CA Datacom®/AD

Implementation Guide

Version 14.0

CA Technologies Product References

This document references the following CA Technologies products:

- CA Datacom®/DB
- CA Datacom® CICS Services
- CA Datacom® Datadictionary™
- CA Datacom® DB2 Transparency
- CA Datacom® DL1 Transparency
- CA Datacom® Fast Restore
- CA Datacom® Presspack
- CA Datacom® Server
- CA Datacom® SQL
- CA Datacom® STAR
- CA Datacom® TOTAL Transparency
- CA Datacom® VSAM Transparency
- CA Dataquery™ for CA Datacom® (CA Dataquery)
- CA Ideal™ for CA Datacom® (CA Ideal)
- CA Ideal™ Option for DB2
- CA IPC
- CA Workload Automation CA 7® Edition (CA WA CA 7 Edition)
- CA Workload Automation Restart Option for z/OS Schedulers (CA WA Restart Option for z/OS Schedulers), formerly CA 11™ Workload Automation Restart and Tracking
- CA APCDDS™ Automated Report Balancing
- CA Common Services for z/OS
- CA Disk™ Backup and Restore (CA Disk)
- CA Dynam®/D Disk Management for z/VSE (CA Dynam/D for z/VSE)
- CA JARS® Resource Accounting (CA JARS)
- CA Jobtrac™ Job Management (CA Jobtrac)
- CA NetMaster® File Transfer Management
- CA NetMaster® Network Management for TCP/IP
- CA PMA Chargeback™

- CA Scheduler® Job Management (CA Scheduler)
- CA Top Secret®

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: Introduction

Overview

This guide provides an introduction to the CA Datacom/AD environment, an overview of CA Datacom/AD post-installation tasks, and a description of regular CA Datacom/AD operation and tuning tasks related to the products that use CA Datacom/AD facilities.

CA Datacom/AD is a suite of CA Datacom products that have been packaged together to provide a common relational database management system (DBMS) used by other CA products.

This guide is intended for those who do one or more of the following tasks:

- Install CA Datacom/AD system software
- Support CA Datacom/AD system software
- Administer the operations of the CA Datacom/AD system
- Maintain data integrity
- Ensure data accessibility
- Maintain system performance and software code currency

For more information, see Products that Use CA Datacom/AD (see page 53).

Terminology

For a list of definitions of many of the terms used in this guide, see the Glossary. In particular, note that in this guide a *region* equates to an *address space* in z/OS environments. A z/OS address space is the area (assigned to a user) in which instructions can be executed and data stored.

Sample Report Headers

The report headers for the sample reports contained in this guide are shown here. Report headers have the following format:

```
Date: mm/dd/ccyy    ********************************************************************************         Page:     n
                    *                                CA Datacom/DB                                 *
Time: hh.mm.ss      *                                General Utility                               *      Version: nn.n
                    *                Copyright © 1990-2011 CA. All rights reserved.                *
Base:     dbid      ******************************************************************************** Directory:    name
```

Base:

The *dbid* is the DATACOM-ID (DBID) of the database (base) in use when the report was executed.

Note: Base does not appear in the header if it is not appropriate for the report that was generated or not known at the time the report is produced.

Date:

The date when the report was executed is shown in the format *mm/dd/ccyy*:

mm

month

dd

day

cc

century

yy

year

Directory:

The *name* is the internal name of the Directory (CXX), assigned with the INIT CXX function that was in use when the report was executed and if known at the time the report was produced.

Note: Directory does not appear in the header if it is not appropriate for the report that was generated.

Page:

The *n* is the page number of the report.

Time:

The time when the report was assembled is shown in the format *hh.mm.ss*:

hh

hour

mm

minutes

ss

seconds

Version:

The version of CA Datacom/DB being executed when the report was executed is shown in the format *nn.n,* for example, Version: 14.0.

Reading Syntax Diagrams

Syntax diagrams illustrate the format of statements and some basic language elements. Read syntax diagrams from left to right and top to bottom.

Syntax diagrams use the following terminology, symbols, and concepts:

- Keywords appear in uppercase letters, for example, COMMAND or PARM. Enter these words exactly as shown.
- Variables appear in italicized lowercase letters, for example, *variable*.
- Required keywords and variables appear on a main line.
- Optional keywords and variables appear below a main line.
- Default keywords and variables appear above a main line.
- Double arrowheads pointing to the right indicate the beginning of a statement.
- Double arrowheads pointing to each other indicate the end of a statement.
- Single arrowheads pointing to the right indicate a portion of a statement, or that the statement continues in another diagram.
- Punctuation marks or arithmetic symbols that are shown with a keyword or variable must be entered as part of the statement or command. Punctuation marks and arithmetic symbols can include the following:

,	comma	>	greater than symbol
.	period	<-	less than symbol
(	open parenthesis	=	equal sign

)	close parenthesis	¬	not sign
+	addition	-	subtraction
*	multiplication	/	division

Statement Without Parameters

The following is a diagram of a statement without parameters:

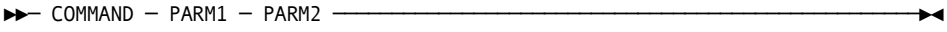
For this statement, you must write the following:

```
COMMAND
```

Statement with Required Parameters

Required parameters appear on the same horizontal line, the main path of the diagram, as the command or statement. Separate the parameters with one or more blanks.

The following is a diagram of a statement with required parameters:



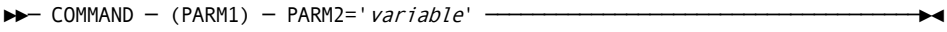
You must write the following:

```
COMMAND PARM1 PARM2
```

Delimiters Around Parameters

Delimiters, such as parentheses or quotation marks, around parameters or clauses must be included.

The following is a diagram of a statement with delimiters around parameters:



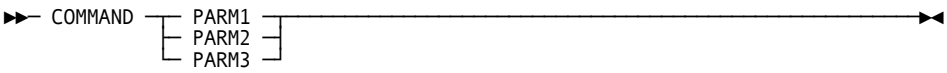
If the word *variable* is a valid entry, you must write the following:

```
COMMAND (PARM1) PARM2='variable'
```

Choice of Required Parameters

When you see a vertical list of parameters as shown in the following example, choose one of the parameters. This format indicates that one entry is required, and only one of the displayed parameters is allowed in the statement.

The following is a diagram of a statement with a choice of required parameters:



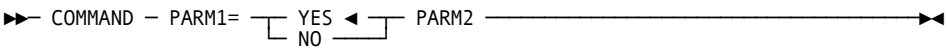
Choose one of the parameters from the vertical list, such as in the following examples:

```
COMMAND PARM1
COMMAND PARM2
COMMAND PARM3
```

Default Value for a Required Parameter

When a required parameter in a syntax diagram has a default value, the default value appears above the main line, and it indicates the value for the parameter if the command is not specified. If you specify the command, code the parameter and specify one of the displayed values.

The following is a diagram of a statement with a default value for a required parameter:



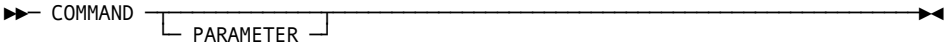
If you specify the command, write one of the following:

```
COMMAND PARM1=NO PARM2
COMMAND PARM1=YES PARM2
```

Optional Parameter

A single optional parameter appears below the horizontal line that marks the main path.

The following is a diagram of a statement with an optional parameter:



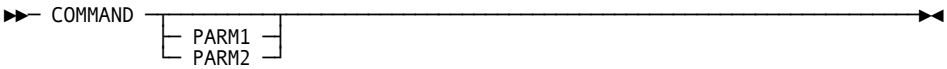
You can choose to use the optional parameter, as shown in the following examples:

```
COMMAND
COMMAND PARAMETER
```

Choice of Optional Parameters

If you have a choice of more than one optional parameter, the parameters appear in a vertical list below the main path.

The following is a diagram of a statement with a choice of optional parameters:



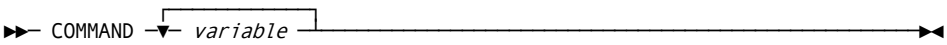
You can choose any of the parameters from the vertical list, or you can write the statement without an optional parameter, such as in the following examples:

```
COMMAND
COMMAND PARM1
COMMAND PARM2
```

Repeatable Variable Parameter

In some statements, you can specify a single parameter more than once. A repeat symbol indicates that you can specify multiple parameters.

The following is a diagram of a statement with a repeatable variable parameter:



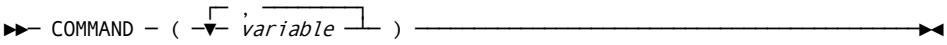
In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply, but it is also on the main path, which means that you are required to specify at least one entry. The repeat symbol indicates that you can specify a parameter more than once. Assume that you have three values named VALUEX, VALUEY, and VALUEZ for the variable. The following are some of the statements you might write:

```
COMMAND VALUEX
COMMAND VALUEX VALUEY
COMMAND VALUEX VALUEX VALUEZ
```

Separator with Repeatable Variable and Delimiter

If the repeat symbol contains punctuation such as a comma, you must separate multiple parameters with the punctuation. The following diagram includes the repeat symbol, a comma, and parentheses:

The following is a diagram of a statement with a separator with a repeatable variable and a delimiter:



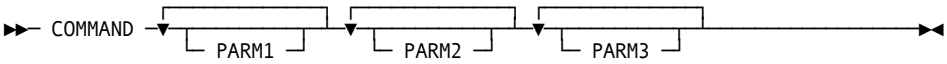
In the preceding diagram, the word *variable* is in lowercase italics, indicating that it is a value you supply. It is also on the main path, which means that you must specify at least one entry. The repeat symbol indicates that you can specify more than one variable and that you must separate the entries with commas. The parentheses indicate that the group of entries must be enclosed within parentheses. Assume that you have three values named VALUEA, VALUEB, and VALUEC for the variable.

