

CA Database Management Solutions for DB2 for z/OS

Best Practices Guide

Version 17.0.00



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Bind Analyzer™ for DB2 for z/OS (CA Bind Analyzer)
- CA Database Analyzer™ for DB2 for z/OS (CA Database Analyzer)
- CA Detector® for DB2 for z/OS (CA Detector)
- CA Endeavor® Software Change Manager (CA Endeavor SCM)
- CA Fast Check® for DB2 for z/OS (CA Fast Check)
- CA Fast Index® for DB2 for z/OS (CA Fast Index)
- CA Fast Load for DB2 for z/OS (CA Fast Load)
- CA Fast Recover™ for DB2 for z/OS (CA Fast Recover)
- CA Fast Unload® for DB2 for z/OS (CA Fast Unload)
- CA Insight™ Database Performance Monitor for DB2 for z/OS (CA Insight DPM)
- CA Log Analyzer™ for DB2 for z/OS (CA Log Analyzer)
- CA LogCompress™ for DB2 for z/OS (CA LogCompress)
- CA Merge/Modify™ for DB2 for z/OS (CA Merge/Modify)
- CA Partition Expert™ for DB2 for z/OS (CA Partition Expert)
- CA Plan Analyzer® for DB2 for z/OS (CA Plan Analyzer)
- CA Quick Copy for DB2 for z/OS (CA Quick Copy)
- CA Rapid Reorg® for DB2 for z/OS (CA Rapid Reorg)
- CA RC/Compare™ for DB2 for z/OS (CA RC/Compare)
- CA RC/Extract™ for DB2 for z/OS (CA RC/Extract)
- CA RC/Migrator™ for DB2 for z/OS (CA RC/Migrator)
- CA RC/Query® for DB2 for z/OS (CA RC/Query)
- CA RC/Secure™ for DB2 for z/OS (CA RC/Secure)
- CA RC/Update™ for DB2 for z/OS (CA RC/Update)
- CA Recovery Analyzer™ for DB2 for z/OS (CA Recovery Analyzer)
- CA Subsystem Analyzer for DB2 for z/OS (CA Subsystem Analyzer)
- CA SYSVIEW® Performance Management (CA SYSVIEW PM)

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.

Documentation Changes

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

- [How to Identify Access Path Changes and SQL Cost Changes](#) (see page 81)—Contains an overview and cross-reference to a new user scenario that describes how to configure CA Plan Analyzer to perform compare processing, and how to explain SQL sources and identify changes to the access paths and SQL costs.

Contents

Chapter 1: Introduction	11
Chapter 2: Installation Best Practices	13
Implement a Proactive Preventive Maintenance Strategy	13
Using CA MSM and Electronic Software Delivery	15
Signing Up for Hyper Notifications	16
Registering on CA Support Website	16
Staying Current on Maintenance and Releases	17
Applying IBM Maintenance	18
Reviewing Health Check Messages	18
How to Optimize CA Insight DPM Installation Setup	19
Starting Data Collector and PC Tasks Automatically at IPL	19
Enabling the Appropriate Security Authorizations.....	19
Adjusting Dispatching Priority to Avoid Lost Trace Data.....	20
Creating an OBID Table Translation File.....	20
Allocating Adequate Trace Collection Buffer Space	20
Using One Exception Data Set for All Data Collector Tasks.....	22
Enabling Data Sharing if Monitoring a Data-Sharing Subsystem	22
Adjusting Default REGION Parameter Value According to Your Needs	23
How to Make Adjustments to Avoid CA Insight DPM Data Collector Overhead.....	23
Installing the History Component	23
Enabling the Exception Component and Optimizing Exception Settings	25
Adjusting IQL Requests if Overhead is Excessive	26
Specifying IFI=YES and IFCID2-NOTIFI=NO for Tracing Parameters	27
How to Review and Evaluate CA Insight DPM Optional Components	27
Evaluating Product Security and Limiting User Access Based on Needs.....	28
Installing the GSS Interface Only if Specifically Needed.....	28
Installing the CA SYSVIEW PM Interface	29
Installing the CA Insight DPM SCM Component for Integration to CA SYSVIEW PM	29
Refreshing the OBID Table Translation File Periodically	30
Evaluating Archive DB2 Tables	30
Installing RAF Interface for Cross-System Data Access	30
Installing VTAM Interface for Access Outside of TSO/ISPF	31
Avoiding Use of Shadow Catalog Tables to Perform EXPLAIN Analysis	31
Customizing Exceptions	31

Chapter 3: Configuration Best Practices 33

General Configuration Best Practices.....	33
Customizing Parmlib Members	33
Using the Global Defaults Facility.....	33
Database Administration Configuration Best Practices	34
Reviewing CA RC/Migrator Model Services Changes	35
Implementing Batch Processor Security Exits	36
Configuring CA Fast Unload for Optimal Use	36
Maintaining CA Fast Unload Data Integrity.....	38
Using CA Fast Unload to Automate the Creation of Test Tables.....	38
Database Performance Management Configuration Best Practices	39
Using SQLID for CA Database Analyzer Maintenance Authority	39
Generating a Consistent Number of CA Database Analyzer Maintenance Jobs	40
Implementing a Naming Convention for CA Database Analyzer Extract and Action Procedures	41
Integrate CA Plan Analyzer with Your Change Management Process	42
Database Backup and Recovery Configuration Best Practices	44
Enable CA Fast Recover Specific Parmlib Member Settings.....	44
Enable CA Merge/Modify Specific Parmlib Member Settings.....	44
Enable CA Log Analyzer Specific Parmlib Member Settings.....	45
Enable CA Quick Copy Specific Parmlib Member Settings	45
How to Enable CA Quick Copy Dynamic Allocation Parmlib Member Settings.....	47

Chapter 4: Performance Management Best Practices 51

CA Bind Analyzer Best Practices	51
Implementing CA Bind Analyzer BIND Card Modeling	51
CA Database Analyzer Best Practices	53
Retrieving Values Directly from Statistics Tables for CA Database Analyzer Action Procedures	54
Use CA Database Analyzer to Monitor the IBM RTS Tables for Missing Information	56
Use #INCLUDE Statements to Specify Longer Predicate Criteria	58
Use #INCLUDE Statements to Create New Action Conditions	60
Using Real-Time Statistics with CA Database Analyzer	63
Using a Combination of IBM RUNSTATS and CA Database Analyzer Extract Procedures	66
Using RTOS to Dynamically Select Objects for Maintenance	68
Using RTOS to Reorganize Hash PBGs	73
CA Insight DPM Best Practices	76
Using CA Insight DPM Batch Reports for System Metrics and Application Performance Monitoring	77
CA Plan Analyzer Best Practices	78
Finding Performance Problems in Programs During a Development Cycle	78
How to Identify Access Path Changes and SQL Cost Changes.....	81
CA Rapid Reorg Best Practices	81
How to Maximize DATA-AVAILABLE CRITICAL Reorgs in CA Rapid Reorg	82

How to Maximize DATA-AVAILABLE HIGH Reorgs in CA Rapid Reorg	83
How to Maximize DATA-AVAILABLE NORMAL Reorgs in CA Rapid Reorg	83

Chapter 5: Database Administration Best Practices **85**

General Database Administration Best Practices	85
How to Maximize CA Fast Index Performance	85
How to Maximize CA Fast Load Performance	86
How to Maximize CA Fast Unload Performance	87
CA RC/Migrator Best Practices	88
Reducing DB2 Catalog Contention and Overhead	88
Review Analysis Options	90
How to Use CA RC/Migrator to Retain Current DDL	90

Chapter 6: Backup and Recovery Best Practices **93**

CA Fast Check Best Practices	93
Using CA Fast Check DELETE Statements	93
How to Maximize CA Fast Check Performance	94
CA Fast Recover Best Practices	94
How to Improve CA Fast Recover Performance	95
CA Merge/Modify Best Practices	95
How to Improve CA Merge/Modify Performance	95
CA Log Analyzer Best Practices	96
How to Improve CA Log Analyzer Performance	96
Using DB2 Logs for Compliance Reporting	97
CA Quick Copy Best Practices	98
How to Maximize CA Quick Copy Performance	98
Reviewing Use of Snapshot General Services	100
Using Wildcards to Select Multiple Objects for Copy	101
CA Recovery Analyzer Best Practices	101
How to Improve CA Recovery Analyzer Performance	102
Scheduling PRA#LOAD Jobs to Execute Frequently	102

Chapter 7: Integration Points **105**

Integrate with CA Service Desk	105
Integration of CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer	106
Integration of CA Quick Copy and CA Rapid Reorg	106
Integration of CA Quick Copy and CA Merge/Modify	107
Integration of CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload	107
Integration of CA Fast Load, CA Quick Copy, and CA Rapid Reorg	108
Integration of CA Bind Analyzer, CA Detector, and CA Plan Analyzer	109

Integration of CA Fast Recover, CA Merge/Modify, and CA LogCompress	109
Integration of CA Insight DPM and CA SYSVIEW PM	110
CA RC/Query Integration.....	110

Index	113
--------------	------------

Chapter 1: Introduction

The guide introduces the CA Technologies mainframe management strategy and features, and describes the best practices for installing and configuring your product.

The intended audience of this guide is systems programmers and administrators who install, maintain, deploy, and configure your product.

Chapter 2: Installation Best Practices

The following topics provide best practices for implementing Mainframe 2.0 strategies, and installing and configuring CA Database Management Solutions for DB2 for z/OS. This release includes the following Mainframe 2.0 features:

- CA MSM and Electronic Software Delivery
- Standardized installation using SMP/E
- Integration with z/OS Health Checker

This section contains the following topics:

[Implement a Proactive Preventive Maintenance Strategy](#) (see page 13)

[Using CA MSM and Electronic Software Delivery](#) (see page 15)

[Signing Up for Hyper Notifications](#) (see page 16)

[Registering on CA Support Website](#) (see page 16)

[Staying Current on Maintenance and Releases](#) (see page 17)

[Applying IBM Maintenance](#) (see page 18)

[Reviewing Health Check Messages](#) (see page 18)

[How to Optimize CA Insight DPM Installation Setup](#) (see page 19)

[How to Make Adjustments to Avoid CA Insight DPM Data Collector Overhead](#) (see page 23)

[How to Review and Evaluate CA Insight DPM Optional Components](#) (see page 27)

Implement a Proactive Preventive Maintenance Strategy

CA Technologies formerly delivered product maintenance using Service Packs. We have replaced this model with [CA Recommended Service \(CA RS\) for z/OS](#), which provides more flexibility and granular application intervals. CA RS is patterned after the IBM preventive maintenance model, Recommended Service Upgrade (RSU). With CA RS, you can install preventive maintenance for most CA Technologies z/OS-based products in a consistent way on a schedule that you select (for example, monthly, quarterly, annually).

We recommend that you develop and implement a proactive preventive maintenance strategy whereby you regularly apply maintenance. You could follow the same schedule that you use to apply IBM maintenance, or you could implement a schedule for CA Technologies products only.

Business Value:

Keeping your products current with maintenance helps your team remain productive and minimize errors while safely protecting your systems. If you do not install preventive maintenance regularly, you risk encountering known problems for which we have published and tested fixes.

Our mainframe maintenance philosophy is predicated upon granting you the flexibility to maintain your sites and systems consistent with industry best practices and site-specific requirements. Our philosophy focuses on two maintenance types. Understanding each type can help you maintain your systems in the most efficient manner.

Note: This philosophy applies to the [CA Next-Generation Mainframe Management stack products](#). For legacy products, contact CA Support for maintenance details.

Corrective Maintenance

Helps you address a specific and immediate issue. This type of maintenance is necessary after you encounter a problem. We may provide a test APAR when a new problem is uncovered, or a confirmed PTF when the problem has been resolved. Your primary goal is to return your system to the same functional state that it was before you experienced the issue. This type of maintenance is applied on an as-needed basis.

Preventive Maintenance

Lets you apply PTFs that we have created and made public. You may have experienced the issues that each PTF addresses. CA RS provides a way to identify all published maintenance that has been successfully integration-tested. This maintenance has been tested with other CA Technologies products, current z/OS releases, and IBM subsystems, such as CICS and DB2. Major CA RS service levels are published quarterly, with updates for HIPER and PE-resolving PTFs that are published monthly. After you test the CA RS level, we recommend that you accept that level before you apply a new CA RS level.

You can initiate a maintenance installation activity at any time. You can then install the current CA RS level of maintenance (recommended) or an earlier level. Additionally, you can install maintenance to support a new hardware device, software upgrade, or function using our [FIXCAT](#) method.

For all maintenance, *before* you initiate any maintenance action, obtain the current SMP/E HOLDDATA.

Important! [CA Chorus™ Software Manager \(CA MSM\)](#) - formerly known as CA Mainframe Software Manager™ (CA MSM) - is an intuitive web-based tool that can automate and simplify many CA Technologies product installation and maintenance activities. We strongly recommend that you use CA MSM to maintain your CA Technologies z/OS-based products.

More Information:

To apply preventive maintenance using CA MSM or from CA Support Online on <http://ca.com/support>, see the *Installation Guide* for your product and the CA MSM online help.

Using CA MSM and Electronic Software Delivery

CA MSM provides a web interface, which works with Electronic Software Delivery (ESD) and standardized installation, to provide a common way to manage CA mainframe products. Use CA MSM and ESD to acquire, install, and maintain the CA Database Management Solutions for DB2 for z/OS.

Business Value:

CA MSM lets you download product and maintenance releases over the Internet directly to your system from the CA Support website. The same interface is used to install the downloaded software packages using SMP/E.

When you order the CA Database Management Solutions for DB2 for z/OS, you receive the authorizations and instructions to access, download, and prepare the installation files without the need for a physical tape. Maintenance can be applied as needed without waiting to obtain a special tape or cartridge.

Additional Considerations:

Downloaded files are delivered in a new pax.Z format that eliminates the need to create an installation cartridge from an ESD image. You can download the pax.Z file from the CA Support website.

To download the files and create the installation data sets, perform the following steps:

1. Review the instructions for installing from DASD provided in the *CA Enhanced Electronic Software Delivery Pax Enhanced ESD User Guide* before proceeding with the remaining steps. This guide is available from the following link:

<https://support.ca.com/phpdocs/7/common/TEC342790.pdf>

2. Prepare your mainframe environment to receive the pax.Z file.

Important! This step requires a UNIX System Services (USS) directory to receive the PAX file and to perform the unpack steps. Verify that all users working with ESD pax files have writer authority to the directory and that enough space has been allocated to receive the file *before* you start the download.

Note: For additional ESD information, go to the mainframe web page on the CA website. Under Events, we offer an ESD webcast to explain the enhanced ESD process.

3. Download the pax.Z file (include your USS directory and TCP/IP profile data set in the JCL).
4. Unpack the ESD pax file and copy the ESD installation files to z/OS data sets.
5. Proceed with the product installation as described in the *CA Database Management Solutions for DB2 for z/OS Installation Guide*.

To install a new release of the products or apply maintenance (application of all or selective fixes within the same release), you must perform SMP/E RECEIVE, APPLY, and ACCEPT processing through native SMP/E.

Note: If you used ESD to install a new or maintenance release of the CA Database Management Solutions for DB2 for z/OS, you must also specify the DASD high-level qualifier to use the files obtained from ESD in the installation process. Detailed information is provided in the *CA Database Management Solutions for DB2 for z/OS Installation Guide*.

More Information:

For more information about CA MSM, see the *CA MSM Product Guide* or *CA MSM Readme*.

Signing Up for Hyper Notifications

CA Support is always alert to any known problems that could affect customers. We recommend that you sign up for hyper notifications for all CA Technologies products on the CA Support website so that you receive notification of potential problems as soon as possible. Select Hyper notifications and follow that dialog to receive notifications for your products.

Business Value:

Early notification about the potential need to fix the CA Database Management Solutions for DB2 for z/OS helps avoid unscheduled downtime and assist you in proactively maintaining your software.

Registering on CA Support Website

Register your database team members on the CA Support website. The CA Support web pages contain not only fixes, but also valuable information, technical articles, and recommendations regarding CA Technologies products.

Go to <http://ca.com/support> and register your email address and the CA Database Management Solutions for DB2 for z/OS that you have.

Business Value:

The CA Support website has guides available for download and searchable knowledge forums:

- Knowledge Base Search
- Support By Product or Solution

- Download Center
- Licensing
- CA Tech Insider Subscription Compatibilities

CA Support users are able to review tickets (both open and closed), search the knowledge database, and download copies of guides. This access gives them a significant productivity boost, without having to disturb other employees with questions. Programmers and DBAs can be more productive by being able to get questions answered quickly by searching knowledge bases and communicating with CA Support on any open issues.

Learning how to navigate the CA Support website can help you become more knowledgeable about CA Technologies products. Greater product knowledge helps you get more value out of them. You can benefit from shorter time to resolution when other users report problems on the CA Support website, and better quality assurance when upgrading to new software.

Reviewing technical documents for your products can unearth previously unused functionality or can provide enhancements to the usage of known functionality. The technical documents can also describe ways of using the products that had not been considered and could be beneficial.

Regular monitoring of the CA Support website verifies that you are aware of the latest software changes and information about upgrade functionality that meets future requirements.

Staying Current on Maintenance and Releases

Download and install all current maintenance that is available. Electronic Software Delivery (ESD) and service pack “buckets” are available on <http://ca.com/support>. These maintenance buckets are updated on a weekly basis.

Business Value:

Being current on release levels can help you take advantage of new product functionality sooner and increase usability and application capability, which can help you achieve higher productivity. By keeping maintenance as current as possible, you will not encounter problems identified by CA Quality Assurance or other customers, and will receive new released features.

Applying IBM Maintenance

We recommend that you review the current list of IBM APARs and apply the APARs that are appropriate to your environment. To obtain a current list of IBM APARs, contact CA Support at <http://ca.com/support>.

Business Value:

Missing APARs can affect the operation or performance of the CA Database Management Solutions for DB2 for z/OS.

Reviewing Health Check Messages

The CA Database Management Solutions for DB2 for z/OS are now integrated with the IBM Health Checker for z/OS through the CA Health Checker Common Service to verify that deployed software follows our best practices. Review the messages generated from the health checks to identify potential problems automatically.

Business Value:

The health checks continually monitor the system and software to provide feedback on whether the software continues to be configured optimally.

Additional Considerations:

When one or more subtasks contain errors, the health check generates an exception error that assigns a severity level to the exception and lists the abending tasks. This message provides an explanation, system action, operator response, system programmer response, problem determination, source, and reference documentation.

Additional messages are also generated to help diagnose and correct the condition.

More Information:

For more information about the messages generated from the health checks, see the *CA Database Management Solutions for DB2 for z/OS Message Reference Guides*. For more information about product-specific health checks, see the product documentation.

How to Optimize CA Insight DPM Installation Setup

We recommend the following activities regarding CA Insight DPM installation setup:

- [Start data collector and PC tasks automatically at IPL](#) (see page 19).
- [Enable the appropriate security authorizations](#) (see page 19).
- [Adjust dispatching priority to avoid lost trace data](#) (see page 20).
- [Create an OBID table translation file](#) (see page 20).
- [Allocate adequate trace collection buffer space](#) (see page 20).
- [Use one exception data set for all data collector tasks](#) (see page 22).
- [Enable data sharing if monitoring a data-sharing subsystem](#) (see page 22).
- [Adjust the default REGION parameter value according to your needs](#) (see page 23).

Starting Data Collector and PC Tasks Automatically at IPL

We recommend setting up the data collector and PC tasks as started tasks that start automatically upon IPL, to allow uninterrupted monitoring of the DB2 subsystem.

Note: The data collector task must remain active at all times, unless circumstances require that the task be brought down or restarted.

Business Value:

By starting at IPL, the CA Insight DPM data collector can monitor the DB2 subsystem and report problems as soon as the DB2 subsystem becomes available.

Enabling the Appropriate Security Authorizations

Verify that the following security-related authorizations are in place for the user ID associated with the data collector task:

- The user ID has the appropriate authority to update the exception VSAM data set and the online history VSAM data sets.
- The user ID minimally has TRACE, MONITOR1, and MONITOR2 DB2 privileges.

Business Value:

CA Insight DPM needs the proper DB2 authorizations to interact with DB2 and start the appropriate DB2 traces for monitoring the DB2 subsystem.

Adjusting Dispatching Priority to Avoid Lost Trace Data

After the started tasks are defined, set the dispatching priority for the data collector task just below the IRLM dispatching priority and above the DB2 address spaces and DB2 applications. The dispatching priority must be high enough to minimize the chance of lost DB2 trace data.

Business Value:

This practice minimizes the risk of losing performance data that might be critical in resolving DB2 performance problems.

Creating an OBID Table Translation File

Create an OBID translation file during the installation process. CA Insight DPM uses this file to translate pageset, index, and table OBIDs into the actual object names.

