CA Datacom®/AD

Best Practices Guide

Version 14.0

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA Datacom®/AD
- CA Datacom® Datadictionary™
- CA Common Services for z/OS
- CA SYSVIEW® Performance Management (CA SYSVIEW)

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Best Practices Guide Process

These best practices represent years of product experience, much of which is based on customer experience reported through interviews with development, technical support, and technical services. Therefore, many of these best practices are truly a collaborative effort stemming from customer feedback.

To continue and build on this process, we encourage users to share common themes of product use that might benefit other users. Please consider sharing your best practices with us.

To share your best practices, contact us at techpubs@ca.com and preface your email subject line with "Best Practices for *product name*" so that we can easily identify and categorize them.

Contents

Chapter 1: Introduction

This section contains the following topics:

Purpose of this Guide

The guide provides a brief introduction to CA's Mainframe 2.0 strategy and features, and describes the best practices for installing and configuring CA Datacom/AD.

Audience

The intended audience of this guide is systems programmers and administrators who install, configure, deploy, and maintain CA Datacom/AD.

Mainframe 2.0 Overview

Mainframe 2.0 is our strategy for providing leadership in the mainframe operating environment. We intend to lead the mainframe marketplace for customer experience, Out-Tasking solutions, and solution innovation. After listening to customer needs and requirements to keep the mainframe operating environment viable and cost-effective, we are providing new tools to simplify usage and to energize this operating environment for years to come.

CA Mainframe Software Manager™ (CA MSM) is an important step in realizing the Mainframe 2.0 strategy. CA MSM simplifies and standardizes the delivery, installation, and maintenance of mainframe products on z/OS systems. CA MSM has a browser-based user interface (UI) with a modern look and feel for managing those solutions. As products adopt Mainframe 2.0 features and CA MSM services, you can acquire, install, and manage your software in a common way.

CA MSM provides software acquisition and installation that make it easier for you to obtain and install CA mainframe products, and apply the recommended maintenance. The services within CA MSM enable you to manage your software easily based on industry accepted best practices. The common browser-based UI makes the look and feel of the environment friendly and familiar.

We follow the IBM z/OS packaging standards using SMP/E, with some additional CA qualities of service added, to make installation simple and consistent. Additionally, through the synchronization of product releases and the use of common test environments, we will declare a yearly mainframe software stack that includes many new releases with enhanced functionality. This stack is certified for interoperability across the CA mainframe product portfolio and the base IBM z/OS product stack.

Mainframe 2.0 Features

Mainframe 2.0 has the following main features:

CA Mainframe Software Manager (CA MSM)

Delivers simplified acquisition, installation, and deployment capabilities using a common z/OS-based web application delivered through a browser-based UI. CA MSM includes the following services:

Product Acquisition Service (PAS)

Facilitates the acquisition of our mainframe products and services, including product base installation packages and program temporary fixes (PTFs). This service integrates the inventory of products available on your system with CA Support, providing a seamless environment for managing and downloading software and fixes onto your system.

Software Installation Service (SIS)

Facilitates the installation and maintenance of our mainframe products in the software inventory of the driving system. This service enables you to browse and manage the software inventory using a web interface, and automates tasks for products that use SMP/E to manage installation. You can browse downloaded software packages, and browse and manage one or more consolidated software inventories (CSIs) on the driving system.

Software Deployment Service (SDS)

Facilitates the deployment of CA Technologies mainframe products from the software inventory of the driving system. This service enables you to deploy installed products that are policy driven with a set of appropriate transport mechanisms across a known topology. The enterprise system topology can include shared DASD environments, networked environments, and z/OS systems. Policies represent a combination of metadata input and user-supplied input. Metadata input identifies the component parts of a product. User-supplied input identifies the deployment criteria, such as where it will go and what it will be called.

Systems Configuration Services (SCS)

Facilitates the Configuration (New Install) of CA Technologies mainframe products. This service works hand in hand with the Software Deployment Service (SDS). Upon providing a set of deployed target libraries, the Systems Configuration Services has the capability of performing a new installation of a CA Technologies mainframe product that has been targeted for configuration. The process is end-to-end. The client gets to provide the high-level qualifiers, based on naming conventions from the site, the DASD allocation, and a fully configured environment, with the capability of modifying parameters to reflect a fully operational system.

Electronic Software Delivery (ESD)

Enables you to get our products from an FTP server. We have improved this process so that you no longer need to build a tape to install the product.

