CA Gen

Host Encyclopedia Construction User Guide

Release 8.5

Third Edition

CA Technologies Product References

This document references the following CA Technologies products:

- CA Gen
- AllFusion 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.

Documentation Changes

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

- DTF Error Codes (see page 189) - Updated the topic with all the common DTF Abends.
- Intelligent Regeneration (see page 324) - Added the last bullet in the topic.
- Batch Job (see page 184) - Removed duplicate information.

Contents

Chapter 9: Testing Applications 177

Chapter 1: Host Encyclopedia Construction User

About Construction

Construction generates an executable application in a CA Gen model. On the mainframe, or Host, CA Gen supports the construction of the COBOL applications that execute under z/OS accessing DB2 UDB for z/OS databases.

References to DB2 refer to DB2 UDB for z/OS.

You can use Host Encyclopedia Construction to construct online, batch, and server applications. Online applications can execute under IMS, CICS, or TSO. Servers can execute under IMS or CICS.

CA Gen supports Construction with these toolsets:

- Database Construction
- Referential Integrity Trigger Construction
- Application System Construction, source, and executable code

CA Gen z/OS Runtime

CA Gen z/OS runtime is a multiple languages execution environment that includes C, COBOL, and Assembler code. This section summarizes the z/OS Runtime changes since AllFusion Gen 7, and is an extract of the *Release Notes* for those releases.

PDSEs

Since AllFusion Gen 7, runtimes are Program Objects that must reside in a PDSE library, a data set type of LIBRARY. The Host Encyclopedia installation jobs create PDSE data sets for CA Gen CEHBPLD0 and CEHBPLD1 libraries.

Since AllFusion Gen 7, the generated code is Program Objects. When you use the Business System data sets specified for NCAL, Executable, and RI Trigger compiled load modules and compatibility libraries, they must be PDSEs. When you use the DYNAM(DLL) option to create CA Gen user exits, the External System Load Libraries containing these user exit modules must also be PDSEs. When you use the External Action Block and Compatibility External Action Block libraries, they must be PDSEs.

C Runtime DLLs and Code Page Customization

Since AllFusion Gen 7, the C runtimes are in IBM C and LE conformant. Since the applications created with Gen releases earlier than release 7 cannot use these C runtime modules, CA Gen includes new runtimes, TIRCRUNC for CICS and TIRCRUNI for IMS.

To use the TIRCRUNC and TIRCRUNI dynamic runtime modules, requires user modifications to the code page translation routines. CA Gen includes a sample utility, MKCRUN, to facilitate the modifications.

LE Changes

Since AllFusion Gen 7.5, the z/OS runtime fully conforms to IBM LE, and uses the standard LE call interface, reducing runtime code complexity. The runtimes are fully re-entrant and thread-safe, improving reliability and performance for the runtime and the generated applications that they support.

Runtimes exploit LE storage management in the assembler routines, eliminating the need to call GETMAIN and FREEMAIN, reducing CPU usage for runtime modules in certain generated applications, such as batch jobs. The CA Gen C and COBOL runtimes use similar functionality.

Migrating the AllFusion Gen Assembler code to LE functionality decreased the number of OS storage calls and decreases CPU usage when a generated application invokes Assembler runtime functions. Your LE heap and stack settings and how each generated application uses these runtimes, block mode, batch, or distributed processing server, determines the benefit of this change.

Assembler and COBOL Runtime DLLs

Since AllFusion Gen 7.5, C runtime routines are in IBM C and DLLs enable dynamic linking to the other Gen runtimes, COBOL, and Assembler. Most Assembler and the COBOL runtimes are DLLs, and as multiple processes share a single copy of the runtime, significantly decreasing the overall load module size for a Gen application.

AllFusion Gen 7.5 improved serviceability over previous releases by allowing update of the runtime without requiring static linking of the maintenance items into every generated application. Maintenance that is applied to a DLL is available to all generated applications that use the DLL.

TSOAE

Since AllFusion Gen 7.5, the TSOAE environment that the z/OS IT and the Application Test Facility use is a 31-bit, LE-compliant application. The use of 24-bit storage is limited to TSO and I/O functions that must be addressed below the 16-MB line. The TSOAE includes virtual storage constraint relief for testing or implementing large CA Gen applications in TSO and batch.

Remove 32K CFB Limit

For CA Gen Release 8.5, the amount of customer data that can be sent between CA Gen Client and Servers using Common Format Buffer (CFB) based cooperative flows has increased. The previous CFB limit was 32K. The new limit is 16.7 MB. The runtimes were modified to handle the larger CFB size.

Application Migration

There are different application migration requirements depending on the CA release that the application is migrating from and the type of linkage the application uses to invoke the different application components.

Note: For more information about migration, see the *Release Notes.*

More information:

Application Migration in HE (see page 200)

Database Construction Toolset

The Database Construction Toolset builds the Database Definition for the database as defined in the Data Structure list and Data Store list. It generates the JCL and DDL statements that allocate and construct the DB2 database objects. This toolset also generates the JCL to execute several DB2 utilities.

Referential Integrity Trigger Construction Toolset

The Referential Integrity Trigger Construction Toolset supports generating source code and object code modules that maintain the validity of data when deleting an entity or disassociating a relationship. The logic for Referential Integrity is based on the relationships between entity types that are defined in the Data Model.

Application System Construction Toolset

The Application System Construction Toolset supports system generation and installation of online, batch, and server applications. During system generation, CA Gen constructs COBOL programs. During the installation, CA Gen produces executable load modules for local installations or Implementation Packages (remote files)-for remote installations.

Prerequisites

The Construction toolset uses information that is stored in the Host Encyclopedia for a CA Gen model. The Host Encyclopedia is the primary source of information for system generation and database generation. Before Construction begins, each type of generation requires complete information from certain diagrams. The type of generation and the required diagrams for each type are:

Database generation requires:

- Data Store
- Data Structure

System generation requires:

- Data Model
- Dialog Flow Diagram
- Screen Design Diagrams for online applications
- Procedure Action Diagrams
- Data Structure

Referential Integrity generation requires:

- Data Structure
- Data Model

To produce these diagrams, use CA Gen at the workstation.

CA Gen requires consistency for the portion of the model for which it is constructing the application system. The CA Gen Consistency Check feature verifies consistency for the model, or selected objects from the model. The workstation toolset includes a Consistency Check option to use to check the model before uploading it to the Host Encyclopedia.

After uploading the model to the Host Encyclopedia, you can use the host Consistency Check to validate the model. The host and the workstation Consistency Check functions use the same rule base.

Construction Functions

Construction functions, described in detail, includes these primary functions:

- Generate COBOL source
- Generate Databases
- Generate Referential Integrity modules
- Package Load Modules
- Specify Target Environment and Construction Libraries
- Generate the Application System
- Test the Application System

Generate Databases

The host database generation tools create the data definition language, (DDL) statements that allocate and construct the DB2 objects for a model. CA Gen uses the completed Data Structure, which is stored in the Host Encyclopedia, to generate the DDL and the JCL to execute the DDL.

For local DB2 applications, use the Generate JCL for DB2 utilities option to generate JCL to run several DB2 utilities.

More information:

Database Generation in HE (see page 37)
DB2 Utilities (see page 233)

Generate Referential Integrity Trigger Modules

The separate libraries store the source code, object code, DBMS, and control information in action blocks named Referential Integrity (RI) triggers. Specify the target environment and construction libraries for the RI trigger modules separately from the environment libraries of the application.

More information:

Referential Integrity (see page 49)

Package Load Modules

The Host provides load module packaging for online and batch load modules, but not for client/server applications. Perform server load module packaging on the workstation and upload it to the Host.

Load module packaging defines procedure steps grouping for installation. The packaging identifies the procedure steps that are contained in each load module. For online and server modules, it identifies the transaction codes that invoke each procedure step. For batch modules, packaging identifies the job and job step that execute the procedure step and the DB2 attach facility that is used. The name that is assigned to the load module is also the DB2 Plan name.

Code generation, not load module packaging, creates the load modules. Define the packaging before using CA Gen to generate the load module or its components. The system generation functions use the packaging definition to determine the components to include in the generated load module. The model stores and updates the packaging definition for each load module.

More information:

Load Module Packaging in HE (see page 107)

Specify Target Environment and Construction Libraries

CA Gen requires that you specify the Construction libraries before it can generate application system components. The Construction libraries store the objects that are created during system generation. The libraries also contain some of the objects that are used during Construction that are external to CA Gen. For example, they can contain external action blocks, programs that are created outside CA Gen.

Also specify the target environment where the generated application runs.

More information:

Specify Target Environment for HE (see page 77)

Generate the Application System

The system generation tools support the following functions:

- Run Consistency Check
- List Action Diagram
- Generate Load Module
  - Generate the COBOL source
  - Generate Dialog Manager or Server Manager
  - Generate Batch Manager for DB2
  - Generate Screen Definitions
  - Generate JCL to Execute Batch Applications
  - Generate Installation Control Information
  - Compile Generated Components (local)
  - Link-edit Compiled Source (local)
  - Bind for DB2 applications (local)
- Generate the Remote Installation files
- Generate a Load Module Component
- Rename Source Members
- Generate an External Action Block Stub
- Resolve an External Action Block
- Install a Load Module (local or remote)
- Test the Application System for DB2

Note: The terms COBOL and COBOL for z/OS refer to the COBOL version in the *CA Gen Technical Requirements* document for this release.

Run Consistency Check

You can execute a Consistency Check from the Construction Tools. The host Consistency Check feature uses the same rule base as the workstation Consistency Check.

List Action Diagram

You can view an Action Diagram (AD) for a selected procedure step or action block on the host without returning to the workstation tools.

More information:

System Generation and Installation (see page 117)

Generate Remote Installation Files

CA Gen can generate files for a remote z/OS installation, during which CA Gen can create a remote file for each of the following:

- Each load module
- All Referential Integrity Triggers
- Database

If multiple load modules exist in the business system, CA Gen generates a remote file for each one.

More information:

System Generation and Installation (see page 117)

Generate a Load Module Component

Using the CA Gen Construction Tools, you can generate a single component of a load module. This is useful to generate only the parts of the load module that is affected by changes to the model since it was last generated.

More information:

System Generation and Installation (see page 117)

Rename Source Members

You can use certain panels in the CA Gen Construction Tools to name or rename generated source code member names, and to name external action blocks and referential integrity triggers.

More information:

System Generation and Installation (see page 117)

Generate an External Action Block Stub

When your application uses logic that is created outside CA Gen, such as external subroutines, generate an external action block stub to provide an interface between the external subroutine and CA Gen generated code.

More information:

External Action Blocks (see page 207)

Resolve an External Action Block

After generating an external action block stub, modify it to access your external subroutine. When DB2 is the target database, and the external subroutine uses embedded SQL, resolve the external action block by identifying the Database Request Modules (DBRMs).

More information:

External Action Blocks (see page 207)

Local and Remote Load Module Installation

CA Gen supports local and remote load module installation.

For a local installation, installing the load module is the final step in system generation. CA Gen compiles the uncompiled generated components, performs a preliminary NCAL (nocall) link edit, and edits the executable load module, resolving all external references.

If the target DB2 subsystem is on the same CPU as CA Gen or is defined as a remote DB2 subsystem, CA Gen also binds the application. If necessary, you can customize CLIST TICINSTX to meet site-specific application bind requirements.

Note: For more information about this CLIST and other customization features, see the *z/OS Installation Guide*.

For DB2, you can use Package Bind. Using package bind lets you make changes in a program within an application and rebind only the affected DBRM.

For a remote installation, CA Gen prepares a remote file with all the generated components the Implementation Toolset to install the application on the remote platform.

Note: For more information about using the Implementation Toolset, see *z/OS Implementation Toolset User Guide*.

More information:

System Generation and Installation (see page 117)
DB2 Package Bind Options (see page 65)

Test the Application System

When the target database is DB2, the Application Test Facility lets you test new systems or changes to systems in a TSO testing environment. This runtime environment simulates your target production environment. The testing facility includes a trace function to debug CA Gen generated programs at the model level. It lets you debug the action diagram logic rather than the COBOL code generated from them.

Online Help

Online help is available as you perform Construction functions:

- When an error occurs, the system displays a short message in the upper-right corner of the screen and beep to alert you.
- To see more information about an error message, or to see an explanation of a panel, press the Help key or type Help on the COMMAND line and press Enter. F1 is the default Help key.

Customizing CA Gen Applications

You customize a CA Gen generated application to use code that is created outside CA Gen. Use these features to customize an application:

- External Action Blocks
- Exit Routines
- Reporting

External Action Blocks

External action blocks let an application use logic defined outside of CA Gen. For example, they let an application use existing subroutines or access non-DB2 databases.

More information:

External Action Blocks (see page 207)

User Exits

User exit routines allow customization of applications for error handling, help processing, security, and other system functions.

Note: For more information, see the *User Exit Reference Guide*.

Reporting

The Construction Toolset does not support report generation from CA Gen generated application code. For the report generation, use your own subroutines, external action blocks, or other report writing software such as QMF.

CA Gen Main Menu

The CA Gen Main Menu lists the host functions:

- Host Encyclopedia functions
- Public Interface functions
- Host Encyclopedia reports
- Application system functions
- Environment specifications

Host Encyclopedia Functions include functions such as user access, model management, and subset management that let you add, change, and delete data in the Host Encyclopedia. They also let you monitor user access to CA Gen and to the contents of models and subsets.

Public Interface Functions include model export and import functions. The export function stores model data in separate tables, for access by PI views. Using the PI views, you can create reports that are tailored to your needs. The import model function creates a model from files that are created outside of CA Gen, typically from data that is contained in some other data management product.

Application System Functions is the name for the Construction functions used to generate and install application systems. The Application system functions are described in this guide. All other CA Gen host capabilities are described in other CA Gen Encyclopedia guides.

Environment specification lets you specify JCL for CA Gen functions that execute in a batch environment, such as, copy model, generate DDL, and generate source code. You can also select CA Gen panel colors for your environment and sort options for selection lists using this option.

Open the Main Menu

Follow these steps:

1. Log on to TSO/ISPF.

   The ISPF Main Menu displays.

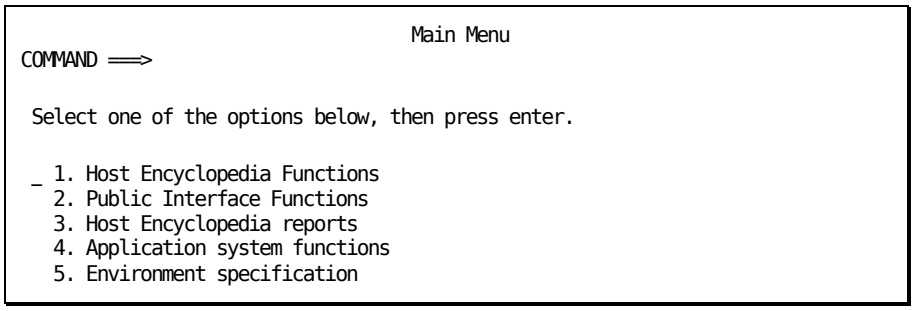
   If your installation accesses the menu some other way, contact your local CA Gen coordinator for instructions.

2. Select CA Gen and press Enter.

   The copyright screen displays.

3. Press Enter.

   The CA Gen Main Menu displays.



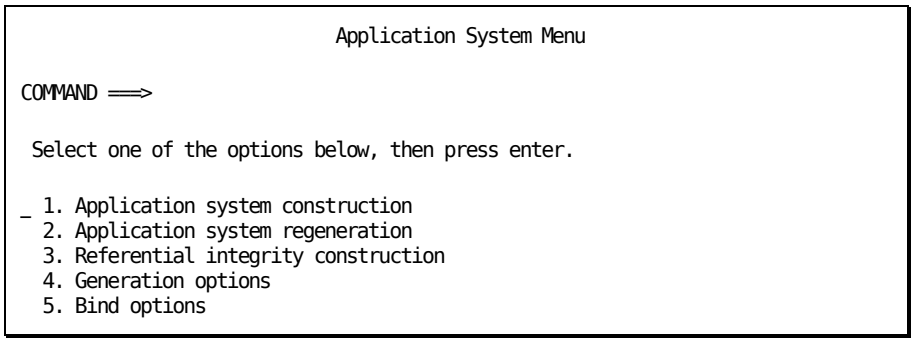
Open the Application System Menu

The Application System menu lists the Host Encyclopedia Construction functions.

Follow these steps:

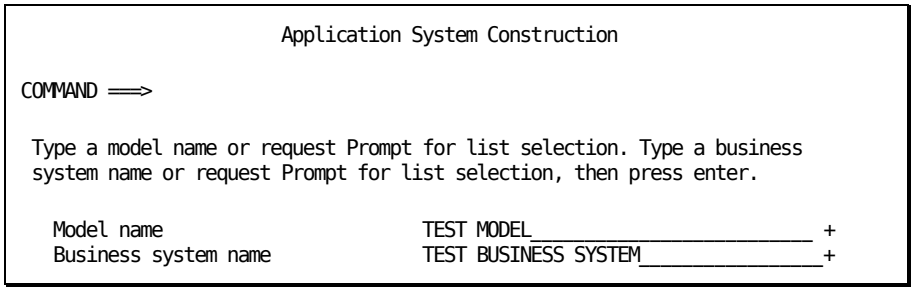
1. Open the CA Gen Main Menu as defined in "Open the Main Menu".
2. Type 4 for Application system functions and press Enter.

   The Application System menu displays.



3. Press 1 to select Application system construction and press Enter.

   The Application System Construction panel to specify the model and business system displays.



4. Type the Host Encyclopedia model name.

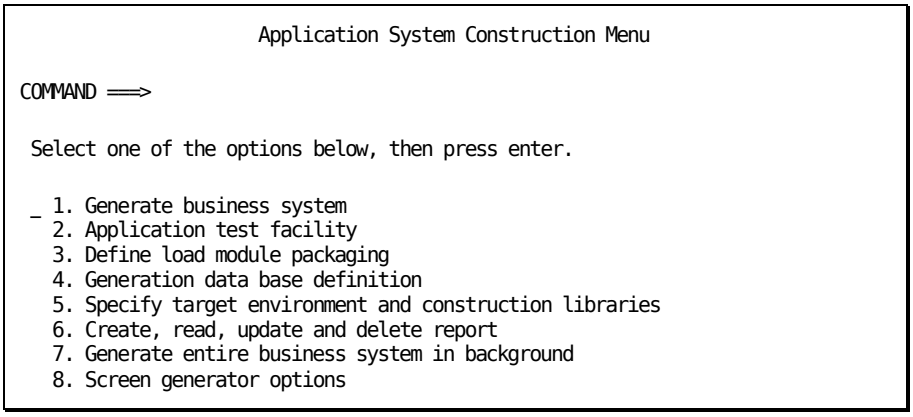
   To list model name choices, use the Tab key to position the cursor at the field and press the Prompt key (PF4). Select a model in the list by typing any character except a period (.) or a space next to the name and press Enter.

5. Type the Host Encyclopedia business system name.

   To list business system name choices, use the Tab key to position the cursor at the field and press the Prompt key. Select a business system from the list by typing any character except a period (.) or a space next to a name and press Enter.

6. Press Enter to accept the model and business system names.

   The Application System Construction Menu panel displays.



CA Gen automatically retrieves the construction library and target environment specifications that are stored in the Host Encyclopedia for the business system.

More information:

System Generation and Installation (see page 117)

Selection Lists in HE

When you select a Construction function, a panel prompts for the model and business system names.

CA Gen lists model and business systems when you press the Prompt key, PF4 by default. CA Gen supports multiple ways to specify the model and business system names:

- Type the model and business system names.
- Select the model name from the Model Selection List and the business system name from the Business System Selection List.
- Type one name and retrieve the one name from the list.

Model Selection List

The Model Selection List screen lists models to which you have access. The upper-right corner of the panel displays the row number of the model at the top of the list and the total number of models in the list.

```
                                           Model Selection List                                     Row 1 of 4
COMMAND ===>                                                                                  SCROLL ===> PAGE

 Type / next to one model name, then press enter.

                                                            Model         ------------Modified---------         Checked
      Model name                                       Owner              Date           Time          User         Out    To
.  TEST MODEL 1                                    User ID       YYYY-MM-DD     HH:MM      USER ID     USER  ID
.  TEST MODEL 2                                    User ID       YYYY-MM-DD     HH:MM      USER ID     USER  ID
.  TEST MODEL 3                                    User ID       YYYY-MM-DD     HH:MM      USER ID     USER  ID
.  TEST MODEL 4                                    User ID       YYYY-MM-DD     HH:MM      USER ID     USER  ID
**************************************Bottom of data *********************************************************
```

The list shows the following information for each model:

Model Name

Displays the Host Encyclopedia model name.

Model Owner

Displays the model administrator's TSO user ID. The model administrator's ID was intentionally removed from this illustration.

Date

Displays the date that the model was last modified.

Time

Displays the time that the model was last modified.

User

Displays the TSO user ID of the person who last modified the model. The ID for the last user that modified these models was intentionally removed from this illustration.

Checked Out To

Displays the TSO user ID of the person who has the model that is checked out. If the model is not checked out, this field is blank. If a subset is checked out, this field contains asterisks.

Business System Selection List

The Business System Selection List lists business systems that are defined for the selected model. The upper-right corner of the panel displays the row number of the business system at the top of the list and the total number of business systems in the list. For example, Row 1 of 4 indicates that the business system at the top of the list is the first of four defined for the model.

```
                    Business System Selection List        Row 1 of 4

COMMAND ===>

 Type / next to one business system name, then press enter.
        Business System name        Creator     Date
. Management_I                      User ID  YYYY/MM/DD
. Management_II                     User ID  YYYY/MM/DD
. Management_III                    User ID  YYYY/MM/DD
. Management_IV                     User ID  YYYY/MM/DD
```

For each business system, the list shows the TSO user ID of the person who initially checked in or uploaded the business system and the date it was first checked in.

To select a business system, type / (slash) in the column next to the business system name and press Enter.

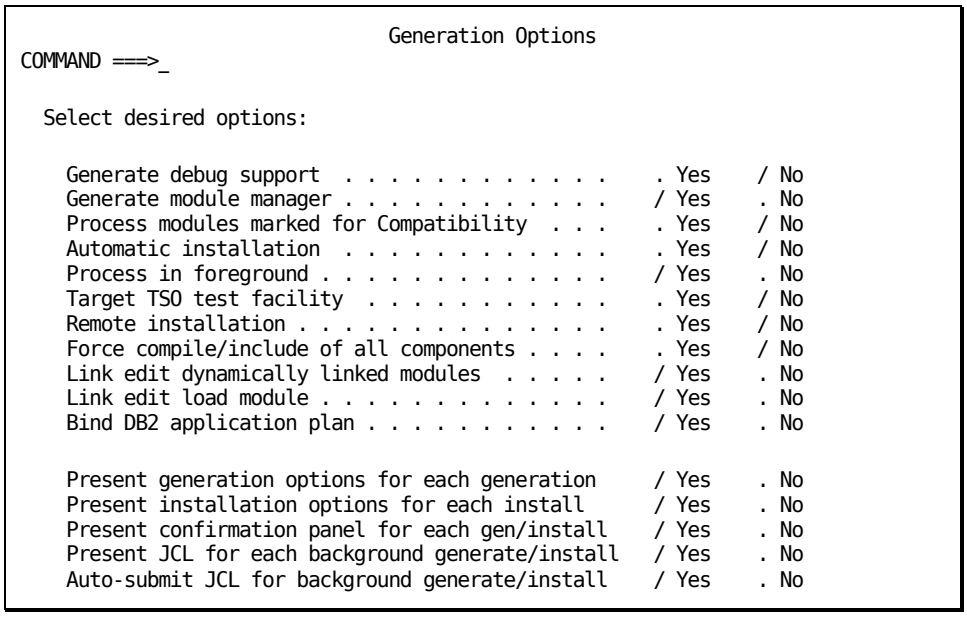
Note: Some panels that request the model and business system name let you set parameters that change functionality. If you access the Business System Selection List from a panel in which you specified the business systems to use as ALL, it does not display a selection list.

Chapter 2: Generation Options

Generation Options Panel in HE

The Generation Options panel displays user-customizable values for generation options. Each user has a separate set of options. Storing the values standardizes and expedites generation and installation. All generation and installation processes, Applications System Construction, Application System Regeneration (Intelligent Regeneration), and Referential Integrity Trigger Construction processes use these options.

To open the Generation Options panel, type 4 in the Application system menu screen.



The first group of settings lists the options for generation and installation. CA Gen presents these options in the generation dialog, allowing you to override each option for a procedure. To use the defaults, use the dialog options to suppress presentation of the options.

The dialog options, the second group of settings, that control the panels that are displayed when you initiate a generate or install process. You can only set these options in this panel. Answering No to a dialog option suppresses display of a specific panel.

Options

The options and their descriptions are:

Generate debug support

Specifies to generate action diagram code with the trace function. Options are:

- Yes - include the trace function
- No - omit the trace function

Use the Application Test Facility to debug.

Generate module manager

Specifies to generate the dialog, batch, or server manager for the load module, or only for the selected components. Options are:

- Yes - generate the dialog, batch, or server manager for the load module
- No - generate the dialog, batch, or server manager for only the selected components

Process modules marked for Compatibility

Specifies the modules to process. Options are:

- Yes - process modules packaged as compatibility and static modules included in the compatibility module during generation or installation. When you select this option when processing RI triggers or Compatibility modules that include statically linked action blocks, the generator compiles the RI triggers and Action Blocks twice, with NODLL and DLL compile options.
- No - bypass compatibility modules during generation and installation

Automatic installation

Specifies to perform the install process automatically after generation completes. Options are:

- Yes - automatically install when generation completes.
- No - generate without the installation.

Process in foreground

Specifies to execute in the foreground or background. Options are:

- Yes - execute the generate and install process in the foreground.
- No - run the process in the background as a batch job.

Target TSO test facility

Specifies to target the TSO Test Facility. Options are:

- Yes - override the Target Environment parameters for the install process and target the TSO Test Facility
- No - use the Target Environment parameters.

The generator ignores this option for server load modules installation. Testing servers under the TSO test facility is not supported.

Remote installation

Specifies if the target installation is a remote system. Options are:

- Yes - when your target environment is a z/OS system different than the system generating the code.
- No - when the target environment is the local system.

Force compile/include of all components

Specifies the components to generate. Options are:

- Yes - compile load module components for local installs, or include source for all components in the remote file for remote installs.
- No - compile only the components that are generated since the last load module installation.

The installation compiles and includes components with a dynamically link packaging property set to compatibility only when you select the Process modules marked for Compatibility option.

Link edit dynamically linked modules

Specifies the modules to link edit. Options are:

- Yes - link-edit modules that are specified as dynamically linked that the load module references.
- No - omits dynamically linked modules that are referenced by the load module, unless they are compiled.

The installation links components with a dynamically link packaging property set to compatibility only when you select the Process modules marked for Compatibility option.

Link edit load module

Specifies to link edit the load module. Options are:

- Yes - link edit the load module and resolve all external references to create an executable load module.
- No - bypass link-editing the load module.

Bind DB2 application plan

Specifies to include the DB2 application plan when performing the bind. Options are:

- Yes - bind the DB2 application plan for the load module.
- No - omit the DB2 application plan bind.

Present generation options for each generation

Set to Yes to review and override stored options for the current activity in an abbreviated version of this panel before each generation. Change the full Generation Options panel to change the defaults.

Present installation options for each installation

Set to Yes to review and override stored installation options each time you install. CA Gen presents the Installation Options panel before each installation activity. Change the full Generation Options panel to change the defaults.

Present confirmation panel for each generation/installation

Set to Yes to present a Confirm Generation or Confirm Installation panel before each generation or installation process.

Set to No to suppress the confirmation panel.

Present JCL for each background generation/installation

Specifies to display the JCL JOB card. Options are:

- Yes - review the card for possible modifications. CA Gen displays the Job card information in the Update JCL for Background Code Generation panel.
- No - use stored job card.

Auto-submit JCL for background generation/installation

Specifies to automatically submit the JCL. Options are:

- Yes - automatically submit the batch job. The generated JCL does not appear in ISPF Edit mode and you do not have to manually submit the job.
- No - display the JCL in Edit mode and manually submit it.

More information:

Update JCL for Background Code Generation Panel (see page 36)
Confirm Generation and Installation Panels (see page 36)

Process Modules Marked for Compatibility

This option applies to MVS applications with components configured with the dynamically link packaging property set to Compatibility. Compatibility is intended for use by applications that dynamically call modules that must reside in a non-DLL module.

Important! All modules within an application are subject to the support policy of the CA Gen release in which they are produced. Using the Compatibility feature without selecting the Process modules marked for Compatibility option does not change the support policy or support timeframe that is associated with those modules that are marked for Compatibility. Therefore, modules that are marked for Compatibility and their components are supported at the release level at which they are built.

Components that are marked for Compatibility include:

- Routines that are built for dynamic linking using a release of Gen earlier than Release 7
- External Action Blocks (EABs) that contain a dynamic program call to a non-Gen routine that resides in a non-DLL module
- Routines that are the target of a dynamic program call that is initiated from a routine that resides in a non-DLL module

Use the Process modules marked for Compatibility option to generate or install modules that are marked for Compatibility. Failing to set the option bypasses processing of components that are marked for Compatibility.

Selecting the Process modules marked for Compatibility option generates and precompiles the RI Trigger modules and all Action Blocks statically named by the module marked for Compatibility, and compiles them twice, once using the NODLL compiler option and again using the DLL compiler option.

The result of the NODLL compile is linked into the module marked for Compatibility. The result of the DLL compile is provided so that it can link into any DLL applications that statically call these triggers or action blocks. If the action blocks are External Action Blocks, they must be compiled using the appropriate compiler options and is available to include in the final load module install.

Separate the libraries in the target environment hold the separate NCAL modules resulting from the two compile steps.

Confirm Generation and Installation Panels

The Confirm Generation Options and Confirm Installation Options panels display when you initiate a generation or installation process. The panels display the options to use for generation and installation and ask you to confirm the operation by pressing Enter.

To cancel the process, press End, press Cancel, or type Cancel on the COMMAND line and press Enter. When generating in the foreground, this is the only opportunity to cancel the process. After execution begins, you cannot cancel it.

Update JCL for Background Code Generation Panel

This panel displays when you select background processing to verify and modify Job card information for the current job submission only, without updating the Host Encyclopedia.

Press Enter to submit the job when the auto-submit option is set to Yes. When the auto-submit option set to No, CA Gen displays the generated JCL in ISPF Edit. Manually submit the job or save it for later submission.

Chapter 3: Database Generation in HE

Creating Databases

Database generation creates the databases that implement a data model from diagrams that are completed in the workstation toolsets. The host tool uses the completed Data Structure List (DSL) and Data Store List (DOL) to generate the data definition language (DDL) statements and JCL to create the database objects.

The Data Store List and Data Structure List are CA Gen workstation tools, which are based on the model's Data Model, that define databases. The Data Model is a conceptual representation of the information in the application. The Data Store List and Data Structure List are a physical representation of the information.

Transform the Data Model on the workstation to create the data structure. This provides a suggested database design that the Database Administrator (DBA) modifies.