Business Value:

Collected data is more meaningful to your business. This practice allows CA Insight DPM online displays to show object names rather than DBID/OBID numbers.

Allocating Adequate Trace Collection Buffer Space

In addition to having a high-enough dispatching priority, the data collector must have a sufficient amount of trace buffer space to collect trace data. Otherwise, the buffers overflow and trace data is lost.

Two sets of buffers are involved:

- Buffers located in CSA
 - DB2 (IFI or service trace exit) uses these buffers to pass trace data to the data collector.
- Buffers residing in the data collector private area
 - These buffers flush the CSA buffers and process the records.

The following data collector SYSPARM options control these buffers:

SRVBUFSZ and SRVBUFNUM

Control the size of the CSA trace buffers.

PRVBUFNUM and PRVMAXSZ

Control the size of the private area buffers.

Set your parameters based on the following:

- Type of DB2 subsystem being monitored
- Expected trace workload (such as the volume of trace data)
- Current system paging rate

Higher buffer sizes reduce the chance of lost trace data but increase the demand on real, common, and virtual storage.

On higher-volume systems, we recommend the following settings (to start with a CSA buffer allocation of 4 MB):

- SRVBUFSZ=8
- SRVBUFNUM=128
- PRVBUFNUM=128
- PRVMAXSZ=4096

On a low-volume system, use the following default values:

- SRVBUFSZ=8
- SRVBUFNUM=8
- PRVBUFNUM=12
- PRVMAXSZ=1024

Note: You can use the INS IFI command from the user interface to determine the current CSA (IFI monitor size) and private buffer sizes in use. This command also shows the number of trace records that have been lost due to unavailable buffers. If records are being lost, and the dispatching priority of the data collector is set appropriately, increase these values to reduce the number of lost records.

Business Value:

Allocating sufficient trace buffer space minimizes the risk of losing performance data that might be critical in resolving DB2 performance problems.

Using One Exception Data Set for All Data Collector Tasks

When performing CA Insight DPM post-installation processing tasks, you can create a *common* exception data set or a *unique* exception data set for each data collector. We recommend sharing a single common exception data set between all data collector tasks. In the common exception data set, you can do the following:

- Define unique exceptions by DB2 subsystem name, if needed. Only those DB2 subsystems that qualify by name will use these exceptions.
- Define customized exceptions for production systems as opposed to test or development systems.

Thus, in the common file, you can customize exceptions by subsystem name or type.

Business Value:

Using a common exception data set reduces the complexity of managing exceptions by allowing you to have a single set of exceptions to manage multiple DB2 subsystems.

Enabling Data Sharing if Monitoring a Data-Sharing Subsystem

CA Insight DPM has a limited number of displays to combine data from members of a data-sharing group. If the data collector is monitoring a data-sharing DB2 subsystem, we recommend enabling the data-sharing option.

Business Value:

Enabling this data sharing provides additional monitoring capabilities for data-sharing environments.

More Information:

For more information about data-sharing implementation, see the *CA Insight DPM System Reference Guide*.

Adjusting Default REGION Parameter Value According to Your Needs

CA Insight DPM provides a SYSPARM to override the default REGION size and REGION limit. If your system is storage-constrained, you can reduce the value to reduce system paging. However, we do *not* recommend setting a region size below 128 MB. We recommend leaving the value at 128 MB or increasing the value to help ensure that the data collector does not run out of virtual storage (particularly in environments where multiple application traces are running).

Business Value:

Overriding the default REGION size when needed lets you collect performance trace data based on your business needs and storage constraints.

How to Make Adjustments to Avoid CA Insight DPM Data Collector Overhead

If collection overhead is a major concern in CA Insight DPM, evaluate the following activities:

- [Install the history component](#) (see page 23).
- [Enable the exception component and optimize exception settings](#) (see page 25).
- [Adjust IQL requests if overhead is excessive](#) (see page 26).
- [Specify IFI=YES and IFCID2-NOTIFI=NO for tracing parameters](#) (see page 27).

Installing the History Component

We recommend installing the optional history component of CA Insight DPM. The history component lets you view past application and subsystem performance data, which is collected and stored in VSAM history files.

Business Value:

The history component enables the monitoring of DB2 applications and subsystem to help you understand past performance problems.

Additional Considerations:

Consider the following recommendations if you install the history component:

- Pay attention to how the file is allocated (size and secondary extents) during the initial installation process to avoid needing to redefine history and copy history files:
 - Verify that the file is properly sized (primary allocation) for the amount of data to be collected.

Note: For information about working with the Online History component (including adjusting allocations for the Online History data sets and using tables/formulas to estimate the proper size for history files), see the *CA Insight DPM System Reference Guide*.

- We recommend *not* defining history files with a secondary extents specification. When the history file is initialized, CA Insight DPM initializes the entire file, which causes the data set to extend into as many extents as possible. Use secondary extents only if you cannot allocate enough space in a primary allocation.
- Filter the history data (using the HISTORY-ACCT=FILTER SYSPARM option) when implementing history on systems that generate a high volume of accounting records. Filtering the records lets you specify which history records to keep in the history file for subsequent online viewing and which records to discard. Trying to store all records on a high-volume system can cause the data on the history file to wrap too quickly and generate unnecessary CPU overhead. You can enable filtering in the HISTORY-ACCT SYSPARM option and customize the FILTERAC IQL request to generate a simple IQL WHERE clause that ignores unwanted records. We recommend using a WHERE clause similar to the following:

```
WHERE TOTAL-TIME-DB2 > .5
```

This clause causes CA Insight DPM to store records that have an “in DB2” elapsed time greater than ½ second. You can set the time to a value that makes sense for your applications and volume of data.

- Enable hardware compression for the history data. Compression lets you store, on average, more than twice the data of uncompressed records, for little CPU overhead. When the data collector closes the history files during termination, it indicates the amount of CPU overhead used to compress data. If CPU usage is of primary concern, and you find that this CPU overhead is too much, you can disable compression. However, disabling compression causes more data to be written to the history files, thus eliminating some CPU savings.
- For high-volume systems, use an accounting trace setting for the ACCTTRACE SYSPARM option that is consistent with the default accounting trace classes that are started and written to SMF. For low-volume and development systems, we recommend a value of 8 or higher.

- Accept the default of starting all optional IFCIDs for the HIST-STATS-RECS SYSPARM option. These IFCIDs do not generate much overhead. If CPU overhead is of primary concern, and you have disabled data set statistics for your DB2 subsystem, remove 199 from this option to disable collection of IFCID 199 records.
- If CPU overhead is of primary concern, disable the collection of any additional thread data (through the HIST-ACCT-DETAIL SYSPARM option). Otherwise, enable the PDTSQL setting if CA Detector is installed. Evaluate all other settings for this SYSPARM option on an as-needed basis with regard to the overhead that each setting can generate.

More Information:

For information about estimating the size of history files, see the *CA Insight DPM System Reference Guide*.

Enabling the Exception Component and Optimizing Exception Settings

We recommend accepting the default of having the exception component enabled. If CPU overhead is of primary concern, try the following recommendations to minimize the overhead of the exception component before disabling the entire component:

- Set the overall exception interval to a value, such as 5 to 15 seconds, that provides timely detection of exceptions. However, if CPU overhead is of concern, you can set a higher value. The default is 10 seconds.

Note: We do *not* recommend a value higher than 60 seconds.

- If a large number of pagesets are allocated to DB2, database exceptions can generate a temporary spike in CPU usage during the database exception interval. Set the exception database interval EXCEPTION-DB-INTERVAL SYSPARM option to 900 seconds (15 minutes). If CPU overhead is of primary concern, consider increasing this value with regard to how quickly you want to be notified of a database or pageset problem.
- After the data collector is started, evaluate the application and database exceptions and disable any exceptions that are not used or needed. In general, subsystem exceptions generate little overhead. If there is a larger number of concurrent threads or a high volume of threads, limit the application exceptions to those that are absolutely required. If many pagesets are allocated to DB2, disable any database exceptions that are not beneficial. You can view the overhead of the exception system by using the MVSTASKS display for the data collector task to view the CPU used by the AMXSSXPE task. Try enabling and disabling different exceptions to see how CPU overhead is affected, until it is at an acceptable level.

Business Value:

Enabling exceptions allows CA Insight DPM to automatically monitor DB2 and alert you about performance problems. Customizing the exceptions allows the exception mechanism to execute more efficiently and reduce CPU and storage overhead.

Adjusting IQL Requests if Overhead is Excessive

Most IQL requests that are started by default generate little overhead. In some instances, depending on the type of applications, the following IQL requests can generate significant overhead. If CPU overhead of the data collector is of concern, evaluate these IQL requests and edit or remove them from the STARTUP list if needed.

- SORTSUM

This IQL request collects *begin* and *end* data for all sorts performed on behalf of SQL. SORTSUM is most beneficial for transaction-based applications using static SQL. If there is a high rate of ad hoc or dynamic SQL using sorts, we recommend removing this IQL request from the STARTUP list. If you remove this request, you must also remove the SORT-ENDED section of the APPEXCP IQL request.

- CONNACTV/CORRACTV/PLANHIST/THRDHIST

In a high-volume DB2 subsystem, we recommend evaluating these IQL requests and removing them from the STARTUP list if they are not useful. For these requests, the data collector accumulates data for each accounting record produced, which can generate significant overhead.

Business Value:

Adjusting IQL requests can reduce data collector CPU and storage overhead.

Specifying IFI=YES and IFCID2-NOTIFI=NO for Tracing Parameters

We recommend the following tracing parameter settings in the SYSPARMS file:

- IFI=YES

This setting specifies how the data collector retrieves data from DB2.

Note: If CPU overhead is of primary concern, consider changing the option to IFI=READS or IFI=NO. However, if IFI=YES is not used, data can be limited or unavailable.

- IFCID2-NOTIFI=NO

This setting tells the data collector to get IFCID 2 records through the IFI.

Note: If data collector CPU usage is a problem, you can set this value to YES to see if data collector CPU usage is affected.

Business Value:

These options allow for less data collector CPU overhead in constrained environments (at the expense of some loss of functionality).

More Information:

For a description of each setting, see the *CA Insight DPM System Reference Guide*.

How to Review and Evaluate CA Insight DPM Optional Components

Review the following CA Insight DPM optional component installation considerations:

- [Evaluate product security and limit user access based on needs](#) (see page 28).
- [Install GSS interface only if needed](#) (see page 28).
- [Install the CA SYSVIEW interface](#) (see page 29).
- [Install the System Condition Monitor for CA SYSVIEW PM](#) (see page 29).
- [Refresh the OBID table translation file periodically](#) (see page 30).
- [Evaluate archive DB2 tables](#) (see page 30).
- [Install the RAF interface for cross-system data access](#) (see page 30).
- [Install the VTAM interface for access outside of TSO/ISPF](#) (see page 31).
- [Do not use shadow catalog tables to perform EXPLAIN analysis](#) (see page 31).
- [Customize exceptions](#) (see page 31).

Evaluating Product Security and Limiting User Access Based on Needs

We recommend evaluating product security and limiting the different CA Insight DPM features to users that require access to those features.

Business Value:

Limiting access based on needs prevents users from interfering with data or functions for which they should not be authorized.

More Information:

For details about the security options and how to implement security, see the *CA Insight DPM System Reference Guide*.

Installing the GSS Interface Only if Specifically Needed

We recommend installing the Global Subsystem (GSS) if you need one or more of the following features:

- Collection, logging, and playback of DB2 console messages
- Logging and playback of exception messages
- Logging and playback of DB2 utility phase change messages
- Logging and playback of DB2 commands executed
- Invocation of REXX automated operations routines from IQL requests and exceptions
- Installation and use of the System Condition Monitor

Business Value:

The GSS provides additional information and functionality. However, if you do not need its functionality, you do not need to install it and can avoid the additional installation time and effort.

More Information:

For information about enabling the GSS user interface, see the *CA Insight DPM System Reference Guide*.

Installing the CA SYSVIEW PM Interface

If you are a licensed user of CA SYSVIEW PM, we recommend installing the CA SYSVIEW PM interface.

Business Value:

Installing the CA SYSVIEW PM interface allows CA SYSVIEW PM users to access CA Insight DPM from within CA SYSVIEW PM. This avoids the need for a separate terminal session to monitor DB2 and avoids the extra process of moving from one product to another product.

More Information:

For information about enabling the CA SYSVIEW PM user interface, see the *CA Insight DPM System Reference Guide*.

Installing the CA Insight DPM SCM Component for Integration to CA SYSVIEW PM

If you have installed the System Condition Monitor for CA SYSVIEW PM, we recommend installing the System Condition Monitor (SCM) component of CA Insight DPM. This feature lets CA Insight DPM pass information about the state of DB2 to CA SYSVIEW PM.

Business Value:

This feature allows the CA Insight DPM user to monitor the status of many DB2 subsystems across systems from a single point of interaction.

More Information:

For information about installing the SCM, see the *CA Insight DPM System Reference Guide*.

Refreshing the OBID Table Translation File Periodically

We recommend setting up a batch job that periodically updates the OBID translation file and refreshes the file in the data collector. We recommend the following refresh intervals:

- Once per day for development or test systems where objects are frequently created and dropped
- Once per week for production systems where objects do not change much

Business Value:

As objects in DB2 are created and dropped, this setup allows CA Insight DPM online displays to show current object names rather than obsolete names or DBID/OBID numbers.

Evaluating Archive DB2 Tables

We recommend evaluating the need to maintain DB2 performance data in DB2 tables. CA Insight DPM provides a facility for storing thread accounting and subsystem statistics data in DB2 tables that can be used for customized querying and reporting of DB2 performance data.

Business Value:

Long-term reporting is useful for performing trend analysis to see how the performance of the DB2 subsystem and its applications is changing over time. Customized reporting allows you to perform detailed performance analysis customized to your specific needs and for specific performance questions or problems that arise.

Installing RAF Interface for Cross-System Data Access

If you need to access data collectors across systems, we recommend installing the remote access facility (RAF) interface. The RAF interface uses VTAM connections, so you can access DB2 performance data from any data collector that has RAF enabled.

Business Value:

Installing the RAF interface lets you access data collectors across systems without having to connect locally for each system.

Installing VTAM Interface for Access Outside of TSO/ISPF

If you need to access the CA Insight DPM user interface outside of the TSO/ISPF environment, or if you want to use the CA SYSVIEW PM interface, we recommend installing the VTAM user interface.

Business Value:

Installing the VTAM user interface lets you monitor DB2 subsystems without tying up a TSO/ISPF session.

Avoiding Use of Shadow Catalog Tables to Perform EXPLAIN Analysis

We do *not* recommend creating and maintaining shadow tables of the DB2 catalog solely for use with the EXPLAIN component of CA Insight DPM. If you have shadow tables already installed for other usage and have a system for maintaining currency, you can use these shadow tables to perform dynamic EXPLAIN analysis. If catalog contention is not a problem, we recommend using the real DB2 catalog tables to perform EXPLAIN analysis.

Business Value:

By not using shadow catalog tables to perform EXPLAIN analysis, you avoid unnecessary maintenance and avoid the overhead of shadow tables.

Customizing Exceptions

After you install a new data collector, we recommend using the Exception Monitor option on the CA Insight DPM Main Menu to view exceptions and recommend using the U line command to customize exception threshold values as needed so that exceptions are detected only when desired.

We also recommend using the Permanent Exception Definitions option (on the Main Menu) to review all subsystem exception definitions, evaluate the threshold values for each definition, and disable exceptions that are not needed.

Business Value:

Performing this customization on exceptions avoids unnecessary alerts from the exception system and allows only true alerts to be displayed.

Chapter 3: Configuration Best Practices

This section contains the following topics:

[General Configuration Best Practices](#) (see page 33)

[Database Administration Configuration Best Practices](#) (see page 34)

[Database Performance Management Configuration Best Practices](#) (see page 39)

[Database Backup and Recovery Configuration Best Practices](#) (see page 44)

General Configuration Best Practices

General configuration best practices include the following:

- Customize parmlib members.
- Use the global defaults facility.

Customizing Parmlib Members

Most of the CA Database Management Solutions for DB2 for z/OS have a parmlib member that you can customize to set default product use values at your site. We recommend that you set these values as soon as possible after installation so that users have these default values set when they use the products. You can edit the members in *high-level.CDBAPARM* or by specifying EP on the CA Database Management Solutions for DB2 for z/OS Main Menu, and then selecting the member.

Business Value:

Customizing your parmlib members helps take advantage of your site-specific requirements, avoid having to specify overly complex syntax, and optimize the execution parameters to your environment.

Using the Global Defaults Facility

Use the Global Defaults facility to specify defaults for variables that are stored in the ISPF profiles on DB2 subsystems. These defaults can be applied to individual users or groups of users. The default values are saved inside a DB2 table on each subsystem. To use the global defaults facility, set DEFAULTS to YES in the SETUPxx global parmlib member. The default values can then be customized using the GBLDEF command on the CA Database Management Solutions for DB2 for z/OS Main Menu.

Business Value:

The Global Defaults facility lets you specify different defaults on different subsystems. For example, if you use the Interactive SQL (ISQL) facility, you can specify that SQL changes on a test subsystem are committed, but that SQL changes on a production subsystem are rolled back.

More Information:

For more information about using GBLDEF to set global defaults, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference Guide*.

Database Administration Configuration Best Practices

Database administration configuration best practices include the following:

- Reviewing CA RC/Migrator Model Services changes
- Implementing Batch Processor security exits
- Configuring CA Fast Unload as follows:
 - For optimal use
 - To maintain data integrity
 - Automate the creation of test tables

Reviewing CA RC/Migrator Model Services Changes

If you have upgraded to a new release, review the changes that have been made to the default model utility definitions and apply the differences to your customized models to bring them up to date with the new release.

Note: You must execute the RCM Load and Customize the Model Services task during post-install processing to load and refresh the model services table for the new release and create a new @DEFAULT utility model.

Business Value:

Keeping your model utility definitions current helps ensure that you are making use of any new functionality added to the definitions.

More Information:

For more information about the RCM Load and Customize the Model Services task, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide*. For more information about Utility Model Services and how to identify changes that have been made to the @DEFAULT model in a new release, see the *CA RC/Migrator User Guide*.

Implementing Batch Processor Security Exits

A security exit is called for every Batch Processor command that provides access or authority to a resource. We recommend that you implement the Batch Processor security exits to control command execution using the Batch Processor interface. To implement the security exits, edit the sample members provided in *high-level.CDBASRC* or create your own, and then assemble and link the source code.

Note: The delivered EXIT01 exit requires customization to permit use of the .AUTH command (to switch authorization IDs for connections to DB2). As delivered, EXIT01 uses a SAF call to your external security system to verify the user's authorization to switch to the specified ID. You need to set up external security definitions for your site to permit use of the .AUTH command.

Business Value:

EXIT01 (.AUTH command) is very important because of its powerful nature. By implementing EXIT01 and the other security exits, you can gain desired control over the use of commands that provide access or authority to resources.

More Information:

For more information about the .AUTH command, as well as setting up and implementing the Batch Processor security exits, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide* and the *Batch Processor Reference Guide*.

Configuring CA Fast Unload for Optimal Use

To help ensure that CA Fast Unload meets the requirements of occasional users, review the following CA Fast Unload parameters and consider specifying these settings in the PFU member of *high-level.CDBAPARM* to minimize the choices that typical users must make in their jobs:

- Specify SORTNUM, SORTDEVT, or both to enable sort to dynamically allocate sort workspace. (The DYNALLOC sort parameter is not generated unless one of these values has been specified.) The SORTSIZE, SORTSIZE-BELOW (or SORT-APPSIZE), and SORTFLAG commands are also available for use in processing ORDER BY within SELECT statements.
- Specify EXTENSION for SQL-ACCESS to override the default value of NONE. Non-SQL access is used wherever possible for improved speed, but more complex SELECT statements are still processed without intervention by changing over to SQL access when necessary.

- Specify ASCII-OUT EBCDIC in the PFU parmlib member if you want to mimic the old data translation behavior in CA Fast Unload, which unloads ASCII table character data as EBCDIC. If you do not have time to review your CA Fast Unload jobs and modify them based on the new data translation behavior of CA Fast Unload, which unloads character data in the encoding scheme of the table, this setting can be used to ease the transition from the old to new behavior and lets you continue current processing until you have time to modify your jobs. This setting is *not* recommended as a permanent solution.
- Use DISPLAY-STATUS in one of its forms to monitor CA Fast Unload execution progress.
- Use PRINT-OPTIONS PTIMSG or the ALLMSG command to review the options that are in effect for a given CA Fast Unload execution.
- Change the EMPTY-RC default value if you do not want to have a CA Fast Unload execution end with return code 4 when a SELECT statement returns no rows. This setting is especially important if you often have tables that are completely empty.