Best Practices Management

Integrates with IBM Health Checker for z/OS to verify that deployed software follows our best practices. The health checks continually monitor the system and software to provide feedback on whether the software continues to be configured optimally.

Best Practices Guide

Provides best practices for product installation and configuration.

Note: For additional information about the CA Mainframe 2.0 initiative, see http://ca.com//mainframe2.

Chapter 2: Installation

Use CA MSM to acquire, install, and maintain CA Datacom.

Business Value

CA MSM provides a web interface, which works with ESD and standardized installation, to provide a common way to manage CA mainframe products. You can use it to download and install CA Datacom.

CA MSM lets you download product and maintenance releases over the Internet directly to your system from the CA Support Online website. After you use CA MSM to download your product or maintenance, you use the same interface to install the downloaded software packages using SMP/E.

More Information:

- For more information about CA MSM, see the *CA Mainframe Software Manager Guide*.

- For more information about product installation, see the *CA Datacom/AD Installation Guide.*

Chapter 3: Initialization Best Practices

Protect Critical System and User Data Sets

Secure the CA Datacom critical system and user data sets to protect against an accidental or planned intrusion. These data sets are both non-database "system" data sets as well as those data sets that house system and user databases. Certain system data sets can contain customer data in a readable form.

While the CA Datacom external security interface can protect access to the system data sets, systems database data sets, and user database data sets, it does not protect against direct data set access using tools such as IEBGENR or IEBCOPY.

Business Value

Using external security data set protection for these data sets increases your ability to secure and protect your database environment from inadvertent or planned intrusion.

Systems Data Sets

Each system data set is commonly referred to by its z/OS DDNAME. The actual data set name is dependent on the options chosen during installation.

The following is a list of the critical system data sets that must be protected:

CXX

Contains the definitions of all databases accessible in this CA Datacom environment.

- Required to do any function in CA Datacom

- Contains condensed metadata about the CA Datacom/DB databases

- Should be externally secured

    - Must be fully accessible to the user ID used to submit the MUF startup

    - Must be fully accessible to user IDs of users who want to execute any DBUTLTY functions opening the CXX

LXX

Contains user transaction data (including data rows) used to support transaction back out and database restart (after a failure).

- Required to start the MUF

- Contains user data from CA Datacom/DB databases

- Should be externally secured

  - Must be fully accessible to the user ID used to submit the MUF startup

  - Must be fully accessible to the user ID used to execute any DBUTLTY function that empties or reports on the contents of the LXX

FXX

Contains user transaction data (not including data rows) used to support fast database restart (after a failure).

- Required to start the MUF

- Does not contain any user data from CA Datacom/DB databases

- Should be externally secured

  - Must be fully accessible to the user ID used to submit the MUF startup

  - Must be fully accessible to the user ID used to execute the DBUTLTY function that empties the contents of the FXX

RXX

Contains user transaction data (including data rows) used to support database recovery from a backup after a data loss.

- Typically not required to start the MUF. For more information, see LOGRECV=YES in the *CA Datacom/DB Database and Systems Administration Guide*.

- Contains user data from CA Datacom/DB databases

- Should be externally secured

  - Must be fully accessible to the user ID used to submit the MUF startup with LOGRECV=YES specified

  - Must be fully accessible to the user ID used to execute the DBUTLTY function to spill LXX records to the RXX

  - Must be read accessible to the user ID used to execute any DBUTLTY function that reports on the contents of the RXX

The external security administrator should be very careful in granting access to the LXX and RXX data sets as user data rows from any table in the environment can appear on the data set. This data set should be as secure as the highest secured table in the environment.

Access to the CXX and FXX is typically not as secure. However, any inadvertent damage to the CXX could completely shut down the CA Datacom environment.

Systems Databases

Each system database is commonly referred to by its functional purpose and by its assigned database ID (DBID). There is one or more index areas and one or more data areas for each database. Each of these areas has one associated z/OS data set. To refer to the data sets making up the database, we typically use the z/OS DDNAME. The actual data set name is dependent on the options chosen during installation.

The following is a list of the critical system databases that should be protected:

CA Datacom Datadictionary

- Database 0002

    - Index data set, IXX002

    - Data area data set, DD1002

    Contains the metadata definitions of all databases accessible in this CA Datacom environment.