Transformation uses a rule-based approach to examine the Data Model objects and determine appropriate implementation. CA Gen generates the objects, associations, and properties. Use the data structure tools to refine the lists the transformation process generates.

The database generation tool also generates the JCL for DB2 utilities. The utilities help maintain database integrity through backup and restore, and monitor and tune your DB2 databases. The tool supports the COPY, RECOVER, REORG, and RUNSTATS utilities.

Generation Steps

The following procedure shows a typical activity sequence to generate a database definition for a model:

Follow these steps:

1. Generate the JCL and DDL for the entire database system or the required database objects.
2. Review and edit the generated DDL.
3. Save the job if you do not intend to submit it immediately. CA Gen does not save the job for you.
4. Use the SUBMIT command to submit the job that creates the database objects.

CA Gen allows you to generate any or all of the subordinates from the object level.

The DB2 database objects and the subordinates at each level are:

Database

Table Spaces, Tables, Index Spaces, Indexes

Table Space

Tables, Index Spaces, Indexes

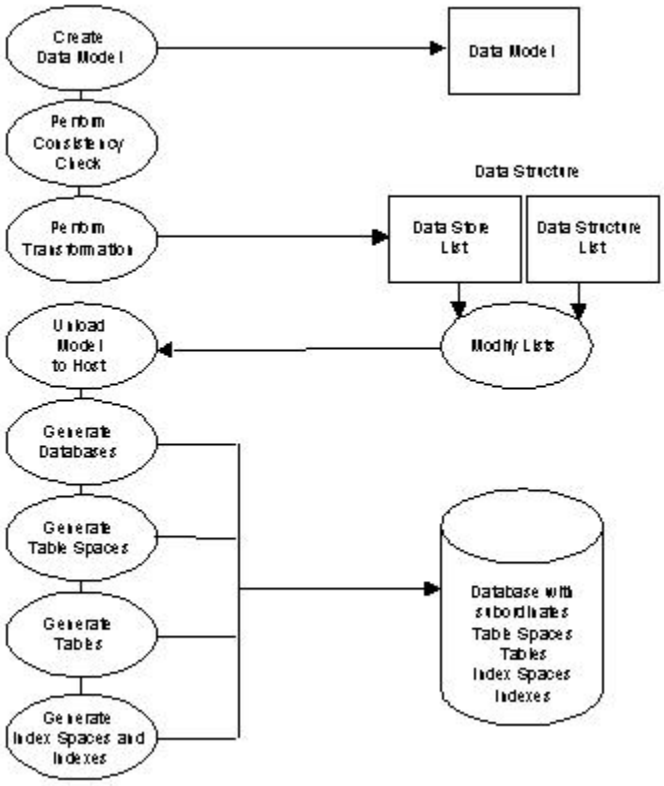
Table

Index Spaces, Indexes

Index Space and Index

No subordinates

Complete the Data Store List and Data Structure List before creating the database because Database generation uses these lists as its primary source of information about the databases.



Prerequisites for Database Generation

Database generation uses the physical definitions from the Data Structure and the Data Store. Complete this information before uploading the model to the host and generating the database.

- Complete the Data Model
- Perform a Consistency Check
- Perform Transformation, or Retransformation if the model has been transformed once
- Modify the Data Store and Data Structure as necessary
- Upload the model to the Host Encyclopedia

Complete the Data Model

The Data Store List and the Data Structure List, which is stored in the Host Encyclopedia, are the primary sources of information for database generation and must be complete and consistent to successfully generate the DDL for a model. Since the Data Store List and the Data Structure List are based on the model's Data Model, the Data Model must also be consistent.

To only generate part of the database definition, for example, a table and its indexes, the portion of the Data Model, the Data Store List, and the Data Structure List that describe the objects to define must be consistent.

Perform a Consistency Check

The Consistency Check verifies that a model, or a part of a model, is consistent. Tools in the workstation toolset allow you to check the Data Model and the Data Structure for consistency. Use these toolsets to check the diagrams before uploading the model to the Host Encyclopedia. The Data Model must be consistent before you can transform the Data Model to create a Data Structure.

Note: After uploading the model to the Host Encyclopedia, validate the model with the host Consistency Check.

Perform Transformation and Retransformation

You can transform the complete Data Model or a portion of it. Use the Diagram and Open options to control the data structure definition that is created during the Transformation Process to select the DB2 z/OS Technical Design defaults. In general, the TD defaults do not need adjusting. Review them with a Database Administrator (DBA) before implementation on the host.

You cannot use some words as ordinary identifiers in a context in which they could be interpreted as SQL keywords. For example, you cannot use COUNT as a column name in a SELECT statement. You can use words as an ordinary identifier in other context. In the statements where the word can never be an SQL keyword, you can use the word as a delimited identifier in context. For example, when double quotation marks (") begin and end delimited identifiers, as in "COUNT". You can use "COUNT" as a column name in a SELECT statement. For a complete list of Reserved Words, see the IBM documentation for DB2.

During Transformation, CA Gen attempts to use the TD names that are specified in the Data Model for data definition components. When CA Gen detects a name that violates the DBMS reserved word conventions and Reserved Word Support is set to YES, CA Gen automatically changes the name to a non-reserved word.

Plan a partial Transformation carefully to avoid problems during system generation. For example, transform all entities that are referenced by the business system you intend to generate. It is possible to forget to transform entities that are not directly referenced by the action diagrams for the business system, but referenced by the generated code.

As an example, when implementing referential integrity, the generated code includes referential integrity logic for mandatory relationships. The logic reference entities that are not directly referenced by the action diagrams. These entities must be transformed for successful system generation.

Similar problems occur if you change the Data Model after transformation. The impact of changes to the Data Model is not always obvious, and a partial retransformation cause inconsistencies. Perform a consistency check on the Data Store List and Data Structure List to detect these inconsistencies after a partial transformation or retransformation. The When Changed Report can help in planning a partial transformation.

Note: For more information, see the *Host Encyclopedia User Guide*.

For DB2 applications, identify the DB2 subsystem in which the generated application runs before creating the DDL.

After Transformation, when you change the Data Model, make an associated TD change to maintain consistency. Keep the TD synchronized with the Data Model by running transformation. When you run transformation, it replaces the initial database design, including changes that are made and creates a database design. It deletes a relationship triggers and adds new ones.

To leave the previous database design triggers intact and change only what is inconsistent or incomplete, use the Retransformation option.

More information:

Specify Target Environment for HE (see page 77)

Upload Model to the Host

To upload the model to the Host Encyclopedia, use the workstation communication tools.

Note: For more information, see the online help in the workstation toolset.

Allocating Space for Table Spaces and Indexes

To acquire space for DB2 table spaces and indexes, use IDCAMS to allocate a VSAM data set for each table space or index, or reference a storage group when creating the table space or index. A storage group identifies one or more DASD volumes available for DB2 table spaces and indexes.

The Data Store List documents how to decide to use storage groups or IDCAMS to acquire DB2 space. When you use storage groups, CA Gen generates the JCL to run the appropriate DDL. When you use IDCAMS, CA Gen generates the JCL to run the IDCAMS control statements and DDL.

Database Authority

CA Gen generates JCL based on choices in the panels that are described in the Database Definition Using DB2 sections. To run the JCL successfully, requires DB2 authority for the activity. For DB2, the reference guide for DB2 SQL describes the authority that is required to execute the DDL statements.

Database Definition

Database definition describes the physical storage of the database objects that are identified by the database design. CA Gen creates the database definition that is based on the information in the completed Data Store List and the Data Structure List stored in the Host Encyclopedia.

For DB2 users that omit the Storage Group option, CA Gen generates IDCAMS control statements to define the VSAM storage allocation for table spaces and index spaces. CA Gen also generates the JCL to execute the SQL DDL statements and IDCAMS control statements.

CA Gen uses the DBMS selected on the Target Environment panel to implement the databases, and uses SQL as the data definition language (DDL) to define the databases.

CA Gen allows you to select the objects for which to generate DDL:

- databases
- table spaces, with DB2 only
- tables
- index spaces, with DB2 only
- indexes

You can generate the JCL and DDL for the entire database system, or for selected objects, as needed.

Use the ISPF Edit commands to review and make changes to the generated JCL and DDL. Use the SUBMIT command to submit the job that allocates the data sets and creates the database objects.

CA Gen does not automatically save the generated JCL and DDL. When you plan to submit the job later, save it. After the job executes successfully, delete it. If you regenerate the DDL for an object, CA Gen creates a job.

To open the database generation panels, select Option 4, Generate data base definition on the Application System Construction Menu.

As it creates the database definition, CA Gen moves through a series of panels. The section Database Definition Using DB2 describes the panels when your system uses a DB2 DBMS.

Database Definition Using DB2

When your system uses DB2, and you select Generate data base definition in the Application System Construction Menu, CA Gen opens this Data Base List panel.

Although this panel identifies the business system, the database function applies to all business systems in the model.

The Data Base List panel supports these action codes:

U

Opens the DB2 Utilities panel to create JCL for DB2 utilities. Use the DB2 Utilities window to generate the JCL to execute one of these DB2 Utilities:

COPY

Creates an image of the selected table spaces in the database.

RECOVER

Restores the previous version of selected table spaces and indexes in the database.

REORG

Reorganizes selected table spaces or indexes to improve performance.

RUNSTATS

Updates the information DB2 uses to select index usage and improve performance.

G

Type G next to an object in the Data Base List panel to open the JCL Job Statement Information Input panel.

A

Type A as the action code next to an object Data Base List panel to open the DDL Options panel. Use the DDL Options panel to generate all the DDL and JCL to create the storage group and the database and its subordinate objects, including tables, table spaces, indexes, and index spaces.

After typing Y or N for each option, press Enter. The system opens the Generated DDL Text panel with the DDL the system generated based on your choices.

Type Go and press Enter to open the JCL Job Statement Information Input panel.

E

Type E as the action code next to a database name in the Data Base List panel to expand to table spaces. CA Gen opens the Table Space List panel.

The Table Space List panel supports these action codes:

G

Type G as the action code next to a table space name to open the JCL Job Statement Information Input panel.

A

Type A as the action code next to a table space name to open the DDL Options panel to generate DDL and JCL for the table space and its subordinate objects, including indexes and index spaces.

E

Type E as the action code next to a table space name to expand the Table List to a panel.

More information:

Set Table List Options (see page 44)
JCL Job Statement Information Input (see page 45)
Set DDL Options for a DB2 Database (see page 45)

Set Table List Options

The Table List panel expands the table space and allows you to select options to generate DDL and JCL for the table, and for the table and all its subordinate indexes and index spaces.

The Table List panel supports these action codes:

G

Type G as the action code next to a table name to open the JCL Job Statement Information Input panel.

A

Type A as the action code next to a table name to open the DDL Options panel to generate DDL and JCL for the table.

E

Type E as the action code next to an object to open the Index List panel to expand the index spaces.

The Index List panel supports G as an action code. Type G next to an Index or Index Space to open the DDL Options panel to generate DDL and JCL for the index.

More information:

JCL Job Statement Information Input (see page 45)
Set DDL Options for a DB2 Database (see page 45)

Set DDL Options for a DB2 Database

To generate the DDL and JCL to create the database, tables, and indexes; use the DDL Options panel.

After typing Y or N for each option, press Enter. The system opens the JCL Job Statement Information Input panel.

More information:

JCL Job Statement Information Input (see page 45)

JCL Job Statement Information Input

The JCL Job Statement Information Input panel displays the information that is collected during the database definition process. Use this panel to specify job card statement information before submitting the JCL and DDL for execution. If you omit the job statement information, CA Gen generates JCL, but cannot execute the job until you add the information. CA Gen does not save the job. To submit the job at another time, save the job.

To change the job statement information:

- Use the database generation dialog panels
- Use the Environment option on CA Gen Main Menu

Both options specify and update the same set of JCL.

Your job statement information meet your organization's requirements and standards. The time specifications vary based on the number of database objects being created.

Edit Generated JCL and DDL

When JCL and DDL generation for the selected objects completes, CA Gen automatically displays the JCL in an ISPF Edit session to review and edit.

For example, if the DB2 subsystem in which the generated application executes is not identified when you specify the target environment, the generated JCL does not include the DB2 subsystem ID. The DB2 subsystem ID must be added to the JCL before it is submitted.

The following example shows part of an EDIT session display of the DB2 JCL and DDL generated to create a table space, a table, and a unique index.

```
EDIT ---- USERID.XXCTEMP1.TIDDLJCL -------------------COLUMNS 001
COMMAND ===>                                          SCROLL ===> PAGE
000049 //*****************************************************************
000050 //*        JCL TO EXECUTE DDL SQL STATEMENTS                *
000051 //*****************************************************************
000052 //*
000053 //STEPSQL EXEC PGM=IKJEFT01,DYNAMNBR=30,COND=(0,NE)
000054 //SYSPRINT DD  SYSOUT=*
000055 //SYSTSPRT DD   SYSOUT=*
000056 //SYSIN  DD  *
000057      CREATE TABLESPACE      "T0041531" IN      "D0041483"
000058        BUFFERPOOL BP0
000059        LOCKSIZE PAGE
000060        CLOSE     NO
000061        USING STOGROUP SYSDEFLT
000062        ;
000063       COMMIT;
000064       CREATE TABLE       "WORK_AREA"
000065         (
000066         "OPTION"                 CHAR(1)
000067          NOT NULL
000068         "ERROR_COUNT"         INTEGER
000069         )
000070        IN  "D0041483".  "T0041531";
000071       COMMIT;
000072        CREATE UNIQUE INDEX    "I0041653"
000073         ON     "WORK_AREA"
000074         ("OPTION"                   ASC
000075         )
000076         SUBPAGES    16
000077         BUFFERPOOL BP0
000078         CLOSE   NO
000079         USING STOGROUP SYSDEFLT
000080         ;
000081        COMMIT;
```

```
000082 /*
000083 //SYSTSIN  DD   *
000084  DSN SYSTEM(DB2T)
000085   RUN PROGRAM(TIUUSQLX) PLAN(GENUSQL) +
000086    LIB(AAAC.GEN.LOAD')
000087   END
000088 /*
000089 //
****************************** BOTTOM OF DATA ******************
```

CA Gen generates the IDCAMS control statements when you do not use storage groups.

Submit Generated JCL and DDL

If your environment allows and you have authority, submit the job to create the database objects. Database security requires that all users have authority, explicitly or implicitly, to do everything they do. For example, to create a DB2 object, a table, table space, or index, requires the authority to create that type of object. Usually, the Database Administrator (DBA) is responsible for granting and revoking DB2 authority. Consult your DBA before creating the database objects.

Note: If you do not intend to submit a job immediately, save it to submit later. CA Gen does not save the job for you.

To run the job that builds the DB2 database, requires DB2 CREATEDBA or SYSADM authority. Many installations restrict the use of these authorities. As an alternative, your DB2 Administrator can define the database for you and can grant you DBADM authority to the database. This allows you to create all of the table spaces, tables, and indexes in the database.

The authorization ID of the person who creates the database objects is the Creator ID. The creator is the owner of the objects. Select authorization ID for object creation carefully considering the following questions:

- Is there a need for a transferable Creator ID?
- Who will bind the generated programs?
- Does that authorization ID have authority over other tables that you also need to access?

Know the Creator ID to execute DB2 utilities on the database objects.

Note: For more information about DB2 security, see the IBM DB2 documentation.

When you are satisfied with the content, submit the job to run as a batch process in the background. You can view the results outside of the encyclopedia.

When the job completes, the encyclopedia returns to the Data Base List panel.

Chapter 4: Referential Integrity

Relationship Participation

When a relationship exists between two entity types, each entity references the other entity through a primary key. Each entity stores the other entity's primary key in the foreign key field in each entity's record.

Referential Integrity (RI) rules verify that when pairing is required, a reference exists, and that the reference is removed when the record to which it points is removed. Rules in the Data Model require setting the reference to null or deleting a record that that has no reference.

CA Gen implements RI triggers as generated action blocks that enforce the referencing rules the Data Model defines when an entity is deleted or a relationship is disassociated.

CA Gen generates and maintains RI triggers for each model separately, based on the complete Data Model and Data Structure for a model.

Generate and install RI triggers before application system construction. This avoids unresolved references when linking application load modules, unless dynamic RI linking is used.

Prerequisites for Generating RI Triggers

To generate RI triggers requires completing tasks on the workstation and the host.

Before generating RI triggers, complete these tasks on the workstation:

- Successful Transformation of the Data Model
- Successful completion of a TD Consistency Check to ensure consistency in the Data Store List and Data Structure List
- Uploading the model to the Host Encyclopedia

Before generating RI triggers, complete these tasks on the host:

- Create and allocate libraries on the host
- Specify the target environment, construction libraries, and DB2 Package Bind defaults for the RI triggers

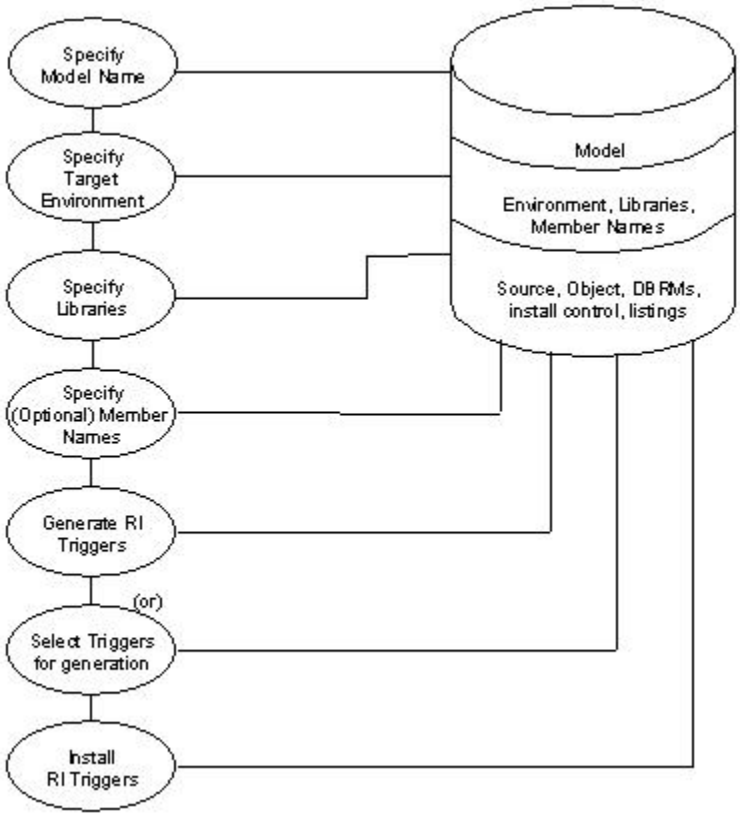
Generating Triggers

To generate triggers, use the Host Encyclopedia Construction Tools.

Follow these steps:

1. Select the model name.
2. Specify the target environment's operating system, DBMS, and language.
3. Specify Referential Integrity Trigger target environment operating system, generated source language, database management system, and DB2 subsystem for package binds.
4. Set the optional DB2 Bind Package defaults for Referential Integrity modules, or the model, or individual members.
5. Specify the libraries for the source code, load modules, Database Request Modules (DBRMs) for DB2, installation control, and listings (optional).
6. Set Dynamically Link RI Triggers flag. When you set this flag to Yes, verify that the RI TP Monitor and RI Trigger Name are set correctly.
7. Generate all Referential Integrity triggers or selectively generate RI triggers.
8. Install, that is compile and link edit, the generated RI triggers.

The following graphic illustrates the trigger generation process.



Types of Referential Integrity Triggers

Generation of RI trigger action blocks results in these two types of modules:

- ENTITY trigger modules (DELETE)
- RELATIONSHIP trigger modules (DISASSOCIATE)

ENTITY Trigger Modules

ENTITY trigger modules delete rows, enforce restrict logic, and manage associate row processing. Each entity type has one ENTITY trigger module.

To ensure the uniqueness within the model, the default member name for ENTITY trigger modules is an E followed by a number. To use specific member names, specify the member names before generation.

An action block or procedure step calls an ENTITY trigger module to manage the delete action, or by another trigger module when the RI rules imply that delete.

RELATIONSHIP Trigger Modules

RELATIONSHIP trigger modules execute the update, that is, set the foreign key to null, to implement a DISASSOCIATE or TRANSFER in an action diagram. They also identify more processing that is required, based on the ERD rules and the type of action that is requested, a DELETE, DISASSOCIATE, or TRANSFER. A call to the appropriate ENTITY trigger module processes additional deletions.

The RELATIONSHIP trigger module manages the delete or disassociate of the link records that are required for many-to-many relationships. One RELATIONSHIP trigger module exists for each relationship.

To ensure the uniqueness within the model, the default member name for RELATIONSHIP trigger modules is an F followed by a number. CA Gen assigns default names when the entity types are added on the workstation, or when foreign key linkage is added to a data record as the relationship is transformed. To use specific names, specify member names before generation.

Specify the DB2 Bind Package Defaults for Referential Integrity Triggers

You only identify the DB2 Package Bind defaults for RI triggers when the target database is DB2.

Follow these steps:

1. Log on to TSO/ISPF.

   The ISPF Main Menu displays.

2. Select CA Gen and press Enter.

   The CA copyright screen displays.

3. Press Enter to display the Main Menu.

4. Type 4 for Application system functions and press Enter.

5. Type 5 for the Bind options. Press Enter.

6. Type 2 for Set Bind Package Defaults for RI Triggers and press Enter.

7. Move the cursor to each field to make an entry and type the entry. Press Prompt for context-sensitive Help for the field.

8. Press Accept to verify the entries when you finish.

9. Press Save to save the entries.

   The Bind Options menu displays.

10. Press End to return to the Application System Functions menu.

Open the Referential Integrity Construction Menu

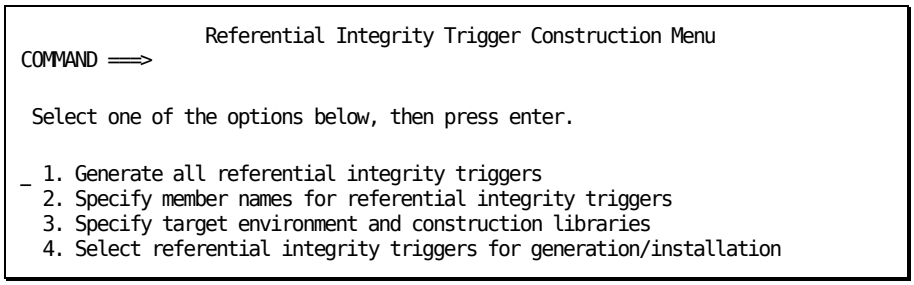
To display the panels to generate RI triggers, use the RI Construction menu.

Follow these steps:

1. Type 4 for Application system functions on the CA Gen Main Menu and press Enter.
2. Type 3 for Referential Integrity Trigger Construction and press Enter.
3. Type the Host Encyclopedia model name and press Enter.

   or

   Use the Tab key to move the cursor to Model name and press the Prompt key for a list of models. Select a model from the list by typing any character except a period (.) or a space character next to one name and press Enter.



Note: When you use DB2 package bind to generate RI Triggers, specify the DB2 Subsystem for Package Binds on the Specify RI Trigger Target Environment Parameters panel.

Specify the Referential Integrity Trigger Target Environment

Follow these steps:

1. On the Application System Menu, select option 3, Referential integrity construction.
2. Type a Model Name and press Enter.

3. Type 3 for Specify target environment and construction libraries and press Enter.

   or

   Use the Tab key or the cursor keys to move the cursor to the left of the option and press Enter.

```
        Specify RI Trigger Target Environment and Construction Libraries
COMMAND ===>

Model name  . . : ABLK 04 TD PERMITTED VALUES

Select one of the options below, then press enter.

_ 1. Specify RI Trigger Target Environment
  2. Specify RI Trigger Construction Libraries
  3. Specify RI Trigger Compatibility Libraries
  4. Specify RI Trigger Dynamic Link Libraries
```

   Note: CA Gen only requires the RI Trigger Compatibility libraries when you select the Process modules marked for Compatibility option in the Generation Options panel.

4. Type 1 for Specify RI Trigger Target Environment and press Enter.

   The Specify RI Trigger Target Environment Parameters panel displays with the target environment parameters:

```
                Specify RI Trigger Target Environment Parameters


  COMMAND ===>

  Specify desired target environment parameters

     Operating System . . . . . .   MVS
     Generated Source Language  . . . . .   COBOL
     Database Management System . . . . .   DB2
     DB2 Subsystem for package binds. . .   SSD2
     TP Monitor . . . . . .   CICS         (CICS, IMS, IEFAE)
     Dynamically Link RI Triggers . . . .   YES          (Yes, No)
     RI Trigger Name (Cascade). . . . . .   CASCADE
```

5. Type DB2 for Database Management System. When using DB2, type a one to four character DB2 Subsystem for package binds name.

   Note: CA Gen completes the Operating System and Generated Source Language. You cannot modify these values.

6. Override the TP Monitor for dynamically linked RI trigger modules, if necessary.

7. Type YES or NO for Dynamically Link RI Triggers.

   When you type YES, CA Gen builds the RI trigger modules as fully resolved dynamic DLL modules.

8. Modify the RI Trigger Name, if necessary.
9. Press Accept to save changes.

   The Specify RI Trigger Target Environment and Construction Libraries menu displays with the phrase Referential Integrity Trigger Target Environment accepted under the Command prompt.

   CA Gen saves the Dynamically Link RI triggers option and RI trigger Name settings in the model.
10. Press End or Cancel to return to the previous panel.

More information:

Generation Options (see page 31)
Specify the Referential Integrity Trigger Construction Libraries (see page 55)

Specify the Referential Integrity Trigger Construction Libraries

When specifying the RI Trigger Construction Libraries, also specify the RI Trigger Compiled non-DLL Load Modules library, the RI Trigger Compiled Load Modules libraries, and the RI Trigger Dynamic Link libraries. The RI Trigger Compatibility Compile listings library is optional.

Follow these steps:

1. Display the Referential Integrity Construction Menu.
2. Type 3 for Specify Target Environment and Construction Libraries and press Enter.
3. Type 2 for Specify RI Trigger Construction Libraries and press Enter.
4. Use Tab to move the cursor to each field to make an entry and type the required library name.

   Note: Allocate the construction libraries outside of CA Gen before using them. CA Gen does not allocate the libraries.

   Enter a valid PDSE (DSNtype=library) data set name for the Compiled Load Modules and valid partitioned data set (PDS) library names for these other construction libraries:

   - Generated source code
   - Compiled load modules
   - DB2 DBRM modules – required for DB2

- Installation control
- Compile the listings (optional)

Use fully qualified data set names, which are enclosed in single quotes. Specify all required libraries before exiting to successfully generate.

5. Select a command to exit the panel:

   Accept

   When entry is complete.

   CA Gen accepts the library names that you typed to temporarily override parameters that are saved in the Host Encyclopedia, returns to the previous menu, and displays this message:

   ```
   Temporary override of RI trigger libraries accepted.
   ```

   Save

   Stores the parameters in the Host Encyclopedia, returns to the previous menu, and displays this message:

   ```
   Referential integrity trigger libraries saved on encyclopedia.
   ```

   End

   Exits without accepting the parameters and displays this message:

   ```
   User exited from function in progress.
   ```

   Cancel

   Exits the panel, returns to the previous menu or panel without accepting parameters, and displays this message:

   ```
   User canceled previous screen display.
   ```

When the model contains modules that are marked for Compatibility, specify the RI Trigger Compiled non-DLL Load Modules library and the RI Trigger Compiled Load Modules libraries. The RI Trigger Compatibility Compile listings library is optional.

When Dynamically Link RI Triggers is set to YES, specify the RI Trigger Dynamic Link libraries.

More information:

Specify the RI Trigger Compatibility Libraries (see page 57)
Specify the RI Trigger Dynamic Link Libraries (see page 58)

Specify the RI Trigger Compatibility Libraries

When the model contains modules that are marked for Compatibility, specify the RI Trigger Compiled non-DLL Load Modules library and the RI Trigger Compiled Load Modules libraries. The RI Trigger Compatibility Compile listings library is optional.

Follow these steps:

1. Type 3 for Specify RI Trigger Compatibility Libraries and press Enter.
2. Use Tab to move the cursor to each field and type the required library name.

   Important! Specify different libraries for RI Trigger Compatibility and RI Trigger Construction.

   Since CA Gen does not allocate libraries, allocate the libraries outside of CA Gen before using them.

   - Enter a valid PDSE (DSNtype=library) data set name for the Compiled non-DLL Load Modules and Compile listings (optional).
   - Use fully qualified data set names that are enclosed in single quotes.
   - Specify the Compiled non-DLL Load Modules library before installing RI triggers.
3. Select a command to exit the panel:

   Accept

   When entry is complete.

   CA Gen accepts the library names that you typed to temporarily override parameters that are saved in the Host Encyclopedia, returns to the previous menu, and displays this message:

   ```
   Temporary override of compatibility libraries accepted.
   ```

   Save

   Stores the parameters in the Host Encyclopedia, returns to the previous menu, and displays this message:

   ```
   Compatibility libraries saved on encyclopedia.
   ```

   End

   Exits without accepting the parameters and displays this message:

   ```
   User exited from function in progress.
   ```

   Cancel

   Exits the panel, returns to the previous menu or panel without accepting parameters, and displays this message:

   ```
   User canceled previous screen display.
   ```

Specify the RI Trigger Dynamic Link Libraries

When Dynamically Link RI Triggers is set to YES, specify the RI Trigger Dynamic Link libraries.

Follow these steps:

1. Type 4 for Specify RI Trigger Dynamic Link Libraries and press Enter.
2. Use Tab to move the cursor to the field to type the library name for Executable Load modules.
3. If necessary, specify optional library names for Executable Batch Load modules, Binder control cards, and Batch Binder control cards.

   Important! Referential integrity trigger executable library must be different from other load libraries specified in this model.

   Since CA Gen does not allocate libraries, allocate the libraries outside of CA Gen before using them.

   - Enter a valid PDSE (DSNtype=library) data set name for the Executable Load Modules and valid partitioned data set (PDS) library name for Binder control cards.
   - Use fully qualified data set names that are enclosed in single quotes.
   - Specify the Compiled non-DLL Load Modules library before installing RI triggers.
4. Select a command to exit the panel:

   Accept

   When entry is complete.

