CA IDMS™

Best Practices Guide

Release 18.5.00

CA Technologies Product References

This document references the following CA products:

- CA IDMS™/DB and the CA IDMS family of products
- CA Chorus™ Software Manager (CA CSM), formerly CA MSM

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 are based on customer experience reported through interviews with development, technical support, and technical services. Therefore, many of these best practices are a collaborative effort stemming from customer feedback.

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

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

Contents

Chapter 1: Introduction

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.

This section contains the following topics:

Active and Heartbeat Event Management through CA OPS/MVS EMA

CA Technologies mainframe products can automatically communicate both active status events and heartbeat events to CA OPS/MVS in a consistent manner. The enabling technology for this feature is through a generic event API call that CA OPS/MVS provides to the other products so that they can communicate events to CA OPS/MVS.

Two versions of this API call are provided to support this initiative:

- An active status event API call that allows other products to generate events for the CA OPS/MVS EMA System State Manager (SSM) component when they are starting, up, stopping, or down.
- A heartbeat API call that allows other CA Technologies products to communicate a normal, warning, or problem overall health status and reasoning to CA OPS/MVS EMA on a regular interval.

After a CA Technologies product begins generating heart beat events for CA OPS/MVS, CA OPS/MVS can also react to the lack of a heart beat event from another CA Technologies product address space, treating this as an indication that there is either a potential problem with the CA Technologies product address space, or there is a larger system-level problem.

SSM is a built-in feature of CA OPS/MVS that uses an internal relational data framework to proactively monitor and manage started tasks, online applications, subsystems, JES initiators, and other z/OS resources including your CA Technologies mainframe products. SSM compares the current state of online systems, hardware devices, and the other resources with their desired state, and then automatically makes the necessary corrections when a resource is not in its desired state. This provides proactive and reactive state management of critical resources. As previously noted, SSM is particularly interested in receiving active status events consistently from all CA Technologies products when they are starting, up, stopping, or down. Without this consistent type of events, SSM must maintain separate rules in CA OPS/MVS for each product unique messages that are associated with starting and stopping.

Chapter 2: Installation Best Practices

This chapter is not intended to describe all installation considerations for CA IDMS. Its purpose is to complement and not replace the CA IDMS product documentation set.

This section contains the following topics:

Use Electronic Software Delivery

Download the installation files from ca.com/support and install directly from your disk.

Business Value:

Using electronic software delivery (ESD) avoids ordering, shipping, and processing physical tape media to install CA IDMS. Using ESD is more timely, more cost-effective, and environmentally friendly. Because ESD uses standard z/OS utilities to prepare the product installation image on your system there is no need to learn new tools.

More Information:

For information about the steps to download your CA products from the CA Support Online web site for installation using the enhanced ESD pax process, see the *Mainframe Enhanced Electronic Software Delivery Guide* posted on the Download page of ca.com/support.

For additional information about the CA IDMS ESD process, see the following sections in the *CA IDMS Installation and Maintenance Guide - z/OS:* Installation from Disk, Delivery Media, and The Electronic Software Delivery Process.

Check for Platform Requirements

Check for any CA IDMS prerequisites for the operating system release or CICS Transaction Server (CICS TS) release that you are using or planning to upgrade to.

CA IDMS is certified for use with new releases of IBM operating systems and CICS TS when they become generally available. If there have been any changes in the operating system or in the CICS TS release that affect CA IDMS, the requirements for using CA IDMS on the platform are described in a Product Information Bulletin (PIB). The Product Information Bulletins are located at ca.com/support.

When upgrading to a new IBM processor such as the zNext or z196, check with IBM in advance for recommendations on how to preserve performance levels of existing software across the upgrade.

Business Value:

Ensuring that pre-requisites are addressed will provide for continued operation of CA IDMS at the current or optimal level of performance after upgrading to the new IBM software release or hardware.

Additional Considerations:

The CA Support Online website (ca.com/support) maintains a Compatibilities link that describes the product specific requirements for z/OS, z/VSE, z/VM, and CICS TS releases. After selecting the specific platform link, the CA IDMS requirements are located under the Databases link. For example, Product Support notice QI82743 provides information for configuring CA IDMS for use with z/OS 1.9 and above.

Keep Current on CA Common Services

Install the current release of CA Common Services.

Business Value:

The latest release of CA Common Services contains the most recent infrastructure updates, allowing you to use the latest software for SVC installation, license checking, service desk integration, and communication.

More Information:

For more information about CA Common Services, see CA Common Services for z/OS Requirements in the *CA IDMS Installation and Maintenance Guide - z/OS*.

Use the Appropriate Type of Configuration

Determine what type of configuration you will do and reference relevant information about that type of configuration.

The available types of configuration are:

- **Full base**–Allocates and formats new database files and new system definitions.
- **Upgrade**–Preserves previous database and system definitions.
- **Add-on**–Configures additional CA IDMS products into an existing CA IDMS environment.

Use the following table to help determine which type of configuration to perform.

	Full Base	Upgrade	Add-on
Allocates and formats database areas	Yes	No	Depends on product
Updates existing database areas	No	Yes	Depends on product

Business Value:

Each type of configuration has its own special considerations that may affect which type of configuration you decide to perform to achieve your objective. Choosing the correct type of configuration avoids execution of unnecessary jobs saving time and CPU costs.

More Information:

For more information about the types of configuration available for CA IDMS, see the following sections in the *CA IDMS Installation and Maintenance Guide - z/OS:* Configuration Types. For an upgrade installation, also see the *CA IDMS Release Notes Version 18.0.00*, Upgrading to Version 18.0.

Use Naming Conventions

Use of appropriate and meaningful naming conventions when defining your SMP/E and CA IDMS data sets will simplify identification and maintenance of your CA IDMS environments.

When performing a CA IDMS SMP/E installation we recommended you take the following step:

- Put a release indicator (like R180) in the high-level qualifiers when creating the SMP/E environment.

When performing a CA IDMS configuration we recommended you take the following steps:

- Put a release indicator (like R180) in the CFGPFX variable in VARBLIST.
- Do not include a release indicator in the DBPFX variable in VARBLIST.

Business Value:

This practice clearly segregates your SMP/E environments by release and makes it easy to identify the release for which an environment was created. Not including a release indicator in database file names eliminates the need for manually creating new files each time you upgrade to a new release. The same set of database files will be used for all releases of CA IDMS.

Additional Considerations:

Any type of installation creates new SMP/E data sets. An upgrade does not create new database files.

Upgrade All Dictionaries and Systems

On an upgrade, be sure to update all application dictionaries and DC/UCF system definitions. If you maintain multiple SYSDIRL dictionaries or multiple CA IDMS message areas, be sure to upgrade each of those as well. For more information about SYSDIRL dictionaries, see Use a Single SYSDIRL Dictionary later in this guide.

Business Value:

By ensuring that all application dictionaries, SYSDIRL dictionaries, message areas, and DC/UCF systems are updated with the changes for the release being installed, you avoid problems caused when the definition of objects does not match what the software expects.

Additional Considerations:

If you do a manual (not using CA CSM) upgrade, configuration Job 10 updates only the application dictionary identified in the VARBLIST with parameter APPLDICT. To update additional application dictionaries, rerun steps APPLDEFS, DLODAIDN, APPLPROT, and APPLARSQ in JOB10 for each additional dictionary. You must alter the job stream on each execution to name the DMCL and target dictionary to be updated.

Additionally, you must update the definition of every DC/UCF system in use with new and revised task and program definitions. This two-step process is described next.

1. Update SYSTEM 99 in each of your SYSTEM dictionaries. The JCL generated for Job09 contains a section that adds sysgen source to the system dictionary. This section uses IDMSDDDL to load the DLOD*xxxx* or DNOD*xxxx* members, depending on the option you chose for the STORPROT variable. Subsequent steps update SYSTEM 99 for various products. Extract the sysgen steps and run them against each of your system dictionaries, altering the job stream on each execution to identify the appropriate DMCL.

2. Update each of your system definitions.

   The simplest way to update your system definitions is to copy the task and program definitions from SYSTEM 99 to your DC/UCF system. Copying can be done by rerunning the SGN90GJU step from JOB16 and changing the number of the target system from 90 to that of your DC/UCF system. The input to JOB16 is shown next:

```
//SYSIPT DD *
  SIGNON DICTIONARY SYSTEM
  USA UPDATE FOR DDLDML
  USA UPDATE FOR DDLDCLOD
  USA RET FOR DDLDCMSG
  SET OPT FOR SESSION DEF ON NO LIST INP 1 THRU 72.
  MODIFY SYSTEM nnn.
  COPY TASK FROM SYSTEM 99.
  COPY PROGRAM FROM SYSTEM 99.
  GENERATE.
```

Reference the CA IDMS Product Page on Support Online

The CA IDMS product page on CA Support Online provides links to:

- Recorded webcasts on CA IDMS technical topics
- Videos on how to use selected features of CA IDMS
- Knowledge documents providing focused topics for improved use of CA IDMS
- CA IDMS product news
- CA IDMS product roadmap
- CA IDMS release and support lifecycle
- CA IDMS product documentation
- CA IDMS product download and order form
- CA IDMS published solutions

Business Value:

Using the CA IDMS product page on Support Online enables you to directly access news, documentation, release and support information, and published solutions about the CA IDMS products. You will be able to keep up to date with the current CA IDMS product information.

More Information:

You can access CA Support Online using the URL http://ca.com/support. Select CA IDMS from the drop-down list in the Support by Product section.

Note: Access to some of the CA IDMS product page links requires a user logon.

Chapter 3: Configuration Best Practices

This chapter is not intended to describe all configuration considerations for CA IDMS. Its purpose is to complement and not replace the CA IDMS product documentation set.

This section contains the following topics:

Dictionary Setup and Maintenance

The following sections contain best practices for dictionary setup and maintenance:

- Segregate system dictionaries
- Use a single SYSDIRL dictionary

Segregate System Dictionaries

Use a separate system dictionary for each CA IDMS central version and reserve it only for system information.