Business Value:

Reviewing your default settings and specifying values that are appropriate for most jobs makes CA Fast Unload easier to use for occasional and experienced users. Specifying global options lets CA Fast Unload execute in a designated way site-wide yet still let users customize at the individual CA Fast Unload execution level by entering commands in the SYSIN stream.

Additional Considerations:

CA Fast Unload supports many other global options that provide various installation-level tailoring choices. Most commands in the PFU parmlib member can also be specified in the SYSIN stream (where the SYSIN values override the parmlib values). However, many commands can be specified only in the SYSIN stream.

More Information:

For more information about CA Fast Unload keywords and global options, see the *CA Fast Unload User Guide*.

Maintaining CA Fast Unload Data Integrity

To help ensure CA Fast Unload data integrity, use SHRLEVEL REFERENCE for SQL-ACCESS NONE or EXTENSION, or consider using INPUT-FORMAT IMAGECOPY, DSN1COPY, or CONCURRENTCOPY. Reserve the use of SHRLEVEL CHANGE (with or without BP-LOOKUP) for times when absolute data integrity is *not* a requirement.

Note: SHRLEVEL CHANGE, in all its forms, is a "dirty read." As an alternative, you can consider unloading the data from some form of image copy of the tablespace in question.

Business Value:

Maintaining CA Fast Unload data integrity is important where operational systems depend on the completeness and accuracy of the unloaded data. For situations with less strict requirements, using SHRLEVEL CHANGE (in its various forms) provides less impact on data availability.

More Information:

For more information about this syntax, see the *CA Fast Unload User Guide*.

Using CA Fast Unload to Automate the Creation of Test Tables

You can automate the creation of test tables by using CA Fast Unload to unload data from production tables into test tables. CA Fast Unload can create single tables in the same or different DB2 subsystems. These tables can contain some or all columns from the original table, with added column data initialized based on various criteria. For example, in your SELECT statement you can specify the literals and columns to unload, and then use field specifications in the INTO clause to specify alternate column formatting for the unloaded data. You can mask sensitive client data by including one or more DEFAULTIF (*condition*) INITIAL (*value*) clauses for each column. The LIMIT keyword lets you limit the total number of rows to unload. The SAMPLE keyword lets you unload a physical subset of the rows from the source table. (For example, you can unload two rows, skip the next 198 rows, unload another two rows, skip the next 198 rows, and so on.)

See the following sample SELECT statement:

```
SELECT 'fred', column_1, column_2, 'abc' INTO AAA CHAR(4), BBB VARCHAR(12)
DEFAULTIF(column_1 = ' ') INITIAL('ZZZZZZZZZZZZ') DEFAULTIF(column_1 = 'abc')
INITIAL('BBB'), CCC, DDD VARCHAR(5)
```

When you submit the CA Fast Unload job to unload the specified data, you can also automatically generate CREATE TABLE DDL and load control statements by including the DDL-CONTROL and LOAD-CONTROL keywords. You can then use CA Fast Load (or another load utility) to load the selected data into a new table.

Note: If the SELECT statement is complex enough to require SQL-ACCESS ONLY processing, creation of all control statements may not be possible. If you require control statement generation, we recommend limiting complex SELECT syntax that requires SQL-ACCESS ONLY processing.

Business Value:

Using existing CA Fast Unload capabilities to automate the creation of test tables eliminates the need to create these tables manually or develop a custom solution, which reduces the potential for error. CA Fast Unload has powerful selection and manipulation logic that provides great flexibility in data selection and potentially provides faster processing than a manual or custom solution.

More Information:

For more information about the CA Fast Unload syntax discussed in this best practice, see the *CA Fast Unload User Guide*.

Database Performance Management Configuration Best Practices

Database performance management configuration best practices include the following:

- Configuring CA Database Analyzer to do the following:
 - Use SQLID for maintenance authority
 - Generate a consistent number of maintenance jobs
 - Implement naming conventions for extract and action procedures
- Integrating CA Plan Analyzer with your change management process

Using SQLID for CA Database Analyzer Maintenance Authority

To maintain CA Database Analyzer procedures (extract, action, and so on) with greater ease and flexibility, specify SQLID for the DEF_PROC_CREATOR parameter in the PDA member of *high-level.CDBAPARM* (or for the Default Procedure Creator field in the Edit Parmlib Member panel). This specification assigns an SQLID value instead of a specific TSO ID as the authority ID (Default Procedure Creator value) to create, view, or update procedures.

Business Value:

If a group of users is responsible for a set of CA Database Analyzer procedures, creating a shared esoteric SQLID (such as PAYADMIN) lets other group members maintain the procedures when one member is unavailable (for example, on vacation).

Additional Considerations:

To convert existing procedure creators to an SQLID, run the SQL in the *high-level.CDBASRC(PDASQL7)* library.

Important! Before you run the SQL, back up all tables that the SQL statements will change, so that you can recover the previous table data if the SQL changes do not provide the expected results.

After you convert the procedure creators to an SQLID, you can specify N for the ALLOW_DPC_PROFILE_OVERRIDE parameter in the PDA parmlib member (or for the Allow DPC to be overridden in PROFILE field in the Edit Parmlib Member panel) to force all users to use the new Default Procedure Creator. Specifying N prevents individual users from controlling the creator type in their CA Database Analyzer profiles.

More Information:

For more information about editing the PDA parmlib member, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide*. For more information about editing the CA Database Analyzer profile values, see the *CA Database Analyzer User Guide*.

Generating a Consistent Number of CA Database Analyzer Maintenance Jobs

Many users tie CA Database Analyzer extract procedures to action procedures to generate action JCL for maintenance tasks such as creating image copies or reorganizing tablespaces. These procedures can select hundreds of database objects and put them in multiple z/OS jobs. Users generally use job schedulers to control these production jobs, and need a way to consistently generate jobs through a CA Database Analyzer extract.

When you create or update an extract procedure, use the Job Generation Parm panel to control how jobs are generated. Specifically, use the Number of Jobs (NOJ) field to specify the number of jobs to generate each time the extract is executed. You can then set up your job scheduler to expect a specific number of jobs.

Business Value:

Generating a consistent number of maintenance action JCL jobs simplifies setup, because it lets the job scheduler reference a specific number of jobs.

Additional Considerations:

For CA Database Analyzer extracts that reorganize objects or collect statistics to update the DB2 catalog and also generate REBIND steps, specify REBIND_JOB (UNIQUE) in the PDA parmlib member. This value places the REBIND step in a separate job that is always generated last. The job scheduler can then set up dependencies for the REBIND job to run after the preceding jobs have completed.

More Information:

For more information about extract procedures and the Job Generation Parms options, see the *CA Database Analyzer User Guide*.

Implementing a Naming Convention for CA Database Analyzer Extract and Action Procedures

Implement a naming convention for your extract and action procedures that follows a systems or database administrator's group standard. A naming convention enables easy recognition and identification, and the ability to specify filter criteria when looking at a long list of procedures.

Extract and action procedures have 8-character names. Each procedure also has a 25-byte long description field. The names and their descriptions can be filtered by using the standard percent (%) and underscore (_) DB2 wildcarding characters, as shown in the following example:

```

PROCEDURE SELECTION LIST PARAMETERS

  Name ==> HR%      Creator ==> *
  Desc ==> %REBIND%
-----
0 EXT PROC PROCEDURE DESCRIPTION
_-----
_ HR#A#001 RTOS REORG W/REBIND

```

Business Value:

Implementing a naming convention helps you maintain your procedures more efficiently and provides an organized way of locating procedures to be changed or reviewed.

Integrate CA Plan Analyzer with Your Change Management Process

Integrate CA Plan Analyzer into the change management process in place at your site (such as CA Endeavor SCM) so that programs and packages that are being promoted are explained automatically and compared to the previous access path. The Compare report highlights the access path changes.

To integrate CA Plan Analyzer with your change management process, append the following job step to the change management workflow that is used to move and promote programs that have DB2 statements:

Note: Modify all bold parameters to meet your site's environment.

```
//STEP1 EXEC PGM=PTLDRIVM,REGION=4M,
//          PARM=' SUFFIX=00,EP=BPLBCTL '
//STEPLIB DD DISP=SHR,DSN=high-level.CDBALOAD
//PTILIB DD DISP=SHR,DSN=high-level.CDBALOAD
//PTIPARM DD DISP=SHR,DSN=high-level.CDBAPARM
//SYSOUT DD SYSOUT=X
//UTPRINT DD SYSOUT=X
//ABNLIGNR DD DUMMY          SUPPRESS ABENDAID DUMPS
//BPIIPT DD *
.CALL EXPLAIN
.DATA

RULESSID = (ssid)
ACM      = (N,userid)
STRATEGY = (ssid,program,userid,@AUTO)
PLANTAB  = (ROLLBACK)
SQLQUAL  = (userid)
LINES    = (55)
PROCDDF  = (N)
FLOATFMT = (SCI)
SAVEHOST = (Y)
PROCVIEW = (Y)
EXPLTYPE = (FUTURE)
ISOLATE  = (CS)
TARGET   = (ssid(@DEFAULT))
PERFTIE  = (N)
DATABASE = (STRATEGY)
REPORT   = (ACCESS/SHORT,PREDICATE,
            DEPENDENCY/SHORT,
            STATISTICS,SQLRULE/SHORT,
            PREDRULE,COMPARE)
STATSRPT = (TSP,TB,IX,IXP,CO)
```

```
COMPOPTS = (ABNAEE CY, ,N,08,05,)  
SRCDBRML = (dbrmlib(program), )  
  
.ENDDATA  
/*  
//BPIOPT DD *  
.CONTROL BPID(BP-USERA01-114817-20080731) +  
LOGID(ssid) UNIT(SYSDA)  
.LIST SYSOUT(X, , )  
.OPTION NOERRORS  
.RESTART OVERRIDE  
.CONNECT ssid
```

If CA Plan Analyzer has not previously analyzed an access path for this program, it creates a new strategy and names it after the program and the first version of the access path created. If CA Plan Analyzer has already analyzed an access path for this program, it creates a new EXPLAIN version and generates the Compare report (if requested) and any other specified reports.

After CA Plan Analyzer creates its strategies, you can access Explain Strategy Services from the CA Plan Analyzer main menu to view the strategies and versions for every strategy and program. In Explain Strategy Services, you can also enter the R line command next to a version to view report output for the version (for further analysis).

Business Value:

Performance analysts can proactively review performance changes before the new or changed program is used in production and reactively go back in time to verify when performance changed for a specific program or package. The generated reports include access path comparison and catalog statistics, so you can quickly identify the exact reason for the performance change.

Additional Considerations:

You can establish this process in batch and using the online panels and parameters.

More Information:

For more information about access path analysis, see the *CA Plan Analyzer User Guide*.

Database Backup and Recovery Configuration Best Practices

Database backup and recovery configuration best practices include enabling parmlib member settings for the following:

- CA Fast Recover
- CA Merge/Modify
- CA Log Analyzer
- CA Quick Copy (including dynamic allocation)

Enable CA Fast Recover Specific Parmlib Member Settings

We recommend that you specify the following values in the PFR member of *high-level.CDBAPARM*:

- LOG-BACKUP YES
- RETAIN YES

Business Value:

Specifying LOG-BACKUP YES enables the product to look for log records to complete the recovery if the image copy does not exist or the allocation fails. Specifying RETAIN YES optimizes the object recovery so that a VOLSER is mounted only once during the recover statement.

More Information:

For more information about these keywords, see the *CA Fast Recover User Guide*.

Enable CA Merge/Modify Specific Parmlib Member Settings

We recommend that you specify LOG-BACKUP YES in the PMM member of *high-level.CDBAPARM*.

Business Value:

Specifying LOG-BACKUP YES enables the CA Merge/Modify ACCUMULATE command to look for log records to complete the recovery if the image copy does not exist or the allocation of the image copy fails.

More Information:

For more information about CA Merge/Modify syntax, see the *CA Merge/Modify User Guide*.

Enable CA Log Analyzer Specific Parmlib Member Settings

We recommend the following settings in the PLA member of *high-level.CDBAPARM*:

- Set UR_WORK_LOAD to L (large) to maximize performance at the cost of more memory. This parameter specifies how much storage CA Log Analyzer uses to manage DB2 units of recovery (URs) and is a trade-off between run-time memory and the CPU and elapsed time for CA Log Analyzer jobs.
- Set PRE_ALLOCATE to Y to pre-allocate DB2 log data sets. This setting is a trade-off in elapsed time against the number of allocated data sets. If CA Log Analyzer preallocates logs, it may allocate additional log data sets that are not needed for the processing range. However, if your standard log range covers multiple log data sets, a value of Y lets CA Log Analyzer reduce the amount of time it spends waiting for an archive log data set to be available.
- Set KEEPDICTIONARY to Y to read the underlying tablespace to retrieve the compression dictionary instead of first reading an image copy.
- Set the INDEX_SEARCH to PCO (primary index (P), clustering index (C), and other unique index (O)). The letter order determines the order in which CA Log Analyzer looks at and uses the indexes, if they are available.

Business Value:

These parmliib settings will generally allow you to optimize the performance of CA Log Analyzer in your environment.

More Information:

For more information about CA Log Analyzer parmliib settings, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide*.

Enable CA Quick Copy Specific Parmlib Member Settings

We recommend that you specify the following CA Quick Copy values as needed in the UTIL parmliib member:

- Set OBJ-ORDER to SIZE when copying multiple objects of variable sizes with a COPY-TASKS value greater than 1. OBJ-ORDER can be defined in syntax or globally in the UTIL member of *high-level.CDBAPARM*. OBJ-ORDER SIZE sorts the objects in descending size order and copies large objects first.

- Set MODIFYBITS YES for full copies and MODIFYBITS NO for incremental copies if you take weekly full copies with daily incremental copies.

Consider the following:

- MODIFYBITS YES can impact the availability of an object for user applications and increase the copy time. It also updates the space map pages of the tablespace and requires that the tablespace be stopped and started at the end of the copy to flush out any updated pages from the DB2 buffers.

Note: SHRLEVEL REFERENCE or NONE is required, which makes the object unavailable to other applications. If you cannot tolerate any downtime with the use of MODIFYBITS YES and SHRLEVEL REFERENCE, consider taking an LRSNCOPY to create incremental image copies.

- MODIFYBITS NO with MERGE-COPY YES keeps only the last incremental image copy data set entry in SYSIBM.SYSCOPY between full image copies. MERGE-COPY YES processing is performed only for a DSNUM ALL copy, that is, when a single image copy data set is created from a COPY that contains all partitions. MERGE-COPY YES can increase execution time.
- Evaluate use of the SINGLE-ATTACH keyword to see whether improvements can be made in CPU and elapsed time. This keyword works best when copying a large number of objects with multitasking (that is, with COPY-TASKS > 1 specified). This keyword can be defined in syntax or globally in the UTIL member of *high-level.CDBAPARM*.
- Evaluate use of the CA Quick Copy access method by setting EXCP YES and EXCP NO to determine which option performs better. EXCP YES generally performs better, but cannot be used for data sets that use the STRIPED attribute. EXCP NO uses Media Manager, supports all data set attributes, and generally supports newer devices.

Business Value:

By specifying the most commonly used values for CA Quick Copy processing in the UTIL member of *high-level.CDBAPARM*, you no longer need to provide the values for these keywords in your CA Quick Copy jobs. This simplifies the syntax that is needed for your copy jobs. If necessary, you can easily override any UTIL member value by specifying the replacement keyword in the syntax.

OBJ-ORDER SIZE processing may prevent a single task from running longer when other tasks have completed their work, which can result in shorter processing times.

Additional Considerations:

An LRSNCOPY can achieve the results of a SHRLEVEL CHANGE, MODIFYBITS YES copy, without making the object unavailable to applications. If you specify LRSNCOPY YES, MODIFYBITS NO, and MERGE-COPY NO, CA Quick Copy uses the RBA/LRSN from the most recent full or incremental copy defined in SYSIBM.SYSCOPY and copies only those pages that have changed since this last copy. An LRSNCOPY processes the tablespace sequentially and does not use the information from the spacemap pages to determine which pages to copy. For large objects with few updates, taking an LRSNCOPY may increase execution time. An LRSNCOPY can be used effectively for objects with heavy updating.

More Information:

Many keywords work in combination with other keywords. For more information about the keywords and processing options, see the *CA Quick Copy User Guide*.

How to Enable CA Quick Copy Dynamic Allocation Parmlib Member Settings

We recommend that you dynamically allocate image copy data sets. You can implement dynamic allocation by doing any of the following:

- Using TEMPLATE control statements in the copy syntax or a TEMPLATE DD statement in the JCL.
- Defining customized parameters within the PQC member of *high-level.CDBAPARM*.
- Specifying the PQCPARM options within the COPY syntax.

Note: You can enable dynamic allocation in the COPY syntax with PQCPARM or within the PQC member of *high-level.CDBAPARM* only if you are not using LISTDEF with TEMPLATE.

Note the following:

- Specify the %DSNUM symbolic for the data set name model if more than one data set is generated for a single image copy. CA Quick Copy generates more than one image copy data set if COPY-TASKS is greater than 1 for a partitioned object or if PARTS-PER-COPY is specified.
- Specify DISP=NEW,CATLG,CATLG if you need to restart jobs that fail.
- Specify FILESEQ=0 if you want to stack copies onto the same tape. You can stack image copies of the same type onto the same tape. (For example, all LP copies are copied to the same tape if FILESEQ=0 is specified for the LP model and UNIT is a tape device.) When you stack copies on tape, COPY-TASKS cannot exceed 16.
- Use the VOL parameter to specify a list of volumes for the image copy data set. If you require more than five volumes, you may also need to specify the VOLCNT parameter.

- (Optional) Use DATACLAS, MGMTCLAS, or STORCLAS for data set placement.
- (Optional) Use EXPDT (expiration date) or RETPD (retention period) for a given data set model.
- Specify DCB or LIKE to provide a model data set name. If GDG data set names are required, CA Quick Copy automatically builds the base data set entry if one does not exist. You can specify a default GDG-LIMIT in the UTIL member of *high-level.CDBAPARM*.
- Specify the PQCPARM keyword within the COPY syntax to override the settings in the PQC parmlib member.

Business Value:

Using dynamic allocation allows for simple creation of image copy data sets and a simple restart of a copy job that fails. The data set name is stored in the restart rows for the job and is used again for a restarted job. Thus, no syntax or JCL changes are required to restart jobs.

When using dynamic allocation, CA Quick Copy computes the required amount of space needed for the image copy data set. As a result, jobs do not fail because of inadequate space specifications. Dynamic allocation also supports the use of symbolics, so you can generate unique data set names for all image copy data sets.

Additional Considerations:

You can create up to eight copies of a given object. These copies include the standard Local Primary (LP), Local Backup (LB), Recovery Primary (RP), Recovery Backup (RB), and an additional copy of each type, designated with LP2, LB2, RP2, and RB2.

Note: You must define a model for each copy type before using the model in a COPY job.

When creating multiple copies of an object, verify that the output image copy data set names are unique. Otherwise, the data sets are not cataloged.

The following example shows parameters for multiple copies:

```
LP      DSN=PROD.ACCTG.%.DBNAME.%.TSNAME.%.LP%DSNUM.%.T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=20
UNIT=SYSDA
STORCLAS=PRODACCT
```

```
LB      DSN=PROD.ACCTG.%DBNAME. .%TSNAME. .LB%DNUM. .T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=60
UNIT=VTAPE
RP      DSN=PROD.ACCTG.%DBNAME. .%TSNAME. .RP%DNUM. .T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=20
UNIT=SYSDA
STORCLAS=RMTACCT
RB      DSN=PROD.ACCTG.%DBNAME. .%TSNAME. .RB%DNUM. .T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=60
UNIT=VTAPE
LP2     DSN=PROD.ACCTG.%DBNAME. .%TSNAME. .LP2%DNUM. .T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=30
UNIT=SYSDA
LB2     DSN=PROD.ACCTG.%DBNAME. .%TSNAME. .LB2%DNUM. .T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=30
UNIT=TAPE
RP2     DSN=PROD.ACCTG.%DBNAME. .%TSNAME. .RP2%DNUM. .T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
RETPD=60
UNIT=TAPE
```

```
RB2      DSN=PROD.ACCTG.%DBNAME..%TSNAME..RB2%DNUM..T%TIME
DISP=NEW
NDISP=CATLG
CDISP=CATLG
FILESEQ=0
EXPDT=2010365
UNIT=TAPE
VOL=(Vol1,Vol2,Vol3,....)
VOLCNT=9
```

More Information:

Several symbolic variables are supported for use with the DSN parameter. For more information about dynamic allocation and the related parameters, see the *CA Quick Copy User Guide*.