   CA Gen accepts the library names that you typed to temporarily override parameters that are saved in the Host Encyclopedia, returns to the previous menu, and displays this message:

   ```
   Temporary override of dynamic link libraries accepted.
   ```

   Save

   Stores the parameters in the Host Encyclopedia, returns to the previous menu, and displays this message:

   ```
   Dynamic RI Trigger libraries saved on encyclopedia.
   ```

   End

   Exits without accepting the parameters and displays this message:

   ```
   User exited from function in progress.
   ```

Cancel

Exits the panel, returns to the previous menu or panel without accepting parameters, and displays this message:

```
User canceled previous screen display.
```

Specify the Referential Integrity Trigger Member Names

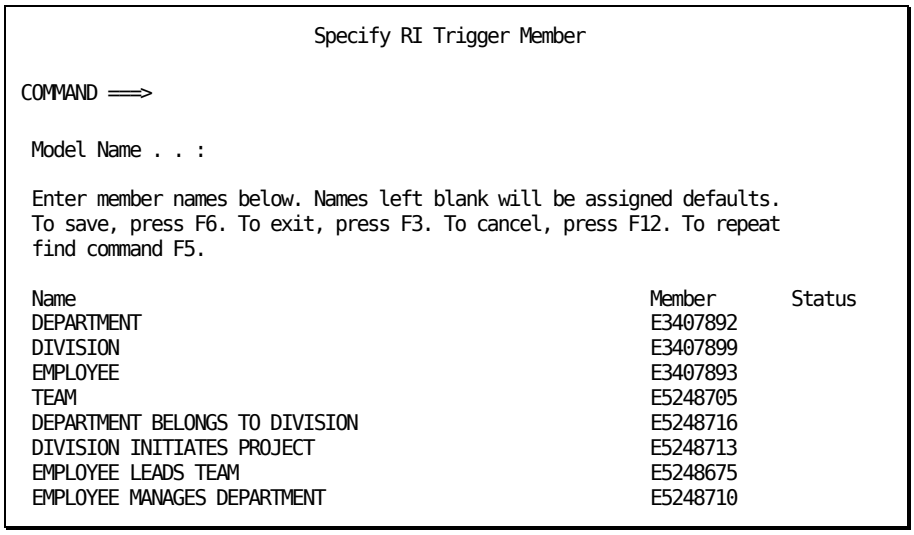
You do not need to specify RI trigger member names before generating triggers. When you omit member names, CA Gen creates default names before generating triggers.

Follow these steps:

1. Display the Referential Integrity Trigger Menu.

   For more information about opening this menu, see the section Open the Referential Integrity Trigger Menu.

2. Type 2 for Specify member names for referential integrity triggers. You can also move the cursor to the left of the option using the Tab key or the cursor keys. Press Enter.



```
                          Specify RI Trigger Member

COMMAND ==>

 Model Name . . :

 Enter member names below. Names left blank will be assigned defaults.
 To save, press F6. To exit, press F3. To cancel, press F12. To repeat
 find command F5.

 Name                                                  Member      Status
 DEPARTMENT                                            E3407892
 DIVISION                                              E3407899
 EMPLOYEE                                              E3407893
 TEAM                                                  E5248705
 DEPARTMENT BELONGS TO DIVISION                        E5248716
 DIVISION INITIATES PROJECT                            E5248713
 EMPLOYEE LEADS TEAM                                   E5248675
 EMPLOYEE MANAGES DEPARTMENT                           E5248710
```

3. Use Tab to move the cursor to each field in which you want to make an entry and type the member name.

   - RI trigger member names must conform to these rules:
   - Must be valid partition data set (PDS) member names
   - Can be up to eight bytes long
   - Can begin with an alphabetic or national character, #, $, or @.
   - Must be unique within the model

4. Select a command to exit the panel:

   Save

   Stores the member names in the model, returns to the previous menu, and displays this message:

   ```
   Member names saved.
   ```

   End

   Exits without saving member name changes and displays this message:

   ```
   User exited from function in progress.
   ```

   Cancel

   Exits the panel, returns to the previous menu or panel without saving member name changes, and displays the message:

   ```
   User canceled previous screen display.
   ```

Generate all Referential Integrity Triggers for a Model

You can generate triggers for entities that are not implemented if they are not referenced directly or by a referential integrity trigger module. Referential integrity generation fails if it encounters an entity that is referenced but not implemented. This means that the entity has not been transformed for the Data Structure Diagram.

Follow these steps:

1. Display the Referential Integrity Trigger Menu.

   Note: Be certain you specify the target environment and construction libraries before selecting to generate all triggers option or selecting individual triggers for generation.

2. Type 1 for Generate all referential integrity triggers and press Enter.

   The Referential Integrity Processing Options panel displays.

3. Type any character except a period (.) or a space in the space next to the option and press Enter.

The options are mutually exclusive. You cannot enable both.

Selections in the Generation Options panel can cause CA Gen to display the Confirm Referential Integrity Trigger Generation panel. Press Enter to begin processing or Cancel to stop processing.

When you select the foreground processing, CA Gen displays the Generation Status panel. The system displays log messages to track progress during trigger generation and installation.

Press Enter to close the Generation Status panel when generation completes.

If you select background processing and set the options to display the Update JCL for Background Generation panel in the Generation Options panel, CA Gen displays the panel. To suppress displaying the Update JCL for Background Generation panel, modify the Job card and press Enter to submit the job.

More information:

Generation Options (see page 31)

Selectively Generating Referential Integrity Triggers

Use this panel to select only the triggers that did not generate or compile. Successfully generate all of the RI Triggers before attempting to generate the source code for the application.

Settings in the Generation Options panel affect panels that open as a result of choices in this section.

Note: This feature does not replace Intelligent Regeneration for the regeneration of triggers after changes to the model.

Follow these steps:

1. Display the Referential Integrity Trigger Menu.
2. Type 4 for Select referential integrity triggers for generation/installation. You can also move the cursor to the left of the option using Tab or the cursor keys. Press Enter.

The Specify RI Triggers for Generation panel displays.

```
                    Select RI Triggers for Generation          Row 1 of 10
COMMAND ===>                                                SCROLL ===> PAGE

 Model Name . . : TEST INSTALLATION MODEL

Type action codes next to one or more triggers, then press enter.
To exit, press F3.  To cancel, press F12.  To repeat a find command, F5.

G=Generate  I=Install  B=Generate and Install

    Name                                Generation      Compilation
                                        Date    Time     Date     Time  Member
Opt
 _ DEPARTMENT                       01/01/9x   8:00  01/01/9X    8:00 E3407892
 _ DIVISION                         01/01/9x   8:00  01/01/9X    8:00 E3407899
 _ EMPLOYEE                         01/01/9x   8:01  01/01/9X    8:01 E3407893
 _ TEAM                             01/01/9x   8:01  01/01/9X    8:01 E5248705
 _ DEPARTMENT                       01/01/9x   8:01  01/01/9X    8:01 E5248716
```

3. Choose an option and press Enter:

   - Type G to generate
   - Type I to install
   - Type B to generate and install

   Install compiles the RI trigger and creates unresolved (NCAL) load modules when the Dynamically Link RI triggers option is set to NO in the Specify RI Trigger Target Environment Parameters panel. When the option is YES, the installation compiles the RI triggers and creates a fully resolved DLL executable module comprising RI triggers. If RI Executable Batch Load modules library is specified, the installation process creates a batch version of dynamic RI DLL executable and saves it in this library automatically. DB2 attachment type for batch dynamic RI will be as specified by the TIRDBATT parameter, defaulting to DSN.

   When you chose to display the Referential Integrity Processing Options panel on the Generation Options panel, the panel displays to select processing parameters.

   For generation, you can select background or foreground processing.

   For the installation, you can select background or foreground processing, remote or local installation, forced compilation of triggers, and Process modules that are marked for Compatibility.

   Select the Process modules marked for Compatibility option in the Generation Options panel when the model for which you are installing the RI Triggers contains modules that are marked for compatibility.

   The Process modules marked for Compatibility option cannot be set to Yes when the Dynamically Link RI triggers is set to Yes. When both are set to Yes, the RI Generation and Installation process displays this message:

   ```
   Cannot select Process Compatibility when Dynamically Link RI is YES
   ```

4. Type any character except a period (.) or a space next to the option to enable. The options are mutually exclusive. You cannot enable both. Press Enter.

   The Confirm Referential Integrity Trigger Generation panel displays if chose that option on the Generation Options panel. Press Enter to begin processing or click Cancel to stop processing.

5. CA Gen displays the Generation Status panel when you chose foreground processing. Press Enter to close the Generation Status panel when generation completes.

   The system displays log messages to track progress during trigger generation and installation.

6. CA Gen displays the Update JCL for Background Generation panel for job card entry if you selected background processing and you set the options to display this panel in the Generation Options panel. To suppress Update JCL for Background Generation, modify the job card and press Enter to submit the job.

7. Press Exit to leave the RI trigger construction menu when generation completes.

More information:

Specify the Referential Integrity Trigger Target Environment (see page 53)
Generation Options (see page 31)

Modifying Referential Integrity Triggers for Migration

Job CEJOB05A modifies operational rules for the migration of referential integrity (RI) triggers with their parent objects: entity types and data records.

The decision about whether to modify the default rules for migration must be made within your development group.

Note: For information about instructions for preparing the JCL to run CEJOB05A and the effects of running it, see Modifying Referential Integrity Triggers for Migration section of the Migration Rules chapter in the *Host Encyclopedia Version Control Guide*.

Chapter 5: DB2 Package Bind Options

Bind an Individual DBRM

The DB2 Package Bind option provides a more granular approach to binding than Plan Bind option. Construction gives the operator options to bind an individual Database Request Module (DBRM) as a package.

The Package Bind option is selected by entering a Collection ID on one of the bind options panels.

Advantages

Using package bind provides definite advantages. You can rebind a package (one DBRM) rather than a whole plan. When you change a program in an application, you can rebind the package for that program's DBRM without having to rebind all unchanged DBRMs in the application plan.

If you do not use packages, and a changed program resides in other plans, rebind those other plans.

Package Bind Terms

The following terms are related to the DB2 Package Bind Option:

Package

An individually bound DBRM.

Location

The database server's unique name. An application uses location to access a DB2 database server. The application can use a database alias to override the location name when accessing a remote server.

Collection

A logical grouping of packages. The collection identifier (ID) is part of the package name. The collection is not a physical entity. DB2 uses the collection ID to allow a plan to reference multiple packages by a single identifier. Specify the collection ID on one of the bind options panels to do package binding.

Package List

Input to a plan bind that defines the search order that DB2 is to use when looking for packages. It can include specific package names, or a collection ID with a wildcard (collection_ID.*) for the package ID. In the latter case, the entire collection is searched for a match on package ID and consistency token. Collection_ID can be qualified with a location also, for example, location.collection_ID.

Version

An identifier that differentiates between two precompilations of the same source member at different points in time. It can be anything meaningful to the user such as TEST and PROD or Release 1, and Release 2. Version allows a collection to include more than one package with the same name. The package that is used is always the one with the matching consistency token. Version allows easy backout of changes. This can also be done without version by using more than one collection.

Owner

The package owner's authorization ID.

Qualifier

Implicit the qualifier for the unqualified table, view, index, and alias names in the static SQL statements of the package.

Package Bind Options and Levels Affected

You can specify package bind options are specified at four levels - Model, RI Triggers, Business System, and Member and affect different systems or members at each level:

- At the Model level, package bind options affect all members in the model
- At the RI Triggers level, package bind options affect all RI Triggers
- At the Business System level, package bind options affect all members in the business system
- At the Member level, level, package bind options affect selected members only

Package List Order

Package List Order option lets you select the order of locations, collection, and packages in the PKLIST parameter of the plan bind. If omitted, the PKLIST consists of one or more COLLECTION_ID.* entries in alphabetical order.

Package List Order option also lets you specify locations, collections, packages that are not referenced by any members. This option enables you to include packages for external action blocks.

Package list order is specified at three levels - Model, Business System, and Load Module and affect different modules depending on the level:

- At the Model level, package list order affects all load modules in the model
- At the Business System level, package list order affects all load modules in the business system
- At the Load Module level, package list order affects selected load modules only

Package Bind Procedures

These procedures document the instructions to use CA Gen to bind DB2 packages.

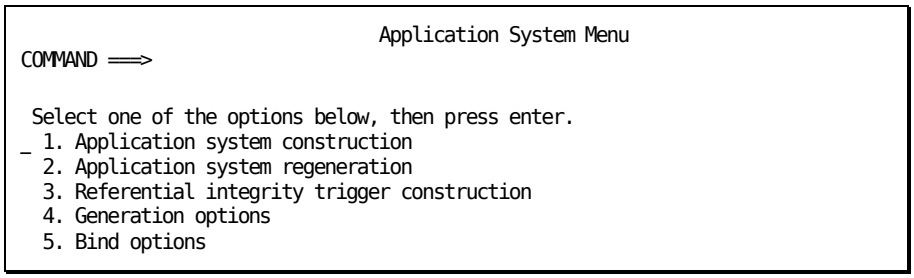
Note: For more information, see the Menu Online Help (F1) and Help Prompt (F4).For database-specific information, see DB2 documentation.

Set Package Bind Defaults

Use the Application System Menu to open the Bind Options menu to set Package Bind default values.

Follow these steps:

1. From the CA Gen Main menu, select option 4 and press Enter.



2. From the Application System Functions menu, select option 5 and press Enter.

```
                                    Bind Options
COMMAND ===>

 Model Name: _____________________________+

 Select one of the options below, or by number, or by positioning the cursor.

 Set Package Bind Defaults
  _ 1. Model
    2. RI Triggers
    3. Business Systems
    4. Members

 Set Package List Order
    5. Model
    6. Business Systems
    7. Load Modules

NOTE: This panel is required only if you want support for DB2 package bind.
      There are no changes to plan bind beyond adding the PKLIST parameter.
```

In the Bind Options panels in which you can enter parameters, use the following function keys:

F1 (Help)

Displays context-sensitive Help for the current screen.

F2 (Accept)

Checks that the values are valid. Does not update the encyclopedia or exit the screen.

F3 (End)

Exits the screen without saving and returns to the previous panel.

Note: If the input data has changed, you cannot use F3 to End. Use F12 to End without saving the changes to the encyclopedia.

F4 (Prompt)

Displays context-sensitive help for data entry fields.

F6 (Save)

Updates the encyclopedia, exits the screen, and displays the previous panel.

Note: This PF key is available only in certain screens.

F12 (Cancel)

Always exits without saving and displays the previous panel.

Set Package Bind Defaults for the Model

To open the Set Package Bind Defaults for the Model panel, use the Bind Options menu.

Follow these steps:

1. Select option 1 on the Bind Options menu and press Enter.

```
              Set Package Bind Defaults for The Model

COMMAND ===>_

Enter package bind defaults.  To save, press F6.  To cancel, press F12.
Location:     _______________       Spaces = Use the default location ID
Collection:   ________________      Spaces = Do NOT do a package bind
Version:      ____________________________________________________________
Owner:        ________              Spaces = Use the primary authorization ID
Qualifier:    ________              Spaces = Do NOT use a qualifier ID
Isolation:    __                  CS, RR, RS or UR
Validate:     ____                BIND or RUN
Release:      _________           COMMIT or DEALLOCATE


To delete the defaults, type DEL on the command line
```

2. Position the cursor in each field and type the entry. Use Prompt to display context-sensitive Help.
3. Accept verifies your entries.
4. Save stores your entries.

Set Package Bind Defaults for Referential Integrity Triggers

To set the package bind defaults for referential integrity triggers, use the Set Package Bind Defaults for RI Triggers menu.

Follow these steps:

1. On the Bind Options menu, select option 2 and press Enter.

```
              Set Package Bind Defaults for RI Triggers
COMMAND ===>

Enter package bind defaults.  To save, press F6.  To cancel, press F12.

Location:    _______________      Spaces = Use the default location ID
Collection:  ________________     Spaces = Do NOT do a package bind
Version:     ____________________________________________________________
Owner:       ________             Spaces = Use the primary authorization ID
Qualifier:   ________             Spaces = Do NOT use a qualifier ID


Isolation:   __                  CS, RR, RS or UR
Validate:    ____                BIND or RUN
Release:     _________           COMMIT or DEALLOCATE



To delete the defaults, type DEL on the command line
```

2. Position the cursor in each field and type the entry. Use Prompt for context-sensitive Help.
3. Use Accept to verify your entries.
4. Click Save to store your entries.

Set Package Bind Defaults for Business Systems

Follow these steps:

1. Select Option 3 on the Bind Options menu and press Enter.

```
                          Select Business System                Row 1 of 2
COMMAND ===>                                                  SCROLL ===> PAGE

Type / next to one or more names, then press Enter to set defaults for all.

 Name                           Collection         Version       Owner    I V R
.APPLICATION_HELP
     Location:                    Qualifier:
.HELP_MAINTENANCE
     Location:                    Qualifier:
************************ BOTTOM OF DATA ************************************
```

2. Select one or more business systems and press Enter to set defaults for all selections.

```
              Set Package Bind Defaults for Business Systems
COMMAND ===>

Enter package bind defaults.  To save, press F6.  To cancel, press F12.

Location:    ________________     Spaces = Use the default location ID
Collection:  ________________     Spaces = Do NOT do a package bind
Version:     ____________________________________________________________
Owner:       ________             Spaces = Use the primary authorization ID
Qualifier:   ________             Spaces = Do NOT use a qualifier ID

Isolation:   __                 CS, RR, RS or UR
Validate:    ____               BIND or RUN
Release:     _________          COMMIT or DEALLOCATE


To delete the defaults, type DEL on the command line


F1=Help  F2=Accept  F3=End  F4=Prompt  F6=Save  F12=Cancel
```

3. Position the cursor in each field and type the entry.

   Press Prompt for context-sensitive Help.

4. Press Accept to verify your entries.

5. Click Save to save your entries.

   CA Gen displays the Select Business System panel with Defaults updated under the Command prompt and default choices abbreviated in the IVR columns.

6. Click End to return to Bind Options menu.

To set the package bind defaults for members

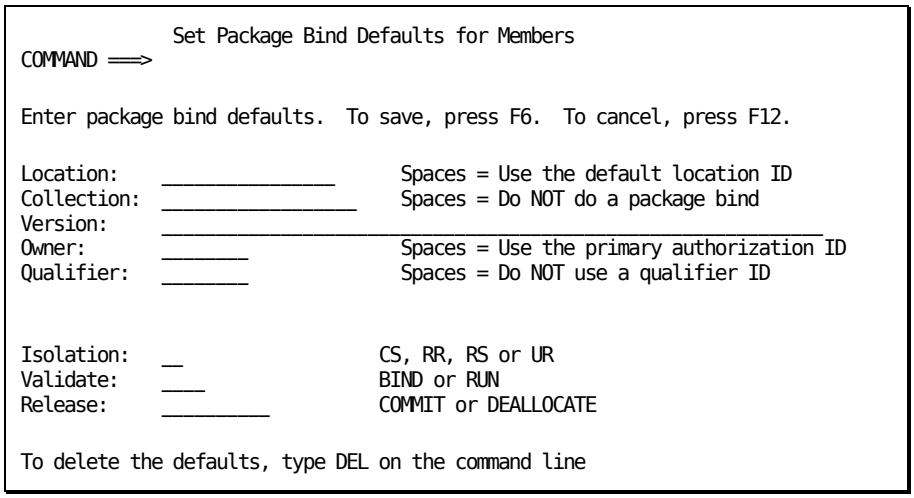
1. Select option 4 on the Bind Options menu and press Enter.

```
                                Select Member                 Row 1 of 2
COMMAND ===>                                                SCROLL ===> PAGE

Type / next to one or more names, then press Enter to set defaults for all.

 Name          Collection              Version        Owner           I V R
. ADDDIV
      Location:                   Qualifier:
. ADDEMP
      Location:                   Qualifier:
. A8914094
      Location:                   Qualifier:
. A8914195
      Location:                   Qualifier:
. A8914106
      Location:                   Qualifier:
```

2. Select one or more members. Press Enter to set defaults for all selections.



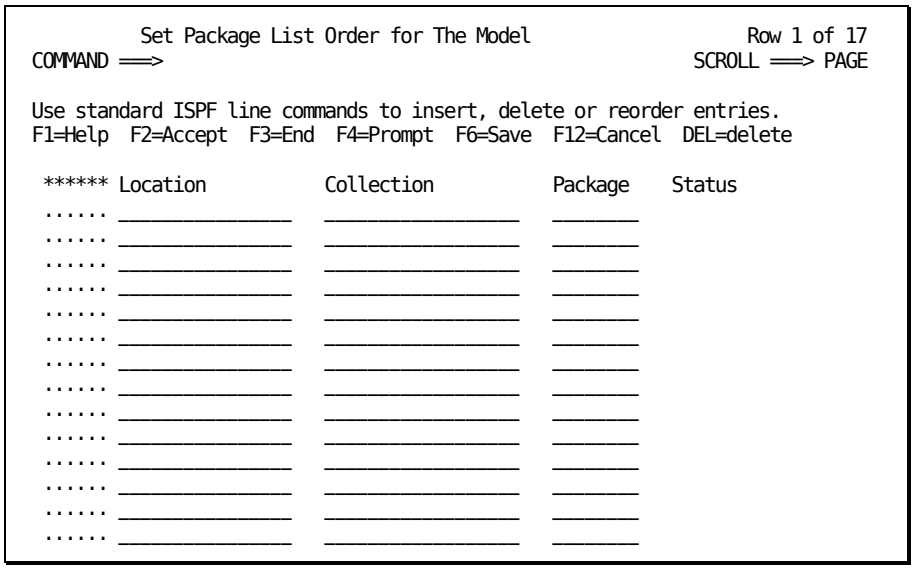
3. Position the cursor in each field and type the entry. Press Prompt for context-sensitive Help.
4. Press Accept to verify your entries.
5. Click Save to save your entries.

   CA Gen displays the Select Member panel with Defaults updated under the Command prompt and default choices abbreviated in the IVR columns.
6. Click End to return to Bind Options menu.

Set Package List Order for the Model

Follow these steps:

1. Select option 5 on the Bind Options menu and press Enter.

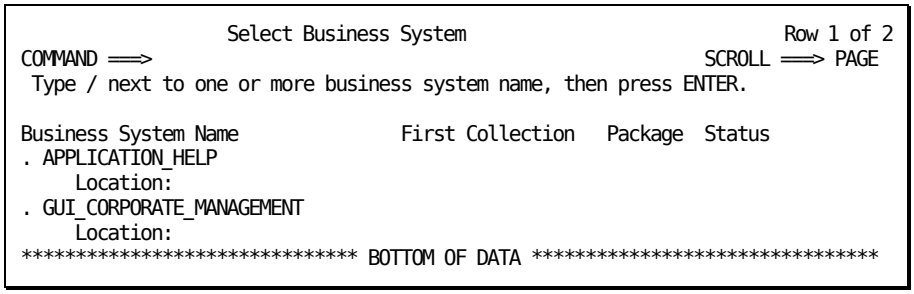


2. Use standard ISPF commands to insert, delete, or reorder entries.
3. Click Accept to verify your changes.
4. Click Save to save your changes.

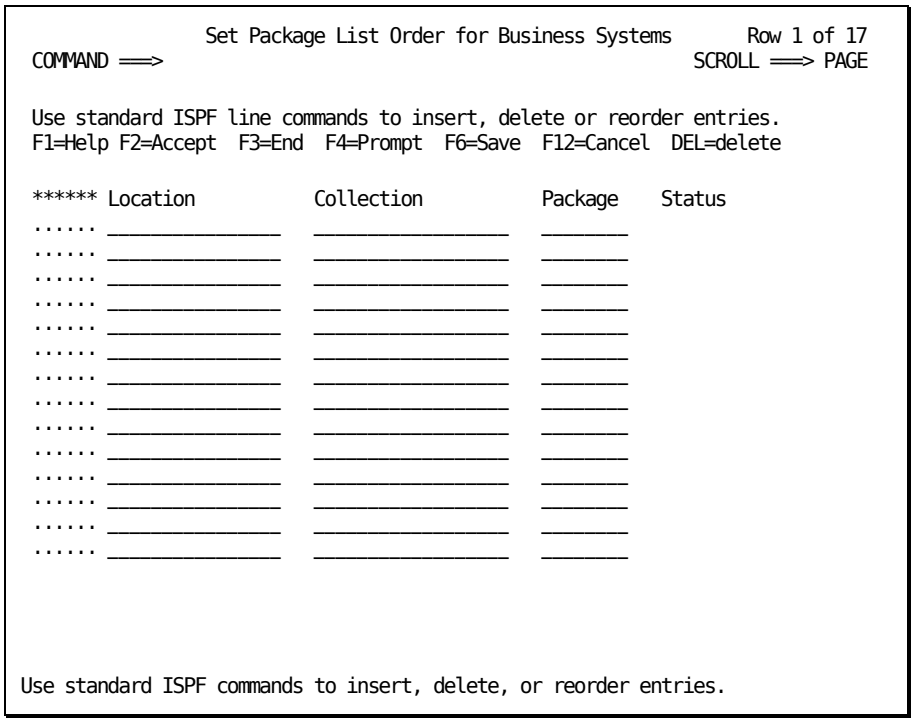
Set Package List Order for Business Systems

Follow these steps:

1. Select option 6 on the Bind Options menu and press Enter.



2. Select one or more business systems and press Enter to set defaults for all selections.

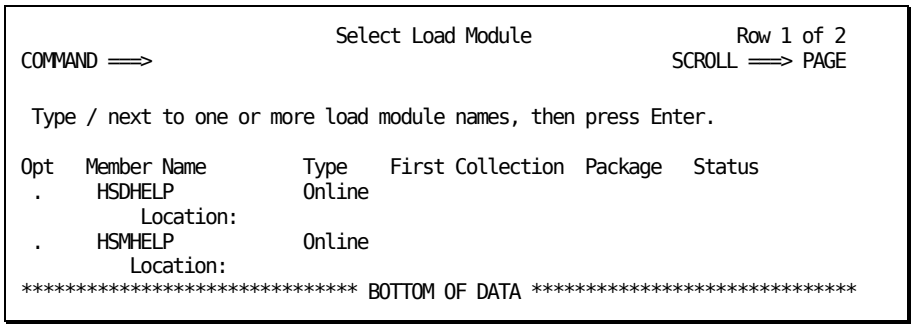


3. Click Save to save your entries.
4. CA Gen displays the Select Business System panel with Defaults updated under the Command prompt and default choices abbreviated in the IVR columns.
5. Click End to return to Bind Options menu.

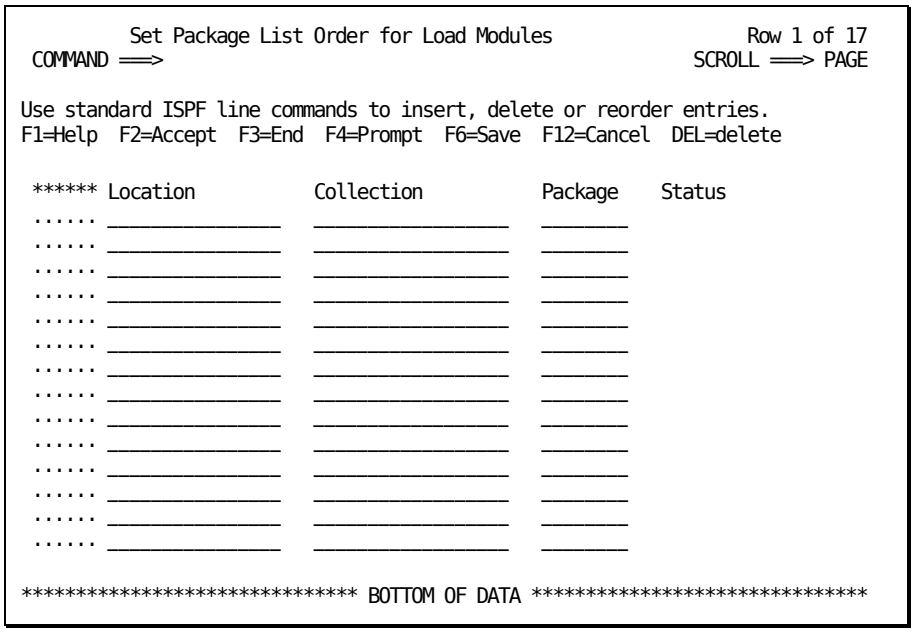
Set Package List Order for Load Modules

Follow these steps:

1. Select option 7 on the Bind Options menu and press Enter.



2. Select one or more load modules and press Enter.



3. Use standard ISPF commands to insert, delete, or reorder entries.
4. Click Save to save your entries.

   CA Gen displays the Select Load Modules panel with status column updated.
5. Click End to return to the Bind Options menu and Application System Functions menu.

Chapter 6: Specify Target Environment for HE

Before generating an application, specify the target environment and construction libraries. This chapter explains how to do so.

Specify the target environment and construction libraries at the business system level. CA Gen stores the specifications in the Host Encyclopedia. They are modified as needed. When you select a business system for a construction function, CA Gen retrieves the library and environment specifications from the Host Encyclopedia.

The target environment is the runtime environment in which the generated application executes. The construction libraries store the objects CA Gen creates during construction and the objects external to CA Gen used during construction.

Introduction

Before generating an application, specify the target environment and construction libraries. This chapter explains how to do so.

Specify the target environment and construction libraries at the business system level. CA Gen stores the specifications in the Host Encyclopedia. They are modified as needed. When you select a business system for a construction function, CA Gen retrieves the library and environment specifications from the Host Encyclopedia.

The target environment is the runtime environment in which the generated application executes. The construction libraries store the objects CA Gen creates during construction and the objects external to CA Gen used during construction.

Rules for Specifying Environment and Libraries

CA Gen allows flexibility in how you specify the target environment and the construction libraries. You use the same specifications for all business systems in a model, or you use different specifications for separate business systems. The following rules apply:

- The target environment and the construction libraries must be specified before the system generation functions can be executed.
- The libraries must be allocated outside of CA Gen. The allocation of libraries is not a function of CA Gen.

  Note: The Business System data sets specified for NCAL, Executable, and RI Trigger Compiled load modules and Compatibility libraries, if used, must be PDSEs. Likewise, the External Action Block, External System Load, and Compatibility External Action Block Libraries, if used, must be allocated as PDSEs.
- The specifications are made at the business system level and are stored in the Host Encyclopedia. This means that you must specify the libraries and the environment for each business system.

Library and Environment Functions

The target environment and library specifications for each business system are stored in the Host Encyclopedia. Each time that you select a construction function, CA Gen prompts you for the model and business system name. CA Gen automatically retrieves the specifications for that business system from the Host Encyclopedia.

CA Gen stores the specifications as ISPF shared variables. They are the current business system specifications. You can temporarily override these specifications without updating the Host Encyclopedia. The specifications remain in effect until you restore the specifications from the Encyclopedia, specify another business system, exit CA Gen, or log out from TSO/ISPF.

CA Gen provides the functions setup and update the specifications initially. The functions are selected from a menu for the Target Environment and Construction Libraries.

Target Application Environment

Before using the construction functions to generate an application system, you must define the target environment for the generated application.

Note: For server applications, specify the target operating system, DBMS, generated source language, and TP monitors for each server load module. Construction of server load modules does not use the operating system, generated source language, DBMS, or TP monitor values on this panel. Instead, server load module construction uses the values in the Specify Server Target Environment panel.