    - Required to populate information into the CXX and the DDD database (see Database 0015)

    - Should be externally secured

        – Must be fully accessible to the user ID used to submit the MUF startup

        – Must be fully accessible to user IDs used to execute DBUTLTY functions against this database

Data Definition Directory - DDD

- Database 0015

    - Index data set, IXX015

    - Data area data set, DDD015

    Contains the SQL metadata definitions of all databases that are SQL accessible in this CA Datacom environment.

    - Should be externally secured

        – Must be fully accessible to the user ID used to submit the MUF startup

        – Must be fully accessible to user IDs used to execute DBUTLTY functions against this database

User Databases

Each user database is commonly referred to by its functional purpose and by its assigned database ID (DBID). There are one or more index areas and one or more data areas for each database. Each of these areas has one associated z/OS data set. To refer to the data sets making up the database, we typically use the z/OS DDNAME. The actual data set name is dependent on the site.

For the AD site, user databases refer to the databases that have been implemented to support the CA product that is using the AD environment. The CA product documentation lists the user databases that are being installed.

User database data set protection depends on the data contents of the tables in that data set (data area) and should be externally secured as follows:

- Must be fully accessible to the user ID used to submit the MUF startup
- Must be fully accessible to user IDs used to execute DBUTLTY functions against this database

The user database index data sets, Index Area (IXX) and Index Area (Inn),, contain index entries that are constructed from data values on the table rows in the data areas of that database. These index entries are typically compressed and would be hard to read outside the CA Datacom environment. It is possible to decipher the index blocks in the IXX and Inn and retrieve a secured data item such as an account number.

For this reason, the IXX and Inn should be externally secured at the same level as the highest secured data table in the database as follows:

- Must be fully accessible to the user ID used to submit the MUF startup
- Must be fully accessible to user IDs used to execute DBUTLTY functions against this database

Make 64-bit Memory Available to MUF Region

Verify that your MUF batch job or started task allows the use of 64-bit memory.

Business Value

The 64-bit memory feature provides additional significant memory to z/OS regions making memory-resident processing much more available. Using memory-resident processing can greatly reduce physical I/O within the database region and can reduce CPU usage.

The Memory Resident Data Facility was enhanced to attempt to place VIRTUAL and COVERED areas into 64-bit memory. If the memory allocation does not fit in 64-bit memory, the MRDF allocations are placed into 31-bit data spaces.

MRDF COVERED data sets can significantly improve MUF request processing. For example, one user site employed MRDF COVERED processing on several high-use data sets. They were able to document a 40 percent reduction in I/O which delivered a 15 percent reduction in CPU consumption per request.

Additional Considerations

The 64-bit memory option is specified in the region parameter in your JCL. However, your site can have certain limitations on 64-bit memory set across your LPAR. Before implementing the change to add 64-bit memory, consult with your systems programming team.

Chapter 4: Configuration Best Practices

Most of the CA Datacom configuration options are implemented through the Multi-User Facility (MUF) startup options. These options provide a wide variety of flexibility for each individual CA Datacom environment.

The *CA Datacom/DB Database and System Administration Guide* provides detailed descriptions of every startup option and each of their parameters. Study the startup options and your respective parameters.

Using Memory Resident Data Facility COVERED Index and Data Areas to Reduce Physical I/O Processing

The primary function of a CA Datacom/DB database system is the ability to store large volumes of tables and rows while providing quick data access and updates capabilities. A unique technique for the CA Datacom site is the ability to make certain data tables memory-resident. The area is copied into memory as the blocks are read (that is, physical I/O) for normal processing. Subsequent read processing that would typically require the area block to be reacquired by physical I/O can be copied back from the block that was stored in the COVERED memory.

Unlike VIRTUAL databases, COVERED databases use memory to save blocks and replace future physical read I/Os, but all updates to the database are still done to the Index and data areas. The memory is allocated when the COVERED specification is encountered. The area in memory is created and as blocks are accessed they are copied into the memory area. The COVERED definition can be done at MUF startup (recommended) or can be added while MUF is active for databases that are currently not open to MUF. All updates to index or data information is protected by logging and recovery.

Unlike VIRTUAL, the COVERED specification can help any data and index processing where a database has significant physical read activity. In addition, the COVERED specification can be done for the specific Index or data area, allowing you to target only the high-read physical I/O data sets.