Chapter 4: Performance Management Best Practices

This section contains the following topics:

[CA Bind Analyzer Best Practices](#) (see page 51)

[CA Database Analyzer Best Practices](#) (see page 53)

[CA Insight DPM Best Practices](#) (see page 76)

[CA Plan Analyzer Best Practices](#) (see page 78)

[CA Rapid Reorg Best Practices](#) (see page 81)

CA Bind Analyzer Best Practices

As a CA Bind Analyzer best practice, we recommend implementing CA Bind Analyzer BIND card modeling.

Implementing CA Bind Analyzer BIND Card Modeling

Integrate CA Bind Analyzer BIND card modeling into build JCL, so that programs and packages with SQL changes or new SQL statements are bound automatically. If the program has not previously been bound to DB2, a package bind “model” can be created, with the name being masked for programs that have specific naming conventions. If the DBRM or program has already been bound to DB2, the package bind is generated from the existing catalog entry for that package.

This process involves performing a BIND PACKAGE command for a model package on each DB2 subsystem where the program build is targeted.

Business Value:

Establishing a model package in the DB2 subsystem provides the following benefits:

- Proactive package binds

You can compare DBRM changes that exist after the current precompile takes place, and you can generate bind cards when the new DBRM has changed (as opposed to the DBRM used in production). This helps avoid costly and resource-consuming rebinding.

- Consistent set of bind parameters for a large group of DB2 packages
You can bind new programs in the same fashion as existing programs. Not having to specify values for each package individually saves time and manpower.
- Automation of the time-consuming manual process of creating package bind cards for new programs

Additional Considerations:

We suggest the following bind card template as a base package model:

```

BIND PACKAGE ( CA_BIND_ANALYZER ) -
OWNER ( userid ) -
QUALIFIER ( userid ) -
ACTION ( REPLACE ) DEGREE ( 1 ) -
-
RELEASE ( COMMIT ) ISOLATION ( CS ) -
VALIDATE ( RUN ) EXPLAIN ( NO ) -
DYNAMICRULES ( ) -
-
LIBRARY ( 'high-level.CDBADBRM' ) -
MEMBER ( MODEL ) -
-
SQLERROR ( NOPACKAGE ) CURRENTDATA ( NO ) -
KEEPDYNAMIC ( NO ) -
DBPROTOCOL ( DRDA ) NOREOPT ( VARS ) -
IMMEDWRITE ( NO ) ENCODING ( 500 ) -
;

```

Sample JCL Precompile Jobstep:

This job step performs a precompile of a source program/package named PBAV9CM. The PARM area of the EXEC statement provides a model package for the subsystem where the program would get bound. By including the PTIBIND DD and the PBA(B,K) and SSID(D91A) parameters, the CA Bind Analyzer precompiler generates BIND control cards for the resulting DBRM.

Because the model contains PBA*, the mask is inclusive of the source program name, PBAV9CM. Here is the JCL job step definition:

```

001000 //*=====*
001100 //PC      EXEC PGM=PBLHPC,
001200 //  PARM=('HOST(IBMCOB) APOST APOSTSQL PBA(B,K) SSID(D91A)',
001300 //      'MODEL(K,PBA*,MODEL,CA_BIND_ANALYZER)')
001500 //*
001600 //PTILIB   DD  DISP=SHR,DSN=PTIPROD.RD115P3.CDBALOAD
001800 //STEPLIB  DD  DISP=SHR,DSN=PTIPROD.RD115P3.CDBALOAD
002000 //PTIPARM   DD  DISP=SHR,DSN=PTIPROD.RD115P3.CDBAPARM
002200 //DSNHPC   DD  DISP=SHR,DSN=D91A.PRIVATE.SDSNEXIT
002300 //          DD  DISP=SHR,DSN=DB2.DB2910.GA.SDSNLOAD
002400 //*
002410 //PBAPARM  DD  DISP=SHR,DSN=PBA.AUTOTEST.JCLLIB(PBDEBUGA)
002500 //PTIDBRMO DD  DSN=PBA.CTRP.CDBADBRM(DBRMOLD),DISP=(OLD,KEEP)
002510 //PTIBIND  DD  DSN=&&PTIBIND,DISP=(MOD,PASS),UNIT=SYSDA,
002520 //          SPACE=(CYL,(1,1))
002600 //*
002900 //DBRMLIB  DD  DSN=PBA.AUTOTEST.DBRMLIB(PBAV9CM),DISP=SHR /* DBRM */
003000 //SYSCIN   DD  DSN=PBA.AUTOTEST.SYSCINV9,DISP=OLD
003100 //*
003200 //SYSIN    DD  DISP=SHR,DSN=PBA.AUTOTEST.CDBASRC(PBAV9CM) /*SOURCE */
003300 //SYSLIB   DD  DISP=SHR,DSN=CANKEB1.DB2.DCLGEN /* PDS */
003500 //*
003900 //SYSUT1   DD  SPACE=(CYL,(2,1),,ROUND),UNIT=SYSDA
004000 //SYSUT2   DD  SPACE=(CYL,(2,1),,ROUND),UNIT=SYSDA
004100 //SYSPRINT DD  SYSOUT=*
004200 //SYSTEM   DD  SYSOUT=*
004300 //*

```

More Information:

For more information about modeling and using the MODEL keyword, see the *CA Bind Analyzer User Guide*.

CA Database Analyzer Best Practices

CA Database Analyzer best practices include the following:

- [Retrieve values directly from statistics tables for CA Database Analyzer action procedures](#) (see page 54).
- [Use CA Database Analyzer to monitor the IBM RTS tables for missing information](#) (see page 56).
- [Use #INCLUDE statements to specify longer predicate criteria](#) (see page 58).

- [Use #INCLUDE statements to create new action conditions](#) (see page 60).
- [Use real-time statistics with CA Database Analyzer](#) (see page 63).
- [Use a combination of IBM RUNSTATS and CA Database Analyzer extract procedures](#) (see page 66).
- [Use RTOS to dynamically select objects for maintenance](#) (see page 68).
- [Use RTOS to reorganize PBG tablespaces with hash access](#) (see page 73).

Retrieving Values Directly from Statistics Tables for CA Database Analyzer Action Procedures

When you create action procedures, you can include various symbolics in your model JCL to perform space allocations and decision-making logic. CA Database Analyzer action procedures support the following types of symbolics:

- User symbolics (US), which are defined by the user (for example, %FULL)
- User automatic symbolics (UAS), which are defined using #SET (for example, #SET %MYTYPE = %SUBSTR(1,1,%OBJTYPE))
- Automatic symbolics (AS), which are defined by the product (for example, %REORGP)
- System automatic symbolics (SAS), which are defined by the product (for example, %PTILIB1)

If statistics or object definition values are not available as automatic symbolics, you can retrieve these values from the CA Database Analyzer (PDA), Real-Time Statistics (RTS), or DB2 catalog tables by creating your own user symbolics. These are created by applying the following tablespace and index action conditions to your action procedures:

- (TS)/(TSP) Set USER1 w/PDA/RTS/DB2 Col
- (TS)/(TSP) Set USER2 w/PDA/RTS/DB2 Col
- (TS)/(TSP) Set USER3 w/PDA/RTS/DB2 Col
- (TS)/(TSP) Set USER4 w/PDA/RTS/DB2 Col
- (IX)/(IXP) Set USER1 w/PDA/RTS/DB2 Col
- (IX)/(IXP) Set USER2 w/PDA/RTS/DB2 Col
- (IX)/(IXP) Set USER3 w/PDA/RTS/DB2 Col
- (IX)/(IXP) Set USER4 w/PDA/RTS/DB2 Col

For each of the preceding action conditions, you can specify a column name from the PDA, RTS, or DB2 catalog table. These conditions set a symbolic variable named %USER1, %USER2, %USER3, or %USER4, which you can then reference in your action procedure using %CALC or #IF-THEN logic.

Note: For more information about a particular action condition, including common usage and examples, type **H** (Help) as a line command next to the action condition and press Enter.

Example: Set %USER1 and %USER2 to Values from SYSIBM.SYSTABLESPACESTATS

This example sets %USER1 to the NACTIVE value from SYSIBM.SYSTABLESPACESTATS and sets %USER2 to the COPYUPDATEDPAGES value in SYSIBM.SYSTABLESPACESTATS:

```

O DESCRIPTION                                T CONDITION
S (TS)/(TSP) Set USER1 w/PDA/RTS/DB2 Col  R | I,TSS.NACTIVE
S (TS)/(TSP) Set USER2 w/PDA/RTS/DB2 Col  R | I,TSS.COPYUPDATEDPAGES

```

Example: Customize an IBM Image Copy Model

This example customizes an IBM image copy (IC Utility Code) model. Review where %USER1 and %USER2 are used in the #IF statement.

Note: The default model is in *high-level.CDBAMDL(MJUTLIC)*.

```

//%SYSCPY DD DSN=BACKUP.%DBNAME..%TSNAME..%DATE..%TIME,
// UNIT=TAPE, LABEL=(%LBL,SL),DISP=(,CATLG),
// VOL=(,RETAIN%REFER)
#CM USER1 is set to the value of TSS.NACTIVE
#CM USER1LIT is set to the literal 'TSS.NACTIVE'
#CM USER2 is set to the value of COPYUPDATEDPAGES
#CM USER2LIT is set to the literal 'COPYUPDATEDPAGES'
/* %USER1LIT = %USER1 %USER2LIT = %USER2 %CALC(%USER2*100/%USER1)
COPY TABLESPACE %DBNAME..%TSNAME DEVT TAPE COPYDDN %SYSCPY
#CM If the number of pages updated is > 20%, then do a full copy.
#SET %PERCCHGD = %CALC(%USER2*100/%USER1)
#IF(%PERCCHGD,>,20)
    %PARTLBL %PART FULL YES SHRLEVEL REFERENCE
#ELSE
    %PARTLBL %PART FULL NO SHRLEVEL REFERENCE
#ENDIF

```

Business Value:

The ability to obtain statistics or object definition information from the PDA, RTS, or DB2 catalog tables gives you greater flexibility in customizing your action procedures to meet specific site requirements.

Additional Considerations:

Automatic symbolic variables are very useful for customizing your models to meet the needs of your site. CA Database Analyzer has many automatic symbolic variables that can be referenced in customizable utility models (skeletons) in such things as %CALC space calculations or #IF model processing logic. For more information about these variables, select **T** (Tutorial) from the CA Database Analyzer main menu, then select **7** (Symbolic Variable Descriptions).

More Information:

For more information about creating action procedures and applying action conditions to them, see the *CA Database Analyzer User Guide* and *CA Database Analyzer Reference Guide*.

Use CA Database Analyzer to Monitor the IBM RTS Tables for Missing Information

CA Database Analyzer provides a statistics maintenance option that checks for missing rows in the IBM Real-Time Statistics (RTS) tables. RTS tables may have missing information if locks were unresolved, RTS objects were stopped, or a utility did not accurately update the RTS statistics or insert rows. Missing RTS rows can cause join SQL+100s or statistics retrieval problems in RTOS extract procedures, or false REJECT-2s for RTS-based action conditions within an action procedure.

The statistics maintenance option also checks for missing RTS utility timestamp information. Many processes, such as RTOS extract or action procedures that have RTS-based conditions, depend on accurate utility execution timestamps to select objects for processing.

Note: Missing information is more common on DB2 V8 systems or on DB2 9 systems that were upgraded from DB2 V8.

The statistics maintenance option provides two functions: The Verify function verifies whether information is missing, and the Update function updates the RTS tables with the missing information. We recommend that you run these functions on a regular basis. You can also submit them from a job scheduler.

To monitor the RTS tables for missing information

1. Select option **3** (Statistics Maintenance) from the CA Database Analyzer main menu and press Enter.

The STATISTICS MAINTENANCE MAIN MENU appears.

2. Select option **4** (Update Statistics for IBM Real-Time Statistics) and press Enter.

The REAL-TIME STATISTICS MAINTENANCE panel appears.

3. Verify whether information is missing:
 - a. Enter your selection criteria for the database name.
 - b. Type **Y** next to the rest of the fields.
 - c. Type **V** (Verify) in the command line.

Note: If you type **U** (Update) instead of **V**, CA Database Analyzer will automatically update the RTS tables with the missing information. We recommend that you type **V** so that you can view updates before they are carried out.

- d. Press Enter.

CA Database Analyzer generates batch processor JCL and SQL SELECT statements and displays them in a report that lists the missing RTS and utility values.

4. Update the RTS tables with the missing information by doing *one* of the following:
 - Type **SUBMIT** on the command line and press Enter.
 - Save the JCL for later submission (either manually or by job scheduler).

When you submit the JCL, the RTS tables are updated with the missing information.

Business Value:

Having complete and accurate RTS statistics and timestamps helps ensure that all RTS-based processing works as expected. Missing or outdated RTS information may cause maintenance actions such as the creation of image copies to be missed.

Additional Considerations:

If the statistics maintenance option regularly reports missing or outdated RTS utility timestamp information, review the utility being run to determine why this is happening. For example, verify that the CA utility has UPDTE-RTS-TABLES YES specified in *high-level.CDBAPARM(UTIL)*. If more than one parmlib is used, update all of them. Also, if you use CA Database Analyzer instead of RUNSTATS to update the DB2 catalog, verify that UPDATE_RTS D is specified in *high-level.CDBAPARM(PDA)*. You can use the parmlib editing facility to verify these settings.

More Information:

For more information about the statistics maintenance option, access the REAL-TIME STATISTICS MAINTENANCE panel and press PF1 to view its online help. Also see the JCL comment cards that are generated with the job.

Use #INCLUDE Statements to Specify Longer Predicate Criteria

When creating an action condition for an action procedure, you are limited to a 30-byte maximum. This limits the criteria that you can specify for object selection, which can result in unnecessary processing (for example, reorganizing tablespaces that are too small to benefit).

You can exceed this 30-byte limitation and specify common or large predicate criteria by placing SQL statements into a model PDS member. You then insert a #INCLUDE <model member> statement in the 30-byte area of your action condition. (This concept is similar to the use of a #INCLUDE <model member> statement with utility code models and similar to the INCLUDE processing done in z/OS JCL processing.) CA Database Analyzer uses a CA common model service that supports the ability to append (copy) a model member onto the SQL that is currently being processed. The appended model is processed as if it were originally part of the current SQL.

The following shows an example of large predicate criteria that can be added to a model PDS member:

```
AND DBNAME||NAME NOT IN (SELECT DB||TS FROM users.reorg_exclusion_table WHERE...)
```

Note: You can also use the IBM exclusion table, DSNACC.EXCEPT_TBL, instead of the user table. We recommend that you use the IBM table for REORG and COPY functions and for statistics collection, because other CA processes, such as RTOS, automatically use this table.

The following example adds “#INCLUDE RTSTS#01” to the reorg action conditions. Model member RTSTS#01 contains “AND NACTIVE > 360” to avoid reorganizing small tablespaces:

```
(TS) RTS reorg percent of ins, del, upd | > 25 #INCLUDE RTSTS#01 |
```

```
(TS) RTS reorg percent of unclustered | > 10 #INCLUDE RTSTS#01 |
```

You can also add up to approximately 32 KB of additional predicates and table expressions to this model.

To use a #INCLUDE statement to specify longer predicate criteria

1. Select option **2** (Action Procedures) from the CA Database Analyzer main menu.
The ACTION PROCEDURE SERVICES panel appears.
2. Enter the **C** (Create) or **U** (Update) command and press Enter.
The ACTION CONDITIONS panel appears.
3. Enter **A** (And) or **O** (Or) in the Condition Type field and press Enter.
A list of action conditions appears.

4. Enter **U** (Update) next to an action condition, enter **#INCLUDE <model member>** in the **CONDITION** column, and press Enter.

You are prompted to select a model library for the new model member you are about to create.

5. Select a model library. You can select any library in your *high-level.CDBAPARM(DSNAMEx)* member. Press Enter.

An empty model member appears.

6. Create an SQL statement that defines the predicate criteria, and then press PF3 (End).

The **ACTION CONDITIONS** panel reappears. You have added a **#INCLUDE** statement that specifies longer predicate criteria in your action condition. When CA Database Analyzer evaluates the action condition, it will read the contents of the model library member specified in the **#INCLUDE** statement and append your SQL predicate to the end of the existing condition.

Business Value:

Extending the action conditions for your action procedures lets you provide additional technical criteria that prevent unnecessary processing (for example, reorganizing tablespaces that are too small to benefit).

Additional Considerations:

The **#INCLUDE** criteria to limit objects based on size are part of the extract options that are available with an action-oriented (RTOS) extract procedure, so we recommend using that type of extract instead of the classic name-oriented extract.

Also, consider implementing a naming convention for your **#INCLUDE** statements that indicates which tables the predicates are associated with. In the previous example, **RTSTS#01** is associated with the IBM Real-Time Statistics (RTS) tables (specifically, the one related to tablespaces). You might have the first three characters indicate the type of table being processed and have characters 4 through 6 match the table. For example, you might use **RTS**, **PDA**, or **DB2** as the first three characters:

- **RTS** indicates that the predicate accesses tables **SYSIBM.SYSTABLESPACESTATS** or **SYSIBM.SYSINDEXSPACESTATS**.
- **PDA** indicates that the predicate accesses tables **PTI.RATS_STATS_XXXX** or **PTI.RAIX_STATS_XXXX**.
- **DB2** indicates that the predicate accesses tables **SYSIBM.SYSTABLESPACE**, **SYSIBM.SYSINDEXPART**, and so on.

For characters 4 through 6, you could use TS, IX, TSP, IXP, and so on, as shown in the following examples:

- RTSTS indicates SYSIBM.SYSTABLESPACESTATS.
- RTSIX indicates SYSIBM.SYSINDEXSPACESTATS.
- DB2IXP indicates SYSIBM.SYSINDEXPARTS.
- PDATS indicates PTI.RATS_STATS_XXXX.

You can also use characters 7 and 8 to add a suffix such as 01. You would increment from 01 to 02 to 03, and so on (for example, RTSTS01, RTSTS02). Using number suffixes instead of letters can differentiate this type of member, which includes SQL to be appended to an existing SELECT statement, from similarly named members that contain entire SELECT statements to create new action conditions.

More Information:

For a sample #INCLUDE statement that includes predicate criteria, browse the *high-level.CDBAMDLMCEAC03* member provided with CA Database Analyzer. This member is fully commented.

For more information about #INCLUDE processing, see the online help for the ACTION CONDITIONS panel. (Specifically, display the online help for this panel, place your cursor on the action conditions hyperlink, and press Enter.)

Use #INCLUDE Statements to Create New Action Conditions

In the past, creating a new action condition for an action procedure required adding a SQL SELECT statement to assembler tables (RAAACTB2 and RAAACTB3), creating a load module, testing the load module, and then placing it into the production load libraries. This process was complicated and error-prone, and some sites did not allow new load modules to be placed into production.

You can now create a new action condition by placing a complete SELECT statement into a model PDS member. You then insert a #INCLUDE <model member> statement in a new, generic action condition. (CA Database Analyzer provides generic action conditions for tablespaces and indexes.) CA Database Analyzer uses a CA common model service that lets you include (copy) another model member into the model that is currently being processed. The included model is processed as if it were originally in the current model. This concept is similar to the INCLUDE processing done in z/OS JCL processing.

The following example adds “#INCLUDE MCEAC01” to a generic action condition for tablespaces:

```
(TS)/(TSP) Customizable SQL condition | #INCLUDE MCEAC01 |
```

The #INCLUDE statement references model member MCEAC01, which contains a complete SELECT statement with several predicate conditions, and is similar to the existing action condition:

```
(TS) RTS DSNACCOX REORG recommended
```

MCEAC01 also contains the following line, which demonstrates that you can customize the 40-byte action condition description. The #HCCD (Help Customized Condition Description) keyword at the beginning signifies that it is the description replacement:

```
#HCCD (TS) RTS REORG recommended – CUSTOM
```

To use a #INCLUDE statement to create new action conditions

1. Select option **2** (Action Procedures) from the CA Database Analyzer main menu.
The ACTION PROCEDURE SERVICES panel appears.

2. Enter the **C** (Create) or **U** (Update) command and press Enter.

The ACTION CONDITIONS panel appears.

3. Enter **A** (And) or **O** (Or) in the Condition Type field and press Enter.

A list of action conditions appears.

4. Enter **U** (Update) next to a generic action condition, and then enter **#INCLUDE <model member>** in the CONDITION column.

Note: CA Database Analyzer provides 10 generic tablespace action conditions and 10 generic index action conditions. They appear on the ACTION CONDITIONS panel as "(TS)/(TSP) Customizable SQL condition" and "(IX)/(IXP) Customizable SQL condition."

Press Enter.

You are prompted to select a model library for the new model member you are about to create.

5. Select a model library. You can select any library in your *high-level.CDBAPARM(DSNAMEx)* member. Press Enter.

An empty model member appears.

