should have a new AUTHID different from the ones in the other database. The SQLNAME and DATACOM-NAME for each table should remain the same. The SQLNAME for DATABASE and AREA must be changed if it is supplied, or it can be left blank.

Using SQC Table to Cancel SQL Requests

The SYSADM.SQL_STATUS_CURRENT (SQC) (see page 149) Dynamic System Table describes the current status of all active SQL transactions. Detailed information is provided about all currently executing SQL transactions including the following:

- Processing steps (for example, joins, unions, sorts)
- What tables and indexes are being accessed
- Number of rows read, and so on

If you identify particular SQL activities as "runaway" tasks, you can cancel them by "deleting" the appropriate rows from the SQC table. Any selection criteria can be used. While it is convenient, care must be taken to avoid canceling more work than you intend, and proper security provided to restrict access to the SQC table.

For more information, see Processing Details (see page 15).

Examples

Canceling Specific Requests by Accessor ID

Following is an example of canceling SQL requests for a specific accessor ID by issuing an SQL statement to delete the ACCESSOR_ID row from the SYSADM.SQL_STATUS_CURRENT table:

```
DELETE FROM SYSADM.SQL_STATUS_CURRENT
 WHERE ACCESSOR_ID = 'accessorid';
```

Canceling a Specific Transaction

Following is an example of canceling a specific transaction:

```
DELETE FROM SYSADM.SQL_STATUS_CURRENT
 WHERE LUW_BEG_REQ_NBR = 999999;
```

Canceling a Specific Plan

Following is an example of canceling SQL requests for a specific plan:

```
DELETE FROM SYSADM.SQL_STATUS_CURRENT
 WHERE PLAN_AUTH = authid
  AND PLAN_NAME = planname;
```

Canceling a Specific Job

Following is an example of canceling SQL requests for a specific job:

```
DELETE FROM SYSADM.SQL_STATUS_CURRENT
 WHERE JOB_NAME = 'jobname';
```

or, alternatively:

```
DELETE FROM SYSADM.SQL_STATUS_CURRENT
 WHERE RUN_UNIT = 99999;
```

Processing Details

When a delete is processed, the rows in the SQC Dynamic System Table are not actually deleted. Instead, the following steps occur:

1. A flag is set for each corresponding transaction to be aborted when the next row is read by either SQL or Compound Boolean Selection (CBS).
2. When the request is aborted, SQLCODE -128, **LUW ABORTED,** is generated.

3. The application is then given the opportunity to issue a commit or rollback.
4. Until the transaction is ended, all subsequent requests to read rows are also aborted with the SQLCODE -128, but subsequent transactions initiated by the application are *not* affected.

Note: Once the abort flag has been set, there is no way to reset it until the transaction ends. Neither is there a way to report on which transactions have the abort flag set but have not yet ended.

Alternatives to SQC Table Deletes

MUF Canceling of Applications

Periodically, the MUF checks to see if an application job is still active. If the MUF determines that an application job is not active, the abort flag is set to abort any currently running tasks for that job in the MUF. While the interval of time between checking on active jobs by the MUF can vary, it typically occurs approximately every 2 minutes.

REQABORT Command

You can abort a request with the REQABORT command as either a console command or a DBUTLTY function. To use REQABORT, first use a STATUS command to obtain the request sequence number, by which you specify the request you want to abort. A limitation of the REQABORT command is that it only aborts a single currently running request, while canceling SQL requests by using the SQC table not only flags current requests as "to be aborted" but also flags all future requests that read rows within the transaction.

Note: For more information about the REQABORT and STATUS commands, see the *CA Datacom/DB DBUTLTY Reference Guide*.

Data Reporter Accessing Dynamic System Tables

Access is limited to Dynamic System Tables in the following circumstances:

- Excludes tables starting with SQL_ that are only accessible using SQL
- RAAT commands REDKX, REDKG, and REDNX only
- The MUFNAME field or DIRNAME field is forced to the MUF being used and the CXX being used
- CA Datacom/DB Reporting Facility now has access to simple fields for fields defined as TIMESTAMP

Chapter 3: Directory (CXX) Related Tables

Twelve Dynamic System Tables (located in the SYS area, base 1000) have been set up to access the Directory (CXX) related data. This group of tables provides in real time most of the Directory (CXX) information. The data is made available in a relatively stable release-independent (or version-independent) manner.

Note: The CXX table data is from the MUF view. Once the definitions have been accessed, changes done in Single User or another non-MUFplex MUF are not visible until the database is next opened in this MUF.

DIR_AREA (DRA)

One row exists in this table for every area in the Directory (CXX). Each row includes information that exists for that specific area except for the data set information which is in the DIR_DATASET table (see DIR_DATASET (DRZ)). For reporting, the DIR_AREA and DIR_DATASET can be easily joined, as can most of the Dynamic System Tables.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	DATACOM-NAME of the area.
AREA_SQLNAME	CHAR(32)	No	If any table in the area is defined in CA Datacom Datadictionary as having SQL-INTENT, this contains the SQLNAME of the area. It is blank if any table in the area is *not* defined in CA Datacom Datadictionary as having SQL-INTENT. It is also blank after a previous Version to a current Version conversion, until the next catalog of the database.
BINARY_SEARCH	CHAR(1)	No	Contains Y or N. Specifies whether this data area uses the binary search algorithm. Single table areas that do not use compression, and that have a sufficient number of possible rows per block, use a binary search algorithm for speed.

Column Name	SQL Data Type	Nullable	Description
DATA_SPACE_OPTION	CHAR(1)	No	Data space option. **0** No reclamation **1** Random reuse **2** Wrap sequential reuse **3** Clustering reuse **4** Random reuse with OLREORG support **5** Wrap sequential reuse with OLREORG support
DBID	SMALLINT	No	The database identifier.
DIR_LINK	CHAR(1)	No	Specifies whether the CXX link option is in place for this data area. The CXX link option provides an edit to assure an area is being accessed through the correct directory.
DIR_NAME	CHAR(8)	No	Name of the Directory (CXX) used by this MUF.
DYN_EXT_TYPE	CHAR(1)	No	Contains the type of value specified for dynamic extend, blanks if none, C if cylinders, or T if tracks.
DYN_EXT_VALUE	INTEGER	No	Contains the number of tracks or cylinders to request if dynamic extend is selected. Contains a zero if dynamic extend is not selected. When dynamic extend is selected, tracks or cylinders specified here (if not zero) are used during each dynamic extend. If zero here, the VTOC JCL secondary allocation is used. This column is not used during an EXTEND function in DBUTLTY.
DYNAMIC_EXTEND	CHAR(1)	No	Specifies whether dynamic extends are allowed for this data area.
GENERATION	SMALLINT	NO	Value returned is 1 for generation1.

Column Name	SQL Data Type	Nullable	Description
HIGH_URI	DECIMAL(11,0)	Yes	If this area has no loaded table, or if it was loaded with URI=NO, this field is undefined and is set to NULL. For URI, each row in an area is assigned a Unique Row Identifier when it is added or when it is loaded except for a LOAD from a RECID=YES BACKUP. HIGH_URI is the last assigned number.
INSERTS	INTEGER	No	Counter is set to zero during a LOAD. Incremented by one each time a row is inserted by either SQL or record-at-a-time into a table in this area.
LOADED_TIME	TIMESTAMP	Yes	Date and time of area load if any table in the area is loaded. If no loaded table in the area, this field is null.
LOADING	CHAR(1)	No	Specifies whether the area is currently being loaded.
MOVED_RECORDS	INTEGER	No	Number of moved records. Records are moved during an update to a compressed table where the size of stored data is larger and does not fit in the data block. Counter is set to zero during a load. Counter is only an indication of volume of moved records and can range from 0 through 32767 only. You can use this value to help tune SLACK either in the LOAD utility or in the definition of the area in CA Datacom Datadictionary. You can also use it to determine if a reorganize is desirable.
OCCURRENCE	CHAR(32)	No	Contains the CA Datacom Datadictionary entity-occurrence name of this area.
SLACK	SMALLINT	No	Contains the value of SLACK to reserve space in a data block when inserting records. Value is provided through CA Datacom Datadictionary in the area definition. It does NOT relate to the SLACK that can be specified during the LOAD function of DBUTLTY.

Column Name	SQL Data Type	Nullable	Description
STATUS	CHAR(1)	No	Status of the occurrence as provided in CA Datacom Datadictionary. P is production (PROD) status and T is TEST status.
UNLOADING	CHAR(1)	No	Specifies whether the area is currently being unloaded. Unloading reflects that either a BACKUP or an EXTRACT is being done and with either the SEQ=NATIVE, or UPDATE=YES options selected. Other types of BACKUP or EXTRACT are not tracked in the Directory and not available here.
URI	CHAR(1)	Yes	The URI status for an area with no loaded tables is considered unknown and is null. If a loaded table exists, the area is URI or not.
URI_REUSE	CHAR(1)	No	Contains the value (Y or N) specified for the URI-REUSE attribute.
VERSION	SMALLINT	No	Specifies the version of the CA Datacom Datadictionary occurrence for this area.

DIR_COLUMN (DRC)

One record exists in this table for every column in the Directory. Each record includes information that exists for that specific column. Column information exists only for those fields defined to CA Datacom Datadictionary that can contain null values or fields which have the DBEDITS turned on or both. Fields without these options are *not* reflected in this table.

Column Name	SQL Data Type	Nullable	Description
CHANGEABLE	CHAR(1)	No	Indicates whether a change is allowed.

Column Name	SQL Data Type	Nullable	Description
COLUMN_DEFAULT	CHAR(1)	Yes	This column is null if the CA Datacom Datadictionary attribute DBEDITS is set to NO. Otherwise, the column contains: **D** Standard data type defaults. **O** A literal is the default value. **N** No default which indicates that the field must be provided during every insert. **S** Security accessor ID is the default value. **U** The SQL AUTHID is the default value.
COLUMN_LENGTH	INTEGER	No	The length of the column as it is stored in the CA Datacom/DB row. This includes 1 byte if the column can be null plus 2 bytes if the column is variable-length.
COLUMN_OFFSET	INTEGER	No	The displacement or offset of this column from the start of the row.
COLUMN_SEQ	SMALLINT	No	Sequence number of this column within the columns of the row. The first physical column in the row (which is nullable or has DBEDITS) has a sequence number of 1, the second 2, and so on.
COLUMN_SIGN	CHAR(1)	Yes	This column is null if the field is not one of the numeric data types. Otherwise, this column specifies whether this column was defined as signed numeric.

Column Name	SQL Data Type	Nullable	Description
COLUMN_TYPE	CHAR(1)	Yes	Column type is null if the CA Datacom Datadictionary DBEDITS attribute is set to NO. Otherwise, this field represents the CA Datacom Datadictionary field type. **N** Numeric **D** Decimal **L** Long float **S** Short float **E** Extended float **B** Binary **W** Variable graphics **G** DBCS **C** Character **V** Variable character
COLUMN_VALUE_CHAR	CHAR(20)	Yes	This field is null if the column did not have the CA Datacom Datadictionary DBEDITS attribute set YES or the default to be other than literal. Otherwise, this is the character form of the data placed in the column. For variable-length fields the first two positions is the length of the following data moved to the field.
COLUMN_VALUE_LEN	SMALLINT	Yes	This field is null if the column did not have the CA Datacom Datadictionary DBEDITS attribute set YES or the default to be other than literal. Otherwise, this is the length of the literal that is placed in the column. If this is a variable character column, then this length includes 2 bytes for the length of the data.

Column Name	SQL Data Type	Nullable	Description
DBEDITS	CHAR(1)	No	Indicates whether the field was defined with the CA Datacom Datadictionary DBEDITS attribute on or off.
DBID	SMALLINT	No	The database identifier of the database containing the table containing this column.
DIR_NAME	CHAR(8)	No	Name of the Directory that the MUF is using.
FORCE_ADD	CHAR(1)	No	Indicates whether CA Datacom/DB is to force new data to this field when inserting a row.
FORCE_UPDATE	CHAR(1)	No	Indicates whether CA Datacom/DB is to force new data to this field when updating a row.
NULLABLE	CHAR(1)	No	Indicates whether the column can be set to NULL.
OCCURRENCE	CHAR(32)	No	Contains the CA Datacom Datadictionary entity occurrence name of this column.
SEMANTIC_TYPE	CHAR(8)	Yes	Set to NULL if this column was not defined to CA Datacom Datadictionary with the DBEDITS set to YES or if the column has no semantic data type. Otherwise, it can contain one of the following: **DATE** SQL type date **TIME** SQL type time **TIMESTMP** SQL type timestamp **MIXED** Mixed case data **BIT-DATA** For-bit-data

Column Name	SQL Data Type	Nullable	Description
SQL_TYPE	CHAR(8)	Yes	Set to NULL if this column was not defined to CA Datacom Datadictionary with the DBEDITS set to YES or if this column cannot be represented by a data type recognized by SQL. Otherwise, it can contain one of the following: **NUMERIC** Zone decimal (signed or unsigned) **DECIMAL** Packed decimal (signed or unsigned) **FLOAT** Floating-point numbers **SQL-DATE** SQL DATE **SQL-TIME** SQL TIME **SQL-STMP** SQL TIMESTAMP **SMALLINT** 2-byte binary signed number **INTEGER** 4-byte binary signed number **CHAR** Character field **VARCHAR** Variable character field **VARG** Variable graphics field **GRAPHIC** Graphics field
STATUS	CHAR(1)	No	Status of the CA Datacom Datadictionary occurrence name for this column.
TABLE_NAME	CHAR(3)	No	Contains the DATACOM-NAME attribute-value of the table containing this column.
VERSION	SMALLINT	No	Version of the CA Datacom Datadictionary occurrence name for this column.

DIR_DATABASE (DRB)

One row exists in this table for every database in the Directory. Each row includes information for that specific database except for the data set information which is in the DIR_DATASET table with an area name of IXX.

Note: Changes in the Version 12.0 DIR_DATABASE Dynamic System Table reflect that the information is for the IXX data set, not all of the index data sets. Columns that were removed from DIR_DATABASE in Version 12.0 were related to index information and can now be found in the DIR_INDEX (DRI) (see page 34) table.

Column Name	SQL Data Type	Nullable	Description
BASE_FORMAT	SMALLINT	No	Column contains the current database format for this DBID. Formats 1 and 2 were used by prior releases. A base open by Version 15.0 is set to Format 3.
DBID	SMALLINT	No	The database identifier.
DIR_NAME	CHAR(8)	No	The name of the Directory (CXX) being used.
MAINTENANCE_TIME	TIMESTAMP	Yes	Date and time that the database definition was last cataloged, or was maintained using the CXXMAINT options except the ALTER option.
OCCURRENCE	CHAR(32)	No	The entity-occurrence name for the database from CA Datacom Datadictionary.
OPEN_TIME	TIMESTAMP	Yes	Date and time of the last User Requirements Table open for a table in this database. If the database is currently open for read-only, the date and time can reflect a previous open for update. If the table was opened by multiple tasks concurrently, the date and time can reflect any of the tasks. Can be null if a new database is cataloged and no User Requirements Tables have been opened.
OPEN_UPDATE_TIME	TIMESTAMP	Yes	Date and time of the last User Requirements Table open for update. If the table was opened by multiple tasks concurrently, the date and time can reflect any of the tasks. Can be null if a new database is cataloged and no User Requirements Tables have been opened.

Column Name	SQL Data Type	Nullable	Description
PARTITION	CHAR(1)	No	Indicates whether the database is partitioned. See the *CA Datacom STAR Administrator Guide* for details about partitioning.
READ_ONLY_USERS	CHAR(1)	No	Reflects whether any of the tables in the database are currently open with an UPDATE=NO User Requirements Table.
REPLICA	CHAR(1)	No	Indicates whether the database can be replicated. See the *CA Datacom STAR Administrator Guide* for details about replicating.
RESTART_TIME	TIMESTAMP	Yes	Is null if a new database is cataloged and the RESTART processing at MUF enable has not occurred for this database nor has the RESET function of DBUTLTY been done for this base (when the database was open). Otherwise, it is the date and time of the last RESTART processing or RESET function for this base.
SQL_SECURITY	CHAR(1)	No	Indicates whether SQL Security has been selected for the tables in this database. For more information about security, see the *CA Datacom Security Reference Guide*.
STATUS	CHAR(1)	No	Specifies the status of the occurrence of the database CA Datacom Datadictionary definition. P is PROD status and T is TEST status.
UPDATE_OWNER	CHAR(8)	Yes	If this DBID is open for update, this is the job name from the last open for update. If this DBID is not open for update, this column is null.
UPDATE_USERS	CHAR(1)	No	Specifies whether the database is currently open for update.
VERSION	SMALLINT	No	Specifies the version of the occurrence of the database CA Datacom Datadictionary definition.

DIR_DATASET (DRZ)

One row exists for the Directory (CXX), one for every index (IXX), and one for every data area. Each row includes information that exists for that specific data set.

In many cases, this table can be used alone. In other cases, you can join the DIR_DIRECTORY table with the CXX row, the DIR_DATABASE table with its IXX row for that database, and each DIR_AREA table row with its area name and database.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	The name of the area (CXX, IXX, or data area).
BLOCK_LENGTH	SMALLINT	No	The length of each block in this area.
DATASET_NAME	CHAR(44)	Yes	This field is null if the data set name is not known. This can occur for new databases before the areas are initialized, or the database loaded from a Directory (CXX) backup and the database is new, or for areas which have not been initialized for several versions of CA Datacom/DB. Otherwise, the data set name is present. This field contains *virtual* if this database is defined as virtual.
DBID	SMALLINT	No	The database identifier (ID) containing this area. For the Directory (CXX) row, this field is zero.

Column Name	SQL Data Type	Nullable	Description
DEVICE	CHAR(6)	Yes	This field is null if the area has not been initialized since a catalog of a new database, or the load of a new database from a CXX backup. Otherwise, it contains a name representing the type of device CA Datacom/DB is using to determine device geometries, that is, records per track and tracks per cylinder. **Note:** The virtual areas are treated as if they reside on 3380 devices. For example, an actual physical Index Area (IXX) might be on a 3390 device. However, CA Datacom/DB treats it as if it resides on a 3380 device during the run of the MUF and reports it that way in the DIR_DATASET (DRZ) Dynamic System Table.
DIR_NAME	CHAR(8)	No	Name of the Directory being used by this MUF.
GENERATION	SMALLINT	No	Value returned is 1 for generation 1.
IN_USE_BLOCKS	INTEGER	No	Can be zero if the area is not initialized. Contains a count of the number of blocks that contains active information. If a system failure has occurred during maintenance processing, this number can be inaccurate. In CA Datacom/DB reporting, IN_USE_BLOCKS and TOTAL_BLOCKS are used to calculate percent full.
MAX_USED_BLOCKS	INTEGER	No	This field can be zero if the area has not been initialized. Otherwise, this is a high water mark as to how full the area has been. In CA Datacom/DB reporting, MAX_USED_BLOCKS and TOTAL_BLOCKS are used to calculate maximum percent full.

Column Name	SQL Data Type	Nullable	Description
PARTIAL_BLOCKS	INTEGER	No	This field is zero in the rows for the Directory (CXX) and Index Area (IXX), if the area is not initialized, or if space option 0 (no reclamation) is selected. Otherwise, this field contains the number of blocks that probably have space to add one or more records. In compressed tables and areas with multiple tables of different lengths, for a specific add, space may not exist. In support of transaction backout, some space is not actually available until a transaction commit. If a system failure has occurred during maintenance processing, this number can be inaccurate.

Column Name	SQL Data Type	Nullable	Description
SECONDARY_CYLS	SMALLINT	Yes	The secondary allocation is only reported in cylinders. The number reported is a maximum size of 64k-1 (65535). This number reflects size 65535 and all larger sizes. The size reported was the last time the area was opened when the base was opened for update. During a dynamic extend, the size is subject to the data set having a specific JCL DD statement with a different secondary. The size is subject to a CXX BACKUP function and restored with the base restore. The information is not disabled if input into a back level of CA Datacom and then subject to a backup there and restored into a 15.0 CXX. If the information is not currently correct, then it is corrected the next open for update of the data set. If the allocation was done in average block then the secondary size from the JCL is converted into tracks (rounding down) and then into cylinders (rounding down). If the allocation was done in tracks then the secondary size from the JCL is converted into cylinders (rounding down). Therefore, a value reported as 0 might mean no secondary or a secondary of a size less than a full cylinder. This field may be null if: ■ The data set has not been opened in the past with 15.0 code installed. ■ It was opened without the database being opened for update. ■ The area was opened in MUF and defined as Virtual. ■ The area was opened for update but was defined in the JCL as a 'TEMP' data set (DSN starting with &&).
TOTAL_BLOCKS	INTEGER	No	This field may be zero if the area has not been initialized. Otherwise, it contains the total number of blocks in the area.

Column Name	SQL Data Type	Nullable	Description
TOTAL_TRACKS	INTEGER	No	This field may contain zeros for an area not initialized. Otherwise, it contains the number of tracks allocated and accepted by CA Datacom/DB for use.

DIR_DIRECTORY (DRD)

A single row exists in this table to provide information about the Directory in general. Also Directory information is provided with one row in the DIR_DATASET table with a database of zero and an area name of CXX.

Column Name	SQL Data Type	Nullable	Description
AREAEV	CHAR(1)	No	This column contains the current setting for Dataset Extent Validation, Y or N. This can be changed by running CXXMAINT.
CURRENT_DATETIME	TIMESTAMP	No	Current timestamp from the MUF where the request as processed.
CXX_ENQ	CHAR(8)	No	This column contains the level of CXX locking in effect, either LOCAL or SYSPLEX. This can be changed by running CXXMAINT.
CXX_LEVEL	SMALLINT	No	The CXX compatibility level.
CXX_VERSION	CHAR(6)	No	The directory's version, for example '14.0' (that is 14.0 plus two blanks to make six characters). Note that Version 15.0 runs with a CXX_VERSION setting at 14.0.
DATACOM_TYPE	CHAR(2)	No	Type of CA Datacom/DB Directory being processed. DB for full CA Datacom/DB or AD for a restricted CA Datacom/AD Directory.
DATAFS	CHAR(1)	No	This column contains the current setting for Data Fast Search, Y or N. This can be changed by running CXXMAINT.

Column Name	SQL Data Type	Nullable	Description
DATAHU	CHAR(1)	No	This column contains the current setting for the Data High Used Mark, Y or N. This can be changed by running CXXMAINT.
DBCS_DEFAULT_MIXED	CHAR(1)	No	Specifies a default that character fields are assumed to be Single Byte Character Set (S) or Mixed characters (M) allowing shift out/in characters.
DBCS_SHIFT_CODES	CHAR(1)	No	Specifies the type of shift codes embedded in the mixed character fields. **I** Represents z/OS **F** Represents Fujitsu MSP **N** Means that the DBCS feature has been disabled in DBUTLTY in CXXMAINT OPTION=ALTER,DBCS= by a value of DBCS=OFF.
DIR_NAME	CHAR(8)	No	Name of the Directory used by this MUF.
GENERATION	SMALLINT	No	Value returned is 1 for generation 1.
HIGH_RUN_UNIT	INTEGER	No	Either the last run unit number assigned to an executing CA Datacom/DB job or reserved for a job.
LANGUAGE_SUPPORT	CHAR(1)	No	K represents that Katakana language support has been enabled. A blank represents no Katakana language support.
PRIMARY_DD_DBID	SMALLINT	No	The database identifier of the primary CA Datacom Datadictionary containing the DATA-DICT database and the High-Speed Directory. The DBID is specified as the first value of the Multi-User startup option DICTIONARY.

Column Name	SQL Data Type	Nullable	Description
PRIMARY_DDD_DBID	SMALLINT	No	The database identifier of the Data Definition Directory (DDD) used in SQL processing and by products using SQL. The DBID is specified as the second value of the Multi-User startup option DICTIONARY.
SECURE_JOBNAME	CHAR(1)	No	This column is provided for compatibility with prior versions.
SECURE_SINGLE_USER	CHAR(1)	No	Indicates whether you can run in a Single User environment with security installed. See the *CA Datacom/DB Database and System Administration Guide* for more information about this topic.
SIMPLIFY_MODE	CHAR(1)	No	This column contains a Y if the CXX was initiated with SIMPLIFY mode as Yes, otherwise, it is N.
SQL_SUPPORT	CHAR(1)	No	Indicates if a MUF with Multi-User startup option SQLOPTION has executed.

DIR_ELEMENT (DRE)

One record exists in this table for every element in the Directory. Each record includes information that exists for that specific element.

Column Name	SQL Data Type	Nullable	Description
DBID	SMALLINT	No	Identifies the database containing this element.
DIR_NAME	CHAR(8)	No	Name of the Directory used by the MUF.
ELEMENT_LENGTH	INTEGER	No	Contains the length of the element.
ELEMENT_NAME	CHAR(5)	No	Contains the short name (DATACOM-NAME attribute-value) assigned to the element.

Column Name	SQL Data Type	Nullable	Description
ELEMENT_OFFSET	INTEGER	No	Contains the offset or displacement from the start of the row to the start of this element.
OCCURRENCE	CHAR(32)	No	Contains the CA Datacom Datadictionary entity-occurrence name for this element.
SECURITY_CODE	SMALLINT	No	The element security code in numeric form.
STATUS	CHAR(1)	No	Status of the CA Datacom Datadictionary entity-occurrence for this element.
TABLE_NAME	CHAR(3)	No	Contains the DATACOM-NAME of the table containing this element.
VERSION	SMALLINT	No	Version of the CA Datacom Datadictionary entity-occurrence for this element.

DIR_INDEX (DRI)

One row exists for this table for every Index Area in the Directory (CXX). Each row includes information that exists for that specific Index Area, except for the data set information that is in the DIR_DATASET table. Changes to the DIR_DATASET (DRZ) table in Version 12.0 provide information about multiple index data sets. For reporting, the DIR_INDEX and the DIR_DATASET tables can be easily joined, as can most of the Dynamic System Tables.