Business Value:

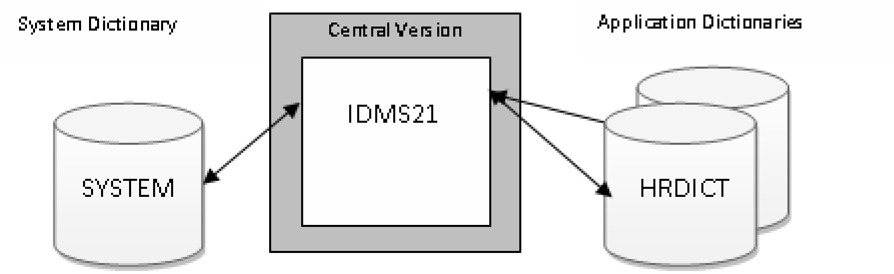
Using separate system dictionaries ensures that if recovery of a system dictionary is necessary only a single CA IDMS central version is impacted. This practice also ensures that changes made to such a dictionary are infrequent and controlled, minimizing the likelihood of damage.

Additional Considerations:

Each CA IDMS central version must have one, and only one, system dictionary. A dictionary is designated as a system dictionary by assigning it a name of SYSTEM in the database name table used by the central version. A central version accesses its system dictionary to retrieve configuration information, such as the DC/UCF system definitions, database definitions, and security information that control the execution of the central version.

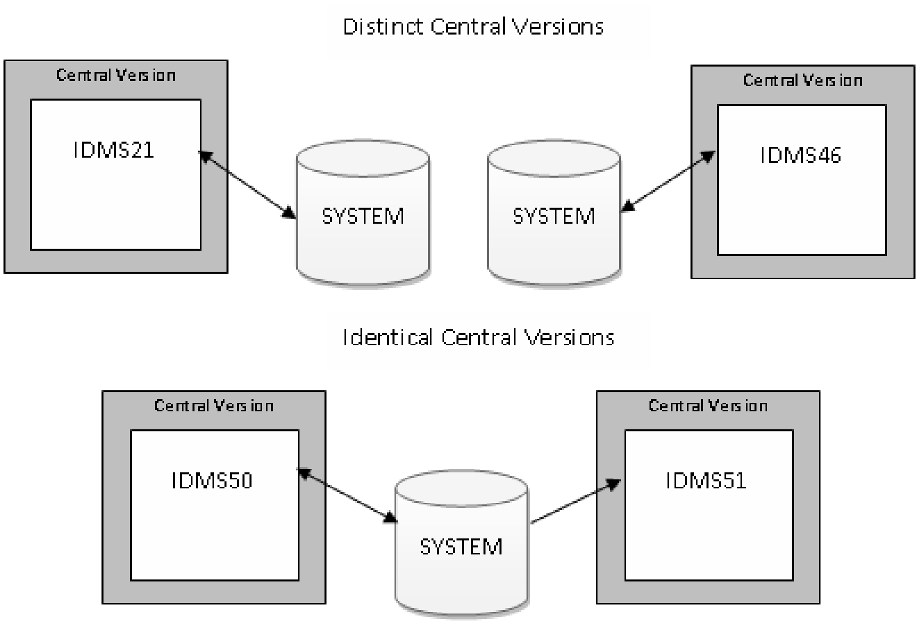
While it is possible to use a system dictionary as a repository for application definitions, we recommend that this not be done. Define one or more application dictionaries for this purpose instead. Clearly separating the use of the two ensures that the central version is not affected if application information must be recovered. It also makes it easier to control changes to configuration settings.

The following diagram illustrates the separation of system and application dictionaries on central version IDMS21.



We also recommend that each central version have its own system dictionary. Doing this eliminates the interdependence between central versions and makes it possible to restore one central version's configuration without affecting that of another. In most cases this is the best way to set up your environment; however, sometimes it is useful for multiple central versions to share a single system dictionary. This is the case when central versions are intended to always have the same runtime attributes. In this situation maintaining the configurations is easier if they all reside in one system dictionary.

The following diagram illustrates both segregating and sharing system dictionaries and configuration information. Central versions IDMS21 and IDMS46 each have their own system dictionary and configuration information because their runtime attributes are not similar. Central versions IDMS50 and IDMS51 share a system dictionary and configuration information because their runtime attributes are meant to be identical.



More Information:

For more information about system dictionaries, see the *CA IDMS Database Administration Guide – Volume 2*.

Use a Single SYSDIRL Dictionary

Define one, and only one, SYSDIRL dictionary for each release of CA IDMS.

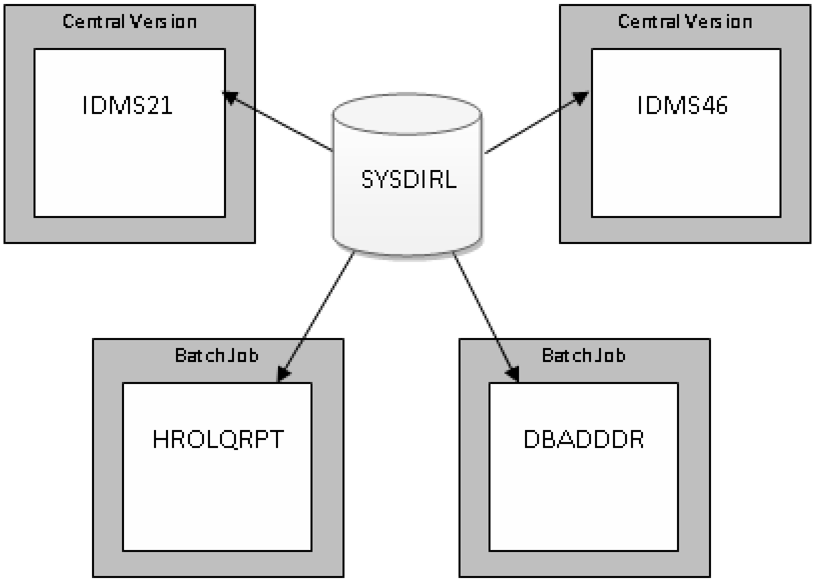
Business Value:

Defining only one SYSDIRL dictionary saves disk space because only one copy of the dictionary schema definition exists. A single copy also eliminates the need for updating multiple dictionaries each time the definition changes.

Additional Considerations:

A SYSDIRL dictionary contains the schema and subschema definitions that describe the dictionary itself. These definitions are used when running dictionary reports and using query tools to report on dictionary contents. The SYSDIRL dictionary also contains all of the CA-provided CA Culprit report source modules. Each copy of the SYSDIRL data is about 20 megabytes in size, so you can save considerable disk space by storing it only once. The dictionary in which you place it is referred to as a SYSDIRL dictionary and it can be shared by all central versions and local mode jobs.

The following diagram illustrates the use of a single SYSDIRL dictionary shared by central versions IDMS21 and IDMS46 and batch jobs HROLQRPT and DBADDDR.



If you decide to maintain multiple SYSDIRL dictionaries, you must update the definitions in each dictionary when upgrading to a new release of CA IDMS. If you are doing manual configuration, the steps to update the definitions in one dictionary instance are executed as part of Job 9.

To update additional dictionaries, you must rerun steps DIRLDEFS, DIRLPROT, DIRLARSQ, and DIRLDICT for each additional dictionary and change the job stream on each execution to name the target dictionary and the DMCL that describes it.

More Information:

For more information about the types of dictionaries and their uses, see the *CA IDMS Database Administration Guide – Volume 2*.

Practices for Improved Productivity and System Availability

Use the following practices to improve productivity and system availability:

- Use a single startup module
- Establish defaults using SYSIDMS
- Use change tracking
- Automate journal offload
- Automate log offload

Use a Single Startup Module

Use a single startup module for all CA IDMS central versions and specify the runtime parameters for an individual central version as keyword/value pairs in the PARM field on the EXEC statement in the startup JCL.

Business Value:

This practice simplifies maintenance and eliminates the effort involved in creating and applying maintenance to multiple startup modules. Use of keyword/value pairs makes it easy to specify options and see those in effect for a central version simply by looking at the JCL.

Additional Considerations:

You must use startup parameters to identify the CA IDMS central version to be started and the DMCL to be used. You can specify values for these parameters in one of two ways:

- Assembling a #DCPARM macro and linking the resulting module with the CA-supplied startup module (RHDCOMVS)
- Specifying the parameter values in the central version's startup JCL

We strongly recommend using the second approach because it eliminates the need for creating and maintaining a separate startup module for each central version.

To provide startup parameters through JCL, specify them in the PARM field of the central version's EXEC statement. You can specify parameter values either as keyword/value pairs or as positional parameters in specific locations within the PARM field. Not only are keyword/value pairs easier to specify and understand than positional parameters, but only keyword/value pairs support all startup parameters. As an example, the name of the DMCL to use can be specified only as a keyword/value pair and not as a positional parameter.

The following EXEC statement starts CA IDMS system 73 with zIIP enabled, a DMCL of CUSTCV, and a WTO exit of CUSTWTO.

```
//CVSTART   EXEC PGM=RHDCOMVS,
//          PARM='S=73,ZIIP=Y,WTO=DEVWTOX,DMCL=CVDMCL'
```

A further means for eliminating tailored startup modules is to avoid linking Write-to-Operator (WTO) and Write-to-Operator-Reply (WTOR) exit routines with the CA-supplied startup module. This can be done either by linking them as stand-alone modules under the names of WTOEXIT and WTOREXIT respectively, or by specifying their load module names as startup parameters as shown in the previous example.

More Information:

For more information about the alternatives for specifying runtime parameters at startup, see System Startup in the *CA IDMS System Operations Guide*.

Establish Defaults Using SYSIDMS

Create a customized SYSIDMS load module to establish default options for your environment.

Business Value:

Creating a customized SYSIDMS load module eliminates the repetitive specification of parameter values in each job stream. It also helps to establish standards for your site.

Additional Considerations:

A SYSIDMS load module can establish defaults for several options that affect CA IDMS execution such as scratch space attributes, the dictionary to be accessed, and whether uppercase and lowercase messages are generated.

To define a SYSIDMS load module, simply define an assembler program that consists of 80-byte character constants, each of which contains one or more SYSIDMS parameters. You code the parameters just as you would when specifying them at runtime through the card-image SYSIDMS parameter file. The resultant module must then be linked with a name of SYSIDMS.

The following is an example of a program for creating a SYSIDMS load module. It establishes defaults for the DMCL and the use of in-memory scratch. It also turns on echoing so that all specified SYSIDMS parameters are displayed on the job's log.

```
SYSIDMS  CSECT
         DC CL80'ECHO=ON DMCL=GLBLDMCL'
         DC CL80'SCRATCH_IN_MEMORY=ON'
         DC CL80'END SYSIDMS DEFAULTS'
         END
```

More Information:

To create a customized SYSIDMS load module, follow the steps described in the *CA IDMS Common Facilities Guide*.

Use Change Tracking

Use change tracking in each CA IDMS central version by referencing a SYSTRK file in the execution JCL of the central version.

