CA Gen

Best Practices Guide

Release 8.5

Third Edition

CA Technologies Product References

This document references the following CA Technologies products:

- CA Gen

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, c*ontact 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:

- Introduction (see page 9)—Streamlined and improved.
- Your Product Installation and Configuration Best Practices > Implement a proactive Preventive Maintenance Strategy (see page 11)—Added to the guide.
- Inline Code (see page 27)-Added best practices that must be followed while adding Inline Code Statements.
- BLOB Functions in Action Diagram (see page 29) - Added best practices that must be followed while using the BLOB functions in an Action Diagram.

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.

Chapter 2: Your Product Installation and Configuration Best Practices

This section contains the following topics:

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 Chorus Software Manager Enabled 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. CA RS levels are published monthly that include PTFs, HIPERs and PRPs (PE-resolving PTFs). Before you download, apply, and test a new CA RS level, we recommend that you accept the previous 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 the FIXCAT method.

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

Important! CA Chorus™ Software Manager (CA CSM) - 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 CSM to maintain your CA Technologies z/OS-based products.

More Information:

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

Chapter 3: Installation

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

Business Value

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

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

Note:

- For more information about CA MSM, see the *CA Mainframe Software Manager Product Guide*.
- For more information about product installation, see the *CA Gen z/OS Installation Guide*.

Chapter 4: Host Encyclopedia

File Transfer Methods

Host Encyclopedia uses two file transfer methods - Seamless File Transfer and Manual File Transfer. The manual file transfer method is the only one available if your site communications are not compatible with the seamless file transfer method. With FTP, you must be aware of where the file is transferred from or to on the mainframe. Manually use the Host Encyclopedia to create the transaction file on checkout before doing the file transfer, or read the transaction file on check-in after doing the file transfer. The trade-off for using the manual method as opposed to the seamless method is determined by whether your site supports compatible communications and if so, the size of the transaction file transferred. If the file is large, it is worth taking the extra steps that are involved in the manual file transfer as the file is transferred much faster. If the file is small, the seamless method is preferable due to its simplicity.

Use Seamless File Transfer

Seamless file transfer uses TSO to transfer update.trn files to the Host Encyclopedia and checkout.trn files from the Host Encyclopedia.

Business Value

Seamless file transfer allows for easier file transfers because you do not need to be aware of where the file is transferred to or from on the mainframe. This saves time for you since it increases system performance.

Additional Considerations

Seamless file transfer moves data in 2K chunks so it is best suited for use with small model or subset downloads, or small uploads.

Use Manual File Transfer

Manual file transfer uses FTP (File Transfer Protocol) to transfer update.trn files to the Host Encyclopedia and checkout.trn files from the Host Encyclopedia.

Business Value

FTP transfers files at a high rate of speed so file transfers complete faster, saving time and system resources.

Model Management

There are three preferred models when working with the Host Encyclopedia:

- Integrated Project Model
- Repeated Data Project Model
- Master and Partial View Data Model

Use Integrated Project Models Method for Small Groups

Integrate all development project models into a single production model to simplify production model management.

Business Value

When you integrate all project models into a single production model, you can make a change to all project models by making a single change to the production model. Model Management is simplified because changes are controlled through one model.

Additional Considerations

For large projects, the production model may become unmanageable so this method is recommended for small groups of integrated applications or as a starting point for larger projects.

Use Repeated Data Models Method to Simplify Model Management

A master model is used to manage the logical and physical data design so the data model, technical design, and Referential Integrity (RI) triggers are migrated to other project models from the master model. Changes to the logical or physical design are only allowed in the master data model.

Business Value

Using a master model allows the model size to be more manageable. In addition, logical and physical design maintenance is isolated from projects because the designs are managed in the master model. Also, diverse project schedules now can be accommodated. Since changes are controlled through the master model, model management is simplified.

Additional Considerations

Since the data model and data structure are identical in all models, the RI triggers will be the same regardless of the model used as the source. Model management may become more complex because of multiple project models. Project models may contain data definitions that are not used for that project.

Use Master and Partial View Data Models Method

Similar to the Repeated Data Models method, a single model is used to manage the logical and physical data design. The information removal action blocks (Delete, Disassociate and Transfer) are maintained in this model. Project models only contain the data required to support the project and reference the information removal action blocks from the master data model rather than implement them.

Business Value

By isolating the logical and physical data models as well as the action blocks that call triggers, the Referential Integrity (RI) logic maintains its integrity. Using a master and partial view data model allows the model size to be more manageable. Since changes are controlled through the master and partial view data model, model management is simplified.

Additional Considerations

Some manual procedures are required to ensure that the RI triggers and information removal action blocks are available to other projects. Like the Repeated Data Model method, model size is more manageable as model management becomes more complex. Unlike the Repeated Data Model method, each project model only contains the data definitions it requires from the master data model.