The DIR_INDEX contains the following columns:

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Datacom name of this index area, such as IXX for standard indexes or other values for partitioned indexes.
DBID	SMALLINT	No	The database identifier (database ID).
DIR_LINK	CHAR(1)	No	Specifies whether the CXX link option is in place for this Index Area. The CXX link option provides an edit to help ensure that an area is being accessed through the correct Directory (CXX).

Column Name	SQL Data Type	Nullable	Description
DIR_NAME	CHAR(8)	No	Name of the Directory (CXX) used by this MUF.
DYNAMIC_EXTEND	CHAR(1)	No	Specifies whether dynamic extends are allowed for this Index Area.
DYN_EXT_TYPE	CHAR(1)	No	Contains the type of value specified for dynamic extend. Blank if none, C if cylinders, or T if tracks.
DYN_EXT_VALUE	INTEGER	No	Contains the number of tracks or cylinders to request if dynamic extend is selected. Contains a zero if dynamic extend is not selected. When dynamic extend is selected, tracks or cylinders specified here (if not zero) are used during each dynamic extend. If zero here, the VTOC JCL secondary allocation is used. This field is not used during an EXTEND function in DBUTLTY.
INDEX_LEVELS	SMALLINT	No	The count of the number of high-level index levels.
OLDEP	CHAR(1)	No	Contains a Y or N to indicate whether this Index Area was initiated with OLDEP (old entry point) support. The index works more efficiently with OLDEP N.

DIR_KEY (DRK)

One row exists in this table for every key in the Directory. Each row includes information that exists for that specific key.

Column Name	SQL Data Type	Nullable	Description
AVERAGE_L0_CHGS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data block changes in a DXX level index block. Contains zero unless CBS has performed certain types of selection, then is initialized as 204, but could possibly be set to a more accurate number if necessary and enough DXX level index blocks exist for this key.

Column Name	SQL Data Type	Nullable	Description
AVERAGE_L0_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data records in a DXX level index block. Contains zero unless CBS has performed certain types of selection, then is initialized as 300, but could possibly be set to a more accurate number if necessary and enough DXX level index blocks exist for this key.
AVERAGE_L1_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data records represented in a level one IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 200, but could possibly be set to a more accurate number if CBS determines it necessary.
AVERAGE_L2_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data records represented in a level two IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 200, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 200 in an index without two levels.
AVERAGE_L3_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data records represented in a level three IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 200, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 200 in an index without three levels.
AVERAGE_L4_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data records represented in a level four IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 200, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 200 in an index without four levels.

Column Name	SQL Data Type	Nullable	Description
AVERAGE_L5_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Average data records represented in a level five IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 200, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 200 in an index without five levels.
CBS_IGNORE	CHAR(1)	No	Indicates whether CBS is allowed to use this key.. Column is provided for prior release compatibility. See the KEY_USE column.
CURRENT_RECORDS	INTEGER	No	The number of records indexed by this key. If a system failure has occurred during maintenance, this number can be inaccurate.
DBID	SMALLINT	No	The database ID of the database for the table that contains this key field.
DIR_NAME	CHAR(8)	No	Name of the Directory used by the MUF.
DIRECT	CHAR(1)	No	Indicates whether this key is a direct record access key definition.
DUPLICATE_MASTER	CHAR(1)	Yes	This field is null if this is not the master key. For a master key, this field indicates if duplicate master keys are permitted.
GENERATION	SMALLINT	No	Value returned is 1 for generation 1.
INDEX_NAME	CHAR(3)	No	This column contains the index name. It is either IXX (the default) or an index area name when using Multi-Dataset Indexes.
KEY_ID	SMALLINT	No	The key ID of this key.
KEY_LENGTH	SMALLINT	No	Total length of the key to index.
KEY_NAME	CHAR(5)	No	The name of this key.

Column Name	SQL Data Type	Nullable	Description
KEY_STATUS	CHAR(8)	No	This column contains status information for the key. More than one status can be listed for a given key as appropriate. Status codes are as follows: **AD** AD signifies the key was added through an APPLYCXX but the process of populating the index which makes the key usable is not complete. **DL** DL indicates that the key was deleted through an APPLYCXX but the process of removing the key from the index is not complete. **LD** LD signifies that the key is loaded and ready for use. **NL** NL signifies that the key is not loaded or available to the user.
KEY_SQLNAME	CHAR(32)	No	The SQL name for this key. It contains blanks if the table is not SQLable, or if the database has not been cataloged since installation of Release 9.0.
KEY_USE	CHAR(8)	No	Contains the value specified for the KEY_USAGE attribute. Possible values are: ■ ANY ■ RAAT ■ CBS/SQL ■ NONE
MASTER	CHAR(1)	No	Indicates whether this key is the master key for this table.
NATIVE	CHAR(1)	No	Specifies whether this key is the native sequence key for this table.
NIL	CHAR(1)	No	Specifies whether this key is to have nil values placed in the index. Nil represents low values or blanks.

Column Name	SQL Data Type	Nullable	Description
OCCURRENCE	CHAR(32)	No	The CA Datacom Datadictionary entity-occurrence name for this key definition.
PARENT_DBID	SMALLINT	Yes	This field is null if this key is not subject to an SQL referential integrity constraint. If this is a FOREIGN key, this field contains the database identifier of database for the table which is referenced.
PARENT_TABLE_NAME	CHAR(3)	Yes	This field is null if this key is not subject to an SQL referential integrity constraint. If this is a FOREIGN key, this field contains the short name (DATACOM-NAME attribute-value) of the table which is referenced.
STATUS	CHAR(1)	No	Status of the CA Datacom Datadictionary entity-occurrence for this key.
TABLE_NAME	CHAR(3)	No	Contains the short name (DATACOM-NAME attribute-value) of the table containing key field.
UNIQUE_KEY	CHAR(1)	No	Specifies whether this key is subject to the unique key constraint.
UNIQUE_SEARCHED	INTEGER	No	Number of rows indexed when the unique counts in this keys fields was generated. This is done during a LOAD, RETIX (except KEYNAME=*data), and REPORT ...,TYPE=G,UPDATE=YES. **Note:** If a key is dynamically added through an APPLYCXX, these are updated after the key entries for the new key are added to the index.
UNIQUE_TIME	TIMESTAMP	Yes	This field can be null if a LOAD, RETIX (except KEYNAME=*DATA) or REPORT ...,TYPE=G,UPDATE=YES has not been done since this key was defined or added new to the Directory. Otherwise, it is the date when the index was scanned and the unique count fields were last computed. **Note:** If a key is dynamically added through an APPLYCXX, these are updated after the key entries for the new key are added to the index.

Column Name	SQL Data Type	Nullable	Description
USES_RAAT	INTEGER	No	A count of initial locate/read record-at-a-time commands against this key definition. The count is incremented by one for every LOC/RED/RDU/CNT command with KY/KG/KR/KL/KX/KI or GSETL command. This counter stops being incremented at one less than two gigs until reset to zero during the next LOAD, RETIX, or CXXMAINT with OPTION=ALTER,OPTION2= RESET_KEY_USES.
USES_SAAT	INTEGER	No	Provides a general key usage count, the times a SELFR command selected this key for a traversal set. This does *not* include population counting or key usage to determine if the key is used in the completion of the request. This count does *not* include use of the command by SQL. This counter stops being incremented at one less than two gigs until reset to zero during the next LOAD, RETIX, or CXXMAINT with OPTION=ALTER,OPTION2= RESET_KEY_USES.
USES_SQL	INTEGER	No	Provides a general key usage count, the times a SELFR command selected this key for a traversal set. This does *not* include population counting or key usage to determine if the key is used in the completion of the request. This count *only* includes use of the command by SQL. This counter stops being incremented at one less than two gigs until reset to zero during the next LOAD, RETIX, or CXXMAINT with OPTION=ALTER,OPTION2= RESET_KEY_USES.
VARIANCE_L0_CHGS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a DXX level index block. Contains zero unless CBS has performed certain types of selection, then is initialized to zero, but could possibly be set to a more accurate number if CBS determines it necessary.

Column Name	SQL Data Type	Nullable	Description
VARIANCE_L0_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a DXX level index block. Contains zero unless CBS has performed certain types of selection, then is initialized to zero, but could possibly be set to a more accurate number if CBS determines it necessary.
VARIANCE_L1_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a level one IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 20, but could possibly be set to a more accurate number if CBS determines it necessary.
VARIANCE_L2_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a level two IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 20, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 20 in an index without two levels.
VARIANCE_L3_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a level three IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 20, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 20 in an index without three levels.

Column Name	SQL Data Type	Nullable	Description
VARIANCE_L4_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a level four IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 20, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 20 in an index without four levels.
VARIANCE_L5_RIDS	SMALLINT	No	Internal CBS information provided as a possible assist in support issues. Variance in the average data records represented in a level five IXX index block. Contains zero unless CBS has performed certain types of selection, then is initialized to 20, but could possibly be set to a more accurate number if CBS determines it necessary. Remains 0 or 20 in an index without five levels.
VERSION	SMALLINT	No	Version of the CA Datacom Datadictionary entity-occurrence for this key.

DIR_KEY_FIELD (DRF)

One record exists in this table for every key field in the Directory. Each record includes information that exists for that specific key field.

Column Name	SQL Data Type	Nullable	Description
DBID	SMALLINT	No	Identifies the database containing this key field.
DIR_NAME	CHAR(8)	No	Name of the Directory used by this MUF.
FIELD_DECIMAL	CHAR(1)	No	Indicates whether this field is Packed Decimal type sensitive.
FIELD_FLOAT	CHAR(1)	No	Indicates whether this field is FLOAT data type sensitive.
FIELD_LENGTH	SMALLINT	No	Contains the length of this key field.

Column Name	SQL Data Type	Nullable	Description
FIELD_NULLABLE	CHAR(1)	No	Indicates whether this field can be null.
FIELD_NUMERIC	CHAR(1)	No	Indicates whether this field is numeric (zoned decimal) data type sensitive.
FIELD_OFFSET	INTEGER	No	Offset or displacement from the start of the row to the start of this field.
FIELD_ORDER	CHAR(1)	No	Order of the field with either an A for ascending or a D for descending.
FIELD_SIGN_INT	CHAR(1)	No	Indicates whether this field is a signed integer (binary number) data type sensitive.
FIELD_UNIQUE	INTEGER	No	Number of rows indexed to be unique assuming all fields of this key are from the left through this key field. This field can be zero for a new table or key. The value is set during the LOAD, RETIX (not if KEYNAME=*DATA), or REPORT ...TYPE=G,UPDATE=YES.
KEY_FIELD_SEQ	SMALLINT	No	Sequence number (1—n) of this field within this key definition.
KEY_NAME	CHAR(5)	No	The short name (DATACOM-NAME attribute-value) of the key containing this field.
OCCURRENCE	CHAR(32)	No	Contains the CA Datacom Datadictionary entity-occurrence name for this field.
STATUS	CHAR(1)	No	Contains the CA Datacom Datadictionary status of this field.
TABLE_NAME	CHAR(3)	No	Contains the short name (DATACOM-NAME attribute-value) of the table containing key field.
VERSION	SMALLINT	No	Contains the version of the CA Datacom Datadictionary entity-occurrence for this key field.

DIR_PART_COLUMNS (DRP)

This table contains information about partitioning columns.

Column Name	SQL Data Type	Nullable	Description
COLUMN_LENGTH	INTEGER	No	Contains the column length being used. It is the entire field as stored in the data row.
COLUMN_OFFSET	INTEGER	No	Contains the column offset into the row, starting from zero.
COLUMN_SEQ	SMALLINT	No	Contains the column sequence within partitioning columns.
DATA_FORMAT	CHAR(2)	No	Contains the format of the data, where: **CH** Indicates character. **DT** Indicates an SQL DATE. **TM** Indicates an SQL TIME. **TS** Indicates an SQL TIMESTAMP. **NM** Indicates numeric.
DBID	SMALLINT	No	Indicates the database ID containing the partitioned table.
DIR_NAME	CHAR(8)	No	Name of the Directory (CXX) used by this MUF.
FULL_TABLE	CHAR(3)	No	Provides the table name of the full table.

Column Name	SQL Data Type	Nullable	Description
HIGH_LENGTH	SMALLINT	No	Contains the length of the highest value to reside in this partition. A length of zeros indicates the concept of high-value. Must match COLUMN_LENGTH when not character. If character, this is the length used to compare, such as a 100 byte-character column, but only the first 25 bytes are compared.
HIGHEST_VALUE	CHAR(60)	Yes	Contains a character representation of the highest value to reside in this partition when the format is not character. When it is character, this contains the actual high value and so might not be printable as is. Character data is left justified. Numeric data is right-justified. If numeric data is signed, the rightmost position contains a minus sign for negative or a blank for positive.
OCCURRENCE	CHAR(32)	No	Contains the CA Datacom Datadictionary occurrence name for the field.
PARTITIONS	SMALLINT	No	Contains the number of partitions for this Parent table.
TABLE_NAME	CHAR(3)	No	Contains the table name of the partition.
TABLE_NEXT	CHAR(3)	No	Contains the table name of the next partition of this Parent table, in data value order.
TABLE_ORDER	SMALLINT	No	Indicates the table order within partitions. Specifies that this is the first partition, or the second, and so on.
TABLE_PREV	CHAR(3)	No	Contains the table name of the previous partition of this Parent table, in data value order.

DIR_TABLE (DRT)

One record exists in this table for every table in the Directory. Each record includes information that exists for that specific table.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	The short name (DATACOM-NAME attribute-value) of the area containing the table.
ANY_AUTHID	CHAR(32)	Yes	If this is a partition, this is the Any Parent SQL AUTHID.
ANY_SQLNAME	CHAR(32)	Yes	If this is a partition, this is the Any Parent SQL name.
ANY_TABLE	CHAR(3)	Yes	If this is a partition, this is the Any Parent name.
CHILD_RESTRICTED	CHAR(1)	Yes	If this table is partitioned, this column indicates whether users can access the Child table directly, Y or N. If this table is not partitioned, this column is null.
CHNG_MASTER_KEY	CHAR(1)	No	Indicates whether the Master Key can be changed during update processing.
CLUSTER_KEY_ID	SMALLINT	No	Specifies the Cluster Key ID. (If clustering by table, the Cluster Key is 0.)
CLUSTER_KEY_LEN	SMALLINT	No	Specifies the Cluster Key length for the table. (If clustering is by table, the Cluster Key length is 3.)
COMPRESS_EXIT	CHAR(8)	No	Name of the user compression routine. Blanks if no user compression for this table.
COMPRESSION	CHAR(1)	No	Indicates whether CA Datacom/DB compression is active in the table.
CURR_CONSTRAINT_ER	CHAR(1)	No	Indicates whether this table currently has a constraint error.

Column Name	SQL Data Type	Nullable	Description
CURR_FWD_RCV_PEND	CHAR(1)	No	Indicates whether this table currently is in check pending status due to a forward recovery utility execution.
CURR_LOAD_PEND	CHAR(1)	No	Indicates whether this table currently is in check pending status due to a load utility execution.
CURR_RELATED_PEND	CHAR(1)	No	Indicates whether this table currently is in a check pending status due to a related table being in check pending status.
DBID	SMALLINT	No	Specifies the database identifier of the database containing this table.
DD_DBID	SMALLINT	No	Contains the database identifier of the CA Datacom Datadictionary that last cataloged this definition to the Directory (CXX). The field can be zeros for bases built during a CXX load.
DEFINITION_POSTED	CHAR(1)	No	Contains a C if the definition was last posted to the Directory (CXX) with a CA Datacom Datadictionary CATALOG or an SQL CREATE TABLE. Contains a D if the definition was last posted to the CXX with a DBUTLTY CXXMAINT OPTION=DDPROD. Contains a blank if the definition was last posted to the Directory (CXX) with a DBUTLTY LOAD function.
DIR_NAME	CHAR(8)	No	Name of the Directory (CXX) used by this MUF.
DL1T_CONSTRAINT	CHAR(1)	No	Indicates that CA Datacom DL1 Transparency constraints are in effect for this table. See the *CA Datacom DL1 Transparency User Guide* for more information.
DOMAIN_CONSTRAINT	CHAR(1)	No	Indicates that CA Datacom/DB domain constraints are in effect for this table.

Column Name	SQL Data Type	Nullable	Description
DUP_MASTER_KEY	CHAR(1)	No	Indicates whether duplicate Master Key values are allowed.
ENCRYPTION_KEY	CHAR(8)	No	Encryption key provided through CA Datacom Datadictionary for the table when using user compression with the CA Datacom Presspack feature.
ENCRYPTION_METHOD	CHAR(1)	No	Indicates the method of encryption to use. Contains one of the following: - Blank - No encryption - A - AES-128 - B - AES-191 - C - AES-256
ENCRYPTION_TYPE	CHAR(1)	No	Indicates the type of encryption to use. Contains one of the following: - Blank - No encryption - B - Basic encryption performed by CA Datacom/DB with no key management
FULL_AUTHID	CHAR(32)	Yes	If this is a partition, this is the Full Parent SQL AUTHID.
FULL_SQLNAME	CHAR(32)	Yes	If this is a partition, this is the Full Parent SQL name.
FULL_TABLE	CHAR(3)	Yes	If this is a partition, this is the Full Parent name.
GENERATION	SMALLINT	No	Value returned is 1 for generation 1.
HIST_CONSTRAINT_ER	CHAR(1)	No	Indicates whether data in the row or rows violated an integrity constraint or constraints, but was confirmed with the force option.
HIST_FORCED_OFF	CHAR(1)	No	Indicates whether a constraint condition was forced off.

Column Name	SQL Data Type	Nullable	Description
HIST_FWD_RCV_PEND	CHAR(1)	No	Indicates whether the table was in check pending status due to forward recovery when the table was confirmed with the force option.
HIST_LOAD_PEND	CHAR(1)	No	Indicates whether the table was in check pending status due to a LOAD utility when the table was confirmed with the force option.
HIST_RELATED_PEND	CHAR(1)	No	Indicates whether the table was in check pending status due to a related table when this table was confirmed with the force option.
INDEX_LOADED	CHAR(1)	No	Indicates whether the Index is loaded for data in this area.
KEY_ELEMENT_LENGTH	INTEGER	No	Number of bytes of memory that is required to contain this tables key, element and any column information.
LOGOPTION	CHAR(1)	No	Indicates whether logging is active for the table.
MOVE_TO_NEW_PART	CHAR(1)	Yes	If this table is partitioned, this column indicates whether an update is allowed to move a row to a different partition, Y or N. If this table is not partitioned, this column is null.
NEW_DBID	CHAR(1)	No	Indicates whether the DBID was changed using the NEWDBID option during a LOAD AREA=CXX or CXXMAINT OPTION=DDPROD utility.
OCCURRENCE	CHAR(32)	No	CA Datacom Datadictionary occurrence name for this table.
PARTITION_TABLE	CHAR(1)	No	Contains a Y if this table is a partition, or contains an N if this table is not a partition.
PIPELINE	CHAR(1)	No	Indicates whether the pipeline maintenance option is active for this table.

Column Name	SQL Data Type	Nullable	Description
RECORDS	INTEGER	No	Number of records in table. If a system failure has occurred during add/delete processing, this number can be inaccurate.
RECORD_LENGTH	INTEGER	No	The length of the user data. Does *not* include the CA Datacom/DB Record Control Element (RCE) of 12 (or 4) bytes.
RECOVERY	CHAR(1)	No	Indicates whether the recovery option is activated for this table.
REFERENCES_IN_BASE	CHAR(1)	Yes	Is set to null if the table is not defined with referential integrity to another table. If the table is referenced or referencing, then this field indicates if all relationships to this table exist in this same database.
SQL_AUTHID	CHAR(32)	No	Contains the SQL AUTHID (authorization ID) of this table. Contains blanks if the table has no SQL access.
SQL_INTENT	CHAR(1)	No	Specifies the status of the SQL: ■ **Y** - if the table has been defined with full SQL access ■ **R** - if the table is SQL read-only ■ **N** - if no SQL access is allowed ■ *** (asterisk)** - if unknown because the structure was cataloged before the SQL INTENT option was supported.
STATUS	CHAR(1)	No	Specifies the status of the CA Datacom Datadictionary occurrence which defined this table.
TABLE_CONSTRAINED	CHAR(1)	No	Indicates whether SQL has constrained this table.
TABLE_ID	SMALLINT	No	Internal table identifier for this table.
TABLE_LOADED	CHAR(1)	No	Indicates whether the table is loaded.

Column Name	SQL Data Type	Nullable	Description
TABLE_NAME	CHAR(3)	No	Contains the DATACOM-NAME attribute-value of the table.
TABLE_REFERENCED	CHAR(1)	No	Indicates whether the table references another table (is the subject of a relationship with another table).
TABLE_REFERENCING	CHAR(1)	No	Indicates whether the table is referenced by another table (the object of a relationship with another table).
TABLE_SQLNAME	CHAR(32)	No	The SQL name for this table if it is SQL accessible. It contains blanks if the table is not SQL accessible.
TABLE_TRIGGER	CHAR(1)	No	Indicates whether the table has one or more SQL triggers.
VERSION	SMALLINT	No	The version of the CA Datacom Datadictionary occurrence defining this table.

DIR_VOLUMES (DRV)

Volume information is NOT stored in the Directory (CXX) however, the data set name is known most of the time in z/OS and also in z/VSE when CA-DYNAM/D is in use. In the case where the data set name is known, the z/OS catalog or CA-DYNAM/D catalog is queried to get a list of volume serial numbers for cataloged areas. Each known volume serial number for each data set is represented in this table with a unique record.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Contains CXX for the Directory row, IXX for each Index Area row, or the data area name for each data area row.
DBID	SMALLINT	No	Contains zero for the Directory row, else contains the database ID where the area is contained.
DIR_NAME	CHAR(8)	No	Contains the name of the Directory used by the MUF.
VOLSER	CHAR(6)	No	Volume serial number which contains one or more extents of this area.

Column Name	SQL Data Type	Nullable	Description
VOLUME_SEQ	SMALLINT	No	Within one area, contains the sequence number of this volume serial. The volume containing the first extent (or set of extents) has a sequence number of 1. The volume (if any) containing the next extent (or set of extents) has a sequence number of 2, and so forth.

Chapter 4: Multi-User Facility Related Tables

The following Dynamic System Tables (located in the SYS area, base 1000) provide information about the MUF environment and activity. They provide information in real time about current conditions or a summary of information/events that has occurred from the start of this MUF to the current time. This information can be acquired and used immediately, or could be saved in other user-defined tables. The information could be requested at intervals, and the differences used or saved in other user-defined tables.

System Statistical Information

The following tables exist to provide data equivalent to the PXX system statistics.

- MUF_AREA_STATS (MFA) (see page 64)
- MUF_SYSTEMS_STATS (MSS) (see page 129)
- MUF_TABLE_STATS (MFT) (see page 138)

Startup Option Information

The following tables exist to provide data about the MUF startup options (or the current values for this type of information).

- MUF_ACCESS (MFZ) (see page 54)
- MUF_ACCOUNTING (MFY) (see page 56)
- MUF_CDC (MCD) (see page 70)
- MUF_CDC_BASE_TABLE (MCT) (see page 74)
- MUF_CDCL (MCL) (see page 75)
- MUF_CONFIG (MCF) (see page 70)
- MUF_COVEREDVIRTUAL (MFC)
- MUF_OPTIONS (MFO) (see page 110)
- MUF_PRODUCTS (MFP) (see page 117)
- MUF_RC_DUMP_OPT (MDO) (see page 119)
- MUF-SYSOUT (MFG) (see page 129)
- MUF_XCF (MFX) (see page 141)

Other Information

The following tables provide various other kinds of information.

- MUF_ACCESS_AREA (MFN) (see page 55)
- MUF_ACTIVE_TASKS (MFQ)
- MUF_BUF_USE (MBU) (see page 66)
- MUF_CBS (MCB) (see page 67)
- MUF_CBS_OLD_SETS (MCO) (see page 69)
- MUF-DATA_SHARING (MFD) (see page 84)
- MUF_IDENTITY (MFI) (see page 86)
- MUF_INTERNAL_STATS (MFV) (see page 89)
- MUF_LOCKS_VALUE (MFK) (see page 92)
- MUF_LOGGING (MFL)
- MUF_MEM_DETAIL (MMD) (see page 101)
- MUF_MEM_SUMMARY (MMS) (see page 102)
- MUF_ML_DUMPS_PREV (MFF) (see page 103)
- MUF_ML_DUMPS_PXX (MFE) (see page 105)
- MUF_OPEN_BASES (MFB) (see page 107)
- MUF_RATES (MFR) (see page 117)
- MUF_RETURN_CODES (MFU) (see page 121)
- MUF_SMP_STATS (MFW) (see page 122)
- MUF_SMP_TASK (MFM) (see page 127)
- MUF_SRB_ZIIP (MZI) (see page 127)
- MUF_TCB_OR_SRB (MTC) (see page 139)

MUF_ACCESS (MFZ)

This table contains access information. One row exists in this table for every database known to the Directory (CXX). This table reflects the Multi-User startup option ACCESS setting, as modified by any utility or console ACCESS changes. The ACCESS can be set for databases which are not currently defined to the Directory, but they *cannot,* however, be queried with this table.

Column Name	SQL Data Type	Nullable	Description
DBID	SMALLINT	No	Contains the database ID.