Business Value

By specifying certain high-read databases (specifically the Index and data areas) as COVERED, you can remove significant amounts of physical read I/O for that database. By removing a significant amount of the physical I/O for the database, the response time for those database requests is improved and the overall I/O reduced. Reductions in physical I/O can also include reductions in CPU usage.

Additional Considerations

For AD sites, the CA Products that are using the AD repository can recommend that a specific index or data area be COVERED to improve processing performance.

In most sites, there are Index or data areas that have high physical read I/O content. Focus on these areas. To look for particular areas that are candidates, review the PXX summary statistics or the MUF_AREA_STATS dynamic systems table to find the areas with high-read I/O.

Implement Symmetrical Multi-Processing

The CA Datacom MUF can use multiple TCBs in parallel to do user request processing. This multiple TCB ability allows sites that have a significant MUF workload to spread that workload across multiple available processors or engines, that is, GP CPUs.

In standard SMP mode, the MUF automatically has one main task TCB where MUF internal processing occurs. This main task TCB is also used to process user requests. In addition to the main task TCB, you can specify a number of additional TCBs that can be started when needed. In this SMP environment, when additional work is available to be processed and the main task TCB is already busy with work, the additional TCB is started as needed to process the latent demand.

Business Value

The user site with heavy MUF workloads can use this feature to spread the MUF workload across multiple TCBs which are able to use multiple processors or engines to complete the workload. While this mode does not typically reduce the total GP CPU consumed, it improves the request processing elapsed time and users' response from the MUF.

The CA Datacom SMP implementation does not allow a specific user request to use more than one TCB. By having multiple TCBs available, a poorly constructed query that is keeping a TCB busy, does not stop other better performance requests from processing on the other TCBs. This protects sites from seeing major performance degradation when these heavy queries are processing.

Additional Considerations

Sites with minimal MUF workloads, should not use SMP processing because it incurs additional overhead required for parallel task locking that is not going to be useful unless multiple task processing is occurring. Also sites having only one available processor or engine would not typically benefit from SMP processing.

SMP processing is implemented in the MUF by specifying the startup option SMPTASK.

SMPTASK *maxtcb,currenttcb,readytorun*

maxtcb

> Sets the high-water mark for additional TCBs that can be added to do request processing.

currenttcb

> Sets the current number of additional TCBs that can be added to do request processing when the amount of demand is present. The MUF adds TCBs as the workload backlog increases until it has reached the value set in *currenttcb*.

readytorun

> Sets the amount of workload backload that is needed before an additional TCB is started. A workload backload is basically a request that is ready to run (that is, ready to be processed) that the MUF has not processed because it is busy on another request.

For example, a *readytorun* of 4, waits until there are four requests that are ready to run and are not currently being processed before it starts the additional TCB (TCB2). The workload backlog would have to grow to 8 *readytorun* requests before another TCB is started (TCB3). This continues until the number of additional TCBs equal the value set in *currenttcb*. Once the workload backlog drops back to zero, the additional TCBs are freed. If the workload backlog redevelops again, the TCB add-on process repeats.

Console adjustments

> The user can dynamically adjust the *currenttcb* number and the *readytorun* number using the MUF console commands SMPTASK and SMPTASKR.

For more information about using SMP processing, see the *CA Datacom/DB Database and System Administration Guide*.

Implementing the zIIP Specialty Processor

IBM has introduced the zIIP Specialty Processor, a CPU engine, for the z9 and z10 hardware systems. The zIIP processor, like its predecessors, the IFL and zAAP processors, can be added to a hardware box without increasing the MIPS rating of that hardware operating environment. Many software license fees are based on the MIPS rating of the hardware, but the specialty processor allows you to increase the processing power of the hardware without increasing the MIPS rating and therefore without increasing the MIPS-based license fees.

Unlike the IFL and the zAAP processor, the zIIP processor allows certain existing z/OS workloads to be executed on the zIIP processor. For these workloads to execute on the zIIP processor, they must be executed as a Work Load Managed (WLM) Enclave SRB.

The SMP processing covered in the section titled Implement Symmetrical Multi-Processing can be changed from a main task TCB with additional TCBs for request processing to a main task TCB and WLM Enclave SRB/TCB pair for request processing. This enhancement allows portions of the existing CA Datacom workloads to be available for off-loading to available zIIP processors. The TCB of the SRB/TCB pair is used to process those parts of the workload not available for execution on the SRB.