The following are some of the statements you can write:

```
COMMAND (VALUEC)
COMMAND (VALUEB,VALUEC)
COMMAND (VALUEB,VALUEA)
COMMAND (VALUEA,VALUEB,VALUEC)
```

Optional Repeatable Parameters

The following diagram shows a list of parameters with the repeat symbol for optional repeatable parameters:



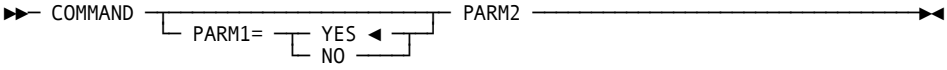
The following are some of the statements you can write:

```
COMMAND PARM1
COMMAND PARM1 PARM2 PARM3
COMMAND PARM1 PARM1 PARM3
```

Default Value for a Parameter

The placement of YES in the following diagram indicates that it is the default value for the parameter. If you do not include the parameter when you write the statement, the result is the same as if you had actually specified the parameter with the default value.

The following is a diagram of a statement with a default value for an optional parameter:

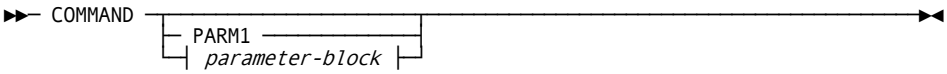


For this command, COMMAND PARM2 is the equivalent of COMMAND PARM1=YES PARM2.

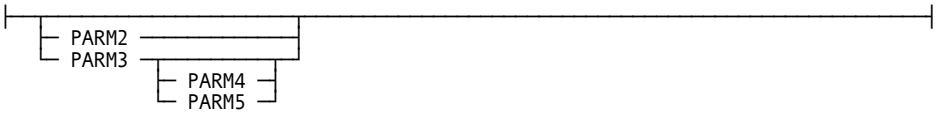
Variables Representing Several Parameters

In some syntax diagrams, a set of several parameters is represented by a single reference.

The following is a diagram of a statement with variables representing several parameters:



Expansion of parameter-block



The *parameter-block* can be displayed in a separate syntax diagram.

Choices you can make from this syntax diagram include, but are not limited to, the following:

```
COMMAND PARM1
COMMAND PARM3
COMMAND PARM3 PARM4
```

Note: Before you can specify PARM4 or PARM5 in this command, you must specify PARM3.

Listing Libraries for CA Datacom Products in JCL

Guidelines to assist you in preparing your JCL are provided in this manual. The sample code provided in this document is intended for use as a reference aid only and no warranty of any kind is made as to completeness or correctness for your specific installation.

Samples for JCL and programs are provided in the install library. You can copy and modify these samples for your specific requirements.

Any JOB statements should be coded to your site standards and specifications. All data set names and library names should be specified with the correct names for the installation at your site. In many examples, a REGION= or SIZE= parameter is displayed in an EXEC statement. The value displayed should be adequate in most instances, but you can adjust the value to your specific needs.

The libraries listed for searching must include the following in the order shown:

1. User customized libraries (CUSMAC, CUSLIB, CUSPROC) you could have defined for specially assembled and linked tables, such as DBMSTLST, DBSIDPR, DDSRTLM, DQSYSTBL, or User Requirements Tables (CUSLIB)
2. CA Datacom/AD base libraries (CAAXLOAD and CAAXMAC): CA Datacom/DB, CA Datacom Datadictionary, CA Dataquery, CA IPC, CA Datacom CICS Services, SQL, and DSV.
3. CA Common Services base libraries

CA Dataquery (optional DBID) users also need the following libraries and data sets for the following specific functions:

- The z/OS data set DQOUT or the z/VSE data set DQOUTD is used only if the DQBATCH execution uses the EXPORT function.
- Running deferred queries with separate JCL members in batch in z/OS requires, in addition to the SYSIN statement DEFER, the inclusion of a DD statement for the internal reader used by VPE. This DD statement should be:

```
//IRDR DD SYSOUT=(A,INTRDR)
```

Chapter 2: CA Datacom/AD Environment

Overview

CA Datacom/AD is the relational database management system used by many CA Technologies products. The CA Datacom/AD environment is a subset of the CA Datacom product line. CA Datacom/AD provides for the access and storage of the data of various using products (application products), providing significant enhancements to performance while ensuring data integrity and allowing concurrent update. The complete line of CA Datacom products is optionally available for use with this relational data.

The heart of the CA Datacom/AD is CA Datacom/DB, a full service relational database management system. It provides for the storage, maintenance, and retrieval of your data.

- Stores information
- Processes the stored information
- Logs each database update
- Monitors database system performance

The CA Datacom Datadictionary component is a central repository of consistent, descriptive information about data and applications.

- Stores data definitions and other information about the database
- Enables you to enforce standards
- Provides documentation of your system
- Generates a variety of reports about the data it stores

The logical structure of the databases required by the product using CA Datacom/AD is defined to CA Datacom Datadictionary and those definitions are stored in CA Datacom Datadictionary.

CA Datacom Datadictionary is an inventory control system for data and processes. It stores information about your data and the processes that act upon that data. It also affords security protection for data definitions. CA Datacom Datadictionary occupies two databases within CA Datacom/DB. You can run batch reports or you can check online displays (requires CICS and CA Datacom CICS Services) with CA Datacom Datadictionary to get the information you need to access and maintain the data stored in CA Datacom/DB databases.

A single environment consists of the following:

- A set of data sets containing unique information about and for the environment.
- One logical server/controller known as a Multi-User (Multiple User) Facility (MUF).

CA Datacom/DB Information Base Structure

This is a description of the structure of the CA Datacom/DB database. It is installed for and used by the product using CA Datacom/AD. For a list of the CA products that currently store their data in this RDBMS, see Products that Use CA Datacom/AD (see page 53).

Databases

The CA Datacom/DB information base can include up to 5000 databases. Each database is identified by a 5-digit numeric ID and consists of Index and data area data sets. The different types of databases include system, sample and CA product. Client databases are prohibited.

Areas

An area is a physical storage dataset that contains either index keys and pointers or data records. Each area can contain one or more logical tables.

There are two types of areas:

- Index (IXX) areas that contain index keys and data record pointers
- Data areas that contain the actual data records which are prefixed with control information

Index Areas

An index area provides the physical storage of key values with pointers to the corresponding records in the data area.

The Index Area (IXX) provides fast access to the data. Each database can contain one or more index areas (but normally has only one), which provides accessibility to all data areas within that database. The index allows any table in the database to be retrieved based on the contents of the table. CA Datacom/DB places an entry in the index for each key within each table in the database.

The Index Area is a compound relational index. All key types for a given database are contained in one compound index. The relational index system stores a given key value only once regardless of the number of tables in the databases that contain that key value. Proper selection of keys can optimize data retrieval.

Data Areas

The data areas are the physical storage medium used by databases to store tables. Every data area is a data set or file. The data areas are divided into blocks based on the block size defined to CA Datacom Datadictionary. The data block is the unit of physical transfer between the DASD device and CA Datacom/DB. As you populate your tables, these blocks are populated with multiple rows from one or more tables defined to that data area. Database maintenance, such as backups and loads, is performed on data areas.

Tables

All data is stored in logical files that are referred to as tables. Tables are two-dimensional, logically made up of rows (records) and columns (fields). Tables are easily extendable with the addition of both rows and columns.

Rows

Rows (records) are the smallest physical data structure that is comprised of one or more columns (fields) or groups of columns referred to as elements.

Columns

Columns (fields) are the smallest logical data structures in the information base. For SQL access, columns are the unit of transfer between the database and the application program.

Keys

Keys are structures used to optimize data access and order data retrieval. A key is composed of one or more columns. The columns can be non-contiguous and in any sequence. Each column in a key can be either ascending or descending in value. Any key can be defined as unique, that is, requiring that each row in the table have a unique value for the key.

All tables must have a Master Key and a Native Key defined. The Master Key functions as any other key but it can be defined as updatable or non-updatable. The Native Key dictates the physical sequence in which the data is stored. The Native Key can be the same as the Master Key.

Elements

For access with CA Datacom/DB commands, columns make up elements and elements are the unit of transfer. The columns that make up elements must be contiguous.

Conceptual Rows

Conceptual or virtual rows are comprised of SQL views and traditional Datacom dataviews.

SQL Views

Views are alternative virtual representations of the data from one or more tables. A view is a "derived table" which can include all or some of the columns contained in the table or tables on which it is defined. Views can also be defined on other views. Views are created by and accessed by SQL statements only.

Dataviews

Dataviews are similar in functionality to SQL Views. However, dataviews are special structures used by CA Ideal and Meta COBOL. Dataviews provide a logical view of the data. They can be composed of elements or keys. The elements (or keys) which make up a dataview are not required to be contiguous.

CA Datacom/DB Components and System Areas

The CA Datacom/DB components and system areas are described in the following sections.

Multi-User Facility (MUF)

The Multi-User Facility (MUF) is the manager of the system and functionally acts as an operating system for the data. It receives a request from the application and determines how it should be processed. It coordinates the activities that must take place to service the request.

The MUF runs in a separate region in the z/OS environment. The required space for the MUF region varies according to the MUF startup options that are specified using control cards.

The MUF startup options specify the operating environment to CA Datacom/DB. You can change the startup options, but most changes take effect only when the MUF is restarted. Some options can be modified with console commands or by updating some fields in the Dynamic System Tables. Use the options for such things as specifying buffers and information pertaining to the environment and control.

When you specify the options, you provide the resources that CA Datacom/DB uses to operate. Proper specification is the key to optimizing system performance.

CA provides a list of suggested options for each of the products distributed to use CA Datacom/AD. You do not need to make any initial changes to the suggested values in most cases. You can, however, if deemed necessary review the specifications in the sample and make the appropriate adjustments to meet your needs.

Shadow MUF Environment

In a Shadow MUF environment, a primary (that is, a full read/write) MUF and a Shadow MUF (full, but in sleep mode) work together as a pair to minimize any loss of data access if a failure of the full MUF or some other unplanned outage occurs. Using a Shadow MUF environment also prevents the loss of access to data during planned or unplanned outages.

You can only have one Shadow MUF per primary MUF within the Sysplex.

For more information and resources pertaining to the Shadow facility, see the *CA Datacom/DB Database and System Administration Guide*.