6. Create a SELECT statement that defines the action condition, and then press PF3 (End).

The ACTION CONDITIONS panel reappears. You have added a #INCLUDE statement to create a new action condition. When CA Database Analyzer evaluates the action condition, it will read the contents of the model library member specified in the #INCLUDE statement and use it as the complete SELECT statement.

Business Value:

Using a #INCLUDE statement to create new action conditions is easier and less error-prone than the previous method, thereby making it easier for users to customize their sites and removing the need to place new load modules into production.

Additional Considerations:

Consider implementing a naming convention for your #INCLUDE statements that indicates which tables the action conditions are associated with. You might have the first three characters indicate the type of table being processed and have characters 4 through 6 match the table. For example, you might use RTS, PDA, or DB2 as the first three characters:

- RTS indicates that the action condition accesses tables SYSIBM.SYSTABLESPACESTATS or SYSIBM.SYSINDEXSPACESTATS.
- PDA indicates that the action condition accesses tables PTI.RATS_STATS_XXXX or PTI.RAIX_STATS_XXXX.
- DB2 indicates that the action condition accesses tables SYSIBM.SYSTABLESPACE, SYSIBM.SYSINDEXPART, and so on.

For characters 4 through 6, you could use TS, IX, TSP, IXP, and so on, as shown in the following examples:

- RTSTS indicates SYSIBM.SYSTABLESPACESTATS.
- RTSIX indicates SYSIBM.SYSINDEXSPACESTATS.
- DB2IXP indicates SYSIBM.SYSINDEXPARTS.
- PDATS indicates PTI.RATS_STATS_XXXX.

You can also use characters 7 and 8 to add a suffix such as AA. You would increment from AA to AB to AC, and so on (for example, RTSTSAA, RTSTSBB). Using letter suffixes instead of numbers can differentiate this type of member, which contains a complete SELECT statement, from similarly named members that contain SQL to extend the predicate of an existing action condition.

More Information:

For more information about #INCLUDE processing, see the online help for the ACTION CONDITIONS panel. (Specifically, display the online help for this panel, place your cursor on the action conditions hyperlink, and press Enter.)

For a sample #INCLUDE statement, browse the *high-level.CDBAMDL(MCEAC01)* member provided with CA Database Analyzer. This model member demonstrates how to build a complex statement with several predicates where the threshold values can easily be adjusted using #SET commands. The sample also demonstrates how you can add up to 36 lines of help text for the new action condition. Other users can type **H** (Help) next to the new action condition to display the help text, just like they do for built-in action conditions. Senior DBAs can use this to help train junior DBAs.

Using Real-Time Statistics with CA Database Analyzer

Use IBM Real-Time Statistics (RTS) with CA Database Analyzer to select objects and generate utility action JCL. Real-time statistics are grouped into two categories:

- *Globals* are general statistics like NACTIVE and TOTALROWS for tablespaces or NACTIVE, NLEAF, and TOTALENTRIES for indexes.
- *Utility incrementals* are counter statistics that indicate what has changed since the last time a specific utility was run (for example, REORGDELETES, STATSINSERTS, and COPYUPDATEDPAGES).

CA Database Analyzer fully exploits all IBM RTS in an action procedure, and can use them to generate utility action JCL without collecting CA Database Analyzer statistics. This is possible because object selection processing and JCL data set space values (space symbolics) can use RTS as their source (using PDA PARMLIB keyword SPACE_SYM_SOURCE(RTS)).

Note: Using RTS-based conditions within CA Database Analyzer can reduce how frequently you collect CA Database Analyzer statistics. However, we recommend that you still collect these statistics on a weekly or monthly basis for detailed line reports and trend/graph analysis reports. These reports are useful for proactively analyzing and identifying data trends and determining why an object may have been selected for utility maintenance (for example, because of extents growth or objects fragmentation). When collecting CA Database Analyzer statistics, you can use the fast scan option to minimize CPU usage and elapsed time.

You can use RTS (to generate utility action JCL or to collect statistics) in the following ways:

- Specify the use of RTS in the extract procedure.

When you build an extract procedure, select MODE = A (action oriented) to invoke Real Time Object Selection (RTOS), and then select a maintenance action (for example, reorganization), CA Database Analyzer automatically selects extract conditions that use the RTS utility incremental statistics to select the objects to include in the extract. The extract procedure must be tied to an action procedure to generate the maintenance action JCL, but the action procedure typically will not have any action conditions applied. (Action conditions are used to filter the objects to be selected, but this is unnecessary because the extract conditions have already filtered the objects.)

This method narrows down the objects to be processed as soon as possible, which uses less memory and CPU resources. For example, if 20,000 tablespace objects exist but only 300 qualify to be reorganized via RTS statistics, then only 300 tablespaces are defined in the extract.

- Specify the use of RTS in the action procedure.

When you create an action procedure, you can add RTS-based action conditions. When this action procedure is tied to a classic MODE = N (name) extract procedure, CA Database Analyzer will use the extract procedure to select objects by their names (instead of by their statistics), and then apply the associated RTS-based action conditions to further filter the objects to be selected.

This method narrows down the objects to be processed by creating a subset from all of the objects that are defined in the associated extract procedure (the superset). For example, 20,000 tablespaces may be defined in the extract (versus 300 using RTOS extracts), but only 300 of them qualify to be reorganized via the RTS referenced in the action conditions.

Note: If you are not using action conditions based on CA Database Analyzer statistics tables, specify Show PDA table related Action Conds = N in the PDA parmlib editor.

- Specify the use of RTS in the extract procedure and action procedure.

This method lets you use RTS statistics to select a superset of objects, and then use RTS to generate JCL for a subset of those objects. This is done in the following manner:

- a. Build an RTOS-based extract procedure, change the two "Promote to a TS auto-build ..." options to Y, and select the option to collect CA Database Analyzer statistics when the extract is submitted.

The extract procedure, when submitted, will provide detailed CA Database Analyzer statistics of the *entire* tablespace for future analysis even if it selects, for example, only one index or tablespace partition for reorganization.

- b. Create an action procedure with conditions and trigger values that mirror the values in the extract procedure, and tie it to the extract procedure.

The action procedure will generate maintenance action JCL only for the selected objects. For example, it may generate reorganization JCL for the index or tablespace partition that was selected.

Note: Instead of mirroring the conditions and trigger values, you can make the action condition slightly more restrictive than the extract condition. For example, if the extract condition specifies a trigger of 40 extents, specify 50 extents in the corresponding action condition. CA Database Analyzer will start collecting statistics when the object reaches 40 extents, but will not create the JCL to reorganize the object until it reaches 50 extents. You can also accomplish the same thing by inserting the same `#INCLUDE <model member>` statement in the extract procedure and the action condition. In this case, you must create the action condition yourself using one of the generic “Customizable SQL condition” conditions. See the example provided with CA Database Analyzer in *high-level.CDBAMDL(MCEAC01)*.

You can use RTS to resolve data set allocation space-related symbolics when generating utility action JCL by specifying Stat source for space-related symbolics = AUTO_RTS in the PDA parmlib editor.

Business Value:

Using IBM RTS to select objects and generate utility action JCL lets you use your resources more efficiently, and can reduce CPU consumption and the elapsed times for your statistics jobs. Because real-time statistics are captured while DB2 is running, you can collect CA Database Analyzer statistics less frequently and only on those objects that meet your criteria.

Additional Considerations:

To help ensure that the IBM RTS statistics are complete, use CA Database Analyzer to monitor the IBM RTS tables.

More Information:

For more information about the CA Database Analyzer RTOS feature, see the RTOS extract help panels. To access the help panels, press PF1 from the CA Database Analyzer main menu and then select 10 (Real Time Object Selection help).

For more information about extract and action procedures and conditions, see the *CA Database Analyzer User Guide* and *CA Database Analyzer Reference Guide*.

For information about checking for missing rows in RTS tables, see the best practice for using CA Database Analyzer to monitor the IBM RTS tables for missing information.

Using a Combination of IBM RUNSTATS and CA Database Analyzer Extract Procedures

Use a combination of IBM RUNSTATS and CA Database Analyzer extract procedures to collect statistics on your system.

Statistics are gathered for two primary reasons:

- To update the DB2 catalog with statistics that the optimizer can use to choose the best access path. This is generally useful only when the object has changed by something like INSERTs or after an object reorganization.
- To gather statistics on a weekly or monthly basis and store them in history tables for analysis. This analysis could be done when exceptions occur, such as when an object is selected to be reorganized, and include the trending data that led to the reorg selection. You could also review The anatsysis could for other reasons (such as space usage over time).

Statistics are commonly gathered using the following methods:

1. IBM RUNSTATS
 - a. DB2 catalog updates are performed using the UPDATE YES keyword.
 - b. IBM history tables (the _HIST tables) are populated using the HISTORY YES keyword.
2. CA Database Analyzer extract procedures
 - a. DB2 catalog updates are performed by specifying the following options:
 - Submit - Build JCL Parameters panel: Collection Type ==> (R)egular or (F)ast Scan
 - Build Extract Procedures panel: Auto Catalog Update ==> Y.
 - b. CA Database Analyzer history tables (named RAxx_STATS_####) are populated by specifying Collection Type ==> (R)egular or (F)ast Scan on the Submit - Build JCL Parameters panel.
 - c. IBM history tables (the _HIST tables) are populated by specifying ADD_IBM_HIST ALSO or ONLY in the PDA parmlib member.

Note: CA Database Analyzer provides flexible maintenance options for the CA Database Analyzer history tables, including automatic statistics deletion (menu option 3.1), in which you can specify the conditions under which object statistics for a DB2 subsystem are deleted, the frequency of deletions, and how old statistics must be before they are deleted.

By using RUNSTATS and CA Database Analyzer extract procedures, you can take advantage of the strengths of each method.

For example, you can use RUNSTATS to collect optimizer statistics (1a above) and use CA Database Analyzer to collect historical statistics (2b above). Using RUNSTATS to collect optimizer statistics lets you collect advanced statistics such as HISTOGRAM NUMCOLS 1 NUMQUANTILES 10, and may also allow for future RUNSTATS autonomic table column and index key column specific keywords. Using CA Database Analyzer to collect history statistics lets you use Real Time Object Selection (RTOS) to limit the number of objects that are processed, thereby reducing CPU consumption and the elapsed times for your statistics jobs. CA Database Analyzer also provides online queries and batch reports that let you review the CA Database Analyzer statistics history.

For a basic implementation, you can tie the CA Database Analyzer extract procedure to an action procedure that contains an RS (RUNSTATS) utility code. You can then use the CA Database Analyzer online queries or batch reports menu options to view the current and past details of objects selected for statistics gathering.

For a more advanced implementation, you can tie the CA Database Analyzer extract procedure to an action procedure that contains an RS (RUNSTATS) utility code and a PR (report) utility code. The PR code lets you generate an object level CA Database Analyzer history report when the object is selected for the RS code. You can use the CA Database Analyzer Reporting Services (menu option 8) to create a report procedure using a tablespace SPACE SUMMARY report. That report procedure would be selected from the list when prompted by the Action Procedure PR Utility Code selection.

With the basic and advanced implementations, the action procedure typically does not have any action conditions selected because the extract that it is tied to contains Extract Object Conditions (EOCs) that determine the objects to be selected for statistics gathering.

Note: For an example that shows an RTOS extract object condition combined with an action procedure action condition, see the CA Database Analyzer best practice about using RTS. In the example, the extract object condition has less restrictive trigger values, and the action conditions have more restrictive trigger values. As a result, CA Database Analyzer collects historical statistics more frequently than optimizer statistics.

Business Value:

Using RTOS to limit the number of objects that are processed reduces CPU consumption and elapsed times for your statistics jobs.

Using RUNSTATS lets you collect advanced statistics (such as HISTOGRAM NUMCOLS 1 NUMQUANTILES 10). The IBM DB2 optimizer uses the statistics to pick the most efficient data access path, thus reducing transaction time (and associated cost).

Using CA Database Analyzer to collect history statistics provides online queries and batch reports that help you solve performance problems and tune objects accordingly. For example, you can determine why a REORG is running so frequently and address the problem to reduce downtime and CPU time (and associated cost).

More Information:

For information about CA Database Analyzer extract procedures (including how to tie the procedures to other procedures), see the *CA Database Analyzer User Guide*. For information about utility codes (such as RS for RUNSTATS), see the *CA Database Analyzer Reference Guide*. For information about IBM RUNSTATS, see the IBM documentation.

For information about customizing the PDA parmlib member, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide*.

The following best practices provide additional information about RTS and RTOS, respectively:

- Use Real-Time Statistics with CA Database Analyzer
- Use RTOS to Dynamically Select Objects for Maintenance

Using RTOS to Dynamically Select Objects for Maintenance

CA Database Analyzer extract procedures provide a Real Time Object Selection (RTOS) option that can dynamically select the objects that require maintenance (such as a reorganization, an image copy, or statistics collection) at execution time, instead of requiring you to statically select the objects when you create the extract procedure. Using RTOS to select objects for maintenance will automatically apply CA default settings that have been identified as the best practices for the selected maintenance action. These settings can be customized per DB2 subsystem level (site) and per procedure creator (user or SQLID).

In an RTOS-based extract procedure, objects are selected for inclusion based on the maintenance action that you specify (for example, to reorganize objects). RTOS extracts can be referred to as being action oriented, whereas classic extracts are referred to as name oriented because you select objects by name. RTOS extracts are created by specifying MODE = A (action) on the Build Extract Procedures panel and then specifying an action in the Maint Action field (such as (R)eorganization / Rebuild, (F)ull Image Copy, (I)ncr Image Copy, Image (C)opy, or (S)tatistics gathering). Classic extracts are created by specifying MODE = N (name).

You can view the CA "best practice" settings that have been applied to the extract by changing the Maint. Action value to E to explode and review the include, exclude, and auto promote object selection level options. As an example of an include option, for reorgs and image copies, all objects in a restrictive state (such as REORP or COPY) or advisory state (such as AREO* or ICOPY) are automatically included in the extract. You can select only those objects that are in a restrictive state by using the exclude EQF named RTOSDAKR (Del Advrsy & Keep Restrictive). For statistics collection, the overall concept is to collect statistics only on objects that are new or have changed.

RTOS has several powerful exclude options. Two options are automatically applied when you select (R)eorganization / Rebuild for your maintenance action: Objects smaller than two cylinders (30 tracks) are excluded, and the IBM exception table (DSNACC.EXCEPT_TBL) is evaluated to identify objects that should be excluded from reorganization. The latter option is also automatically applied when you select an image copy option or statistics collection for your maintenance action. When you select (S)tatistics gathering, both of the preceding exclude options are applied, plus objects related to volatile tables are automatically excluded. The content for these types of tables can vary from empty to very large and the DB2 optimizer can take the volatile characteristic into account without statistics. See SQL reference documentation for CREATE TABLE VOLATILE and related DB2 catalog information: SYSIBM.SYSTABLES SPLIT_ROWS=Y.

Note: The IBM exception table is user-defined and populated. For more information, see the IBM documentation. For reorganizations, CA Database Analyzer searches the QUERYTYPE column of this table for the %REORG% value. For image copies, it searches QUERYTYPE for the %COPY% value. For statistics collection, it searches QUERYTYPE for the %RUNSTATS% value.

Although an RTOS extract selects the objects that require maintenance, it must still be tied to an action procedure to generate the maintenance JCL (for reorg, copy, or statistics collection). The CA Database Analyzer best practice on using real-time statistics (RTS) contains an example that combines an RTOS extract condition with an action procedure.

Note: If you have existing action procedures for reorg, copy, or statistics collection, they can still be tied to an RTOS extract, but the action conditions should probably be removed. Action conditions are used to filter the objects to be selected, which is unnecessary in this case because the RTOS extract filters the objects.

Business Value:

RTOS extracts narrow down the objects to be processed as soon as possible, resulting in lower memory and CPU resource usage. The predefined RTOS settings also automatically apply CA default settings that have been identified as best practices for the selected maintenance action.

Additional Considerations:

- RTOS defaults:

The defaults for reorgs are set by looking at Extract Default Procedures for reorg models named RTOS\$R\$U (user or SQLID), RTOS\$R\$S (site), and RTOS\$R\$C (CA).

The defaults for image copies are set by looking at Extract Default Procedures for image copy models named RTOS\$C\$U (user or SQLID), RTOS\$C\$S (site), and RTOS\$C\$C (CA).

The defaults for statistics collection are set by looking at Extract Default Procedures for statistics gathering models named RTOS\$\$U (user or SQLID), RTOS\$\$S (site), and RTOS\$\$C (CA).

The CA Extract Default Procedures (RTOS\$_C) are shipped with CA Database Analyzer, but we strongly recommend that you copy them to site Extract Default Procedures (RTOS\$_S), even if you keep the default RTOS settings, because the CA defaults may change in the future based on DB2 changes or user input. Also, the default procedure creator (user) can be changed by specifying SQLID in the PDA parmlib member, which allows multiple customized defaults to be set up. For example, SQLID=HR or SQLID=PAYROLL could have their own RTOS defaults.

Note: For more information, type COMMAND ==> DEF (default) from the Extract Build panel.

- RTOS object selection source:

By default, CA Database Analyzer uses the IBM DB2 Real-Time Statistics (RTS) tables to select objects for maintenance actions. However, you can specify a different table by using the new conditions named "Customizable SQL condition" with a #INCLUDE <model member> statement. This statement lets you use virtually any table, such as a user-defined custom table or the DB2 catalog.

Note: To see #INCLUDE <model member> setup examples, select (T)utorial from the CA Database Analyzer main menu, select 10 (Glossary Terms), and then select S3 (Select Column Literals).

- Objects processed by RTOS extracts:

RTOS selects objects for maintenance actions by going against the entire DB2 subsystem (like the IBM DSNACCOR/X stored procedure does). You can easily limit the included objects by using the extended Extract Object Condition (EOC) option of #INCLUDE <model member>. CA Database Analyzer provides a sample RTOS reorg (RTOS\$R1C), sample image copy (RTOS\$C1C), and sample statistics collection (RTOS\$S1C) extracts that can be copied and then edited.

Note: For more information about using RTOS\$_1C RTOS preinstalled examples, press PF1 from the CA Database Analyzer main menu, select option 10 (RTOS), and then select option 4 (limiting object inclusion).

- Generating reorg JCL at tablespace level instead of partition level:

- For CA reorg utility generation, the DU Utility Code has two sub-utility codes (RR and RX) that specify whether to process one partition per step (RR) or all selected partitions in one step (RX). RX is the typical selection.

Note: For more information about sample models for RX (named DMPTIRX*), see the *CA Database Analyzer Reference Guide*.

- For IBM reorg utility generation for partitioned objects, CA Database Analyzer builds JCL that processes one partition at a time. The IBM product provides sample reorg models (MJUTLRO for offline processing and MJUTLROO for online processing).
- Starting with DB2 9, the IBM reorg utility rebuilds the NPIs instead of updating them in place. To generate reorgs at the tablespace level (instead of at partition level) so that NPIs can be rebuilt, the following must be done:
 - Remove any partition-related symbolics in the models being used.
 - Use the following CA Database Analyzer Action Procedure Action Conditions:
 - (TS) RTS Once per tablespace (part 0/n) – Condition value “;”
 - (IX) RTS Once per index (part 0/n) – Condition value “;”
 - Use the following CA Database Analyzer Extract Procedure RTOS options:
 - Promote to a TS auto-build when any TS is selected ==> Y
 - Promote to a TS auto-build when any IX is selected ==> Y

- Generating copy JCL at tablespace level instead of partition level:
 - For CA image copy utility generation, the DU Utility Code has two sub-utility codes (QC and CX) that specify whether to process one partition per step (QC), or all selected partitions in one step (CX). CX is the typical selection.
Note: For more information about sample models for CX (named DMPTICX*), see the *CA Database Analyzer Reference Guide*.
 - For IBM image copy utility generation for partitioned objects, CA Database Analyzer builds JCL that processes one partition at a time. The IBM product provides sample image copy models (MJUTLIC for partition level processing and MJUTLICP for tablespace level processing).
 - Some sites copy partitioned objects at the tablespace level to make the recovery process simpler. To generate image copies at the tablespace level (instead of at partition level), the following must be done:
 - Remove any partition-related symbolics in the model being used.
 - Use the following CA Database Analyzer Action Procedure Action Conditions:
(TS) RTS Once per tablespace (part 0/n) – Condition value “;”
(IX) RTS Once per index (part 0/n) – Condition value “;”
 - Use the following CA Database Analyzer Extract Procedure RTOS options:
Promote to a TS auto-build when any TS is selected ==> Y
Promote to a TS auto-build when any IX is selected ==> Y
- Generating statistics JCL at tablespace level instead of partition level:
 - For IBM RUNSTATS utility generation for partitioned objects, CA Database Analyzer builds JCL that processes one partition at a time. The IBM product provides a sample statistics gathering model (MJUTLRS for partition level processing).
 - Some sites collect statistics for partitioned objects at the tablespace level to make the higher level tablespace analysis simpler. To gather statistics at the tablespace level (instead of at the partition level), the following must be done:
 - Copy MJUTLRS and remove any partition level-symbolics such as %PARTLBL and %PART.
 - Use the following CA Database Analyzer Action Procedure Action Conditions:
(TS) RTS Once per tablespace (part 0/n) – Condition value “;”
(IX) RTS Once per index (part 0/n) – Condition value “;”
 - Use Extract Procedure RTOS options:
Promote to a TS auto-build when any TS is selected ==> Y
Promote to a TS auto-build when any IX is selected ==> Y

- Taking unconditional image copies:

Using RTOS may not be the best option if your goal is to take an image copy of all objects in a particular database or set of databases (for example, when performing a monthly full DR backup). In this case, none of the advanced inclusion or exclusion criteria (such as the number of pages that have changed) apply, so there is no need to use RTOS MODE=A processing. Classic extract MODE=N (name) processing may be faster.