Business Value

For sites with zIIP processors, this ability allows you to offload portions of the MUF workload to the zIIP processor which could free up the General Purpose (GP) processors for other workloads. By freeing up GP processors, you can run additional workloads or possibly delay a planned increase to the GP processor capacity.

Additional Considerations

Sites with minimal MUF workloads should not use SMP processing as it incurs additional overhead required for parallel task locking that is not going to be useful unless multiple task processing is occurring.

Sites that do not have a zIIP processor would not typically benefit from WLM Enclave SRB processing.

WLM Enclave SRB processing is implemented in the MUF by specifying the startup option SMPTASK with the additional *tcborsrb* parameter.

SMPTASK *maxtcb,currenttcb,readytorun,tcborsrb*

tcborsrb

> Defaults to TCB which is the standard processing mode for r11 (and before). Changing the specification to SRB tells the MUF to convert from multiple TCB processing to the WLM Enclave SRB/TCB pair processing.

For sites that want the MUF to take advantage of zIIP processors, but want to limit the amount of the workload that is offloaded, the MUF startup option ZIIP_USER_LIMIT can be specified. Typically this limit would only be implemented for test systems where you want to use zIIP processors for some processing but also keep the test systems from putting a significant load on the zIIP processor.

For more information about using zIIP processing, see the *CA Datacom/DB Database and System Administration Guide*.

If you want to estimate the amount of workload that could be offloaded even though you do not have an available zIIP processor, you should implement the SMPTASK parameter with the SRB specification. Even though the SRBs are not dispatched to a zIIP processor, the MUF provides statistics on the amount of workload that would have been dispatched if the zIIP was available.

Enable the History Database for Data Access Statistics

The History Database table CADTCM_HISTORY_ADS, provides a persistent record of data area access. It provides a snapshot of logical read and physical activity read for each area that has been opened since the MUF started up. These statistics can be used to determine how much sequential access is being done per data area. By reviewing these statistics, you can analyze data access trends and determine if the data area is experiencing a decline in the ratio of logical processing requests to physical I/Os. In most cases, a non-sequential data access is not affected by the data order, and its logical requests per physical I/O ratio remains the same.

However, in some cases, sequential data access processing can be negatively affected when the data rows get out of order. Typically, such a decline in important sequential processing applications could indicate the need for a data area or database level reorganization.

Business Value

In many cases, sites run regularly scheduled data reorganizations to reorder the physical sequence of the data. For a data area or database to be physically reorganized, it must be taken offline from user access.

A site also runs these data reorganizations on a regularly scheduled basis. However, in today's 24 hours a day, 7 days a week environment, the times that data can be offline from users is greatly diminishing.

By using the History Database table CADTCM_HISTORY_ADS, you can monitor data area processing trends and determine if and when offline data reorganization is needed.

Additional Considerations

The History Database should be defined automatically in any system that has been installed or upgraded from the r11 level. Before starting the History Database, verify that the index and data areas have been allocated, initialized and loaded null.

The MUF startup option HISTORY must be added to the MUF startup to enable the statistical collection.

HISTORY *dbid*

dbid

> Provides the database ID for the History Database. The standard implementation uses DBID 1007 for the History Database.

The statistics are normally collected once a day at midnight (24:00:00). If you want to change the time of the once-a-day collection, you can specify the MUF startup option HISTORY_END_HOUR to change the hour of the collection.

There is no automated deletion of the statistics records collected in the CADTCM_HISTORY_ADS table. Determine when and how to remove the statistical records.

For more information about using the History Database, see the *CA Datacom/DB Database and System Administration Guide*.

Enable the History Database for Spill Statistics

The History Database table CADTCM_HISTORY_SPL provides a persistent record of DBUTLTY SPILL executions. Detailed information is provided about each of the spills of the Log Area (LXX) to the Recovery File (RXX). This information can be used to determine the contents of each recovery file.

This information can be helpful when you find that you have a damaged database and need to recover the database from a previous backup. Once the backup is reloaded to the database, the database is at the point in time of the backup.

The spill history records help you to determine what RXX files are necessary to run a DBUTLTY forward recovery to bring the restored database forward in time.

Business Value

Typically when database damage occurs, and the user must restore a backup and do forward recovery, time-to-recover is of great importance. Unfortunately, the need to recover quickly can add additional pressure to the job of creating the recovery JCL.