Directory (CXX)

The CXX (or Control Directory) functions as a system catalog. The significant pieces of the data definitions stored in the CA Datacom Datadictionary are used to create the CXX. The CXX contains all of the information that is needed to satisfy any non-SQL request to CA Datacom/DB. The CA Datacom Dictionary facility is comprised of the following databases:

- Data Dictionary (DD or DBID 0002) - Stores the basic structure definitions and attributes of the majority of the Datacom entities needed for RAAT or SAAT.
- Data Definition Dictionary (DDD or DBID 00015) - Used for SQL processing, information concerning tables, views, and synonyms is obtained from both the CXX and the CA Datacom Datadictionary. The CA Datacom Datadictionary portion of the information is obtained from a high performance cache called the Data Definition Directory (DDD). This enhances SQL performance.

CA distributes the CXX with all required definitions for the base CA Datacom/AD system and recommended sample databases. Each CA using product that installs CA Datacom/AD adds its own database definitions to the CXX.

The name for the CXX that you assign at your site is critical and needs to be unique for each CXX data set. If you have the need for multiple MUF environments, each is normally going to have a unique CXX name and a unique set of data sets that it is going to own.

Because a site has many z/OS Systems and multiple Sysplex occurrences, there could be a need for more than one CA Datacom environment.

To separate the identity of each environment, a unique name must be assigned. The name should be seven (7) or less positions long and meet the requirements needed to be part of a data set name node. To make data set ownership clear and prevent problems caused by having data sets used by multiple (or wrong) environments, it is a best practice (though not a requirement) to make the name a node in each data set.

The CA Datacom system data set controlling the environment is the Directory with a data set DDNAME of CXX. The name selected for each environment is provided to CA Datacom as the CXXNAME. Access to the data set environment is primarily through the MUF and secondarily through the utility function processor named DBUTLTY. The MUF is often a single instance of an executing program DBMUFPR to provide services to all accesses to the data in this environment. In this configuration, the name of the MUF would be the same as the name of the CXX. The MUF can also be configured as two instances or Jobs with one the primary and active server and the second doing some tracking as a shadow to take over processing quickly should the primary fail. In this configuration, the CXX name would become the MUFPLEX name as the identity of the group of MUF instances, and each MUF instance would need its own identity or MUF name that would be the CXX name suffix with numbers 1 and 2 or the letters A and B.

To define the specific environment, CA Datacom provides one Macro (DBSYSID) with key word options providing the names of the CXX and the MUF instances. In addition, it provides many other options that define the environment configuration.

Important! The CXX, CA Datacom Datadictionary, and the DDD-Database are the heart of the CA Datacom/AD production system. Protect the heart of the system by retaining the backups taken at the end of each install or upgrade. Always back up the CXX and the two CA Datacom Datadictionary databases (DBIDs 2 and 15) at the same time.

Print, Statistics and Diagnostics Area (PXX)

CA Datacom/DB uses the Print, Statistics and Diagnostics Area (PXX) to hold various operational information. This information is used to produce printed reports, dumps, and traces. It also holds various system and job statistics that can be used to monitor and tune the system or to solve problems. The statistics are collected each time a batch job ends or CA Datacom CICS Services is shut down. These dumps, traces, and statistics are available for printing any time through the REPORT AREA=PXX option of DBUTLTY. For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

CA Datacom/DB opens and clears the PXX as output each time the MUF is started. This data set also contains statistics information collected from the MUF and jobs.

The PXX can become full during the execution of the MUF. If it does, a message is posted and job statistics and return code dumps are no longer collected. This does not have an adverse effect on the actual execution of the MUF.

MUF startup option SYSOUT is the preferred mechanism to collect this kind of documentation. SYSOUT does not remove this requirement for a PXX data set but can reduce the size of the data set that the PXX is allocated for this DD name.

If the MUF startup option PXXSTATS is set to EOJPRT, the MUF writes the summary statistics for the MUF EOJ report in SYSPRINT DD and not to the PXX.

If you use these MUF startup options, PXXSTATS EOJPRT and SYSOUT, the PXX has very little activity, but the data set for the PXX is still required. Both of these MUF options are the recommended and best practice usage.

Logging and Recovery System

The logging and recovery system consists of the Log Area (LXX), Force Area (FXX), and the Recovery File (RXX). The LXX and the RXX are used when recovering from a system failure or program malfunction. The LXX and FXX are used for RESTART processing.

The LXX is short term permanent storage for the maintenance transactions (such as, the OPEN, CLOSE, ADD, UPDATE, and DELETE functions) logged by application programs.

The RXX permanently stores the log records after they have been written (*spilled or archived*) from the LXX. If the LXX becomes full, all MUF processing stops. Once a log record is spilled, its space is available for re-use. CA recommends that you use a tape data set for the RXX.

If CA Datacom/DB Transaction Backout is specified for an application that abends, the MUF backs out all data changes made by the program to tables with logging specified since the last *checkpoint*. For programs not using checkpoint, the backout point is the beginning of when the program started. CA Datacom/DB uses the active LXX and sometimes the latest RXX to accomplish the backout. If required, the RXX can also be used to perform forward or backward recovery of lost data and changes.

Force Area (FXX)

The Force Area (FXX) contains logging-related information that has been *forced* from the Log Area (LXX). The FXX is paired with the LXX in that it must have the same physical and logical block size. The number of tracks required is dependent upon the number of tasks generated in the MUF.

System Databases

During the installation of CA Datacom/AD, databases 1 through 20, 1000 through 1020, 2000 through 2020, 3000 through 3020, 4000 through 4020, and 5000 are reserved for use by CA Datacom/DB, CA Datacom Datadictionary, and other CA products. The following identifies the reserved IDs.

- CA Datacom Datadictionary databases (DBID 002 and 0015)
- CA Dataquery database (DBID 003) - Optional
- Accounting Facility (DBID 0004 and 0005) - Optional
- Compound Boolean Selection (DBS) database (DBID 0006 and 1006)
- SQL Default database (DBID 0016)
- SQL Temporary Table Manager (TTM) database (DBID 0017)
- Dynamic System Table database (DBID 1000)
- MUF History database (DBID 1007) optional
- Auto-Status and Auto-Collect databases (DBID 10018, 1019, and 1020) optional
- Change Data Capture database (DBID 2009) optional
- Sample databases (DBID 1 and 10) optional

Note: Since the CBS and SQL TTM are work databases, they are usually defined to MUF as virtual databases that do not require DASD.

Sample Databases

Sample databases are provided for several reasons:

- Utilized by our IVPs to ensure the DBMS is completely installed and working
- Provided as educational tools for learning
- Provides a non product database that can be used to test procedures
- Provides a common database structure and data to be used in recreating problems

The following are the sample databases:

- Human Resource database (DBID 0001)
- Order Entry database (DBID 0010)
- SQLDemoDB (DBID 1001)

Databases for CA Using Products

For more information, see Products that Use CA Datacom/AD (see page 53).

CA Datacom DBUTLTY

DBUTLTY is the main CA Datacom/DB utility that you use to perform most of the CA Datacom/AD data and index maintenance functions. For more information and syntax formats for the functions, see the *CA Datacom/DB DBUTLTY Reference Guide*.

Commonly Used CA Datacom Utilities

The following are the CA Datacom utilities that are used by product:

- DBUTLTY and DRREPORT used by CA Datacom/DB
- DDUTILTY, DDUPDATE, DDRMFLM, and DDBTGLM used by CA Datacom Datadictionary
- DDRTVCAT used by CA Datacom/AD (only)
- DBXMMPR and DBSQLPR used by SQL Option
- DQBATCH used by CA Dataquery
- VLSUTIL used by CA IPC

More Information:

- For more information about DBUTLTY, see the *CA Datacom/DB DBUTLTY Reference Guide for z/OS*
- For more information about DRREPORT, see the *CA Datacom/DB Reporting Facility User Guide*
- For more information about DDUTILTY and DDUPDATE, see the *CA Datacom Datadictionary Batch Guide*
- For more information about DDRTVCAT, see DDRTVCAT (see page 31).
- For more information about DBXMMPR and DBSQLPR, see the *CA Datacom/DB SQL User Guide*
- For more information about DQBATCH, see the *CA Dataquery for CA Datacom Administration Guide*
- For more information about VLSUTIL, see the *CA IPC for z/OS Implementation Guide*

Chapter 3: CA Datacom Datadictionary

Overview

CA Datacom Datadictionary is a tightly coupled CA Datacom application facility used to manage and store database structure definitions including predefined entities (or record types) and corresponding relationships. The standard entities, as they are interconnected by standard and internal relationships, form the model for the database environment as seen from the perspective of the using product line. This model assures that the format of the data stored in the database matches what a program expects.

As part of the Datacom/AD installation, the CA Datacom Datadictionary will have various CA Datacom system and sample databases defined in the CA Datacom Datadictionary. Each structured database will be promoted from a TEST to PRODuction status and then cataloged from the CA Datacom Datadictionary to the CXX.

During the post-installation for many of the CA using products, your CA Datacom Datadictionary is fully populated with the definitions required by the product. Therefore, you are not required to use any of the CA Datacom Datadictionary online or batch facilities to customize it. If you have to tailor CA Datacom Datadictionary for your system, see additional CA Datacom Datadictionary manuals as needed, including the following:

- *CA Datacom Datadictionary User Guide*
- *CA Datacom Datadictionary Batch Reference Guide*
- *CA Datacom Datadictionary Online Reference Guide*

Restrictions

Restrictions regarding how to use CA Datacom Datadictionary in the CA Datacom/AD environment are as follows:

- You cannot create database structures other than the ones that are provided by the CA using product.
- You are only allowed to redefine existing columns (fields) and add keys to the tables in the database approved for use with the CA using product.
- You cannot add columns (FIELD entity-occurrences), change the size of the supplied keys, or create your own databases, areas, or tables to use with the CA using product. You can make a duplicate of an existing database structure.
- Specific naming conventions are enforced.
- You must use the DDRTVCAT utility to catalog the database from PROD status in the CA Datacom Datadictionary to the CXX after you make any changes. DDRTVCAT validates and ensures that no critical changes are made to the shipped CA database structure.

CA Datacom Datadictionary Terminology

CA Datacom Datadictionary is a collection of information in categories that are called entities. Any data in CA Datacom Datadictionary is associated with an entity-type (such as, record type). DATABASE, AREA, and TABLE are some specific entity-types.