A SYSTRK file contains a description of the database environment most recently in use by the central version. During startup, an image of the current DMCL is written to SYSTRK along with information about database and journal files defined in the JCL.

Business Value:

The use of change tracking improves DBA productivity and provides for enhanced system availability by eliminating the need for manual intervention if a failure occurs during certain maintenance activities.

Change tracking lets you change the database environment of a central version in a fault-tolerant manner. Specifically, it permits the DBA to perform the following dynamic actions:

- Vary the data set name of a journal or database file within a central version without introducing the need for manual intervention in case of a central version failure.
- Vary a new version of a DMCL without introducing the potential for a warmstart failure.
- Vary the status of an area or segment permanently on a central version regardless of subsequent page range changes.
- Change the journal files in use by a central version and coordinate those changes with the associated archive journal jobs.

If the central version fails, the runtime database definition is restored from SYSTRK files during the restart, ensuring that the files being updated at the time of failure are the ones recovered by the warmstart unless explicitly overridden by changes in the JCL used to restart the central version.

Additional Consideration:

Change tracking requires the use of SYSTRK files for recording the current state of a central version's database definition.

To implement change tracking for a central version, take the following steps:

1. Create a model SYSTRK file whose dataset name establishes the pattern for the individual SYSTRK files.
2. Create and format two to four SYSTRK files whose data set names are the same as that of the model SYSTRK file suffixed with a unique digit from 1 to 9. A minimum of two SYSTRK files are required due to internal mirroring, which is used to provide fault tolerance and recoverability in case of file damage.
3. Alter the central version execution JCL to reference the model SYSTRK file. Use a ddname established by the SYSIDMS parameter SYSTRK_DDNAME_PREFIX. The default value for this parameter is SYSTRK. SYSIDMS is a parameter file added to the JCL stream of batch jobs running in local mode or under the central version.
4. Change the JCL for the associated archive journal job to also reference the model SYSTRK file and to remove references to the disk journal files.
5. Optionally, change the JCL of other local mode jobs to reference the model SYSTRK file and remove explicit references to database files.

More Information:

For more information about the use of change tracking and how to allocate and format SYSTRK files, see the *CA IDMS System Operations Guide*.

Automate Journal Offload

Use the Write-to-Operator exit (WTOEXIT) to automate the offloading of a full disk journal file.

Business Value:

This practice avoids processing delays caused when a CA IDMS central version must halt database activity due to a lack of journal space.

Additional Considerations:

A CA IDMS central version uses its disk journal files to record update activity against its database files. Journal information is needed for recovery in case of failures such as system outages or damaged files.

A central version uses at least two and sometimes several disk journal files. When one journal file fills, CA IDMS automatically switches to another journal file. However if all journal files fill, CA IDMS must halt update activity until the ARCHIVE JOURNAL utility is executed to archive the journal data and reclaim space on the journal file. To avoid processing delays, it is critical to run this utility as soon as a central version switches to a new journal file.

CA IDMS switches to another disk journal file when:

- The active disk journal becomes full.
- A DCMT VARY JOURNAL command is issued under the central version.
- An I/O error is detected on the active disk journal file.

When CA IDMS switches to another disk journal file, it writes a DC205003 message to the operator. This message indicates that a swap has occurred and that the previously active journal file needs offloading. Usually an operator must respond to this message by submitting a job to archive the full file. You can eliminate the need for operator intervention and ensure immediate job submission by using a Write-to-Operator exit routine. The WTO exit intercepts and reviews the message to the operator and responds by automatically submitting a job through the internal reader.

During configuration, the source for a sample WTO exit module called WTOEXIT is assembled and linked. The exit looks for DC205003 messages. Each time one is encountered, the exit submits the contents of the file identified by the AJNLJOB DD statement that must be included in your startup JCL. The startup JCL created in Job 2 shows how to use the sample WTOEXIT module.

The module does not have to be named WTOEXIT. To use a different name, assemble and link the sample source from Job 2 using the name of your choice and specify the name as a startup parameter to your central version.

More Information:

For more information about journals, see the *CA IDMS Database Administration Guide*. For information about the WTOEXIT user exit and sample routines, see the *CA IDMS System Operations Guide*.

Automate Log Offload

Use the Write-to-Operator exit (WTOEXIT) to automate the offloading of log information.

Business Value:

This practice avoids processing delays caused when a CA IDMS system must wait for space in the log area.

Additional Considerations:

A CA IDMS central version uses its log area to record runtime events such as task failures, configuration changes and resource shortages. The log may also be used to record statistics for performance monitoring and chargeback purposes. If its log area becomes full, a central version must wait until the ARCHIVE LOG utility is executed to archive the information and reclaim space in the area. To avoid processing delays, it is critical to run this utility in a timely fashion.

A central version has a single log area. As the log fills CA IDMS issues DC050001 messages indicating the percentage of the area that has been used. Usually an operator must respond to this message by submitting a job to offload the log information. You can eliminate the need for operator intervention and ensure immediate archive job submission by using a Write-to-Operator exit routine. The WTO exit intercepts and reviews the message to the operator and responds by automatically submitting a job to offload the log.

A sample WTO exit module called WTOEXIT is assembled and linked during CA IDMS configuration. It looks for DC050001 messages. Each time one is encountered the exit submits the contents of the file identified by a PLOGJOB DD statement that must be included in your startup JCL. To use the sample WTOEXIT, specify the load module name as a startup parameter to your central version.

More Information:

For information about the WTOEXIT user exit and sample routines, see the *CA IDMS System Operations Guide*.

Practices for Monitoring and Managing Your Operation

The following practices help to ensure a smoothly running operation:

- Plan backup and recovery
- Monitor key system resources
- Use CA IDMS Visual DBA

Plan Backup and Recovery

Be sure to establish a backup and recovery plan. The procedures that you establish for backing up your data should enable recovery within a time period that meets the needs of your organization. This will vary depending on the volume and frequency of updates and how critical the data is to your organization. You should establish a plan that meets your business needs, and you should regularly test the plan to ensure that it continues to meet those needs.

Business Value:

Having a backup and recovery strategy helps to ensure that you can successfully recover data in the event of a failure such as a hardware or software malfunction.

More Information:

For more information about backup and recovery, see the *CA IDMS Database Administration - Volume 2*.

Monitor Key System Resources

Establish regular monitoring of key system resources, such as space, CPU, I/O, response time, and so on.

Business Value:

By monitoring critical resources, you can anticipate increased demands and avoid unanticipated interruptions in service by preemptively taking corrective action.

Additional Considerations:

You should monitor resources such as database space, index efficiency, response times, CPU, I/O, and buffer utilization. To monitor these resources, periodically capture relevant statistics and review them looking for trends and sudden changes.

- Detecting trends enables you to plan for upgrades or schedule maintenance at a time when it is least disruptive to your operations. For example, a database area that is approaching 70% full should be expanded before it causes degraded performance or becomes completely full and results in an interruption in service.

- Detecting sudden changes in resource utilization assists in problem diagnoses so that corrective action can be taken in a timely manner. For example, a sudden increase in response time might be due to hardware or software changes, the relocation of files or the implementation of a new application.

CA IDMS provides several monitoring facilities. The following table identifies facilities that can be used to monitor critical resources.

Resource	Monitoring Facilities
Space in a database area	PRINT SPACE utility
Index efficiency	PRINT INDEX utility
CPU Time	DCMT DISPLAY STATISTICS SYSTEM command System Statistics report (Statistics report 3) Summary Recap report (CA IDMS Performance Monitor PMARPT03)
Response Time	Summary Recap report (CA IDMS Performance Monitor PMARPT03)
I/O	DCMT DISPLAY STATISTICS SYSTEM command DCMT DISPLAY STATISTICS AREA|BUFFER|FILE commands Program Summary (Journal report 4) Database Summary report (CA IDMS Performance Monitor PMARPT18)
Buffer Utilization	DCMT DISPLAY STATISTICS AREA|BUFFER|FILE commands Program Summary (Journal report 4)

More Information:

For more information about monitoring your CA IDMS environment, see the *CA IDMS Database Administration Guide – Volume 2* and the *CA IDMS System Operations Guide.* For more information about how to use the monitoring facilities, see the *CA IDMS Utilities Guide, CA IDMS Reports, CA IDMS Performance Monitor Guide*, and *CA IDMS System Tasks and Operator Commands Guide*.

Use CA IDMS Visual DBA

Use CA IDMS Visual DBA for defining, maintaining, and monitoring CA IDMS systems.

Business Value:

CA IDMS Visual DBA is an easy-to-use graphical user interface (GUI) tool for performing database administration tasks. It does not require use of JCL or knowledge of syntax to perform tasks. It has an extensive online help system with detailed context-sensitive information for all of its capabilities.

Use of this tool reduces the learning curve for a new DBA and increases the productivity of an experienced DBA. CA IDMS Visual DBA provides a single point-and-click interface for simultaneously managing and monitoring multiple CA IDMS systems and databases.

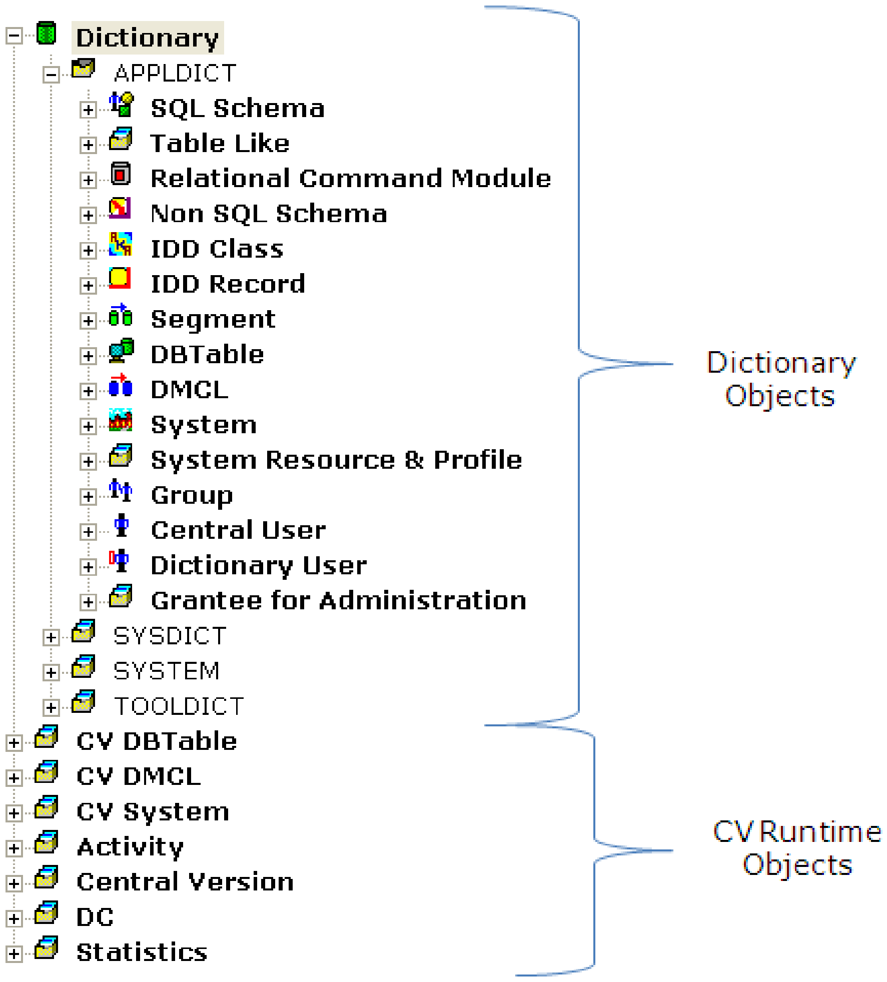
CA IDMS Visual DBA is available to all CA IDMS/DB clients at no charge.