Space Management

Host Encyclopedia space management consists of estimating the database size, partitioning large tablespaces, cleaning up the IEFXLOG files, and disabling the model object history.

Estimate Encyclopedia Database Size

You can control the amount of DB2 disk space by allocating for a small or medium sized host encyclopedia. For more information on space requirements, see guidelines in the *CA Gen z/OS Installation Guide*.

Business Value

Allocating disk space for a small or medium sized host encyclopedia improves disk usage and will reduce space costs. In addition, it improves the I/O throughput.

Partition Large Encyclopedia Tablespaces

To enable parallel execution of DB2 utilities, partition large encyclopedia tablespaces and indexes. For more information, see the partitioning guidelines in the *CA Gen Host Encyclopedia Administration Guide*.

Business Value

Parallel execution of DB2 utilities may reduce the total elapsed time used by the encyclopedia housekeeping jobs. This improves the encyclopedia availability for the developers, improving development throughput.

Cleanup IEFXLOG files

This sequential file is allocated for each CA Gen encyclopedia user as *userid*.IEFXLOG. It contains a log of upload and download processes with their CPU times and elapsed times, and can be used to record similar statistics for all encyclopedia activities if the statistics option is turned on using %TIUDEBUG R command.

Business Value

The *userid*.IEFXLOG files increase in size because new log records are appended to them. Archive and delete these files at regular intervals to reduce the space consumption and to improve performance. CA Gen would reallocate them automatically if they are missing. Archiving and deleting the *userid*.IEFXLOG files improves disk usage, reduces space costs, and improves I/O throughput.

Disable Model Object History

The Host Encyclopedia Object History facility can capture more detailed history of changes to specific objects in DHOBJ if you enable the model object history.

Business Value

Logging can be enabled or disabled for object migration and object changes. The default for a new model is logging enabled. Disable this option if the history information is not required. Review objects in the DHOBJ table. Disabling object history reduces the additional processing required to capture the object history.

Host Construction

Host Construction code generation is often the highest overall CPU consuming encyclopedia activity, but is rarely identified as such. This is probably because the code generation is fast and interactive.

Reduce Code Generations

The following measures could reduce the number of code generations:

- Prioritize and group enhancements to avoid duplicate generation
- Use Application System Regeneration to generate modules which require to be regenerated only.
- Dynamically link frequently changed modules
- Use non-host code generation using the Workstation or the CSE construction where possible.

Business Value

Reducing code generations lowers system costs.

Migration

Managing the performance of migration consists of reducing migration size and avoiding trial migration.

Reduce Migration Size

Migration does not perform interim database commits but commits at the end of processing. As a result, large migrations may cause database contentions. It is important to keep the aggregate sets small. Consider Model Copy as an alternative to a large migration.

Business Value

Small migrations improve the encyclopedia concurrency.

Avoid Trial Migration

Large migrations may cause database contentions. The trial migration is equivalent to a migration, but it also performs a database rollback in the end; therefore, avoid trial migrations. A combination of When Changed report and a Compare report can provide a complete list of objects needing migration.

Business Value

Avoiding trial migrations reduces the system costs and improves the encyclopedia concurrency.

Subsetting

Managing the performance of subsetting consists of defining the smallest subset possible, requesting minimum protection and expansion, avoiding the use of subsetting to delete objects, avoiding long check out times of subsets, and using an expansion conflict report before subset checkout.

Define the Smallest Subset Possible

When defining a subset, scope only on objects that are necessary. For example, do not scope on a business system unless you need to change the system's default information (system commands, PF keys, screen templates, exit state, video attributes, custom video properties, and edit patterns).

Business Value

Smaller subsets result in:

- Lower cost for checkout
- Reduced protection conflicts between users of a model
- Minimum impact of current access to the encyclopedia

Note: For more information about how to define subsets, see the *CA Gen Host Encyclopedia Subsetting User Guide*.

Request the Minimum Protection and Expansion

When scoping objects, request the minimum protection and expansion needed to accomplish your task. For example, do not request full expansion unless it is required for the task. Do not request Delete protection for a scoping object unless you need to delete a component. Any other subset that needs the object will get it with only Read protection. Avoid requesting Modify protection when Access works as well.

Business Value

Minimize potential conflict over objects that may be shared across subsets by requesting the minimum protection and expansion needed to accomplish your task.

Avoid Using Subsetting to Delete Objects

Use the Delete Object option from the Host Encyclopedia Model Management menu whenever possible instead of subsetting to delete a shared object.

Business Value

Using the Delete Object option to delete a shared object eliminates the need to define and check out a subset which includes the object with delete protection along with all the objects that reference it. This can be an extremely large subset.