Column Name	SQL Data Type	Nullable	Description
MAINTENANCE	CHAR(1)	No	Indicates whether maintenance can currently be done to this database.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
OPEN_STATUS	CHAR(6)	No	Contains the current open status of this database. Contains CLOSED if the MUF does not have this database open. Contains WRITE if this MUF has the database open with update authority. Contains READ if this MUF has the database open without update authority.
OPTIMIZE_OPEN	CHAR(1)	No	Contains Y or N to indicate the setting of the OPTIMIZE Multi-User startup option for this database ID.
TYPE_ACCESS	CHAR(5)	No	Contains the type of access allowed for User Requirements Table (URT) opens. OFF indicates NO opens are allowed. UTLTY indicates opens are only allowed by DBUTLTY functions. READ indicates opens are allowed for URTs with UPDATE=NO. WRITE indicates that all opens are allowed.

MUF_ACCESS_AREA (MFN)

This table contains information about the area level access function. One row exists in this table for every area that has one of the following conditions:

- Has been specified in the DBUTLTY function ACCESS specifying both the DBID= and AREA= keywords.
- Has been opened by an execution of DBUTLTY function RECOVERY OPTION=BACKWARD.
- Has been opened by an execution of a DBUTLTY function that included the keyword MULTUSE=YES (BACKUP, EXTEND, EXTRACT, INIT, LOAD, REORG, RETIX).

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Contains the area name.
DBID	SMALLINT	No	Contains the database ID.

Column Name	SQL Data Type	Nullable	Description
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
TYPE_ACCESS	CHAR(5)	No	Type of access in effect, OFF or READ (WRITE is implied for missing rows).
TYPE_ACCESS_USER	CHAR(5)	No	Type of access set by user using DBUTLTY ACCESS with DBID and area name. Possible values are OFF, READ, UTLTY, or WRITE.
TYPE_ACCESS_UTLTY	CHAR(5)	No	Type of access set by a DBUTLTY function (BACKUP, EXTEND, EXTRACT, INIT, LOAD, REORG, RETIX) using the keyword MULTUSE=YES (excludes the ACCESS function). Possible values are READ, UTLTY, or WRITE.
WHICH_UTLTY	CHAR(8)	No	Contains blanks if the previously shown TYPE_ACCESS_UTLTY column contains WRITE, else contains BACKUP, EXTEND, EXTRACT, INIT, LOAD, RECOVERY, REORGB, REORGL, or RETIX.

MUF_ACCOUNTING (MFY)

This table is for the Accounting Facility. It provides the startup information and the current information about each Accounting table. One row exists for each Accounting table defined as a startup option.

Before Version 12.0, you could perform certain kinds of maintenance (inserts and deletes) on this table, but beginning with Version 12.0, that functionality has been moved to the SQL_CONSOLE (SQX) (see page 143) Dynamic System Table.

Column Name	SQL Data Type	Nullable	Description
BUFFER_SIZE	INTEGER	No	The size of the buffer as provided in the startup option.

Column Name	SQL Data Type	Nullable	Description
CONFLICTS	INTEGER	No	Number of entries in the Accounting table that had exclusive control held by a user application when a spill was initiated.
DBID	SMALLINT	No	The database ID containing this Accounting table.
ENTRIES	INTEGER	No	Number of entries in the in-memory Accounting buffer.
LOCATES	INTEGER	No	Number of entries retrieved into the Accounting buffers including existing entries in the Accounting table with reclaimed internal space.
LOCATES_STARTED	INTEGER	No	Number of times that a group of locates were performed for the Accounting table.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
PHYSICAL_READWRITE	INTEGER	No	Physical I/O (EXCPs) issued on behalf of requests for this Accounting table.
RECLAIMED	INTEGER	No	Number of in-memory table entries reused after a spill, that is, the number of entries for which a LOCATE was not required.
SKIPPED	INTEGER	No	Number of entries that were not immediately processed due to exclusive control conflicts.
SPILLING	CHAR(1)	No	Indicates whether this Accounting table is currently being spilled.
SPILLS_INTERNAL	INTEGER	No	Number of spills initiated to decrease the number of entries unavailable.
SPILLS_THRESHOLD	INTEGER	No	Number of times that the threshold parameter value was exceeded and a spill was done.
TABLE_NAME	CHAR(3)	No	Name of the Accounting table.

Column Name	SQL Data Type	Nullable	Description
TABLE_STATUS	CHAR(1)	No	Current Accounting table status. Contains O for open or C for closed.
THRESHOLD	INTEGER	No	Current value (nnnnn) of the threshold counter. Defines how long changes to an Accounting table can be buffered in the Accounting buffer before being spilled to the normal buffers.
TOTAL_ADDS	INTEGER	No	Number of adds which were done to this Accounting table.
TOTAL_REQUESTS	DECIMAL(13,0)	No	Number of times this Accounting table was used. This value is determined by the conditional expression, if there is one.
TOTAL_UPDATES	INTEGER	No	Number of times that records in the Accounting table were updated.
UNAVAILABLE	INTEGER	No	Number of user application requests that had to wait due to insufficient memory allocated by the buffer parameter.

MUF_ACTIVE_TASKS (MFQ)

This table provides status information about active tasks. One row exists for every task area which is currently attached. This table has characteristics in common with the console status command and the DBUTLTY status function. You can query task areas that are currently attached. This includes query task areas being attached with normal user jobs, being attached by CA Datacom STAR, or being attached by internal CA Datacom/DB system tasks such as the ACCT (Accounting Facility) or index queue functions.

Column Name	SQL Data Type	Nullable	Description
BUFFER_REFERENCES	INTEGER	No	The number of index and data buffers referenced by this task, starting when the request is accepted by the MUF.

Column Name	SQL Data Type	Nullable	Description
CPU_TIME	CHAR(6)	Yes	This column contains the total CPU time for this request if you have chosen to keep track of CPU time by specifying YES for the *cpu-time* parameter on the ACCTPRM Multi-User startup option. The format is *mmm:ss*, where *mmm* is minutes and *ss* is seconds. Otherwise this column is null. For more information, see the performance note for *cpu-time* in the information about the ACCTPRM Multi-User startup option in the *CA Datacom/DB Database and System Administration Guide*. Because CPU time is only computed periodically, tasks that are actively running might not include all of their CPU time.
CURRENT_STATUS	CHAR(15)	No	The current status of a task area is the same as is documented for the console status command. The list is long. For more information, see the COMM STATUS information in the *CA Datacom/DB DBUTLTY Reference Guide* for details.
DB_COMMAND	CHAR(5)	No	The five-character record-at-a-time or set-at-a-time command being executed or last executed. This field can contain certain internal commands that are only used by CA Support.
DBID	SMALLINT	No	Contains the database ID.
DURATION	CHAR(6)	No	The duration of the current request in the format *mmm:ss* where *mmm* is minutes and *ss* is seconds. A duration of 000:00 is provided if the request is complete or not yet accepted by the MUF. The minutes wrap after 999.

Column Name	SQL Data Type	Nullable	Description
EOJ_OK	CHAR(1)	No	The EOJ_OK option allows MUF to EOJ without the application closing all URTs. The column can be either Y, S, or N. - Y and N reflect the URT DBURSTR macro parameter EOJ_OK. - S indicates MUF EOJ_OK S=YES has been set for this CA Datacom Server task. For more information, see the *CA Datacom/DB Database and System Administration Guide*.
JOB_NAME	CHAR(8)	No	The name of the job currently in the task area. Names starting with an asterisk (*) are internal to CA Datacom/DB.
LOCK_VALUE	CHAR(64)	No	Provides a character string of the row lock or value lock upon which is being waited, similar in format to the console STATUS function. The first six characters of the lock value are provided in the status after WAIT E/C.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
MUFPLEX_OWNER	CHAR(2)	No	The MUFplex number at which the task owning a row or value lock resides, if this is a remote wait for exclusive control.
OPTIONAL_ID	CHAR(16)	No	For CICS transactions, contains the transaction ID, terminal ID, operator ID, and sequence number. For CA Datacom Server, contains the logical terminal name. For DBUTLTY, contains literal "DBUTLTY" followed by the DBUTLTY utility function. Otherwise, this field is blank.

Column Name	SQL Data Type	Nullable	Description
OWNER_TASK	SMALL INTEGER	No	If the status is WAIT E/C, the number of the task which owns the lock is reported. Some locks have an owner which is the MUF itself and not a specific task, in which case the OWNER_TASK is reported as a zero (0). This does not occur for data row locks but does for some of the value locks.
PHYSICAL_EXCPS	INTEGER	No	The number of physical I/O events done by this task, starting when the request is accepted by the MUF.
REQUEST_SEQ_NO	INTEGER	No	The internal request sequence number. It starts at one when the MUF starts and is incremented for each request. It wraps as necessary. It is typically used to see that the system is performing new work and to determine if a long-running user application is looping. The entire number (if less than 6 digits or else the low order 6 digits) can be used in the REQABORT utility to cancel a request if desired.
RUN_TIME	CHAR(6)	No	The amount of time this request has been running, minus the amount of time it has been in a CA Datacom controlled wait. The format is *mmm:ss*, where *mmm* is minutes and *ss* is seconds.
RUN_UNIT	INTEGER	No	The internal number assigned by CA Datacom/DB to each CA Datacom job. This number is reported in various possible messages relating to the job.
TABLE_NAME	CHAR(3)	No	Name of the table.

Column Name	SQL Data Type	Nullable	Description
TASK_NUMBER	SMALL INTEGER	No	The number of the task area being used for this request, also known as the RWTSA number. The low numbers are allocated for user applications based upon the TASKS Multi-User startup option, followed by internal CA Datacom/DB system task areas.
TRN_SEQ_NO	CHAR(8)	No	Provides the transaction sequence number (TSN) if the current task is in a transaction, or provides blanks if the current task is not in a transaction.
TSN_DURATION	CHAR(6)	No	Provides the duration of the transaction if the current task is in a transaction, or provides blanks if the current task is not in a transaction. The format of the duration is similar to the current request in that it is provided as mmm:ss, except that a time of zero is set to all blanks and a time greater than 999 minutes is set to +++:++.

Column Name	SQL Data Type	Nullable	Description
USER_PATH	Char (5)	No	USER_PATH specifies the path from the user application to the MUF and can be as follows: **LOCAL** LOCAL is the PATH if the user and the MUF are on the same system and none of the following options were selected. **XCF** XCF is the PATH if the z/OS XCF facility is being used for communication. **CCI** CCI is the PATH if the CA CCI facility is being used for communication. **IUCV** IUCV is the PATH if the z/VSE IUCV facility is being used for communication. ***MUF*** Request was generated by MUF itself. This includes system requests, and console or console like requests which require an RWTSA, and restart rollback requests. **UNKN** UNKN is the PATH if the path cannot be determined (this PATH is not expected to occur).
USER_JOBID	Char (8)	No	USER_JOBID is the z/OS JOBID identification of the user application. This may be blanks.

Column Name	SQL Data Type	Nullable	Description
USER_RQ_DATA	CHAR(32)	No	Associated with each active request is 32 bytes of user request data. For record-at-a-time and set-at-a-time commands this data is passed by the user application program. When running a transparency, the transparency provides the information. Requests that are a part of SQL contain a variety of values. The format and content from all sources is totally unpredictable.
USER_SYSTEM_NAME	Char (8)	No	USER_SYSTEM_NAME is the system name, where the user application called the interface. This may be blanks.
WAIT_DURATION	CHAR(6)	Yes	If this task is currently waiting, this column contains the total amount of time since this wait began. The format is *mmm:ss*, where *mmm* is minutes and *ss* is seconds. If the task is not in a wait, this column is null.
WAIT_TIME	Char(6)	No	Specifies the total wait time that this request was not selected by the MUF for processing, excluding a current wait, if any, whose time is in WAIT DURATION.

MUF_AREA_STATS (MFA)

In this table, one row exists for every area that has been opened by the MUF including the control areas Directory (CXX), Log Area (LXX), and Force Area (FXX), and for user index and data areas.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Area name. Includes the Directory (CXX), Log Area (LXX), Force Area (FXX), Index Area (IXX), Statistics and Diagnostics Area (PXX) and user data areas.

Column Name	SQL Data Type	Nullable	Description
BEGIN_TIME	TIMESTAMP	No	The date and time of the MUF enable.
CURRENT_DATETIME	TIMESTAMP	No	The current date and time.
DBID	SMALLINT	No	The database ID for a data area or the index to a data area. It is zero for the Directory, Log Area, Force Area, and the Statistics and Diagnostics Area (PXX).
LOGICAL_READS	DECIMAL(11,0)	No	Logical read of any block in this area. A Directory, log, index, or data block having been physically read from DASD may stay in memory and be used again and again. Each of these uses is a logical read.
LOGICAL_WRITES	DECIMAL(11,0)	No	Logical write of any block in this area. An add, update, or delete of a record generates one logical write to the data area.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
PHYSICAL_READS	DECIMAL(11,0)	No	Number of I/O requests made against this area to read one or more blocks of DASD information.
PHYSICAL_WRITES	DECIMAL(11,0)	No	Number of I/O requests made against this area to write one or more blocks of DASD information.

MUF_BUF_CONTENT (MBC)

In the MUF_BUF_CONTENT Dynamic System Table, one row exists for each DBID or area and DBID which has been defined to use an alternate buffer pool. The table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Datacom name of the area if the buffer pool content was set at the area level. If the buffer pool content was set at the database level, the value is blanks.

Column Name	SQL Data Type	Nullable	Description
DBID	SMALLINT	No	The database Identifier.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
POOL_NAME	CHAR(6)	No	The name of the alternate buffer pool this entity is using.

MUF_BUF_USE (MBU)

In the MUF_BUF_USE Dynamic System Table, one row exists for each buffer pool defined to the MUF. This includes the four standard buffer pools. The table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
BUFFER_SIZE	INTEGER	No	Contains the buffer size for this pool.
BUFFER_USED_1	DECIMAL(13,0)	No	Number of times a buffer in this pool was referenced for the first-time, after the buffer was read form DASD or copied from MRDF.
BUFFER_USED_2	DECIMAL(11,0)	No	Number of times a buffer in this pool was referenced for a second time, after the buffer was read form DASD or copied from MRDF.
BUFFER_USED_3	DECIMAL(11,0)	No	Number of times a buffer in this pool was referenced for a third time, after the buffer was read form DASD or copied from MRDF.
BUFFER_USED_4	DECIMAL(11,0)	No	Number of times a buffer in this pool was referenced for a fourth time, after the buffer was read form DASD or copied from MRDF.

Column Name	SQL Data Type	Nullable	Description
BUFFER_USED_5	DECIMAL(13,0)	No	Number of times a buffer in this pool was referenced for a fifth time (and a sixth time, and a seventh time, and so on), after the buffer was read form DASD or copied from MRDF.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
NUMBER_OF_BUFFERS	INTEGER	No	Count of buffers currently specified for this pool. For the standard buffer pools, the count includes any FLEXPOOL buffers currently specified.
POOL_NAME	CHAR(6)	No	The name of the standard pool. The standard pools are DATA, DATA_2, DXX, and IXX.

MUF_CBS (MCB)

The MUF_CBS Dynamic System Table contains CBS information and statistics. The MCB table contains one row and the following columns:

Column Name	SQL Data Type	Nullable	Description
BUF_CUR_PERC_FULL	SMALLINT	No	This is the percentage of the Compound Boolean Selection (CBS) buffer size specification currently being used for open set definitions.
BUF_MAX_PERC_FULL	SMALLINT	No	This is the high water percentage of the Compound Boolean Selection (CBS) buffer size specification that was used for open set definitions. Once the CBS buffer percent used reaches 100, it causes the least recently used sets to be spilled to the CBS index.
CUR_USED_SET_MEM	INTEGER	No	The amount of memory being used for open set definitions.

Column Name	SQL Data Type	Nullable	Description
DURATION_AT_SPILL	CHAR(6)	Yes	The duration or length of time the most recently spilled set existed before it was spilled, in the format *mmm:ss*. If no sets have been spilled, this column is null.
MAX_USED_SET_MEM	INTEGER	No	The maximum amount of memory used for open set definitions. The maximum amount of memory used is limited by the Compound Boolean Selection (CBS) buffer specification. If this limit is reached, open sets must be spilled to the CBS index.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
SETS_DEL_MAX_AGE	DECIMAL(11,0)	No	The number of open sets in the Compound Boolean Selection (CBS) buffer that were deleted because of the CBS max age specification. If this number is not zero, it can indicate that applications are not releasing sets when they are finished with them.
SETS_FETCHED	DECIMAL(11,0)	No	Count of the number of times a spilled set was referenced, to retrieve more rows. The ratio between sets spilled and sets fetched can be used to understand if sets that are truly in use are being spilled to the CBS index, or if it is an absence of the SELPR command in user programs that is causing sets which are no longer in use to be paged out.
SETS_PROCESSED	DECIMAL(11,0)	No	Specifies the number of sets processed by Compound Boolean Selection (CBS).
SETS_SPILLED	DECIMAL(11,0)	No	Count of sets which had to be paged to the Compound Boolean Selection (CBS) index. This is an interaction between the specified CBS buffer size and the number and size of open CBS sets. If SETS_SPILLED in not zero, you might want to increase the CBS buffer size or examine your applications for missing SELPR commands.
SETS_TEMP_INDEX	DECIMAL(11,0)	No	The number of sets requiring a temporary index.

Column Name	SQL Data Type	Nullable	Description
SPILLED_JOB_NAME	CHAR(8)	Yes	This is the job name that owned the most recently paged-out set, because the Compound Boolean Selection (CBS) buffer was full. If no sets have been spilled, this column is null.
SPILLED_UIB	CHAR(32)	Yes	This is the 32 UIB associated with the most recently paged-out set, because the Compound Boolean Selection (CBS) buffer was full. If no sets have been spilled, this column is null.
TEMP_INDEX_ENTRIES	DECIMAL(11,0)	No	The total number of temporary index entries built to support set selection criteria.

MUF_CBS_OLD_SETS (MCO)

In the MUF_CBS_OLD_SETS Dynamic System Table, one row exists for each of the ten oldest open Compound Boolean Selection (CBS) sets that reside in the CBS buffer and have therefore not been spilled. There can be fewer than ten rows in this table if there are less than ten sets currently in the CBS buffer. The same set can appear more than once in this table if the set is used while your query is running and after its use is in the ten least recently used sets.

The MUF_CBS_OLD_SETS Dynamic System Table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
JOB_NAME	CHAR(8)	No	The name of the job that opened this set.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
RUN_UNIT	INTEGER	No	The internal run unit assigned to the job that opened this set.
SEQUENCE_NUMBER	SMALLINT	No	A sequence number of from one to ten (1—10).
SET_AGE	CHAR(6)	No	The amount of time that has passed since this set was last used, in the format *mmm:ss*.

Column Name	SQL Data Type	Nullable	Description
SET_NUMBER	INTEGER	No	This is a sequence number assigned when a set is created. For example, a number of 110 represents the one hundred and tenth set created during the life of this MUF.
UIB	CHAR(32)	No	The User Information Block (UIB) associated with this set.

MUF_CDC (MCD)

This Dynamic System Table contains information about Change Data Capture (CDC) and the current MUF. The MUF_CDC table contains one row if the current MUF had CDC Multi-User startup option defining this MUF as a source MUF.

The MUF_CDC contains the following columns:

Column Name	SQL Data Type	Nullable	Description
CDC_BUFFER_SIZE	INTEGER	No	The CDC work buffer size, taken from the *size* parameter of the CDC Multi-User startup option.
CDC_DBID	SMALLINT	No	The database identifier (database ID) for the CDC database, taken from the CDCM_DBID Multi-User startup option (or default).
CDCM_ERROR_REASON	CHAR(16)	Yes	Is set to null if the CDCM task last request returned a blank return code. Otherwise, the field contains information about the request, the command, the table name if not an open or close, the external return code, and the internal return code.
CDCM_POLL_SECONDS	SMALLINT	No	The interval in seconds, as specified with the CDCM_POLL Multi-User startup option, between subsequent checkings (by the subtask) of the target MUF, to determine the status of the CDCL and CDCU programs.

Column Name	SQL Data Type	Nullable	Description
CDCM_POLL_STATUS	CHAR(8)	No	Contains a value of either WAITENBL, CLOSED, WAITPOLL, WAIT-RQ-, or RUNNING, described as follows: **WAITENBL** WAITENBL means waiting on the source MUF to be enabled, where CDCM is not running its target MUF. **CLOSED** CLOSED means CDCM is currently defined as closed and is waiting for a CDCM_OPEN request. **WAITPOLL** WAITPOLL means CDCM is connected to the source MUF and is waiting for the next poll interval to verify the status again. **WAIT-RQ-** WAIT-RQ- means CDCM is connected or connecting to the source MUF and waiting on a request. **RUNNING** RUNNING means CDCM is actively running or waiting on a CP to run.
CDCM_SIDNAME	CHAR(8)	No	As specified with the CDCM_SIDNAME Multi-User startup option, identifies the target MUF to the CDCM subtask.

Column Name	SQL Data Type	Nullable	Description
CDCM_SUBTSK_STATUS	CHAR(7)	No	Can be OPEN, CLOSED, CLS>OPN, OPN>CLS, OPN-ING, or CLS-ING as follows: **OPEN** OPEN indicates that the CDCM subtask is open. **CLOSED** CLOSED indicates that the CDCM subtask is closed. **CLS>OPN** CLS>OPN indicates that the CDCM subtask is closed, but you have requested it to open. **OPN>CLS** OPN>CLS indicates that the CDCM subtask is open, but you have requested it to close. **OPN-ING** OPN-ING indicates that the CDCM subtask is in the process of opening. **CLS-ING** CLS-ING indicates that the CDCM subtask is in the process of closing.
CDCM_WARN	SMALLINT	No	Identifies how many seconds, as specified with the CDCM_WARN Multi-User startup option, that the CDCL or CDCU program can fall behind the LXX before a warning message is issued. The CDCL task is measured from the last log record it saw compared to the current last log record on DASD. The CDCU task is measured from the oldest TSN record in the CDC database to the current last log record on DASD.
CONTROL_ID	CHAR(1)	No	Identifies to the CDC monitor, as specified with the CDCM_ID Multi-User startup option, which level of support is being used by the CDCL task. This information is used in the CNTKR commands to track the correct records. The letter A is the valid control ID.

Column Name	SQL Data Type	Nullable	Description
CURRENT_DATETIME	TIMESTAMP	No	Current date and timestamp from the MUF.
LAST_CDC_RECORD	CHAR(6)	Yes	The time, given in minutes *mm* and seconds *ss,* that CDCL is behind in processing LXX records. If the time is greater than 999 minutes, the display shows +++:++. If the CDCM subtask has not yet successfully opened the CDCL, the time is technically unknown and is therefore indicated by UNKN.
LAST_LOG_RECORD	CHAR(6)	No	The time, given in minutes *mm* and seconds *ss,* that CDCU is behind in processing TSN records. If the time is greater than 999 minutes, the display shows +++:++. If the CDCU user task has not yet started, the time is technically unknown and is therefore indicated by UNKN.
MAINT_POLL_COUNT	INTEGER	No	At the time of the last CDCM poll, this is the count of MNT table records based upon the rows in the table returned from either a CNTTB or CNTKY command.
MAINT_REQUESTS	DECIMAL(11,0)	No	Count of maintenance requests defined as subject to CDC during the physical life of this MUF.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
MUFPLEX_NAME	CHAR(8)	No	If running full Data Sharing as a MUFplex, the name of the group.
TSN_POLL_COUNT	INTEGER	No	At the time of the last CDCM poll, this was the count of TSN table records based upon the rows in the table returned from either a CNTTB or CNTKY command.

Column Name	SQL Data Type	Nullable	Description
USE_CNTTB	CHAR(1)	No	Provides information regarding the performance option, as specified with the CDCM_CNTTB Multi-User startup option, that allows the CDCM subtask to use CNTTB commands to determine the number of transaction sequence number (TSN) records and maintenance (MNT) LXX records in the CDC database. This count of records is checked at the polling intervals set with the CDCM_POLL Multi-User startup option. In a MUFplex with two or more MUFs enabled, a CNTTB command provides only the number known to the local MUF, not to the MUFplex as a whole.

MUF_CDC_BASE_TABLE (MCT)

This Dynamic System Table reflects the current settings of the CDC_BASE and CDC_TABLE Multi-User startup options as changed by console-like CDC_BASE and CDC_TABLE commands. For CDC_TABLE entries, one row exists for each database with table name that is currently set with the YES or NO values. No row exists for any database with table name that is currently set with the IGN value. The presence of a database with table name row is considered an exception to the normally expected processing done using CDC_BASE. One row also exists for each database (table name blank) that is currently set with the YES value using CDC_BASE. Table entries have priority over a database entry if both have the same DBID.

Note: Tables can exist that are not represented as rows in this table. Do not be confused by the apparent conflict of having a database set to NO where every table in the database is set to YES.

The ordering for this table includes the MUF name, the DBID, and the table name. Database entries have the table name set to blanks and are lower in order than all other table names in that database.

The MUF_CDC_BASE_TABLE contains the following columns:

Column Name	SQL Data Type	Nullable	Description
DBID	SMALLINT	No	The database identifier (database ID).

Column Name	SQL Data Type	Nullable	Description
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
PROPAGATE	CHAR(1)	No	Contains a letter Y if this entry is subject to CDC. Contains a letter N if this entry is not subject to CDC. **Note:** CDC_BASE databases that are currently set to NO (the default) do not have a row in this table, and therefore every present CDC_BASE entry with its blank table name always has the PROPAGATE value set to the letter Y.
TABLE_NAME	CHAR(3)	No	For CDC_TABLE entries, contains the table name that has been set to participate or to not participate. For CDC_BASE entries, contains blanks.

MUF_CDCL (MCL)

One row exists for this table for each Change Data Capture Listener (CDCL) task defined to the current MUF.

The MUF_CDCL contains the following columns:

Column Name	SQL Data Type	Nullable	Description
CDCL_DBID	SMALLINT	No	The database identifier specified with the CDCL_DBID Multi-User startup option, that is, the DBID containing the CDC table(s).
CDCL_ERROR_REASON	CHAR(32)	Yes	Either contains blanks if no error has occurred since this listener was last opened successfully, or contains information about the last error. The many values that are possible are described in the explanatory description of message DB03153E.
CDCL_NAME	CHAR(8)	No	Identifies the name, as specified in the CDCL Multi-User startup option, of the source MUF owning the LXX.