Additional Considerations:

CA IDMS Visual DBA runs on the Windows operating system and can access multiple CA IDMS central versions from a single PC.

The CA IDMS Visual DBA user interface includes an object tree and detailed information pane components. Objects in the tree can be expanded into lower level objects. By right-clicking an object, you can create, change, or drop instances of that object type. Standard CA IDMS security is used to control capabilities available to the user.

At the root level of the object tree, there are dictionary objects and central version (CV) runtime objects.



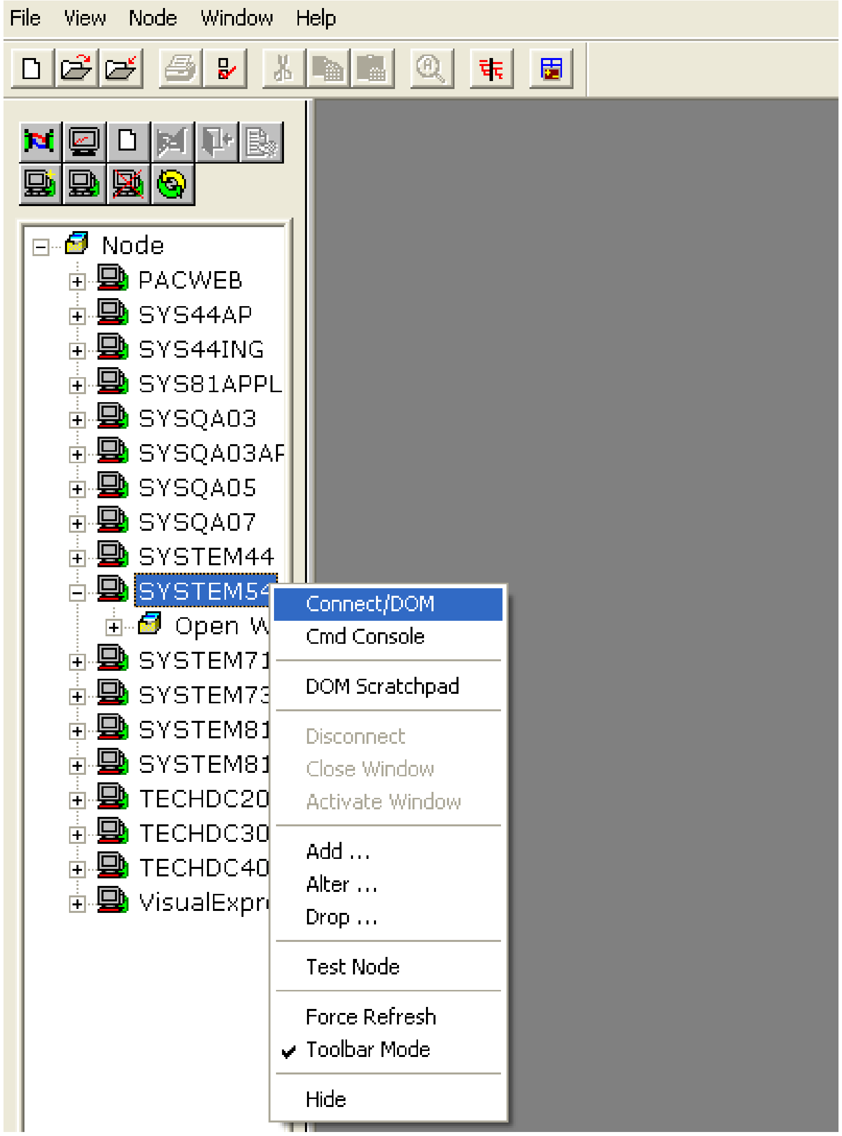
Dictionary objects include databases, systems, and users. The database objects expand into objects for logical and physical database definitions for both non-SQL and SQL-defined databases. You can define new database, system, and user object instances as well as modify and delete existing definitions. You can also use point and click to display rows in the database.

Central version runtime objects enable active monitoring and dynamic tuning of central versions. By connecting to multiple CA IDMS nodes and using tear-out windows, you can actively monitor multiple CA IDMS central versions simultaneously.

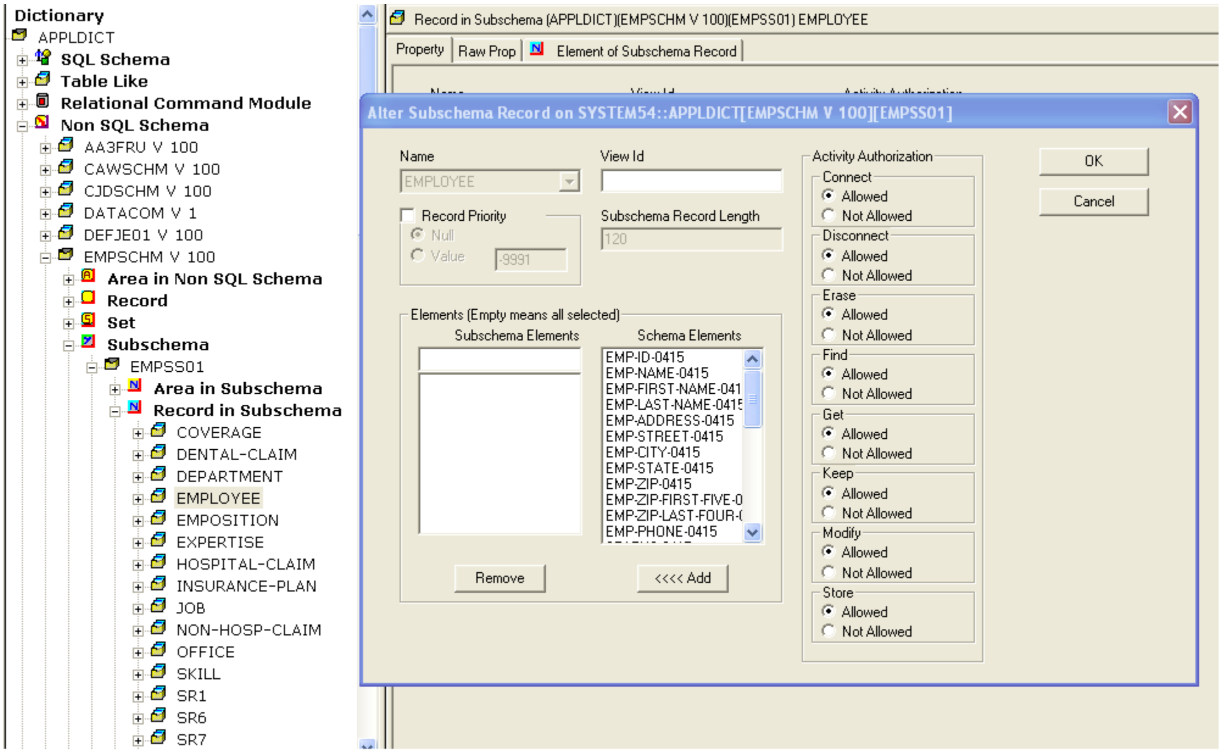
CA IDMS Visual DBA also includes a Command Console that enables you to edit, execute, and view the results of commands and scripts for the DCMT, DCUF, IDD, OCF, Schema, SQL SSC, and Sysgen command processors.

Here are some examples of the capabilities available in CA IDMS Visual DBA.

- Connect to a CA IDMS system using the Node window. You can connect to multiple CA IDMS systems simultaneously. After you are connected, you can hide the display of the Node window.



- Modify a subschema record. Using the drill-down capabilities of the object tree, you can navigate to a specific subschema record. Right-clicking on the subschema record name opens a dialog with the current definitions. You can use the controls in the dialog to change the current definition and save the modified record definition.