Each instance or value within the entity-type is an entity-occurrence (or occurrence). For example, defining a database entails storing information about the database in a DATABASE entity-occurrence. Each database is listed by its unique name as an entity-occurrence within the DATABASE entity-type.

The classification of information stored about each entity-occurrence is called an attribute. The actual information stored for each attribute is an attribute-value. The attribute-value provides specific information about the entity-occurrence.

Examples from DBID 0001

ENTITY-TYPE - DATABASE
ENTITY-OCCURRENCE - HUMAN RESOURCE

ENTITY-TYPE - TABLE
ENTITY-OCCURRENCE - PERSONNEL

ENTITY-TYPE - FIELD
PARENT ENTITY-OCCURRENCE - PERSONNEL
ENTITY-OCCURRENCE - STREET-ADDRESS

ATTRIBUTE:	**VALUE:**
- JUSTIFICATION	LEFT
- TYPE	CHARACTER
- SIGN	NO
- LENGTH	24
- DESCRIPTION	EMPLOYEE STREET ADDRESS

The CA Datacom Datadictionary manages multiple copies of a structure using a combination of version numbers and status levels. The version number is a three digit number that is used to identify a specific version of a structure. That version number follows the structure throughout the Dictionary.

There are three statuses where a structure can reside:

- TEST
- PROD
- HIST

New or altered structures start in TEST status. When it is finalized, the structure is verified and PRODDED which means that the current PROD structure moves to HIST status and the TEST structure is copied to PROD status.

Batch Utilities

Use the CA Datacom Datadictionary batch utilities when you are working on a large amount of database definition updates and to catalog a database.

For details on these utilities, see the *CA Datacom Datadictionary Batch Reference Guide*. When you choose to perform CA Datacom Datadictionary tasks in batch, you use an editor to create and edit input which you submit with JCL to execute the various CA Datacom Datadictionary utilities. The utilities execute CA Datacom Datadictionary transactions to accomplish each task.

The following are the batch utilities that are important to the using products:

DDRTVCAT

This processing utility is provided for use with the CA using products only. It reads the data definitions in CA Datacom Datadictionary and applies them directly to the CXX. For more details about formatting input to the CXX, see the section on defining multiple databases.

DDBTGLM

The Backup/Transport Generator program creates transactions for use with the DDUPDATE utility from selected entity-occurrences in one CA Datacom Datadictionary which can be used to add or update entity-occurrences in another CA Datacom Datadictionary system. You can also use DDBTGLM to create an archive copy of entity-occurrence information in batch transaction format.

DDRMFLM

This utility is used to reformat CA Datacom data records.

DDUPDATE

This update utility is used in the general maintenance processing of entity-occurrences. In addition, you can perform all CA Datacom Datadictionary report functions with this utility.

DDUTILTY

This report utility provides printing batch reports and generating source language statements. The Source Language Generator (SLG) facility allows the field definitions of ELEMENT, KEY, and RECORD entity-occurrences to be reproduced as high-level language (copybook) statements suitable for COBOL, PL/I, or Assembler.

DDRTVCAT - Runtime and Verify Catalog Program

For CA Datacom/AD, use the DDRTVCAT (CA Datacom Datadictionary Runtime Verify and Catalog) Utility to perform the following functions. When these functions are executed, the CA Datacom Datadictionary VERIFY function is also executed and any resulting error messages are issued. See the CA Datacom Datadictionary error messages in the *CA Datacom/DB Message Reference Guide*.

-CXX CATALOG

Specifies the occurrence name of the database to be cataloged to the CXX.

-RTV BASIS

Specifies the occurrence name of the database to be used as a model or base when running the catalog program. The -RTV BASIS transaction is optional. It is used when making an exact copy of an existing database in order to have multiple databases.

Important!

- The database being cataloged cannot be open for update by the MUF or any Single-User job when running DDRTVCAT.
- The CXX is the heart of the CA Datacom/AD production system. Therefore, we recommend that you run the DBUTLTY BACKUP AREA=CXX function prior to executing the DDRTVCAT program. See the *CA Datacom/DB DBUTLTY Reference Guide* for information on the BACKUP function.

  Note: The name of the CXX is unique for each occurance of the physical CXX dataset.. Starting in this version of this product the name will be enforced to be unique.
- The DDRTVCAT program first compares the database description that was shipped with the CA using product to the newly modified database description and verifies changes. You are only allowed to redefine existing fields and add keys. DDRTVCAT also makes sure that no other changes are made to the shipped database structure. If the DDRTVCAT checks meet the established criteria, the program copies the database to PRODuction status, catalogs the database structure, and enables it for use with CA Dataquery or CA Ideal for CA Datacom.

Error Messages

The DDRTVCAT error messages are listed in the *CA Datacom/DB Message Reference Guide* in the chapter containing CA Datacom Datadictionary numbered messages. The format is DDRTV*cnnnn*.

When to Use

The DDRTVCAT utility is provided for use with CA Datacom/AD. It can, however, also be executed in a full CA Datacom/DB environment.

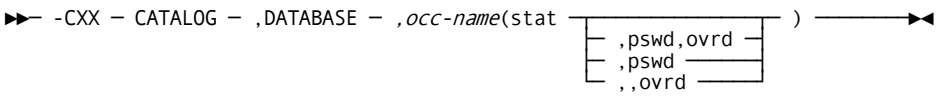
During the installation of the CA using product, your CA Datacom Datadictionary is fully populated with the definitions required by the CA using product. If you find that you need to tailor CA Datacom Datadictionary for your particular requirements, use the DDRTVCAT utility to catalog these changes to the Directory (CXX).

-CXX CATALOG - Cataloging the Definitions to the Directory

Use the -CXX CATALOG transaction submitted through the DDRTVCAT utility to verify the database structure and catalog the occurrence definitions to the CA Datacom/AD Directory (CXX). You can catalog TEST and PRODuction status DATABASE structures.

For the CA Datacom/DB Accounting database, use the DBUTLTY ACCT CATALOG option. See the *CA Datacom/DB DBUTLTY Reference Guide*.

The following is the syntax for this transaction:



,DATABASE

(Required) Indicates that the occurrence is a DATABASE occurrence.

,occ-name

(Required) Specifies a valid DATABASE occurrence name.

Valid Entries:

Any valid DATABASE occurrence name

Default Value:

(No default)

(stat)

(Required) Specifies the status of the occurrence used as input to the DDRTVCAT execution.

Valid Entries:

T001 to T999
PROD

Default Value:

(No default)

,pswd

(Optional) Enter the user-assigned password for the DATABASE occurrence within the parentheses surrounding the stat parameter. If the occurrence is not protected by a password and you must enter an override code, enter the preceding comma.

Valid Entries:

Password assigned to the occurrence

Default Value:

(No default)

,ovrd

(Optional) Enter the system override code within the parentheses surrounding the stat parameter if you are selecting an occurrence which is protected from unauthorized update or access by a lock level. If the occurrence is not protected by a lock level, you can omit the override code and its preceding comma.

Valid Entries:

System override code

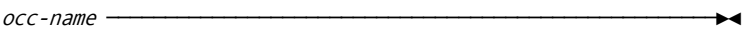
Default Value:

(No default)

-RTV BASIS - Modeling a Database

Use the -RTV BASIS transaction to identify the existing database structure on which to model the new database structure.

The following is the format for this transaction:



,occ-name

(Required) Specifies the DATABASE occurrence name for the structure provided by the CA product that is being installed or upgraded.

Valid Entries:

Any valid DATABASE occurrence name

Default Value:

(No default)

Sample Job Streams and Report

z/OS Example JCL

The following sample JCL catalogs the DATABASE structure MY-CLONED-DB (which is validated against the CA-ORIGINAL-DB structure) definition to the CXX.

Note: Use the following as a guide to prepare your JCL. The JCL statements are for example only. Lowercase letters in a statement indicate a value you must supply. Code all statements to your site and installation standards.

```
//jobname   See the note above and Listing Libraries for CA Datacom Products in JCL.
//       EXEC PGM=DDRTVCAT,REGION=600K
//STEPLIB   See the note above and Listing Libraries for CA Datacom Products in JCL.
//SYSPRINT DD   SYSOUT=a                      Print output
//SYSPUNCH DD   SYSOUT=b                      Punch output
//SNAPER   DD   SYSOUT=a                      dumps
//SYSUDUMP DD   SYSOUT=a                      System dumps
//SYSIN    DD   *                             Input transactions
-USR sample-user,password                     Authorized user information
-CXX CATALOG,DATABASE,MY-CLONED-DB(T001)
-RTV BASIS,CA-ORIGINAL-DB
/*
```

Sample Report - DDRTVCAT

The report contains the following:

- The information that can be entered for this transaction followed by the transaction as entered.

  ```
  CODE FUNCTION,ENTITY-TYPE,OCCURRENCE(VERS,PSWD,OVRD)
  -CXX CATALOG,DATABASE,MP602(T001)                          ;

  CODE FUNCTION,BASIS
  -RTV  BASIS,MP300
  ```

- The first function is the CA Datacom Datadictionary VERIFY function to compare the existing database definition to the new database definition.

  ```
  FUNCTION = VERIFY-OLD-TO-NEW
         ENTITY-TYPE       = DATABASE
         OCCURRENCE        = MP303
  ```

- The next VERIFY function compares the new database definition to the existing database definition.

```
FUNCTION = VERIFY-NEW-TO-OLD
       ENTITY-TYPE       = DATABASE
       OCCURRENCE        = MP303

*** VERIFY ERROR MESSAGES ***

TOTAL ERRORS              0
```

- CA Datacom Datadictionary copies the new database definition from TEST to PRODuction status.

```
FUNCTION = COPY-TO-PROD
       ENTITY-TYPE       = DATABASE
       OCCURRENCE        = MP303
```

- CA Datacom Datadictionary catalogs the new database to the Directory (CXX).

```
FUNCTION = CATALOG-STRUCTURE
           ENTITY-TYPE       = DATABASE
           OCCURRENCE        = MP303
```

- CA Datacom Datadictionary enables the new database definition.

```
FUNCTION = ENABLE-STRUCTURE
           ENTITY-TYPE       = DATABASE
           OCCURRENCE        = MP303
```

- A message indicating that the job has ended.

```
*                          END OF JOB
```

Online Facilities

CA Datacom/AD Version 14.0 does not deliver any of the CICS functionality. The last version of CA Datacom/AD to deliver this functionality was CA Datacom/AD Version 12.0.