Follow these steps:

1. Open the Target Environment and Construction Libraries menu.
2. Type 1 for Specify Target Environment and press Enter.

```
                    Specify Target Environment Parameters
COMMAND ===>

   Operating System . . . . . .  MVS_____ (MVS)
   Generated Source Language  .  COBOL___ (COBOL)
   Database Management System .  DB2_____ (DB2)
   TP Monitor . . . . . . . . .  CICS____ (CICS,IMS,IEFAE)
   Screen Format Type . . . . .  BYPASS__ (BYPASS,MFS,MFS/EOF,HSTFAC)
   Profile Type . . . . . . . .  TSQ
   Extended Attribute Support .  YES
   Enforce DM Constraints . . .  NO_
   Optimize import view initialization   NO_
   Restartable Application  . .  YES
   Clear Screen Default Command  RESET__  (RESET,RESTART)
   DB2 Subsystem  . . . . . . .  SSD2     (DB2 only)
   Dynamically link procedure steps . .  NO_      (YES, NO, CMP)
   Dynamically link action blocks . . .  NO_      (YES, NO, CMP)
   Dynamically link screen managers . .  NO_      (YES, NO, CMP)
   Pseudoconversational Support . . . .  NO_      (CICS only)
   Handle CICS Command Abends . . . . .  NO_      (CICS only)
   XCTL for flows when possible . . . .  NO_      (CICS only)
```

3. Use the Tab key to move the cursor to each field and type the value.
4. Select a command to exit the panel:

   Accept

   Uses the values only for this CA Gen session. CA Gen displays the following message:

   `Temporary override of target environment accepted.`

   Exit

   Leaves the screen without saving and returns to the previous panel. CA Gen displays the following message:

   `User exited from function in progress.`

Save

Updates the Host Encyclopedia, exits the screen, and displays the previous panel. The system displays the following message:

```
Target environment saved on encyclopedia.
```

Cancel

Always exits without saving and displays the previous panel. CA Gen displays the following message:

```
User cancelled function in progress.
```

Operating System

The operating system on which the generated application executes. Host Encyclopedia Construction supports the z/OS operating system, which is specified in this panel as MVS.

Generated Source Language

The language in which CA Gen generates the application. Host Encyclopedia Construction supports COBOL generation.

Database Management System

The Database Management system the generated application uses. This must be DB2 for applications that are generated on the Host.

Teleprocessing (TP) Monitor

For generated online applications, the Teleprocessing Monitor under which the generated application runs. Support is available for three teleprocessing monitors: CICS, IMS, or IEFAE (TSO).

Screen Format Type

The types of screen formats available for each TP monitor are:

CICS

BYPASS for native 3270 formatting support.

IEFAE (TSO)

BYPASS for native 3270 formatting support.

IMS

MFS (Message Formatting Services) restores the field during input after the erase-end-of-field and erase-input keys.

or

MFS/EOF, a variation of the MFS that erases the field during input after the erase-end-of-field and erase-input keys.

Some considerations affecting performance occur when using this feature due to the extra processing required. Basic Mapping Services (BMS) generation for CICS is not used.

Profile Type

This option controls the storage method for the profile table (stack), which CA Gen uses to retain information for processing and error recovery. For IMS and IEFAE (TSO), use the SQL profile manager. CICS DB2 users have the option of using the SQL or the TSQ profile manager. The SQL profile manager maintains the profile stack on a DB2 database. The TSQ profile manager maintains the profile stack in a CICS temporary storage queue. Use of the TSQ profile manager can improve application performance.

Extended Attribute Support

The extended attribute support option identifies if the generated application runs in an environment whose devices, such as terminals or communication devices, support extended attributes. The extended attributes are reverse video, underline, blinking, and color.

When the target environment supports extended attributes, select Y. When the environment does not support extended attributes, select N. The results are unpredictable when a generated application sends extended attributes to a device that does not support them.

In the CICS environment, CICS checks the CICS terminal definition to determine if the device supports extended attributes. The device definition overrides the Extended Attribute Support option. When the Extended Attribute Support is Yes, the generated application checks the definition before sending the attributes and does not send the attributes when the definition shows that the device does not support extended attributes.

In the TSO and IMS environments, the TIRDEV exit responds that the device supports extended attributes even when it may not actually do so. The user must provide a TIRDEV exit that returns a true indication of the devices capabilities, or to select the relevant extended attribute supported characteristic correctly based on the target devices.

Enforce DM Constraints

The Enforce DM (Data Modeling) Constraints option captures information describing the rules to use in maintaining the integrity of the database. Some rules are enforced by the target DBMS. Other rules are enforced by CA Gen-generated referential integrity triggers when DELETE, TRANSFER, or DISASSOCIATE actions execute.

When you set the Enforce DM Constraints option, CA Gen-generated code enforces four more constraints. This eliminates the need to enforce them with action language in action blocks.

You cannot set these constraints independently of each other. They are selected as a group.

The four more constraints are:

- Enforce One-to-One Relationships
- Enforce Mutually Exclusives
- Enforce Mandatory Relationships
- Prevent Quiet Disassociations

The Enforce DM Constraints option can potentially degrade performance. Therefore, we recommend that you only using this option only during testing to debug. If performance is a serious consideration, regenerate the application with this option set to No before moving into production.

Enforce One-to-One Relationships

CA Gen checks both ends of the relationship before permitting an ASSOCIATE action to prevent the foreign keys of multiple rows from pointing to the same related row. For example, if A is related to B, and A contains the foreign key, the transaction fails with a fatal error unless the following tests are true:

- The foreign key in the view of A to be associated with B must be null.
- No references to B exist in any other A.

Enforce Mutually Exclusives

CA Gen generated code ensures that no ASSOCIATE action violates a defined mutually exclusive constraint. If any member of the mutually exclusive set exists, none of the others are allowed to exist. For example, if A is related to B and to C, in a mutually exclusive set, an action block reads A and attempts to associate it to B. If A is already associated with C, the transaction fails with a fatal error.

Enforce Mandatory Relationships

CA Gen generated code performs an ASSOCIATE action for each mandatory relationship membership when it creates an entity occurrence. CA Gen enforces this constraint in action blocks with CREATE actions. After executing a CREATE on View A, an attempt to READ into View A or exit the action block without associating A or its mandatory relationship results in a runtime error.

Prevent Quiet Disassociations

Consider the effect of overlaying the value in a foreign key when an ASSOCIATE action is applied to a one-to-many relationship. If the foreign key is populated and no cascade delete logic is performed, the effect is the same as a DISASSOCIATE action. This is named *a quiet disassociation* because it executes without performing database integrity checks.

CA Gen generated code checks before performing the ASSOCIATE to see if the attribute on which the foreign key is based is null. If not, the association that is requested must match the existing association. The association exists if the foreign key value equals the value of the attribute on which it is based. The transaction fails with a fatal error when the foreign key is not null and is not the same value as the new key requested.

Optimize Import View Initialization

Indicates to CA Gen that the logic for initializing import views can reside in the named action block. This action reduces the amount of time to initialize import views.

Restartable Application

Specifies if a business system is restartable. A restartable application supports screen clear, other transaction execution, and using the RESTART command to return to the application. Setting an application as restartable improves the system performance by reducing use of the runtime profile table.

CA Gen generated applications use a temporary runtime stack, the Profile Database, to maintain execution context. CA Gen maintains one stack for each business system for each user. For example, when working in three CA Gen generated business systems, you have three stacks, one for each business system.

If an application is non-restartable, CA Gen still generates the profile table, but only accesses it to support the following situations:

- Returns from links
- Hidden views
- Scrollable repeating groups
- HELP and PROMPT requests

If the business system is non-restartable, RESTART still works in certain situations. The profile table supports the previous situations regardless of the restartable specification. Therefore, a transaction restarts in these situations:

- It uses hidden views
- A SET NEXTTRAN is generated

A business system is always restartable in all situations if it is specified as restartable.

Clear Screen Default Command

For online transactions, CA Gen supports two clear screen commands: RESET and RESTART. RESET clears the stack, the profile, for the user and business system and begins a new dialog. RESTART continues a dialog using the top entry from the stack. The top entry is always the last active screen for the business system, regardless of the transaction code entered.

The Clear Screen Default Command option specifies the command that the generated application executes when a transaction code is entered without a command on a clear screen. To execute the command that is not the specified default, the user of the generated application must clear the screen and must enter the transaction code followed by the command. The format is:

```
trancode command
```

For example, if the default command selected is RESET, when the application user clears the screen and enters a transaction code, the stack is cleared and a new dialog begins. To restart a dialog, the user must enter the clear screen transaction code followed by the RESTART command. The format is:

```
trancode RESTART
```

Note: When the application is non-restartable, the clear screen default command is RESET. When you type the transaction code without the RESET command, the table is not cleared because the application is not configured to use a runtime profile table. To clear the table, specifically enter the clear screen transaction code followed by the RESET command.

DB2 Subsystem

The DB2 Subsystem is the name of the database location where the generated application runs. CA Gen uses the name to bind packages or plans and when invoking application programs.

Dynamic Linking

CA Gen generates different COBOL code when calling a routine that is statically linked within an executable and when calling a routine that resides in a separate executable. For calls to routines that are statically linked within an executable, CA Gen generates a COBOL CALL literal statement.

For calls to routines that reside in an executable separate from the calling executable, CA Gen generates a COBOL CALL identifier statement that references a variable. The variable contains the name of the routine that is to be dynamically invoked. A program call to a routine that resides in a separate executable is known as dynamic linking.

Dynamic linking is implemented by CA Gen using a z/OS specific packaging option that indicates how a routine is built. How the routine is built determines how it is called. The specific dynamically link packaging property that is associated with each procedure step, screen, or action block (including external action blocks) identifies how that component is resolved during the installation of the CA Gen load module in which it is packaged.

The designation of the dynamically link packaging option for a procedure step, screen, or action block can be set to Default. In this case, the dynamically link packaging option is derived from the dynamically link packaging option that is established in the business system owning the CA Gen load module in which the given procedure step, screen, or action block is defined.

The following values can be explicitly set for each individual procedure step, screen, or action block (or if set to Default, the value is derived from their respective Dynamically Link packaging option that is obtained from the default value that is established in the Business System):

- No—the routine is statically linked into the application.
- Yes—the routine is resolved as the target of a dynamic program call and as such is considered to be dynamically linked at runtime. During the installation of the load module, those components that have their associated dynamically link packaging property set, or derived, to Yes are built so they reside in their own separately loadable executables. These application routines reside in DLLs.
- CMP—(Compatibility) the routine is resolved as the target of a dynamic program call and as such is considered to be dynamically linked at runtime. A routine that is designated as Compatibility reside in a non-DLL executable.

A dynamic program call to a routine that resides in a DLL is invoked directly by the generated COBOL CALL statement. A dynamic program call to a routine that resides in a non-DLL module is indirectly invoked by CA Gen z/OS runtime.

Note: Every module that makes a dynamic program call to a routine marked for Compatibility must be regenerated and reinstalled to incorporate the call to the runtime routine that handles the indirect call processing.

For a module built before AllFusion Gen 7, identifying it for Compatibility allows that module to be dynamically called by a CA Gen routine that resides in a DLL.

It is possible to migrate procedure steps, screens, or action block routines that were built before AllFusion Gen 7 and must continue to reside in a non-DLL executable. CA Gen allows a module that is explicitly set to, or defaulted to, Compatibility to be built as a non-DLL executable. These migrated non-DLL executables use the same CA Gen z/OS runtime as those application routines that reside in DLLs.

The CA Gen Toolset, Client Server Encyclopedia, and Host Encyclopedia each provide an option that indicates if modules marked for Compatibility should be processed, that is generated or installed.

- If the intent is to use routines that are created with a release of AllFusion Gen before Release 7, Process modules that are marked for Compatibility is not set when generating and/or installing a load module that contains the item marked for Compatibility.
- If the intent is to create a current version of the non-DLL routine, Process modules that are marked for Compatibility is set when generating or installing a load module that contains the item marked for Compatibility.

  Selecting the Process modules marked for Compatibility option causes the RI Trigger modules and all Action Blocks that are statically linked into a Compatibility module to be compiled twice—once using the NODLL compiler option and again using the DLL compiler option. If the action blocks are External Action Blocks these must also be compiled with the NODLL option to be included in the Compatibility load module.

  The Compatibility option is intended to enable a phased migration of an existing application. It allows routines that have been migrated and reside in DLLs to interoperate with routines that reside in non-DLL executables. The non-DLL executables can themselves be migrated and built using the current release of CA Gen or they can remain as is, having been built with a release of AllFusion Gen before Release 7.

Note: Feature enhancements in future CA Gen releases require that routines using this feature reside in a DLL.

When CICS is the target environment, the following options are available:

- Pseudoconversational Support
- Handle CICS Command Abends
- XCTL for Flows when Possible

These options affect the choice of CICS commands that are used by the application system. The default for each of these options is N (No). This allows you to generate with the testing facility as the target and then to reinstall for CICS without having to regenerate the Dialog Manager.

It is possible to have different combinations of options in each load module. You can flow from a load module with any combination of options to a load module with any other combination. However, you can have only one combination of options per load module, even if you have packaged several steps together. This occurs because each load module contains only one Dialog Manager.

While the options are selected on the Target Environment panel, they do not go into effect until the Dialog Manager is generated and installed. When you change an option, change it on the Target Environment panel, regenerate the Dialog Manager, and install.

Pseudoconversational Support

Choosing Yes for this option prevents terminal users from being able to clear the screen. The CLEAR, PA1, PA2, and PA3 function keys simply refresh the screen. Users are able to execute only the procedure steps that are defined in the Dialog Flow Diagram, or transactions that are called through the NEXTTRAN command.

Choosing Yes is appropriate when you do not want to allow users to clear the screen and enter another transaction code. This option allows you to build systems in which the user is tied to the system (dedicated terminal systems).

The choice is not between conversational or pseudoconversational transactions. All CA Gen generated transactions are pseudoconversational in that they do not remain in memory or wait for a response from the terminal.

If you Click Yes, the generated application behaves differently in two ways.

- When the transaction ends, the command to return to CICS changes to force a return to the transaction to allow it to intercept the CLEAR, PA1, PA2, and PA3 function keys. The command changes from the following command:

  ```
  EXEC CICS RETURN END-EXEC
  ```

  to the following command:

  ```
  EXEC CICS RETURN TRANSID trancode END-EXEC
  ```

- The generated application traps the CLEAR, PA1, PA2, and PA3 function keys and refreshes the screen when these keys are pressed. The application behaves as if you had typed the transaction from a clear screen, using the RESTART parameter.

You need to provide a means of logging off the system because users will not have access to a clear screen. No CA Gen log out routine is supplied, but you can create your own. Assign a PF key to a LOGOFF command. Use a CASE OF command to perform a NEXTTRAN to a non-CA Gen procedure. Or, you can write a non-CA Gen procedure to log out the system. It should include a command such as:

```
EXEC CICS ISSUE DISCONNECT END-EXEC
```

For greater flexibility, set pseudoconversational support to No when testing. Set it to Yes when you are ready to install, or when you want to see if the option works correctly.

Handle CICS Command Abends

Allows you to define the recovery from CICS command abends. If you specify Yes for this option, HANDLE ABEND and HANDLE CONDITION statements are used. For the commands that fail, the user sees a CA Gen runtime error message rather than a CICS error message.

Specifying No causes the CICS error message to display on the screen. A user could clear the screen and could start over. Set this option to Yes when Pseudoconversational support is set to Yes because Pseudoconversational support is to prevent clear screen access.

Choosing Yes means that the CA Gen abend recovery module TIRFAIL intercepts CICS command abends. If you have written an error recovery exit such as TIRTERM, abends also pass through it. However, you do not get a CICS dump for these abends.

Note: For more information about the user exits, see the *User Exit Reference Guide*.

XCTL for Flows when Possible

Specifies to use EXEC CICS XCTL in place of EXEC CICS START when flowing from one CA Gen procedure to another. An XCTL is quicker and uses fewer the CICS resources.

XCTL for Flows when Possible – DB2 Considerations

When the target database management system is DB2, use the Dynamic Plan Selection exit included with CA Gen. When you specify Y for this option, a lengthy online message displays to explain how to use the option.

If you click Yes for the XCTL for Flows when the Possible option, CA Gen uses XCTL for dialog flows when the module being flowed to is in the same CICS region. A START command is issued when flowing to a module in another region.

DB2 can dynamically change plans during the execution of an application. This allows CA Gen to use XCTL, when it can, to implement dialog flows.

When you flow from one CA Gen procedure step to another procedure step that is not located in the same load module, you change plan names. DB2 uses the RCT to associate a transaction name to a plan name. If you have hard-coded plan names in the RCT, you receive the wrong plan name when you attempt to do DB2 calls in the target procedure step. Instead of hard coding plan names, use a dynamic plan selection exit.

CA Gen provides a dynamic plan selection exit, named TIRC$EXT, as part of the installation software. When executed, a CA Gen load module writes a signature record containing the name of the program to a temporary storage queue. The queue name is a combination of the transaction name and the terminal ID. If the terminal ID is not present, the queue name is the task number.

By convention, CA Gen program names and the plan names are the same. DB2 uses the RCT to find the plan name when it encounters the first load module DB2 call. At this point, TIRC$EXT reads the temporary storage queue and returns the program name as the plan name. The temporary storage queue is deleted on exit from the CA Gen load module.

The *z/OS Installation Guide* has instructions to install TIRC$EXT. The exit is compatible with load modules that use START commands. After installing TIRC$EXT, any combination of modules with XCTL and without XCTL can work together. It is not necessary to change the RCT entry back if you drop the XCTL option.

Note: For more information, see the *z/OS Installation Guide.*

More information:

Process Modules Marked for Compatibility (see page 35)
Dynamic Link Features and Considerations (see page 89)

Dynamic Link Features and Considerations

Using dynamic linking can reduce total memory and CPU resources that are required by a TP monitor to process a load module. Dynamic linking common routines eliminate the need to link all of the load modules that use it. Applications that are generated for DB2 require a bind or rebind.

The key features and requirements are:

- Only procedure steps, action blocks, and screens can dynamically link.
- Each dynamic linked module must be a fully resolved module.

  If this fully resolved module is a DLL, the Referential Integrity triggers and statically called action blocks in a procedure step, screen, or action block must be compiled as DLL and with the applicable CA Gen runtimes must be linked into each dynamic linked DLL.

  If this fully resolved module is a module marked for Compatibility, the Referential Integrity triggers and action blocks that are called statically within a procedure step, screen, or action block must be compiled as NODLL and with the applicable CA Gen runtimes must be linked into the dynamic linked Compatibility load module.
- The dynamically link packaging property of individual procedure steps, screens, and action blocks can be set to Default. Default indicates that the value of the dynamically link property for that component is determined by the corresponding default setting, which is established at the business system level.

- Marking a module for Compatibility causes its calling module to be generated and installed so that it processes the call using a module that is provided as part of the CA Gen runtime. The runtime code performs the dynamic program call to the non-DLL load module. In the cases where a module marked for Compatibility issues a dynamic program call to another non-DLL dynamic load module, only the module issuing the first dynamic program call to a Compatibility module requires generation and installation.
- Applications that run under TSOAE and contain modules that are marked for Compatibility cannot dynamically call modules that are built with a release before AllFusion Gen 7.6.
- Enhanced Map Block Mode applications containing screens that are marked for Compatibility cannot dynamically call screen managers that are built with a release before AllFusion Gen 7.6.
- Modules that are marked for Compatibility must follow standard OS or LE linkage conventions and must operate in the same AMODE as the caller. These modules must be non-DLL and must be stand-alone, fully resolved programs, eligible for a dynamic, OS style call.
- The ability to call a procedure step, action block, or screen dynamically allows application changes to become effective without having to link every load module using these modules. However, changes made to a module that uses DB2 SQL requires that the DB2 plan corresponding to the load module containing that module be rebound.
- In CICS, a PPT entry is required for each module that is called dynamically. For more information about defining CICS programs, see CICS documentation.

The decision to dynamically link CA Gen modules, as opposed to generating one executable load module with a static link, is based on the following factors:

- Compatibility—CA Gen creates DLLs except for those modules marked for Compatibility. The Compatibility option uses specific CA Gen runtime to enable DLLs to issue a dynamic program call to non-DLL load modules. This requires that the module issuing the dynamic program call be regenerated and reinstalled.
- CA Gen allows modules that are marked for Compatibility to be rebuilt in such a way that they are able to use CA Gen runtime DLLs. This requires that the modules marked for Compatibility be reinstalled and, if necessary, regenerated.

RI triggers and action blocks, including EABs, statically linked into Compatibility modules must be built using the NODLL compiler option. When these RI triggers and action blocks are also used in CA Gen applications that are built as DLLs, they must be compiled using the DLL option.

Selecting the option Process modules that are marked for Compatibility causes the RI triggers and the statically linked action blocks to be generated and precompiled once but compiled twice – once as NODLL and again as DLL.

- DLLs—The CA Gen runtimes are DLLs. These DLL runtimes are no longer included in generated modules, except for a few that are statically linked with the CA Gen Batch, Online, or Server Managers. This applies to static and dynamic linked CA Gen modules.
- Size—A dynamic linked module must be fully resolved and include components that it calls using a static call. This includes some CA Gen runtimes and Referential Integrity triggers. When more than one dynamic linked modules use the same RI modules, these RI modules are included in each dynamic linked module. Most of the CA Gen runtimes are no longer included in each of the DLL applications that are built by CA Gen so they do not increase the size of these modules.
- Volatility—When a static linked module is changed, the calling module containing the statically linked module must be relinked. Linking a frequently changed module dynamically eliminates the requirement to link every load module that calls it.
- Shared modules—Action blocks that are widely reused within an application environment are good candidates for dynamic linking. Linking a shared module dynamically decreases the amount of storage that is required during execution.
- Frequency of use—Dynamic linking is considered for load modules that are invoked infrequently. Dynamic linking modules that are called infrequently would decrease the size of the base load module. Examples of modules that are called infrequently are exception handling routines and processing options that are rarely selected by the user.

Construction Libraries in HE

The construction libraries, which are PDS and PDSEs, store the objects that are created during system generation. The generated objects are the PDS and PDSE members.

You are responsible for selecting the libraries to be used during construction, identifying these libraries to CA Gen, and creating the libraries. CA Gen *does not* create or allocate the libraries. This must be done outside CA Gen, before the libraries are used.

The construction libraries are allocated with SHR disposition but enqueues are performed to prevent corruption of the PDS and PDSE. Use a unique library for each type of CA Gen generated output. That is, use different libraries for the generated source, JCL, load modules, and so forth.

No construction libraries are needed for database generation because no permanent data sets are created by database generation. Database generation creates the JCL and the DDL used to define a model's DB2 databases, and the IDCAMS data set allocation statements that define the VSAM data set allocation for the DB2 database (table spaces and index spaces). This information is stored in a temporary sequential data set that can be saved.

CA Gen uses construction libraries that are specified for a business system, System Load Libraries, and CA Gen Libraries. The System Load Libraries and CA Gen Libraries are defined during the CA Gen installation. You cannot modify them through the CA Gen Construction Library panels.

The System Load Libraries are external product libraries to run CA Gen-generated applications. For example, the system load libraries typically contain the runtime components for products such as COBOL, PL/I, DB2, IMS, CICS, and TSO.

CA Gen Libraries are the libraries in which CA Gen itself resides. They include one load library and one DB2 DBRM library.

There are two categories of construction libraries - internal and external libraries.

Internal Libraries

The Internal Libraries store the objects that are created during system generation.

You are responsible for creating the internal libraries before CA Gen uses them. CA Gen does not create or allocate the libraries.

CA Gen requires that the internal libraries have certain characteristics. Some libraries are required, and some are required only when the application is installed locally.

The following list summarizes the Data Control Block (DCB) characteristics to specify when allocating each library, if the library is required, the recommended blocksizes (BLKSIZE) for IBM 3380 disk drives. All storage devices use PO organization (ORG) and logical record length (LRECL) of 80 unless otherwise noted. Record format (RECFM) is for all storage devices.

Generated source code - Required

RECFM: FB

BLKSIZE: 9440

NCAL load modules (see Note 3) - Required to install application locally

RECFM: U

BLKSIZE: 19069

Executable load modules (see Note 1) - Required to install application locally

RECFM: U

BLKSIZE: 19069

Executable batch dynamic action block library (see Note 1) – Optional to capture dynamic batch modules

RECFM: U

BLKSIZE: 19069

DB2 Database Request Modules (see Note 2) - Required to install application locally

RECFM: FB

BLKSIZE: 3120

Installation Control - Required

RECFM: FB

BLKSIZE:3120

Binder Control Cards - Optional

RECFM: FB

Batch Binder Control Cards - Optional

RECFM: FB

Generated MFS Source - Required to use MFS

RECFM: FB

BLKSIZE:3120

Generated batch JCL – Required to generate JCL

RECFM: FB

BLKSIZE:3120

Compile listings (see Note 3) -Optional

RECFM:FBA

LRECL: 133

BLKSIZE: 23408

Static non-DLL NCAL modules (see Note 3) - Required for Compatibility application that includes static Action Blocks installed locally

RECFM: U

BLKSIZE: 19069

Static non-DLL NCAL listings (see Note 3) – Optional, used only for Compatibility application that includes static Action Blocks installed locally

RECFM: FBA

LRECL: 133

BLKSIZE: 23408

Notes:

- The DCB characteristics for the internal load library is the same as those used to allocate the System Load Library that is defined when CA Gen is installed and the CA Gen Load Library in which CA Gen itself resides.
- The DCB characteristics for the internal DB2 DBRM Library is the same as those of the System DB2 DBRM Library that is defined when CA Gen is installed and the CA Gen DB2 DBRM Library in which CA Gen DB2 DBRMs reside.
- The Static non-DLL NCAL modules library is only used to store the NCAL modules that are produced by the NODLL compile (and ensuing NCAL link) of action blocks that are statically linked into Compatibility modules. In addition, these same action blocks are compiled with the DLL option so they can be used by DLL applications. The NCALs resulting from this DLL compile are stored in the NCAL load modules library. The compile listings are stored in the corresponding listings libraries, if a library is specified.

You are responsible for monitoring the space utilization of the data sets using the ISPF utility. If system generation terminates abnormally with a D37 or B37 system abend, it is likely that the data sets ran out of space or directory blocks.

The following panel shows CA Gen internal libraries that you can specify:

```
                          Specify Internal Libraries
Command ===>

Enter or verify the following business system library specifications:

 Generated source code . . .   'AAAC.TEST.COBOL'_____________________
 NCAL load modules . . . . .   'AAAC.TEST.NCAL'_____________________
 Executable load modules . .   'AAAC.TEST.LOAD'_____________________
 Exec batch dynam AB library   'AAAC.TEST.LOAD.BATCH'_______________
 DB2 DBRM modules  . . . . .   'AAAC.TEST.DBRM'_____________________
 Installation control  . . .   'AAAC.TEST.INSTCTL'___________________
 Binder control cards  . . .   'AAAC.TEST.BNDCTL'___________________
 Batch binder control cards    'AAAC.TEST.BNDCTL.BATCH'_____________
 Generated MFS/DECforms  . .   'AAAC.TEST.MFSSRC'_____________________
 Generated batch JCL . . . .   'AAAC.TEST.IDCAMS'_____________________
 Compile listings  . . . . .   'AAAC.TEST.LISTING'_____________________
```

You can also specify Static non-DLL NCAL modules and listings.

Generated Source Code Library

The Generated Source Code Library contains source members CA Gen creates during system generation. CA Gen generates source members for Dialog Managers, Batch Managers, Server Managers, screens, procedure steps, and action blocks.

NCAL Load Modules Library

The NCAL Load Library stores the compiled load modules that are created by CA Gen local install process, except for the modules that are produced by the NODLL compile of action blocks that are statically linked into Compatibility modules. The linkage editor creates these members using the NCAL (no-call option). Therefore, they contain unresolved external references and are not executable. The library member name is the component member name. The library member name for the Manager is the load module name.

Using this library allows you to specify a separate library for compiled NCAL load modules. This is useful when relinking the load module for execution in an IMS or CICS environment after testing in the CA Gen Test Facility (TS0).

Important! Do not give the NCAL Load Modules Library the same library name as the Static non-DLL NCAL Modules Library because these are mutually exclusive. Do not give the NCAL Load Modules Library the same library name as the Executable Load Modules Library. Keep the NCAL load modules separate from executable load modules to eliminate the need to regenerate the Manager after changing environment variables.

Executable Load Modules Library

The Install option in system generation is used to create an executable load module. All external references are resolved during installation. The executable load modules are stored in the Executable Load Modules Library. The member name is the load module name.

Executable Batch Dynamic Action Block Library

The batch versions of dynamic action blocks including the z/OS Library DLLs are stored in the Executable Batch Dynamic Action Block library if specified. The member name is the fully executable dynamic action block name or z/OS Library DLL name.

Important! Do not give the Executable Batch Dynamic Action Block Library the same library name as the Executable Load Modules Library.

DB2 DBRM Library

The DB2 DBRM Library stores the Database Request Modules that are generated by the DB2 precompiler. CA Gen generates a DB2 DBRM for each procedure step and action block in a load module that accesses a DB2 table.

Installation Control Library

The Install option automatically creates the installation control information for the load module and stores it in the Installation Control Library. The library member name is the load module name.

The Local Install option of system generation uses the control information to create an executable load module and to bind the DB2 Plan for DB2 targets, or the Remote Install option uses the control information to create a remote file.

Binder Control Cards Library

The Install option automatically creates the binder control cards information for the load module and for each of its components that are built as fully resolved executables in the Binder Control Cards Library. The library member name is the load module name, or the component name.

CA Gen generates the binder control cards information during local application installation. It is a copy of SYSLIN DD used during link edit.

Batch Binder Control Cards Library

This library contain the binder control cards for batch processing. If not specified, a temporary library is used.

Important! Do not give the Batch Binder Control Cards Library the same library name as the Binder Control Cards Library.

Generated MFS Source Library

If the application uses MFS, an MFS Source Library must be specified. This library contains MFS source members that are created during source code generation.

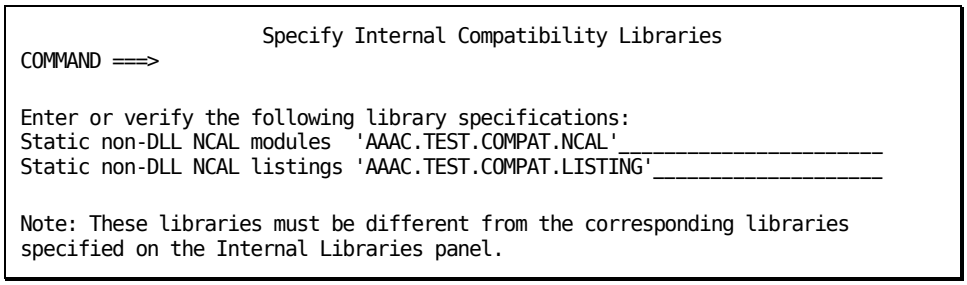
Generated Batch JCL Library

This library contains the generated JCL for the batch procedures.

Compile Listings Library

The Listings Library contains the listings that are created by the compiler for all but the NODLL compiled action blocks that are statically called by modules that are marked for Compatibility. Use of this library is optional, and is recommended.

The following illustration shows CA Gen internal compatibility libraries that you can specify:



Static non-DLL NCAL Modules Library

The Static non-DLL NCAL Modules Library stores only the compiled load modules that are created by CA Gen local install process for modules that are produced by the NODLL compile of action blocks statically linked into Compatibility modules. The linkage editor creates these members using the NCAL (no-call option). Therefore, they contain unresolved external references and are not executable. The library member name is the static action block name.