Column Name	SQL Data Type	Nullable	Description
CDCL_POLL_SECONDS	SMALLINT	No	Indicates how long, as specified by the CDCL_POLL Multi-User startup option, the CDCL task is to wait when there are no LXX records to process. Only used when the CDCL task operates in remote mode.
CONTROL_ID	CHAR(1)	No	Identifies to the CDC monitor, as specified with the CDCM_ID Multi-User startup option, which level of support is being used by the CDCL task. This information is used in the CNTKR commands to track the correct records. The letter A is the valid control ID.
CURRENT_DATETIME	TIMESTAMP	No	Current date and timestamp from the MUF.
CURRENT_STATUS	CHAR(7)	No	Is one of the following: **OPN>CLS** OPN>CLS indicates it is open but has been requested to close. **CLS>OPN** CLS>OPN indicates it is closed but has been requested to open. **OPEN** OPEN indicates it is currently open. **CLOSED** CLOSED indicates it is currently closed.
DDNAME	CHAR(8)	No	Name of the DD statement used to read the LXX data set of the source MUF.
DSNAME	CHAR(44)	No	Name of the data set that is the LXX of the source MUF.
GAP_WAIT	CHAR(1)	No	The letter Y indicates that this CDCL task is in a GAP situation and waiting to be notified that it has been resolved and needs to try again. The letter N indicates that this CDCL task is not in a GAP situation.
LAST_CDC_RECORD	TIMESTAMP	Yes	Timestamp in SQL format for the last log record that was subject to CDC.
LAST_LOG_RECORD	TIMESTAMP	Yes	Timestamp in SQL format for the last log record processed, CDC or not.

Column Name	SQL Data Type	Nullable	Description
MAINT_PROCESSED	DECIMAL(11,0)	No	The number of maintenance log records processed that were subject to CDC.
MODE	CHAR(10)	No	Can be LOCAL, REMOTE(DS), REMOTE(MP), REMOTE(WB), or REMOTE(XD) as follows: **LOCAL** LOCAL refers to local processing. **REMOTE(DS)** REMOTE(DS) indicates a remote status because the data set name does not match the local LXX. **REMOTE(WB)** REMOTE(WB) indicates a remote status because the MUF CDC startup option did not provide a work buffer. **REMOTE(MP)** REMOTE(MP) indicates that it is a remote status because the MUF currently has two or more MUFplex MUFs enabled. **REMOTE(XD)** REMOTE(XD) indicates that a remote status has been forced with a debugging option. **Note:** The debugging option just mentioned is intended only for use by those who have received information about it, that is, by CA.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
PHYSICAL_READS	DECIMAL(11,0)	No	The number of I/O events directed to the LXX used by the source MUF.
WRAP_COUNT	INTEGER	No	The number of times the CDCL task has read the LXX and found it newer than expected. This indicates that the LXX has completely circled around (wrapped) while the CDCL was running or stopped.

MUF_CONFIG (MCF)

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

The only rows that exist in this table are those that have been provided by the user. If a user does not provide an option such as FORCE_NO_64BIT YES, it is not in the table. The user can deduce that missing items are the default values.

Because users can set the same item many times in a MUF execution, the last value for that item is the only one saved, allowing this table to have a reasonable number of records.

The key of the table is the MUF_NAME and the CONFIG_COMMAND.

Some console commands are excluded, including ?MEM, ZAP and DISPLAY.

Because the key fields are required to be unique, no command exists in this table that violates this rule.

The following Multi-User startup options and console-like commands can appear in this table:

- ARM
- CPU_TSN_READ_DELAY_SEC
- DIAGOPTION
- DUMP_OUTPUT
- DYNAMIC_EXTEND_MSGS
- FORCE_ABEND_FULL_SNAPER
- FORCE_NO_64BIT
- HISTORY
- HISTORY_END_HOUR
- LOG_CONTROL_STEPDOWN
- LOG_RECORD_ENCRYPT
- LOG_ROLLBACK_MINUTES
- LOG_ROLLBACK_RECORDS

- LUW_STATEMENT_LIMIT
- RXX_VARIABLE_BLK
- SDI_ABEND_1
- SNAPCLS
- SQL_COND_INFO_AREAS
- SQL_DATACOM_LOOPLIMIT
- SQL_KATAKANA_CODE_PAGE
- SQL_LRU_STATEMENT_CACHE
- SQL_OPTIMIZATION_LEVEL
- SQL_PLAN_WAIT
- SQL_SQUARE_BRACKETS
- SQLOPTION_TIMEOUT
- VTOC_DATE_DAILY
- X_BFMTRACE_BFR_UNLK
- X_COVERED_MAX_SIZE
- X_CXX_ALLOW_SHARING
- X_DUMP_NUMBER_BLKS
- X_IO_HIGH_THRESHOLD
- X_IO_LOW_THRESHOLD
- X_IO_MSG_THRESHOLD
- X_IO_24BIT_SIZE
- X_LOG_AHEAD_BLKS
- X_LOG_CONTROL_BLKS
- X_LOG_DELAY_BATCH
- X_LOG_DELAY_BLK
- X_LOG_DELAY_SEC
- X_LOG_DELAY_TIME
- X_LOG_FULL_BLK
- X_LOG_HASH_LR
- X_LOG_INFO1_MSG
- X_LOG_INFO2_MSG
- X_LOG_INFO3_MSG

- X_LOG_MINIMUM_BLK
- X_LOG_SPILL_BLKS
- X_LOG_STABLE_BLKS
- X_LOG_WRITE_OLD_BLKS
- X_ML_DUMP_RQP
- X_MSB_MSGS
- X_OPEN_CLOSE_MSGS
- X_PEND_WRITE_BLKS
- X_SNAPER_OPTION_ALL
- X_TRACE_BFM_SIZE
- X_TRACE_DSM
- X_TRACE_RQ_SIZE
- X_TRACE_XCF_OPTION
- X_TRACE_XCF_SIZE

The MUF_CONFIG (MCF) table has three columns with the following characteristics:

Column Name	SQL Data Type	Nullable	Description
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
CONFIG_COMMAND	CHAR(25)	No	The configuration command entered by the user, up to field maximum. The command is left justified with blanks padded on the right. One or more blanks between the command and operands are removed.
CONFIG_PARMS	CHAR(65)	No	The operand or operands entered by the user, up to field maximum. The operand is exactly as the user specified it, to the 65-byte limit. One or more blanks between the command and operands are removed.

MUF_COVEREDVIRTUAL (MFC)

This table is for MRDF (Memory Resident Data Facility). This table contains combined information from the definitions of the desired covered and virtual usage with the current covered and virtual status.

Four types of rows can exist to provide all the information.

- The first is a database level definition. This type of definition serves as a default for all areas in the database which have no specific matching area definition.
- The second type of row is for specific area definitions for which the area has not yet been opened.
- The third type is a specific covered area which is open or has been opened.
- The last is a specific virtual area which is open or has been opened.

The following table explains what data is present for the various types of conditions.

Column Name	SQL Data Type	Nullable	Description
ACTIVE_READS	INTEGER	Yes	This field is null for a database level definition or an area level definition of an unopened area. It is also null for a virtual area. Provides the number of blocks read from disk to support an active covered area which is open or has been open. Also provides the number of blocks read from disk for covered using FIRST.
AREA_NAME	CHAR(3)	No	Contains blanks for a database level definition. Otherwise, it contains a specific area name from either a specific definition or the open of an area in a database with a database level definition but no specific definition for this area.
BEGIN_BLOCK	SMALLINT	Yes	Contains the percent of area blocks used when covered with FIRST option. Contains null if virtual or covered with the ACTIVE option.

Column Name	SQL Data Type	Nullable	Description
BLOCKS_CURRENT	INTEGER	Yes	Contains null if a database level definition or an unopened specific area definition. Otherwise, it contains the number of blocks currently available for use for this specific area.
BLOCKS_MAXIMUM	INTEGER	Yes	Contains null if a database level definition or an unopened specific area definition. Otherwise, it contains the number of blocks allocated for use for this specific area.
DATASPACE	CHAR(1)	Yes	If this field is null, it indicates that this row is a database level definition or an unopened specific area definition. If this field contains a Y, it indicates that this area is in a data space. If this field contains an N, it indicates that normal address space memory is in use. If this field contains an X, it indicates that the space used is 64-bit. A VIRTUAL area can have its initial allocation in 64-bit or a data space. Later extends only occur in a data space. The value reported here is the initial allocation.
DBID	SMALLINT	No	Contains the database ID of this database.
FIRST_ACTIVE	CHAR(1)	Yes	For database level definitions or specific area definitions to unopened areas, if either did NOT provide the FIRST or ACTIVE option specified, this field can be null. But the field can contain an F to indicate FIRST is to be used, or an A to indicate ACTIVE usage, even if you did not specify an F or an A. See the sections on COVERED FIRST and COVERED ACTIVE in the chapter on the Memory Resident Data Facility in the *CA Datacom/DB Database and System Administration Guide* for more information.

Column Name	SQL Data Type	Nullable	Description
HIGH_VIRTUAL_BLOCK	INTEGER	Yes	This field contains null except for areas opened in a VIRTUAL database. This field contains the number of the last block in the virtual data set.
MEMORY_SIZE	DECIMAL(11,0)	No	The size for a database definition or an unopened specific area definition is what was provided in the covered or virtual definition. For these definitions with covered, a size value of 1 through 100 represent a percentage of the area to cover. A value greater than 100 represents the number of bytes to cover. Definition sizes ending in K,M,G are converted to bytes for this size field. For an area which has been opened, the size is always the number of bytes and reflects the actual memory allocated. This size for an open area has been rounded up to one block (covered) or one track (virtual), if necessary. Otherwise, it was rounded down to an even number of blocks (covered) or an even number of tracks (virtual).
MRDF_READS	INTEGER	Yes	This field is null in definitions other than areas which have been opened. It contains the number of blocks that were read from the MRDF space instead of from DASD.
MUF_NAME	CHAR(8)	No	Name of the Multi-User Facility specified in the MUF Multi-User startup option or the name of the MUF job name.
NONFIRST_READS	INTEGER	Yes	This field is null in definitions other than covered areas which have been opened and that are being processed with the FIRST option. It contains the number of blocks read from DASD which are higher than the first covered blocks.

Column Name	SQL Data Type	Nullable	Description
TOTAL_READS	INTEGER	Yes	Null if this row is not an opened area. Otherwise, this is the number of blocks in this covered/virtual area that are needed in normal LXX, IXX, or data buffers that are requested from MRDF. For a virtual area, each of these is also considered an MRDF_READ. For a covered area using the FIRST option, each of these is also counted in either the MRDF_READS or the NONFIRST_READS. For a covered area using the ACTIVE option, each of these is also counted in either the MRDF_READS or the ACTIVE_READS.
VIRTUAL_COVERED	CHAR(1)	No	Contains a V to represent a virtual base or a C to represent a covered base.
VIRTUAL_WRITES	INTEGER	Yes	Null unless this is an opened area in a base defined as virtual. For opened virtual areas this is the number of times a block was written from the normal buffers to a MRDF buffer.

MUF_DATA_SHARING (MFD)

This table is provided to allow the monitoring of the MUF in a Data Sharing environment. It provides real-time information about the types and frequencies of requests being made through the Coupling Facility to/from this MUF.

Column Name	SQL Data Type	Nullable	Description
ALL_CALLS	INTEGER	No	Total number of calls made to the Coupling Facility using XES.
ENT_RATIO	SMALLINT	No	Denominator of fraction representing ratio of elements to entries.

Column Name	SQL Data Type	Nullable	Description
EL_RATIO	SMALLINT	No	Numerator of the fraction representing ratio of elements to entries. The mean list entry size should be approximated by EL_RATIO * 256 / ENT_RATIO.
HIGH_USED_LIST_ENT	INTEGER	No	High water mark for number of entries in list structure.
HIGH_USED_LIST_EL	INTEGER	No	High water mark for number of 256-byte entries in list structure.
HIGH_USED_BUFL_ENT	INTEGER	No	High water mark for number of 256-byte entries used for buffers.
LIST_CALLS	INTEGER	No	Total calls made to the Coupling Facility to support list structure processing.
LIST_READS	INTEGER	No	Times a list record was requested for read.
LIST_RESETS	INTEGER	No	Times a list lock was released.
LIST_SETS	INTEGER	No	Times a list lock was requested.
LIST_WRITES	INTEGER	No	Times a list record was requested to be written or deleted.
LOCK_ALTERS	INTEGER	No	Times a lock structure lock was escalated from share to exclusive (excluding buffers).
LOCK_B_ALTERS	INTEGER	No	Times a lock structure buffer lock was escalated from share to exclusive.
LOCK_B_CONTENTIONS	INTEGER	No	Times contentions occurred for a lock structure buffer lock.
LOCK_B_OBTAINS	INTEGER	No	Times a lock structure buffer lock was requested.
LOCK_B_RELEASES	INTEGER	No	Times a lock structure buffer lock was released.
LOCK_CALLS	INTEGER	No	Total calls made to the Coupling Facility to support lock structure processing.
LOCK_CONTENTIONS	INTEGER	No	Times contentions occurred for a lock structure lock (excluding buffers).

Column Name	SQL Data Type	Nullable	Description
LOCK_OBTAINS	INTEGER	No	Times a lock structure lock was requested (excluding buffers).
LOCK_RELEASES	INTEGER	No	Times a lock structure lock was released (excluding buffers).
MAX_LIST_ENT	INTEGER	No	The approximate maximum number of entries allowed in the list structure.
MAX_LIST_EL	INTEGER	No	The approximate maximum number of 256-byte entries allowed in the list structure.
MAX_LIST_STR_SIZE	DECIMAL(11,0)	No	Maximum storage of list structure.
MESSAGES_IN	INTEGER	No	Times a message was received from another MUF in support of Data Sharing processing.
MESSAGES_OUT	INTEGER	No	Times a message was sent to another MUF in support of Data Sharing processing.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
REFUSED_LOCK_RQ	INTEGER	No	Number of times that contention processing rejected a lock request.
SRB_USES	INTEGER	No	Times an SRB was executed in support of the Coupling Facility processing.

MUF_IDENTITY (MFI)

This table is for system identification. It contains one row. It also provides MUF-wide information which is not provided through the startup options.

Column Name	SQL Data Type	Nullable	Description
CURRENT_DATETIME	TIMESTAMP	No	Contains the current date and time from the MUF.
DIR_NAME	CHAR(8)	No	The Directory (CXX) name being used by this MUF.
ENABLE_TIME	TIMESTAMP	No	The date and time that the MUF was enabled.

Column Name	SQL Data Type	Nullable	Description
MUF_JOBID	CHAR(8)	No	Contains the MUF JOB ID if z/OS or blanks if z/VSE.
MUF_JOBNAME	CHAR(8)	No	The MUF job name provided by the operating system.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
MUF_SYSTEM_NAME	Char (8)	No	Name of the system on which MUF is running if z/OS, or blanks if z/VSE.
NODE_NAME	CHAR(8)	No	Contains either the CA Datacom STAR node name or blanks.
PC	Char (1)	No	In z/OS, this column contains a P if the production PC is being used, or T if the test PC is being used. This is specified in the DBSYSID macro, PC parameter and normally is P. For z/VSE, it contains the SUBID of the DBSYSID macro.
SECURITY_DD	CHAR(8)	No	CA Datacom Datadictionary security is being done through EXTERNAL security checking or INTERNAL CA Datacom Datadictionary coding.
SECURITY_DQ	CHAR(8)	No	CA Dataquery security is being done through EXTERNAL security checking or INTERNAL CA Dataquery coding.
SECURITY_DQR_CICS	CHAR(2)	No	Occurs with table used for externally securing DQ RAAT requests in CICS.
SECURITY_DQS_CICS	CHAR(2)	No	Occurs with table used for externally securing DQ SQL requests in CICS.
SECURITY_DQR_OTHER	CHAR(2)	No	Occurs with table used for externally securing DQ RAAT requests *not* in CICS.
SECURITY_DQS_OTHER	CHAR(2)	No	Occurs with table used for externally securing DQ SQL requests *not* in CICS.

Column Name	SQL Data Type	Nullable	Description
SECURITY_RAT_CICS	CHAR(2)	No	Defines external security for record-at-a-time and set-at-a-time calls made through CA Datacom CICS Services. Contains NO if not externally secured or two characters indicating which external security table is to be used for checking authorizations.
SECURITY_RAT_SRVR	CHAR(2)	No	Defines external security for record-at-a-time and set-at-a-time calls made through CA Datacom Server or Windows ODBC or gateway access. Contains NO if not externally secured or two characters indicating which external security table is to be used for checking authorizations.
SECURITY_RAT_OTHER	CHAR(2)	No	Defines external security for record-at-a-time and set-at-a-time calls made through any interface that is not one of the preceding two (SECURITY_RAT_CICS or SECURITY_RAT_SRVR). Contains NO if not externally secured or two characters indicating which external security table is to be used for checking authorizations.
SECURITY_SQL_CICS	CHAR(2)	No	Defines external security for SQL calls made through CA Datacom CICS Services. Contains NO if not externally secured or two characters indicating which external security table is to be used for checking authorizations.
SECURITY_SQL_SRVR	CHAR(2)	No	Defines external security for SQL calls made through CA Datacom Server or Windows ODBC or gateway access. Contains NO if not externally secured or two characters indicating which external security table is to be used for checking authorizations.

Column Name	SQL Data Type	Nullable	Description
SECURITY_SQL_OTHER	CHAR(2)	No	Defines external security for SQL calls made through any interface that is not one of the preceding two (SECURITY_SQL_CICS or SECURITY_SQL_SRVR). Contains NO if not externally secured or two characters indicating which external security table is to be used for checking authorizations.
SERVICE_PACK	CHAR(2)	No	Set to blank, column exists for compatibility with prior versions.
VERSION_LEVEL	CHAR(6)	No	Version level of CA Datacom/DB, for example, '14.0 ' (that is 14.0 plus two blanks to make six characters).

MUF_INTERNAL_STATS (MFV)

This table is designed to provide what is termed internal statistics and contains one row. This information could be important for the current version but is subject to change between versions and possibly within a version. The addition or removal of fields is very likely.

Column Name	SQL Data Type	Nullable	Description
BREAKS_DONE	DECIMAL(11,0)	No	Number of times the startup option BREAK occurred to cause a task to allow a task switch. It does not indicate if another task of equal or higher priority was ready to run and was dispatched.
DEFRAG_REDBR	INTEGER	No	Contains the count of the number of times backwards processing (such as LOCBR) interrupted an active DBUTLTY DEFRAG function.
INDEX_Q_CUR_ENTS	INTEGER	No	Current number of unprocessed entries in the index queue.
INDEX_Q_MAX_ENTS	INTEGER	No	Maximum number of unprocessed entries in the index queue for the life of this MUF.

Column Name	SQL Data Type	Nullable	Description
INDEX_Q_CUR_DBIDS	SMALLINT	No	Contains the current number of unique DBIDs with unprocessed entries in the index queue.
INDEX_Q_MAX_DBIDS	SMALLINT	No	Contains the maximum number of unique DBIDs with unprocessed entries in the index queue for the life of this MUF.
IO_DASD_RETRY	INTEGER	No	Count of times the operating system required CA Datacom to reissue a physical I/O because of a "retry" return code.
IO_MEM_SHORT	INTEGER	No	Times I/O restricted in z/OS due to 24-bit memory shortage.
MRDF_IXX_DIRECT	DECIMAL(11,0)	No	Contains the count of times the MRDF facility referenced an IXX block directly, without moving it to an IXX buffer.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
TIMES_READ_1	INTEGER	No	During sequential GETIT/GETPS, CBS, and ADDIT processing it is possible that CA Datacom/DB would choose to read multiple blocks with one EXCP. This counter reflects the number of times that only one block was read.
TIMES_READ_2_8	INTEGER	No	During sequential GETIT/GETPS, CBS, and ADDIT processing it is possible that CA Datacom/DB would choose to read multiple blocks with one EXCP. This counter reflects the number of times that two through eight blocks were read.
TIMES_READ_9_32	INTEGER	No	During sequential GETIT/GETPS, CBS, and ADDIT processing it is possible that CA Datacom/DB would choose to read multiple blocks with one EXCP. This counter reflects the number of times that 9 through 32 blocks were read.

Column Name	SQL Data Type	Nullable	Description
TIMES_READ_33_99	INTEGER	No	During sequential GETIT/GETPS, CBS, and ADDIT processing it is possible that ,CA Datacom/DB would choose to read multiple blocks with one EXCP. This counter reflects the number of times that 33 or more blocks were read.
TIMES_WRITE_1	INTEGER	No	With pipelining, multiple blocks in an area frequently need to be written. If these blocks are physically close on DASD, CA Datacom/DB attempts to write more than one at a time. This count is the times a write was necessary to the index or data and only one block needed to be or could be written.
TIMES_WRITE_2_12	INTEGER	No	See previous description. This is a count of the times that 2 through 12 blocks could be written together in one EXCP.
TIMES_WRITE_13_24	INTEGER	No	See previous description. This is a count of the times that 13 through 24 blocks could be written together in one EXCP.
TIMES_WRITE_25_48	INTEGER	No	See previous description. This is a count of the times that 25 through 48 blocks could be written together in one EXCP.
XCF_ALL_PATHS_BUSY	INTEGER	No	Count of times an XCF message send failed because all XCF signaling paths were busy resulting in a retry.
XCF_MISC_ENV_ERROR	INTEGER	No	Count of times an XCF message send failed for environmental reasons other than paths busy or too few message buffers resulting in a retry.
XCF_REJECT_MSG_IN	INTEGER	No	Count of messages received by MUF with characteristics which made them unacceptable for processing.

Column Name	SQL Data Type	Nullable	Description
XCF_RETRY_LIM_HIT	INTEGER	No	Count of the number of XCF messages for which the retry limit was exceeded and therefore had to be queried for action by a subtask.
XCF_SUBSYS_FAILURE	INTEGER	No	Count of times an XCF message send failed because the MVS XCF component failed.
XCF_TOO_FEW_MSGBUF	INTEGER	No	Count of times an XCF message send failed due to a shortage of XCF buffers resulting in a retry.

MUF_LOCKS_VALUE (MFK)

This table contains information about the current contents of the value lock table. One row exists for each value lock in existence. A list of specific value locks is provided with the DBUTLTY function COMM OPTION=STATUS. Value locks are utilized by a MUF to protect a process that needs protection.

Column Name	SQL Data Type	Nullable	Description
DURATION	CHAR(6)	No	Contains the time the lock has been held in the format *mmm:ss* where *mmm* is minutes and *ss* is seconds. If the lock is a share, then the time is for the first task getting the share. The formatting is the same as the ACTIVE_TASKS and STATUS in that a value of less than one second is reflected as all blanks and a value of greater than 999:59 is reflected as +++:++.
JOBNAME	CHAR(8)	No	Contains the job name owning this task. If a job name begins with three asterisks (***), it means that the value lock is owned by a system task.
LOCK_LENGTH	SMALLINT	No	Contains the length of the lock in its internal form. A length of over 50 indicates the value has been truncated. This condition allows duplicates to not be provided.

Column Name	SQL Data Type	Nullable	Description
LOCK_TYPE	CHAR(9)	No	Contains the type of lock. It can be EXCLUSIVE or SHARE.
LOCK_VALUE	CHAR(64)	No	Contains the value lock with limited formatting. Known values locked, that have binary or hexadecimal fields, are formatted in a useful way. Unknown values are formatted in another way with character and hexadecimal values provided. This formatted format is how the table must be queried, if all values are being acquired. Because of the truncation of the value, actual unique entries that are duplicated within the first externalized 64 bytes have only the first occurrence externalized (this condition is very unlikely). The formatting sets the first 6 bytes to be the first 6 bytes of the value lock, followed by a blank, and then match that done for the ACTIVE_TASKS conditions of waiting for a value lock. It is also the same format as that used for the console STATUS command and the DBUTLTY FUNCTION=COMM,OPTION=STATUS.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
TASK_NUMBER	SMALLINT	No	Contains the task number owning this value lock. Some internal locks are acquired and assigned to system tasks or are assigned to the primary task of the job.

MUF_LOGGING (MFL)

This table exists to provide information about the logging environment and status. The startup options pertaining to logging are provided in the MUF_OPTIONS table. This table has a single row.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Contains LXX as the area name to easily join this row with the MUF_AREA_STATS table for logical and physical write information.
BLOCK_LENGTH	SMALLINT	No	Always set to 32760. MUF optimizes blocks that are written to match the data needing to be written.
BLOCKS_SPILLABLE	INTEGER	No	Contains zero if currently spilling. Otherwise, it contains the number of tracks that are spillable at this point in time (which may also be zero).
CYCLE_NUMBER	INTEGER	No	A cycle is a contiguous group of LXX tracks to be written to the Recovery File (RXX). If currently a spill is being done (SPILL=Y), then this cycle number represents the cycle being spilled. If a spill is not being done (SPILL=N), then this cycle number represents the current unspilled LXX tracks, some or all of which become the next spilled cycle.
DBID	SMALLINT	No	Contains zero as the database ID to easily join this row with the MUF_AREA_STATS table for logical and physical write information.
GENERATION	SMALLINT	No	Value returned is 1 for generation 1.