Dictionary
- APPLDICT
  - SQL Schema
  - Table Like
  - Relational Command Module
  - Non SQL Schema
    - AA3FRU V 100
    - CAWSCHM V 100
    - CJDSCHM V 100
    - DATACOM V 1
    - DEFJE01 V 100
    - EMPSCHM V 100
      - Area in Non SQL Schema
      - Record
      - Set
      - Subschema
        - EMPSS01
          - Area in Subschema
          - Record in Subschema
            - COVERAGE
            - DENTAL-CLAIM
            - DEPARTMENT
            - EMPLOYEE
            - EMPOSITION
            - EXPERTISE
            - HOSPITAL-CLAIM
            - INSURANCE-PLAN
            - JOB
            - NON-HOSP-CLAIM
            - OFFICE
            - SKILL
            - SR1
            - SR6
            - SR7

Record in Subschema (APPLDICT)(EMPSCHM V 100)(EMPSS01) EMPLOYEE

Alter Subschema Record on SYSTEM54::APPLDICT[EMPSCHM V 100][EMPSS01]

Name: EMPLOYEE

View Id:

☐ Record Priority
● Null
○ Value: -9991

Subschema Record Length: 120

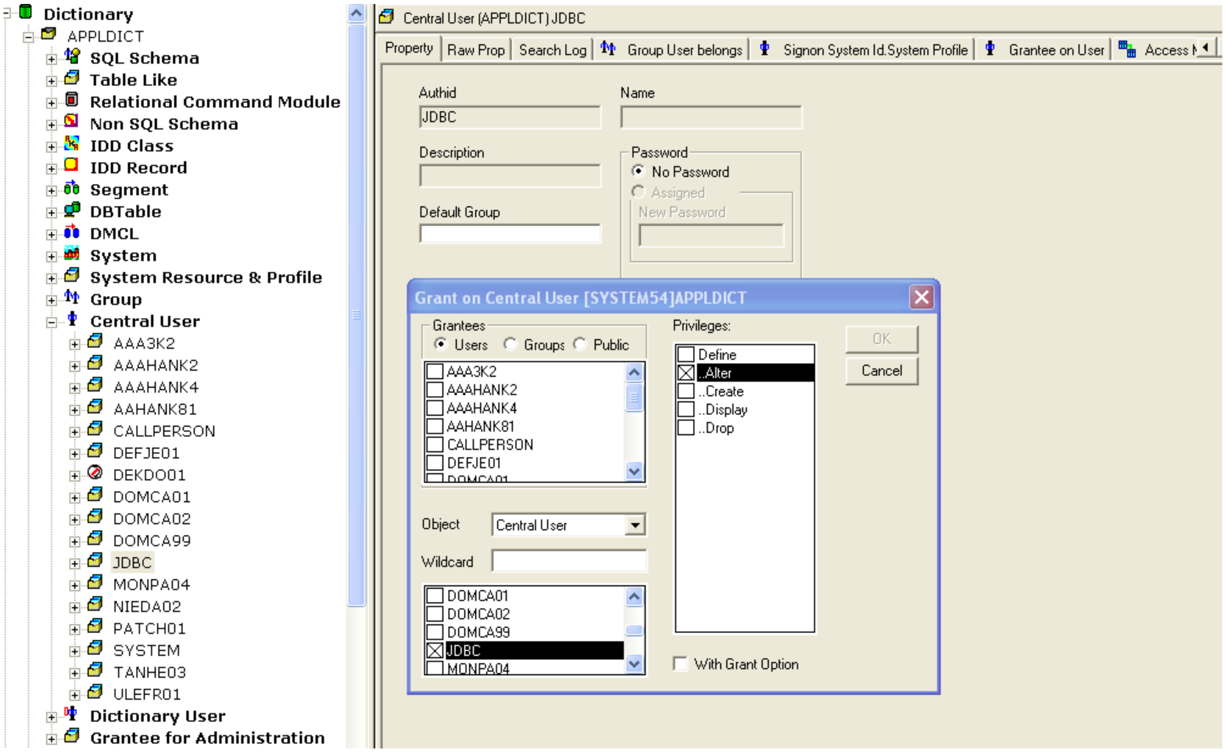
Elements (Empty means all selected)

Subschema Elements	Schema Elements
	EMP-ID-0415
	EMP-NAME-0415
	EMP-FIRST-NAME-041
	EMP-LAST-NAME-0415
	EMP-ADDRESS-0415
	EMP-STREET-0415
	EMP-CITY-0415
	EMP-STATE-0415
	EMP-ZIP-0415
	EMP-ZIP-FIRST-FIVE-0
	EMP-ZIP-LAST-FOUR-(
	EMP-PHONE-0415

Activity Authorization

Activity	Allowed	Not Allowed
Connect	●	○
Disconnect	●	○
Erase	●	○
Find	●	○
Get	●	○
Keep	●	○
Modify	●	○
Store	●	○

- Grant and revoke privileges to users. In addition to maintaining individual users, the drag-and-drop capabilities of dictionary objects can be used to simplify maintenance of users across multiple dictionaries.



Authid: JDBC

Password: ● No Password ○ Assigned

Grantees: ● Users ○ Groups ○ Public

□ AAA3K2
□ AAAHANK2
□ AAAHANK4
□ AAHANK81
□ CALLPERSON
□ DEFJE01

Privileges:
□ Define
☒ ..Alter
□ ..Create
□ ..Display
□ ..Drop

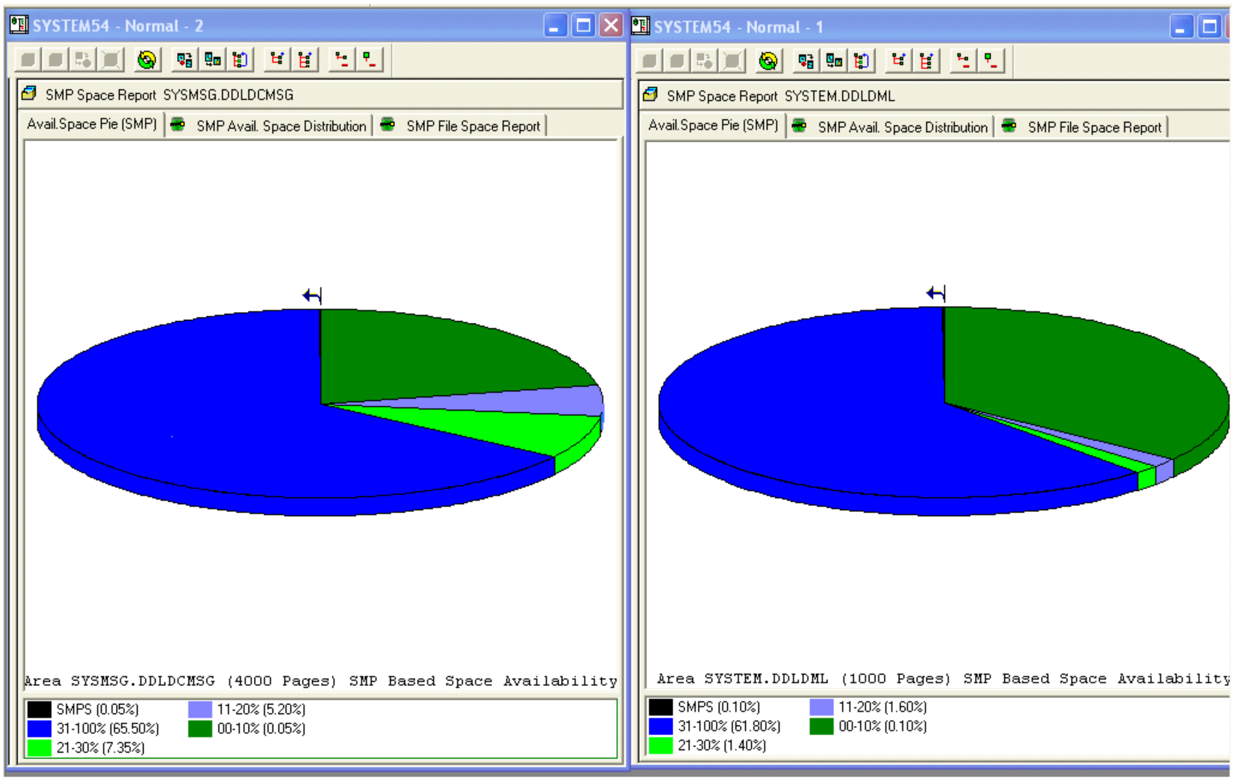
Object: Central User

Wildcard:

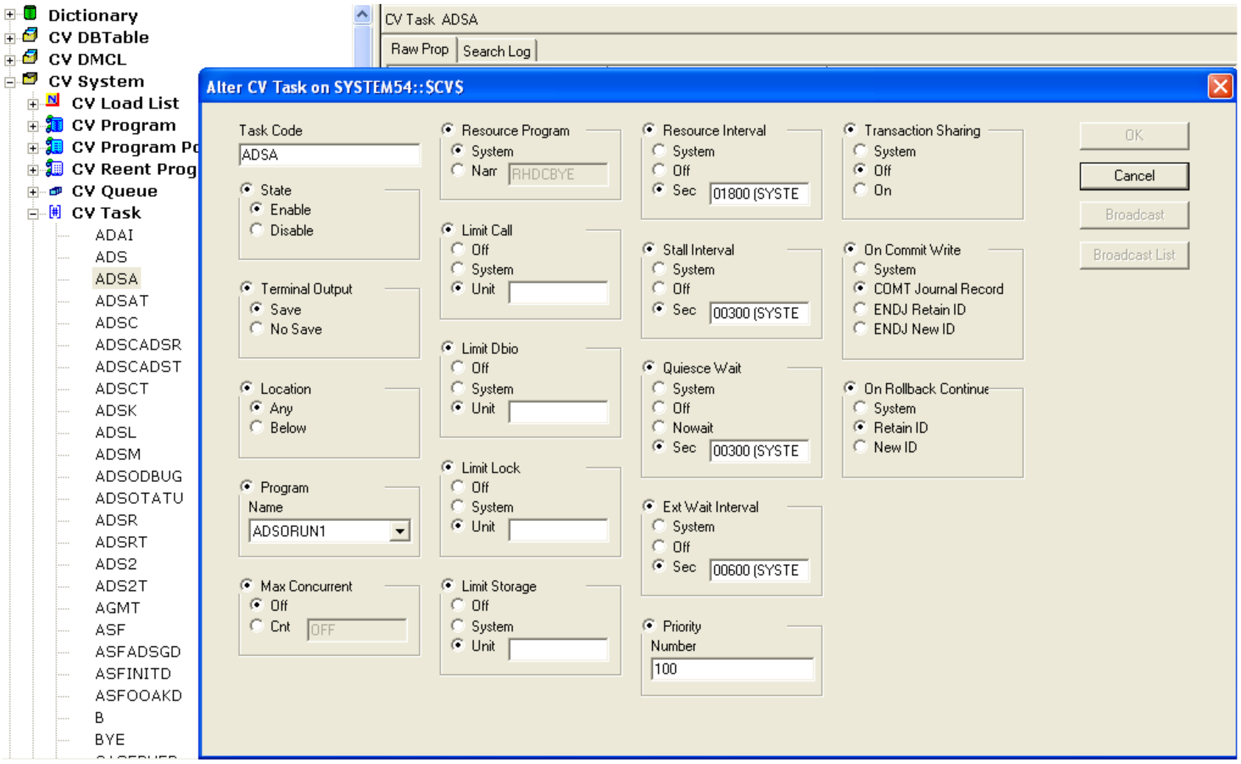
□ DOMCA01
□ DOMCA02
□ DOMCA99
☒ JDBC
□ MONPA04

□ With Grant Option

- Dynamically monitor space utilization of database areas using color-coded pie charts. To display this pie chart for an area, expand the CV DMCL object, and then navigate to the CV area that you want. This example also illustrates the tear-out capabilities of CA IDMS Visual DBA that allows multiple panes to be displayed simultaneously.