Important! Do not give the Static non-DLL NCAL Modules Library the same name as the NCAL Load Modules Library or the Executable Load Modules Library.

Static non-DLL NCAL Listings Library

The Static non-DLL NCAL Listings Library contains only the listings that are created by the NODLL compile of action blocks statically linked into modules that are marked for Compatibility. Use of this library is optional, and is recommended.

More information:

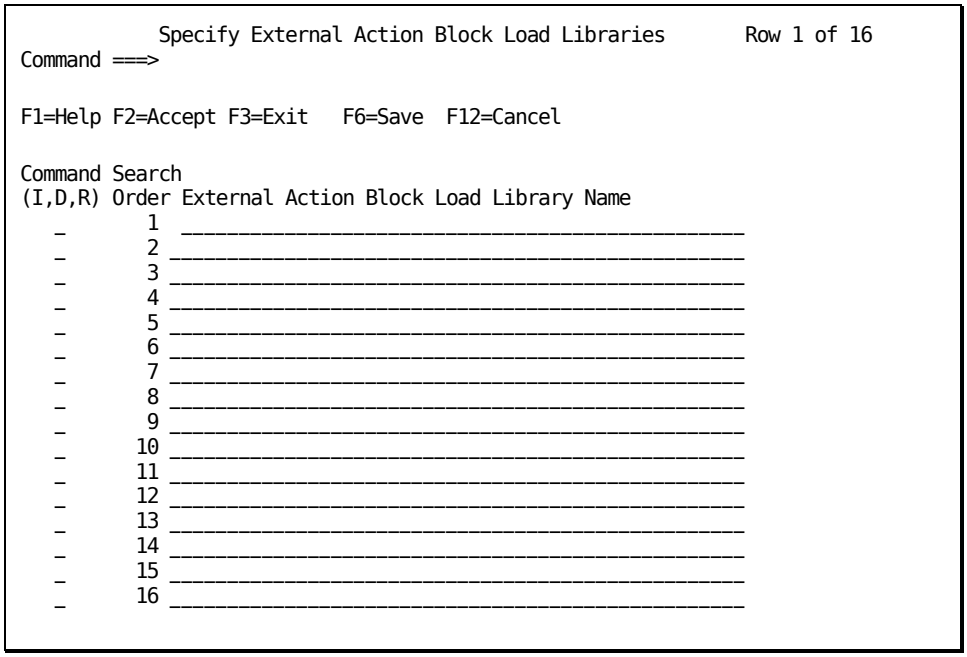
System Generation and Installation (see page 117)

External Libraries

The external libraries store objects, load modules, and DBRMs, external to CA Gen that CA Gen generated load module reference, including the following:

- External Action Block Load Libraries
- External Action Block DBRM Libraries (for target DBMS DB2)
- External System Load Libraries
- External Compatibility Libraries

CA Gen allows you to specify up to 16 external libraries in each category. A separate panel is used for each category. The panels are identical except for their titles and a note included in the External Compatibility Library panel. The following screen shows where to specify External Libraries:



When CA Gen generates and compiles a load module, it creates installation control information that includes the names of external libraries that are specified for the business system. When CA Gen executes the link edit that creates the executable load module, it resolves all external references. The link-editor searches the external libraries for the members the load module references. The External Compatibility Library names must be different from the External Action Block Load Library names as they contain different compiled versions of the same External Action Block (the same member name). The libraries are searched in the order in which they are specified on the panels.

You are responsible for creating the external libraries before CA Gen uses them. CA Gen does not create or allocate the libraries. Your site determines the characteristics of the external libraries. The external libraries are concatenated with other load libraries and DB2 DBRM libraries when a load module is installed, link edited, and bound.

To avoid concatenation problems, the DCB characteristics for all load libraries that are used during construction is the same and the DCB characteristics for all DB2 DBRM libraries are the same.

More information:

Library Concatenation for Link edit (see page 100)
Library Concatenation for DB2 Bind (see page 101)

External Action Block Load Libraries

The External Action Block Load Libraries store the load modules for external action blocks except for the modules that are produced by the NODLL compile of external action blocks statically linked into Compatibility modules. External action blocks are programs that are created outside CA Gen that CA Gen-generated programs use.

Important! Do not include the same libraries in the External Action Block Load Libraries and External Compatibility Libraries concatenation. These libraries are mutually exclusive.

External Action Block DBRM Libraries

The External Action Block DBRM Libraries store the DBRMs associated with external action blocks.

External System Load Libraries

The External System Load Libraries are optional libraries to use with other load libraries CA Gen uses. They contain external system load modules run a specific CA Gen-generated application. For example, the external system load libraries could store security subroutines that are required by a specific application. External system load libraries also store user-written termination exits.

Note: For more information, see the *User Exit Reference Guide.*

The External System Load Libraries must not contain External Action Blocks that are statically linked into Compatibility modules. These EABs must be in the External Compatibility Libraries.

CA Gen does not generate the external system load modules, and they are not part of the standard environment that is defined when installing CA Gen.

The External System Load Libraries are different than the System Load Libraries defined when installing CA Gen. The System Load Libraries are product libraries run CA Gen generated applications. The system load libraries typically contain the runtime components for products such as COBOL, PL/I, DB2, IMS, CICS, and TSO.

External Compatibility Libraries

The External Compatibility Libraries store the load modules for external action blocks that are statically linked into Compatibility modules. These external action blocks must be compiled using the NODLL compiler option. If the same external action block is used by a CA Gen generated program that is not marked for Compatibility the external action block must also be compiled with the DLL compiler option and placed in an External Action Block Load Library.

Important! Do not include the same libraries in the External Compatibility Libraries and External Action Block Load Libraries concatenation. These libraries are mutually exclusive.

Library Concatenation for Link edit

CA Gen uses the following library concatenation sequence to link-edit CA Gen-generated applications except for those marked for Compatibility:

APPLOAD

1. Internal NCAL Load Library for Business System
2. Referential Integrity Trigger Load Library
3. Internal NCAL Load Library for Business Systems owning foreign action blocks that are referenced in the load module
4. External Action Block Load Libraries for the Business System
5. External Action Block Load Libraries for the Business Systems owning foreign action blocks that are referenced in the load module

SYSLIB

1. System Load Libraries
2. External System Load Libraries for the Business System
3. External System Load Libraries for the Business Systems owning foreign action blocks that are referenced in the load module
4. CA Gen Load Library
5. External Action Block Load Libraries for the Business System
6. External Action Block Load Libraries for the Business Systems owning foreign action blocks that are referenced in the load module

Important! Since the External libraries are still in SYSLIB concatenation, if using the z/OS Library feature, ensure that none of the External System or External Action Block libraries contains an NCAL module named same as the named zLIB action blocks. Otherwise these zLIB action blocks are statically included to the caller during link edit through auto-call.

CA Gen uses the following library concatenation sequence to link-edit CA Gen-generated applications marked for Compatibility and that include statically linked Action Blocks. The Process modules marked for Compatibility option was selected when building these applications:

APPLOAD

1. Static non-DLL NCAL Modules Library for Business System
2. Referential Integrity Trigger Compiled non-DLL Load Modules Library
3. Static non-DLL NCAL Modules Library for Business Systems owning foreign action blocks that are referenced in the load module
4. External Compatibility Action Block Load Libraries for the Business System
5. External Compatibility Action Block Load Libraries for the Business Systems owning foreign action blocks that are referenced in the load module

Note: Action Blocks that are not statically linked are built as Compatibility or DLLs.

SYSLIB

1. System Load Libraries
2. External System Load Libraries for the Business System
3. External System Load Libraries for the Business Systems owning foreign action blocks that are referenced in the load module
4. CA Gen Load Library
5. External Compatibility Action Block Load Libraries for the Business System
6. External Compatibility Action Block Load Libraries for the Business Systems owning foreign action blocks that are referenced in the load module

Library Concatenation for DB2 Bind

CA Gen uses the following library concatenation sequence for the DB2 bind of CA Gen-generated applications:

1. CA Gen DB2 DBRM Library
2. Internal DB2 DBRM Library for the Business System
3. Referential Integrity DB2 DBRM Library
4. Internal DB2 DBRM Library for Business Systems owning foreign action blocks that are referenced in the load module
5. External Action Block DB2 DBRM Libraries for the Business System
6. External Action Block DB2 DBRM Libraries for the Business System owning foreign action blocks that are referenced in the load module

Display Environment and Library Panels

Follow these steps:

1. Display the Application System Construction panel.
2. Type the Host Encyclopedia name of the model.
3. Type the Host Encyclopedia name of the business system and press Enter.

   CA Gen displays the Application System Construction Menu.CA Gen automatically retrieves the construction library and target environment specifications that are stored in the Host Encyclopedia for the selected business system. If no environment or library information was stored for the selected business system, CA Gen displays the following message:

   ```
   No target environment was previously saved for this business system.
   ```

4. Type 5 for Target environment and construction libraries and press Enter.

```
                Specify Target Environment and Construction Libraries
COMMAND ===>
Model name  . . . :  Test model
Business system   :  Test system

Specify one of the options below, then press enter.
_ 1.     Specify Target Environment
  2.    Specify Construction Libraries
  3.        Restore Target Environment and Construction Libraries
       specifications from the encyclopedia.
```

Transactions Under IMS or CICS

Note: For information about configuring the IMS or CICS environments to implement CA Gen applications, see the *z/OS Installation Guide*.

Specifying Internal Libraries

Follow these steps:

1. Display the Application System Construction menu.
2. Type 5 for Specify target environment and construction libraries. You can also position the cursor opposite the option using the Tab key or the cursor keys. Press Enter.
3. Type 2 for Specify Construction Libraries. You can also position the cursor opposite the option using the Tab key or the cursor keys. Press Enter.
4. Select 1 to specify internal libraries or 5 to specify internal compatibility libraries. Press Enter.

5. Use the Tab key to move the cursor to each field and type the entry. Use fully-qualified data set names and enter the library names in single quotes. Specify all required libraries before exiting.
6. Select a command to exit the panel:

   Accept

   Uses the values only for this CA Gen session. CA Gen displays the Specify Construction Libraries and one of the following messages:

   ```
   Temporary override of internal libraries accepted.
   Temporary override of compatibility libraries accepted.
   ```

   Exit

   Leaves the screen without saving. CA Gen displays the Specify Construction Libraries panel and the following message:

   ```
   User exited from function in progress.
   ```

   Save

   Updates the relevant internal libraries in the encyclopedia for this business system. CA Gen displays the Specify Construction Libraries panel one of the following messages:

   ```
   Internal libraries saved on encyclopedia.
   Compatibility libraries saved on encyclopedia.
   ```

   Cancel

   Stops library name entry and exits without saving. CA Gen displays the Specify Construction Libraries panel and the following message:

   ```
   User canceled previous screen display.
   ```

Specifying External Libraries

Follow these steps:

1. Display the Application System Construction menu.
2. Type 5 for Specify target environment and construction libraries and press Enter.
3. Type 2 for Construction Libraries and press Enter.
4. Select one of the External Library options:

   2

   Specify external action block load libraries.

   3

   Specify external action block DB2 DBRM libraries.

4

Specify external system load libraries.

6

Specify external compatibility libraries.

5. Press Enter

   The selected External Library panel displays with the current specifications for the business system.

6. Type the required entry or entries.

   Use fully-qualified data set names. Enter library names in single quotes. Specify all required libraries before exiting.

7. Select a command to exit the panel:

   Accept

   Uses the values only for this CA Gen session. CA Gen displays the Specify Construction Libraries and one of the following messages:

   ```
   Temporary override of external action block load libraries accepted.
   Temporary override of external action block DBRM libraries accepted.
   Temporary override of external system load libraries accepted.
   Temporary override of compatibility external AB load libraries accepted.
   ```

   Exit

   Leaves the screen without saving. CA Gen displays the Specify Construction Libraries panel and the following message:

   ```
   User exited from function in progress.
   ```

   Save

   Updates the relevant internal libraries in the encyclopedia for this business system. CA Gen displays the Specify Construction Libraries panel and one of the following messages:

   ```
   External action block load libraries saved on encyclopedia.
   External action block DBRM libraries saved on encyclopedia.
   External system load libraries saved on encyclopedia.
   Compatibility external AB load libraries saved on encyclopedia.
   ```

   Cancel

   Stops library name entry and exits without saving. CA Gen displays the Specify Construction Libraries panel and the following message:

   ```
   User canceled previous screen display.
   ```

Restoring the Target Environment and Construction Libraries

To restore the the target environment and construction libraries from the Host Encyclopedia, use the Target Environment and Construction Libraries panel.

Follow these steps:

1. Display the Target Environment and Construction Libraries menu.
2. Type 3 for Restore Target Environment and Construction Libraries Specifications from the encyclopedia and press Enter.

   CA Gen retrieves the environment and library specifications and displays the message:
   `Target environment and libraries retrieved.`

Special Considerations for large Common Format Buffers

For CA Gen Release 8.5, the amount of customer data that can be sent between CA Gen Client and Servers using Common Format Buffer (CFB) based cooperative flows has increased. The previous CFB limit was 32K. The new limit is 16.7 MB.

Installing the modules larger than 32K require modifications for resources that are used during Compile and Link.

- Internal libraries that are used for source code, listing, ncal loadmodules, executable load modules are increased in size. You are responsible for monitoring the space utilization of the data sets using the ISPF utility. If system generation terminates abnormally with a D37 or B37 system abend, it is likely that the data sets ran out of space or directory blocks.
- Temporary work space that is allocated by parameters TICASUTP and TICASUTS need to be evaluated and appropriately increased. These parameters can be found in the PARMLIB member TIUHE for the Host Encyclopedia and member TIUIT for the z/OS IT and can be modified as necessary.

Chapter 7: Load Module Packaging in HE

Specifying Components

Specify which components comprise the load modules for a business system before generating the executable components of your application system. The specification of the components is named load module packaging.

The packaging itself does not create the load modules; it simply identifies what makes up a load module. A load module is created when you generate and install it.

These methods of packaging are available on the Host:

- Online
- Batch if the target database is DB2

Note: Cooperative packaging for client/server applications and z/OS Library packaging cannot be done using Host Construction. To generate a server on the Host, specify its packaging using the workstation toolset or client/server encyclopedia and upload the packaging to the Host Encyclopedia. To generate a z/OS Library on the Host, specify its packaging using the workstation toolset and upload the packaging to the Host Encyclopedia.

More information:

System Generation and Installation (see page 117)

Online Packaging

To package for online processing, CA Gen displays a list of the online procedure step names for the selected business system. You supply the names for these components:

- Load Modules
- Clear Screen Transaction Codes
- Dialog Flow Transaction Codes

Batch Packaging for DB2

To package for batch processing, CA Gen displays a list of the batch procedure step names for the selected business system. You supply the names for these components:

- Load Modules
- Job names
- Job Stepnames

A procedure equates to a batch job and a procedure step to a job step.

Also specify which DB2 attach facility you want. The three choices are DSN command, DLIBATCH, and IMS BMP. If you select DLIBATCH or IMS BMP, specify the name of the Program Specification Block (PSB).

Prerequisites

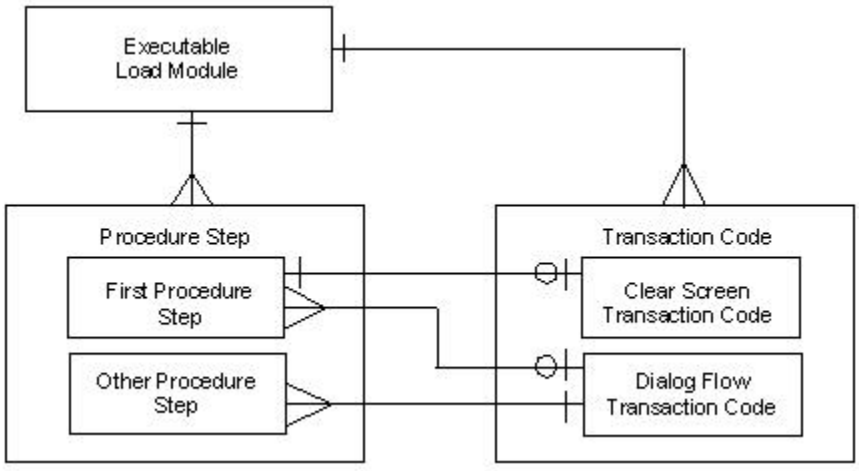
Before you can use the packaging functions, you complete the diagrams in the model and upload the model to the Host Encyclopedia.

The way that you package your load modules depends on how you established the dialog flow in the Dialog Flow Diagram (DLG). The portion of the model you intend to generate or install must have at least one procedure with at least one procedure step and must be consistent.

Packaging information that is entered in the Workstation Construction toolset or client/server encyclopedia is always uploaded. A model does not be checked in to be packaged using the Host Encyclopedia Construction Tools. However, when packaged procedure steps are checked out to a workstation, certain types of packaging changes are restricted to prevent model corruptions.

Online Packaging Rules

CA Gen lets you package online load modules to meet your system performance and user productivity goals. The factors such as the size of procedure steps and the frequency of use influences packaging decisions.



During load module packaging, identify the transaction codes and load module that is associated with each procedure step. During program execution, the TP monitor uses the transaction code to determine the load module to execute. The TP monitor through a transaction code invokes every procedure step. Procedure steps cannot call each other directly.

The packaging definition provides CA Gen with the transaction code information it cross-reference procedure steps and load modules with transaction codes. CA Gen does not install these transaction codes into IMS or CICS. This must be done outside CA Gen, before the load module is executed.

CA Gen uses two types of transaction codes (trancodes):

- Clear Screen Transaction Codes
- Dialog Flow Transaction Codes

Clear Screen Transaction Codes

A clear screen transaction code allows the user to invoke a transaction, that is a load module, from a clear screen. A clear screen is a screen that contains no user input other than the transaction code and data that is passed with the code. The clear screen transaction code also invokes the transaction from another program, for example, using NEXTTRAN, IMS message switch, or CICS START command.

In CA Gen, a clear screen transaction code can only invoke the first step in a procedure. To invoke a first procedure step from a clear screen or another program, assign a unique clear screen transaction code to the procedure step. This allows the TP monitor and Dialog Manager to identify correctly the procedure step to execute.

CA Gen assumes a procedure step that is not a first procedure step can be invoked only through a dialog flow, not from a clear screen or another program. Therefore, you cannot assign a clear screen transaction code to a procedure step that is not a first procedure step.

More information:

System Generation and Installation (see page 117)

Dialog Flow Transaction Codes

The Dialog Manager uses a dialog flow transaction code to manage flows, that is links and transfers, between procedure steps. During load module packaging, a dialog flow transaction code must be assigned to every procedure step that is flowed to from another procedure step. The dialog flows are identified in the Dialog Flow Diagram.

Every procedure step that is not a first procedure step must have a dialog flow transaction code. This type of procedure step is always accessed (flowed to) from another procedure step. It cannot be invoked from a clear screen or another program.

First procedure steps require a dialog flow transaction code if they can be flowed to from another procedure step. The clear screen and dialog flow transaction code can be the same. A first procedure step that is accessed only from a clear screen does not require a dialog flow transaction code.

Dialog flow transaction codes do not have to be unique. You can assign the same code to more than one procedure step. However, a dialog flow transaction code must uniquely identify a load module. This means that all procedure steps associated with the same dialog flow transaction code must be packaged in the same load module.

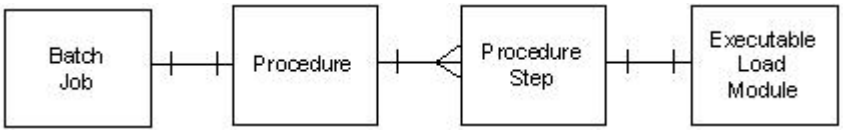
The TP monitor uses the dialog flow transaction code to determine which load module to execute. However, in this case, the Dialog Manager that is based on information that is passed on the message queue determines the procedure step. Therefore, the dialog flow transaction code must uniquely identify the load module, but does not be associated with only one procedure step in the load module.

Note: The dialog flow transaction code is not used to determine the load module to execute on a return from a link. The transaction code that is used on a return from link is the transaction code that was active when the link was initiated.

Batch Packaging Rules

When packaging batch applications, think of each procedure as a batch job, and each procedure step as a job step. The following rules apply to batch packaging:

- Only one procedure step is packaged into a load module.
- All procedure steps in a procedure must be packaged into the same batch job.
- Procedure steps from different procedures cannot be packaged into the same batch job.
- Procedure step (job step) names must be unique within a procedure (job).
- Job names and load module names must be unique within the model.



Repackaging Load Modules

The procedure for repackaging load modules is the same as the procedure for packaging load modules. The load module packaging function lets you repackage load modules without changing the generated procedure steps, action blocks, and screens. The change occurs when you generate the Dialog Managers and install the load modules.

The chapters on system generation and installation and intelligent regeneration discuss the rules that apply to system generation based on repackaging a load module. Note, however, that changing the packaging of an online load module affects the Dialog Managers of all online load modules that are affected by the change.

For example, if a procedure step is moved from one online load module to another, the change affects:

- The load module that contained the procedure step before the change.
- The load module that will contain the procedure step after the change.
- The load modules that contain procedure steps that transferred or linked to the procedure step before the change.
- The load modules that contain procedure steps which transfer or link to the procedure step after the change.

Commands in the Packaging Panels

Type the following commands on the COMMAND line during packaging: SORT, FIND, CANCEL, FINDP, and SAVE.

When the packaging panel is displayed, the procedure steps are sorted by load module name. If no load modules have been defined, the procedure steps are listed in the order in which they appear on the Dialog Flow Diagram. The packaging commands are available to assist you in finding particular load modules or procedure steps:

SORT

Sorts procedure steps by load module name and displays them. On the command line, type:
`SORT`

FIND

Moves to the first procedure step for a specified load module. One the command line, type:
`FIND <Load Module Name>`

CANCEL

Cancels the updates that are made to the load module packaging. On the command line, type:
`CANCEL`

FINDP

Locates a specified procedure step. On the command line, type:
`FINDP <Procedure Step Name>`

SAVE

Saves the changes that are made to the load module packaging and updates the Host Encyclopedia. Changes are lost when you exit the panel unless you press the Save function key or type SAVE on the COMMAND line and press Enter.

Scrolling

A business system often contains more procedure steps than can display on a single packaging panel. Use the TSO/ISPF scroll commands to scroll through the procedure steps. The defaults are PF8 to scroll forward and PF7 to scroll backward.

Packaging Online Load Modules

Follow these steps:

1. Display the Application System Construction panel.
2. Type the Host Encyclopedia name of the model.

   You can also use the Tab key to position the cursor at the required field and press the Prompt key for a list of valid choices. Select a model from the list by typing any character except a period (.) or a space character next to one name and press Enter.
3. Type the Host Encyclopedia name of the business system.

   You can also use the Tab key to position the cursor at the required field and press the Prompt key for a list of valid choices. Select a business system from the list by typing any character except a period (.) or a space character next to one name and press Enter.
4. Press Enter.

   The Application System Construction menu displays.
5. Type 3 for Define load module packaging and press Enter.

   The Select Load Module Packaging Environment panel displays.
6. Type 1 for online packaging and press Enter.

   `Enter packaging updates.`

```
                        Package Online Load Module              Row 1 of 10
COMMAND ===> _                                                SCROLL ===> PAGE

Model name  . . : TEST MODEL 1
Business system : APPLICATION_HELP

Enter packaging updates.  To save, press F6.  To cancel, press F12.

 Load                                            Clear Screen  Dialog Flow
Module          Procedure Step Name               Trancode      Trancode

HSDHELP_   SCREEN_DESCRIPTION                      HSDA____      HSDA____
HSDHELP_   FIELD_DESCRIPTION                       HSDB____      HSDB____
HSDHELP_   PERMITTED_VALUES                        HSDC____      HSDC____

****************************** Bottom of data ********************************
```

   If necessary, use the command functions to locate and display the procedure step name. You can also scroll the procedure step names using the TSO/ISPF scroll commands. The default key functions are F8 to scroll forward and F7 to scroll back.
7. Type the load module name in the Load Module column.

   Limits: eight alphanumeric characters. The first character must be a letter.

8. Type a clear screen transaction code next to the load module name if the procedure step is a first procedure step, and can be invoked from a clear screen or directly by another program.

   The clear screen transaction code for IMS and TSO is a maximum of eight alphanumeric characters. The CICS clear screen transaction code is a maximum of four alphanumeric characters. The first character must be a letter for all systems.

9. Type the dialog flow transaction code next to the load module name. All procedure steps that are linked or transferred to require a dialog flow transaction code.

   The dialog flow transaction code for IMS and TSO is a maximum of eight alphanumeric characters. The CICS dialog flow transaction code is a maximum of four alphanumeric characters. The first character must be a letter for all systems.

10. Repeat the online packaging steps as needed.

    If CA Gen detects an error in the input, a message displays under the COMMAND line and CA Gen puts an X between the load module and procedure step names on the lines which are in error.

11. Click Save to keep the updates or Cancel to cancel the updates.

    The Select Load Module Packaging Environment panel displays.

Packaging Batch Load Modules

To package batch load modules

1. Display the Application System Construction panel.
2. Type the Host Encyclopedia name of the model.

   You can also use the Tab key to position the cursor at the required field and press the Prompt key for a list of valid choices. Select a model from the list by typing any character except a period (.) or a space character next to one name and press Enter.

3. Type the Host Encyclopedia name of the business system.

   You can also use the Tab key to position the cursor at the required field and press the Prompt key for a list of valid choices. Select a business system from the list by typing any character except a period (.) or a space character next to one name and press Enter.

4. Press Enter.

   The Application System Construction menu displays.

   CA Gen automatically retrieves the Construction library and target environment specifications stored in the Host Encyclopedia for the selected business system.

5. Type 3 for Define load module packaging. Press Enter.
6. Type 2 for Batch packaging. Press Enter.

```
                        Package Batch Load Module          Row 1 to 11 of 13
COMMAND ===>                                                SCROLL ===> CSR

Model name  . . : TEST MODEL
Business system : CORPORATE_MANAGEMENT

Enter packaging updates. To save, press F6. To cancel, press F12.

           Job      Load Module                              DB2          IMS
Jobname    Stepname                 Procedure Step Name     Attach       PSBname
BJOB0001   BSTEP01  P489            MENU                    DB2_DSN_     _____
BJOB0009   BSTEP01  P482            DIVISION_MAINT          DB2_DSN_     _____
BJOB0010   BSTEP01  P481            COST_CTR_MAINT          DB2_DSN_     _____
BJOB0013   BSTEP01  P480            EMPLOYEE_LIST           DB2_DSN_     _____
```

Use the command functions to locate and display the required procedure step name. You can also scroll the procedure step names using the TSO/ISPF scroll commands. The default key functions are F8 to scroll forward and F7 to scroll back.

7. Type the job name in the Jobname column.

   This is the name of the job used in the JCL for the batch job. The job name is a maximum of eight alphanumeric characters. The first character must be a letter. Standard JCL naming conventions apply.

8. Type the name of the job step in the Job Stepname column.

   This is the name that is used in the JCL for the step name. Standard JCL naming conventions apply.

9. Type the load module name in the Load Module column.

   The load module name is a maximum of eight alphanumeric characters. The first character must be a letter.

10. If your target environment is DB2, type the name of the DB2 attach facility in the DB2 Attach column.

    The three choices are DSN command, DLI (for DLIBATCH), or BMP (for IMS BMP). If you leave the DB2 Attach field blank, a default of DSN command is used.

11. Repeat these steps as needed for each procedure step.

    If CA Gen detects an error in the input, a message displays under the COMMAND line. An X displays between the load module and procedure step names on lines that are in error.

12. Click Save to keep the updates or Cancel to cancel the updates.

    The Select Load Module Packaging Environment panel displays.

Chapter 8: System Generation and Installation

Installation in the HE Environment

Discusses the system generation and installation functions that create the source code and the application system executable. However, the information about the generation and installation tasks described here do not apply to application components with a dynamically link packaging property set, or derived, to Compatibility unless the option Process modules marked for Compatibility is also selected. When the option Process modules marked for Compatibility are not selected, these tasks are bypassed.

The z/OS Library cannot have Compatibility modules. Since Process modules marked for Compatibility option is not applicable for the z/OS Library, z/OS Library generation and installation do not use this option.

Source Code Generation

The two phases in application system construction are:

- Source code generation
- Installation

The system generation and installation functions create an application system's source code and the executable components. During the source code generation, CA Gen generates COBOL source code, MFS source for online IMS systems, and JCL for batch jobs. The COBOL source must be generated for all required components of a load module before the load module can be installed.

The generated source modules are stored in the business system's Internal Source Code library.

Important! Do not modify the generated source code. Always change the related action diagrams, views, and screens, and regenerate the source. The results after installation are unpredictable if you modify the generated source code.

You can generate systems in the foreground or can generate a batch job to run in the background.

The Installation Control Generator creates an Install Control file that controls the installation procedures. Each load module is installed individually from a single Install Control file.

More information:

Specify Target Environment for HE (see page 77)

Installation on z/OS Platform

Select local to install the application for the z/OS operating system under which the generators are running. Select remote to install the application on another z/OS platform.

Local Installations

For the local installation, CA Gen compiles the generated source and link-edits the load modules. The binder control cards that are used in link edit are stored in Binder Control Cards library.

CA Gen local installation link-edits the dynamic action blocks including z/OS Library modules for batch processing in addition to regular link edit if Executable Batch Dynamic Action Block library is specified for the Business System.

For z/OS Libraries, the online version of z/OS Library DLL is populated in the regular Executable Load Modules Library. The batch version of z/OS Library is saved in the Executable Batch Dynamic Action Block library if specified. DB2 attachment type for batch z/OS Library DLLs is as specified by the TIRDBATT parameter, defaulting to DSN.

For dynamic action blocks, if the Executable Batch Dynamic Action Block library is specified, it contain the batch versions of dynamic action blocks. The following rules apply if this library is present.

- Dynamic action block that is linked via the caller batch load module:
  - Batch version of the dynamic action block is populated in Executable Batch Dynamic Action Block library.
- Dynamic action block that is linked via the caller non-batch load module:
  - Non-batch version of the dynamic action block is populated in the regular Executable Load Modules Library.
  - Batch version of dynamic action block is prepared and populated in the Executable Batch Dynamic Action Block library.
- If the batch library is not specified, then the Executable Load Modules Library contain the most recently installed dynamic action block, which have batch or non-batch runtimes. It is recommended to specify the Executable Batch Dynamic Action Block library in this case.

Important! Do not give the Executable Batch Dynamic Action Block Library the same library name as the Executable Load Modules Library.

If your environment permits, CA Gen also executes the DB2 bind that is required for programs that use DB2 databases.

Remote Installations

For the remote installation, CA Gen copies the generated source into remote files to transfer to the target environment and install using the Implementation Toolset. The remote files are also named Implementation Packages (IPs).

You can install systems in the foreground or background mode.