Column Name	SQL Data Type	Nullable	Description
IN_USE_BLOCKS	INTEGER	No	Number of tracks in the Log Area that contain active log records. If spilling is active, this includes all tracks in the cycle being spilled in addition to the tracks built after the blocks being spilled. At the completion of a spill, the spilled tracks are removed from the active tracks. To calculate a percent full, use the IN_USE_BLOCKS and the TOTAL_BLOCKS.
LAST_SPILLED_TIME	TIMESTAMP	Yes	Contains a null if a spill has not been completed during this execution of the MUF. Otherwise, it contains the date and time of the last record spilled.
LOG_BLOCK_SEQ	DECIMAL (21,0)	No	Contains the assigned log track sequence number of the log record in the Log Area that is one number higher than the previous log track.
LOG_RECORD_SEQ	DECIMAL (21,0)	No	Each record written to the Log Area is assigned an ascending sequence number. The last number assigned is provided here. The value of the LOG_RECORD_SEQ cannot be predicted when a MUF is enabled. However, it is predictable that each log record has the next higher number within the same MUF session. This number is available here so that log activity can be measured, such as how many records are being logged in a given period of time (by reading this record at two points in time within the same MUF execution and subtracting the first LOG_RECORD_SEQ value from the second).

Column Name	SQL Data Type	Nullable	Description
LOG_RECORD_SEQ_G0	INTEGER	No	Specifies the Log Record Sequence number as an integer as it existed in the generation 0 table. This column is being maintained for compatibility but may not accurately represent the new larger values supported in the current release. If the true value is larger than will fit in an integer, this column is set to high values.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
PHY_PER_LOG_BLOCK	SMALLINT	No	Contains a 1 with the recommended variable logging. If using spanned logging for fallback to Version 14.0, PHY_PER_LOG_BLOCK contains a 2 or a 3.
RESTART_BLOCK_REDO	INTEGER	No	The MUF takes into account the startup option LOGPEND and also the current activity in active requests and index and data buffers that are pipelined (write pended). From this information comes a minimum number of log tracks that RESTART processing needs to redo, during MUF startup, to ensure all index and data in the pipeline are completed. It can be a little higher than LOGPEND but is usually much lower. This track count provides a good indication of the amount of work and, therefore, the time a restart must reprocess work.

Column Name	SQL Data Type	Nullable	Description
RESTART_BLOCK_SCAN	INTEGER	No	Contains the number of log tracks that the RESTART process during MUF startup processing would have to review to ensure all tasks are seen and processed. The scan part of RESTART must simply read through the Log Area from a point that would cause the current request for each active task (RWTSA) doing maintenance to be seen. Also, the last maintenance request for each task that is subject to transaction backout must be seen. This scan runs quickly, but the number of tracks is dependent on the current activity and is not predictable. This scan is *not* controlled by the LOGPEND parameter.
RXX_NUMBER	INTEGER	No	This is the number which is assigned to the next Recovery File (RXX) produced. In a test system with the Multi-User startup option LOGRCV NEVER specified, this number is unchanged because no Recovery File will be produced. For inactive recovery, each SPILL of log tracks causes a new number to be assigned. For active recovery, the MUF enable starts a new RXX and each COMM NEWRXX causes another. The number starts at zero when the Log Area is initialized. The number represents the number of recovery files built since the Log Area INIT.
SPILL_BLOCKS_DONE	INTEGER	No	As a spill is prepared, this field is set to zero. It is incremented as individual log tracks are read from the Log Area to be copied to the Recovery File.

Column Name	SQL Data Type	Nullable	Description
SPILL_BLOCKS_TOTAL	INTEGER	No	As a spill is prepared, this field is set to contain the number of tracks that are to be spilled. This number of tracks is also called a cycle on the Recovery File.
SPILLING	CHAR(1)	No	Contains an R to indicate that a spill has been prepared and is committed to start. Contains a Y to indicate that a spill has been started and is in process. Contains an N to indicate that no spill is currently requested and none is currently in process. In an active environment, the spill is automatically started and runs through completion. In an inactive environment, the spill is a utility function which must be started either through an operator or could also be started through an automated tool watching for console messages.
TOTAL_BLOCKS	INTEGER	No	Number of log tracks (logical) that are available for logging records. Currently, the first track is reserved as a control block. The last physical track is also reserved.
TOTAL_TRACKS	INTEGER	No	Number of tracks that are being used for the Log Area.
VARIABLE	CHAR(1)	No	Indicates if the LXX was initialized as variable.
VARIABLE_MAX_GT8	SMALLINT	Yes	For variable logging, this column contains a count, if you have allocated more than 8 buffers. It tells you the maximum number of buffers that have actually been used. If not null, it cannot be less than 9 nor more than the number of log-ahead buffers. It represents a high-water mark of buffer usage, because the detail used counts stop at 8.

Column Name	SQL Data Type	Nullable	Description
VARIABLE_USED_1	INTEGER	No	This count is the number of times the first log-ahead buffer is activated. It occurs only when the single current buffer is full and no log-ahead buffers are in use. If activated, the buffer can contain only one record or a full block of records before it is written.
VARIABLE_USED_2_4	INTEGER	No	Count of times the second, third, or fourth log-ahead buffer was activated.
VARIABLE_USED_5_7	INTEGER	No	Count of times the fifth, sixth, or seventh log-ahead buffer was activated.
VARIABLE_USED_8	INTEGER	No	Count of times 8 or more log-ahead buffers were activated.
VARIABLE_WAIT_SMP	INTEGER	No	This is the count of times all log-ahead buffers were full and a wait needed to occur for a current block to free one or more buffers. If this count is large, consider increasing the number of log-ahead buffers. This column indicates whether the log was subjected to an INIT with the variable specified as yes (Y) or no (N).
VARIABLE_WRITE_2_3	INTEGER	No	Contains a count of the number of times the current log block was written with a chain of 2 or 3 log blocks in a single I/O.
VARIABLE_WRITE_4_6	INTEGER	No	Contains a count of the number of times the current log block was written with a chain of 4, 5 or 6 log blocks in a single I/O.

Column Name	SQL Data Type	Nullable	Description
WAIT_COMIT_BAD	INTEGER	No	If this MUF uses the X_LOG_DELAY Multi-User startup option, this column contains the count of times a task doing a transaction-ending command delayed for another task to write the LXX block but none did within the specified time limit. If this MUF does not use the X_LOG_DELAY Multi-User startup option, this column is 0.
WAIT_COMIT_GOOD	INTEGER	No	If this MUF uses the X_LOG_DELAY Multi-User startup option, this column contains the count of times a task doing a transaction-ending command delayed for another task to write the LXX block and another did, therefore saving a possible LXX I/O. If this MUF does not use the X_LOG_DELAY Multi-User startup option, this column is 0.
WAIT_CURR_LOG_BLK	INTEGER	No	Informational count of the times that a task had to wait on access to the log buffer that is used for adding log records. No user action is required related to this count. It provides a clue of contention to the current log block. It is used by tasks adding log records, possibly by transaction backout, or if writing an index or data block which has information in this current log block.
WAIT_CURR_LOG_RCD	INTEGER	No	Informational count of the times a task wanted to add a log record, but had to wait on another task already in the process of adding a log record. This count can be large, but no action is possible. Most counts occur as each log block is written, many tasks may want to log a record during this time and wait. The count is provided as a clue to log contention.

MUF_MEM_DETAIL (MMD)

For the MUF_MEM_DETAIL (MMD) Dynamic System Table, one row exists for each MUF for most memory pools in existence, with the exception that some with the same descriptions are combined.

The MMD table is modeled after the Memory Use Summary Information page of the EOJ report. The MUF_MEM_DETAIL table contains individual pools. The MUF_MEM_SUMMARY table contains summary information. This table and the MUF_MEMORY_SUMMARY table are user-friendly ways of providing the information retrieved using the ?MEM console command.

64-bit MUF memory usage is not reflected in MUF_MEM_DETAIL table.

The MUF_MEM_DETAIL table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
MEM_AMOUNT	DECIMAL(11,0)	No	Total memory allocated to this pool.
MEM_DESC	CHAR(4)	No	A short description that can be blanks or repeated, as appropriate. For a list of these descriptions, refer to the information in the EOJ report.
MEM_TYPE	CHAR(2)	No	Value of 24 or 31, indicating whether memory is 24-bit or 31-bit.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
OCCURENCES	SMALLINT	No	This column lists the number of occurrences of LUW, STME, STMR, and XCL (refer to the information in the EOJ report).

MUF_MEM_SUMMARY (MMS)

The MUF_MEM_SUMMARY Dynamic System Table contains one row that summarizes current MUF memory use. This table and the MUF_MEMORY_DETAIL table are user-friendly ways of providing information retrieved using the ?MEM console command.

The MUF_MEM_SUMMARY table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
LARGE_PAGES_IN_1M	DECIMAL(11,0)	No	The total number of SYSPOOL index buffer 64-bit memory allocations in 1M increments being backed by large pages.
MEM_REQUESTS	DECIMAL(11,0)	No	The number of requests made to the MUF memory manager. This count includes both get memory and free memory requests.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
OUT_OF_MEM_COUNT	INTEGER	No	The number of times memory was requested but could not be honored from the MUF memory manager, because of insufficient available memory.
PGM_MEM_24BIT_IN_K	DECIMAL(11,0)	No	This shows most but not all 24-bit program memory, divided by 1024 (in K).
PGM_MEM_31BIT_IN_K	DECIMAL(11,0)	No	This shows most but not all 31-bit program memory, divided by 1024 (in K).
TOTAL_24BIT_IN_K	DECIMAL(11,0)	No	Number of 1024 segments allocated in 24-bit program memory.
TOTAL_31BIT_IN_M	DECIMAL(11,0)	No	Number of 1,048,576 segments allocated in 31-bit program memory.
TOTAL_64BIT_IN_M	DECIMAL(11,0)	No	Number of 1,048,576 segments allocated above the 64-bit bar.

MUF_ML_DUMPS_PREV (MFF)

Provides information about the last (up to 10) Master List dumps that were requested in the MUF. One row exists for each dump with identifying information. Once the maximum (of 10 rows) is reached, each additional row that is added forces the oldest row to be deleted. The Statistics and Diagnostics Area (PXX) becoming full has no effect on the rows being added to this table. Additionally, clearing the PXX (or optionally, the PXXML in an z/OS environment) has no effect on the rows of this table. The column data and rules match the MUF_ML_DUMPS_PXX (MFE) table.

Column Name	SQL Data Type	Nullable	Description
DB_COMMAND	CHAR(5)	No	The five-character record-at-a-time or set-at-a-time command being executed or last executed. This field can contain certain internal commands that are only used by CA Support.
DBID	SMALLINT	No	Contains the database ID.
DUMP_NUMBER	INTEGER	No	Dumps are numbered from one as the first after the MUF has been enabled.
DUMP_REASON	CHAR(1)	No	The DUMP_REASON can be: **T** TEST (CA Datacom/DB record-at-a-time command). **S** SNAP console or DBUTLTY COMM option. **D** Internal condition (not error) needing a Master List dump. Typically done in a *patch* used in debugging. Not normally seen in the GA version. **M** Special multi-tasking error dump. Should not occur. **E** Error with return code requesting a Master List dump.
EXTERNAL_CODE	CHAR(2)	No	The EXTERNAL_CODE is blanks if the reason is not E.

Column Name	SQL Data Type	Nullable	Description
INTERNAL_CODE	CHAR(3)	No	The INTERNAL_CODE is 000 if the reason is not E.
JOB_NAME	CHAR(8)	No	The job name for a console SNAP request is ***DBCOM.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
PROGRAM_NAME	CHAR(4)	No	The PROGRAM_NAME is blanks if the reason is not E or D.
PROGRAM_OFFSET	CHAR(4)	No	The PROGRAM_OFFSET is blanks if the reason is not E or D. For a few conditions the offset is 0000.
R15_CODE	SMALLINT	No	The R15_CODE is 0 if the reason is not E. The R15 is useful to CA Support, if that becomes necessary.
RUN_UNIT	INTEGER	No	The run unit number (zero for console SNAP).
TABLE_NAME	CHAR(3)	No	Name of the table.
TASK_NUMBER	SMALLINT	No	The number of the task area being used for this request, also known as the RWTSA number. The low numbers are allocated for user applications based upon the TASKS Multi-User startup option. Next are CA Datacom STAR task areas, if any, followed by internal CA Datacom/DB system task areas.

MUF_ML_DUMPS_PXX (MFE)

Provides information about Master List dumps (up to 10) that are currently in the Statistics and Diagnostics Area (PXX) (or the optional PXXML in an z/OS environment). One row exists for each dump with identifying information. Once 10 rows have been added, rows for additional dumps are not written to this table. Clearing the PXX (or PXXML) removes the actual Master List dumps and deletes the corresponding row(s) from this table. Printing the PXX has no effect on the rows of this table. The column data and rules match the MUF_ML_DUMPS_PREV (MFF) table.

Column Name	SQL Data Type	Nullable	Description
DB_COMMAND	CHAR(5)	No	The five-character record-at-a-time or set-at-a-time command being executed or last executed. This field can contain certain internal commands that are only to be used by CA Support.
DBID	SMALLINT	No	Contains the database ID.
DUMP_NUMBER	INTEGER	No	Dumps are numbered from one as the first after the MUF has been enabled.
DUMP_REASON	CHAR(1)	No	The DUMP_REASON can be: **T** TEST (CA Datacom/DB record-at-a-time command). **S** SNAP console or DBUTLTY COMM option. **D** Internal condition (not error) needing a Master List dump. Typically done in a *patch* used in debugging. Not typically seen in the GA version. **M** Special multi-tasking error dump. Should not occur. **E** Error with return code requesting a Master List dump.
EXTERNAL_CODE	CHAR(2)	No	The EXTERNAL_CODE is blanks if the reason is not E.

Column Name	SQL Data Type	Nullable	Description
INTERNAL_CODE	CHAR(3)	No	The INTERNAL_CODE is 000 if the reason is not E.
JOB_NAME	CHAR(8)	No	The job name for a console SNAP request is ***DBCOM.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
PROGRAM_NAME	CHAR(4)	No	The PROGRAM_NAME is blanks if the reason is not E or D.
PROGRAM_OFFSET	CHAR(4)	No	The PROGRAM_OFFSET is blanks if the reason is not E or D. For a few conditions the offset is 0000.
R15_CODE	SMALLINT	No	The R15_CODE is 0 if the reason is not E. The R15 is useful to CA Support, if that becomes necessary.
RUN_UNIT	INTEGER	No	The run unit number (zero for console SNAP).
TABLE_NAME	CHAR(3)	No	Name of the table.
TASK_NUMBER	SMALLINT	No	The number of the task area being used for this request, also known as the RWTSA number. The low numbers are allocated for user applications based upon the TASKS Multi-User startup option. Next are CA Datacom STAR task areas, if any, followed by internal CA Datacom/DB system task areas.

MUF_OPEN_BASES (MFB)

This table provides information about open databases. One row exists for every open User Requirements Table, both external to the MUF and internal to the MUF. It is designed primarily for CA Datacom STAR sites to provide information about what bases are open by what jobs and at what distributed nodes. Also, this table provides what jobs locally have a database open. This can be valuable to CA Datacom STAR and non-CA Datacom STAR sites.

Column Name	SQL Data Type	Nullable	Description
DBID	SMALLINT	No	The database ID for this row. For a CA Datacom STAR environment, this is the data base ID as it is known by the reporting MUF.
JOB_NAME	CHAR(8)	No	Job name of a job that has at least one URT open against a table in this database.
JOB_NODE	CHAR(8)	No	Contains blanks in a non-CA Datacom STAR MUF. In a CA Datacom STAR MUF, this field contains the CA Datacom STAR node name from the MUF that the application is connected to.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
PARTITION	CHAR(1)	No	Indicates whether this base is part of a CA Datacom STAR partitioned database. This flag is set to Y only if the DBID is the object of a $PARTITIONED relationship in CA Datacom Datadictionary therefore, this flag reflects a partitioned child.
REMOTE	CHAR(1)	No	Indicates whether the database ID represents a database which resides at a remote MUF using CA Datacom STAR. STAR_BASE must be Y for this flag to be Y.

Column Name	SQL Data Type	Nullable	Description
REMOTE_DBID	SMALLINT	Yes	Contains null when the REMOTE field contains N. Otherwise, it contains the database ID as it is known at the MUF where the database resides. Thus, this database ID may match or differ from the DBID field.
REMOTE_NODE	CHAR(8)	Yes	Contains null in a non-CA Datacom STAR environment. Also contains null if this database is local or if REMOTE is N. For a remote database, contains the node name of the MUF at the remote site.
REPLICA	CHAR(1)	No	Specifies whether this base is part of a CA Datacom STAR replicated database. This flag is set to Y only if the DBID is the object of a $REPLICATED relationship in CA Datacom Datadictionary. If the DBID is a child of a partitioned base, then the PARTITION flag is also set to Y.
RUN_UNIT	INTEGER	No	Contains the run-unit number for this job. The run unit number could be zero for an opening job, but a job may be assigned the run unit number of zero. For a job running at another MUF, the run unit number is the number assigned the job at the remote location. The run unit number matches the reported run unit number in messages that range from 0 through 99,999.

Column Name	SQL Data Type	Nullable	Description
STAR_BASE	CHAR(1)	No	Indicates whether this database exists to support CA Datacom STAR services. **N** Indicates a normal local database with an index and data areas. **Y** Indicates this database exists only for CA Datacom STAR usage. With a Y, the BAS definition in CA Datacom Datadictionary has either a $RESIDES relationship to a remote MUF or a $PARTITIONED or $REPLICATED relationship to a child base.
URT_DBID	SMALLINT	No	Contains a database ID as it exists in the application programs User Requirements Table (URT). For local databases and non-CA Datacom STAR environments, the URT_DBID matches the DBID. If the JOB_NODE reflects a remote MUF, the URT_DBID is the database ID as it is known at the remote MUF node. If the DBID is a partitioned or replicated database, then this database ID may reflect the root base.
URT_NUMBER	SMALLINT	No	Contains the external User Requirements Table Number for the User Requirements Table with this database open. Each URT open by a job has a unique number ranging from -32768 through 32767 except that zero is now displayed as 10000. The specific number has no importance and may appear random.

MUF_OPTIONS (MFO)

This table contains the general startup options that existed before Version 12.0. For information about newer startup options, see the MUF_CONFIG (MCF) section. Most MUF startup information is available but not all. Options which allow more than one occurrence are in other tables, such as the Accounting table or the CA Datacom products table. This table has only one row. Some of the fields are changeable, and the values reported are the current value at the time this row is read. An example is any of the FLEXPOOL fields. For these changeable fields, the value at the MUF startup is *not* available. CA Datacom/DB takes the liberty of rounding some of the buffer sizes up to doubleword or other boundaries. Expect this possibility in your reporting. A full description of these fields is contained in the section on modifying Multi-User startup options in the *CA Datacom/DB Database and System Administration Guide*.

Column Name	SQL Data Type	Nullable	Description
ACCTPRM_ANN_DBID	SMALLINT	No	Contains zero or the database ID which contains the Accounting data areas.
ACCTPRM_CPU_TIME	CHAR(1)	No	Indicates whether CPU time is accumulated when Accounting is enabled.
ACCTPRM_PRM_DBID	SMALLINT	No	Contains zero or the database ID of the Accounting parameter table.
AGENT	CHAR(1)	No	This column is provided for prior version compatibility and contains an N.
AGENT_LOG_LEVEL	SMALLINT	No	This column is provided for prior version compatibility and contains a 0 (zero).
AGENT_USE_CONFIG	CHAR(1)	No	This column is provided for prior version compatibility and contains an N.
BREAK	INTEGER	No	Contains the current value of the BREAK Multi-User startup option.
CBS_BUFFER	INTEGER	No	Contains the size of the CBS buffer.
CBS_DBID_HEURISTIC	SMALLINT	No	Contains database used for heuristic information or 0 (zero) if no DBID has been specified.
CBS_DBID_TEMP	INTEGER	No	Contains the database ID where CBS stores various index information.

Column Name	SQL Data Type	Nullable	Description
CBS_MAXSTEN	INTEGER	No	Contains the CBS temporary index entry limit.
CBS_MAXSTIO	INTEGER	No	Contains the CBS set I/O limit.
CBS_TIME_DELETE	SMALLINT	No	Age of set not referenced where it can be deleted.
CSAFREE	CHAR(1)	No	This column is provided for prior version compatibility and contains an N.
DATAPOOL_DATALN	SMALLINT	No	Specifies the data buffer size for normal data buffers.
DATAPOOL_DATALN2	SMALLINT	No	Specifies zero or the data buffer size for large data buffers.
DATAPOOL_DATANO	INTEGER	No	Specifies the number of normal data buffers.
DATAPOOL_DATANO2	INTEGER	No	Specifies the number of large data buffers.
DATASPACE_CONNECT	CHAR(1)	No	In z/OS, this is a Y indicating that a data space is used to connect between address spaces. In z/VSE, the only value is N.
DATASPACE_SIZE	INTEGER	No	Contains the size, in bytes, for any data spaces defined. Zero indicates the maximum of 2-GB.
DATETIME_DATE	CHAR(3)	No	Format for SQL date usage.
DATETIME_TIME	CHAR(3)	No	Format for SQL time usage.
DICTIONARY_DD_DBID	SMALLINT	No	The database ID for the primary DATA-DICT database.
DICTIONARY_DDDDBID	SMALLINT	No	The database ID for the primary Data Definition Directory database.
EXCTLNO	INTEGER	No	Zero or record exclusive control limit.
EXPAND_LENGTH	INTEGER	No	This column is provided for prior version compatibility and contains a 0 (zero).

Column Name	SQL Data Type	Nullable	Description
EXPAND_NUMBER	INTEGER	No	This column is provided for prior version compatibility and contains a 0 (zero).
FLEXPOOL_DATANO	INTEGER	No	Number of current data FLEXPOOL buffers.
FLEXPOOL_DATANO2	INTEGER	No	Number of current large data FLEXPOOL buffers.
FLEXPOOL_DXXNO	INTEGER	No	Number of current DXX FLEXPOOL buffers.
FLEXPOOL_IXXNO	INTEGER	No	Number of current IXX FLEXPOOL buffers.
IOTASK_CURRENT	SMALLINT	No	Current number of usable I/O tasks.
IOTASK_MAXIMUM	SMALLINT	No	Maximum number of I/O tasks.
LOGOPTION_OPN_CLS	CHAR(1)	No	Force writing OPEN and CLOSE records to the Log Area (LXX) for maintenance run units if Y. N indicates no OPEN on the log and only CLOSE if it is a commit.
LOGPEND	SMALLINT	No	Size of pipeline of DXX and data buffers, based upon number of log blocks.
LOGPOOL	SMALLINT	No	Number of log buffers for transaction backout.
LOGRCV	CHAR(5)	No	Specifies which logging recovery option was selected.
LOGRSYS	SMALLINT	No	The z/VSE SYS number for the Recovery File (RXX).
LOGSPILL_A	SMALLINT	No	Log percent for console message.
LOGSPILL_B	SMALLINT	No	Log percent for force spill.
LOGSPILL_C	SMALLINT	No	Log percent for starting spill.
LOGSPILL_D	SMALLINT	No	Log percent to force from log scan.
LOGSPILL_E	SMALLINT	No	Log percent to force from log redo.

Column Name	SQL Data Type	Nullable	Description
MAXELRQ	SMALLINT	No	Maximum elements per request.
MUF_ENDED_NO_LOG	CHAR(1)	No	Y or N, with N as the default. With N set or defaulted, a console message (DB00102I) only occurs in MUF if the job ending did maintenance which was logged. With Y set, a console ended message occurs for all jobs. The N option is recommended to reduce the console clutter.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
MUF_RUN_UNIT_GROUP	SMALLINT	No	Number of run units to group during MUF execution to reduce CXX I/O.
MUFMSG_A	CHAR(1)	No	Include MUF job name as message prefix.
MUFMSG_B	CHAR(1)	No	Include SVC number as message prefix.
MUFMSG_C	CHAR(1)	No	Include SUBID number as message prefix.
MUFPLEX_NAME	CHAR(8)	No	If running with full Data Sharing as a MUFplex, name of the group.
MUFPLEX_NUMBER	SMALLINT	No	Number of this specific MUF within this MUFplex group.
MUFPLEX_LOCKS	INTEGER	No	Number of lock hash entries to control performance in Data Sharing.
MUFPLEX_MAX_TASKS	SMALLINT	No	Maximum number of tasks per MUF within this MUFplex group.
NONSWAP	CHAR(1)	No	Request MUF be set nonswappable (z/OS).
PLANSEC_CHECKPLAN	CHAR(1)	No	SQL plan security.
PLANSEC_CHECKWHEN	CHAR(8)	No	SQL plan security, when to check.
PLANSEC_CHECKWHO	CHAR(8)	No	SQL plan security, who to check.

Column Name	SQL Data Type	Nullable	Description
PROCEDURE_CACHE	INTEGER	No	Size of cache used to save information about the execution of procedures.
PROCEDURE_NESTS	SMALLINT	No	Number representing the depth of ability for a procedure that is executing to issue SQL statements that cause another procedure to run.
PROCEDURE_TCBS	SMALLINT	No	Number of operating system subtasks to attach to support the procedure processing.
PXX_STATS	CHAR(6)	No	Level of statistics information in PXX.
READAHD	CHAR(8)	No	Type of sequential reading ahead (pre-8.1 URTs).
RESTART_ACCESS_OFF	CHAR(1)	No	Contains a Y if errors are ignored (as described in RESTART_IGNORE), and additionally indicates that the database ID with the errors is to be set to ACCESS OFF to prevent it from being easily opened until the errors are corrected.
RESTART_IGNORE	CHAR(1)	No	Contains a Y if errors during the reprocessing of the restart phase of the MUF startup are to be ignored and the MUF enabled (not all errors are ignorable). Contains an N if errors are to stop the enable so that users can correct the problem and rerun the MUF for correct results.
RRS	CHAR(8)	No	Contains blanks if not specified, CONTINUE if specified as continue, or FAIL if specified as fail.
RXX_UNIT_COUNT	SMALLINT	No	Count of units for RXX (z/OS).
RXX_UNIT_NAME	CHAR(8)	No	Name for RXX unit (z/OS).
RXX_VOLUME_COUNT	SMALLINT	No	Number of volumes to request.
RXXROLLBACK	CHAR(1)	No	Specifies force spill at Log Area full (Y) versus force checkpoint at Log Area full (N).