Avoid Long Check Out Times of Subsets

Subsets checked out for a long time can cause downgrades for other subset downloads. Downgrades usually result in consecutive overrides and downloads. This cycle degrades the overall throughput of the encyclopedia. In the development phase of an application, the subsets tend to be checked out for a long time. This may not cause many downgrades for other developers since subsets are usually mutually exclusive at this stage. However, in the maintenance phase of an application, the subsets must be scoped for the current task and must have a quicker download and upload turnaround time as compared with the development phase.

Business Value

Avoiding long checkout times improves throughput of the encyclopedia, increases user productivity, and reduces system costs.

Additional Considerations

Examining the cause of each downgrade is a good starting point in determining the maximum duration that a subset can stay checked out for each project and its implementation phases. Alternatively, a report of subsets that have been checked out for more than one week can be brought to project teams' attention regularly. Some Gen sites are known to override all subsets checked out for more than one week to enforce frequent check-ins.

Use Expansion Conflict Report before Subset Checkout

Use the Host Encyclopedia Expansion Conflict report prior to subset checkout to avoid potential downgrades. This report shows the objects currently checked out through other subsets. Review this report prior to a subset checkout to avoid potential downgrades.

Business Value

Expansion Conflict Report may help reduce the system costs where downgrades occur frequently. The report has minimum impact of current access to the encyclopedia and uses less system resources as compared to checkout.

Chapter 5: Implementation Toolset

Implementation Toolset for z/OS

CA Gen lets you select the dataset storage type, initial database size, and secondary storage factor on the Set DB2 Variables panel during configuration.

Sizing the Database

The dataset storage type options are 'S' for Storage Group and 'V' for VSAM. The initial database size options are 'S' for Small (approximately 329 tracks of 3390 DASD) or 'M' for Medium (approximately 2303 tracks of 3390 DASD). The default primary and secondary number of records is derived from the selection of the database size (Small/Medium) for each table of the z/OS Implementation Toolset (IT). The secondary storage factor is used to increase the default primary and secondary sizes by a factor that is input by the customer on the Set DB2 Variables panel. Use the secondary storage factor to fine tune your database size requirements by a factor of .01 to 9.9. For example, if the secondary storage factor is 4.5, the primary size will increase by a factor of 4 (4 times the primary amount), and the secondary size will increase by a factor of 5 (5 times the secondary amount).

Business Value

Fine tuning the size parameters reduces storage resources that may be over-allocated for the database and ultimately would reduce the associated system costs.

Chapter 6: Middleware Best Practices

IBM CICS Transaction Gateway

The IBM CICS transaction gateway consists of the configuration tool and is shipped with a Java runtime environment.

Configuration Tool

When using the IBM CICS Transaction Gateway Configuration tool for the first time, make a backup copy of your original ctg.ini file.

Business Value

If you make changes using the IBM CICS Transaction Gateway Configuration tool and save the file, your current ctg.ini file will be overwritten. A backup version of the ctg.ini file called ctg.bak will be created automatically. Using the Configuration tool to make and save additional configuration changes will overwrite the ctg.ini file and the ctg.bak file so saving the information in the original ctg.ini file will ensure you do not lose any configuration information.

Java Runtime Environment

The IBM CICS Transaction Gateway is shipped with its own Java Runtime Environment. After installing the IBM CICS Transaction Gateway on a new system, open Command Prompt and run the ctgjava command to specify which Java Runtime Environment will be used with the IBM CICS Transaction Gateway. The following example is shown with the default installation path for IBM CICS Transaction Gateway v7.2:

```
ctgjava -s="C:\Program Files\IBM\CICS Transaction Gateway\jvm15\bin\java.exe"
```

Use the 'ctgjava –v' command to display the current JVM being used.

Business Value

Using the correct JVM with the IBM CICS Transaction Gateway ensures the software is configured and works correctly.

CICS Socket Server Timeout Values

It is critical to understand and set appropriate timeout values of the CICS Socket Server.

Business Value

When the timeout values are set to the appropriate values for your business, the client server applications operate with the best throughput possible. By using the TIRSLEXT and TIRSLTMX exits, the value a timeout is set to can be controlled by trancode, CICS region and down to microseconds.

Additional Considerations

The timeout values can be defined in two ways.

- Set the values when the CICS Socket Server is defined in your CICS region. The shortest timeout allowed using this method is 1 second.
- Set the values in the TIRSLEXT and TIRSLTMX exits. Using the exits lets you control the timeout values based on the CICS Socket Server trancode, CICS region, or both as well as being able to set the timeout value to microseconds.

No matter which method you use to set the timeout values, you want to have very few GIVESOCKET or NO INPUT DATA errors when processing requests. It may take several times of trying different settings to find the combination that is right for your environment.