If only one of the RXX files is missed or if the RXX files are specified in the wrong order, the forward recovery can fail and cause the user to restore the backup and run the recovery a second time.

By having the CADTCM_HISTORY_SPL table available, you can quickly determine all of the RXX files that have been created since the backup in question was created. Verify that these RXX files are provided in the correct sequence to the forward recovery process. This information not only speeds the job of creating the recovery JCL, but also improves the accuracy of the JCL creation.

Additional Considerations

The History Database automatically contains the additional CADTCM_HISTORY_SPL table in any system that has been installed or upgraded to the Version 12.0 level or later. This new table shares the same data area (A01) as the other History Database table CADTCM_HISTORY_ADS. Verify that the Index and data areas have been allocated, initialized and loaded null before starting the History Database.

The MUF startup option HISTORY must be added to the MUF startup to enable the statistical collection.

HISTORY *dbid*

dbid

> Provides the database ID for the History Database. The standard implementation uses DBID 1007 for the History Database.

The spill statistics are created each time a DBUTLTY SPILL process is executed in a MUF environment with the LOGRCV=NO option specified.

For more information about using the History Database, see the *CA Datacom/DB Database and System Administration Guide*.

Using Shadow MUF

Use the Shadow MUF to extend the ability for 24x7 MUF availability.

Business Value

The Shadow MUF is a nearly identical MUF job or started task that is usually running on a different LPAR within the z/OS Sysplex. Functionally, the Shadow MUF is waiting in stand-by mode ready to take over the CA Datacom processing should the Primary MUF need to be brought down on goes down due to any unforeseen or unrecoverable error.

Additional Conditions

Typically, the CA Products that are using the AD repository recommend if Shadow MUF processing is needed for this product.

Hot Backups versus Cold Backups

Depending on your requirements, backups generally fall into one of two categories:

- Hot backup - performed when the database or DBMS is still up and servicing read and maintenance requests
- Cold backup - taken when the database or DBMS is down

Business Value

Performing a hot backup provides the following advantages:

- Database and DBMS are actively available to the applications
- DBULTY BACKUP SEQ=PHYSOCAL and non-DBUTLTY backups can be taken at any time.

Performing a cold backup offers the following advantages:

- Simplest form of backup taken whenever all MUFs that reference the database are down.

- Absolute guarantee that all data is backed up

- DBUTLTY BACKUP SEQ=NATIVE can be taken in preparation of a corresponding LOAD which ultimately reorganizes the data into native key sequence. This accesses the records by reading the index for the lowest native key value and retrieving the corresponding data record. This requires exclusive access to the database.

- DBUTLTY BACKUP SEQ=PHYSICAL is the fastest of DBUTLTY backups because it does not read the index at all. It simply starts from the beginning of the data area and copies each record it finds to the backup file. This does not require exclusive access to the data area.

Enlarging the LXX Log File

Enlarging the CA Datacom LXX allows continuous processing.

Business Value

The LXX that is allocated as part of a normal CA Datacom install is sized for an average amount of data records being updated. Depending on the number of records that are updated in some fashion by either an application or the CA Datacom DBMS and how often you do or do not want to spill, we recommend that you enlarge the size of your LXX.

Finding Compatibility Information on CA Datacom

To find the operating systems or CICS release compatibility for any of the CA Datacom family of products go to our support web site at http://support.ca.com. After signing in, click on COMPATIBILITIES in the left column of the support home page. From the CA Product Compatibilities page, select the platform of interest then select the DATABASES category from the platform compatibility page. On the DATABASES category page, find the product, platform release and product release level. The informational solution number that contains the desired information is located on the far right hand side.

Business Value

This process enables you to obtain faster resolution or queries of your research and fewer support calls.

Chapter 5: DBUTLTY Configuration

The CA Datacom/DB DBUTLTY program provides the majority of database utility functions for the CA Datacom environment. There are over 30 different database utility functions that can be executed using the DBUTLTY program.

Enlarging the Buffer Specifications

During installation of CA Datacom, a default DBMSTLST module is assembled. This default assemble determines the various buffers that are allocated for use when running DBUTLTY. For the many of the important database functions performed by DBUTLTY, the buffers specified by DBMSTLST determine the performance metrics of the DBUTLTY execution, such as, elapsed time, I/Os, and CPU usage.