Column Name	SQL Data Type	Nullable	Description
SMPTASK_CURRENT	SMALLINT	No	Current number of SMP tasks.
SMPTASK_MAXIMUM	SMALLINT	No	Maximum number of SMP tasks.
SMPTASK_READY_TASK	SMALLINT	No	Ready-to-run work required to start another SMP task.
SQLDEFAULT_DBID	SMALLINT	No	Zero or the database ID for the default database for SQL tables.
SQLDEFAULT_NAME	CHAR(32)	No	Default area name for SQL tables.
SQLOPTION_BOTH	CHAR(1)	No	A Y indicates that both update and read-only cursors are allowed in a plan. An N indicates that either an update or read-only cursor is allowed in a plan.
SQLOPTION_MODE	CHAR(7)	No	Mode of SQL support.
SQLOPTION_OPTION	CHAR(1)	No	SQL activation option.
SQLOPTION_TIMEOUT	SMALLINT	No	Minutes before a CICS SQL logical unit of work is closed.
SQLOPTION_TTMID	SMALLINT	No	Zero or the database ID of the SQL TTM database.
SQLOPTION_VIEW	CHAR(1)	No	External security checking at the view-level default. All plans which have a specified selection of view security or table-level are excluded from this default. It applies on every compile or rebind as the default. If Y is specified, all security within the plan for a view is done through view security. If N is specified, all security within the plan is done with table-level security.
STAR_MAX_NODES	SMALLINT	No	Maximum number of nodes with which CA Datacom STAR can connect.
STAR_NODE	CHAR(8)	No	Name of the node which this MUF is known as, for (DST) usage.
STAR_SIZE	INTEGER	No	Size of working storage for CA Datacom STAR usage.

Column Name	SQL Data Type	Nullable	Description
STAR_TASKS	SMALLINT	No	Number of tasks (RWTSAs) to reserve for remote CA Datacom STAR requests.
STAR_TIME_BATCH	SMALLINT	No	Seconds CA Datacom STAR waits for a remote request for record locking in batch.
STAR_TIME_CICS	SMALLINT	No	Seconds CA Datacom STAR waits for a remote request for record locking in CICS.
STATBFR	INTEGER	No	This column is provided for prior version compatibility and contains as 0 (zero).
SUBTASK	CHAR(8)	No	Blanks or name of a subtask to run with MUF.
SYSOUT_CLASS	CHAR(1)	No	z/OS throw-away class.
SYSPOOL_CXXNO	SMALLINT	No	Number of Directory (CXX) buffers.
SYSPOOL_DXXNO	INTEGER	No	Current number of DXX buffers.
SYSPOOL_INDEX	SMALLINT	No	Provides the index buffer size.
SYSPOOL_IXXNO	INTEGER	No	Current number of IXX buffers.
SYSTEMDBID_DBID	SMALLINT	No	Zero or the database ID of the database containing the Dynamic System Tables.
SYSTEMDBID_SYSOUT	CHAR(1)	No	In z/OS, Y indicates that changes are written to SYSOUT, but N indicates that changes are not written to SYSOUT. In z/VSE, changes are written to the PXX.
TASKS_NUMBER	SMALLINT	No	Number of tasks (RWTSAs).
TASKS_SIZE	INTEGER	No	Size of the task area.
TASKS_VAE_NUMBER	SMALLINT	No	Number of tasks in the MUF which support XPCC connection (z/VSE).
TASKS_VM_NUMBER	SMALLINT	No	Number of tasks in the MUF which support VM IUCV connection.
TASKS_XCF_NUMBER	SMALLINT	No	Number of tasks in the MUF which support XCF connection (z/OS).

MUF_PRODUCTS (MFP)

This table is for CA products or features that can be in use in the CA Datacom/DB system. One record exists for each optional product which could be provided in the Multi-User startup option DATACOM. A Y (YES) or N (NO) flag indicates whether the product was requested. This provides what was requested during the startup, not necessarily what products are installed or could have been selected.

Column Name	SQL Data Type	Nullable	Description
AVAILABLE	CHAR(1)	No	Indicates whether the product or feature was specified in the Multi-User startup option DATACOM.
MUF_NAME	CHAR(8)	No	Contains the name of the MUF specified in the MUF Multi-User startup option or the operating system job name.
PRODUCT	CHAR(18)	No	One of the products which are allowable in the Multi-User startup option DATACOM.

MUF_RATES (MFR)

This table is a simple, low overhead way of watching Multi-User performance. It contains one row. Approximately every two minutes, Multi-User saves selected information about activity which has been done. If an SQL query is done against the MUF_RATES table, the saved information is reviewed and the fields that follow are calculated as required and provided for access. The data provided is the same during the approximate two-minute interval. The time is based upon operating system facilities and timings and can therefore vary, but the two-minute value cannot be changed by the user.

Column Name	SQL Data Type	Nullable	Description
CPU_USED	INTEGER	No	The number of seconds of CPU time used by the MUF address space during the last completed time interval.

Column Name	SQL Data Type	Nullable	Description
DATA_SEQUENCE	INTEGER	No	The number of data buffer references per second averaged over the last completed time interval. Note that data buffers include both normal and large size buffers.
DB_REQUESTS	INTEGER	No	The number of CA Datacom/DB requests per second averaged over the last completed time interval. Note that this request count is the same as reported in the Statistics and Diagnostics Area (PXX) report summary statistics as DB REQUESTS. It is also reported as DB_REQUESTS in the MUF_SYSTEMS_STATS field.
DXX_SEQUENCE	INTEGER	No	The number of low-level index (DXX) buffer references per second averaged over the last completed time interval.
ELAPSED_SECONDS	INTEGER	No	The number of elapsed seconds which occurred in the last completed time interval. Expect the times to vary based upon operating system features and processor availability.
IXX_SEQUENCE	INTEGER	No	The number of high-level index (IXX) buffer references per second averaged over the last completed time interval.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
PHYSICAL_READS	INTEGER	No	The number of physical read I/O events per second averaged over the last completed time interval. This count includes the Directory (CXX), Log Area (LXX), all Index Areas (IXX...), and data areas (xxx...).

Column Name	SQL Data Type	Nullable	Description
PHYSICAL_WRITES	INTEGER	No	The number of physical write I/O events per second averaged over the last completed time interval. This count includes the Directory (CXX), Log Area (LXX), Dual Log Area (NXX), all Index Areas (IXX...), and data areas (xxx...).

MUF_RC_DUMP_OPT (MDO)

In the MUF_RC_DUMP_OPT Dynamic System Table, one row exists for each CA Datacom return code, not only for those return codes that have occurred. The MUF_RC_DUMP_OPT table allows you to determine the PXX dump options in effect for a given return code. The MUF_RC_DUMP_OPT contains the following columns:

Column Name	SQL Data Type	Nullable	Description
DUMP_OPTION	CHAR(8)	No	Can be one of the following: BUFFER, ML, MINIMAL, or NONE. **BUFFER** BUFFER means that Master List and buffers dump are generated. **ML** ML means that a Master List dump with no buffer dump is generated. **MINIMAL** MINIMAL means that a minimal dump is generated. A minimal dump is a small, tailored dump that is suitable for a few simple errors, for example, an invalid command. **NONE** NONE means that no dump is produced.

Column Name	SQL Data Type	Nullable	Description
EXTERNAL_CODE	CHAR(2)	No	CA Datacom/DB passes a return code, that is, an **external** return code, to the Request Area of a program in response to the program issuing a CA Datacom/DB command. CA Datacom/DB also issues return codes within CA Datacom/DB messages, that is, **internal** return codes. All non-blank external return codes have one or more associated internal return codes. If there is only one internal return code associated with the return code, it is not separately listed in the *CA Datacom/DB Message Reference Guide* since the description in that case of the external and internal return codes is the same.
INTERNAL_CODE	CHAR(3)	No	See the explanation of internal codes given in the description of EXTERNAL_CODE.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.

MUF_RETURN_CODES (MFU)

This table allows you to determine which CA Datacom/DB return codes have occurred in this execution of the MUF. It has a key of the external return code concatenated with the internal return code.

Column Name	SQL Data Type	Nullable	Description
COUNT_CODES	INTEGER	No	Each full return code has a count of occurrences. The table has rows for only those errors that have occurred at least once. Several return codes that occur frequently in non error situations are not counted. These include, but are not restricted to, return codes: 14, 19, and 04(127). Also not counted are certain return codes that occur outside the MUF. These include, but are not restricted to, return codes: 36, 37, 57, 68, 85, and 86.
EXTERNAL_CODE	CHAR(2)	No	CA Datacom/DB passes a return code, that is, an *external* return code, to the Request Area of a program in response to the program issuing a CA Datacom/DB command. CA Datacom/DB also issues return codes within CA Datacom/DB messages, that is, *internal* return codes. All non-blank external return codes have one or more associated internal return codes. If there is only one internal return code associated with the return code, it is not separately listed in the *CA Datacom/DB Message Reference Guide* since the description in that case of the external and internal return codes is the same.
INTERNAL_CODE	CHAR(3)	No	See the explanation of internal codes given in the description of EXTERNAL_CODE.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.

Column Name	SQL Data Type	Nullable	Description
SHORT_TEXT	CHAR(50)	No	Each unique return code has this short text to speed the review of errors and save time that could be spent looking up message meanings.

MUF_SMP_STATS (MFW)

This table is provided to allow monitoring of information relating to the running of the MUF in SMP mode. It contains one row. Two types of fields exist.

The first type of field supplies information to tune the number of SMP tasks and the count to start more. These fields currently include the POSTED_0 through POSTED_24_999. Periodically, CA Datacom/DB samples the list of posted tasks (units of work) that are ready to run. At each sample, one is added to one of the POSTED counters. The counter selected is based on the total entries in the list. The proportion of values in the POSTED counters reflects the depth of the backlog of work outstanding.

All other fields belong to the second type of field. They are informational and no tuning changes them. They provide the number of lock contentions for each of the SMP-type internal locks. All locks are held for a minimum number of instructions. Contention exists between one TCB which owns the lock and another TCB which wants the lock when one of them needs an exclusive lock. In this case, the second TCB must wait. Since no user action is possible, each is briefly described.

Column Name	SQL Data Type	Nullable	Description
ACCESS_AREA	INTEGER	No	Contains number of times wait occurred for the area level access.
ACCOUNTING_SMP	INTEGER	No	Accounting SMP locks.
ACCOUNTING_STATUS	INTEGER	No	The status of an Accounting table is in the process of being changed.
ACCOUNTING_TABLE	INTEGER	No	A specific Accounting tables buffer is being changed.
ADD_DELETE_FLEX	INTEGER	No	The add or delete of a FLEXPOOL buffer. Also, the add or delete of a sequential buffer.
CBS_BUFFER	INTEGER	No	Adding to or deleting from the CBS buffer.

Column Name	SQL Data Type	Nullable	Description
CHECK_INTERLOCK	INTEGER	No	Checking to see if a lock conflict (record, value, system) would cause an interlock.
COVEREDVIRTUAL	INTEGER	No	A specific Covered or Virtual area while processing a block.
CXX_BUFFER	INTEGER	No	Count of times a task waited on a specific Directory (CXX) block to become available.
DATA_BUFFER	INTEGER	No	A specific data buffer being updated.
DATA_HASH	INTEGER	No	The hash-type lookup for a data buffer.
DATA_LRU	INTEGER	No	The Least Recently Used chain of data buffers.
DATA_SPACE_BUFFER	INTEGER	No	Record add looking for space in a specific buffer.
DATA_SPACE_MGMT	INTEGER	No	Data space management adding a block to, or removing a block from, the space index for a specific area.
DATA_WRITE_PEND	INTEGER	No	Changing the data buffer write-pending list.
DSF_IN_MUF	INTEGER	No	CA Datacom Datadictionary requests in the MUF.
DUMP_LIST	INTEGER	No	Count of times a task waited while attempting to manage the CA Datacom/DB dump list because another task was concurrently managing the dump list.
DXX_BUFFER	INTEGER	No	A specific DXX buffer is being updated.
DXX_HASH	INTEGER	No	The hash type lookup for a DXX buffer.
DXX_LRU	INTEGER	No	The Least Recently Used chain of DXX buffers.
DXX_WRITE_PEND	INTEGER	No	Changing the DXX buffer write-pending list.

Column Name	SQL Data Type	Nullable	Description
EXPAND_BUFFER	INTEGER	No	This column is provided for prior version compatibility and contains 0 (zero).
HISTORY_SPILL	INTEGER	No	Count of times a task had to wait on the history database spill SMP lock.
INDEX_QUEUE	INTEGER	No	Changing the index queue.
IXX_BUFFER	INTEGER	No	A specific IXX buffer being updated.
IXX_HASH	INTEGER	No	The hash-type lookup for an IXX buffer.
IXX_LRU	INTEGER	No	The Least Recently Used chain of IXX buffers.
LOCK_LIST	INTEGER	No	Searching or changing the lock (exclusive control) list.
LOG_BUFFER	INTEGER	No	Changing a specific Log buffer.
LOG_LRU	INTEGER	No	The Least Recently Used chain of LOGPOOL log buffers for transaction backout.
LOGGING_CHAINS	INTEGER	No	Log processing chains.
LOGGING_FORCE	INTEGER	No	Log force condition.
MEMORY_MANAGER	INTEGER	No	Change in memory allocation.
MISCELLANEOUS	INTEGER	No	Miscellaneous small structures.
MUF_NAME	CHAR(8)	No	Name of the MUF.
OPEN_TABLE	INTEGER	No	Changes in the list of open tables.
OPEN_USER_EXIT	INTEGER	No	Contains number of times wait occurred to get to *not* reentrant user open exit.
PLEX_CLOSE_STAT	INTEGER	No	Waits to write-close stats.
PLEX_LOCK_BUFFER	INTEGER	No	Waits for index or data buffer coupling record.
PLEX_LOCK_DBYU	INTEGER	No	Waits for lock/read of lock index assignment coupling record.
PLEX_LOCK_LOG_C	INTEGER	No	Waits for lock/read for log control/current coupling record.

Column Name	SQL Data Type	Nullable	Description
PLEX_LOCK_LOG_U	INTEGER	No	Waits for lock/read for log uncertainty coupling record.
PLEX_LOCK_RUNUNIT	INTEGER	No	Waits for lock/read for run unit grouping coupling record.
PLEX_LOCK_TSN	INTEGER	No	Waits for TSN assignment coupling record.
PLEX_LOG_RCD	INTEGER	No	Waits for the current log record from the coupler.
PLEX_REBUILD	INTEGER	No	MUFplex rebuild process.
PLEX_TSN_READ	INTEGER	No	Times a read was required to read the TSN record from the coupler.
PLEX_TSN_RETRY	INTEGER	No	Times a retry occurred during the assignment of a TSN to start a new transaction.
POSTED_0	DECIMAL (11,0)	No	Times the posted list entries were counted and the list was empty.
POSTED_1_5	DECIMAL (11,0)	No	Times the posted list entries were counted and the list contained one through five entries.
POSTED_6_10	INTEGER	No	Times the posted list entries were counted and the list contained six through ten entries.
POSTED_11_23	INTEGER	No	Times the posted list entries were counted and the list contained 11 through 23 entries.
POSTED_24_999	INTEGER	No	Times the posted list entries were counted and the list contained 24 or more entries.
POSTED_LIST	INTEGER	No	Some changes to the list of posted (ready-to-run) units of work.
PXX_CONSOLE	INTEGER	No	Writes to the PXX or writes to the console.
SECURITY_CHECK	INTEGER	No	Changes to the security buffer or tables.
SQL_ATTACH	INTEGER	No	Changes to the SQL attach list.
SQL_GLOBAL	INTEGER	No	Changes to SQL global areas.

Column Name	SQL Data Type	Nullable	Description
SQL_MEM_MGR	INTEGER	No	SQL memory manager.
SQL_MISC	INTEGER	No	Changes to minor SQL blocks.
SQL_PLAN	INTEGER	No	Changes to the SQL plan list.
SQL_PROCEDURE	INTEGER	No	Waits for a procedure TCB.
SQL_STATUS	INTEGER	No	Changes in SQL status conditions.
SQL_TTM	INTEGER	No	Getting/freeing SQL TTM blocks.
SQL_URT	INTEGER	No	Changes to the SQL User Requirement Lists.
STAR	INTEGER	No	General CA Datacom STAR control changes.
STAR_TASK	INTEGER	No	Changes in CA Datacom STAR task status.
TASK_AREA	INTEGER	No	Changes in a task area (RWTSA) status.
TSN	INTEGER	No	Changes to the active transaction list.
USER_COMPRESSION	INTEGER	No	Control information around a user compression exit.
WAIT_COUPLER	INTEGER	No	Waits for Coupling Facility actions.
WAIT_LIST	INTEGER	No	Changes in the list of waiting tasks.
WAIT_MUF_MUFPLEX	INTEGER	No	Waits for another MUF within the MUFplex.
WAIT_PLEX_AREA	INTEGER	No	Waits for lock/read of data area coupling record.
WAIT_PLEX_BASE	INTEGER	No	Waits for lock/read of database coupling record.
WAIT_PLEX_TABLE	INTEGER	No	Waits for lock/read of table coupling record.
WAIT_XCF	INTEGER	No	Waits for a message when using XCF for communication.
XES_MSG_REPLY	INTEGER	No	Waits for a message reply from another MUFplex MUF.

MUF_SMP_TASK (MFM)

This table exists and can be queried in all environments, but it is intended for and useful only in z/OS running the MUF in SMP mode as set by the SMPTASK startup option. When running in SMP mode, multiple z/OS tasks (TCBs) can be dispatched to perform the full range of CA Datacom/DB MUF processing.

This table contains one row for the main task and one for any SMPTASKS generated. It shows the counts of times that each TCB selected and did a unit of work. A request provided MUF, once started, is a unit of work. It continues until a reason to stop occurs. When restarted, it is considered a new unit of work. This table only shows the level of activity done by each TCB. To be most useful, read this table and save the results. After a period of time, repeat and compare the difference to show activity during the time period.

Column Name	SQL Data Type	Nullable	Description
ACTIVE	CHAR(1)	No	Indicates if this SMP task is currently active.
MUF_NAME	CHAR(8)	No	Name of the MUF as specified in the MUF Multi-User startup option or the operating system job name.
SMP_TASK_ID	SMALLINT	No	Contains the number of the SMP task. The MUF main task is always number 1. SMP tasks are numbered 2 through one higher than the number specified in the startup option SMPTASK.
TIMES_USED	DECIMAL(13,0)	No	Number of times this TCB was used to perform an SMP-type unit-of-work.

MUF_SRB_ZIIP (MZI)

(z/OS only) If SRB mode was specified with the SMPTASK Multi-User startup option, the MUF_SRB_ZIIP (MZI) Dynamic System Table has one row for each SMP subtask that was designated on the SMPTASK Multi-User startup option parameter used to specify the maximum number of SMP subtasks,

Information provided by the MZI table is updated periodically by the specific subtask, based on usage counts, which means that it is therefore not necessarily current. It does, however, provide performance information with minimal impact.

The MUF_SRB_ZIIP table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
ENCLAVE_SECONDS	INTEGER	No	CA Datacom SMP SRBs run in an Enclave. This is the total CPU time (in seconds) that this SMP task has used for the life of this MUF.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the MUF job name.
SEQUENCE_NUMBER	SMALLINT	No	A sequence number that uniquely identifies the SMP task. For example, sequence number 1 is the first SMP task, sequence number 2 is the second SMP task, and so on.
ZIIP_ON_CP_SECONDS	INTEGER	No	If the ZIIP Multi-User startup option was specified for this SMP subtask, this is the total CPU seconds spent on a standard (non-zIIP) processor that would have run on a zIIP processor but a zIIP processor was not available. If the ZIIP Multi-User startup option was not specified, this column is 0 (zero).
ZIIP_PERCENT	SMALLINT	No	This is 100 if no user limit has been placed on zIIP processing with the ZIIP_USER_LIMIT Multi-User startup option. Otherwise, this is the user-specified limit.
ZIIP_QUAL_SECONDS	INTEGER	No	The total time in seconds for which this SMP subtask was qualified to run on a zIIP processor.
ZIIP_SECONDS	INTEGER	No	The actual total CPU time this SMP subtask ran on a zIIP processor.

MUF_SYSOUT (MFG)

In z/OS only, this table provides information about the SYSOUT Multi-User startup option that directs diagnostic and tracing information to SYSOUT files instead of to the Statistics and Diagnostics Area (PXX).

Column Name	SQL Data Type	Nullable	Description
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
SYSOUT_NAME	CHAR(3)	No	Name of the directed output: ML, CBS, SQL, DD, or DST.
OUTLIM_NUMBER	INTEGER	No	Output limit or zero.
REDIRECTED_NAME	CHAR(3)	No	If the SYSOUT_NAME output is redirected to another SYSOUT, this is the name of that other SYSOUT. If the SYSOUT_NAME is not redirected to another SYSOUT, the ROUTED_NAME is blanks.

MUF_SYSTEM_STATS (MFS)

MUF_SYSTEM_STATS (MFS) is an old table delivered for compatibility with prior releases. Use MUF_SYSTEMS_STATS (MSS) instead because it has a capacity for larger numbers.

MUF_SYSTEMS_STATS (MSS)

The following table is designed as a replacement table for MUF_SYSTEM_STATS (MFS). It has many columns with a larger capacity than the old table and removes obsolete columns that are in other dynamic systems tables or provide no value. We recommend that you start using this table to collect systems statistics even though the MFS table still exists.

Note: Columns whose lengths are different between the two tables are in italics. Otherwise, all column characteristics are the same as the old table.

Column Name	SQL Data Type	Nullable	Description
BEGIN_TIME	TIMESTAMP	No	Date and time the MUF was enabled. This timestamp would need to be the same if this record was read at two points in time and the difference was being computed to see activity during a time period. (If the timestamp is different on the second reading of the record, the MUF has been recycled.)
CURRENT_CPU	INTEGER	No	Number of CPU seconds used by the MUF address space since startup.
CURRENT_DATETIME	TIMESTAMP	No	Current date and timestamp from the MUF.
DATA_MGR_REQUESTS	DECIMAL(13,0)	No	Specifies the number of calls to the low-level CA Datacom/DB processor. Each request to the MUF that involves a CA Datacom/DB RAAT or SAAT type command generates one data manager request. Each request involving an SQL request generates as many data manager requests as necessary to process the SQL request.
DATA_SEQUENCE	DECIMAL(13,0)	No	Number of times a data buffer was referenced.
DATA_USED_1	DECIMAL(13,0)	No	Number of times a data buffer was referenced the first time after the buffer was read from DASD or copied from MRDF.

Column Name	SQL Data Type	Nullable	Description
DATA_USED_2	DECIMAL(11,0)	No	Number of times a data buffer was referenced a second time after the buffer was read from DASD or copied from MRDF.
DATA_USED_3	DECIMAL(11,0)	No	Number of times a data buffer was referenced a third time after the buffer was read from DASD or copied from MRDF.
DATA_USED_4	DECIMAL(11,0)	No	Number of times a data buffer was referenced a fourth time after the buffer was read from DASD or copied from MRDF.
DATA_USED_5	DECIMAL(13,0)	No	Number of times a data buffer was referenced five times (and six times, and seven times, and so forth) after the buffer was read from DASD or copied from MRDF.
DATA2_USED_1	INTEGER	No	Number of times a data2 buffer (large-size data buffer) was referenced once after the buffer was read from DASD or copied from MRDF.
DATA2_USED_2	INTEGER	No	Number of times a data2 buffer (large-size data buffer) was referenced twice after the buffer was read from DASD or copied from MRDF.
DATA2_USED_3	INTEGER	No	Number of times a data2 buffer (large-size data buffer) was referenced three times after the buffer was read from DASD or copied from MRDF.
DATA2_USED_4	INTEGER	No	Number of times a data2 buffer (large-size data buffer) was referenced four times after the buffer was read from DASD or copied from MRDF.

Column Name	SQL Data Type	Nullable	Description
DATA2_USED_5	DECIMAL(11,0)	No	Number of times a data2 buffer (large-size data buffer) was referenced five times (and six times, and seven times, and so forth) after the buffer was read from DASD or copied from MRDF.
DB_REQUESTS	DECIMAL(13,0)	No	Number of calls from an application program either directly or through another tool (such as SQL) which was shipped to the MUF. Blocked GETIT/GETPS requests are counted here as one request per block. Block SQL requests are the same, one request per block.
DELETED_BLOCKS	INTEGER	No	Number of times the last entry in an index block was deleted.
DXX_SEQUENCE	DECIMAL(13,0)	No	Number of times a DXX buffer was referenced.
DXX_USED_1	DECIMAL(11,0)	No	Number of times a DXX buffer was referenced once after the buffer was read from DASD or copied from MRDF.
DXX_USED_2	DECIMAL(11,0)	No	Number of times a DXX buffer was referenced twice after the buffer was read from DASD or copied from MRDF.
DXX_USED_3	DECIMAL(11,0)	No	Number of times a DXX buffer was referenced three times after the buffer was read from DASD or copied from MRDF.
DXX_USED_4	DECIMAL(11,0)	No	Number of times a DXX buffer was referenced four times after the buffer was read from DASD or copied from MRDF.
DXX_USED_5	DECIMAL(13,0)	No	Number of times a DXX buffer was referenced five times (and six times, and seven times, and so forth) after the buffer was read from DASD or copied from MRDF.