The optional CA Datacom Datadictionary CICS online facility displays menu, prompter, and entry panels for selecting functions and entering information from a CICS/TS environment. You use these panels to specify the entity-types and entity-occurrences and to complete definitions. Within this facility, groups of related tasks are called modes.

Note: Your CA Datacom Datadictionary Administrator must add your user identification code to CA Datacom Datadictionary before you can use any of the CA Datacom Datadictionary functions.

For more information and complete instructions about using CA Datacom Datadictionary online, see the *CA Datacom Datadictionary User Guide*.

Multiple Databases Definitions

The CA using products support the ability to define and use multiple physical databases. This facility allows you to design your database around organizational boundaries. For example, you might want to define a test organization and keep it as a separate physical database. Or you may have two different companies, each processed individually, that you want to keep in different databases so that they can be backed up and maintained separately.

Multiple Databases Implementation

Under CA Datacom/AD, a new CA Datacom Datadictionary definition is required for each new database. The simplest way to create this new definition is make a copy of the CA Datacom Datadictionary update transactions used to create the original database. Some products may produce alternative facilities for generating the transactions.

These transactions are in an 80-byte format that can be viewed and edited with any text editor. You should use your text editor to carefully change all instances of the database name (such as CA-ORIGINAL-DB) to the new database name (such as MY-CLONED-DB).

The transaction fields are positional, so using this method requires that you follow the CA naming scheme. Specifically for the database, use the 2900 DATABASE transaction (2900 in positions 1 through 4) to enter the DATACOM-ID attribute-value, and the new database ID number, in columns 6 through 10.

```
-UPD DATABASE,MY-CLONED-DB(T001)
2900 nnnnn (where nnnnn is the new DBID number)
-END
```

After making your changes to all the transactions, submit them to CA Datacom Datadictionary with the DDUPDATE utility to create the new definitions. Then, use the DDRTVCAT utility to activate the database by copying it to PRODuction status and updating the CXX. Note that the DDRTVCAT BASIS parameter is required and must reference the original database. The BASIS parameter allows DDRTVCAT to check your new definition against the original database for compatibility with the using product.

Finally, allocate and initialize the new database using the CA Datacom/DB DBUTLTY INIT function. See the *CA Datacom/DB DBUTLTY Reference Guide* for details on this function.

See the CA using product installation manual for information on jobs that perform these functions for the original databases. You can copy the JCL in those jobs to create the new database.

Important! You should make a full backup of your CA Datacom Datadictionary databases and CXX plus a backup of the update transactions before starting this procedure. See the job in the SAMPJCL library that can be used for this purpose.

User Requirements Table Requirements

CA Datacom/AD User Requirements Tables (URTs) are required for each program that will access the database. The purpose of the URT is to define which tables in which databases the program will require access, whether the access is R/O or R/W, sequential or random access, SQL or native call, and so forth. URTs are assembled and link edited then included with the executable program or dynamically called at execution.

URTs have three formats:

- Batch
- CICS
- SQL

For batch, there can be several URTs. Often, there is one URT per program.

For CICS, the URT is named DBURT*nnnn* where *nnnn* is the database ID. A CICS PPT entry is required for this name. Some CA using products dynamically build the URT, and for those products this step is not necessary.

SQL URTs do not contain specific tables and DBIDs hardcoded in them. These requirements are determined from the items found in the SQL statements and are dynamically built into the URT at execution time from the PLAN information that is stored in the DDD database.

URTs are generally provided as part of the CA using product post-installation process. For more information about installation jobs that compile and link edit the URTs, see the CA using product documentation.

Chapter 4: Regular Procedures

With regard to the topics discussed in this chapter, it is possible that procedures exist specifically for your CA using product. Check the documentation for your CA using product, and if you still have questions about any topic covered in this chapter, contact Technical Support.

Multi-User Facility

This section contains information on MUF activation, status reports related to the MUF, and MUF termination.

The following list provides a summary of the relationship between the CA Datacom functions to z/OS functions:

MUF	MVS
CXX	OS CATALOG
Area	OS DATASET
LXX/RXX	SMF
PXX	SYSI.DUMP
Task	INITIATOR
URT	JCL
COMM STATUS	D A
COMM EOJ	P jobname
COMM SNAP	DUMP jobname
RUNUNIT	JES #

Activation

As part of the CA Datacom/AD installation process, a sample CA Datacom/DB MUF startup job AD14STRT for z/OS is provided. The MUF must be enabled prior to the time the CA Technologies product you are using wants to access it.

If CA Datacom/AD is being utilized with another CA Technologies system product, run the AD14STRT job/STC (where STC refers to a Started Task) at a priority just under the priority of the JES subsystem.

The DBMUFPR is the program that activates the MUF and performs the following functions. This step stays active until the MUF is terminated.

Note: The DBMUFPR program must be executed from authorized libraries.

Typical MUF Start-up Messages and Phases

Messages are displayed on the console or to the JESMSGLG that are informational or concerning things that require an action to be performed. The details about these messages can be found in the *CA Datacom/DB Message* Reference *Guide*.

Echo the MUF startup options to the console in DB01900I

These messages are reflected in part on the JESMSGLG and in their entirety in the SYSOUT. Descriptions of these parameters can be found in the *CA Datacom/DB Database and System Administration Guide*.

Restart the MUF

During the restart process you can get the following messages:

- DB01201I RESTART, LOG NOT OPEN, NO ACTION REQUIRED - If the MUF did not abend the previous time it was active, you receive this message which means that the Log Area (LXX) was emptied and closed cleanly when the MUF was previously terminated.
- DB01202I RESTART, STARTING LOG SCAN, BLOCKS -n - If the MUF has abended the previous time that it was active, you receive this message followed by different restart messages.
- DB01207I RESTART, COMPLETE - This message means that the restart is successful.

Enable the MUF and Identify DB Software Level

When the MUF becomes enabled, the message DB00201I MULTI-USER ENABLED CXX=cxxname MUFNAME=mufname displays on the console followed by the message DB00215I CA-DATACOM/DB SYSTEM VERSION: v.r.

The DBMUFPR uses the following main required data sets. All access either to or from these data sets must be done through the MUF or DBUTLTY.

- The CXX is a control data set and is required. It contains the definitions for all the databases. Some of the information is constant and only updated when the database is defined. Other information is updated when a database is closed or when the DBUTLTY LOAD or RETIX function is used.

  z/OS

  When using CA Datacom/DB dynamic allocation, the CXX data set contains a DSNAME field which is updated when a database data set is initialized. CA Datacom/DB uses this DSNAME to dynamically allocate the file.

- For each database, CA Datacom/DB uses one or more Index Area (IXX) data sets and one or more data area data sets. The DD or DLBL names for an IXX are IXX*nnnn or* IZZ*nnnn,* where *nnnn* is the numeric database ID, and where ZZ can be 01-99. The DD or DLBL names for a data area are *aaannnn* where *aaa* is the 3-character area name assigned in CA Datacom Datadictionary and *nnnn* is the numeric database ID.
- The Log Area (LXX) is read to determine how the MUF last terminated. If the MUF reached a normal end-of-job, this step completes without any further action.

  If the MUF ended with an abend, this step tries to perform the following:

  1. Reset any databases that were left open.

  2. Re-apply all checkpointed (that is, committed) transactions so that the databases are in a stable state.

  3. After reading the Force Area (FXX), any changes that were not checkpointed are removed. (CA Datacom/DB checkpoints at the end of every batch program and at the end of each CICS transaction. Programs may also issue explicit checkpoints.)

  4. Marks the LXX as spillable
- MUF startup option SYSOUT already has the format dumps or Snaps. SYSPRINT has the EOJ Stats that can be used for monitoring and tuning for CA Datacom/AD. MUF startup option PXXSTATS EOJPRT turns this on.

  Note: If the MUF abends, CA Support *must* have the region dump and the report produced by the MUF. SYSPRINT has MUF options and EOJ stats that were collected at the time MUF abended or EOJ. A SVC dump can be generated as well. Collect all the output that the abending MUF generated and the SVC dump that is related for CA Support.

Standard Communication Messages

When a program establishes a connection with the MUF for the first time through a User Requirements Table, the MUF issues the following job-started message in the program's message log:

DB00101I STARTED JOB-jjjjjjjj NUMBER-nnnnn CXX=cxxname MUFNAME=mufname

The DB00101I message can be followed by either of the following messages if the job referenced in the DB00101I is executing from another environment using XCF or CCI as its communication protocol.

- DB00118I XCF TOGROUP=*name* SYSTEM=*system name*
- DB00111I CCISYS=CAICCI *system name* SYSTEM=MVS *system name*

Contained in the message is additional information such as the following:

- Job name of the MUF
- Unique RUN-UNIT assigned to this job to identify it for logging and recovery
- MUFNAME used to communicate to a particular MUF
- CXXNAME reflects the internal name of the CXX Directory this MUF is using
- And more

A similar message is issued when the first online User Requirements Table is opened. In CICS, the RUN-UNIT is in effect until all online User Requirements Tables associated with the CICS are closed (usually the whole session). For this reason, all CICS tasks also include a SUB RUN-UNIT number that uniquely identifies each CICS transaction within a CICS session.

When a program completely drops connectivity with the MUF, the following job-ended message is posted in the MUF region. You receive this message only if you have installed CA Datacom CICS Services. CA Datacom/AD Version 14.0 does not deliver CA Datacom CICS Services. It was delivered in CA Datacom/AD Version 12.0.

DB00102I ENDED JOB-jjjjjjj NUMBER-nnnnn [TTT-FROM=mmmmmmmm]

The message shows the job name, RUN-UNIT number, and other items. You can use the RUN-UNIT number on the DB00101I and DB00102I to determine when a job started and stopped.

Once the MUF is up, you can begin to run jobs that request CA Datacom CICS Services.

MUF Status

At any time, you can run a batch job or enter a console command to request the current status of the MUF. This status provides specific information on all jobs currently accessing the MUF. We recommend that two to three STATUS commands be issued to see progress over time. For more information about this function, see the description of the DBUTLTY COMM OPTION=STATUS or the COMM STATUS command in the *CA Datacom/DB DBUTLTY Reference Guide*. AD14COMM is a sample JOB that is provided. CA SYSVIEW can optionally be used to monitor the activity within the MUF.