The default buffer allocations for DBMSTLST are quite small to keep the amount of 31-bit space needed to execute the DBUTLTY program at a minimum. However, in most sites today, there is adequate 31-bit memory available and, using these small default allocations is wasting resources and slowing performance of the DBUTLTY functions.

The following are the default buffer allocations for DBMSTLST:

```
CXXNO=10
DATANO=15
DXXNO=16
IXXNO=8
```

During various testing, we have determined that different DBUTLY functions perform better with additional buffers specifications. You can experiment with having different copies of the DBMSTLST module stored in different user libraries. Then, at DBUTLTY execution time, use the library with the DBMSTLST that performs best for that utility function.

For most sites, the multiple DBMSTLST module approach is too complicated and time consuming. While the amount of buffers needed by each DBUTLTY function varies, you should consider the following as a minimal number of buffers in DBTMLST:

```
CXXNO=10
DATANO=128
DXXNO=64
IXXNO=8
```

For sites with available space, some additional performance can be gained with the following slightly higher buffers numbers:

```
CXXNO=10
DATANO=256
DXXNO=512
IXXNO=256
```

After a new installation process, you should immediately rerun the install job that builds the DBMSTLST module and specify the expanded buffer setting you select.

Business Value

Having the additional buffer sizes when not needed does not slow the processing of the utility. However, having the additional buffers for the functions that need them could greatly improve processing time and resource consumption for the DBUTLTY.

Additional Considerations

Certain DBUTLTY functions, such as BACKUP, EXTRACT, and REORG, allow the specification of the SEQBUFS= in the DBUTLTY control cards. For these utility functions, specifying a value for SEQBUFS overrides the number of buffers specified in the DBMSTLST DATANO parameter. In z/OS sites, if the SEQBUFS is not specified, the value is set to 128. This guarantees that the BACKUP, EXTRACT, and REORG functions have at least 128 data buffers for their processing.

Overriding Buffer Specifications at Execution Time

As discussed in the previous section, the default DBUTLTY DBMSTLST module does not specify enough buffers for large DBUTLTY executions. The previous section recommended changes to the DBMSTLST settings to provide a more substantial set of buffers.

However, this approach lacks some flexibility for certain utility operations where more buffers could aid in speeding up the processing. For example, during a DBUTLTY RECOVERY=FORWARD operation having a significantly larger number of IXX, DXX, and data buffers could significantly improve the recovery time and shorten the time it takes to rebuild a database after a failure. However, for other utilities such as a normal backup or extract, these additional buffers may not be beneficial.

For these special cases, the DBUTLTY SET command can be used to override the buffer settings for a given DBUTLTY execution. Once the SET command has been issued, the selected buffer allocation remains in effect until the DBUTLTY execution ends or another SET command is encountered.

Business Value

Using the SET command can allow you to experiment with the number of buffers needed to support long-running DBUTLTY jobs. In some cases, having the ability to add additional buffers to speed recovery or other processing can be critical to sites that have experienced a database outage.

Additional Considerations

The DBUTLTY SET function uses a freeform text processor to pass various commands and settings from the JCL stream to the DBUTLTY program. Providing buffer specifications is just one of the overrides that this function can provide to the DBUTLTY program, as shown in the following example:

```
SET OPTION1='DATANO=9999;DXXNO=9999;IXXNO=999'
```

The SET function shows you how to override data buffers to 9999, DXX buffers to 9999 and IXX buffers to 999.

You need to be careful not to set the buffers numbers so high that the DBUTLTY 31-bit memory size exceeds the size of the region's available 31-bit memory. For DBUTLTY functions that also include a sort, some planning must be done to insure there is ample room for the sort's 31-bit memory requirements.

Always Specify SORT= in DBUTLTY LOAD and RETIX Functions

Many DBUTLTY functions, such as RETIX at the database level, require the SORT option to be specified. However, there are many older area level LOAD or RETIX jobs that are still running at user sites that do not have the SORT= parameter specified.

Running the functions without the SORT option can greatly reduce the performance of the utility and in some cases can create index areas that are not as efficient as they could be.

Business Value

Using the SORT= parameter on LOAD and RETIX functions improves the performance of the job execution and provides the most efficient Index Area possible.

Additional Considerations

SORT=0 is the same as not specifying the SORT parameter. Specify a sort value that is near the amount of key values (that is, keys per table times rows per table) that are processed.

If you do not know the amount of key-values specify a reasonably high number such as 999999 for small to medium tables and 999999999 for large tables.