More information:

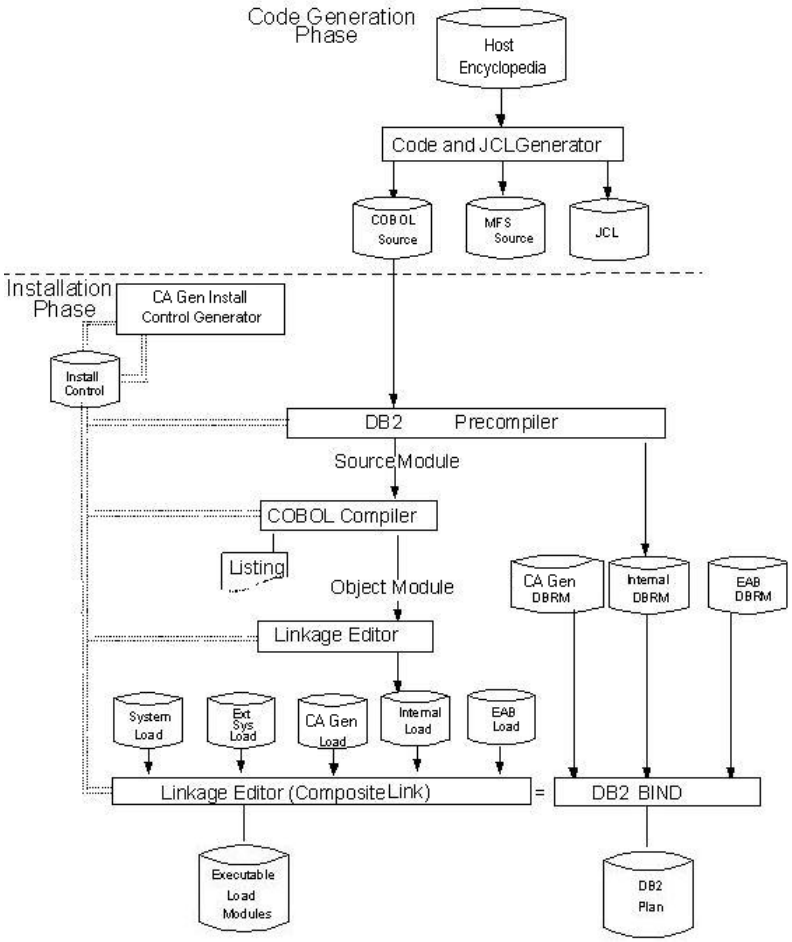
Specify Target Environment for HE (see page 77)

DB2 Bind Options

When using DB2, CA Gen offers a choice of how to bind data for your application. You bind a plan using Database Request Modules (DBRMs), packages, or a combination of DBRMs and packages.

Package bind binds each DBRM into a single package. When the application changes, you can rebind the package for that program's DBRM without having to rebind the application plan.

The following illustration shows the general flow of system generation and installation when DB2 is the target DBMS and you do not use package bind, and the activities and libraries that store the generated and installed components.



Prerequisite Workstation Tasks

Complete certain tasks on the workstation and on the host before generating a system. Although create the DB2 tables before installing a system, you can generate a system without creating the DB2 tables.

Before generating a system, complete these workstation tasks:

- Complete all diagrams for the part of the model to generate.
- Perform Transformation of the Data Model (DM) using the Design toolset to create the Data Structure List and Data Store List.
- Package the server load modules for client/server applications.
- Perform a Consistency Check to ensure that the Data Model and the Technical Design are consistent.
- Upload the model to the Host Encyclopedia.

Complete All Diagrams

The Host Encyclopedia is the main source of information for system generation. The system generation functions use information from the diagrams that are created during Analysis and Design.

Before generating and compiling code successfully, complete the diagrams for the part of the model you are generating. The system generation functions use the following diagrams that create the Host Encyclopedia information:

- Data Modeling Diagram
- Dialog Flow Diagrams
- Procedure Action Diagrams
- Technical Design, Data Structure List, and Data Store List
- Screen Design Diagrams

Perform Transformation on the Data Model

CA Gen allows you to generate code before you create the database tables. However, first transform the Data Model (DM) into a Technical Design using the Design toolset. The Transformation process creates the definitions for the tables that Construction generates. This information is stored in the Host Encyclopedia.

The system generation tools use the information in the Host Encyclopedia to generate code that references the tables as if they already existed. CA Gen also generates SQL DECLARE TABLE ... statements that are embedded in the COBOL code. For DB2, generation does not depend on a DB2 DCLGEN (Declare Table Generation) having run. To run a DCLGEN, the tables must exist.

During Design, you can transform the complete Data Model or a portion of it. Plan a partial Transformation carefully to avoid problems during system generation. For example, transform all entities that are referenced by the business system you intend to generate.

Be sure to transform entities that are not directly referenced by the action diagrams for the business system, but referenced by the generated code. For example, to implement referential integrity, the generated code includes cascade delete logic for mandatory relationships. This logic reference entities that are not directly referenced by the action diagrams. If these entities are not transformed, system generation fails.

Similar problems occur if you change the Data Model after transforming it. When you change the Data Model, perform a Transformation or a Retransformation. Retransformation processes only the part of the model that changed. This leaves untouched the parts of the modified Data Store List and Data Structure List that did not change. The impact of changes to the Data Model is not always obvious, and a partial Retransformation cause inconsistencies. The DSD consistency check detects these inconsistencies, and is used after partial Retransformation to verify that there are no errors.

The host When Changed Report can help in planning a partial Transformation or Retransformation.

Note: For more information, see the *Host Encyclopedia User Guide*.

Package the Server Load Modules

Since there is no load module packaging tool for server load modules on the Host, Server load module packaging must be complete using the workstation toolset or client/server encyclopedia. After packaging is complete, some of the packaging information in the Server Target Environment panel can be modified.

Package the z/OS Library Load Modules

Since there is no load module packaging tool for z/OS Library server load modules on the Host, use the workstation toolset to complete the z/OS Library packaging. After packaging completes, you can modify some of the packaging information in the Server Target Environment panel.

Perform Workstation Consistency Check

The system generation functions require consistency in the part of the model you are generating, such as the procedure steps or action blocks. For example, for a load module to generate successfully, all procedure steps and action blocks in the load module must be consistent, including:

- Entities, functions, and processes associated with the procedure steps and action blocks
- Dialog Flow Diagram, Screen Design Diagrams, and Procedure Action Diagrams for the procedure steps and action blocks
- The portion of the Technical Design that defines the components of the data model that is referenced by the procedure steps and action diagrams

The Analysis and Design Toolsets include a Consistency Check option to check a model before uploading it to the Host Encyclopedia.

Upload Model to Host Encyclopedia

The model must be uploaded to the Host Encyclopedia before using the host tools to generate or install.

Note: For more information, see the online help in the workstation toolsets.

Prerequisite Host Tasks

Before generating a system, complete these tasks on the host:

- Run a host Consistency Check to verify that the load module components are consistent.
- Generate Referential Integrity trigger modules for the model.
- Package the load modules for online and batch applications.
- Specify a collection ID for members that are to bind as packages and provide their bind parameters. You can do this for the entire model, the business system, or for all RI Triggers.
- Specify the target environment and the construction libraries in which CA Gen stores the generated objects (system generation libraries).
- Specify all required external libraries. Be sure to include external action block libraries.
- If the load module uses external action blocks that access tables, resolve the external action blocks, and identify the DB2 DBRMs associated with the external action blocks.

- Before installation, perform the database generation to create the tables and indexes for DB2 related table spaces and index spaces, for the entities that are referenced by the load module.
- Install the load module to resolve all external references and create the executable load module.

Perform Host Consistency Check

After uploading the model, perform a host Consistency Check to ensure model consistency for generation and installation. The host Consistency Check uses the same rule base as the workstation.

Generate Referential Integrity Triggers

Referential Integrity (RI) triggers are CA Gen generated action blocks that implement the cascade delete logic that is required to enforce data modeling constraints that are specified in the Data Model.

More information:

Referential Integrity (see page 49)

Package Load Modules for Online and Batch Applications

Packaging specifies the procedures that comprise a load module. Perform this step before compiling source code.

More information:

Load Module Packaging in HE (see page 107)

Specify DB2 Collection ID and Set Parameters

When the target DBMS is DB2, specifying a DB2 collection ID causes a package bind to follow the compile. Perform this step before generating source code. When you do not specify a collection ID, the default is to include the DBRMs in the application plan.

More information:

DB2 Package Bind Options (see page 65)

Specify Target Environment and Construction Libraries

Before using the system generation functions, specify the target environment and the construction libraries.

More information:

Specify Target Environment for HE (see page 77)

Resolve External Action Blocks for DB2

When a load module uses external action blocks that access DB2 tables, the DB2 DBRMs associated with the external action blocks must be identified before installing the load module. Identifying the DB2 DBRMs is called resolving an external action block.

More information:

External Action Blocks (see page 207)

Generate Database

Source code can be generated without creating the tables. However, generate the database before performing installation.

More information:

Database Generation in HE (see page 37)

Generate the Load Module Components

Generate the source code for the load module components using the system generation tools.

Install the Load Module

Use the installation tools to compile, link edit, and bind the load module. This resolves all external references and creates the executable load module.

Load Module Components

Construction creates a system's executable load modules. Each load module contains certain components depending on if it is an online or batch load module. The following lists show the components each type of load module can contain. CA Gen generates the components marked with an asterisk (*). For external action blocks, CA Gen only generates a stub that you can modify.

Online Load Module

*Dialog Manager

Profile Manager

*Screen Manager – one or more

*Procedure Steps - one or more

*Action Blocks

*External Action Blocks – optional

*Referential Integrity Triggers

CA Gen User Exits

CA Gen Runtime Routines

Batch Load Module

*Batch Manager

Profile Manager

*Procedure Steps - one or more

*Action Blocks

External Action Blocks – optional

*Referential Integrity Triggers

CA Gen User Exits

CA Gen Runtime Routines

Server Load Module

*Server Manager

Procedure Steps - one or more

*Action Blocks

*External Action Blocks – optional

*Referential Integrity Triggers

CA Gen User Exits

CA Gen Runtime Routines

Most CA Gen runtime routines are DLLs and only a few are statically linked with the executable load modules.

Note: For more information about CA Gen DLLs, see the *z/OS Installation Guide*.

Dialog Manager

A Dialog Manager is generated for each online load module. The entry point for all load modules is a CA Gen-provided runtime module named TIRMAIN. It calls the Dialog Manager that controls the dialog flow, link, and transfer, between procedure steps, supports terminal input/output, and maintains execution context in a Profile Database. The Dialog Manager uses information from procedure step import and export views, screen definitions, dialog flows, and load module packaging to isolate the user-defined program logic from the execution environment.

The Dialog Manager contains cross-reference tables of procedure steps, exit states, and transaction codes for the procedure steps that are packaged in the load module. The procedure steps to which the load module procedure steps link or transfer are also cross-referenced. The Dialog Manager uses this information to select the target procedure step within its load module. If the target procedure step is in another load module, the Dialog Manager uses the information to select the transaction code that invokes the load module containing the target procedure step. CA Gen provided runtime routine TIRMSG is named to invoke the other load module.

When a load module is executed, the Dialog Manager determines the procedure step to execute and populates the import view of the procedure step from screen input, data that is passed from another procedure step, and data that is taken from the Profile Database. This allows each procedure step to reference data in its import view without considering how the data values were obtained.

Batch Manager

A Batch Manager is generated for each batch load module that has DB2 as the target DBMS. The Batch Manager functions similarly to the Dialog Manager. The Batch Manager controls the dialog flow between procedure steps. Batch processing only allows forward transfers and does not allow links. Batch processing cannot return to a procedure step after it moves to the next step.

CA Gen batch applications use the following three special files, referenced in the JCL by the Data Definition (DD) names TIRIOVF, TIRMSGF, and TIRERRF. CA Gen accesses these files through the CA Gen-provided runtime routine TIRBTCH.

TIRIOVF

The data set used to pass data, that is views and Batch Manager control information, from one procedure step to another on a dialog flow.

TIRMSGF

The Batch Manager Message File that provides a trace of procedure steps that are executed, exit states set, exit state messages that are issued, and dialog flows taken.

TIRERRF

The Batch Manager Error Message File where CA Gen writes error messages and diagnostic information if a failure occurs. The messages and diagnostic information help determine the cause of the failure.

Server Manager

CA Gen generates a Server Manager for each server load module that isolates the user from the execution environment. The Server Manager accepts the message from the client, invokes the appropriate procedure step, and manages the sending of the response back to the client.

Server-to-Server Flow

Some Distributed Processing Server (DPS) runtime environments, for example, MQSeries, Tuxedo, CICS, and IMS, support the situation where one DPS component initiates a request of another DPS component. These DPS-to-DPS flows are known as the Server-to-Server flows. For server environments that support server-to-server flows between separate server managers, the flow behaves similarly to a client-to-server flow.

Each server represents a unit of work. Commit processing occurs when the server terminates. This is true for Server-to-Server flows where the called server commits its changes before returning to the calling server. The calling server commits its own changes regardless of the status of the called server. The two servers are not part of two-phase commit processing.

Note: For DPS-to-DPS flows within the same server manager, the flow processing is a local function call.

Profile Manager

The Profile Manager, TIRPROF, is a CA Gen provided runtime routine that saves information in the following situations:

- During a link and return
- When a screen supports scrolling
- Retaining the values of hidden fields on a screen
- Processing HELP and PROMPT requests
- When restarting an application

The Profile Manager saves information in a Profile Database. The Profile Database is a temporary runtime pushdown stack that stores the export views of procedure steps. CA Gen maintains one stack per business system per user.

Note: A potential for problems can arise if you open multiple sessions within a single business system using the same User ID. You can modify the User ID user exit to return the terminal ID as the User ID used in the key to the Profile Database. This helps to avoid the contention when using the Profile Manager.

Screen Manager

CA Gen generates a Screen Manager for every online procedure step. The Dialog Manager calls the Screen Manager to perform functions such as input parsing, function key translation, input/output view mapping, scrolling, and Help/Prompt command processing. The Screen Manager calls CA Gen runtime routines to edit input and output and to write output buffers. Screen Managers are statically linked with the load module or dynamically linked when the load module executes.

MFS Source

If the target environment is IMS with a screen format type of MFS, CA Gen generates an MFS source module each time a Screen Manager is generated. Each MFS source module contains the following control blocks:

MID

Message Input Descriptor

MOD

Message Output Descriptor

FMT

Format member. This combines DIF (Device Input Format), and DOF (Device Output Format).

Note: CA Gen does not install the MFS Control Blocks. Installation is performed outside of CA Gen.

Procedure Steps and Action Blocks

CA Gen generates a source module for each procedure step and action block. If the application uses databases, the generated source code includes the SQL/DML statements to access the tables. Each generated source module includes a listing of its action diagram, its views, and related dialog flow details. In the COBOL source, this listing is at the end of the module and is cross-referenced to the generated source code to help analyze problems that result from errors in the action diagram logic. You can also generate a listing of the action diagram separately, without generating the source code.

The procedure steps in each load module are selected from a list of all the procedure steps that are defined for the business system during load module packaging. Recall that online transactions have no restrictions on the number of procedure steps you can package in each load module. Batch applications require that you package only one procedure step per single load module.

Action blocks are included in load modules when the procedure steps packaged in the load module use them, or when other action blocks use them.

Procedure steps and action blocks are statically linked with the load module or dynamically linked when the load module executes.

An other action block or action diagram in that model uses an action block that is defined in a model.

Although many business systems USE an action block, generate it only in the business system that owns it. After generation, other business systems can successfully install modules that USE the action block. The business system that owns an action block is established when it is first USEd, when it is scoped, or if it is copied with CA Gen COPY function. These activities occur on the workstation.

When an action block owned by one business system is USEd by an action block in another business system, it is referred to as a foreign action block. If a foreign action block was not generated and installed, you get an error. If the foreign action block is statically linked, the error is an unresolved reference during the installation. If the foreign action block is dynamically linked, the error is an indication that the program is not found during execution.

It is sometimes necessary to generate an action block in a business system other than the one that owns it. For example, if the owning business system has no load modules that contain the action block, it cannot be generated there.

More information:

Load Module Packaging in HE (see page 107)
List a Procedure Action Diagram (see page 159)

External Action Blocks

External action blocks provide CA Gen-generated applications with a means to access logic in a subroutine that is defined outside of CA Gen. The external action block calls the subroutine's logic or contains it.

More information:

External Action Blocks (see page 207)

Referential Integrity Triggers

Referential Integrity triggers are CA Gen generated action blocks that enforce data modeling constraints.

More information:

Referential Integrity (see page 49)

CA Gen Supplied User Exits

CA Gen provides several users exits to use with your application system. The user exit contains source code that you can modify to meet your system's specific needs.

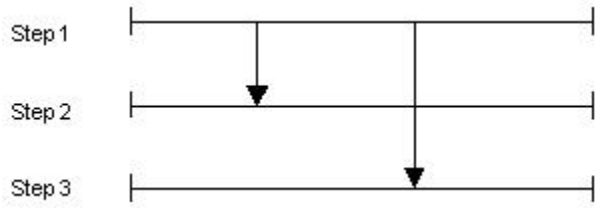
Note: For more information about user exits, see the *User Exit Reference Guide.*

Return Codes in CA Gen z/OS Batch Jobs

CA Gen implements the dialog flow type 'transfer' for z/OS batch jobs using JCL condition codes. As each procedure step completes, the Batch Manager interprets the exit state and determines if a transfer is to take place. If a transfer is to take place, the Batch Manager writes control data and passed views to the TIRIOVF file, and sets a COBOL return code. The JCL uses the return code as the condition code. If the exit state indicates no transfer, the Batch Manager sets a return code of 1000, indicating the successful end of the job.

Procedure Example

Consider the three-step procedure in the following illustration:



Under some conditions, Step 1 sets an exit state to transfer to Step 2. Under other conditions, Step 1 sets an exit state to transfer to Step 3. Step 2 does not transfer to Step 3.

Execute Step 1 and Step 2

When Step 1 sets an exit state that causes a transfer to Step 2, the Batch Manager for Step 1 sets a return code of 120 and terminates. The JCL EXEC statement for Step 2 has a conditional execution parameter of COND=(120,LT). Since 120 is not less than condition codes set, Step 2 begins execution. Since Step 2 does not transfer to another step, its Batch Manager sets a return code of 1000 when it completes. The JCL EXEC statement for Step 3 has a conditional execution parameter of COND=(130,LT). Since 130 is not less than condition codes set, the job terminates without transferring to Step 3.

Execute Step 1 and Step 3

When Step 1 sets an exit state that causes a transfer to Step 3, the Batch Manager for Step 1 sets a return code of 130 and terminates. The JCL EXEC statement for Step 2 has a conditional execution parameter of COND=(120,LT). Since 120 is less than any condition code set, Step 2 is bypassed. The JCL EXEC statement for Step 3 has a conditional execution parameter of COND=(130,LT). Since 130 is not less than any condition code set, Step 3 begins execution. Since Step 3 does not transfer to another step, its Batch Manager sets a return code of 1000 when it completes.

CA Gen Condition Code Numbering

The algorithm for computing the condition code is Step Number times 10 + 100. An increase of 10 allows users to easily insert non-CA Gen steps into JCL. The CA Gen codes all exceed 100 to prevent conflicts with return codes issued by non-CA Gen steps that usually have single digit returns codes such as 4, 8, and so on. The code 1000 indicates successful end of job to ensure that subsequent CA Gen steps would be bypassed, but would leave a wide range of available condition codes (1001-4095) for other uses.

Batch Return Code Override Exit

The Batch Return Code Override Exit allows the user to override the CA Gen defined COBOL return code in batch job steps. It is called by the Batch Manager at the end of each job step, and by program TIRIOVFI.

Note: For more information, see the *User Exit Reference Guide*.

Clear the TIRIOVF file

The first job step in each CA Gen batch job executes program TIRIOVFI that clears the TIRIOVF file that passes control data and views between procedure steps. This program sets the COBOL return code to 110. The next CA Gen job step is a procedure step and has a conditional parameter of COND=(110,LT).

Generation in the HE Environment

To initiate generation and installation tasks, use these panels:

Generate Business System panel

The Generate Business System panel lists the online load modules, server load modules, and batch jobs in a business system and the system generation functions you can perform on them. Use a single letter action code to initiate the functions for each load module. Expand a load module or batch job to initiate functions for components.

Load Module Expansion panel

CA Gen displays the Load Module Expansion panel when you select the expand option for an online, server, z/OS Library, or batch load module from the Business System Generation panel. Use this panel to view and perform generation and other functions on load module components.

Some of the functions in this panel duplicate function in the Business System Generation panel, although at a lower level of component detail. Selecting generate in the Business System Generation panel generates every component in a load module. Selecting generate in the Load Module Expansion panel generates a single component of a load module.

Batch Job Expansion panel for DB2

For batch processing, expand at the job level before the load module expansion level on the Batch Job Expansion panel to generate all components per job step.

CA Gen displays the Batch Job Expansion panel when you select the expand option for a batch job from the Generate Business System panel to view and perform generation and other functions on batch load modules.

Some of the functions in this panel duplicate function from the previous panel, but do so at a lower level of component. Selecting generate in the Generate Business System panel generates every component in every load module in the batch job. Selecting generate in the Batch Job Expansion panel generates every component in a single batch load module.

Batch Load Module Expansion panel for DB2

CA Gen displays the Batch Load Module Expansion panel when you select the expand option for a batch load module from the Batch Job Expansion panel. Use this panel to view and perform generation and other functions on batch load module components.

Some of the functions in this panel duplicate function from the Batch Job Expansion panel, although at a lower level of component detail. Selecting generate in the Batch Job Expansion panel generates every component in the batch load module. Selecting generate in the Batch Load Module Expansion panel generates a single component.

From any of these panels, you can access the Generation Options panel by typing an O or the word Options on the COMMAND line and pressing Enter. You can review the options and can update them.

To open the Target Environment and Construction Libraries panel, type an E or the word Environment on the COMMAND line and press Enter to verify or change the environment parameters or libraries without returning to the Application System Construction menu.

More information:

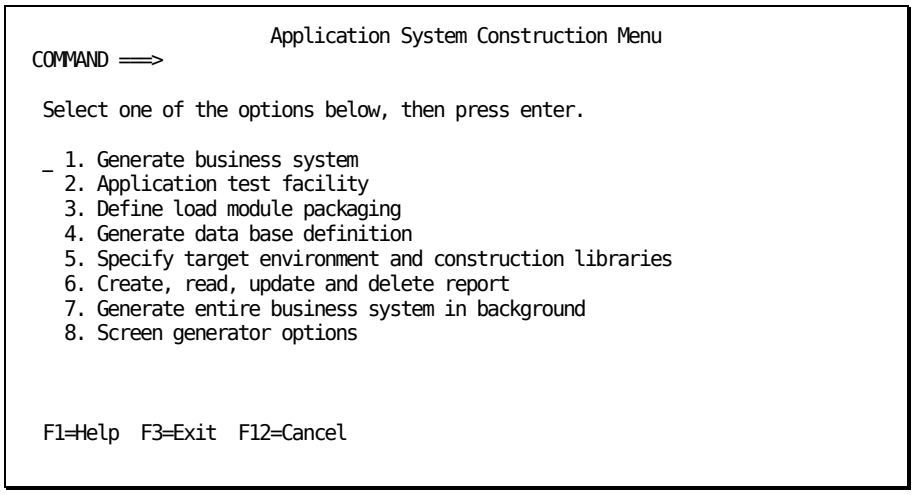
Specify Target Environment for HE (see page 77)
Generation Options (see page 31)

Display the System Generation Functions

Follow these steps:

1. On the Application System Construction panel, type the Host Encyclopedia model name.

2. Type the business system name and press Enter.



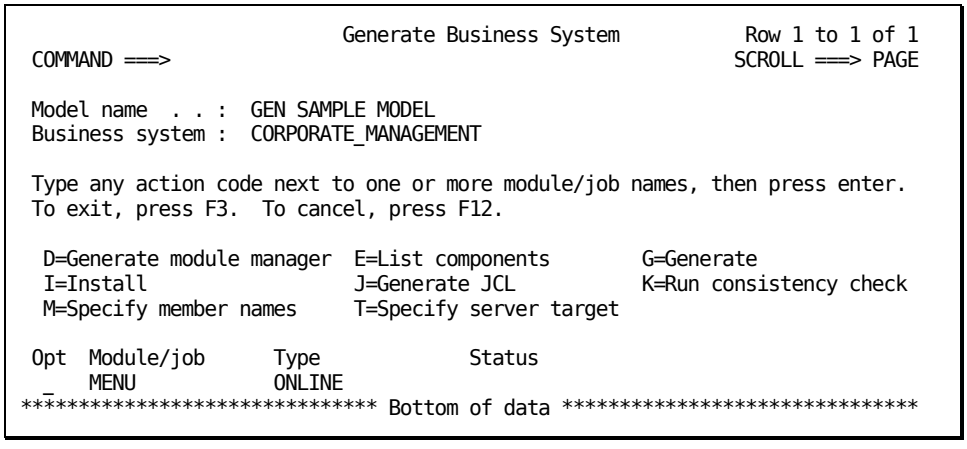
The system generation and installation functions are available through this menu.

CA Gen automatically retrieves the construction library and target environment specifications that are stored in the Host Encyclopedia for the selected business system.

3. Select the number of the item that is needed and press Enter.

Generation and Installation Tasks

The Generate business system option displays the load modules that belong to the selected business system.



Perform these tasks directly from the Generate Business System panel using the following options:

D

Generates the module manager

I

Installs the load module

M

Specifies member names

E

Lists the components

J

Generates JCL

T

Specifies server target

G

Generates the load module components

K

Runs consistency check

Run Consistency Check

Perform a consistency check on the entire model before generating code. Host Consistency Check uses the same rule base as the workstation consistency check.

Follow these steps:

Start Consistency Check from the Generate Business System panel or Batch Job Expansion panel.

1. Display the Generate Business System panel or Batch Job Expansion panel.
2. Type K next to a Module/Job name and press Enter.

   CA Gen displays the Check Model for Consistency panel.
3. Type / (slash) next to the range of objects to check.
4. Type / (slash) next to the execution option appropriate for the module or job.

5. Press Enter.

   CA Gen displays the Consistency Check Report Options panel.

   a. Select the report detail.

   b. Select to include errors and warnings.

   c. Set the diagnostic threshold to the number of messages at which to stop consistency checking.

      Default: 100

      Limit: 1-9999

   d. Select the diagnostic rule level and press Enter.

      Default: ALL

      When you chose ALL, consistency checking begins. Consistency check displays the report when it finishes checking.

   e. Press Exit to close the panel.

   When you chose SELECTED, CA Gen displays the Select Object Types for Model Consistency Check panel.

6. Type a / (slash) by one or more object types and press Enter to display a list of all occurrences of those types in the model.

   CA Gen displays the following message:
   `Input has been accepted, press "enter" or enter "accept" command to proceed.`

7. Press Enter.

   CA Gen displays a list of the object types in the model.

8. Select the expansion option by typing E next to individual occurrences of an object type and press Enter.

   CA Gen displays a list of all individual objects of the object type.

9. Select individual objects by typing S next to the objects to check and press Enter.

   CA Gen displays the Confirm Selected objects panel.

10. Type / (slash) next to an object name to remove it from the list, and press Enter.

11. Type Accept on the COMMAND line and press Enter to begin checking.

    The program checks the model for consistency.

    When complete, CA Gen displays the report.

Note: When the Consistency Check reaches the number of warnings or errors that are specified in the Consistency Check Report Options panel, CA Gen prompts to continue the Consistency Check.

Specify Member Names

To specify member names, press M in the Generate Business System panel or press M in the Batch Job Expansion panel.

Use the Specify member names option to identify or change the source member names for the following generated load module components:

- Procedure steps and Action blocks
- Screen Managers for online procedure steps only
- MFS Control Blocks for online procedure steps targeting IMS

The member names must meet these criteria:

- Be valid PDSE member names
- Contain a maximum of eight alphanumeric characters
- The first character must be a letter
- MFS format names must be a maximum of six characters.
- Each name must be unique within the model and library.

CA Gen displays an error message in the Response column next to the name when the name is invalid or a duplicate.

If you attempt to generate code before specifying the required member names, CA Gen asks you to specify the names during generation.

Follow these steps:

You can specify member names from the Business System Generation panel or from the Batch Job Expansion panel.

1. Display the Business System Generation or Batch Job Expansion panel.
2. Type an M next to the Module/Job name and press Enter.

   CA Gen displays the Specify Member Name for *Load Module/Batch Job* panel.
3. Use the Tab key to position the cursor to each name field and type a member name. CA Gen assigns default names to blank fields.
4. Click Save to store the changes, Cancel to ignore them, or press End to leave the panel without completing the procedure.

   When you click Save, CA Gen displays the message:
   `Member names saved.`

CA Gen displays the previous panel.

Default Member Names

If you omit a name, CA Gen assigns a default member name. For the procedure steps, CA Gen action blocks, screens, and MID and MOD members, the default name is the right-most seven digits of the Host Encyclopedia object ID of the component that is preceded by a letter:

P

Procedure steps

A

Action blocks

M

Screens and MOD names for MFS members

I

MID name for MFS members

For example, P0149508 is the default member name for a procedure step with an object ID of 0149508. CA Gen uses the MOD name as the member name of the MFS source module and defaults to the same name as the screen.

The default name for MFS members is the five rightmost digits of the Host Encyclopedia object ID for the component, which is preceded by the letter F.

The default name for an external action block is the first eight valid characters of the external action block name. If the first eight characters contain a character not valid in a PDS name, such as an underscore, CA Gen ignores the invalid character and truncates the remaining characters. For example, the default member name for the external action block READ_HISTORY is READHIST. If the first character is invalid, the default name is the last seven digits of the object ID preceded by the letter E. Change the default names to descriptive names before generating the member.

Dynamic Linking Parameters

Use the Specify Member Names panel to specify how to dynamically link a particular procedure step, action block, or screen.

- Use DEF to default to the corresponding Dynamically Link value that is specified in the target environment parameters.
- Use YES to designate the module as one to be invoked using a dynamic program call. Programs invoked using a dynamic program call are considered to be dynamically linked at runtime. During load module installation, the components that have their associated dynamically link packaging property set, or derived, to Yes are built so they reside in their own separately loadable executables. These application routines reside in DLLs.

- Use NO to designate the module as one to statically link in the load module in which it is defined.
- Use CMP to designate the module as marked for Compatibility. A module marked for Compatibility is dynamically called at runtime and resides in a non-DLL module. Compatibility modules use specific CA Gen runtime to enable DLLs to issue a dynamic program call to non-DLL load modules. This requires regenerating and reinstalling the module issuing the dynamic program call. CA Gen allows modules that are marked for Compatibility to be rebuilt to use CA Gen runtime DLLs. This requires that the modules marked for Compatibility be reinstalled and, if necessary, regenerated.
- To maintain compatibility with previous Gen releases that only supported static batch procedure steps, CA Gen generates and installs Batch procedures that are set to DEF as static modules, regardless of the default dynamic link setting for the business system
- RI triggers and action blocks, including EABs, statically called by modules that are marked for Compatibility must be built using the NODLL compiler option. When these RI triggers and action blocks are also used in Gen applications that are built as DLLs, they must be compiled using the DLL option.
- Selecting the option Process modules that are marked for Compatibility causes the RI triggers and the statically called action blocks to be generated and precompiled once but compiled twice – once as NODLL and again as DLL.
- Separate the libraries are provided in the target environment to hold the separate NCAL modules resulting from the two compile steps.