Sample Report

Following is a sample report page. For an example report header, see Sample Report Headers (see page 8).

```
MULTI-USER DBDVM02    AVAILABLE         TASKS - ATTACHED-----1, AVAILABLE----24

JOBNAME  R-UNIT TASK STATUS          CMD     DBID TBL     SEQ TIME         I/O ID                      TSN        OWNR/Y TSN-TM
DBDVL132  14160    1 READY TO RUN    ADDIT    997 C02   95727                                          DBDBBBDA             :01

JOBNAME  R-UNIT TASK VALUE/RID
STATJOB2  33654    1     1 PAY
```

AUTOINFO Function

The main purpose of the DBUTLTY AUTOINFO function is to allow you to choose a given MUF and quickly generate information about that environment. This information can be helpful for you and for your communications to Technical Support about a MUF problem.

AUTOINFO should be executed while the MUF is experiencing the problem. However, AUTOINFO can be executed after the MUF has come down or has been cycled. While it will not furnish as much information about a particular situation that occurred in the previous MUF execution, it still provides valuable environmental information.

Sample Report

Following is a sample report page. For an example report header, see Sample Report Headers (see page 8).

```
MUF IS AVAILABLE, INFORMATION IS PROVIDED BY THE MUF.

MUFNAME: MUF4       RELEASE: 14.0               JOB/STC NAME: QA12MUF4
CXXNAME: QAMUF4     JOBID: JOB11528             SYSTEM: CA31
ENABLED TIME:       2008/11/11 13:12:17
CURRENT MUF TIME:   2008/11/11 13:14:59

MUF IS SHADOW:      N                           MUF HAS SHADOW: N
DB SUBSYS PRESENT:  Y                           SQL:            Y
EXTERNAL SECURITY:  N                           ACCOUNTING ON:  N
```

```
DBIDS:     DST: 1000   HISTORY: 1007    CBS TEMP: 0006   CBS HEUR: 1006
TASKS:  TOTAL: 00250_    XCF:  00250_       CCI: 00050_     SIZE: 46080_
```

If AUTOINFO is executed after a MUF has ended but has not been restarted, the function ends with a U0004 condition code. This notifies you that while the report has executed, certain data was not available.

MUF Termination

The MUF can be shut down in the following ways:

Note: In z/OS, a sample shutdown proc AD14STOP is provided in the CUSPROC PDS data set.

The DBUTLTY COMM OPTION=EOJ function or the COMM EOJ console command provides a controlled shutdown. After receiving the EOJ command, the MUF does not allow any new tasks or programs to start accessing the DBMS. However, the MUF does not stop until all existing tasks or programs using the MUF have finished. This could apply to CICS, CA Datacom Server, TSO and other CA products processing, if they are being used. A COMM STATUS reflects jobs are still running but it also reflects an EOJ IN PROGRESS.

For more information about both options, see the *CA Datacom/DB DBUTLTY Reference Guide*. AD14STOP is a sample job provided from the install.

Note: To help determine the problem if the MUF abends, CA Support *must* have the region dump and the report or snaps produced by the MUF.

CA Datacom/AD Environment Maintenance

Consider the following with regard to keeping your CA Datacom/AD system operating successfully.

Backup and Recovery Considerations

Database backup requirements depend on your plans for recovery. For more information about backup and recovery than is found in this section, see the *CA Datacom/DB Database and System Administration Guide*.

AD14BKUP and AX14DD are samples provided on the install. AX14DD is a destructive load example. It refreshes all the databases for DD, DDD, and CASQL DEFAULT with the CXX as it looked when you did the initial CA Datacom/AD Version 14.0 install. You can use it as an example but be certain to modify it to use your backups, not the ones from the install as provided, unless you have a need to go back to the point where you do not want to have any CA Technologies using product installed in the that MUF instance for CA Datacom/AD Version 14.0.

Log Area (LXX)

CA Datacom uses the Log Area (LXX) to allow data to be cached while ensuring the integrity of the data. As updates are made to the database, the LXX is updated with each change that is made. The updates themselves are not necessarily written to the database at the time they are made because CA Datacom uses a pipeline feature to reduce I/O and improve the efficiency of the updates to the database files. Pipelining keeps changes in storage until they can be written to the database in a group, thereby decreasing I/O to the database files and increasing access speed.

In the event of a system failure or CA Datacom MUF outage, the LXX is used to fill in the gaps that may have existed due to this caching. This occurs on startup of CA Datacom if the MUF was not shutdown normally. Database recovery is therefore automatic after an outage if the DASD files are intact and no restore process is necessary. If the database DASD is lost, however, a restore process and forward recovery is needed.

Logged records are kept in the LXX until the database changes are committed to the database DASD, after which completed records can be spilled (removed) from the LXX and written to the RXX. Spilling frees space in the LXX, allowing new log records to then be added. Records spilled to the RXX (typically tapes) can be used to restore a database through the forward recovery process.

Specifications for how the LXX is used by the MUF at your site are determined by the MUF startup options related to logging, which include those shown in the following sample.

Note: The values noted here are illustrative *samples only* and therefore should be reviewed for relevance to your environment.

```
MESSAGE  CRIT,DB00308   Specifies how log messages get treated on console
LOGPEND  100            Number of log blocks to hold records in memory
LOGPOOL  2              Number of log buffers for rollback
LOGRCV   NO             Specifies availability of A DEDICATED RXX TAPE DRIVE
LOGSPILL 75,90,25,30,20 Log % full msg, % force checkpoint,
                        % force spill, % last request to FXX,
                        % active tasks to FXX
```

These parameters, the size of the LXX, and other conditions described in the *CA Datacom/DB Database and System Administration Guide*, determine when spilling of the LXX is needed.

If a program abends or cancels, updates are backed out up to the last checkpoint. Processing stops until the MUF processes the log records. After the backout is complete, the MUF starts processing the transactions that have queued. When a checkpoint is issued or the job completes normally, all uncheckpointed transactions are checkpointed and marked as eligible to be spilled. A CICS transaction boundary is considered a checkpoint.

With the MUF startup option LOGRCV specified as NO, log records are not automatically spilled.

Recovery File (RXX)

The Recovery File (RXX) is a sequential history file of logged maintenance records spilled from the LXX. The RXX is always (except sometimes in z/OS) a tape file that allows you to perform forward recovery for a database that has been restored from a backup. CA Datacom is installed with the RXX option effectively turned off. The creation of the RXX is determined by the CA Datacom/DB LOGRCV MUF startup option.

We recommend you specify the creation of an RXX by coding NO (that is, no dedicated tape drive for RXX) for the CA Datacom/DB LOGRCV MUF startup option. Specifying LOGRCV NO creates an RXX by running a SPILL job from the LXX. With LOGRCV NO specified, the RXX is neither active nor opened during CA Datacom/AD execution. This is the most commonly preferred LOGRCV option.

The other options for LOGRCV are NEVER and YES.

YES specifies active Recovery File. A dedicated tape drive is used for the Recovery File. If you specify the MUF startup option LOGRCV as YES, you are not allowed to specify the MUFPLEX MUF startup option.

NEVER specifies the Recovery File does not exist. NEVER is the value assigned to this parameter during installation. You should carefully evaluate the impact of specifying this because, with a LOGRCV value of NEVER, recovery is not possible and transaction backout can be compromised. Remember that this option affects all databases, including the CA-required databases such as CA Datacom Datadictionary and the DDD.

Note: There is no default for LOGRCV if you code the LOGRCV option. However, if you do not specify this option, the default is an Inactive Recovery File.)

Because log records are not automatically spilled with LOGRCV NO, to write spillable records to the RXX you must mount the RXX tape and run the SPILL function of DBUTLTY. If one or more contiguous blocks are spillable, they are all spilled. Create a spill job using the DBUTLTY SPILLOPT SPILL function as detailed in the *CA Datacom/DB DBUTLTY Reference Guide*.

AD14LXXS is a sample provided on the install.

Note: A tape generation data set works very well for the RXX. However, when combining RXX GDG data sets, be sure to manually concatenate them in reverse order (such as, oldest to most current).

Forward Recovery

Use forward recovery when one or more data areas in a database are not physically accessible, for example when the physical data pack is destroyed. Load to disk the latest backup copy of the database or selected areas in the database and use the RECOVERY OPTION=FORWARD function of DBUTLTY to apply the RXXs that were created during and after the backup was started originally to the backup. Forward recovery is not possible if you specify LOGRCV NEVER in the MUF startup options. For more information, see the chapter on the RECOVERY function in the *CA Datacom/DB DBUTLTY Reference Guide*.

Backward Recovery

Use backward recovery when one or more update jobs need to be reversed (backed out). For example, if you did not specify transaction backout for a program and a failure occurs, the database is not in a logically consistent state. To perform backward recovery, execute the RECOVERY OPTION=BACKWARD function of DBUTLTY. Backward recovery is not possible if you specify LOGRCV NEVER in the MUF startup options. For more information, see the chapter on the RECOVERY function in the *CA Datacom/DB DBUTLTY Reference Guide*.

Hardware Backup and Recovery

Some sites invest in hardware mirroring of volumes to speed in recovery of DASD. This may be local mirroring or across a Geographically Dispersed Parallel Sysplex (GDPS). If you have mirroring and want to use it with CA Datacom and your CA using product, check CA's online support for technical articles on CA Datacom and mirroring, and contact Technical Support for more information if needed.

System Failure or MUF Failure

With CA Datacom, a system outage or MUF failure is automatically recovered when the MUF is restarted. If the system that CA Datacom is on fails, when CA Datacom is restarted on the system after an IPL or on an alternative system with shared DASD, the database is automatically recovered to the point of failure. CA Datacom processes the LXX and brings the database up to date to the time of failure. This type of recovery through restart processing occurs without a need to perform either a restore from backup or a manual recovery.

Important: Do not issue a RESET or INIT function on the LXX without consulting with Technical Support.

DASD Failure or Disaster Recovery

DASD failure recovery or disaster recovery requires a database backup be available to restore the database. To prepare for these types of failures you must do regular backups preferably when MUF is quiesced or down.