Utilize RETIX MINIMAL to Rebuild Indexes

The DBUTLTY RETIX function is used to rebuild a disabled, that is, not loaded, or damaged Index Area. This process accesses the data rows and rebuilds each of the index entries.

Typically, index entries become disabled when the database description has been changed and a change to the keys definition, the columns (fields), was made.

Rebuilding the Index Area is typically faster than reloading the data, which includes rebuilding the Index Area. For large database indexes, this could still be a significant time consuming effort.

The CA Datacom/DB has been updated to track what key IDs in the index are disabled, not loaded, or are invalid for some reason. You can now run the DBUTLTY RETIX function with the MINIMAL=YES parameter. This tells the DBUTLTY to only rebuild the invalid index entries.

Business Value

Using the RETIX with MINIMAL=YES can significant reduce the time it takes to rebuild a damaged or invalid index and greatly reduces the time it takes to get a database back online.

Additional Considerations

With multiple data set indexes, the RETIX function needs to be executed against each of the IXX and I*nn* data sets that have a disabled key ID. RETIX MINIMAL=YES is valid for multiple data set indexes.

Chapter 6: CA Datacom AutoScope Toolset

In AutoScope, we have delivered a new set of tools for the Database Administrator. These tools provide a wide variety of functionality to assist the DBA in managing the CA Datacom environment.

The AutoInfo tool has been designed to assist the AD user in collecting and documenting information about the site's AD repository and its current configuration.

Use AutoInfo for Problem Documentation and Support Interaction

The AutoInfo function provides a simple way for you to collect all of the environmental information for a selected MUF in one simple report.

In addition to providing a report for the user, AutoInfo can also create a simple sequential data set that can be transported to CA Support. Upon reception, CA Support can convert the sequential data set into a working spreadsheet that provides the Support technician with the MUF environmental information.

AutoInfo can be used to reduce the amount of work a DBA must go through to document a given MUF environment. This is particularly useful when working with a CA Support technician on a CA Datacom issue. It can also be useful as a documentation tool, where you can save environmental information for a set of MUFs in case later research is needed.

Business Value

AutoInfo provides a simple process to quickly collect MUF environmental reducing the amount of time needed to document a support issue.

Using this functionality, you can improve the DBA's control of a given system and help to reduce their workload.

Additional Considerations

For the AutoInfo process to provide the maximum amount of information, you must have the Dynamic System Tables Database enabled.

The AutoInfo function is executed using the CA Datacom/DB DBUTLTY AUTOINFO function.

More Information:

- For more information about the CBUTLTY AUTOINFO function, see the *CA Datacom/DB DBUTLTY Guide*.

- For more information about implementation, usage, and recommended implementation scenarios, see the newly created *CA Datacom/DB AutoScope User Guide*.

Index

A

audience for this guide • 7
AutoInfo tool • 33
AutoScope toolset • 33

B

backups
    cold • 26
    hot • 26

C

CADTCM_HISTORY_ADS • 23
cold backups • 26
compatibility informaion • 27
configuring CA Datacom options • 19
configuring DBUTLTY • 29
COVERED index • 19
critical system data sets • 13
critical systems databases • 15
CXX, LXX, RXX, • 13

D

data access statistics • 23
Datadictionary • 15
DBMSTLST buffer specifications • 29
DBUTLTY configuration • 29
DBUTLTY RETIX • 32

E

enlarging Datacom LXX log file • 27
enlarging DBMSTLST buffers • 29
external database security • 13

F

finding compatibility information • 27

H

History Database • 23
hot backups • 26

I

IFL processors • 22
implementing zIIP • 22

installing with MSM • 11
interacting with Support • 33

M

Mainframe 2.0 defined • 7
memory availability • 16
memory for MUF regions • 16
MF 2.0 • 7
MRDF • 19

O

overriding DBMSTLST buffers • 30
overview of Mainframe 2.0 • 7

P

parallel TCBs • 20
preventing non-Datacom access • 13
protecting user database data sets • 16

R

rebuild indexes with RETIX • 32
RETIX MINIMAL • 32
running the SORT option • 31

S

securing CA Datacom/DB access • 13
securing Index areas • 16
shadow MUF • 26
SORT= • 31
specialty processors • 22
strategy for Mainframe 2.0 • 7
symmetrical multi-processing • 20

Z

zIIP processors • 22