Using RTOS to Reorganize Hash PBGs

DB2 10 introduced the hash access method, which can improve performance significantly in certain situations. The hash access method lets DB2 access a single table row directly instead of scanning the entire table or going through an index.

Use an RTOS REORG extract in CA Database Analyzer with the following extended queries to select partition-by-growth (PBG) and partition-by-range (PBR) tablespaces with hash access:

RTOSHG2X: eXclude ts Hash pbG 2+ part

Excludes hash PBG tablespaces with multiple partitions from the RTOS extract. Use this extended query to select tablespaces for reorganization at the partition level. These tablespaces can include hash PBG tablespaces with one partition, standard partitioned tablespaces, nonpartitioned tablespaces, and PBR tablespaces with one or more partitions.

RTOSHG2I: Include ts Hash pbG 2+ part

Includes hash PBG tablespaces with multiple partitions in the RTOS extract. Use this extended query to select hash PBG tablespaces with multiple partitions for reorganization at the tablespace level. This query does *not* select nonhash tablespaces with multiple partitions.

Example: Reorganize a Hash PBG Tablespace With One Partition

The following example explains how to use RTOSHG2X to reorganize a hash PBG tablespace with one partition.

1. Create an RTOS extract procedure that uses the RTOSHG2X extended query to select objects for reorganization:
 - a. Type **1** (Extract Procedures) on the CA Database Analyzer main menu. Press Enter.
 - b. Type **C** (Create) in the option field, enter an extract name in the EXT PROC field, and enter an extract description in the PROCEDURE DESCRIPTION field. Press Enter.

The BUILD EXTRACT PROCS panel appears.

- c. Type **A** in the MODE Field and press Enter.
The action fields appear.
 - d. Type **R** (Reorganization) in the Maint. Action field and press Enter.
Your selection is registered. This setting selects all objects that need reorganization, including both hash and nonhash objects.
 - e. Type **U** (Update) in the Maint. Action field and press Enter.
The BUILD EXTRACT PROCS: RTOS Mode=Act Opts panel appears.
 - f. Type **S RTOSHG2X** in the Add EQF predicate field and press PF3 (End).
The predicate is saved. This predicate removes hash PBG tablespaces with more than one partition.
2. Create an action procedure that has a REORG (RO) utility code containing a model with the %PARTLBL and %PART symbolics.
 3. Tie the RTOS extract procedure to the action procedure.
 4. Submit the extract procedure.

CA Database Analyzer generates REORG JCL with the PART 0001 syntax for the hash PBG tablespaces with one partition. If CA Database Analyzer encounters any hash PBG tablespaces with more than one partition, it uses the RTOSHG2X query to avoid generating REORG JCL for them.

Example: Reorganize a Hash PBG Tablespace With Multiple Partitions

The following example demonstrates how to use RTOSHG2I to reorganize a hash PBG tablespace with multiple partitions.

1. Create an RTOS extract procedure that uses the RTOSHG2I extended query to select objects for reorganization:
 - a. Type **1** (Extract Procedures) on the CA Database Analyzer main menu. Press Enter.
 - b. Type **C** (Create) in the option field, enter an extract name in the EXT PROC field, and enter an extract description in the PROCEDURE DESCRIPTION field. Press Enter.

The BUILD EXTRACT PROCS panel appears.
 - c. Type **A** in the MODE Field and press Enter.
The action fields appear.
 - d. Type **R** (Reorganization) in the Maint. Action field and press Enter.
Your selection is registered. This setting selects all objects that need reorganization, including both hash and nonhash objects.

- e. Type **U** (Update) in the Maint. Action field and press Enter.
The BUILD EXTRACT PROCS: RTOS Mode=Act Opts panel appears.
- f. Type **S RTOSHG2I** in the Add EQF predicate field and press Enter.
The predicate is saved. This predicate removes all nonhash objects and all hash PBG tablespaces with only one partition.
- g. Type **U** (Update) in the Extract Object Conditions field and press Enter.
The EXTRACT CONDITIONS panel appears.
- h. Find (TS) RTS reorg PBG HASH IX ents ratio. Remove the AND TS.PARTITIONS = 1 predicate. Press PF3 (End.)
You return to the previous panel.
- i. Type **Y** in the Promote to a TS auto-build when any TS is selected field. Press PF3 (End).
Your settings are saved. The Promote to a TS auto-build setting specifies that all partitions for a tablespace are selected when any single partition for that tablespace is selected.

2. Create an action procedure with the following settings:
 - Specify a REORG (RO) utility code containing a model *without* the PART keyword.
 - Specify **O** in the Condition Type field and select the following Action Condition: S (TS) RTS Once per tablespace (part 0/n).
This setting selects only the last partition for whole tablespace processing because the RO model does not use the PART keyword.

3. Tie the RTOS extract procedure to the action procedure.
4. Submit the extract procedure.

CA Database Analyzer generates REORG JCL without PART syntax for the hash PBG tablespaces with more than one partition. If CA Database Analyzer encounters any nonhash tablespace or hash PBG tablespaces with only one partition, it uses the RTOSHG2I query to avoid generating REORG JCL for them.

Business Value:

RTOS extract processing takes advantage of built-in best practices to select objects for reorganization automatically. The shipped defaults do not support a DB2 10 hash PBG tablespace with more than one partition. Following this best practice enables you to control the reorganization of these objects.

Additional Considerations:

CA Database Analyzer also provides the following EQFs to control processing of objects that have been configured for hash access. As with the two preceding EQFs, these EQFs are provided in pairs that complement each other. After you use them to create RTOS REORG extracts, you can run the extracts separately, or you can tie them together using an execution procedure and can run them together.

The following extended queries process all hash PBG tablespaces in one RTOS extract and none in the other extract:

- RTOSHGAX: eXclude ts Hash pbG All part
- RTOSHGAI: Include ts Hash pbG All part

The following extended queries process all hash PBR tablespaces in one RTOS extract and none in the other extract:

- RTOSHRAX: eXclude ts Hash pbR All part
- RTOSHRAI: Include ts Hash pbR All part

The following extended queries process all PBR and PBG tablespaces with hash indexes in one RTOS extract and none in the other extract:

- RTOSHXAX: eXclude ts Hash pbX All part
- RTOSHXAI: Include ts Hash pbX All part

Note: For more information about using EQFs and EOCs, see the *CA Database Analyzer User Guide*.

CA Insight DPM Best Practices

As a CA Insight DPM best practice, we recommend [using CA Insight DPM batch reports for system metrics and application performance monitoring](#) (see page 77).

Note: For information about CA Insight DPM installation best practices, see the "Installation Best Practices" chapter.

Using CA Insight DPM Batch Reports for System Metrics and Application Performance Monitoring

We recommend that you use the batch reports that are available with CA Insight DPM. The following reports use the DB2 SMF records for input and provide various levels of summary and detail information for system metrics and application performance:

- BTACTSML Detailed Summary of Accounting Information Report
- BTACTSMS Short Summary of Accounting Information Report
- BTACTTRL Detailed Trace of Accounting Information Report
- BTACTTRS Short Trace of Accounting Information Report
- BTCONTSM Summary of Lock Timeouts and Deadlocks Report
- BTEDMSUM EDM I/O Summary Report
- BTLKCNT Lock Timeouts and Deadlocks Report
- BTPLIOSM I/O Summary by Database, Pageset, and Plan Report
- BTPSIOSM I/O Summary by Database and Pageset Report
- BTRECTRC Detail Trace of DB2 Activity Report
- BTSCNSM1 Summary of Pagesets Scanned Report
- BTSQLSM1 SQL Summary by Statement Report
- BTSQLTRL Trace all SQL Statements Report
- BTSTASM1 DB2 System Services Address Space Statistics Report
- BTSTASM2 Summary of DB2 Database Address Space Statistics Report
- BTSTATR1 Statistics Data Trace Report
- BTSUSPSM Lock Suspension Summary Report
- BTSUSPTR Lock Suspension Details Report
- BTSYSPRM System Parameters Report

Business Value:

Running daily reports for DB2 statistics and accounting data and storing the reports in a commonly accessible place gives database administrators, systems performance programmers, and application programmers instant access to the performance data they require to do their jobs.

More Information:

For more information about the CA Insight DPM batch reports, see the *CA Insight DPM Batch Report Reference Guide*.

CA Plan Analyzer Best Practices

CA Plan Analyzer best practices include the following:

- [Find performance problems in programs during a development cycle](#) (see page 78).
- [Identify access path changes and SQL cost changes](#) (see page 81).

Finding Performance Problems in Programs During a Development Cycle

Incorporating CA Plan Analyzer into your site's program lifecycle can provide explain output that can be automatically evaluated to avoid access path changes and cost differences that can result in performance problems.

CA Plan Analyzer enhanced explain processing lets you create and explain strategies in batch. You can incorporate steps into current application bind jobs to automate a comparison of SQL, host variables, access paths, and cost. This process helps ensure that explain processing runs consistently (without intervention).

Note: This process of creating a strategy and explaining and reporting on the sources (that is, DBRMs, packages, plans) in the strategy can also be done online through the CA Plan Analyzer panels.

Implementation

Append the following JCL to your current application's bind maintenance cycle, updating the JCL to reflect your site's data set names and including additional symbolics to generate unique strategy names.

Note: The strategy name and DBRM name must have a one-to-one relationship (that is, a unique strategy will be created for each DBRM). You can achieve this by making the strategy name (defined in the STRATEGY control card) equal to the DBRM name (defined in the SRCDBRML control card).

```
//STEP1 EXEC PGM=PTLDRIVM,REGION=4M,PARM=' SUFFIX=00,EP=BPLBCTL '  
//STEPLIB DD DISP=SHR,DSN=high-level.CDBALOAD  
// DD DISP=SHR,DSN=ssid.PRIVATE.SDSNEXIT  
// DD DISP=SHR,DSN=high-level.SDSNLOAD  
//PTILIB DD DISP=SHR,DSN=high-level.CDBALOAD  
// DD DISP=SHR,DSN=ssid.PRIVATE.SDSNEXIT  
// DD DISP=SHR,DSN=high-level.SDSNLOAD
```

```

//PTIPARM DD DISP=SHR,DSN=high-level.CDBAPARM
//PTIXMSG DD DISP=SHR,DSN=high-level.CDBAXMSG
//SYSOUT DD SYSOUT=*
//PTIIMG DD SYSOUT=*
//UTPRINT DD SYSOUT=*
//ABNLIGNR DD DUMMY SUPPRESS ABENDAID DUMPS
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(30,30))
//SYSREC DD UNIT=SYSDA,SPACE=(TRK,(1,1))
//BPIIPT DD *
.CALL EXPLAIN
.DATA
RULESSID = (ssid)
ACM = (N,userid)
STRATEGY = (ssid,strategyname,userid,@AUTO)
PLANTAB = (ROLLBACK)
SQLQUAL = (userid)
CASE = (UPPER)
LINES = (55)
PROCDDF = (N)
FLOATFMT = (SCI)
SAVEHOST = (Y)
PROCVIEW = (N)
LINES = (55)
PROCDDF = (N)
FLOATFMT = (SCI)
SAVEHOST = (Y)
PROCVIEW = (N)
EXPLTYPE = (FUTURE)
ISOLATE = (CS)
SAVERPTS = (Y)
VSAM = (N)
TARGET = (ssid(@DEFAULT))
PERFTIE = (N)
DATABASE = (Y,Y,PRIOR,B,Y)
REPORT = (SUMMARY,COST,ACCESS/SHORT,SQLRULE,PHYSRULE,PREDRULE,
COMPARE)
COMPOPTS = (BBNBBB CN,PRIOR,N,,)
SRCDBRML = (dbrmlib name(dbrmname),)
.ENDDATA
//BPIOPT DD *
.CONTROL BPID(PPACMP) +
LOGID(ssid) UNIT(SYSDA)
.LIST SYSOUT(*)
.OPTION NOERRORS SQLERRORS RETRY(10) NOBINDERERRORS
.CONNECT ssid

```

The preceding JCL includes the following commands:

STRATEGY

Identifies the subsystem, strategy name, and strategy creator.

Important! Make the strategy name equal to the DBRM name.

DATABASE

Defines a manual *Hversion*, which refers to the historical version and identifies the data in the CA Plan Analyzer historical database for retrieval for reports, such as the Compare Explain Versions report and subsequent analysis.

In this scenario, the CURRENT Hversion, if it exists, is used to create an Hversion named PRIOR, before the explain begins. Then, the explain process creates a new CURRENT, which is compared to PRIOR.

REPORT

Indicates which reports to generate, including the comparison report.

COMPOPTS

Defines the options used for the compare process. Additional parameters are available to limit the display to SQL whose cost differences exceed defined thresholds.

SRCDBRML

Defines the DBRM library and member.

Results

The first execution of the JCL generates explain output, creates an explain strategy (if it does not already exist), and adds a version to the strategy. Subsequent executions will also generate explain output, compare this output to the previous execution, and add a version to the existing strategy.

The REPORT control card in the preceding JCL generates several reports, including the Summary, Cost, and Compare Explain Versions reports.

The Compare Explain Versions report shows changed/unchanged SQL, changed/unchanged access paths, paired/unpaired SQL statements, and cost differences. These options are controlled through the COMPOPTS control card.

You can analyze these reports to address any performance problems.

Specifying an Hversion through the DATABASE control card lets you view additional online reports from these options:

- Query EXPLAIN Database
- Compare EXPLAIN Versions
- Identify Problem SQL

Business Value:

CA Plan Analyzer enhanced explain processing helps prevent performance problems. Avoiding the performance problems translates to resource savings (such as CPU and I/O).

More Information:

For more information about this process, see the *CA Plan Analyzer User Guide*.

How to Identify Access Path Changes and SQL Cost Changes

As a database administrator (DBA), you want to identify access path changes and changes in SQL costs, such as milliseconds, service units, and total costs because the access path changes can result in costly, poorly performing SQL. These changes can occur in several situations, for example:

- Migrating to a new version of DB2
- DB2 object changes, such as adding or dropping an index
- SQL changes, such as additional predicates or changes to ORDER BY/GROUP BY

You can configure CA Plan Analyzer to perform compare processing and generate reports that identify access path changes and changes in SQL costs. Many filtering options are provided so you can limit your report data. For example, you can limit your results to report only on packages where the SQL costs increased by three percent.

By comparing explain data and identifying the projected access path and SQL cost changes, you can assess potential problems and can take appropriate action. For example, you could decide to proceed with your proposed changes. Alternatively, you could decide to use other CA Plan Analyzer utilities and reports to study the comparison results in more depth before proceeding with your proposed changes.

Note: For more information about how to identify access path changes and SQL cost changes, see the *CA Plan Analyzer User Guide*.

CA Rapid Reorg Best Practices

CA Rapid Reorg best practices include the following:

- Maximize DATA-AVAILABLE CRITICAL reorgs.
- Maximize DATA-AVAILABLE HIGH reorgs.
- Maximize DATA-AVAILABLE NORMAL reorgs.

How to Maximize DATA-AVAILABLE CRITICAL Reorgs in CA Rapid Reorg

We recommend that you review your DATA-AVAILABLE CRITICAL reorg jobs in CA Rapid Reorg and make the following changes to maximize performance:

- Enable sort to dynamically allocate sort workspace.
- Create an image copy during the reorg. The copy occurs while the indexes are being rebuilt. You can also use COPY-TASKS to multi-task the copy.
- Keep statistics reasonably accurate so that CA Rapid Reorg can make a reasonable estimate of the current number of rows in the object.
- Use FASTSWITCH YES for partitioned objects or if the object is in a database comprising mainly partitioned objects.
- Use FASTSWITCH NO when reorganizing segmented tablespaces.
- Use the RIDDISP keyword instead of the RID nnn data sets for small to medium objects so that the mapping RIDs can be stored in a dataspace. This eliminates the need to sort the RIDs and to perform I/Os to the RID nnn data set, which helps improve the performance of the reorganization and log apply phases.
- Use MAXTASKS and UNLOAD-TASKS to help improve the concurrency of the reorg tasks.

Business Value:

Implementing these changes in your CA Rapid Reorg jobs will help reduce I/O and reduce the need for large amounts of secondary storage.

Additional Considerations:

DATA-AVAILABLE CRITICAL reorgs are designed to be used when maximum availability is desired during reorganizations.

If a failure occurs during the unload, reload, or update phases of the reorg, you do not need to perform an object recovery because only shadow data sets are updated until the time of the switch phase; no object updates are performed.

More Information:

For more information about CA Rapid Reorg keywords, see the *CA Rapid Reorg User Guide*.

How to Maximize DATA-AVAILABLE HIGH Reorgs in CA Rapid Reorg

We recommend that you review your DATA-AVAILABLE HIGH reorg jobs in CA Rapid Reorg and make the following changes to maximize performance:

- Enable sort to dynamically allocate sort workspace.
- Create an image copy during the reorg. The copy occurs while the indexes are being rebuilt. You can also use COPY-TASKS to multi-task the copy.
- Keep statistics reasonably accurate so that CA Rapid Reorg can make a reasonable estimate of the current number of rows in the object.
- Use FASTSWITCH YES for partitioned objects or if the object is in a database comprising mainly partitioned objects.
- Use FASTSWITCH NO when reorganizing segmented tablespaces.
- Use EXTENDED-TSREORG YES for small to medium size objects.
Note: Extended mode is available *only* for tablespaces with only one table.
- Use UNLOAD-TASKS and REBUILD-TASKS to improve concurrency of the reorg tasks in extended mode reorgs. Use MAXTASKS for standard mode reorgs.

Business Value:

Implementing these changes in your CA Rapid Reorg jobs will help reduce I/O and reduce the need for large amounts of secondary storage.

Additional Considerations:

DATA-AVAILABLE HIGH reorgs are designed to be used when applications need read-only availability during reorganizations. Both standard and extended modes are available with this option.

More Information:

For more information about CA Rapid Reorg keywords, see the *CA Rapid Reorg User Guide*.

How to Maximize DATA-AVAILABLE NORMAL Reorgs in CA Rapid Reorg

We recommend that you review your DATA-AVAILABLE NORMAL reorg jobs in CA Rapid Reorg and make the following changes to maximize performance:

- Use EXTENDED-TSREORG YES for small to medium size objects.
Note: Extended mode is available *only* for tablespaces with only one table.
- Use UNLOAD-TASKS and REBUILD-TASKS to improve concurrency of the reorg tasks in extended mode reorgs. Use MAXTASKS for standard mode reorgs.

- Use REBUILD-INDEX YES or consider using REBUILD-INDEX NOPI with EXTENDED-TSREORG YES for partition level reorgs where you do not require concurrency. Use REBUILD-INDEX NO for non-unique non-partitioned indexes.
- Specify at least the same number of work data sets (SYSUT1xx) plus one as the number of unload tasks for better performance.

Business Value:

Implementing these changes in your CA Rapid Reorg jobs will help reduce I/O and reduce the need for large amounts of secondary storage.

Additional Considerations:

DATA-AVAILABLE NORMAL reorgs are designed to be used when object availability is not an issue. Both standard and extended modes are available with this option.

More Information:

For more information about CA Rapid Reorg keywords, see the *CA Rapid Reorg User Guide*.

Chapter 5: Database Administration Best Practices

This section contains the following topics:

[General Database Administration Best Practices](#) (see page 85)

[CA RC/Migrator Best Practices](#) (see page 88)

General Database Administration Best Practices

General database administration best practices include the following:

- Maximize CA Fast Index performance.
- Maximize CA Fast Load performance.
- Maximize CA Fast Unload performance.