Prior to any backup, take the appropriate steps to flush the pipeline buffers out to DASD before beginning. This can be controlled from the ACCESS, LOCK, or QUIESCE functions.

We recommend that you use DBUTLTY to perform database backups and restores. The BACKUP function of DBUTLTY backs up all components of a database together, ensuring that they are all in sync with each other when they are restored. Non-DBUTLTY backups can also be used as long as the pipeline is flushed to DASD first. Make sure your backup utility of choice will perform a full track image backup of the data area. Index data sets should be allocated and rebuilt for each DBID using DBUTLTY RETIX.

System Backups

The following backups should be performed.

Directory (CXX)

The Directory (CXX) contains the definitions that are required for any database access. With CA Datacom/AD, these definitions are updated infrequently, if at all depending on your using product. However, the CXX is considered the central control file for the DBMS containing record counts, locks, enques, and the assignment of RUNUNIT members. Be sure to always have a current backup available in case of an emergency.

The following is the DBUTLTY control statement:

```
BACKUP AREA=CXX,DDNAME=dddddddd
```

The *dddddddd* identifies the JCL DD statement for the output data set for each of the backups. For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

CA Datacom Datadictionary and CA Dataquery Databases

Installation of the CA using products populates the CA Datacom Datadictionary in your CA Datacom/AD system while a summarized version is stored in the CXX. This structural detail information is stored in databases 0002 and 0015. The CXX and the CA Datacom Datadictionary and DDD-DATABASE databases should always be backed up together. This should be done when there is no activity in the MUF, but the MUF must be active.

CA Dataquery is an optional product that you can install with CA Datacom/AD. If you install CA Dataquery, its database ID is 0003

If the CA Datacom/AD installation process installs CA Dataquery, it installs DBID 0003 and needs to be part of the backup procedure, as shown in the following:

```
BACKUP DBID=0002,DDNAME=dddddddd
BACKUP DBID=0003,DDNAME=dddddddd
BACKUP DBID=0015,DDNAME=dddddddd
```

Use the DBUTLTY BACKUP function to backup the CA Datacom Datadictionary, DDD-DATABASE, and the CA Dataquery databases (optional DBID). For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

Note: AD14BKUP and AX14DD are samples provided in the install.

Virtual Library System (VLS) File Backups

CA Datacom/AD Version 14.0 does not deliver the CICS interface, DCS, with this version, so this section can be ignored. CA Datacom/AD Version 12.0 was the last version to have this functionality.

Optional. There are several Virtual Library System (VLS) files required to support your CA Datacom/AD environment with CICS. Use the VLSUTIL BACKUP function to backup the Virtual Library System (VLS) files. Similar functions exist to RESTORE, FORMAT, REPORT, and so forth. This is only necessary if you are using CICS with your CA using products. For more information, see the CA IPC *Customization and Tuning Guide.*

System Area Restores

Restoring one of the system areas should never be attempted unless you are fully aware of the consequences of the restore. For each of the system backups described earlier, a restore procedure exists.

Those areas which are backed up using the DBUTLTY BACKUP function are restored using the DBUTLTY LOAD function. For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

CA Datacom/AD Version 14.0 does not deliver support for CICS, and therefore VLSUTIL is not needed.

Those files which are backed up using the VLSUTIL BACKUP function are formatted and restored using the VLSUTIL RESTORE function. For more information, see the CA IPC *Customization and Tuning Guide*.

Database Backups and Restores

Some of the CA using products provide backup, restore, and reorganization procedures for their CA Datacom/AD databases. You can also use standard CA Datacom/DB backup and restore procedures as described in the *CA Datacom/DB DBUTLTY Reference Guide*.

If the CA using product does not provide a backup and restore procedure for its databases, you should create one. Creating and restoring this backup, along with the use of the LXX and the RXX, allows your system to be fully recovered.

There are two types of backups, native and physical. The native backup uses the index to back up the data in Native Key sequence. The physical backup reads the data area with records read in the order they appear in the data area. A physical backup allows processing to continue while the database is being backed up. However, allowing maintenance during a backup can result in backup data that cannot be used without performing forward recovery.

The LOAD function of DBUTLTY performs the restore of the above two types of backups.

Database Areas Extensions

During installation, the CA Datacom/DB data and Index Areas are allocated to handle the existing number of data records and keys and to allow for some growth. As the amount of data stored in your system increases, you may run out of space, causing attempts to add more data to receive return codes 07 for data areas and 08 for Index Areas.

Early Warning of Possible File Full Conditions

On a periodic basis, execute the DBUTLTY REPORT AREA=CXX,TYPE=A function. This report provides statistics about all of the data and Index Areas in your CA Datacom/AD system. One of the statistics reported is the percent-in-use. When this percent begins to approach 100 percent, extend the area. For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

Extending an Area

When a database file is allocated, it should be allocated with primary and secondary allocation. This allows you to run the DBUTLTY EXTEND function. You cannot run this function when the database is open. When the CA Datacom/DB area becomes full, add additional space to the area by executing the DBUTLTY EXTEND function. For more information, see the *CA Datacom/DB DBUTLTY Reference Guide*.

You can use the following ways to enlarge an area:

- Use DBUTLTY BACKUP, DELETE, RE-ALLOCATE LARGER, INIT, LOAD for a data area or Delete, re-allocate larger, DBUTLTY INIT and RETIX
- Use the DBUTLTY EXTEND function to perform the following:
  - Dynamically allocate a secondary extent
  - Supply a secondary space override using JCL
  - Extend the area to another DASD volume
  - Enlarge the area using a non-Datacom utility but then use EXTEND to format the excess blocks
- Extend data and Index Areas dynamically when they become full if the database definition has specified DYNAMIC EXTEND=YES and DYN.EXT.TRACKS

For more information about creating and maintaining data and Index Areas, see the appropriate chapters in the *CA Datacom/DB Database and System Administration Guide*.

Estimating Data Space

The DEVICE option of the DBUTLTY REPORT function provides, for a given device type, an estimate of the logical records per block and the total space required to store particular number of records in a CA Datacom/AD data area.

Appendix A: Products that Use CA Datacom/AD

The following are some of the products that use CA Datacom/AD. For each product, the list contains the assigned database IDs (DBIDs), the CA Datacom components that each uses, and other information as appropriate.

Products

The following z/OS products can use the CA Datacom/AD suite and the CA Datacom products that they require:

CA APCDDS Automated Report Balancing and CA Workload Automation CA 7 Edition

DBID: 100

Prefix: DDS

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary) and CA IPC

CA Scheduler

DBID: 430 through 439

Prefix: CH

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary) and CA IPC

CA JARS (All flavors) and CA PMA Chargeback

DBID: 490

Prefix: PMA

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom SQL, and CA IPC

CA NetMaster Network Management for TCP/IP and CA NetMaster File Transfer Management

DBID: 501 and 502

Prefix: NM

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom SQL, CA IPC, and CA Datacom Server

CA WA Restart Option for z/OS Schedulers (formerly CA 11)

DBID: 601 (602 backup)

Prefix: L7

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary) and CA IPC

CA Disk

DBID: 650 through 699

Prefix: DMS

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom SQL, and CA IPC

CA Ideal Option for DB2

DBID: none (stores definitions in CA Datacom Datadictionary)

Prefix: n/a

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom CICS Services, and CA IPC

CA IMS Toolkit Products

DBID: 615

Prefix: ITK

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom SQL, CA Dataquery, CA Datacom CICS Services, and CA IPC

CA Common Services for z/OS

DBID: 1008, 1009, 1010, 1011

Prefix: TNG

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom SQL, CA Datacom Server, and CA IPC

CA Jobtrac

DBID: 161 - 168

Prefix: HD

CA Datacom products: CA Datacom/DB (includes CA Datacom Datadictionary), CA Datacom SQL, and CA IPC

Glossary

accessor ID (ACID)

(1) An *accessor ID* designates a user in SQL. This is a user ID, not a schema authorization ID. (2) An *accessor ID,* in security products such as CA Top Secret, is a unique character-string identifier by which the product identifies the security record (user characteristics, authorities, and so on) of a user.

address space (z/OS)

For information about an *address space* in z/OS, see the definition of *region.*

alias

An *alias* in CA Datacom Datadictionary is an alternate name for an entity-occurrence.

alternate key

An *alternate key* in a CA Datacom/DB table created through SQL is any unique key other than the primary key. See *key*, *foreign key*, and *primary key*.

area

In CA Datacom/DB, an *area* is an operating system data set that can consist of one logical table or multiple logical tables. Each database consists of one or more index areas (IXX or I*nn*) and one or more data areas.

authorization identifier (AUTHID)

In SQL, an *authorization identifier* (AUTHID) is what identifies a schema. Every table, view, plan, and synonym in SQL is contained in a schema. A fully qualified object name includes the AUTHID of the schema containing the object.

CA Datacom Datadictionary

CA Datacom Datadictionary is a central, integrated, and active control facility that provides the basis for shared and consistent system resource management in the CA Datacom/CA Ideal environment. It is the repository of descriptive data for CA Datacom/DB and other CA products and is a component of CA Datacom/DB.

CA Datacom DB2 Transparency

CA Datacom DB2 Transparency is a software conversion tool that provides applications and data to the CA Datacom/DB environment.

CA Datacom DL1 Transparency

CA Datacom DL1 Transparency automates the reengineering of DL/I and IMS data to the CA Datacom/DB environment, then provides an application transparency which preserves the use of existing DL/I applications.

CA Datacom TOTAL Transparency

CA Datacom TOTAL Transparency automates the reengineering of TOTAL data to the CA Datacom/DB environment, and then provides an application transparency which preserves the use of existing TOTAL applications.

CA Datacom VSAM Transparency

CA Datacom VSAM Transparency allows VSAM applications to execute after VSAM files have been defined to CA Datacom Datadictionary as CA Datacom/DB tables and the data has been transferred to these tables.

CA Datacom/AD

CA Datacom/AD is a subset of the CA Datacom product line that provides for the access and storage of data for various CA using products (application products) while providing significant enhancements to performance and ensuring data integrity.

CA Datacom/DB

CA Datacom/DB is a high-performance relational database management system (DBMS) used to organize, store, update and retrieve your corporate information.