Column Name	SQL Data Type	Nullable	Description
INDEX_Q_PROCESSED	INTEGER	No	The number of index queue entries processed. Queue processing includes: ■ All high-level block processing ■ Block deletions of low-level blocks This statistic is useful primarily to CA Support for debugging purposes.
IXX_SEQUENCE	DECIMAL(13,0)	No	Number of times an IXX buffer was referenced.
IXX_USED_1	DECIMAL(11.0)	No	Number of times an IXX buffer was referenced once after the buffer was read from DASD or copied from MRDF.
IXX_USED_2	DECIMAL(11,0)	No	Number of times an IXX buffer was referenced twice after the buffer was read from DASD or copied from MRDF.
IXX_USED_3	DECIMAL(11,0)	No	Number of times an IXX buffer was referenced three times after the buffer was read from DASD or copied from MRDF.
IXX_USED_4	DECIMAL(11,0)	No	Number of times an IXX buffer was referenced four times after the buffer was read from DASD or copied from MRDF.
IXX_USED_5	DECIMAL(13,0)	No	Number of times an IXX buffer was referenced five times (and six times, and seven times, and so forth) after the buffer was read from DASD or copied from MRDF.
LOG_PERCENT_FULL	INTEGER	No	The highest percent full the Log Area reached.

Column Name	SQL Data Type	Nullable	Description
LOG_WRITE_COMMAND	INTEGER	No	The number of times the Log buffer containing logged data was written because of a command. Commands which cause transaction boundaries (COMIT, LOGCP, LOGCR, ROLBK, CLOSE) are always written. The ADDIT, DELET, and UPDAT maintenance commands are only written if the table definition indicates writes, the URT indicates writes, or the Multi-User startup option LOGPEND indicates no pipelining. Commands which are optional are LOGIT and LOGDW. Some internal system commands can also write the Log block prior to full. Comparing the ratios of COMMAND WRITE to BUFFER FULL WRITE I/O can be used to attempt tuning by reducing the number of LXX writes. The number of writes to the Log Area may not equal these detail counts due to multitasking events.
LOG_WRITE_CONTROL	INTEGER	No	The number of times the Log control block was written. The control block information is required for the restart process during MUF. This counter is part of the information to provide details about the LXX I/O. The number of writes to the Log Area may not equal the detail counts due to multitasking events.
LOG_WRITE_FULL	INTEGER	No	The number of times the log buffer containing logged data was written when it was considered to be full. This count is the number of Log Area (LXX) blocks used. LXX block size tuning is reflected here. The number of writes to the Log Area may not equal these detail counts due to multitasking events.

Column Name	SQL Data Type	Nullable	Description
LOG_WRITE_OTHER	INTEGER	No	Indicates the number of times a non-full Log block had to be written to support RESTART when an index or data block had to be written. The number of writes to the Log Area may not equal these detail counts due to multitasking events.
LOG_WRITE_TXB	INTEGER	No	Indicates the number of times a Log block was written during transaction or request backout.
LOG_WRITE_2_PHASE	INTEGER	No	The number of I/Os done to the log Area in support of the CA Datacom STAR 2 phase commit processing. This is also used in special cases without CA Datacom STAR in the system.
LOGPEND_WRITES	INTEGER	No	Number of writes to an index or data buffer due to the setting of the Multi-User startup option LOGPEND.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
NO_TXB_BUFFER	INTEGER	No	Specifies the number of times a task was in transaction or request backout, but all LOGPOOL buffers were not available.
PHYSICAL_READS	DECIMAL(11,0)	No	Specifies the number of physical I/Os (EXCPs) that have been done to read data from DASD.
PHYSICAL_WRITES	DECIMAL(11,0)	No	Specifies the number of physical I/Os (EXCPs) that have been done to write memory data to DASD.
RECORD_LOCKS	DECIMAL(11,0)	No	Specifies the number of times a record was locked. Includes exclusive control locks and, for SQL, share locks.
RQ_PENDING_1_10	DECIMAL(11,0)	No	Specifies the number of times tasks waited on resources other than I/O and there were already nine or less tasks waiting on resources.

Column Name	SQL Data Type	Nullable	Description
RQ_PENDING_11_20	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 10 through 19 tasks waiting.
RQ_PENDING_21_30	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 20 through 29 tasks waiting.
RQ_PENDING_31_40	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 30 through 39 tasks waiting.
RQ_PENDING_41_50	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 40 through 49 tasks waiting.
RQ_PENDING_51_60	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 50 through 59 tasks waiting.
RQ_PENDING_61_70	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 60 through 69 tasks waiting.
RQ_PENDING_71_80	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 70 through 79 tasks waiting.
RQ_PENDING_81_90	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 80 through 89 tasks waiting.
RQ_PENDING_91_100	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already 90 through 99 tasks waiting.
RQ_PENDING_GT_100	INTEGER	No	Specifies the number of times tasks waited on resources other than I/O and there were already more than 99 other tasks waiting on resources.

Column Name	SQL Data Type	Nullable	Description
SECONDARY_CONFLICT	INTEGER	No	Specifies the number of times a record lock was attempted where it was not currently available because the record was subject to transaction backout.
SEQ_READ_AHEAD	INTEGER	No	Specifies the number of times a read-ahead I/O was done in support of the GETIT or GETPS command.
SPLIT_DXX	INTEGER	No	Specifies the number of times that a DXX-level index block was split. This indicates growth of an index.
SPLIT_IXX	INTEGER	No	Specifies the number of times that an IXX-level index block was split.
SQL_REQUESTS	INTEGER	No	Specifies the number of requests issued to the MUF of the SQL style.
WAIT_ACCOUNTING	INTEGER	No	Specifies the number of times a task had to wait when adding a record to an Accounting buffer because the Accounting buffer was full.
WAIT_LOCK	INTEGER	No	Specifies the number of times a task requested a record-level lock that was not available.
WAIT_REQUEST	INTEGER	No	Specifies the number of times that a task had to wait on an active request of another task. This involves internal CA Datacom/DB processing and currently can occur during a Log Area full condition and during close processing.
WAIT_SPILL	INTEGER	No	Specifies the number of times that a task had to wait on the completion of a spill. This occurs when the Log has no space to add records until a spill is completed and space made available.

Column Name	SQL Data Type	Nullable	Description
WAIT_TASK	INTEGER	No	Specifies the number of times that a task had to wait on the completion of another task. This is done when a secondary conflict exists and the task needs to wait for the commit or rollback of the task last maintaining a data record.

MUF_TABLE_STATS (MFT)

This table provides table-level information about the number and types of requests processed by this MUF. It contains one row for every table that has been opened to this MUF. This information is always from the start of the MUF to the current time. This includes the RESTART process done during the MUF enable, all activity for ended jobs, and all activity for active jobs. This table shows a view of the logical activity for comparison purposes with the area logical and physical I/O activity and the summary statistics.

Column Name	SQL Data Type	Nullable	Description
AREA_NAME	CHAR(3)	No	Name of area which contains the table. Provided to easily join these rows with the area statistics rows.
BEGIN_TIME	TIMESTAMP	No	Date and time of the MUF enable. Helpful if saving copies of this current data in a user-defined table as history information.
CURRENT_DATETIME	TIMESTAMP	No	Current MUF date and time.
DBID	SMALLINT	No	The database ID which contains this table.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
TABLE_NAME	CHAR(3)	No	Name of the table.

Column Name	SQL Data Type	Nullable	Description
TOTAL_ADDS	DECIMAL(11,0)	No	Contains the number of rows added to this table. As a measure of work done, it includes adds attempted but failed and also adds which were reversed during transaction backout.
TOTAL_DELETES	DECIMAL(11,0)	No	Number of rows deleted from this table. Includes deletes done during transaction backouts reversing an add.
TOTAL_READS	DECIMAL(11,0)	No	Number of data read type requests against this table.
TOTAL_REQUESTS	DECIMAL(11,0)	No	Number of data manager requests made against this table.
TOTAL_UPDATES	DECIMAL(11,0)	No	Number of updates to this table. Includes updates done during transaction backouts reversing updates or deletes.

MUF_TCB_OR_SRB (MTC)

The MUF_TCB_OR_SRB Dynamic System Table is similar to the MUF EOJ TCB Use Summary Information report produced at MUF EOJ. Therefore, for additional information see the *CA Datacom/DB DBUTLTY Reference Guide*.

Statistics provided by this table are updated periodically by the specific subtask. The information in this table is not completely current, but it does provide performance information with minimal impact.

The MUF_TCB_OR_SRB table contains the following columns:

Column Name	SQL Data Type	Nullable	Description
CPU_SECONDS	INTEGER	No	Total CPU seconds used by this task for the life of this MUF.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF startup option or the MUF job name.

Column Name	SQL Data Type	Nullable	Description
PROGRAM_NAME	CHAR(8)	No	Name of the program using this TCB or SRB. See the list from the DBUTLTY EOJ report in the *CA Datacom/DB DBUTLTY Reference Guide*.
PHYSICAL_IO	DECIMAL(11,0)	No	The number of I/Os this TCB or SRB performed.
SEQUENCE_NUMBER	SMALLINT	No	A sequence number which uniquely identifies a task, where more than one share the same PROGRAM_NAME (for example, DBSMPPR).
TASK_TYPE	CHAR(4)	No	Contains MAIN for the main MUF task, SMP for SMP subtask, or SUB for other subtask.
TIMES_POSTED	DECIMAL(13,0)	No	Lists number of times individual TCBs or SRBs went into a wait (no work to do) and were posted to start processing. Once any of the TCBs are posted, they run (and take new work) until no work exists to be done. The information provided in this field could be useful in tuning the ready-to-run option on the SMPTASK Multi-User startup option.
TIMES_USED	DECIMAL(13,0)	No	The number of times an internal unit of work was performed by this task. Each program has its own considerations for what a unit of work is, which means that this count is therefore not comparable to other CA Datacom counters.

MUF_XCF (MFX)

This table contains the startup options from the XCF_FROM statement. It contains one row for every XCF_FROM Multi-User startup option specified. The table therefore remains empty except for a Multi-User that is using the XCF Coupling Facility.

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

Column Name	SQL Data Type	Nullable	Description
CURRENT_STATUS	CHAR(6)	No	Either DELETE, NO, or YES, as specified with the X option.
JOBNAME	CHAR(8)	No	The job name from which this MUF can accept requests, as specified with the XCF_FROM option.
JOBS_CURRENT	INTEGER	No	The number of active jobs which are connected to a MUF.
JOBS_TOTAL	INTEGER	No	The number of jobs that were received.
MUF_NAME	CHAR(8)	No	Name of the MUF specified in the MUF Multi-User startup option or the name of the MUF job name.
REQUESTS_CURRENT	INTEGER	No	The number of active requests which are connected.
REQUESTS_TOTAL	DECIMAL(11,0)	No	The number of requests that were received.
SYSTEM_NAME	CHAR(8)	No	System from which this MUF can accept requests, as specified with the XCF_FROM option
XCF_GROUP	CHAR(8)	No	Name of the group containing the jobs identified in the MUF_NAME column from which this MUF can accept requests.

Chapter 5: SQL Tables

SQL Maintenance Table

Maintenance commands to Dynamic System Tables can be issued using a Dynamic System Table named SQL_CONSOLE (SQX). SQL maintenance is only done to the SQX table.

The SQX table lets you do an insert with a character -string that is formatted by all the console commands. The command is processed as if issued through the console. The results, if any, are returned to the caller. The SQX table can therefore be used to issue the same commands that can be requested using the DBUTLTY function COMM OPTION=CONSOLE,OPTION2=. For details about using OPTION2=, see the CA Datacom/DB DBUTLTY Reference Guide).

The ability for SQL to insert, update, or delete Dynamic System Table rows is folded into the common component console APIs that allow the MUF changes to occur. The MUF changes occur the same as they do through all the console-like paths. For example, in a previous release, a user of SQL could change the BREAK value by reading and updating the BREAK field in the MUF_OPTIONS table.After Version 12.0, SQL users can instead issue the change as an insert into the SQL_CONSOLE (SQX) table.

Note: For information about CA Datacom/DB maintenance using console commands, see the CA Datacom/DB Database and System Administration Guide.

If a command would normally produce console output, that output is written to the console and is also returned to the application in the SQL feedback area.

The SQX table can be externally secured, but external security is not content based. With external security, therefore, any user who has ADD rights to a table can issue any insert into the table, causing the console-like request to be executed. If you are concerned about the possibility that this ability could be misused, the table can be deleted from CA Datacom Datadictionary and the database can be cataloged without the deleted table being present.

SQL_CONSOLE (SQX)

The SQL_CONSOLE table consists of two columns with the following characteristics:

Column Name	**SQL Data Type**	**Nullable**	**Description**
MUF_NAME	CHAR(8)	No	Logical name for this MUF.

Column Name	SQL Data Type	Nullable	Description
CONSOLE_COMMAND	CHAR(126)	No	Name of the console command.

SQL Sequence Table

The SQL_X_SEQUENCE (SQE) table provides a way of generating a unique ascending sequence number. The SQE table consists of one row which when queried includes a column with this ascending sequence number.

SQL_X_SEQUENCE (SQE)

The SQL_X_SEQUENCE table consists of one column with the following characteristics:

Column Name	SQL Data Type	Nullable	Description
SEQUENCE_NUMBER	DECIMAL(21,0)	No	A unique ascending sequence number. The number is computed using the current system clock so it does not ascend in increments of one.

SQL Status Tables

Following is a list of the SQL status tables.

Table Name and Location
SQL_MISC_STATS (SQM) (see page 146)
SQL_PROC_DETAIL (SQZ) (see page 147)
SQL_SQLCODES (SQQ) (see page 147)
SQL_STATUS (SQS)
SQL_STATUS_CURRENT (SQC) (see page 149)
SQL_STATUS_PLAN (SQP) (see page 157)
SQL_STATUS_SEGMENT (SQG) (see page 158)
SQL_STATUS_FRONT (SQF) (see page 159)
SQL_STATUS_URT (SQU) (see page 160)

The SQL virtual tables are now in the Dynamic System Tables database. To save the rows in these tables, you can use the searched form of the INSERT statement to insert them into a table of your own. For example, to obtain a profile of SQL activity during the day, you could define a table with the same columns as the SQL_STATUS table and execute the following statement every 30 minutes:

```
INSERT INTO MY_AUTHID.SQL_STATUS
SELECT *
FROM SYSADM.SQL_STATUS;
```

You can access the SQL status tables as you would any other SQL-accessible table. For example, you can use DDOL and DQRY for on demand queries and develop your own customized monitoring tools for ongoing analysis.

Here are examples of other possible uses of the SQL status tables.

- If you have a job running that builds a large temporary table, you can find out how much room is left in the temporary table (TTM) area by doing a SELECT such as the following:

  ```
  SELECT TTM_BLKS_ALLOC - TTM_BLKS_IN_USE
  FROM SYSADM.SQL_STATUS;
  ```

- If an SQL mode CA Dataquery query is running longer than you think it should and you believe production transactions are running longer, you can do a SELECT such as the following to find out what the query is doing and whether it is about to finish or should be canceled:

  ```
  SELECT *
  FROM SYSADM.SQL_STATUS_CURRENT
  WHERE ACCESSOR_ID = :USER_SIGNON;
  ```

- If you think an SQL precompile is hung, you can find out if someone is currently executing a plan by doing a SELECT such as the following:

  ```
  SELECT *
  FROM SYSADM.SQL_STATUS_PLAN
  WHERE PLAN_AUTHID = :AUTHID AND PLAN_NAME = :PLAN;
  ```

- If you have tried to use the CA Datacom/DB Utility (DBUTLTY) to EXTRACT a table and received a return code 46, you can find out if there is a SQL-generated URT open for the table by doing a SELECT such as the following:

  ```
  SELECT *
  FROM SYSADM.SQL_STATUS_URT
  WHERE DBID = :BASE AND DB_TBL_NAME = :TBL;
  ```

SQL_MISC_STATS (SQM)

This table is provided to allow the monitoring of statistical information on selected SQL functionality. These statistics cover the execution of SQL procedure statements within the MUF.

Column Name	SQL Data Type	Nullable	Description
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
PROC_CACHE_REUSE	INTEGER	No	Count of times a procedure definition had to be purged from the LRU (Least Recently Used) procedure definition cache to make room for a newly requested definition. Enlarge the cache size specified in the PROCEDURE Multi-User startup option until this number is as small as possible. If the number is zero, either procedures are not in use or every procedure definition used so far fits into the cache.
PROC_FAILURES	INTEGER	No	Count of times a procedure failed due to some type of abend in the procedure.
PROC_FROM_CALL	INTEGER	No	Count of times a procedure was executed because of a CALL PROCEDURE SQL statement.
PROC_FROM_TRIGGER	INTEGER	No	Count of times a procedure was executed because of a triggered event.
PROC_NESTS	INTEGER	No	Count of times a procedure was executed when it was caused by another procedure SQL statement.
PROC_QUEUED	INTEGER	No	Count of times a procedure needing to execute was placed on a queue because no procedure TCB was available. Used to tune the number of procedure TCBs.
PROC_SQL_STMTS	INTEGER	No	Count of SQL statements executed by a procedure.

SQL_PROC_DETAIL (SQZ)

This table lists the procedure activity since MUF startup.

Column Name	SQL Data Type	Nullable	Description
EXEC_SECONDS	INTEGER	No	The number of seconds of execution, rounded up to the next second.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
PROC_NAME	CHAR(8)	No	Shows the procedure name.
PROC_STARTS	INTEGER	No	Shows the procedure starts.
PROC_TCB_FAILURES	INTEGER	No	Shows the procedure failures.
SQL_STATEMENTS	INTEGER	No	Shows the SQL statements.

SQL_SQLCODES (SQQ)

This table allows you to determine which SQL_SQLCODE has been encountered in this MUF execution (and how often). It has a key of the SQLCODE.

Column Name	SQL Data Type	Nullable	Description
CODE_COUNT	INTEGER	No	Count of occurrences of this SQLCODE.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
SQL_CODE	INTEGER	No	Specific SQLCODE returned from an SQL statement.

SQL_STATUS (SQS)

This one-row table describes the current activity and resources (for example, storage, TTM space, and so forth) used by the SQL engine in the MUF.

Column Name	SQL Data Type	Nullable	Description
CNS_POOL_SIZE	INTEGER	No	Shows the memory used for constraint enforcement.

Column Name	SQL Data Type	Nullable	Description
CURSORS_OPEN	INTEGER	No	Shows the number of open cursors.
LUW_POOL_SIZE	INTEGER	No	Shows the memory used by current requests and open cursors.
LUWS	INTEGER	No	Shows the number of open logical units of work, that is, units of recovery.
MEM_OTHER	INTEGER	No	Lists other dynamically allocated memory.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
PLAN_POOL_SIZE	INTEGER	No	Shows the memory used for a cache copy of statements executed in open plans.
PLANS_ACTIVE	SMALLINT	No	Shows the number of plans currently executing a statement in the MUF (plans with open LUWs) or with at least one open cursor.
PLANS_OPEN	SMALLINT	No	Lists the number of open plans.
REQ_STAMP	TIMESTAMP	No	Shows when you made your request.
STMTS_CACHE	INTEGER	No	Lists the number of statements that have been executed in open plans.
TASKS_ACTIVE	SMALLINT	No	Lists the number of tasks with requests active in the MUF.
TASKS_ATTACHED	SMALLINT	No	Shows the number of tasks currently allocated (less than or equal to the maximum number of tasks specified at MUF start-up).
TTM_BLKS_ALLOC	INTEGER	No	Shows the number of 4K-blocks allocated in the temporary table area.
TTM_BLKS_IN_USE	INTEGER	No	Lists the number of temporary table area 4K-blocks currently in use.
TTM_BLKS_MAX_USE	INTEGER	No	Lists the highest number of 4K-blocks used.

Column Name	SQL Data Type	Nullable	Description
URTS	SMALLINT	No	Lists the number of open User Requirements Tables.

SQL_STATUS_CURRENT (SQC)

This table describes the current status of all active SQL transactions.

Note: For information about how to use the SQC table to cancel SQL requests, see Using SQC Table to Cancel SQL Requests (see page 14).

Column Name	SQL Data Type	Nullable	Description
ACCESSOR_ID	CHAR(18)	No	Shows the accessor identifier.
CMD_TYPE	CHAR(12)	No	Shows the type of request.
DB_TBL_NAME	CHAR(3)	No	Shows the CA Datacom/DB table name. For temporary tables, this is TTM.
DBID	SMALLINT	No	Lists the database ID. For temporary tables, this is the temporary table DBID.
EXEC_CNT	INTEGER	No	Lists the number of times a process has been executed. Zero indicates a process has not started. One indicates a process has started and may have been completed. More than one indicates multiple rows joined to an inner table of a nested loop join or multiple executions of a correlated subquery. See DML Execution (Query Level).
JOB_NAME	CHAR(8)	No	Shows the job name.
LUW_BEG_REQ_NBR	INTEGER	No	Shows the request number of the first request for the LUW. Only cursors can have a prior request number. All other statements are executed in a single MUF request.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
PLAN_AUTH	CHAR(18)	No	Lists the authorization ID of plan.
PLAN_NAME	CHAR(18)	No	Shows the plan name.

Column Name	SQL Data Type	Nullable	Description
PROC_STEP	SMALLINT	No	Lists the process step number within query block. Steps are numbered by 100 per table. For example, a query that reads a single table and uses a sort for ORDER BY might have process steps 101, 102, and 103 for the base table and 201 for the sort.
PROC_TYPE	CHAR(32)	No	Lists the process type. See Process Types (PROC_TYPE) Description (see page 152).
QRY_LVL	SMALLINT	No	Lists the query level. Zero indicates subselect. Levels 1 and greater indicate subqueries.
QRY_NBR	SMALLINT	No	Shows the number of the subselect or subquery at the specified level.
ROWS_OUT	INTEGER	No	Shows the number of rows that were output from the process step. By subtracting the ROWS_OUT from the previous step, you can determine the number of rows filtered by the process. For example, if the ROWS_OUT for process READ INDEX is 1,000 and the next process RESTRICT INDEX is 100, low-order predicates rejected 900 rows. See Process Types (PROC_TYPE) Description (see page 152).
RUN_UNIT	INTEGER	No	Shows the run unit number (a sequential number assigned to each job using the MUF).
STAMP_CUR	TIMESTAMP	No	Shows the timestamp of the cursor to read this table.
STAMP_LAST_REQ	TIMESTAMP	No	Shows the timestamp of your request to read this table.
STAMP_LUW_BEG	TIMESTAMP	No	Shows the timestamp when current or OPEN request began executing in the MUF.

Column Name	SQL Data Type	Nullable	Description
STMT_ID	INTEGER	No	Shows the statement ID. In CA Datacom mode, statement IDs are assigned in the order they physically appear in the host program in increments of 16, that is, the first statement is 16, the second 32, and so on. For dynamic statements, the first statement is 24, the second 40, and so on. All statements referencing a cursor use the statement ID of the DECLARE statement, except the last three bits specify the cursor command: OPEN=1, FETCH=2, UPDATE=3, DELETE=4, CLOSE=5, OPEN(with hold)=6. For example, 18 (16 + 2) is a FETCH for the first statement. In DB2 mode, statement IDs are assigned by the host language precompiler. For COBOL and PL/I, the program line number is used.
STMT_TYPE	CHAR(12)	No	Lists the statement type.
TBL_NBR	SMALLINT	No	Shows the position of the table in the FROM clause. The first table is TBL_NBR 0. Temporary tables are numbered following the tables in the FROM clause.
TERM_ID	CHAR(4)	No	Lists the terminal ID if online. Shows blanks if batch.
TSK_NBR	SMALLINT	No	Shows the RWTSA in which a process is currently executing. Zero indicates an open cursor which is not currently executing in the MUF.

Process Types (PROC_TYPE) Description

The processes that can generate a row are:

- Binding
- DDL execution
- Record-at-a-time constraint enforcement
- DML execution
- LUW summary

Binding

In the SQL_STATUS_CURRENT table, PROC_TYPE shows the following for bind processes:

```
BIND IN PROGRESS
BIND: GET HOST VARIABLE INFO (QHOST)
AUTO VIEW REBIND
CONSTRAINT REBIND IN PROGRESS
```

In the SQL_STATUS_CURRENT table, ROWS_OUT is zero for the preceding processes.

DDL Execution

Tasks executing DDL statements generate a single row. In the SQL_STATUS_CURRENT table, STMT_TYPE indicates the type of statement being executed.

Record-at-a-Time Constraint Enforcement

A task enforcing constraints for record-at-a-time ADDIT, UPDAT, or DELET commands generates a row with a PROC_TYPE (in the SQL_STATUS_CURRENT table) of ENFORCE CONSTRAINTS, while CMD_TYPE in the same table indicates which command is being constrained.

DML Execution (Query Level)

The following rows may appear for cursor open and fetch, and searched insert, update and delete statements. The processes are listed in order of execution. The PROC_TYPE column (in the SQL_STATUS_CURRENT table) indicates the kind of activity, while ROWS_OUT in the same table indicates the number of rows output by the process.

If the process has not taken place yet, ROWS_OUT is zero. For some processes, ROWS_OUT is set while the process is taking place, and in other cases ROWS_OUT is only set at the end of the process.

Some processes can be executed multiple times. EXEC_CNT indicates how many times the process has taken place, including processes *in-flight* at the time of your request. In these cases, ROWS_OUT is the total for all executions.

- Index Access

  ROWS_OUT for process type READ INDEX is the number of index entries read.

- Index Scanning

  Index scanning includes the following process types:

  INDEX MERGE

  - Rows from multiple indexes qualify
  - Temporary index built to eliminate duplicate rows found in multiple indexes

  INDEX INTERSECTION

  - Rows qualify only if found in multiple indexes
  - Only used if indexes restricted to single value

  INDEX RESTRICTION

  - Rows qualify only if low-order predicates are TRUE
  - Low-order predicates reference columns of scan index, but cannot restrict index scan range because they:
    - Reference other than the first column of index, or
    - Preceding columns are not restricted to single value

  If these processes have not or are not currently taking place, no row is produced.