Since the CMP option is not supported for a z/OS Library load module, do not use CMP for the modules that are listed under a z/OS Library. Doing so cause runtime errors that is reported by consistency check.

Renaming Generated Members

After generating a load module or load module component, you can use the Specify Member Names panel to rename members. If a member name changes, all affected load modules must be regenerated.

More information

Intelligent Regeneration (see page 245)

Specify Server Target Environment

Use the Specify target environment option in the Generate Business System panel to specify or change the following for the load module:

- Operating System
- Generated the source language
- DBMS
- TP Monitor

CA Gen uses these values during Construction, overriding the equivalent parameters that are specified for the business system using the Target environment and Construction Libraries option.

Follow these steps:

1. Display the Application System Construction Menu.
2. Type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.
3. Type T in the column next to the module name and press Enter.

   CA Gen displays the Specify Server Target Environment panel. The panel shows the target operating system, generated source language, DBMS, and TP Monitor for the server load module.
4. Use the Tab key to position to cursor at each field to make or change an entry and type the entry.
5. Select a command to exit the panel.

Generate All Components in an Online, Server, or z/OS Load Module

Use the Generate option in the Generate Business System panel to generate all components in an online, server, or z/OS load module.

For the online load module, use the Generate Business System panel to generate:

- Procedure step and action block source. Foreign action blocks, z/OS Library action blocks, and external action block stubs are not generated.
- Dialog Manager source (optional)
- Screen manager source (online only)
- MFS source (online, IMS only)

For the server load module, use the Generate option to generate:

- Procedure step and action block source. Foreign action blocks, z/OS Library action blocks, and external action blocks are not generated.
- Server Manager source (optional)

For a z/OS Library Load Module, use the Generate option to generate action block source. Foreign action blocks that include z/OS Library action blocks that are packaged into another z/OS Library Load Module and external action blocks are not generated.

Follow these steps:

1. On the Application System Construction Menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, z/OS Library load modules, and batch jobs in the business system.

2. Type G in the column next to the online, server, or z/OS Library module name and press Enter.

   If member names were not specified for all components in the load module, CA Gen displays the Specify Member Names panel.

   a. Use the Tab key to position the cursor at each name field and type a member name. CA Gen assigns default names to blank fields.

   b. Click Save.

      When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

3. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

4. Press Enter to accept the parameters or click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

5. Press Enter to exit the Generation Status panel when generation completes.

   When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

6. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options panel was not set to Yes.

More information:

Generate an External Action Block Stub (see page 155)

Generate All Components in a Batch Load Module

When you select G for a batch load module, CA Gen generates:

- Procedure step and action block source. Foreign action blocks and external action block stubs are not generated.
- Batch Manager source (optional)

Follow these steps:

Use this procedure when the batch job includes multiple batch load modules. When the batch job contains one batch load module, use the Generating All Components for All Load Modules in a Batch Job procedure.

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E in the column next to the Job name and press Enter.

   CA Gen displays the Batch Job Expansion panel listing the load modules in the job.

3. Type a G in the column next to a component and press Enter.

   If member names were not specified for all components of the load module, CA Gen displays the Specify Member Names panel.

   a. Use the Tab key to move the cursor to each name field and type a member name. CA Gen assigns default names to empty fields.

   b. Press End.

      When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

4. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

5. Press Enter to accept the parameters or click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

6. Press Enter to exit the Generation Status panel when generation is complete.

   When you selected processing in the background and the Present JCL option is set to Yes in the Generation Options panel, CA Gen displays the Update JCL for Background Generation panel.

7. Make the changes and press Enter to submit the job.

Generate a Dialog Manager

Generating a dialog manager is useful when the dialog flow or packaging that changed, but not the associated action diagrams.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel showing the online load modules, server load modules, z/OS Library load modules, and batch jobs in the business system.

2. Type a D in the column next to the online load module name and press Enter.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

3. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

4. Press Enter to accept the parameters or click Cancel to return to the options panel.

   CA Gen displays the Generation Status panel, when you selected processing in the foreground.

5. Press Enter to exit the Generation Status panel when generation completes.

   When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

6. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

Generate a Server Manager

Generating a server manager is useful when the packaging changed, but not the associated action diagrams.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, z/OS Library load modules, and batch jobs in the business system.

2. Type a D in the column next to the server load module name and press Enter.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

3. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present generation option is set to Yes in the Generation Options, CA Gen displays the Confirm Generation of *Load Module* panel.

4. Press Enter to accept the parameters or click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

5. Press Enter to exit the Generation Status panel when generation completes.

6. When the Present JCL option is set to Yes in the Generation Options panel, and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

7. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

Generate a Batch Manager

Generating a Batch Manager for a batch load module is useful when the dialog flow or packaging that is changed, but not the associated action diagrams.

Follow these steps:

When the batch job contains one batch load module, use the Generate All Batch Managers in a Batch Job procedure.

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, z/OS Library load module, and batch jobs in the business system.

2. Type an E in the column next to a batch job name and press Enter.

   CA Gen displays the Batch Job Expansion panel listing the load modules in the batch job.

3. Type a D in the column next to a batch load module name and press Enter.

   When Present generation options are set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

4. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

5. Press Enter to accept the parameters or click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

6. Press Enter to exit the Generation Status panel when generation completes.

7. When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

8. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

Generate All Load Modules in a Batch Job

Use this option to generate all components for all load modules in a batch job. CA Gen generates for each load module:

- Procedure step and action block source. Foreign action blocks, z/OS Library action blocks, and external action block stubs are not generated.
- Batch Manager source (optional)

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, z/OS Library load modules, and batch jobs in the business system.

2. Type a G in the column next to the Job name and press Enter.

   If member names were not specified for all components being generated, CA Gen displays the Specify Member Name panel.

   a. Use the Tab key to position the cursor to each name field and type a member name. CA Gen assigns default names to blank names.

   b. Click Save.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Batch Job* panel.

3. Override the default values for this generation, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Batch Job* panel.

4. Press Enter to accept the parameters or click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

5. Press Enter to exit the Generation Status panel when generation is complete.

   When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

6. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

Generate All Batch Managers in a Batch Job

Generating all Batch Managers in the batch job is useful when the dialog flow or packaging that is changed, but not the associated action diagrams.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel. The panel shows the online load modules, server load modules, z/OS Library load module, and batch jobs in the business system.

2. Type D next to the Job name and press Enter.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Batch Job* panel.

3. Override the default values for this generation and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Batch Job* panel.

4. Press Enter to accept the parameters or click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

5. Press Enter to exit the Generation Status panel when generation completes.

6. When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

7. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

Expand an Online Load Module

Expand an online load module to list the module's components in the Load Module Expansion panel. Expand a load module to individually generate its components, that are the procedure steps, action blocks, screens, or external action blocks.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel.

2. Type an E in the column next to an online load module name.

   CA Gen displays the Load Module Expansion panel.

Expand a Server Load Module

Expand a server load module to list its components in the load module in the Server Load Module Expansion panel. Expand a load module to generate its components, that is, the procedure steps, actions blocks, or external action blocks.

Follow these steps:

1. In the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel.

2. Type an E in the column next to a server load module name.

   CA Gen displays the Online Load Module Expansion panel.

Use the Online Load Module Expansion panel to select generation, installation, and other options.

Expand a z/OS Library Load Module

Expand the z/OS Library load module to list its components in the Load Module Expansion panel. Expand a load module to individually generate its components, the actions blocks.

Follow these steps:

1. In the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel.

2. Type an E in the column next to a z/OS Library load module name.

CA Gen displays the z/OS Library Load Module Expansion panel to select generation, installation, and other options.

Expand a Batch Job

Expand a batch job to list its load modules in the Batch Job Expansion panel. If the batch job contains one load module, CA Gen displays the components of the load module in the Batch Load Module Expansion panel. You do not have to expand further. Expand a batch job to individually generate the components for a specific batch load module.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E in the column next to a batch job name.

   When the batch job has one batch load module, CA Gen displays the Batch Load Module Expansion panel.

   When the batch job has multiple batch load modules, CA Gen displays the Batch Job Expansion panel. Use this panel for generation, installation, and other tasks. Further expansion displays the Batch Load Module Expansion panel for batch load module components. This provides a lower level of generation functions.

Expand a Batch Load Module

Expand a batch load module to list the components of the selected batch load module in the Batch Load Module Expansion panel. Expand to this level to generate individual components of a batch load module.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules and batch jobs in the business system.

2. Type an E in the column next to a batch job name and press Enter.

   When the batch job includes multiple batch load modules, CA Gen displays the Batch Job Expansion panel.

   When the batch job includes one batch load module, CA Gen displays the Batch Load Module Expansion panel listing the batch load module components available for generation.

3. Type an E in the column next to a batch load module name and press Enter.

   CA Gen displays the Batch Load Module Expansion panel.

4. Select individual batch load module components to generate.

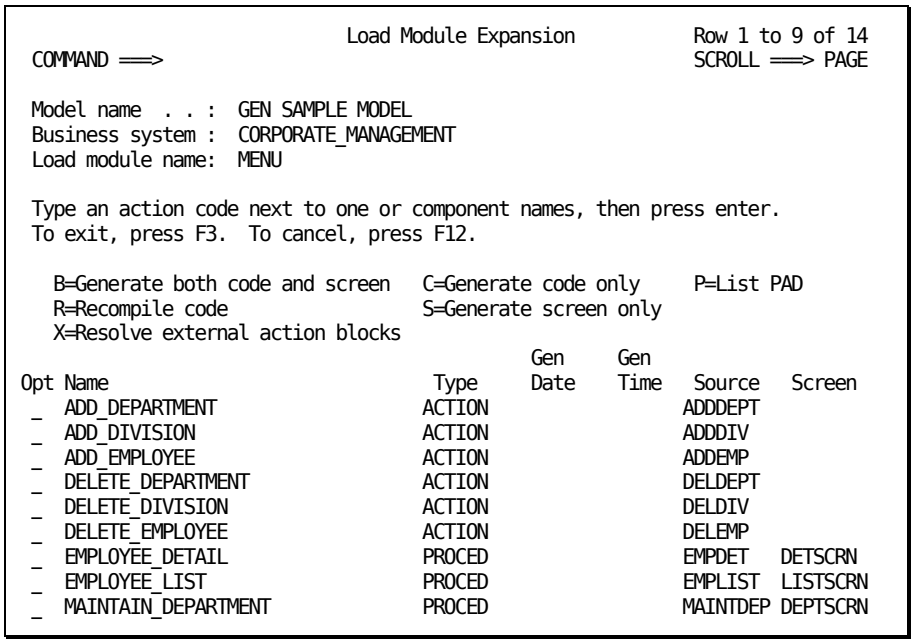
Generate an Online, Server, z/OS Library, or Batch Load Module Component

Use the Load Module Expansion panel to generate an online, server, z/OS Library, or batch load module component.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

2. Type an E in the column next to a load module name and press Enter.



```
                              Load Module Expansion          Row 1 to 9 of 14
COMMAND ===>                                                 SCROLL ===> PAGE

Model name  . . :  GEN SAMPLE MODEL
Business system :  CORPORATE_MANAGEMENT
Load module name:  MENU

Type an action code next to one or component names, then press enter.
To exit, press F3.  To cancel, press F12.

  B=Generate both code and screen   C=Generate code only     P=List PAD
  R=Recompile code                  S=Generate screen only
  X=Resolve external action blocks

                                              Gen     Gen
Opt Name                                      Type    Date    Time   Source   Screen
_   ADD_DEPARTMENT                            ACTION                 ADDDEPT
_   ADD_DIVISION                              ACTION                 ADDDIV
_   ADD_EMPLOYEE                              ACTION                 ADDEMP
_   DELETE_DEPARTMENT                         ACTION                 DELDEPT
_   DELETE_DIVISION                           ACTION                 DELDIV
_   DELETE_EMPLOYEE                           ACTION                 DELEMP
_   EMPLOYEE_DETAIL                           PROCED                 EMPDET   DETSCRN
_   EMPLOYEE_LIST                             PROCED                 EMPLIST  LISTSCRN
_   MAINTAIN_DEPARTMENT                       PROCED                 MAINTDEP DEPTSCRN
```

3. Select a generation option, B, C, or S, for one or more components:

   B

   For procedure steps when generating the source code and the screen.

   C

   To generate only the source code for a component.

   S

   To generate only the screen source code.

4. Press Enter.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

5. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

6. Press Enter to accept the parameters. Click Cancel to return to the options panel.

   If you selected processing in the foreground, CA Gen displays the Generation Status panel.

   To exit the Generation Status panel when generation is complete, press Enter.

   When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

7. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit JCL option on the Generation Options screen was not set to Yes.

To generate a server load module or a z/OS library load module component

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E in the column next to a load module name and press Enter.

   CA Gen displays the Load Module Expansion panel with load module's components.

3. Type a C in the column next to the component name to generate code and press Enter.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

4. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

5. Press Enter to accept the parameters or click Cancel to return to the options panel.

   CA Gen displays the Generation Status panel if you selected processing in the foreground.

6. When generation completes, press Enter to exit the Generation Status panel.

   When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

7. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

To generate a batch load module component

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules and batch jobs in the business system.

2. Type an E in the column next to a batch job name and press Enter.

   When the batch job includes multiple batch load modules, CA Gen displays the Batch Job Expansion panel listing the batch load modules in the batch job.

   - Type an E in the column next to a batch load module name and press Enter.

     CA Gen displays the Batch Load Module Expansion panel with the module's components.

   or

   When the batch job contains one batch load module, CA Gen displays the Batch Load Module Expansion panel with the module's components.

   - Type a C in the column next to a component name to generate code.

     When Present generation options are set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

3. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

4. Press Enter to accept the parameters and click Cancel to return to the options panel.

   CA Gen displays the Generation Status panel when you selected processing in the foreground.

5. Press Enter to exit the Generation Status panel when generation completes.

   When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

6. Make the changes and press Enter to submit the job.

   You only press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

More information:

Generate an External Action Block Stub (see page 155)

Generate Code Only

In the Load Module Expansion panel, type C to generate the source code only for the selected components. You can generate code for procedure steps, action blocks, foreign action blocks, or external action blocks. However, you cannot generate code for action blocks that are packaged into a z/OS Library unless the expanded z/OS Library owns the action block. You can select more than one component by placing a C in the space by several entries before you press Enter.

Note: When you use this option to generate a foreign action block, CA Gen stores the source in the library that is defined for the currently selected business system, not the business system that owns the action block. Because this can lead to undesirable results, use this option to generate a foreign action block only when it is not possible to generate it in its owning business system.

Generate Screen Only

In the Load Module Expansion panel, type S to generate the source code only for the screen that is associated with a procedure step. This option is useful when you changed the content of a screen but not the procedure step that uses it. CA Gen generates MFS code with the screen if the target TP monitor is IMS and the screen format type is MFS or MFS/EOF.

Generate Both Code and Screen

In the Load Module Expansion panel, type B to generate the source code module and the associated screen. CA Gen also generates MFS code with the screen if the target TP monitor is IMS and the screen format type is MFS. Only procedure steps have screens that are associated with them. You cannot test a procedure step without its associated screen.

Generate an External Action Block Stub

When you select C for an external action block component in the Load Module Expansion panel, CA Gen generates the stub of the program that implements an external action block. The generated stub is a COBOL source listing that includes the view definitions that are required to interface with CA Gen generated programs. The stub is stored in the sequential data set:

prefix.*stubname*.COBOL

Where *prefix* is the value that is entered in the TIUPREF variable during host installation. The stub helps you write external action block code that interfaces correctly with CA Gen generated code. It serves as a base to which you add the external action block logic.

Follow these steps:

You can generate the stub from the Load Module Expansion panel for online or batch load modules. This example is for batch load modules.

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E in the column next to a batch job name and press Enter.

   When the batch job contains multiple batch load modules, CA Gen displays the Batch Job Expansion panel listing the batch load modules.

   - Type an E to expand the online load module, server load module, or batch load module that contains the external action block to generate and press Enter.

     CA Gen displays the Load Module Expansion panel.

   or

   When the batch job contains only one batch load module, CA Gen displays the Batch Load Module Expansion listing the batch load module's components.

   - Type a C next to the external action block and press Enter.

     When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for *Load Module* panel.

3. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of *Load Module* panel.

4. Press Enter to accept the parameters or click Cancel to return to the options panel.

   CA Gen displays the Generation Status panel when you selected processing in the foreground.

5. Press Exit to exit the Generation Status panel when generation completes.

6. When the Present JCL option is set to Yes in the Generation Options panel and you selected processing in the background, CA Gen displays the Update JCL for Background Generation panel.

   Make the changes and press Enter to submit the job.

More information:

External Action Blocks (see page 207)

Recompile Code

When you select R in the Load Module Expansion panel, CA Gen performs the compile function on the selected load module component. Normally, it is not necessary to recompile code. As an example, you use this option if you accidentally deleted the executable load module.

You cannot use R (Recompile) option for an action block that is packaged into a z/OS Library, unless the expanded load module is the z/OS Library into which the action block was packaged.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E next to an online load module, server load module, z/OS Library load module, or batch job name to expand and press Enter.

   If you chose a batch job with multiple batch load modules, CA Gen displays the Batch Job Expansion panel listing the batch load modules in the batch job.

   If you chose an online or server load module, CA Gen displays the Load Module Expansion panel listing the components of the load module.

   If you chose a batch job with one batch load module, CA Gen displays the Batch Load Module Expansion panel listing the contents of the batch load module.

3. Type an E next to the batch load module to expand and press Enter.

   CA Gen displays the Batch Load Module Expansion panel listing the batch load module components for which you can generate a procedure action diagram listing.

4. Select the generation option R to recompile the code for the required components and press Enter.

   CA Gen displays the Generation Status panel.

5. Press Enter to exit the Generation Status panel when generation completes.

Resolve External Action Blocks

When you select X in the Load Module Expansion panel, CA Gen displays a panel to enter the names of the DB2 DBRMs for external action block. CA Gen procedure steps and action blocks use external action blocks to access logic that is defined in subroutines that are created outside CA Gen. Resolve external action blocks that use embedded SQL. Resolving an external action block means identifying DB2 DBRMs associated with it. You must do this before installing CA Gen procedure step or action block that uses the external action block.

When you resolve an external action block, you also change the external action block member name. If an external action block does not use embedded SQL, you do not need to resolve it.

Follow these steps:

You do not need to resolve the external action block if it does not contain embedded SQL. You can resolve an external action block from the Load Module Expansion panel for online or batch load modules.

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.
2. On the Generate Business System panel for online and server load modules or the Batch Job Expansion panel for batch load modules.
3. Type an E to expand the load module or job step that contains the external action block to generate.

   CA Gen displays the Load Module Expansion panel.
4. Type an X next to the external action block and press Enter.

   CA Gen displays the External Action Block Resolution panel.
5. When changing the load module name, that is, the external action block member name, type the new name.
6. If the external action block references a database, type the name of the DB2 DBRM.

   You can specify up to 60 DBRMs. A DBRM name is a maximum of eight alphanumeric characters. The first character must be a letter.
7. Press Enter to save the changes or click Cancel to cancel.

   CA Gen displays the Load Module Expansion panel.

More information:

External Action Blocks (see page 207)

List a Procedure Action Diagram

When you select P in the Load Module Expansion panel, CA Gen creates a Procedure Action Diagram for a selected procedure step or action block that you can view without returning to the workstation toolset.

When you select this option, CA Gen generates a listing, stores it in a sequential file with the name:

prefix.*membername*.adlist.*suffix*

where *prefix* and *suffix* are the prefix and suffix set during the installation of the Host Encyclopedia Construction tool, and displays the listing in TSO Browse mode.

Note: For more information about setting the prefix and suffix for default file names, see the *z/OS Installation Guide*.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E next to an online load module or batch job name to expand and press Enter.

   When you select a batch job with multiple batch load modules, CA Gen displays the Batch Job Expansion panel listing the batch load modules.

   When you select an online load module, CA Gen displays the Load Module Expansion panel listing the components of the load module.

   - Type an E next to the batch load module that to expand and press Enter.

     CA Gen displays the Batch Load Module Expansion panel listing the batch load module components for which you can generate a procedure action diagram listing.

   or

   When you select a batch job with one batch load module, CA Gen displays the Batch Load Module Expansion panel listing the contents of the batch load module.

   - Type a P in the column next to the appropriate procedure step or action block and press Enter.

     CA Gen displays the Procedure Action Diagram in TSO Browse mode.

3. Press End to return to the Generate Business System panel.

CA Gen creates the listing from information in the Host Encyclopedia, identical to the one included as part of every generated procedure step and action block source module. It contains the action diagram, its views, and related dialog flow details.

The following illustration is an example of a procedure action diagram listing. The action diagram numbers verbs, not statement lines.

```
BROWSE — AAAC.LAG.LISTING (P0173162) --- LINE 00000000 COL 001 080 —


COMMAND ===>                                                SCROLL ===> PAGE
************************** TOP OF DATA ********************************
P0173162
    +>      MENU                                              MM/DD/YY   HH:MM
    |       IMPORTS: Entity View input division
    |                number FROM sent division
    |                Entity View input cost_center
    |                Entity View input employee
    |                number FROM sent employee
    |      EXPORTS:  Entity View sent division
    |                number
    |                Entity View sent cost_center
    |                number
    |                Entity View sent employee
    |                number
    |      PROCEDURE STATEMENTS
1   |      MOVE input division TO sent division
2   |      MOVE input cost_center TO sent cost_center
3   |      MOVE input employee TO sent employee
4   | +>   CASE OF COIMMAND
4   | |>   CASE DIVLIST
5   | |>   EXIT STATE IS transfer_to_division_maint
4   | |>   CASE CCLIST
6   | |>   EXIT STATE IS transfer_to_cc_maint
4   | |>   CASE EMPLIST
7   | |>   EXIT STATE IS transfer_to_employee_maint
4   | |    OTHERWISE
8   | |    EXIT STATE IS invalid_option
8   | +-----
8   +-----
                            PROCEDURE STEP VIEW MATCHING REP
************************** BOTTOM OF DATA *****************************
```

Procedure Step View Matching Report

The Procedure Step View Matching Report lists all dialog flows to and from the procedure step. The list is at the end of the PAD listing for a procedure step and uses the format in the following illustration.

```
Procedure Step View Matching Report

  Dialog Flows Targeting DISPLAY_TDPV_ORDER

    TD_PERMITTED_VALUES_MENU links to DISPLAY_TDPV_ORDER
      flows on: DISPLAY_DATA
      setting command to CLEAR
      no views matched

    DISPLAY_TDPV_ORDER returns to (DISPLAY FIRST) TD_PERMITTED
      returns on: RETURN
        with autoflow on RETURN
      setting command to <SPACES>
      no views matched

End Of Procedure Step View Matching Report
```

As can be seen from the illustration, the report has an opening and closing title and two headers that separate the report into two groups:

- Flows originating from the listed procedure
- Flows targeting the listed procedure

Each header includes entries for every transfer, link, or return in that direction. Each entry includes the following information:

- The source and destination procedure names and the type of flow.
- The exit state that causes the flow, if any. If no exit state is defined, CA Gen issues one of the following error messages:
  - When the flow is a transfer or a line:
    `*** Error: A dialog flow must have at least one "flows on" exit state.`
  - When the flow is a transfer or a line:
    `*** Error: A "link" type of dialog flow must have at least one "returns on" exit state.`
- The autoflow command, if any. If no autoflow, this line is absent.
- The command is passed, if any, or one of the following three special literals that correspond to choices you can make in the dialog flow diagram when you detail the properties of a flow and can select the SET CMD or RTN CMD buttons:
  `<SPACES> <CURRENT> <PREVIOUS>`
- A list of the views that are matched, if any, or the message no views matched.

Entity views are listed showing the view name and the entity name, while group views show only the group view name, followed by the cardinality in parentheses. Non-repeating group views do not show a cardinality but can be distinguished from an entity view by the absence of an entity name.

When the cardinalities of the group views do not match, CA Gen adds this note below the view:

```
*** Note: Group view cardinalities are not the same.
```

When a repeating group view is matched to a non-repeating group view, CA Gen adds this note below the view:

```
*** Note: Group view matched to non-repeating group view.
```

Error and notes have three asterisks on the left. They do not necessarily indicate that an action must be taken. Use consistency check to determine if the condition is valid.

Install an Online Load Module, Server Load Module, z/OS Library Load Module, Batch Job, or Batch Load Module

When you select I on the Generate Business System panel, CA Gen starts the installation process for an online load module, a server load module, a z/OS Library load module, or all load modules in a batch job. When you select I on the Batch Job Expansion panel, CA Gen starts the installation process for one load module in a batch job.

Automatic Installation

In the Generation Options panel, set automatic install to Yes to start the installation when generation completes. This option can also be set from the Generation Options for *Load Module* panel, if display is not suppressed.

Note: Before installing the load module, create the DB2 tables and related table spaces, index spaces, and indexes for the entities referenced. Installation is the final step in system generation. If a local install is requested, this function compiles the generated source code, link-edits the executable load modules for a business system, and binds the database plan. If a remote install is requested, this function builds the remote files to transfer to the target environment.

The first step of the install function is to create an Install Control file. The Install Control file is stored in the Installation Control library you specified when you set up your internal libraries. The member name is the name of the load module.

Install Control File

The Install Control files that are created for local and remote installs have different formats. The local Install Control file is in a format consistent with earlier CA Gen releases and is processed by the CLIST TICINSTX. Modify this CLIST to customize local installations for your site.

Note: For more information, see the *z/OS Installation Guide*.

The local Install Control file contains the following information:

- Transaction codes that are supported by the load module
- IMS program specification block (PSB) information, if applicable
- MFS control blocks, if applicable
- Name of the load module to link edit
- Names of z/OS Library action blocks to import during link edit
- Names of the SYSLIBs, SYSLIB DD names, to use during the link edit
- Names of the APPLOADs, the APPLOAD DD names, to use during the link edit
- Required link-edit statements, such as the INCLUDE or IMPORT statements for the components of the load module
- Name of the DB2 Plan to bind
- Names of the DB2 DBRM libraries to use during the DB2 application bind
- Names of the DBRMs

Local Installation

The local install function performs a link edit that creates executable load modules. All external references are resolved. If the target DB2 subsystem, the DB2 subsystem in which the generated application runs, is on the same CPU as CA Gen, CA Gen also binds the DB2 Plan. The DB2 Plan name is the load module name.

More information:

DB2 Package Bind Options (see page 65)

Binder Control Cards Library

The Binder Control Cards Library members are created during local installs. They contain the binder control cards information that is used during link-edit processing, as in SYSLIN DD, for example, IMPORT, INCLUDE APPLOAD, ENTRY, and REPLACE statements.

Remote Installation

The remote install function copies the generated source into a remote file for the Implementation Toolset to process in the target environment.

DB2 Bind Authority

To perform the bind process, requires authority over the DB2 tables. The TSO ID must create the tables or must specifically grant BIND authority for the tables. One way to handle the DB2 security and authority requirements is to use a single user ID to install all load modules for an application.

Note: For more information about DB2 authority, see the *z/OS Installation Guide*.

Install an Online Load Module, a Server Load Module, or a z/OS Library Load Module

To Install an online load module, a server load module, or a z/OS library load module

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, z/OS Library load modules, and batch jobs in the business system.

2. Type an I next to the online, server, or z/OS Library load module to install and press Enter.

   When the Present installation option is set to Yes in the Generation Options panel, CA Gen displays the Installation Options for *Load Module* panel.

3. Override the default values for this installation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Installation of *Load Module* panel.

4. Press Enter to begin processing or click Cancel to stop processing.

   If you started processing, CA Gen displays the Generation Status panel. If you cancelled processing, CA Gen displays the Generate Business System panel.

To complete the installation process, see Implementation Considerations.

The binder control cards that are used in link-edit (SYSLIN DD) are saved in the Binder Control Cards library.

The output of the link edit is saved under its member name in a library with the name:

prefix.IEF.LKSPRNT.LINK.*suffix*

where *prefix* and *suffix* are a prefix and suffix set during the installation of the Host Encyclopedia Construction tool. If the link edit fails, the output from the link-editor is displayed in TSO Browse mode.

To install all load modules in a batch job

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an I in the column next to the batch job to install and press Enter.

   When the Present installation option is set to Yes in the Generation Options panel, CA Gen displays the Installation Options for *Batch Job* panel.

3. Override the default values for this installation, if necessary, and press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Installation of *Batch Job* panel.

4. Press Enter to begin processing or click Cancel to stop processing.

If you started processing, CA Gen displays the Generation Status panel. If you cancelled processing, CA Gen displays the Generate Business System panel.

To complete the installation process, see the Implementation Considerations.

CA Gen saves the output of the link edit as its member name in a library named:

prefix.ief.lksprnt.suffix

Where prefix and suffix are a prefix and suffix set during the installation of the Host Encyclopedia Construction tool. If the link edit fails, CA Gen displays the output from the link-editor in TSO Browse mode.

To install a batch load module

Use this procedure when the batch job contains one batch load module. When the batch job contains multiple batch load modules, use the To install all load modules in a batch job procedure.

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type an E in the column next to the Batch Job containing the load module to install and press Enter.

   CA Gen displays the Batch Job Expansion panel.

3. Type an I next to the load module to install and press Enter.

   When the Present installation option is set to Yes in the Generation Options panel, CA Gen displays the Installation Options for *Load Module* panel.

4. Press Enter to continue.

   When the Present confirmation panel option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Installation of *Load Module* panel.

5. Press Enter to begin processing or click Cancel to stop processing.

   If you started processing, CA Gen displays the Generation Status panel. If you cancelled processing, CA Gen displays the Generate Business System panel.

To complete the installation process, see the Implementation Considerations.

CA Gen save the binder control cards that are used in link-edit, SYSLIN DD, in the Binder Control Cards library.

CA Gen save the output of the link edit as its member name in the following library:

prefix.IEF.LKSYSPRT.LINK.*suffix*

where *prefix* and *suffix* are a prefix and suffix set during the installation of the Host Encyclopedia Construction tool. If the link edit fails, CA Gen displays the output from the link-editor in TSO Browse mode.

Generate JCL for a Batch Job

When you select J in the Generate Business System panel, CA Gen generates the JCL to execute a batch job. The member name for the batch job is the job name that is listed on the Generate Business System panel. CA Gen uses the JCL library that you specified when you set up your internal libraries.

After generating the JCL, CA Gen displays it in an ISPF Edit session. You can modify the JCL before saving it. You can submit the modified JCL to run as a batch job.

Follow these steps:

1. On the Application System Construction menu, type 1 for the Generate business system option and press Enter.

   CA Gen displays the Generate Business System panel listing the online load modules, server load modules, and batch jobs in the business system.

2. Type a J next to the job name and press Enter.

   The system generates the JCL and places you in an ISPF Edit session. CA Gen assigns the job name as the generated member. The library is the one you specified when you set up your internal libraries.

3. Modify the JCL and submit the job or leave the panel.

   - Complete the changes and type SUBMIT on the COMMAND line and press Enter to submit the job.
   - Press End to leave ISPF edit and return to the Generate Business System panel without submitting the job.