- Dynamically change a task definition on the system to which you are connected. You can use the point-and-click functionality of CA IDMS Visual DBA to perform the equivalent of a DCMT VARY TASK command.



Alter CV Task on SYSTEM54::SCVS

Task Code: ADSA

● State: ● Enable ○ Disable

● Terminal Output: ● Save ○ No Save

● Location: ● Any ○ Below

● Program Name: ADSORUN1

● Max Concurrent: ● Off ○ Cnt OFF

● Resource Program: ● System ○ Narr RHDCBYE

● Limit Call: ○ Off ○ System ● Unit

● Limit Dbio: ○ Off ○ System ● Unit

● Limit Lock: ○ Off ○ System ● Unit

● Limit Storage: ○ Off ○ System ● Unit

● Resource Interval: ○ System ○ Off ● Sec 01800 (SYSTE

● Stall Interval: ○ System ○ Off ● Sec 00300 (SYSTE

● Quiesce Wait: ○ System ○ Off ○ Nowait ● Sec 00300 (SYSTE

● Ext Wait Interval: ○ System ○ Off ● Sec 00600 (SYSTE

● Priority Number: 100

● Transaction Sharing: ○ System ● Off ○ On

● On Commit Write: ○ System ● COMT Journal Record ○ ENDJ Retain ID ○ ENDJ New ID

● On Rollback Continue: ○ System ● Retain ID ○ New ID

More Information:

For more information about CA IDMS Visual DBA, see the *CA IDMS Visual DBA User Guide*.

CA IDMS Visual DBA is downloadable from ca.com/support.

Tuning for Improved Performance and Lower TCO

The best practices for making optimal use of your computer resources while maximizing CA IDMS performance include:

- Use of the IBM z Integrated Information Processor (zIIP)
- Use of the High Performance Storage Protection Option
- Use of the scratch in memory and expandable scratch features
- Use of recommended settings for system generation parameters
- Use of CICS threadsafe applications
- Use of the type 4 JDBC driver for Java applications
- Use of recommended CICS TS settings

Utilize the IBM z Integrated Information Processor

The zIIP (IBM System z Integrated Information Processor) is a specialty mainframe processor designed to help free general computing capacity from the Central Processor (CP). CA IDMS exploits the zIIP to provide improved Total Cost of Ownership (TCO) for your mainframe environment.

You can easily evaluate the benefit of the CA IDMS zIIP exploitation feature for your environment.

- If you have one or more zIIPs available to the LPAR on which your CA IDMS central version is executing, specify ZIIP=Y as a CA IDMS startup parameter.
- If no physical zIIP is present, you can also estimate potential zIIP redirection by specifying ZIIP=Y at startup.

  Note: Running with ZIIP=Y on a system without a ZIIP engine can result in performance overhead and inaccurate estimates.

In both cases, you can then use CA IDMS system statistics to calculate the actual or potential CP utilization reduction for your environment.

Note: The zIIP exploitation feature was introduced with CA IDMS r17 and enhanced for Version 18.

Business Value:

Use of the zIIP feature can help you lower your TCO (total cost of ownership) on the mainframe. By exploiting zIIP, CA IDMS can help you deliver more computing capacity and throughput without additional system hardware resources.

CA IDMS makes wide use of the zIIP engine. Portions of all typical CA IDMS production workloads—including those generated by online transaction systems, batch processing jobs, and distributed platform requests—can exploit zIIP capacity to offload processing that would otherwise occur on the Central Processor (CP). Because significant portions of all these workloads can run on zIIPs, you can leverage zIIP capacity to scale your database environments without incurring expensive hardware upgrades.

The best results—as measured by percentage of CP processing offloaded to the zIIP and the amount of white space created on the CP—were found with batch/CV processing, CICS/DML requests, JDBC and ODBC distributed requests, CA IDMS/DDS requests, and any kind of external request processing to the database engine. A system architected with CA IDMS application-owning regions (AOR) making requests to CA IDMS database-owning regions (DOR) may also see significant benefits. CA IDMS/DC and CA ADS online applications that execute in the same CA IDMS region as the database will see less benefit because user mode code is not eligible for zIIP execution and cannot be offloaded. Overall results range from a benefit of 10% to 50%. Your actual benefit may vary depending on your mix of CA IDMS work—online, batch, CICS, CA ADS, COBOL, Web, JDBC, ODBC, CA IDMS/DDS, and so on.

Additional Considerations:

Most CA IDMS system code is eligible to run on a zIIP processor. CA IDMS runtime processing ensures that a non-zIIP processor is selected to run non-eligible routines, such as user exits, database procedures, SQL-invoked routines, and application programs.

To implement zIIP support for your system, take the following steps:

1. Ensure z/OS software feature HBB7709 is available on your system.
2. Provide the ZIIP=Y startup parameter in the central version startup JCL.
3. Make sure the specific rules for load module residence for zIIP processing are met:
    - The central version startup routine, typically RHDCOMVS, should reside in an authorized library in the STEPLIB concatenation. It can reside in a linklist library, but this is not recommended.
    - CA IDMS nucleus modules, including all line drivers and service drivers, must be loaded from an authorized load library in the CDMSLIB concatenation or from the Link Pack Area (LPA). The IBM Language Environment library (usually CEE.SCEERUN) must be authorized if it is included in the CDMSLIB concatenation. Alternatively, the following modules must reside in the LPA: CEEPIPI, CEEPLPKA, and CEEEV003.

- z/OS Callable Services library (SYS1.CSSLIB) must be in the linklist or it must be authorized and included in the STEPLIB concatenation.
- Modules that consist of non-executable code or code that is never eligible to run on a zIIP processor do not have to come from a secured location. Most modules that are supplied by a client or that are modifiable at a client site are in this category.

More Information:

For additional information about utilizing zIIPs and using statistics to evaluate the CP utilization reduction, see the ZIIP Exploitation section in the *CA IDMS System Operations Guide*.

Use the High Performance Storage Protect Option

Use the High Performance Storage Protect option (HPSPO) in production CA IDMS systems.

Business Value:

HPSPO protects the integrity of your systems by preventing user programs from overwriting CA IDMS and operating system storage, from overwriting CA IDMS and operating system code, and from executing privileged instructions.

With HPSPO you do not have to make the difficult choice between system integrity and CPU. You can implement storage protection with minimal CPU cost. The use of full storage protection incurs significant CPU time.

Additional Considerations:

While protecting the integrity of CA IDMS and the operating system, HPSPO will not protect non-reentrant user programs and user storage. Use full storage protection in test systems to identify and correct any renegade program before moving it into production.

The High Performance Storage Protect option was introduced with CA IDMS r16 SP2. HPSPO allows you to protect the IDMS system and control blocks from being overlaid by a user mode programs written in CA ADS, COBOL, Assembler, or PL/I.

To use HPSPO, you must specify the following system generation parameters:

- STORAGE KEY IS 9 and PROTECT on the SYSTEM statement.
- PROTECT at the program definition level.

HPSPO also requires special setup for your storage pools. You must segregate all user-oriented storage from CA IDMS system storage. Storage types, user, user-kept, shared, and shared-kept, can be together, but they must be defined to secondary storage pools and must be isolated from any secondary pools that contain database or terminal type storage.

Any user programs that attempt to perform I/O directly through operating system facilities may need to be modified because they must run with storage protection enabled. This ensures that the storage key and the execution key match, which is an operating system requirement for I/O.

To use full storage protection, set the STORAGE KEY IS parameter on the SYSTEM statement to a value between 10 and 15; also specify PROTECT on the SYSTEM statement and at the program definition level.

Use full storage protection for all development, test, non-performance related QA, and pre-production systems. This should help to identify any application errors before the application is moved to production. When full storage protect is enabled, any program attempting to overlay storage that it does not own will abend with a D003 abend code.

For production you should use the High Performance Storage Protect option. This provides for system integrity with negligible CPU overhead in the event that a change is made to an application that has not been tested on a non-production system.

More Information:

For more information about storage protection and the use of HPSPO, see Storage Protection in the *CA IDMS System Generation Guide*.

Maintain Scratch Information in Memory

Use the scratch in memory and extensible scratch options in both central version and local mode environments.

Business Value:

Enabling the use of scratch in memory significantly improves CA IDMS runtime performance by eliminating file I/O to a scratch area. Defining scratch in memory also improves DBA productivity by eliminating the need to define and maintain a scratch area for every CA IDMS system.

Enabling extensible scratch reduces the possibility of task failures due to insufficient scratch space at runtime.

Additional Considerations:

CA IDMS utilizes a temporary working storage area referred to as the scratch area. This temporary storage can either be allocated in memory or allocated on DASD and accessed through standard file I/O.

The SCRATCH IN STORAGE YES system generation parameter and the SCRATCH_IN_STORAGE SYSIDMS parameter specify that the scratch area is memory-resident for central version and local mode processing, respectively. To enable the use of extensible scratch, specify both primary and secondary extent sizes.

If scratch information is not maintained in memory, then a scratch area must be allocated on DASD and physical I/O will be done to the associated file at runtime.

Customers with the following types of workloads will benefit from use of this best practice:

- IDMS SQL, JDBC, or ODBC applications. CA IDMS SQL internal processing utilizes scratch storage extensively.
- CA IDMS DC/UCF COBOL, PL/I, Assembler, or CA ADS applications that issue Scratch commands.

More Information:

For more information about scratch information in memory and extensible scratch, see the *CA IDMS System Generation Guide* and the *CA IDMS Database Administration Guide*.

Tailor System Generation Parameters

Set CA IDMS system generation parameters to the recommended value in the tables shown in Additional Considerations, and monitor your system performance to fine tune the parameter setting. It is usually best to over-allocate resources such as storage pools, program and reentrant pools, RCEs, and so on. As a general rule, it is better to be slightly over-allocated at the beginning and allow the system to run through a peak load period. You can then use the peak load statistics to reconfigure the system to optimum levels.

Business Value:

Adjusting the CA IDMS system generation parameters appropriately for your site will provide optimal runtime performance of a CA IDMS system. This will hel p to meet Service Level Agreements for response time and effectively use available computing resources.

Additional Considerations:

The following tables contain recommended values for CA IDMS system generation parameters. The tables contain recommendations for the following system generation statements:

- SYSTEM
- ADSO
- PROGRAM
- TASK
- STORAGE POOL and XA STORAGE POOL

The parameter values fall into these three categories and are listed under the column Recommendation in the tables that follow.