How to Maximize CA Fast Index Performance

To maximize CA Fast Index performance, review your CA Fast Index jobs and make the following changes:

- Enable sort to dynamically allocate sort workspace.
- Keep statistics reasonably accurate so that CA Fast Index can make a better estimate of the current number of rows in the object.
- Adjust MAXTASKS to control the number of partitions to be multi-tasked when reading key information from the table. Enter a number less than or equal to the number of partitions in the tablespace.
- Adjust MAXSORTS to control the maximum number of concurrent sort tasks of indexes.
- Use UPDATE-CATSTATS to update index statistics in the DB2 catalog.

Business Value:

Implementing these changes in your CA Fast Index jobs will help reduce I/O and reduce the need for large amounts of secondary storage.

More Information:

For additional performance tips, see the *CA Fast Index User Guide*.

How to Maximize CA Fast Load Performance

To maximize CA Fast Load performance, review your CA Fast Load jobs and make the following changes:

- Enable sort to dynamically allocate sort workspace.
- Use OUTPUT-CONTROL BUILD to take input and build the tablespace directly.
Note: Jobs will terminate if CA Fast Load finds duplicate keys for a unique index or if RECLUSTER NO is specified and CA Fast Load finds out-of-sequence keys for the clustering index. If RECLUSTER YES or RECLUSTER NO SORT-CLINDX is specified, an out-of-sequence index entry does not terminate the job. Normal processing takes care of any other errors and the load continues.
- Use RECLUSTER NO if the data is already in clustered order.
- Keep statistics reasonably accurate so that CA Fast Load can make a reasonable estimate of the current number of rows in the object.
- Use multiple SYSULDxx statements if your data is already segregated into separate data sets by partition.
- Create an image copy during the load. The copy occurs while the indexes are being rebuilt. You can also use COPY-TASKS to multi-task the copy.
- Adjust the MAXTASKS parameter to control the number of simultaneous tasks used for indexes.

Business Value:

Implementing these changes in your CA Fast Load jobs will help reduce I/O and reduce the need for large amounts of secondary storage.

More Information:

For more information about the control statements and other options you can use to maximize CA Fast Load performance, see the *CA Fast Load User Guide*.

How to Maximize CA Fast Unload Performance

To maximize CA Fast Unload performance, review your CA Fast Unload jobs and consider making the following changes:

Note: For recommended settings that apply to the entire installation, set them in the PFU member of *high-level.CDBAPARM* when possible so that users do not have to code the recommended commands in individual jobs.

- Enable sort to dynamically allocate sort work data sets for processing ORDER BY clauses in SELECT statements.
- Specify COPY-BUFFERS, IO-BUFFERS, and VSAM-BUFFERS if the default values are not appropriate for your use.
- Use EXCP MM as the default for maximum speed and flexibility. The alternative is EXCP NO, which uses VSAM for access. EXCP YES is no longer available.
- Use CURRENT-DEGREE ANY and MULTI-ROW-FETCH to improve performance with SQL-ACCESS ONLY. MULTI-ROW-FETCH is enabled by default.
- Use OUTPUT-FORMAT LOAD for best performance. However, when unloading long variable length columns, OUTPUT-FORMAT VARIABLE is usually more appropriate.
- Use data translation only as needed because it can impact performance.
- Use data conversion (that is, the INTO clause of the SELECT statement) only as needed because it can impact performance.
- If INPUT-FORMAT TABLE and SQL-ACCESS NONE or EXTENSION are used and the result is non-SQL access of table data, use several SELECT statements that each specify a different range of partitions, and specify PART-INDEPENDENCE YES. CA Fast Unload can then scan the partitions in separate, parallel tasks.

Note: CA Fast Unload writes the SYSREC output to one data set per SELECT statement, unless you specify ONE-SYSREC.

Business Value:

Improving CA Fast Unload performance will help improve data availability for affected DB2 tables and reduce the costs associated with the unload process.

More Information:

For more information about these settings, see the *CA Fast Unload User Guide*.

CA RC/Migrator Best Practices

CA RC/Migrator best practices include the following:

- Reduce DB2 catalog contention and overhead.
- Review analysis options.
- Use CA RC/Migrator to retain current DDL.

Reducing DB2 Catalog Contention and Overhead

Use CA RC/Migrator alternate catalog access methods to reduce DB2 catalog contention and processing overhead due to CA RC/Migrator queries. CA RC/Migrator provides three alternate catalog access methods:

Catalog PDS Unload

Helps reduce DB2 catalog query overhead. The Catalog PDS Unload data sets contain DB2 object information formatted in an internal structure that CA RC/Migrator uses to identify DB2 objects for migration, alteration, and comparison strategies. CA RC/Migrator reads the catalog once to create the catalog PDS with a subset of DB2 objects; subsequent catalog queries use the catalog PDS to identify candidate objects for a strategy. CA RC/Migrator queries the DB2 catalog to retrieve object details that are necessary to properly analyze the strategy.

To use Catalog PDS Unload, access the Catalog Unload JCL Interface panel through the Catalog PDS Unload option on the Expert Profile Menu. From this panel, you can unload the DB2 catalog into a Catalog Unload PDS. Update the MIGRATOR parmlib member and specify this PDS name in the PDSDSN() parameter for the DB2 subsystem whose catalog was unloaded.

The unloaded catalog data set is created at a point in time from the metadata in the DB2 catalog. Over time, this information can become stale, possibly leading to incorrect DDL in CA RC/Migrator strategy analysis scripts. The catalog data sets should be reloaded periodically, preferably daily.

Alternate Catalog Mapping (ACM)

Identifies DB2 objects for strategy creation and analysis by using shadow tables only, views only, or views *and* shadow tables. Each method has unique characteristics that make it suitable for a particular environment; each method can help reduce or eliminate the need for CA RC/Migrator users to have authority to query the DB2 catalog and can also limit the number of rows accessed in the DB2 catalog.

ACM shadow tables create tables and extract data from the DB2 catalog into those tables. The duplication of the DB2 catalog requires additional system and DASD resources, and the metadata in the tables can become obsolete and must be reloaded frequently. However, queries are directed at the shadow tables, reducing overhead and contention for the real DB2 catalog.

ACM views can help improve CA RC/Migrator performance at the minor cost of additional DB2 catalog resources to define the view. Use SQL query predicates on the ACM views to restrict the objects returned for a query and ease contention on the DB2 catalog by limiting the number of rows queried. This helps reduce the amount of DB2 and system resources required to represent a strategy in memory and reduce CA RC/Migrator processing overhead. Because the views are directly on the catalog, the metadata used by CA RC/Migrator is always current, and no additional tables or system resources are required.

Limited Tree Processing

Restricts available DB2 objects for strategies to objects within databases involved in the strategy. This reduces the amount of data returned by DB2, reducing the storage and processor resources required to analyze a strategy.

Note: Limited tree processing is ignored for compare type strategies, strategies in which a storage group is being dropped, and view processing.

The methods can be used individually or used together to maximize benefits. The method that you choose depends on your strategy needs and how much toleration you have for using system resources.

Business Value:

Using the alternate catalog access methods to restrict CA RC/Migrator use of the DB2 catalog helps reduce DB2 catalog contention and overhead caused by the queries needed for strategy creation and analysis.

Additional Considerations:

Using Catalog PDS Unload and ACM together can help improve performance. CA RC/Migrator uses the PDS to build a picture of the DB2 catalog and to determine each object's parent and child objects. To get a complete definition of objects, CA RC/Migrator then performs all SQL against the alternate catalog. The PDS should be created from the ACM catalog if you plan to use ACM and PDS together in an analysis.

More Information:

For more information about Catalog PDS Unload, ACM, and limited tree processing, see the *CA RC/Migrator User Guide*.

Review Analysis Options

Always review your analysis options prior to performing analysis on a strategy.

To review your analysis options

1. Enter **Y** in the UPDATE OPTIONS field on the Strategy Analysis panel.
The Strategy Analysis Options panel appears.
2. Review each setting for its applicability to the strategy that you are analyzing.
This helps ensure that you receive the expected results.

Business Value:

Reviewing your analysis options helps ensure that CA RC/Migrator performs the strategy analysis as expected and generates complete and accurate DDL scripts. This helps avoid wasting resources by re-analyzing strategies.

Additional Considerations:

The Analysis Options panel contains several different sections. Review each section carefully. Unless you save changes to analysis options, you will lose the option changes when you exit your CA RC/Migrator session. To verify that option settings stay in effect the next time you start CA RC/Migrator, use the SAVE primary command.

The analysis options often work with strategy options. For example, you can specify the AUX IMP (auxiliary implode) option in the strategy definition and in the analysis options.

Note: When a conflict exists among the parameters, the analysis options take precedence. You can set the strategy option to act as a default if the analysis option has not been set. For this reason, it is important to review the analysis options before performing the strategy analysis.

CA RC/Migrator evaluates the options you specify on the analysis options panel with regard to the work to be done. Certain actions may not be performed, regardless of the options set. For example, you can specify that a reorg should be done following drop-create processing in the script, but CA RC/Migrator may not generate a reorg if it is not necessary due to the actions taken in the DDL script.

How to Use CA RC/Migrator to Retain Current DDL

Use CA RC/Migrator to keep current DDL scripts that can redefine your applications in an emergency. CA RC/Migrator reads the DB2 catalog to identify relationships between objects and generates DDL that can redefine your application's DB2 objects. Keeping current DDL scripts helps ensure that you have a complete set of DDL to recover a lost or damaged system.

To generate a complete set of DDL for your application, perform the following steps:

1. Create a strategy to migrate the application objects.

Start at the highest object level (for example, the database level). Select all objects associated with that object. For example, place an **A** in the line command for the application database to include all lower-level hierarchical objects.

2. Perform an analysis on the strategy, being sure to do the following:

- a. Verify that the analysis control options include all objects or object attributes (foreign keys) that might not otherwise be included as part of the strategy.

For example, set AUX Implode to include all possible parents of views, indexes, and tablespaces that might exist in other databases.

- b. (Recommended) Set the BND/DAT/STA/SQL/GRNT/RI field to **S** to cause only SQL to be written (but not include Batch Processor commands in the DDL script).
- c. Specify to save the analysis output into a permanently allocated data set.

Business Value:

Retaining complete application DDL protects your application resources in case of lost or damaged DB2 resources. Also, you can use the created DDL to recreate the application objects on other DB2 subsystems.

Additional Considerations:

CA RC/Migrator provides two options for saving the generated DDL: You can store the DDL in a data set or use the managed output option to save the DDL in a DB2 table. Saving the DDL in a data set provides the greatest safety and availability. One good idea is to save the DDL in CA Endeavor SCM, which protects the DDL and tracks changes. You can view the historical changes to the DDL and recover to a point in time, if needed.

It is also a good idea to save the JCL that analyzes the strategy. This lets you recreate the DDL at any time by running the job again. Also, you can enter the JCL into a job automation system to run it automatically at defined intervals.

Note: To save the JCL, enter **D** in the DESTINATION field on the Batch JCL Specification panel, and use OUTPUT JCL DATA SET SPECIFICATIONS to specify the name of a data set and member to receive the generated JCL.

Chapter 6: Backup and Recovery Best Practices

This section contains the following topics:

- [CA Fast Check Best Practices](#) (see page 93)
- [CA Fast Recover Best Practices](#) (see page 94)
- [CA Merge/Modify Best Practices](#) (see page 95)
- [CA Log Analyzer Best Practices](#) (see page 96)
- [CA Quick Copy Best Practices](#) (see page 98)
- [CA Recovery Analyzer Best Practices](#) (see page 101)

CA Fast Check Best Practices

CA Fast Check best practices include the following:

- Use CA Fast Check DELETE statements.
- Maximize CA Fast Check performance.

Using CA Fast Check DELETE Statements

When DB2 executes the delete process using a foreign key and no indexes, it performs tablespace scans for each DELETE statement. These scans result in increased processing times. Therefore, for performance reasons, we recommend that you use CA Fast Check DELETE statements to perform the same task.

The CA Fast Check DELETE FOREIGN-KEY keyword generates DELETE statements based strictly on each foreign key without the requirement of indexes. This lets you correlate the delete to the parent table from the columns in the DELETE statement. When you specify DELETE FOREIGN-KEY and CA Fast Check finds an RI violation, it generates DELETE statements with the predicate using the columns in the foreign key.

This procedure occurs regardless of any process that CA Fast Check uses (DSP, sort, or index on foreign key). It also occurs even if the table has a unique index.

Business Value:

Using CA Fast Check instead of DB2 to generate DELETE statements that delete rows in violation of referential or table check constraints from the tablespace can result in decreased processing times.

More Information:

For more information about the DELETE FOREIGN-KEY option, see the *CA Fast Check User Guide*.

How to Maximize CA Fast Check Performance

To maximize CA Fast Check performance, review your CA Fast Check jobs and make the following changes:

- Specify DATASPACE YES in your CHECK DATA jobs to use a dataspace instead of sort when the index size is less than 2 GB. DATASPACE YES is the default, so verify that it is not disabled.
- Define a secondary index on large child tables when foreign keys exist so that CA Fast Check can compare the indexes directly.

Business Value:

Specifying DATASPACE YES when the index size is less than 2 GB can improve CPU performance because it lets CA Fast Check load the foreign index into a binary tree in a dataspace instead of using sort to match the foreign key in the data rows with the foreign index.

Defining a secondary index for a child table can help improve CPU performance because the secondary index lets CA Fast Check avoid reading and building the foreign key from the data row and sorting it into key sequence before comparing it to the foreign index.

More Information:

For more information about CA Fast Check syntax options, see the *CA Fast Check User Guide*.

CA Fast Recover Best Practices

As a CA Fast Recover best practice, we recommend [reviewing your jobs and making specific changes that will improve CA Fast Recover performance](#) (see page 95).

How to Improve CA Fast Recover Performance

To improve CA Fast Recover performance, review your CA Fast Recover jobs and make the following changes:

- Use a single RECOVER statement for all objects. If you use tape stacking for your image copy data sets, specify RETAIN YES.
- Use SORT-LOG YES or SORT-LOG CONCURRENT for your recovery jobs.
- When using SORT-LOG YES, recover a tablespace and rebuild its indexes in the same job step.

Business Value:

CA Fast Recover reads the data sets once for each RECOVER statement; therefore, using a single RECOVER statement lets CA Fast Recover process the DB2 recovery log more efficiently. Specifying RETAIN YES lets CA Fast Recover optimize the recovery order of the objects and mount a VOLSER only once during the RECOVER statement.

The SORT-LOG options let you sort the log records in object order. An object order sort lets CA Fast Recover make better use of system resources because it can close the data sets after processing all log records for an object.

Combining RECOVER and REBUILD statements in the same job step while using SORT-LOG YES lets CA Fast Recover make a single pass through the tablespace to rebuild the index, which decreases processing time.

More Information:

For more information about these options, see the *CA Fast Recover User Guide*.

CA Merge/Modify Best Practices

As a CA Merge/Modify best practice, we recommend [reviewing your jobs and making specific changes to improve CA Merge/Modify performance](#) (see page 95).

How to Improve CA Merge/Modify Performance

To improve CA Merge/Modify performance, review your CA Merge/Modify jobs and make the following changes:

- Use a single ACCUMULATE or ACCUMULATE CHANGES statement for all objects.
- Specify RETAIN YES.

Business Value:

CA Merge/Modify reads the data sets once for each ACCUMULATE or ACCUMULATE CHANGES statement; therefore, using a single statement for all objects lets CA Merge/Modify process the DB2 recovery log more efficiently.

Specifying RETAIN YES lets CA Merge/Modify optimize the recovery order of the objects and mount a VOLSER only once during the ACCUMULATE or ACCUMULATE CHANGES statement.

More Information:

For more information about these options, see the *CA Merge/Modify User Guide*.

CA Log Analyzer Best Practices

CA Log Analyzer best practices include the following:

- Enable DATA CAPTURE CHANGES to improve CA Log Analyzer performance.
- Use DB2 logs for compliance reporting.

How to Improve CA Log Analyzer Performance

To improve CA Log Analyzer performance, enable DATA CAPTURE CHANGES on your tables.

Note: You can use the Change Data Capture (CDC) Analysis report to determine how enabling or disabling DATA CAPTURE CHANGES will increase or decrease the amount of data written to your log data sets.

Business Value:

Using DATA CAPTURE CHANGES on your tables causes DB2 to log full row images for UPDATE statements. CA Log Analyzer performance is improved because it no longer needs to read image copy data sets to resolve incomplete row data.

More Information:

For more information about DATA CAPTURE CHANGES and the CDC Analysis report, see the *CA Log Analyzer User Guide*.

Using DB2 Logs for Compliance Reporting

Use the DB2 logs for compliance reporting. Part of the purpose of auditing production changes is to identify incorrectly assigned authorities and security problems, such as Trojan horses, where unauthorized users change shared procedures and insert code to run when these procedures are executed by users with more authority. The only true record of these changes is in the DB2 logs.

You can write extraction jobs to provide your auditors with a method to extract the data they need from the DB2 logs, or you can use CA Log Analyzer, which comes with common compliance reports that track the following:

- All DDL changes
- Grant/revoke activity
- Failed access attempts
- SMF audit trace reporting

Business Value:

The integrity of the DB2 subsystem is essential for many business-critical applications, including human resources and financial data. Failing to detect unauthorized changes to these systems could be a catastrophic event, with legal repercussions. Auditing from the DB2 log can potentially generate more accurate and comprehensive reports. Other auditing processes may assume that authorities are correctly assigned and that internal or external hackers cannot circumvent security.

In addition, using the DB2 logs eliminates human error that may be involved if database administrators are required to audit their own actions in a data set that contains all DDL changes that have been made in production. If they forget even once, the data set will not be accurate, whereas the DB2 log is maintained by the system and is designed to always be accurate.

Additional Considerations:

CA Log Analyzer lets you determine how data was changed in DB2 tables and who changed it. You can use the following CA Log Analyzer features:

- UNDO DDL to recover dropped objects

This feature recreates objects that have been dropped accidentally, and integrates with CA RC/Migrator to reverse the effects of an ALTER strategy and generate an application fallback.

- Change Data Capture analysis feature to determine impact on the DB2 log

This feature generates a report that lets you monitor when a table's data capture attribute has been changed and determines what transactions occurred within the duration of that data capture status. A table's data capture attribute dictates how much data is logged for an update transaction to that table. When you use DATA CAPTURE NONE (DCN), the log record contains only the changed data. When you use DATA CAPTURE CHANGES (DCC), the log record contains enough information to build a full before- and after-image.

- Recovery Point analysis to generate quiesce points for recovery purposes

This feature lets you identify safe recovery points in the DB2 log and create a report that lists points in the DB2 log where no update activity was occurring against the objects you are interested in. This can help you choose an appropriate point at which to recover data.

More Information:

For more information about using CA Log Analyzer to create accurate reports for compliance monitoring, see the *CA Log Analyzer User Guide*.

CA Quick Copy Best Practices

CA Quick Copy best practices include the following:

- Maximize CA Quick Copy performance.
- Review use of Snapshot General Services.
- Use wildcards to select multiple objects for copy.

How to Maximize CA Quick Copy Performance

To maximize CA Quick Copy performance, review your CA Quick Copy jobs and make the following changes:

- Combine multistep jobs or multistatement jobs into a single CA Quick Copy statement that specifies multiple objects, a wildcarded set, or a LISTDEF, and review use of the COPY-TASKS setting. Set COPY-TASKS to correspond with your recovery requirements for output image copies and tape stacking.
- Use a single COPY statement to copy a base tablespace and its auxiliary tablespaces.

- Review use of the MODIFYBITS option as follows:
 - If you take full copies, set MODIFYBITS to NO. Time is saved because CA Quick Copy does not need to stop the tablespaces or update the tablespace space map pages.
 - If you take weekly full copies and merge daily incremental copies, set MODIFYBITS to YES to reset all modification bits for the tablespace in the full copy set. Subsequently, whenever a row is updated, added, or deleted, the modification bit is set. For incremental image copies, set MODIFYBITS to NO. Each incremental copy contains all the changes since the last full image copy.

Note: If you specify MERGE-COPY YES in the incremental copy jobs, the previous incremental image copy is deleted from SYSIBM.SYSCOPY. If you always take your incremental image copies with MODIFYBITS set to NO, a recover utility can use this single incremental copy with the full image copy.

- If you take weekly full copies and daily incremental copies with many changes, set MODIFYBITS to YES for the full copy set and incremental copies set. Each incremental copy will contain the changes since the last image copy, full or incremental. The full copy and all subsequent incremental copies are needed for recovery.

If you cannot use MODIFYBITS YES for your full copy, specify MODIFYBITS NO for the full copy and specify LRSNCOPY for the incremental copies. Each incremental copy will contain the changes since the last image copy, full or incremental.