Chapter 7: Toolset

Inline Code

The following list consists of our recommendations while using the inline code in Toolset:

- Use the inline code statement for code that cannot be written in CA Gen Procedure Action Diagram (PAD). For example, IF statements can be written in CA Gen but calls to the external functionality cannot. So, use the inline code statement to call external functionality.
- If the inline code statement involves multiple steps, consider splitting each step into a separate inline code statement. This splitting helps debugging because you can use the CA Gen Diagram Trace Utility to determine which inline code statement is causing problems rather than having to use a source code debugger. Exceeding the limit of number of lines per statement, that is imposed by the compiler, may cause trouble in either displaying, generating, or compiling the statement.
- Copy all references to view variables to locally defined variables in the inline code statement before using them. Copy all locally defined variables back to view references before exiting the statement. This is especially true for host variables because the host variables generated for ENTITY ACTION views are not accessible in the inline code statement.
- To trace the inline code, use the native code debugger. Set the CA Gen Build Tool variables so that you can compile the source code of the parent action diagram with debug enabled and the native code debugger works correctly.
- The Java Import statements and C# Using statements can be specified in the Global Variable Declarations field. All the Import or Using statements must be specified at the beginning of this field and nowhere else.
- If the inline code statement contains the delimiter (##) of view reference as data, these delimiter characters may cause invalid code generation or failure of code generation. To avoid this scenario, follow these steps:

    1. To refer a string literal containing the ## character sequence, declare and use a global variable.
    2. To escape the ## character sequence, use the constructs specific to your language.
        - Example in C: \x23#<text>#\x23
        - Example in C# and Java: \u0023#<text>#\u0023

- The C Include files and COBOL Copybooks may be specified in the Global Variables field of the INLINE CODE statement. References to COBOL Copybooks are included in the WORKING STORAGE section of the code.
- Do not use the TAB key while typing inline code statements or declarations in COBOL.
- For COBOL, the Global Variables field may be used to specify the following sections:
  - INPUT-OUTPUT SECTION.
  - FILE-CONTROL.
  - DATA DIVISION.
  - FILE SECTION.

  These sections may be added to support file access from COBOL action diagrams.

  Note: Type these sections in upper-case followed by a period as shown above.

Important! The Inline Code statement allows the user the ability to add code for any supported language, database, and environment that is supported by CA Gen, but the user will occasionally encounter circumstances where the code will not compile, link, or execute. This may be true even for code that was previously working in an External Action Block.

The following are a couple of mainframe specific types of coding, that we are aware of, that the current implementation of the inline code statement will not support:

EXEC CICS: These types of statements will not work even if CICS is selected as the target environment for the application. The reason is that EXEC CICS code must be translated before it can be compiled and using either the separate CICS translator or the CICS-integrated translator modifies the application code in ways that require extra processing that is currently not handled by Gen application generation and installation.

WebSphere MQ statements: While it may be possible to reference queues that were created to use with the Gen application, because we have no mechanism currently to define other queues to the application so this will make it difficult to use MQ constructs within an Inline Code statement. Also, there are differences in the link deck that are required to use MQ API statements which are currently not supported by Gen application installation.

There are bound to be other circumstances where code will not work in an Inline Code statement and we welcome input from our customers about these situations and will update the Best Practices Guide to document them so that other users will benefit from the experiences. We also welcome any input into how we might be able to support these unsupported coding circumstances and, depending on the level of customer desire for the functionality, we will add this functionality to our backlog to be worked on at the soonest possible time.

BLOB Functions in Action Diagram

The BLOB functions let you add algorithms to the logic in action diagrams and action blocks.

The following points must be considered while using the BLOB functions in an Action Diagram:

- Avoid assigning a NULL value to either of the input BLOBs. The function exits with an invalid LOB handle error message.
- When the length of the input BLOBs are zero, that is, the input BLOBs are empty, the function returns without any action.

CONCATBLOB

The following point must be considered while using the CONCATBLOB function in an Action Diagram:

- If the concatenated length is greater than the length of destination, the output is truncated to available destination length.

SUBSTRBLOB

The following points must be considered while using the SUBSTRBLOB function in an Action Diagram:

- If index is less than zero, the substring of the BLOB is from the zeroth index of the input BLOB.
- If the length of the substring to be retrieved is less than zero, SUBSTRBLOB sets the length to zero.
- If the length attribute given to the function is greater than the length of the BLOB, the output is truncated to hold BLOB until the end of BLOB.

Text Conversion Functions – BLOBDBCS, BLOBMBCS, BLOBTEXT, DBCSBLOB, MBCSBLOB, TEXTBLOB

The following point must be considered while using the text conversion functions (BLOBDBCS, BLOBMBCS, BLOBTEXT, DBCSBLOB, MBCSBLOB, TEXTBLOB) in an Action Diagram:

- If the destination length is less than the length of input BLOB, the output is truncated to length of destination.