- **Best Setting**-If there is a value in the Recommendation column, then this value has been found to be the best setting for that parameter.
- **Set then Tune**-If there is a value in the Recommendation column, followed by <site specific>, then the value listed is a reasonable starting point but will need to be monitored and tuned using available CA IDMS Performance Monitor functions and commands, as well as various DCMT and OPER commands.
- **Site specific**-If <site specific> is listed in the Recommendation column, then you need to determine the appropriate values for your site. Guidelines for determining the appropriate value are provided in the notes following the table. When upgrading to a new release of CA IDMS, the site specific parameters related to Storage and Program Pools should be set at a minimum to the values specified in the previous release; however, we recommend that you set the values 10 to 20 percent higher, and then tune down after running through your busiest processing period.

System Statement

The following table represents recommended values for parameters set using the SYSTEM statement.

Parameter	Recommended Setting	Affects	Benefit
ABEND STORAGE is	600	Program recovery	System stability
CUSHION is	10% of Pool 0	Reserved storage for executing tasks	System stability
DEADLOCK DETECTION INTERVAL is	1	Time to wait before detecting a deadlock	Impacts CPU, affects response time and throughput
JOURNAL	NOJOURNAL retrieval	Reduces I/O to journals	Saves CPU, reduces response time, increases throughput

JOURNAL FRAGMENT INTERVAL	0	Writes status information to journals	Reduced I/O to journal, saves CPU, reduces response time, increases throughput
JOURNAL TRANSACTION LEVEL	5	The fullness of a journal page	Reduced I/O to journal
LIMITS for online and LIMITS for external	OFF	Task resource usage	Saves CPU, reduces response time, increases throughput
LOADLIST is	SYSLOAD	Specifies load sequence for programs and dialogs	Saves CPU, reduces response time, increases throughput; see LOADLIST note
MAXIMUM ERUS is	<site specific>	Limits external concurrent access	Impacts CPU, affects response time and throughput; see MAX ERUS/MAX TASK note
MAXIMUM TASK is	<site specific>	Limits online concurrent access	Impacts CPU, affects response time and throughput; see MAX ERUS/MAX TASK note
PROGRAM POOL is	<site specific>	Size of below the line program pool	Impacts CPU, affects response time and throughput; see PROGRAM POOLs note
PROTECT	PROTECT	Allows the system to protect itself from storage overlays	System stability; see Use the High Performance Storage Protect Option section
QUEUE JOURNAL	BEFORE	Specifies which queue images are written to the journals	Reduced I/O to journal, saves CPU, reduces response time, increases throughput
REENTRANT POOLS is	<site specific>	Size of below the line reentrant program pool	Impacts CPU, affects response time and throughput; see PROGRAM POOLs note
RELOCATABLE THRESHOLD	NO	Writes certain types of storage to the scratch area	Saves CPU, reduces response time, increases throughput
RETRIEVAL	NOLOCK	Maintenance of locks for records in areas accessed in shared retrieval	Saves CPU, reduces response time, increases throughput
RUNAWAY INTERVAL is	5	CPU seconds before abending a task as a runaway	Stability, impacts CPU, affects response time and throughput

RUNUNITS FOR	5 <site specific>	Preallocates run units for LOADER, MSGDICT, QUEUE, SECURITY, SIGNON, SYSTEM/DEST	Saves CPU, reduces response time, increases throughput
SCRATCH IN STORAGE	YES	Eliminates the need for a physical scratch file	Saves CPU, reduces response time, increases throughput
STACKSIZE is	3000	Size of system internal work area	Stability
STORAGE POOL is	1200 <site specific>	Size of below the line storage pool	Impacts CPU, affects response time and throughput; see STORAGE POOL note
SYSLOCKS is	400,000 <site specific>	Number of locks the system should pre-allocate	Impacts CPU, affects response time and throughput
SYSTRACE	OFF	Internal tracing	Saves CPU; see SYSTRACE note
TICKER INTERVAL	1	Time to wake up system to check for work	Impacts CPU, affects response time and throughput, affects all timer functions within CA IDMS
UPDATE	NOLOCK	Maintenance of locks for areas in protected update usage mode	Saves CPU, reduces response time, increases throughput
XA PROGRAM POOL is	<site specific>	Size of above the line 31-bit non- reentrant pool	Impacts CPU, affects response time and throughput; see PROGRAM POOLs note
XA REENTRANT POOL is	<site specific>	Size of above the line 31-bit reentrant pool	Impacts CPU, affects response time and throughput; see PROGRAM POOLs note
XA STORAGE POOL is	<site specific>	Size of above the line 31-bit SYSTEM storage pool	Impacts CPU, affects response time and throughput; see PROGRAM POOLs note

Notes:

LOADLIST

LOADLIST tailoring can provide significant CPU savings and response time reduction but incorrect specification can and will result in exactly the opposite. Initially try using the default SYSLOAD loadlist with the NODYNAMIC option of the Program Definition statement.

MAX ERUS/MAX TASK

There are no magic numbers for these two parameters. They need to be determined by each site. We recommend that you specify high values for both on the system statement, and then use the following command to vary max tasks up or down until the optimum number is found.

```
DCMT VARY ACIVE TASK MAX TASK nnn
```

You could initially set both parameters to 100. If you are using CA IDMS/DC, then you set the MAX ERUS parameter to accommodate any batch to CV jobs you may need to run. Sites that only run CICS COBOL or PL/I DML transactions can set the MAX TASK value to a very low number; 5 is recommended. A low value allows for any online work that may need to be processed (for example, signing on through the system console).

CICS TS sites should also note that if the value of the CA IDMSMAX ERUS parameter is set below the CICS MAXTASK parameter, they could experience 1473 abends. If MAX ERUS is reached, meaning all the allocated ERE control blocks that maintain the link between the external session and the central version are in use, the next CICS task that needs to interact with the central version will not be able to and will abend with a 1473 error code. You can prevent this by specifying a value for MAX ERUS that is higher than the CICS MAXTASK parameter value. In this way, CICS tasks will be able to acquire an ERE. You can then control the number of CICS tasks that are allowed to run within the central version using the DCMT VARY ACTIVE TASK MAX TASK command.

PROGRAM POOLs

Select values for your PROGRAM POOL parameters and then tune them. You can determine the appropriateness of your current settings by using various statistics such as the output of the DCMT DISPLAY ACTIVE PROGRAMS and DCMT DISPLAY ACTIVE REENTRANT PROGRAM commands.

When upgrading to a new release of CA IDMS, the size of all below the line PROGRAM and REENTRANT pools should be allocated at the size they were in the prior release.

STORAGE POOL

The STORAGE POOL parameter specifies the size of storage pool 0 that is used for system type storage. This pool could and would be used for other types of storage, user and shared, if there were no other pools defined in the system; however, you will always define other pools. The 1200 K allocation is probably on the high side, and can be reduced after determining how much storage is used during your peak processing period. Use statistics displays from DCMT, OPER, and CA IDMS Performance Monitor to help determine the appropriateness of your current setting. For example, the OPER WATCH STORAGE command displays the number of times a short-on-storage (SOS) condition occurred and the number of waits for storage.

SYSTRACE

Internal system trace data is sometimes needed for debugging. When a problem exists, CA Technical Support may ask you to set the SYSTRACE parameter to ON and set the number of entries to 9999 using SYSGEN or higher using DCMT vary commands.

XA STORAGE POOL

This parameter controls storage pool 255 that is the XA equivalent of storage pool 0. This pool must be defined. All system type storage is allocated out of this pool. If this pool fills, the storage pool 0 will be used; however, using the storage pool 0 should be avoided. An allocation of 12000 is likely higher than what you will actually need, but it is better to over-allocate than under-allocate, as mentioned earlier.

ADSO Statement

The following table represents recommended values for parameters set using the ADSO statement.

Parameter	Recommended Setting	Affects	Benefit
DIALOG STATISTICS	off <site specific>	Collects dialog statistics	Saves CPU
FAST MODE THRESHOLD is	OFF	Writes RBBs and other information to SCRATCH	Saves CPU, reduces response time, increases throughput
RESOURCES are	FIXED	Writes ADS related storage to SCRATCH	Saves CPU, reduces response time, increases throughput

Program Definition Statement

The following table represents recommended parameter values for each PROGRAM statement.

Parameter	Recommended Setting	Affects	Benefit
NODYNAMIC	NODYNAMIC	Dynamically look for newer versions in LOADLIST	Saves CPU, reduces response time, increases throughput
PROTECT	PROTECT	Program runs with storage protection	System stability; see Use the High Performance Storage Protect Option section

Task Definition Statement

The following table represents a recommended parameter value for each system generation TASK statement.

Parameter	Recommended Setting	Affects	Benefit
PROTOCOL is	EXPRESP	IDMS waits for a response from VTAM	Reduces response time and increases throughput

STORAGE POOL and XA STORAGE POOL Statements

The following table represents recommended parameter values for the STORAGE POOL and XA STORAGE POOL statements. These statements are used to create user-defined secondary storage pools.

Parameter	Recommended Setting	Affects	Benefit
SIZE	<site-specific>	Size of above the line and below the line storage allocated	Impacts CPU utilization, response time, and throughput; see STORAGE POOLs note
CONTAINS TYPES	See STORAGE POOLs note	Ability to use High Performance Storage Protect Option	Impacts system integrity and performance
CUSHION	5% of pool size	System stability	System stability
NOPGFIX		Suppresses page fixing	Impacts performance
RELOCATABLE THRESHOLD	NO	Writing relocatable storage to scratch	Saves CPU, reduces response time, increases throughput

Notes:

STORAGE POOLs

The size of USER storage pools, those numbered 1 thru 127 for below the line pools and numbered 128 thru 254 for above the line storage pools, should initially be allocated to at least the size of the USER storage pools used in the prior release. As mentioned previously, it is best to allocate these pools larger than they were in the prior release. You may want to increase the size by 25 percent initially, and then cut them back after finding the true HWM after processing through your peak period.

The below the line and above the line storage pools should be defined as shown in the following lists.

Below the line storage pools:

- One storage pool that contains types (User, User Kept)
- One storage pool that contains types (Shared, Shared Kept)
- One storage pool that contains types (Terminal, Database)

The sizes of these below the line storage pools can be very small as the amount of below the line storage allocated is minimal.