  Although all of these processes can occur in the same query, only one row is generated where ROWS_OUT is the number of index entries that have been qualified by all processes. If PROC_TYPE is INDEX MERGE, index intersection and restriction may also have taken place. If PROC_TYPE is INDEX INTERSECTION, index restriction may also have taken place.

  You can determine the number of rows rejected by index scanning by subtracting ROWS_OUT from the index access ROWS_OUT.

  If a large percentage of index entries is rejected by this process, you can improve performance when there is a single less than or greater than predicate by rearranging the columns of the index, (or adding a new index), such that the referenced column is either the first column or preceded by columns restricted to a single value.

Following is an example:

```
SELECT *
FROM T1
WHERE COL1 >= :HOST1 AND COL2 = :HOST2

Original Index: (COL1, COL2)
Better Index  : (COL2, COL1)
```

- Read Data

  The number of rows read from the data area is ROWS_OUT for a PROC_TYPE of READ DATA. This is the same value for index scanning, if applicable, or index access, since no rows can be rejected by this process.

- Restrict Data

  This row is generated if there are predicates evaluated by Compound Boolean Selection that reference columns not in the scan index. The number of rows rejected by this process is the difference between ROWS_OUT for this and the previous process step.

  The following predicates are not evaluated by Compound Boolean Selection, so the number of rows rejected is not reflected in this process step:

  - LIKE
  - IN list
  - Operands are expressions or scalar functions (except concatenated contiguous NOT NULL character columns)
  - When there are multiple tables referenced under an OR
  - Correlated or quantified subquery

  If a large percentage of rows is rejected in this step, you can improve performance by including the referenced columns in the scan index. This eliminates the cost of accessing rows in the data area that do not qualify. However, consider the higher cost of the index, especially if the extra columns included are updated frequently.

- Sort for Merge

  If a sort is performed to join tables using the sort-merge technique, a row with the process type SORTED FOR MERGE or SORTING FOR MERGE is generated. ROWS_OUT is the number of rows sorted.

- Reading sort-merge temporary table

  If a table is sorted for a sort-merge join, a row with process step READ SORTED TEMP TABLE is generated. ROWS_OUT is the number of rows read. This value can be greater than the number of rows in the table for the inner table when there are multiple rows with the same join value in the outer table.

- Quantified Subquery Temporary Table

  A quantified subquery can have multiple rows in its result table. These rows are stored in a temporary table. A row with process step SUBQUERY TEMP TABLE is generated when the subquery is executed. EXEC_CNT can be greater than one if the subquery is correlated, in which case ROWS_OUT is the total rows for all executions of the subquery.

- Non-quantified Subquery

  A row is generated with the process step NON-QUANTIFIED SUBQUERY for non-quantified subqueries. Since a non-quantified subquery can have at most one row, ROWS_OUT is either zero or one.

  For correlated, non-quantified subqueries, EXEC_CNT indicates the number of times the subquery has been evaluated, and ROWS_OUT is the sum of all executions.

- Group By

  If a sort is required for GROUP BY, a row is generated with process type SORT FOR GROUP BY.

  If an index is used to process the GROUP BY, a row is generated with process type INDEXED GROUP BY.

  In either case, ROWS_OUT is the number of rows in the grouped table.

- HAVING

  If HAVING is specified, a row is generated with process type HAVING, where ROWS_OUT is the number of qualified grouped rows.

- DISTINCT, UNION or ORDER BY

  DISTINCT and UNION, and ORDER BY not using an index, cause a sort. A row is generated with process type SELECT DISTINCT, SORT SUBSELECT OF UNION, or ORDER BY is generated in these cases. ROWS_OUT indicates the number of rows in the sorted result temporary table.

  Although there are these three reasons for the sort, there is only one sort per subselect and only one row generated. When process type is ORDER BY, DISTINCT or UNION may have been specified, also. When process type is SORT SUBSELECT OF UNION, DISTINCT may also have been specified.

- UNION MERGE

  When there are multiple subselects, a row is generated with process type UNION MERGE. ROWS_OUT is the total number of rows output from the union of all subselects.

DML Execution (Statement Level)

ROWS_OUT for the following process types are for the final result at the statement level. Query-level columns are not significant.

- FETCH ROWS—rows returned
- UPDATE CURSOR—rows updated
- DELETE CURSOR—rows deleted
- INSERT ROWS—rows inserted
- DELETE ROWS—rows deleted
- UPDATE ROWS—rows updated

LUW Summary (Diagnostic Activity)

LUW summary rows have process steps 9900 and higher. Plan and statement level columns are not significant.

Writing SYSADM.SYSMSG and Statistics and Diagnostics Area (PXX) diagnostic information can cause a job to run slowly when triggered by:

- Plan option MSG (SYSMSG and Compound Boolean Selection Diagnostic Report to Statistics and Diagnostics Area (PXX))
- DBUTLTY COMM OPTION=ALTER,TRACE/DUMPS= PXX tracing requests
- PXX dumps for certain return codes

Diagnostic activity is shown in process steps and types:

- 9998—SYSMSG ROWS INSERTED/DELETED
- 9997—SQL AND CBS TRACE LINES

Note: ROWS_OUT for the CBS Diagnostic Report is an estimate.

PXX return code dumps are not included.

LUW Summary (Non-Diagnostic Activity)

The total requests to MUF for the LUW is indicated in process step 9999, process type REQUESTS TO SQL SUBSYSTEM in column ROWS_OUT.

SQL_STATUS_PLAN (SQP)

This table provides a list of all open plans.

Column Name	SQL Data Type	Nullable	Description
CBSIO	INTEGER	No	Shows the plan option CBSIO.
CLOSE_OPT	CHAR(1)	No	Lists the plan close option.
DATE_LAST_USED	DATE	No	Shows the last date a statement of the plan began executing in the MUF.
DATE_OPENED	DATE	No	Shows the date the plan was opened.
ISO_LVL	CHAR(1)	No	Shows the isolation level.
MSG_OPT	CHAR(2)	No	Lists the plan option MSG.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
OPT_MODE	CHAR(1)	No	Shows the plan option OPT.
PLAN_AUTHID	CHAR(18)	No	Shows the plan authorization ID.
PLAN_NAME	CHAR(18)	No	Lists the plan name.
PRIORITY	SMALLINT	No	Shows the plan priority.
REQ_STAMP	TIMESTAMP	No	Lists the timestamp of your request to read this table.
TIME_LAST_USED	TIME	No	Lists the last time a statement of the plan began executing in the MUF.
TIME_OPENED	TIME	No	Shows the time the plan was opened.
USERS_BATCH	SMALLINT	No	Shows the number of batch jobs using the plan.

Column Name	SQL Data Type	Nullable	Description
USERS_ONLINE	SMALLINT	No	Shows the number of online active LUWs using the plan.
WAIT_SECONDS	INTEGER	No	Lists the plan option wait minutes times 60 plus the wait seconds ((wait minutes X 60) + wait seconds).

SQL_STATUS_SEGMENT (SQG)

This table lets you view the SQL statement source for executing SQL statements. It provides the source in a series of rows containing 80-byte segments. The SEGMENT_SEQUENCE column contains a number indicating the segment of the statement source represented by the row, the first row having a SEGMENT_SEQUENCE of 1.

Column Name	SQL Data Type	Nullable	Description
ACCESSOR_ID	CHAR(18)	No	The security accessor ID of the person executing the plan.
BIND_METHOD	CHAR(11)	No	Indicates PERMANENT if the statement was statically prepared (and will be permanently stored in the DBMS), otherwise indicates TEMPORARY.
JOB_NAME	CHAR(8)	No	The job name of the application.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
PLAN_AUTH	CHAR(18)	No	The authorization ID of the plan.
PLAN_NAME	CHAR(18)	No	Shows the plan name.
RUN_UNIT	INTEGER	No	Shows the run unit number (a sequential number assigned to each job using the MUF).
SOURCE_SEGMENT	CHAR(80)	Yes	Contains the text of the SQL statement, or NULL if no text is currently available.

Column Name	SQL Data Type	Nullable	Description
SEGMENT_SEQUENCE	SMALLINT	No	Identifies the segment of source represented by this row. The first segment is segment 1.
STMT_ID	INTEGER	No	This identifier distinguishes the statement from others in the plan.
TERM_ID	CHAR(4)	No	Identifies the terminal from which a CICS user is running.
TSK_NBR	SMALLINT	No	The number of the task in the MUF that is executing this statement.

SQL_STATUS_FRONT (SQF)

This table lets you view the SQL statement source for executing SQL statements. It provides up to 3900 bytes of the statement source in one large VARCHAR column.

Column Name	SQL Data Type	Nullable	Description
ACCESSOR_ID	CHAR(18)	No	The security accessor ID of the person executing the plan.
BIND_METHOD	CHAR(11)	No	Indicates PERMANENT if the statement was statically prepared (and will be permanently stored in the DBMS), otherwise indicates TEMPORARY.
JOB_NAME	CHAR(8)	No	The job name of the application.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.
PLAN_AUTH	CHAR(18)	No	The authorization ID of the plan.
PLAN_NAME	CHAR(18)	No	Shows the plan name.
RUN_UNIT	INTEGER	No	Shows the run unit number (a sequential number assigned to each job using the MUF).

Column Name	SQL Data Type	Nullable	Description
SOURCE_FRONT	VARCHAR(3900)	Yes	Contains the text of the SQL statement, or NULL if no text is currently available.
STMT_ID	INTEGER	No	This identifier distinguishes the statement from others in the plan.
TERM_ID	CHAR(4)	No	Identifies the terminal from which a CICS user is running.
TSK_NBR	SMALLINT	No	The number of the task in the MUF that is executing this statement.

SQL_STATUS_URT (SQU)

This table lists all open User Requirements Tables. You can use DELETE to manually close User Requirements Tables.

Column Name	SQL Data Type	Nullable	Description
CLS_OPT	CHAR(1)	No	Shows when the URT will be closed. A T indicates closure at the end of the last logical unit of work to use the URT (that is, when the USERS column goes to 0). An R means closure at the end of the run-unit. This column is initially set to the close-option of the plan that opens the URT. On subsequent uses, a T is *upgraded* to R if the close-option of the plan instigating the usage is R. Once updated to R, the value never reverts to T. Rows containing values other than T and R are not part of this table.
DB_TBL_NAME	CHAR(3)	No	Shows the CA Datacom/DB table name.
DBID	SMALLINT	No	Lists the CA Datacom/DB database ID.
JOB_NAME	CHAR(8)	No	Shows the job name.
MUF_NAME	CHAR(8)	No	Logical name for this MUF.

Column Name	SQL Data Type	Nullable	Description
REQ_STAMP	TIMESTAMP	No	Lists the timestamp of your request to read this table.
RUN_UNIT	INTEGER	No	Lists the run unit number.
URT_NBR	SMALLINT	No	Shows the URT number (assigned by SQL).
USERS	SMALLINT	No	Lists the number of LUWs using the URT.

SQL Cache Tables

All of the SQL Cache dynamic system tables have an authid of SYSADM.

All SQL Cache tables are prefixed with SQLSC, to represent SQL Source Cache.

Users can operate on the Cache by writing queries against the DSTs of the Cache.

Each column name is prefixed with the three-character table name abbreviation. All abbreviations consist of "SC" and a single character unique to the table.

Each table has a Primary Key consisting of a single, system-generated unique value. Use this column to join to related tables. The value of these Primary Keys can change each time the MUF is restarted.

Except where noted, these SQL Cache DSTs are read-only.

Following are descriptions of the Cache DSTs:

SQLSC_FACILITY (SCF)

The SQLSC_FACILITY (SCF) table consists of one row. The SCF table describes the overall state of the SQL Cache since the MUF was started. The SCF table has the following characteristics:

Note: When the SCF_QUERY_ALLOWED or SCF_MEMORY_ALLOWED column is changed, the SCF_MEMORY_SIZED_TS column is set the next time the Cache is searched, and the SCF_SEARCH_CNT and SCF_MATCH_CNT columns are reset so that the hit ratio can then be computed using the new limits.

Column Name	SQL Data Type	Nullable	Description
SCF_ID	INTEGER	No	Primary Key (always 1)

Column Name	SQL Data Type	Nullable	Description
SCF_MUF_NAME	CHAR(8)	No	Name of the MUF
SCF_CURRENT_TS	TIMESTAMP	No	Current time
SCF_MEMORY_SIZED_TS	TIMESTAMP	No	When the Cache size was last set
SCF_QUERY_ALLOWED	INTEGER	No	Maximum queries the Cache can hold (default is 1000)
SCF_MEMORY_ALLOWED	INTEGER	No	Maximum memory the Cache can use (default is 10M)
SCF_MEMORY_CURRENT	INTEGER	No	Current memory used
SCF_QUERY_CNT	INTEGER	No	Current number of queries in the Cache (see the note aove the table)
SCF_SEARCH_CNT	INTEGER	No	Number of times the Cache was searched--SCF_SEARCH_CNT and SCF_MATCH_CNT are set to zero when SCF_SEARCH_CNT overflows the maximum fullword value (see the note aove the table)
SCF_MATCH_CNT	INTEGER	No	Number of times a match was found

Note: Two additional columns in the SCF table are planned for future releases, SCF_OPTIMIZED_QUERIES and SCF_TERMINATED_CNT. When implemented in a future release, it is planned that optimized statements would never be removed from the Cache.

Cache Size

The size of the Cache can be limited by both the memory used (SCF_MEMORY_ALLOWED) and the number of statements in the Cache (SCF_QUERY_ALLOWED). Because both limits are in effect, the statement count limit can cause the memory limit to never be reached. Likewise, the memory limit can cause the statement count limit to never be reached.

When a new statement is inserted into the Cache, the least-recently-used (LRU) statement is removed from the Cache. Currently-executing statements cannot be removed from the Cache, because statistics are recorded in the cached copy of the statement at end of execution. SCF_QUERY_CNT and SCF_MEMORY_CURRENT can therefore be larger than the respective limits.

The size of the Cache can be tuned by using a “hit ratio” of SCF_MATCH_CNT divided by SCF_SEARCH_CNT. When the Cache size is too small, this hit ratio decreases.

SQLSC_PLAN (SCP)

The SQLSC_PLAN (SCP) table contains a row for each combination of plan options that can affect a query's generated executable object from queries in the Cache. For a query to match in the Cache, these plan options must be identical. Other plan options, however, because they do not affect the generated plan use the current plan, and therefore are not required to match.

Column Name	SQL Data Type	Nullable	Description
SCP_ID	INTEGER	No	Unique ID
Plan Options follow:			For descriptions, see the CA Datacom/DB SQL User Guide
CHK_PLAN	CHAR(1)	No	Plan execute privilege
CHK_WHO_BINDER	CHAR(1)	No	Access rights of binder or executor
CHK_WHEN_BIND	CHAR(1)	No	Table-level privileges checked at bind or runtime
PLAN_HAS_DYN_STMT	CHAR(1)	No	Dynamic statement in plan
CURSOR_BEHAVIOR	INTEGER	No	Commit and rollback behaviour
SQLMODE	CHAR(8)	No	Mode to process the program
ISOLATION	CHAR(1)	No	Isolation level
OPTMODE	CHAR(1)	No	Join optimization mode
TIMEFORM	CHAR(3)	No	Time output format
DATEFORM	CHAR(3)	No	Date output format
DECPOINT	CHAR(1)	No	Period or Comma to be decimal point indicator
STRDELIM	CHAR(1)	No	String delimiter apostrophe or quotation mark

SQLSC_ENTRY (SCE)

The SQLSC_ENTRY (SCE) table consists of one row for each query in the Cache.

Column Name	SQL Data Type	Nullable	Description
SCE_ID	INTEGER	No	Unique ID
SCE_HASH-ID	INTEGER	No	internal use

Column Name	SQL Data Type	Nullable	Description
SCE_PLAN_ID	INTEGER	No	ID of the query's SQLSC_PLAN
SCE_QUERY_NO	INTEGER	No	Identifies the query (implementation pending)
SCE_STMT_TYPE	CHAR(12)	No	query/insert/update/delete

SQLSC_VERSION (SCV)

The SQLSC_VERSION (SCV) table consists of one row for each version of a query in the Cache. A query has multiple versions only when an optimization or restriction has been specified.

Statistics are not saved across MUF executions, except that statistics for the original version are saved at the time the first optimized version is created.

In the SCV table that follows, “RSU” is an abbreviation for Resource Units. This is the same unit of measure used by the CBSIO plan option.

Note: In a future release, it is planned to include in the SCV table a group of columns that provide Elapsed Time Statistics.

Column Name	SQL Data Type	Nullable	Description
SCV_ID	CHAR(16)	No	Unique ID
SCV_SCE_ID	INTEGER	No	References SQLSC_ENTRY
SCV_STMT_ID	INTEGER	No	References an optimized statement, else zero
SCV_VERSION	SMALLINT	No	Version 0 is original query Version 1+ are optimized versions
SCV_ADDED_TS	TIMESTAMP	No	When query first added to Cache
SCV_MATCHES	INTEGER	No	Total matches
SCV_MATCH_LAST_TS	TIMESTAMP	No	When last matched
Elapsed Time Statistics			
SCV_ELAPSED_LIMIT_TS	TIMESTAMP	No	When limit was set last
SCV_ELAPSED_LIMIT	INTEGER	No	maximum seconds query can execute
SCV_TIMED_OUT_CNT	INTEGER	No	Number times query has been timed-out
SCV_ELAPSED_LAST	INTEGER	No	Last elapsed time in MUF in seconds

Column Name	SQL Data Type	Nullable	Description
SCV_ELAPSED_MAX	INTEGER	No	Maximum elapsed time in MUF in seconds
SCV_ELAPSED_MAX_TS	TIMESTAMP	No	When maximum elapsed time occurred
SCV_ELAPSED_MIN_TS	TIMESTAMP	No	When minimum elapsed time occurred
SCV_ELAPSED_MIN	INTEGER	No	Minimum elapsed time in MUF in seconds
SCV_ELAPSED_TOTAL	INTEGER	No	Total elapsed time
Resource Unit (RSU) Statistics			
SCV_RSU_LIMIT	INTEGER	No	CBSIO limit
SCV_RSU_LIMIT_CNT	INTEGER	No	Times limit reached
SCV_RSU_LAST	INTEGER	No	Last RSU in MUF in seconds
SCV_RSU_MAX	INTEGER	No	Maximum RSU in MUF in seconds
SCV_RSU_MAX_TS	TIMESTAMP	No	When maximum RSU occurred
SCV_RSU_MIN_TS	TIMESTAMP	No	When minimum RSU occurred
SCV_RSU_MIN	INTEGER	No	Minimum RSU in MUF in seconds
SCV_RSU_TOTAL	INTEGER	No	Total RSU
Row Statistics			
SCV_ROWS_ACCEPTED_AVG	INTEGER	No	Average rows accepted
SCV_ROWS_SORT_AVG	INETGER	No	Average rows sorted for GROUP/ORDER/UNION
SCV_ROWS_INDEX_AVG	INTEGER	No	Average total index entries scanned
SCV_ROWS_DATA_AVG	INTEGER	No	Average total data area rows scanned
SCV_ROWS_MERGE_AVG	INTEGER	No	Average total rows sorted for a merge join
SCV_ROWS_SUBQRY_AVG	INTEGER	No	Average total rows searched in subqueries

SQLSC_METRICS (SCM)

The SQLSC_METRICS (SCM) table consists of one row for each processing step for each query. Metrics are kept for the original query, and each optimized query.

Processing steps include the following processes (these steps appear for each SELECT block, except UNION and ORDER BY, which can only be specified once per query):

- Reading a base table
- Grouping
- Union
- Sort (for Join/Group/Union/Distinct)
- Build & read subquery temp table

Column Name	SQL Data Type	Nullable	Description
SCM_SCV_ID	CHAR(16)	No	References SQLSC_VERSION
SCM_TABLE_NAME	CHAR(48)	No	authId.tableName correlationName
SCM_STEP	INTEGER	No	Processing step number, a number that reflects the order in which processing occurs
SCM_PROCESS	CHAR(24)	No	Text description of step
SCM_SUBSEL_NO	SMALLINT	No	Subselect number
SCM_SUBQRY_NO	SMALLINT	No	Subquery number within a subselect
SCM_TBL_NO	SMALLINT	No	position in table expression tree relative to zero (see note following table)
SCM_DBID	SMALLINT	No	Database ID of table (see note following table)
SCM_TBL_NM	CHAR(3)	No	3-character table name (see note following table)
SCM_KEY	CHAR(5)	No	5-character key name (see note following table)
SCM_SETS_AVG	INTEGER	No	Times executed
SCM_SCANKY_AVG	INTEGER	No	Number of Index entries scanned (see note following table)
SCM_SCANDT_AVG	INTEGER	No	Number of rows accessed in the data area (see note following table)

Column Name	SQL Data Type	Nullable	Description
SCM_ROWS_AVG	INTEGER	No	Rows accepted (after filtering with predicates not evaluated by Compound Boolean Selection (CBS))

Note: These fields are blank or zero except for table-level processes 1 – 3.

Schema Information Tables (SIT)

The Schema Information Tables (SIT) are located in the SIT area, base 15, and are associated with an authorization ID of SYSADM, for example, SYSADM.SYSCONSTRDEP, SYSADM.SYSCONSTROBJ, and so on. They contain CA Datacom/DB system information that *authorized* users can query. **The SIT information should be properly secured.**

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

The SIT tables described in this chapter are as follows:

Table Name and Location
SYSCONSTRDEP (CND) (see page 167)
SYSCONSTROBJ (CNO) (see page 168)
SYSCONSTRSRC (CNS) (see page 168)
SYSVIEWDEP (VWD) (see page 169)

SYSCONSTRDEP (CND)

SYSCONSTRDEP shows which constraints are related to which columns.

Column Name	Data Type	Description
COLNAME	CHAR(32)	Column Name
TBLNAME	CHAR(32)	Table Name
TBLCREATOR	CHAR(18)	Authorization ID of the Table
TBLTYPE	CHAR	'T' = Table, 'V' = View
CCREATOR	CHAR(18)	Authorization ID of the Constraint
CNAME	CHAR(32)	Constraint Name

SYSCONSTROBJ (CNO)

SYSCONSTROBJ contains the compiled form of constraints, for use by CA Datacom/DB.

Column Name	Data Type
CREATOR	CHAR(18)
CNAME	CHAR(32)
TYPE	CHAR
TCREATOR	CHAR(18)
TNAME	CHAR(32)
DBTBLNAME	CHAR(3)
DBID	SMALLINT
STATUS	CHAR
REFSTAT	CHAR
SEQNO	SMALLINT
TOTLEN	INTEGER
OBJ	CHAR(2048)

SYSCONSTRSRC (CNS)

SYSCONSTRSRC contains the source for constraints.

Column Name	Data Type	Description
CREATOR	CHAR(18)	Authorization ID of the Constraint
CNAME	CHAR(32)	Constraint Name
TYPE	CHAR	'C' - column or table constraint, including view WITH CHECK OPTION constraints 'U' - UNIQUE Constraint 'P' - Primary Key 'F' - Foreign Key
DELRULE	CHAR	' ' - No delete rule is applicable 'R' - ON DELETE RESTRICT 'C' - ON DELETE CASCADE 'N' - ON DELETE SET NULL 'D' - ON DELETE SET DEFAULT

Column Name	Data Type	Description
SEQNO	SMALLINT	Sequence of this row within the series of rows for this constraint (starts at zero)
TOTLEN	SMALLINT	Total length of the source
TEXT	CHAR(254)	A segment of the source

SYSVIEWDEP (VWD)

SYSVIEWDEP has one row for each dependency between a table (or view) and a view.

Column Name	Data Type	Description
BNAME	CHAR(32)	The Table or View DNAME is dependent on.
BCREATOR	CHAR(18)	Authorization ID of BNAME
BTYPE	CHAR	BNAME is 'T' - table, 'V' - view
DNAME	CHAR(32)	Name of the View
DCREATOR	CHAR(18)	Authorization ID of DNAME

Chapter 6: CA Datacom Server Tables

SRV_CLNTINFO (SRC)

This table tracks the JDBC application connections with the Datacom server. If the setClientInfo method is used in the JDBC connection class, then it is there for the life of the MUF.

Column Name	SQL Data Type	Nullable	Description
ACTIVE_CONNECTION	CHAR(1)	No	Contains either a Y or N. Based on the current state of the connection. If the connection is either active or the connection failed to go through the proper statement close, the connection shows that it is currently active. Otherwise, the connection shows as N for the life of the MUF.

Column Name	SQL Data Type	Nullable	Description
APPLICATIONNAME	CHAR(32)	No	This value is set by the invocation of either a JAVA 'setClientInfo(properties)' or a 'setClientInfo(string,string'. For more information,
CLIENTACCTINFO	VARCHAR(200)		This column contains the following information for the invoking application: ■ DatacomJdbcDriver major version, minor version ■ Build number ■ Client host name ■ Server name ■ IP address
CLIENTHOSTNAME	CHAR(18)	No	The is the host name of the computer that the application is using for the connection. This can be passed in the setClientInfo API invocation.
CLIENTUSER	CHAR(16)	Yes	This is the name of the application that is using the connection. This can be passed in the setClientInfo API invocation.
MUF_NAME	CHAR(8)	Yes	The column is filled in on behalf of the setClientInfo at the time the row is created. This is the name of the MUF that this CA Datacom Server is communicating with.
PLAN_NAME	CHAR(32)	Yes	This column is filled in on behalf of the setClientInfo at the time the row is created. It contains the plan name that is assigned by the CA Datacom Server that the application is connected to.
SERVER_NAME	CHAR(32)	Yes	This column contains the name of the server that the application is connected to. It is created at the time the setClientInfo updates the system tables.