Business Value:

Because there is a significant amount of overhead with the initialization of a COPY statement, you can get more throughput with CA Quick Copy when you copy many objects with a single COPY statement. By copying many objects in a single statement (as opposed to a single object per statement), you eliminate the repetitive initialization processing.

Additional Considerations:

Objects become unavailable to applications with SHRLEVEL REFERENCE. At the start of a COPY statement, all objects specified for a given statement are placed in RO status. Each object becomes available for other processing when its copy is complete.

You need to determine the acceptable level of performance improvement and number of DB2 objects that become unavailable to applications for SHRLEVEL REFERENCE. If SHRLEVEL CHANGE is specified, the status of an object is not changed for the copy. After CA Quick Copy completes the initialization phase, the copying process is quick. CA Quick Copy simply schedules each object to one or more copy tasks.

When setting up your jobs for creating image copies, consider how you set up your jobs for a recovery. If image copies are registered at the partition level, the recovery job must recover the object at the partition level. When taking image copies, be consistent with respect to how copies are registered. You should always take copies at the partition level, or always take a DSNUM ALL copy where only a single image copy data set is registered. This can impact the MERGE-COPY YES processing for incremental copies.

More Information:

For more information about the CA Quick Copy syntax options, see the *CA Quick Copy User Guide*.

Reviewing Use of Snapshot General Services

Review the following options for Snapshot General Services:

- Review your ability to quiesce your objects. If possible, use SNAPSHOTACCESS YES with SHRLEVEL REFERENCE to take quick, consistent copies.
- Use SNAPSHOTACCESS INSTANT to take quick, hardware-assisted backups right before critical processing. When you use this option to take copies, you can then specify SNAPSHOTACCESS INSTANT in a CA Fast Recover job to perform hardware-assisted recoveries, if necessary. You can also use CA Merge/Modify COPY-IMAGE-COPY LAST-INSTANT to convert your SNAPSHOTACCESS INSTANT copies into standard image copies.

Business Value:

By taking snapshot copies, you can reduce the amount of time that your DB2 tablespaces are unavailable to applications. The snapshot image copies can be used by CA Fast Recover.

More Information:

For more information about Snapshot General Services, see the *CA Quick Copy User Guide*. For more information about the SNAPSHOTACCESS option in CA Fast Recover and CA Merge/Modify, see the *CA Fast Recover User Guide* and the *CA Merge/Modify User Guide*.

Using Wildcards to Select Multiple Objects for Copy

We recommend that you use wildcards in your CA Quick Copy COPY statements to quickly select multiple objects with similar names and qualifiers for copy.

You can implement wildcarding by specifying an underscore sign (`_`), asterisk (`*`), or percent sign (`%`) in your COPY statement to copy specific databases and tablespaces. The percent sign (`%`) or asterisk sign (`*`) indicates any string of zero or more characters. The asterisk (`*`) and percent sign (`%`) characters are interchangeable. An underscore (`_`) indicates any single character.

Note: You can also use COPY LIST and provide the name of a LISTDEF. A LISTDEF control statement can include wildcarded tablespaces, wildcarded indexspaces, and objects that are referentially related. You can specify the EXCLUDE and CLONED keywords to filter the list.

When you use wildcarding, you must dynamically allocate your image copy data sets. Invoke dynamic allocation by removing all image copy DD statements from your CA Quick Copy job. When CA Quick Copy processes the job, it uses the dynamic allocation parameters that have been specified in the PQC member of *high-level.CDBAPARM*.

Business Value:

Wildcarding reduces overhead by eliminating the need to update production jobs when adding or dropping tablespaces to your DB2 applications. CA Quick Copy dynamically allocates the image copy data sets, saving you time because you no longer need to specify the required DD statements in your jobs.

More Information:

For more information about wildcarding and dynamic allocation of data sets, see the *CA Quick Copy User Guide*.

CA Recovery Analyzer Best Practices

CA Recovery Analyzer best practices include the following:

- Improve CA Recovery Analyzer performance.
- Schedule PRA#LOAD jobs to execute frequently.

How to Improve CA Recovery Analyzer Performance

To improve CA Recovery Analyzer performance, perform the following tasks:

- Collect catalog statistics on the tables used by CA Recovery Analyzer, and then rebind the packages.
- Review strategy use and regenerate older strategies.

Business Value:

CA Recovery Analyzer benefits from current statistics just like any other DB2 application. Regenerating older strategies will help improve performance.

More Information:

For more information about generating recovery strategies, see the *CA Recovery Analyzer User Guide*.

Scheduling PRA#LOAD Jobs to Execute Frequently

Schedule your PRA#LOAD jobs to execute frequently enough so that newly created objects are detected. Be sure to run PRA#LOAD at least every night.

If any of the following circumstances exist, run the PRA#LOAD batch job periodically during the day so that the VTOC information remains current:

- Users create tablespaces during the day.
- Information flow is so heavy that tablespaces go into new extents during the day.
- A DASD device is disabled or is no longer available. If the device is not available to retrieve space information, the old quantities are used.

Business Value:

CA Recovery Analyzer must have access to current system information to effectively perform recoveries. PRA#LOAD retrieves this system information and saves it in product-specific versions of the DB2 tables. CA Recovery Analyzer uses these tables to quickly perform analysis instead of accessing the DB2 catalog directly. Running PRA#LOAD frequently keeps these tables up to date and helps CA Recovery Analyzer recover objects quickly and effectively.

Additional Considerations:

Before running PRA#LOAD for the first time, verify that the generation data group (GDG) base data sets needed for the PRA#COPY step of the PRA#LOAD job have been created. The base data sets are created by running the PRA#DGDG job.

When upgrading to a new release of CA Recovery Analyzer, regenerate all PRA#LOAD jobs for all DB2 systems to help ensure the jobs are consistent with release changes.

Important! If you change the names for the product suite database (the default name is PTDB) or storage group (the default name is PTSG), verify that they are valid in the SETUPxx member for the target DB2 system before running PRA#LOAD. For more information about valid SETUP member entries, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide*.

More Information:

For more information about the PRA#LOAD job, see the *CA Recovery Analyzer User Guide*.

Chapter 7: Integration Points

This chapter describes the CA Database Management Solutions for DB2 for z/OS integrations that are currently available in support of the CA Enterprise IT Management Strategy (EITM).

Note: Appropriate licensing is assumed to enable this integration.

This section contains the following topics:

[Integrate with CA Service Desk](#) (see page 105)

[Integration of CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer](#) (see page 106)

[Integration of CA Quick Copy and CA Rapid Reorg](#) (see page 106)

[Integration of CA Quick Copy and CA Merge/Modify](#) (see page 107)

[Integration of CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload](#) (see page 107)

[Integration of CA Fast Load, CA Quick Copy, and CA Rapid Reorg](#) (see page 108)

[Integration of CA Bind Analyzer, CA Detector, and CA Plan Analyzer](#) (see page 109)

[Integration of CA Fast Recover, CA Merge/Modify, and CA LogCompress](#) (see page 109)

[Integration of CA Insight DPM and CA SYSVIEW PM](#) (see page 110)

[CA RC/Query Integration](#) (see page 110)

Integrate with CA Service Desk

If one of the CA Database Management Solutions for DB2 for z/OS encounters a problem, it can automatically raise a CA Service Desk ticket for the error. To integrate with CA Service Desk, set the Service Desk Interface option to YES in the SETUP global parmlib member.

Business Value:

This practice speeds time to resolution by automating the notification that the product encountered an error and needed to terminate.

More Information:

For more information about integrating the CA Database Management Solutions for DB2 for z/OS with CA Service Desk, see the *CA Database Management Solutions for DB2 for z/OS Implementation Guide*.

Integration of CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer

If CA Detector identifies a query that is causing performance problems, you can immediately access CA Plan Analyzer for detailed SQL performance information and recommendations for improving the SQL. CA Subsystem Analyzer and CA Detector share information on data collection intervals. Users can move directly between the products to analyze buffer pool hit ratios, and getpage requests and physical object I/O. This enables problem tracking to be extended to database objects as well as programs and SQL statements.

Business Value:

You move directly into other products, when seeking additional information, to more quickly determine the root cause of a performance issue.

More Information:

For more information about how these products are integrated, see the product documentation.

Integration of CA Quick Copy and CA Rapid Reorg

CA Rapid Reorg can invoke CA Quick Copy to produce required image copies as part of the reorganization process. To integrate CA Quick Copy and CA Rapid Reorg, use the QUICKCOPY keyword in your CA Rapid Reorg jobs.

Business Value:

Eliminating additional steps in the reorganization process shortens the outage required for reorganization operations, which helps increase data availability and improve service levels.

More Information:

For more information about integrating CA Rapid Reorg and CA Quick Copy, see the *CA Rapid Reorg User Guide* and *CA Quick Copy User Guide*.

Integration of CA Quick Copy and CA Merge/Modify

CA Quick Copy is designed to work with various tools and utilities to offer the best possible solutions to meet your needs. Use the COPY-IMAGE-COPY function of the MERGECOPY option of CA Merge/Modify to make additional copies of an image copy.

Business Value:

Using CA Merge/Modify to produce additional image copies offline speeds up the initial image copy (backup) process and restores access to the database more quickly, for better service availability.

More Information:

For more information about the COPY-IMAGE-COPY function, see the *CA Merge/Modify User Guide*.

Integration of CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload

When analyzing a recovery strategy, CA Recovery Analyzer can generate the JCL and control statements for CA Fast Recover, CA Quick Copy, and CA Fast Unload to speed up the recovery process.

Business Value:

Using CA Recovery Analyzer to generate JCL for the CA utilities helps reduce the time needed to perform the recovery, which restores access to the databases more quickly and increases service availability.

More Information:

For more information about setting options for recovery strategies and using CA Recovery Analyzer to generate CA utility JCL, see the *CA Recovery Analyzer User Guide*.

Integration of CA Fast Load, CA Quick Copy, and CA Rapid Reorg

CA Fast Load can invoke CA Quick Copy to take an inline copy using the QUICKCOPY keyword and other related copy keywords in the LOAD syntax. CA Fast Load can also invoke CA Rapid Reorg to do a load and reorganization as a single step.

Business Value:

Eliminating additional steps in the load process shortens the outage required for load operations, thereby increasing data availability and improving service levels.

The speedy and effective reorganization of DB2 tablespaces and indexspaces reduces the window of data inaccessibility. In one execution, CA Rapid Reorg performs reorganizations, collects CA Database Analyzer statistics, produces up to eight image copies, and updates DB2 catalog statistics. Using online mode, you can even perform reorganizations while data is in read/write mode. CA Rapid Reorg offers speed and effective control and management of the reorganization requirements in a DB2 data processing center.

Additional Considerations:

CA Fast Load provides CA Rapid Reorg with a SYSREC after it finishes the load job. The REORG type load and a description of the syntax necessary to include CA Fast Load in the reorganization job stream are described in the *CA Fast Load User Guide*.

CA Fast Load can call CA Quick Copy to create up to eight image copies during execution, rather than as a separate phase. For more information about the QUICKCOPY keyword, see the *CA Fast Load User Guide*.

More Information:

For more information about invoking CA Quick Copy and CA Rapid Reorg from CA Fast Load, see the *CA Fast Load User Guide*.

Integration of CA Bind Analyzer, CA Detector, and CA Plan Analyzer

CA Bind Analyzer and CA Plan Analyzer interface to provide reports for DBRM explain access path and DBRM access path comparison. If CA Detector identifies a query that is causing performance problems, you can immediately access CA Plan Analyzer for detailed SQL performance information and recommendations for improving the SQL.

Business Value:

This integration helps improve application performance by proactively preventing deprecated access paths. It enables you to move, in context, directly into other products when seeking additional information to more quickly determine the root cause of a performance issue.

More Information:

For more information about using CA Bind Analyzer, CA Detector, and CA Plan Analyzer to improve application performance, see the product-specific documentation.

Integration of CA Fast Recover, CA Merge/Modify, and CA LogCompress

Integration can provide the following product benefits:

- CA LogCompress can make your archive logs smaller (by removing unnecessary records) so that you can store more logs on DASD.
- CA Merge/Modify can create a change accumulation log that contains only the log records required for recovery for certain objects.
- CA Fast Recover can read the change accumulation logs and compressed archive logs from CA LogCompress during recovery.

Business Value:

If you store more archive logs on DASD and create change accumulation logs for critical objects, your recovery process will take less time, and your objects will be available more quickly.

More Information:

For more information about CA Fast Recover, CA Merge/Modify, and CA LogCompress, see the product-specific documentation.

Integration of CA Insight DPM and CA SYSVIEW PM

CA Insight DPM can provide the CA SYSVIEW PM System Condition Monitor with DB2 status information, which allows CA SYSVIEW PM to monitor the status of the DB2 subsystems defined.

Business Value:

This integration enables rapid investigation of a system performance problem that may be caused by DB2.

More Information:

CA SYSVIEW PM includes a System Condition Monitor (SCM), which is a color-coded, high-level summary of resources that are currently being monitored. It uses intelligent modules (IMODs) written in compiled REXX with additional CA-supplied functions, or IMODs, to communicate with monitored subsystems.

For more information about IMODs, see the *CA SYSVIEW PM Administration Guide*.

CA RC/Query Integration

CA RC/Query works with CA Fast Check, CA Fast Load, CA Fast Unload, CA Fast Recover, CA Merge/Modify, CA Quick Copy, CA Rapid Reorg, CA RC/Compare, CA RC/Extract, CA RC/Migrator, CA RC/Secure, and CA RC/Update. From any CA RC/Query report panel, you can perform the following tasks:

- Edit, browse, or copy table data.
- Alter, create, or drop objects.
- Check tablespace DASD statistics.
- Move whole environments between subsystems.
- Invoke CA fast utilities.
- Perform many other tasks.

While viewing a list of selected objects within various areas of CA RC/Update, you can enter CA RC/Query commands to view reports and obtain more information about the particular object before making a request. CA RC/Query provides more than 140 different types of reports that let you view and analyze virtually any element that is present in the DB2 system catalog.

Business Value:

The tight integration among products simplifies database administration, performance management, and backup and recovery efforts, which improves DBA productivity and database availability.

More Information:

For more information about the reports that can be generated from CA RC/Query, see the *CA RC/Query User Guide*.

Index

B

backup and recovery • 93

C

CA Bind Analyzer best practices

integrating CA Bind Analyzer, CA Detector, and CA Plan Analyzer • 109

CA Database Analyzer best practices

generating a consistent number of maintenance jobs • 40

implementing a naming convention for extract and action procedures • 41

using IBM RUNSTATS and extract procedures • 66

using RTOS to dynamically select objects for maintenance • 68

CA Detector best practices

integrating CA Bind Analyzer, CA Detector, and CA Plan Analyzer • 109

integrating CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer • 106

CA Fast Check best practices

maximizing performance • 94

using DELETE statements • 93

CA Fast Index best practices

maximizing performance • 85

CA Fast Load best practices

integrating CA Fast Load, CA Quick Copy, and CA Rapid Reorg • 108

maximizing performance • 86

CA Fast Recover best practices

improving performance • 95

integrating CA Fast Recover, CA Merge/Modify, and CA LogCompress • 109

integrating CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload • 107

CA Fast Unload best practices

automating the creation of test tables • 38

integrating CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload • 107

maintaining CA Fast Unload data integrity • 38

CA Insight DPM best practices

adjusting default REGION parameter value • 23

adjusting dispatching priority to avoid lost trace data • 20

adjusting IQL requests if overhead is excessive • 26

allocating adequate trace collection buffer space • 20

avoiding CA Insight DPM data collector overhead • 23

creating an OBID table translation file • 20

customizing exceptions • 31

enabling appropriate security authorizations • 19

enabling data sharing if monitoring a data-sharing subsystem • 22

enabling the exception component and optimizing settings • 25

evaluating archive DB2 tables • 30

evaluating product security and limiting user access • 28

installing RAF interface for cross-system data access • 30

installing the CA SYSVIEW interface • 29

installing the GSS interface • 28

installing the history component • 23

installing the SCM component for integration to CA SYSVIEW PM • 29

installing VTAM interface for access outside of TSO/ISPF • 31

integrating CA Insight DPM and CA SYSVIEW PM • 110

performing EXPLAIN analysis • 31

refreshing the OBID table translation file periodically • 30

specifying IFI=YES and IFCID2-NOTIFI=NO for tracing parameters • 27

starting data collector and PC tasks automatically at IPL • 19

using one exception data set for all data collector tasks • 22

using reports for metrics and performance monitoring • 77

CA Log Analyzer best practices

improving performance • 96

using DB2 logs for compliance reporting • 97

CA LogCompress best practices

- integrating CA Fast Recover, CA Merge/Modify, and CA LogCompress • 109
- CA Merge/Modify best practices
 - improving performance • 95
 - integrating CA Fast Recover, CA Merge/Modify, and CA LogCompress • 109
 - integrating CA Quick Copy and CA Merge/Modify • 107
- CA Plan Analyzer best practices
 - integrating CA Bind Analyzer, CA Detector, and CA Plan Analyzer • 109
- CA Quick Copy best practices
 - integrating CA Fast Load, CA Quick Copy, and CA Rapid Reorg • 108
 - integrating CA Quick Copy and CA Merge/Modify • 107
 - integrating CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload • 107
 - maximizing performance • 98
 - reviewing use of Snapshot General Services • 100
- CA Rapid Reorg best practices
 - integrating CA Fast Load, CA Quick Copy, and CA Rapid Reorg • 108
 - maximizing DATA-AVAILABLE CRITICAL reorgs • 82
 - maximizing DATA-AVAILABLE HIGH reorgs • 83
 - maximizing DATA-AVAILABLE NORMAL reorgs • 83
- CA RC/Migrator best practices
 - reducing DB2 catalog contention and overhead • 88
 - retaining current DDL • 90
 - reviewing analysis options • 90
 - reviewing Model Services changes • 35
- CA Recovery Analyzer best practices
 - improving performance • 102
 - integrating CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload • 107
 - scheduling PRA#LOAD jobs to execute frequently • 102
- CA Service Desk, integrating with • 105
- CA Subsystem Analyzer best practices
 - integrating CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer • 106
- catalog contention reduction • 88
- configuration • 33
- contention reduction • 88

- CPU savings
 - in CA Database Analyzer • 66, 68
 - in CA Fast Check • 94
 - in CA Insight DPM • 23, 25, 26, 27

D

- database administration • 85

H

- health checks • 18
- hyper notifications • 16

I

- I/O reduction
 - in CA Fast Index • 85
 - in CA Fast Load • 86
 - in CA Rapid Reorg • 82, 83
- IBM maintenance • 18
- installation • 13
- integration points
 - integrating CA Bind Analyzer, CA Detector, CA Index Expert, and CA Plan Analyzer • 109
 - integrating CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer • 106
 - integrating CA Fast Load, CA Quick Copy, and CA Rapid Reorg • 108
 - integrating CA Fast Recover, CA Merge/Modify, and CA LogCompress • 109
 - integrating CA Insight DPM and CA SYSVIEW PM • 110
 - integrating CA Quick Copy and CA Merge/Modify • 107
 - integrating CA Quick Copy and CA Rapid Reorg • 106
 - integrating CA Recovery Analyzer with CA Fast Recover, CA Quick Copy, and CA Fast Unload • 107
 - integrating with CA Service Desk • 105

O

- outages, shortening • 106, 108
- overhead reduction
 - in CA Insight DPM • 23, 31
 - in CA Quick Copy • 98
 - in CA RC/Migrator • 88

P

performance improvements

- applying IBM maintenance • 18
- improving CA Fast Recover performance • 95
- improving CA Log Analyzer performance • 96
- improving CA Merge/Modify performance • 95
- improving CA Recovery Analyzer performance • 102
- integrating CA Bind Analyzer, CA Detector, CA Index Expert, and CA Plan Analyzer • 109
- integrating CA Detector, CA Plan Analyzer, and CA Subsystem Analyzer • 106
- integrating CA Fast Recover, CA Merge/Modify, and CA LogCompress • 109
- integrating CA Insight DPM and CA SYSVIEW PM • 110
- maximizing CA Fast Check performance • 94
- maximizing CA Fast Index performance • 85
- maximizing CA Fast Load performance • 86
- maximizing CA Quick Copy performance • 98
- maximizing DATA-AVAILABLE CRITICAL reorgs in CA Rapid Reorg • 82
- maximizing DATA-AVAILABLE HIGH reorgs in CA Rapid Reorg • 83
- maximizing DATA-AVAILABLE NORMAL reorgs in CA Rapid Reorg • 83
- reducing DB2 catalog contention and overhead • 88

performance management • 51

S

security

- compliance reporting • 97
 - enabling authorizations • 19
 - evaluation • 28
 - limiting user access • 28
- service desk, integrating with • 105
- support website
- applying IBM maintenance • 18
 - registering • 16
 - signing up for hyper notifications • 16

T

throughput, increasing • 98