CA Datacom/DB Reporting Facility

CA Datacom/DB Reporting Facility generates customized reports from data stored in CA Datacom/DB.

CA Dataquery

CA Dataquery is a query product designed to access, manipulate, and report mainframe data. Provides personal databases and supports its own language (DQL mode) and SQL language (SQL mode). CA Dataquery requires CA Datacom/DB.

CA Ideal

CA Ideal is a CA fourth-generation programming language.

column

(1) In CA Datacom/DB, a *column* is a vertical set of data in a table. Each column has a name and a specific data type. All the values in a column have the same characteristics. Each row in a table is defined to contain the same columns in the same sequence. (2) In CA Datacom Datadictionary, a *column* is defined as a FIELD entity-occurrence.

commit

In CA Datacom/DB, a *commit* frees locks so that data which has been changed by an application or user can be referenced by other applications or users. When data has been committed and a new commit point is established, committed data cannot be rolled back.

commit point

In CA Datacom/DB, a *commit point* is the point at which data is considered to be consistent.

CXX

See Directory.

database

(1) In CA Datacom/DB, *database* is a collection of data and an index for finding data that can be accessed according to data values. (2) When conforming to the CA Datacom Datadictionary CA Datacom/DB Model, a *database* is a component of an information base, where a DATABASE entity-occurrence is composed of one or more TABLE entity-occurrences where data is stored.

Database Administrator

A *Database Administrator* is a person responsible for administering the operation of the CA Datacom system. A Database Administrator's responsibilities include maintaining data integrity, maintaining accessibility of data, and monitoring performance.

database management system (DBMS)

A *database management system (DBMS)* is a layer of software used to manage and access databases. A DBMS shields users from hardware-level detail and provides a higher-level view of databases. For example, CA Datacom/DB is a DBMS.

dataview

A *dataview* is a logical user view of the data, independent of the area in which the data is stored. Used by CA Ideal and CA MetaCOBOL+, a dataview provides improved insulation of user requests from the physical location of the data.

Directory (CXX)

The *Directory (CXX)* is the CA Datacom/DB information base used to store definitions of the databases, areas, tables, fields, keys, and elements. It acts as a high-speed directory when data access is requested. CXX definitions are maintained through the CA Datacom Datadictionary.

domain

In SQL, the *domain* is the range of possible values for a column in a table.

DQL mode

DQL mode provides CA Dataquery features that differ in varying degrees from the version of CA Dataquery accessible to users operating in SQL mode and using interactive SQL commands.

element

All data in CA Datacom/DB is defined as a component of an *element.* An element is defined as the smallest logical grouping of data that can be accessed by CA Datacom/DB commands. An element consists of a single field or a group of contiguous fields defined to a single table or a record.

element list

In CA Datacom/DB programming, an *element list* specifies which data elements to retrieve, update, or add.

embedded SQL

Embedded SQL is defined as SQL statements embedded in a host language program that are prepared during the program preparation process before the program is executed. After it is prepared, the statement itself does not change, although values of host variables specified within the statement could change.

field

(1) In a CA Datacom/DB table, a *field* is the logical portion that contains a value. (2) Columns specified in SQL processing are defined as FIELD entity-occurrences in CA Datacom Datadictionary. See the CA Datacom Datadictionary documentation for discussions of fields as defined by CA Datacom Datadictionary. (3) In most cases, CA Dataquery uses the term *column* unless referring to a unique CA Datacom Datadictionary characteristic. Also see *column, compound field, entry field*, and *repeating field.*

Force Area (FXX)

The *Force Area (FXX)* contains logging-related information that has been forced from the Log Area (LXX). The FXX is used in conjunction with the LXX for startup recovery. The FXX is paired with the LXX in that it must have the same physical and logical block size.

foreign key

In SQL, a *foreign key* is a key defined to hold the same values as the primary key of a related table for the purpose of ensuring referential integrity of related data.

FXX

See *Force Area.*

Index Area (IXX)

In CA Datacom/DB, the Index Area (IXX) is the area of the database that contains the index for tables in that database.

index-only processing

In CA Datacom/DB, retrieving data from the index without reading the related record.

information base

The information base is the total collection of databases operational under one copy of CA Datacom/DB.

IXX

See *Index Area (IXX).*

JCL

Job Control Language. A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

JES

JES stands for Job Entry Subsystems.

key

(1) A *key* is a value stored in an index with pointers to its position in the physical database. A key is used to locate a row quickly. (2) A *key* is composed of one or more columns of a table, where its logical significance depends upon its type. See *alternate key, foreign key, Master Key, Native Key*, and *primary key*. (3) A *key* is reserved for referencing logical concepts which impact functionality as opposed to "index," which is reserved for referencing physical addressing mechanisms which impact performance.

key value

A *key value* is the actual contents of a key at a given time.

Log Area (LXX)

In CA Datacom/DB, the *Log Area (LXX)* is a data set used as a temporary storage area for transactions. When the Log Area reaches a user-defined spill point, it is written to the Recovery File (RXX), which can be used to restore the database. The LXX used with the pipeline feature prevents data loss. All updates are logged directly to the LXX. See also *pipeline* and *Force Area (FXX).*

logging

In CA Datacom/DB, *logging* is the building and writing of log records when a CA Datacom/DB maintenance function is performed or when specific log commands (such as LOGCP and LOGIT) are issued.

logging cycle

In CA Datacom/DB, a *logging cycle* is a logical division on the Recovery File (RXX) consisting of a cycle start record, log data records, and a cycle end record.

LPAR

LPAR stands for logical partition. Dynamic logical partitioning allows hardware resources to be shared by more than one independent operating environment. This is accomplished by dividing a single server into several independent virtual servers or LPARs (logical partitions).

LXX

See *Log Area.*

Master Key

Each table must have one of its keys designated as the *Master Key.* This key is the key (a) for which a value always exists in the index (even when the value is blanks or zeroes) and (b) which can be defined as nonduplicatable and nonupdatable.

Master List

A *Master List* is a definition of the system environment in which an application program is to be executed.

Native Key

Each table must have one of its keys designated as the *Native Key.* This key is the key used to sequence the records when the table is unloaded. When the table is loaded, the records are therefore loaded in the Native Key sequence.

object

An *object* is anything that can be created or manipulated with SQL, for example, tables or views.

partition

(1) For distributed database management system, see *partitioned database*. (2) For z/OS environments, see *partitioned table.* (3) For z/VSE environments, see *region.*

partitioned database

In CA Datacom/DB, a *partitioned database* is a physical database composed of one or more tables, the records of which are limited to those with Native Key values within a range specified by partitioning criteria defined to CA Datacom Datadictionary. Each partition of the logical parent database has its own index and can be distributed to a different location for local access. However, the logical parent of the partitions can still be accessed as a single image.

partitioned table

In CA Datacom/DB, a *partitioned table* is a table whose rows are stored in table partitions located in multiple data areas. The table is partitioned based on user-defined partitioning criteria specified in a CA Datacom Datadictionary PARTITION-COLUMN-VALUE entity-occurrence.

partitioning

The term *partitioning* is synonymous with horizontal partitioning, that is, dividing a logical database into two or more discrete, physical databases according to Native Key ranges defined for each individual partitioned database.

partitioning criteria

Partitioning criteria is criteria for partitioning a database into two or more databases based on a high-order part of the Native Key values of its records, as defined in the CA Datacom Datadictionary PRT table.

password

A *password* is a unique string of characters that are supplied at sign-on to meet security requirements. A password may be but is not necessarily always required.

pipeline

The CA Datacom/DB *pipeline* feature caches database updates to reduce DASD activity and increase database access speed.

plan

A *plan* is the control structure produced during the bind process and used by CA Datacom/DB to process SQL statements encountered during application execution. The plan contains information about SQL statements. Each program containing embedded SQL statements must have a plan before being executed.

positioned operations

In SQL, *positioned operations* are operations performed on the row of a table identified by the positioning information maintained with the cursor control structure, such as the SQL FETCH, positioned UPDATE, and positioned DELETE statements.

primary exclusive control

A task gains *primary exclusive control* over a record when that task acquires the record for update or deletion. Once the update or delete has taken place, primary exclusive control is dropped over that particular record. Primary exclusive control may also be dropped by the RELES and RELFL commands and for online programs, a program exit. See also *secondary exclusive control*.

primary key

Defined to a CA Datacom/DB table through SQL, a *primary key* is the one required unique key that provides a guaranteed method of addressing each row of the table by using the data value of all participating columns. See *alternate key, foreign key*, and *key*.

primary table

In CA Dataquery, the *primary table* is the table read first when more than one table is to be read in the search for data. If the primary table is to be joined to another database table for this search, it must share a key or column with that table.

privilege

A *privilege* defines what kind of access (SELECT, INSERT, UPDATE, DELETE) is authorized for a particular resource. You must be granted the necessary privilege to execute specific commands, or to grant privileges to other users. The CA Datacom/DB Security Facility must be installed before privileges may be defined.

record

(1) A *record* is a collection of fields. (2) In CA Datacom Datadictionary, a *record* is the substructure in the CA FILE Model within a FILE structure. The RECORD substructure contains fields and can contain ELEMENT and KEY substructures. Also see *row*.

record-at-a-time (RAAT)

In CA Datacom/DB, *record-at-a-time* programming is an access technique in which records are selected by specifying a previously defined key name and a key value.

recovery

The *recovery* process restores a database to its last known point of stability. This process can mean the reconstruction of a database from a previous backup copy and a Recovery File (RXX) that contains all updates (forward recovery), or it can mean reversing the effect upon the database (backward recovery) of one or more jobs.

Recovery File (RXX)

The *Recovery File (RXX)* is the history file of logged data records.

set-at-a-time (SAAT)

In CA Datacom/DB, the set-at-a-time (SAAT) commands access a set of rows (records) that are defined based on the data values contained in the table. Individual rows of the set or a count of the rows in the set can be accessed. The set is selected by specifying a search condition. The rows of the set are returned based on a user-specified sequence of columns in the row, either ascending or descending or as selected by CA Datacom/DB, if no user sequence is requested.

SQL

SQL is a database sub-language which can be used to define, manipulate, and control data in a database.

Index