Above the line (XA) storage pools:

- One storage pool that contains types (User, User Kept)
- One storage pool that contains types (Shared, Shared Kept)
- One storage pool that contains types (Terminal, Database)

As a size recommendation, we suggest that you overallocate. If you had one XA storage pool that contained all types in the prior release and its size was 10 M, allocate each pool at 10 M. You will not need all of this storage. After running through your peak processing time ,you can determine the HWMs and adjust as needed.

The High Performance Storage Protect Option requires the storage pools be defined in this manner. These definitions are also valid for non-protect, as well as full storage protect, usage.

More Information:

For detailed descriptions of all system generation (SYSGEN) options and parameters, see the *CA IDMS System Generation Guide*.

For descriptions and examples of all DCMT and OPER commands that can be used to monitor resources within the CA IDMS system, see the *CA IDMS System Tasks and Operator Commands Guide*.

For detailed online and batch reporting facilities available in the CA IDMS Performance Monitor product, see the *CA IDMS Performance Monitor User Guide* and the *CA IDMS Performance Monitor System Administration Guide*.

Use CICS Threadsafe Applications

Define eligible CICS threadsafe application programs that use the CA IDMS interface as CONCURRENCY (THREADSAFE), API (OPENAPI), or both in the CICS TS System Definition (CSD) file.

Business Value:

Using CICS threadsafe applications can significantly increase throughput of tasks in the CICS region, improve response time and decrease costs. The CICS TS Open Transaction Environment (OTE) enables multiple tasks to execute the same THREADSAFE program simultaneously operating on different TCBs.

Additional Considerations:

CICS programs that are truly threadsafe should be declared with the CONCURRENCY attribute of THREADSAFE.

Note: A program may be threadsafe even though it contains a non-threadsafe EXEC CICS command.

A program that meets the following criteria should also be declared with the API attribute of OPENAPI:

- The program is threadsafe.
- The program contains no non-threadsafe commands or minimal non-threadsafe EXEC CICS commands (for example, a single EXEC CICS SEND command that is executed as the last command before the program terminates).

Programs defined with the OPENAPI attribute will be dispatched on an OPEN TCB and returned to an OPEN TCB after temporary assignment to the Quasi-reentrant (QR) TCB during the execution of a non-threadsafe command.

Carefully evaluate each program before declaring it with program attributes of THREADSAFE or OPENAPI. If a non-threadsafe program is defined as THREADSAFE, unpredictable results may occur, including task or system abends and data corruption. If a program containing many non-threadsafe EXEC CICS commands is declared as OPENAPI, performance may suffer due to excessive swaps between the OPEN TCB and the QR TCB.

More Information:

For general information on CICS TS threadsafe programming, consult the appropriate IBM documentation.

For more information on implementing CICS threadsafe programming with the CA IDMS interface, see the *CA IDMS System Operations Guide*.

Use the Type 4 JDBC Driver

Use the CA IDMS Server type 4 JDBC driver when executing Java programs that use JDBC to access CA IDMS databases.

Business Value:

The CA IDMS Server JDBC driver is tightly integrated with CA IDMS SQL processing for optimized SQL access to both non-SQL and SQL-defined CA IDMS databases.

The type 4 JDBC driver provides for optimal performance and response time for web and client/server Java applications accessing CA IDMS. With the type 4 JDBC driver, the Java client program is able to communicate directly with the CA IDMS region through native TCP/IP. This eliminates the need for any middleware address space, such as CCI, and frees mainframe processing for other workloads.

Additional Considerations:

CA IDMS Server supports the four types of JDBC drivers as defined by Sun.

The four types of JDBC drivers differ in the way they communicate with the database and whether they use native code on the client platform:

- Type 1 JDBC driver

  Uses an ODBC driver to communicate with the database. ODBC drivers are always implemented in native code. The JDBC-ODBC Bridge is a basic Type 1 driver that is rarely used now that database specific drivers are available.

- Type 2 JDBC driver

  Invokes the native client interface to communicate with the database.

- Type 3 JDBC driver

  Uses a generic network protocol to communicate with a middleware server that invokes the native client interface to communicate with the database. It uses no native code on the client platform.

- Type 4 JDBC driver

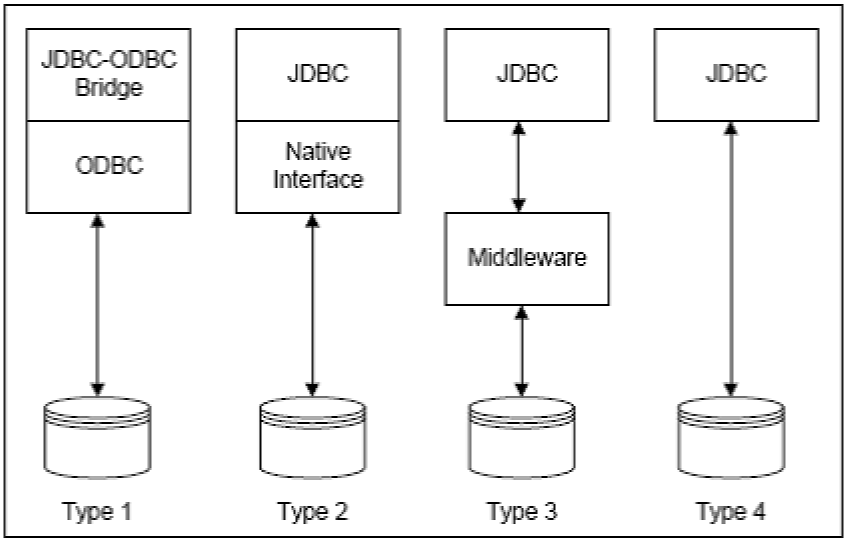
  Communicates directly with the database using its proprietary protocol. It uses no native code on the client platform. This provides the most flexible and efficient way of communicating with the back-end database.

  To enable the type 4 JDBC driver, define a TCP/IP line and a listener PTERM on the mainframe and specify the host name and port on the client. A Java application can establish a connection through the type 4 JDBC driver using either an IdmsDataSource object or the static DriverManager class.

The type 4 JDBC driver supports a large number of concurrent connections and is essentially limited only by the resources allocated to your CA IDMS system. In particular, you may need to increase the size of storage pools to support high volume applications that create many concurrent connections.

CA IDMS uses a minimum of 250 KB of storage for each dynamic SQL session, whether started by the type 4 driver, the type 2 or ODBC drivers, OCF, or IDMSBCF.

The following illustration shows the layers of processing involved for each type of JDBC driver in the client/server architecture.



More Information:

For more information about the type 4 JDBC driver and detailed information on how to specify client settings, see the *CA IDMS Server User Guide*.

For detailed information about defining the TCP/IP line and listener, see the *CA IDMS System Generation Guide*.

Configure Your CICS Environment

The IBM CICS Transaction Server (CICS TS) product can be used as a front-end TP monitor with back-end CA IDMS central versions in multiple ways. The two most common ways are listed here:

- Use CICS TS as the online application program execution environment and access CA IDMS only for database services. This type of usage requires assembling the CICSOPT macro to create an IDMSINTC module. The CA IDMS CICS interface program IDMSINTC is created by assembling the CICSOPT macro and linking it as described in the *CA IDMS System Operations Guide*.

- Use the CA IDMS/UCF product with CICS TS to enable CICS terminals to access online application programs that execute in the CA IDMS/DC environment. In addition to creating an IDMSINTC module, this type of usage also requires assembling the #UCFCICS macro. The CA IDMS/UCF front-end module UCFCICS is created by assembling the #UCFCICS macro with other macros and linking the resulting object modules as described in the *CA IDMS System Operations Guide*.

When new features are introduced for a new CA IDMS release, the default values for the CICSOPT and #UCFCICS macros are set so that the new features are not enabled. This ensures that customers using the default values will not encounter unexpected results when they upgrade to a new release. As a result, however, there are cases where the recommended best practice differs from the default setting.

The following best practices apply when configuring CICS TS for use with CA IDMS:

- Set the TPNAME parameter of the CICSOPT macro to null or blanks and modify the CICS TS System Initialization Table (SIT) for each CICS region accessing the same CA IDMS system to have a unique SYSIDENT value.

  Business Value:

  Multiple CICS regions can use the same IDMSINTC load module to access the same CA IDMS back-end system. This best practice facilitates the setup and maintenance for customers that have multiple CICS regions that access the same CA IDMS central version. This improves DBA productivity and provides for consistent CICS TS usage with CA IDMS. It enables additional CICS regions to be easily added if there is increased workload for CA IDMS processing.

  Additional Considerations:

  In addition to coding the CICSOPT TPNAME parameter value as null or blanks, you must also modify the CICS TS System Initialization Table (SIT) for each CICS region to ensure that each region has a unique SYSIDENT value. When the CICSOPT TPNAME parameter is null or blanks, then the IDMSINTC interface will use the SYSIDENT of the CICS region as a unique identifier.

  The alternative is to have multiple CICS regions that use the same SYSIDENT value accessing the same CA IDMS region. In this case, each CICS region must have its own IDMSINTC interface created by coding a CICSOPT macro with a unique TPNAME value. This creates more setup work for the DBA and requires more work to consistently maintain each IDMSINTC module.

  More Information:

  For more information about the CICS TS SIT parameters, see the appropriate IBM documentation. For more information about the CICSOPT parameters, see the *CA IDMS System Operations Guide*.

- Use the SYSCTL DDNAME parameter of the CICSOPT macro instead of the SVC and CVNUM parameters to specify the CA IDMS system to access.

Business Value:

If the same CICS region is used to access multiple CA IDMS backend systems, then a separate IDMSINTC interface module must be created for each unique CA IDMS central version. By using this best practice, the name of the CA IDMS system to be accessed can be controlled at runtime. This provides for flexibility and enables a customer to respond quickly to changes in system workloads by bringing up additional CA IDMS back-end systems.

Additional Information:

The CICS TS startup JCL must include a ddname that matches the CICSOPT SYSCTL DDNAME parameter. It must point to the same dataset as the one specified in the SYSCTL DD statement in the CA IDMS system startup JCL.

More Information:

For more information about the CICSOPT parameters, see the *CA IDMS System Operations Guide*.

- Use the USERCHK parameter on the #UCFCICS macro and require each user to sign on to CICS TS.

  Business Value:

  Using this best practice provides enhanced security. If a UCF session terminates abnormally, a subsequent user will not inadvertently gain access to the information from the previous user's session.

  Additional Information:

  Use CICS TS installation security to require each user to sign on to CICS TS.

  More Information:

  For more information about CICS TS security, see the appropriate IBM documentation. For more information about #UCFCICS parameters, see the *CA IDMS System Operations Guide*.

Index