More information:

Specify Target Environment for HE (see page 77)

Generate for Remote Installations

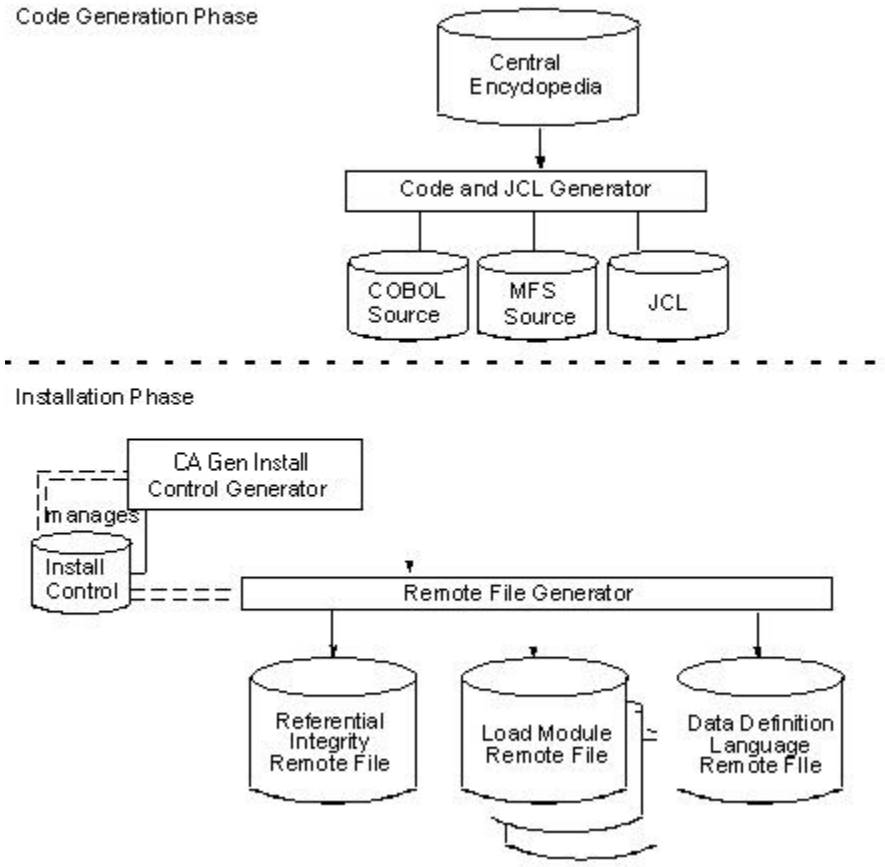
The procedures to generate code for a remote installation are the same as the procedures for a local installation. Setting the Remote Installation option to Yes in the Generation Options panel causes CA Gen install process to prepare the generated code for transfer to a remote environment. CA Gen stores the generated code in a set of remote files.

Transfer the IPs to the target platform after generation. Use the Implementation Toolset on the target platform to install the remote files.

Note: For more information, see the *z/OS Implementation Toolset User Guide*.

The following illustrations show the basic flow of the generation process. Each step in generating the applications system, that is database generation, referential integrity generation, and system generation, produces files for a remote installation.

The following illustration shows the flow of remote system generation with libraries.



The database DDL remote file is named *prefix*.IEF.*dbname*.REMINST. One DDL remote file exists for all load modules in the application.

Load module remote files are named *prefix*.IEF.*loadname*.REMINST. One load module remote file exists for every load module in the application. Be sure to transfer them all.

The RI trigger remote file is named *prefix*.IEF.CASCADE.REMINST. Referential integrity is implemented with the use of RI trigger modules. Each CA Gen application has only one RI trigger remote file, regardless of the number of load modules in the application.

The Implementation Toolset on the target environment splits the remote file, generating a command procedure from the control information and populating libraries/directories with the other components of the remote file. The command procedure is used to compile, link, and install the load modules.

The remote installation involves these major activities:

- Use CA Gen Host Encyclopedia Construction tools to define and generate the database definition (DDL), referential integrity (RI) triggers, and application source code for remote installation. CA Gen requires that the portion of the model for which the application system is being constructed be consistent.
- Transfer the generated database definition, RI triggers, and application code, in the form of the remote files, to the remote environment.
- Use the Implementation Toolset to construct and install the generated database and application in the remote environment.
- Test the application system in the remote environment using CA Gen Application Execution Facility.

Special Considerations for large Common Format Buffers

For CA Gen Release 8.5, the amount of customer data that can be sent between CA Gen Client and Servers using Common Format Buffer (CFB) based cooperative flows has increased. The previous CFB limit was 32K. The new limit is 16.7 MB.

Installing the modules larger than 32K use more resources during deployment.

- Runtime routines for encryption and decryption now use a temporary buffer that is a max size of 16.7MB. This large buffer is released as soon as the algorithm has completed processing the buffer.
- Runtime user exit TIRSIPEX (CICS Sockets Server Exit) need modifications to the timeout parameters SELECT-TIMEOUT-SECS and SELECT-TIMEOUT-MICROSEC. These parameters are evaluated and increased as appropriate.
- The default MQSeries message length is evaluated and increased as appropriate.

Implementation Considerations

After CA Gen completes the link-edit and bind, you have some final considerations before implementing your system. You can think of these considerations as completing the installation process.

Use the installation control files that are created during system generation, that is the members in the Installation Control library, to identify the application components and actions to move your transaction into the execution environment, (IMS or CICS).

Make the CA Gen runtime modules available in the Target system.

Final considerations before implementing your system:

- Never modify the generated source code. Modify the action diagram instead.
- CA Gen requires that the DB2 application plan and the load module have the same name.
- Do not try to combine load modules that are generated from different versions of the same model.
- CA Gen does not grant the execution authority for generated applications when they are installed in DB2. The correct set of users must be authorized for an execution as a separate process.
- CA Gen generated applications reference a DB2 profile table and the tables the application uses unless the CICS temporary storage queue was selected. The DB2 application plans include references to the profile table. A DB2 synonym for the profile table must be established if the owner of the application tables is different from the owner of the profile table.

Note: For information about implementing CA Gen applications in CICS, see the *z/OS Installation Guide*.

More information:

Specify Target Environment for HE (see page 77)

Other Construction Options

Generate the Create/Read/Update/Delete Report

This report allows you to view which actions for a business system—create, read, update, or delete—are being conducted against the data model entity types. The report lists the actions by three groupings:

- By load module, action block, and entity action view
- By load module
- By load module and action block.

The Create/Read/Update/Delete Report (First Grouping) screen lists the names of the load modules, business system, source members, and action blocks. This grouping also lists the procedure step to which the action blocks belong. For example, the procedure step DIVISION_MAINT contains the action blocks BSD_ADD_DIVISION, BSD_MODIFY_DIVISION, and BSD_DELETE_DIVISION.

The abbreviations CRE, UPD, and DEL indicates the create, update, and delete actions, respectively. The READ action is implied because a read must occur before any other action. For example, the procedure step MAINTAIN_DIVISION reads REQUIRED DIVISION and EMPLOYEE DIVISION.

```
                        CREATE/READ/UPDATE/DELETE REPORT
                    BY LOAD MODULE, ACTION BLOCK AND VIEW
LOAD MOD    BUSINESS SYSTEM
  MEMBER      ACTION BLOCK
    CRE UPD DEL  VIEW                                   ENTITY
--------------------------------------------------------------------------------
MENU        CORPORATE_MANAGEMENT
  MAINMENU    MENU
  MAINTDIV    MAINTAIN_DIVISION
                 REQUIRED                               DIVISION
                 REQUIRED                               EMPLOYEE
  ADDDIV      ADD_DIVISION
       UPD       REQUIRED                               EMPLOYEE
    CRE UPD      NEW                                    DIVISION
  MODDIV      MODIFY_DIVISION
       UPD       REQUIRED                               EMPLOYEE
       UPD       NEW                                    EMPLOYEE
       UPD       REQUIRED                               DIVISION
  DELDIV      DELETE_DIVISION
                 EXISTING                               DEPARTMENT
            DEL  EXISTING                               DIVISION
```

The Create/Read/Update/Delete Report (Second Grouping) screen lists more information for each load module. EXT indicates an external action block with DB2 code. A Y indicates that the procedure step includes that action or external action block. Conversely, an N indicates that the procedure does not contain the action or external action block. At least one Y listed for CRE, UPD, DEL, or EXT indicates a repeatable read isolation level (RR) if target DBMS is DB2. If all of the entries are Ns, this indicates a cursor stability isolation level (CS) if target DBMS is DB2.

```
                          CREATE/READ/UPDATE/DELETE REPORT
                               BY LOAD MODULE

BUSINESS SYSTEM:  CORPORATE_MANAGEMENT
LOAD MOD  CRE UPD DEL EXT CS TRAN  DLG TRAN PROCEDURE STEP NAME
MENU        Y   Y   Y   N  MENU     MENU     MENU
                           MDIV     MDIV     MAINTAIN_DIVISION
                           MDEP     MDEP     MAINTAIN_DEPARTMENT
                           MEMP     MEMP     EMPLOYEE_LIST
                                    MEMP     EMPLOYEE_DETAIL
```

This grouping also lists the clear screen transaction code (CS TRAN), the dialog flow transaction code (DLG TRAN), and the name of the procedure step to which the transaction codes are associated.

The Create/Delete/Update/Delete Report (Third Grouping) screen is similar to the second grouping. In the third grouping, however, you can see the action block that conducts the actions.

```
                    CREATE/READ/UPDATE/DELETE REPORT
                  BY LOAD MODULE AND ACTION BLOCK

BUSINESS SYSTEM:  CORPORATE_MANAGEMENT
MODULE       ACTION BLOCK NAME              CRE UPD DEL EXT CS TRAN  DLG TRAN


MENU            *                            Y   Y   Y   N
MAINMENU     MENU                            N   N   N   N   MENU     MENU
  MAINTDIV   MAINTAIN_DIVISION               N   N   N   N   MDIV     MDIV
  ADDDIV     ADD_DIVISION                    Y   Y   N   N
  MODDIV     MODIFY_DIVISION                 N   Y   N   N
  DELDIV     DELETE_DIVISION                 N   N   Y   N
  MAINTDEP   MAINTAIN_DEPARTMENT             N   N   N   N   MDEP     MDEP
  ADDDEPT    ADD_DEPARTMENT                  Y   Y   N   N
  MODDEPT    MODIFY_DEPARTMENT               N   Y   N   N
  DELDEPT    DELETE_DEPARTMENT               N   N   Y   N
  EMPLIST    EMPLOYEE_LIST                   N   N   N   N   MEMP     MEMP
  DELEMP     DELETE_EMPLOYEE                 N   N   Y   N
  EMPDET     EMPLOYEE_DETAIL                 N   N   N   N       MEMP
  ADDEMP     ADD_EMPLOYEE                    Y   Y   N   N
  MODEMP     MODIFY_EMPLOYEE                 N   Y   N   N
```

To generate the Create/Read/Update/Delete report

1. On the Application System Construction Menu, type 6 for the Create/Read/Update/Delete report and press Enter.

   CA Gen displays the report in ISPF Browse mode. You can scroll up and down in the report with the ISPF scroll keys. The default keys are PF7 to scroll up and PF8 to scroll down.

2. Press the End key to leave ISPF Browse when you finish viewing the report. CA Gen displays a panel to print the report.

   CA Gen automatically creates a sequentially named data set for the report. The data set is named *prefix*.IEF.RPT*n,* where *prefix* is a value set during host installation in the TIUPREF variable and *n* is a number from one to ten. The CA Gen numbers the data sets from one through ten, and then restarts at one.

3. Type on of the following print options on the COMMAND line and press Enter:

   PK

   Print the data set and keep it.

   PD

   Print the data set, then delete it.

K

Keep the data set but do not print it.

D

Delete the data set without printing it.

When you print the data set, specify to print on a local printer or as a batch job on a system printer.

For a local printer, specify the print mode as local and the printer ID. To submit the print job as a batch job, specify the print mode as batch, the sysout class, and the JCL Job card information.

CA Gen displays the Application System Construction menu when the print completes.

Generate Entire Business System in Background

This option generates the source code for each load module within an entire business system. The generation occurs completely in the background. This option is especially useful if you have copied a model from one version of CA Gen to a newer version. Changing the versions sometimes requires that you regenerate and reinstall the load modules for each business system. The option is also useful when you are ready to move your application system into production. The background generation option provides a convenient way to generate all of the load modules for a business system.

Follow these steps:

1. On the Application System Construction menu, type 7 for the Generate entire business system in background option and press Enter.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Generation Options for Business System panel.

2. Override the default values for this generation, if necessary, and press Enter to continue.

   When the Present generation option is set to Yes in the Generation Options panel, CA Gen displays the Confirm Generation of Business System panel.

3. Press Enter to accept the parameters. Click Cancel to return to the options panel.

   When the Present JCL option is set to Yes in the Generation Options panel, CA Gen displays the Update JCL for Background Generation panel.

   If you specify the Job card during installation, CA Gen displays the information in this panel.

   The Job card keyword parameter USER= may or may not be required at your site depending on the security system and how it is installed. If you receive the following message:

   ```
   USER UNKNOWN DOES NOT HAVE CE ACCESS
   ```

   resubmit the job with the USER= parameter.

4. Make the changes and press Enter to submit the job.

   When processing completes, CA Gen displays the Background Job JCL Submission Panel listing messages that are produced during the JCL generation. The panel can display up to 10 messages. CA Gen writes other messages to the file:

   prefix.IEF.SYSPRINT

   where *prefix* is the value set for the TIUPREF variable during host installation.

5. Press Enter to submit the job or press End to cancel submission.

   If you submit the job, CA Gen issues a TSO message indicating that the job was submitted and press Enter to return to the Application System Construction menu.

   If you do not submit the job, CA Gen displays the Application System Construction menu.

Screen Generator Options

Use the Screen Generator options to set the TIMAPGEN environment variable when the target platform is z/OS. The options are Enhanced Screen Generation or Standard Screen Generation. The default is enhanced. The Enhanced Screen Generator is available when the variable is set to 2, or when it is undefined. Standard screen generation is available when the variable is set to 1.

Set TIMAPGEN to 1 to use the previous version of screen generation. TIMAPGEN is specific to each User ID. After setting TIMAPGEN, it remains set for all applications that are developed or generated with your User ID.

Follow these steps:

1. On the Application System Construction Menu, type 8 for the Screen generator options and press Enter.

   CA Gen displays the Screen Generator Version panel listing the current screen generator version option, ENHANCED or STANDARD.

2. To select an option, type the number or position the cursor, and press Enter to continue.
3. Press End to return to the Application System Construction Menu.

Chapter 9: Testing Applications

Testing Systems in the HE Environment

CA Gen provides an Application Test Facility to test new systems or changes to systems in a TSO testing environment that simulates the target production environment. You can use the Test Facility to test batch applications and online applications, that is IMS, CICS, and TSO applications. To test an application system, generate and install the load modules into the testing environment and execute the test. During the test, you see how your system behaves at runtime.

CA Gen also includes a Debug Trace Facility (DTF) to debug the CA Gen-generated programs at the model input level. With Debug Trace, you can execute and debug CA Gen action diagrams instead of debugging the COBOL code that is generated for them. As your generated application executes, you see the sequence of action block calls and action diagram statements executing, and can step through the execution of the application. You can also display and modify the views, that is, the data stores, and system variables for each action diagram. To use the Debug Trace Facility, select debug support when you generate your application.

You can execute the Debug Trace Facility under CICS to test CICS online transactions and Distributed Process Server (DPS) applications. This facility lets the developer test the application in the target environment.

For all other types of applications (IMS, TSO and batch), testing with the Debug Trace Facility must be done within the Application Test Facility.

Test a Load Module or Application

The following is a summary of a typical sequence of activities to build and test an executable load module, and test an application system.

Follow these steps:

1. Specify the libraries in which the CA Gen stores the generated objects and the external libraries that are used during construction.
2. Specify the target environment.
3. Specify the load module packaging.
4. Check the load module components for consistency.

5. Generate the load modules.

   Because this activity generates the source code, set the Generate Debug Support option to Yes when generating load modules that you test using the Debug Trace Facility.

6. Install the load modules in the testing environment.

   This activity compiles, link-edits, and binds the generated source code, resolving external references.

7. Test the application transactions.

8. Review the results of the test process.

   To correct errors, change toolset diagrams, not generated code.

9. Regenerate and reinstall selected components that are based on the changes to the diagrams.

10. Repeat the steps, beginning with generating the load modules, until the load modules are debugged.

    Note: As testing continues, generate components that have been fully tested (or are not a critical part of the test) without the debug support option.

11. When the application or load module is fully tested, use the system generation tools to generate (without Debug Support) and install the application in the target production environment.

Update your construction library specifications to reflect the production environment rather than the testing environment.

Application Test Facility Considerations

The Application Test Facility does not support external subroutine logic such as DL/I calls, IMS calls, or CICS calls. You can only update DB2 databases during test executions.

There are two ways to test applications that use DL/I, IMS, or CICS calls:

- Create CA Gen procedure steps without the USE statements that call the external logic before generating and testing CA Gen load modules. After debugging the load modules, add the USE statements to the action diagrams.
- Create CA Gen procedure steps, including the USE statements in your action diagrams. Create the stubs for the external routines that do not use DL/I, IMS, or CICS calls. Then generate and test CA Gen load modules. After the load modules have been debugged, add the DL/I, IMS, or CICS calls to the stubs.

When testing batch applications, it is important to consider the amount of test data used. Since the Application Test Facility executes online, we recommend that you use a small volume of data when testing batch applications.

Note: For information about moving an application that is installed into the testing environment using the z/OS IT or Host Encyclopedia Construction to a different TSO System see the z/OS *Implementation Toolset User Guide*.

CICS Debug Trace Facility Considerations

The Debug Trace Facility is executed under CICS to test CICS online and server applications. The TIRCRUNC DLL is required in CICS to test CICS online and server applications. Under CICS, external subroutine logic such as DL/I and CICS calls are tested. The Temporary Storage Queue (TSQ) Profile Manager is used too. Using the Debug Trace Facility under CICS result in long-running transactions in your test region.

Beginning with AllFusion Gen 7, servers no longer require TWASIZE set to 8. TWASIZE is set to zero.

Trace Facility Display Interface

The Trace Facility Display Interface (TFDI) is a transaction that is used to display a panel of debug information when the Debug Trace Facility is used to trace CICS online and server applications. This transaction must be installed to operate the Debug Trace Facility under CICS.

Generate Application System for Testing

Before testing an application system, generate the load modules for the transactions to test. When determining which load modules must be generated, consider the following points:

- Generate all load modules that you need for the test. When testing the flow between two procedure steps, be sure that both procedure steps were generated.
- During testing, the test fails when attempting to access a load module that has not been generated.
- When generating load module components separately, all components in a load module must be generated to install the load module or the install fails because the link-edit step has unresolved references.
- To test part of a multi-step procedure, always generate the first procedure step in addition to the procedure steps to test because the first procedure step provides the access to all other procedure steps in the procedure.

More information:

System Generation and Installation (see page 117)

Generate with Debug Support

Use the Debug Trace Facility to debug CA Gen-generated programs at the CA Gen diagram level. As your program executes, you see the sequence of action block calls and the sequence of action diagram statements executing. You can also display and modify the views and the system variables for each action diagram.

To test a load module or its components with the Debug Trace Facility, generate selecting the generate debug support option. Debug support is the code the CA Gen-generates to drive the Debug Trace Facility.

Install in the Test Environment

After generating a load module, install it to build the executable. The load module can be executed in TSO or in CICS, with or without Debug Support.

TSO Testing

To test using the Debug Trace Facility under TSO, specify IEFAE as the target environment and install selecting the Target TSO Test Facility option. If the application was generated with Debug Support the Debug Trace Facility is automatically enabled.

CICS Testing

To test using the Debug Trace Facility under CICS, specify CICS as the target environment and do not select the Target TSO Test Facility option. The application, the Trace Facility Display Interface (TFDI), and Debug Trace Facility (DTF) load modules must be installed in the CICS region. These modules are located in the CA Gen load library.

The TIRCRUNC DLL is required in CICS to trace CICS online and server applications.

Creating the Executable

The install function resolves all external references and performs a final link edit that creates executable load modules. Most of the CA Gen runtimes are no longer statically linked into the executables. Each CA Gen application type requires different runtimes.

The list of the runtime DLLs required to execute CA Gen TSO, Batch, and CICS applications are listed in the sections that follow. For a full list of the runtime DLLs used by CA Gen CICS and the IMS applications, see the z/OS Installation Guide.

More information:

System Generation and Installation (see page 117)

Execute Test Under the Application Test Facility

After installing a CA Gen load module in the test environment, you can test it. The application requires the CA Gen runtime DLL modules to run.

The TSOAE-specific DLLs that are required are TSOAE, TIRTSOCS, TIRTSODV, TIRTSOST, and TIRTSXSU. The CA Gen runtime DLLs that are required for a CA Gen generated application to run are TIRARUNT, TIRORUNT, TIRMTQBZ, TIRTERAZ, TIRPRFQZ, TIRRTLDT, TIRDEVTZ, TIRIRTRZ, TIRIURTZ, and TIRTSYSZ. In addition, the DLLs that are required for generated Block Mode Enhanced Map applications are TIRCGSPZ, TIRCHPZ, TIRCHPRZ, TIRCIIMZ, TIRCO2PZ, TIRCO2SZ, TIRCPINZ, TIRCPUIZ, TIRCVINZ, and TIRIEXZ. The TIRIEXSZ DLL is required for generated Block Mode Standard Map applications.

To test under the Application Test Facility, use the following sequence:

- Specify the Test Execution Environment
- Begin the Test
- Complete the Test and Return to CA Gen

Specify the Test Execution Environment

Specify or verify the test execution environment each time you execute a test.

The Application Test Facility panel defines the test environment. The specifications for the test execution environment include the following:

- DB2 subsystem
- Application load library

- Test SYSLIBs
- End function key assignment

DB2 Subsystem

The DB2 subsystem is the DB2 SYSID of the DB2 system to access for the test. The default for this field is the DB2 subsystem that is specified for your target environment. You can override the target environment specification on the Target Environment and Construction Libraries panel.

For testing, use a DB2 subsystem that is not used for production. This is especially important when testing changes to a program that is also running in production.

More information:

Specify Target Environment for HE (see page 77)

Application Load Library

The Application Load Library is the load library containing the load modules to test.

Test SYSLIBS

Test SYSLIBS must be specified when your application includes external routines that use dynamically called system routines. The test SYSLIBS is the libraries in which the dynamically called system routines can be found. You can specify as many as four SYSLIBS.

The external files must be allocated outside of CA Gen. A convenient way to do this is create a TSO CLIST that allocates the files. Execute the CLIST on the TSO COMMAND line:

```
TSO ex clist dataset_member_name for PDS CLISTS
TSO ex clist dataset_name for SEQ CLISTS
```

End Function Key Assignment

The End function key assignment lets you specify the key to be used to return to a clear screen and end the test session.

Be careful not to assign a function key that is used by your application system. The End function key assignment overrides function keys that are assigned within your application. For example, if your application uses PF15 to return to a master menu and you assign PF15 (the default) as the End function key, the Test Facility recognizes PF15 as an End command only. When you press PF15 from an application screen, the Test Facility does not return to the master menu.

Begin The Test

After you specify the test execution environment, you can begin testing the application immediately. To begin the test, press Enter from the Specify Test Execution Environment panel. A clear screen displays.

Online Load Module

To start the test session for an online load module, type the clear screen transaction code of the first transaction you want to test. Be sure that it is a transaction whose load module has been installed. CA Gen does not recognize the transaction code if the load module has not been installed.

Note: Begin a test of a multistep, online procedure by entering the transaction code for the first procedure step.

The Testing Facility begins simulating your application. During the test, you can use any of the application functions whose load modules were installed into the testing environment.

You can return to a clear screen from an application screen at any time. From a clear screen, you can test another transaction by entering its transaction code or you can exit the test. To exit the test, click End.

If you are not using the Debug Trace Facility, the Test Facility displays the application screens exactly as they would appear in a production environment. If you are using the Debug Trace Facility, several debug panels are displayed in addition to your application screens.

More information:

Common Testing Errors (see page 190)

Batch

To execute the CA Gen Batch applications under IEFAE the following DLLs are required for IEFAE: any of the exits listed in the batch section of User Exit Reference Guide, TSOAE, TIRTSOCS, TIRTSODV, TIRTSOST and TIRTSXSU, TIRARUNT, TIRBRUNB, TIRBRUNT, TIRORUNT, TIRBRTRZ, TIRBURTZ, TIRMTQBZ, TIRITIAZ, TIRTERBZ, TIRIRTRZ. Depending on what functions are used by the action blocks the TIRDATXZ and TIRUPPRZ also be needed.

Batch Job

To start the test session for a batch job, type the load module name as the input transaction code. Because it is an online test, we recommend that you use a small volume of data during testing.

Note: The Test Facility does not support DL/I access by external action blocks.

For batch procedures, the Batch Manager uses three special files. These files are allocated with data set names prefix.IEF.TIRxxxF. The value for prefix is specified during host installation.

After executing the test, examine the TIRMSGF and TIRERRF files to view the results. The TIRIOVF file is cleared when a procedure completes successfully.

When a procedure completes successfully, an end of job message displays on the screen. If the procedure fails, CA Gen runtime error messages are written to the screen only as when an online procedure step fails. Clear the screen before initiating another test.

Rerun a Failed Batch Job

You can restart a failed batch procedure step under the Test Facility by entering the load module name of the failed procedure step. TIRIOVF contains the import view that is passed on the previous transfer from another step or from itself. Restore files that are controlled by external action blocks.

Use the following procedure to rerun a failed batch job from the beginning while in the Test Facility:

1. Back out the changes to the DB2 tables and to any files controlled by external action blocks.
2. Run the CLEARIOV CLIST to reset the TIRIOVF file. Attempting to execute without clearing the TIRIOVF results in a fatal error because the Batch Manager expects to restart a different step. Execute the CLIST by entering TSO CLEARIOV on the TSO COMMAND line when the Specify Test Execution Environment Panel is displayed.
3. Enter the load module name of the first procedure step again.

Complete Testing and Return to CA Gen

To return to a clear screen from an application, press the End key that is specified for the test execution environment. From a clear screen, you can test another transaction by entering its transaction code or you can exit the test. To exit the test, press End.

After executing the test of a batch procedure, you can examine the TIRMSGF file to view the results. The TIRIOVF file is cleared when a procedure completes successfully. When a batch procedure completes successfully, an end of job message displays on the screen.

Abnormal Ends (ABENDS)

The Dialog Manager, a part of every CA Gen-generated online load module, handles abnormal endings, abends.

Note: For more information about handling runtime errors, see the *User Exit Reference Guide.*

If a batch procedure fails, CA Gen writes a failure message to the screen the same it does for an online step failure.

CA Gen performs these following steps to process a runtime error in an online load module:

1. The Dialog Manager performs all necessary rollbacks.

   CA Gen displays an error screen that lists the appropriate CA Gen runtime error messages, similar to this illustration:

```
TIRM030E:   APPLICATION FAILED - UPDATES HAVE BEEN BACKED OUT
TIRM042E:   *** PROFILE ERROR DETECTED ***  9 0007
--------------------------------------------------------------------------------
DSNT408I SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN
         UNAVAILABLE RESOURCE. REASON 00E30305, TYPE OF RESOURCE
         00000801, AND RESOURCE NAME D80H.TIRPROFD.1898BDDF1A94C6A2
DSNT418I SQLSTATE   = 57011 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXEAAL SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -150  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'FFFFFF6A'  X'00000000'  X'00000000'
         X'FFFFFFFF'  X'00000000'  X'00000000' SQL DIAGNOSTIC
         INFORMATION
--------------------------------------------------------------------------------
TIRM046E:   *** TRANSACTION PROCESSING TERMINATED  IEFC1
TIRM044E:   *** PRESS PA2 TO CONTINUE
***
```

2. When you press PA2, the NEXT PAGE key, from the error message screen, CA Gen displays the last screen for the transaction being processed when the error occurred. For a batch procedure, CA Gen displays the following message:

   ```
   No page available
   ```

CA Gen recovers all data in the import views at the time the error occurred. CA Gen recovers user input and displays it on the screen. The screen fields that are only in the export view may or may not be populated, depending on when the error occurred.

The application remains active.

An error message displays in the system error message area that is defined for the screen. This message is distinct from the runtime error messages that are displayed on the error message screen. The default error message is:
`SYSTEM ERROR OCCURRED - CONTACT SUPPORT.`

To clear the screen to try another transaction code, press the End key that is defined for your test environment.

If a runtime error occurs in a batch load module, CA Gen performs the following processing:

1. The Batch Manager performs all necessary rollbacks.
2. CA Gen displays an error screen that lists the appropriate CA Gen runtime error messages.
3. You can clear the screen to try another load module by pressing the End key that is defined for your environment.

Batch Testing

Batch applications can execute and be tested under the Application Test Facility or by running under JES.

The CA Gen runtime modules required by CA Gen generated batch applications running under JES are the following: any of the exits that are listed in the batch section of User Exit Reference Guide, TIRARUNT, TIRBRUNB, TIRORUNT, TIRBRTRZ, TIRBURTZ, TIRMTQBZ, TIRTERBZ, TIRRETCZ, TIRIOVFI. If your batch application is IMS DLIBATCH or IMS BMP, the TIRBRUNI DLL is needed. Depending on what functions are used by the action blocks the TIRDATXZ and TIRUPPRZ also be needed. These must be included in the STEPLIB DD.

Execute Test Under CICS

Perform the following activities to test under the CICS Debug Test Facility:

- Install your DPS application
- Install the Trace Facility Display Interface (TFDI)

- Install the Debug Trace Facility (DTF)

All load modules must be installed in the CICS region. The TFDI and DTF modules are found in the CA Gen load library.

The CA Gen runtime DLLs that are required for all types of generated CICS applications targeting DB2 are TIRORUNC, TIRMTQBZ, TIRTERAZ, TIRARUNC, TIRSDTNZ, TIRCDPTZ, TIRCRTRZ, TIRCSYSZ, TIRCTIAZ, TIRCURTZ, TIRCUSRZ, TIRDATXZ, TIRDEVCZ, TIRDLCTZ, TIRHELPZ, TIRQCNTZ, TIRSECRZ, TIRUPPRZ, TIRYYXZ, and TIRRTLDC. In addition, the DLLs required for generated Block Mode Enhanced Map applications are TIRCGSPZ, TIRCHPZ, TIRCHPRZ, TIRCIIMZ, TIRCO2PZ, TIRCO2SZ, TIRCPINZ, TIRCPUIZ, TIRCVINZ, and TIRIEXZ. The TIRIEXSZ DLL is required for generated Block Mode Standard Map applications.

For more information about the list of required runtime DLLs to execute, see the z/OS Installation Guide.

CICS DTF User Interface

The DTF user interface control the trace facility by specifying the transaction to place in the trace mode. DTF manipulates the debug control queue by adding or removing records, thereby placing the transactions in or out of the trace mode. While the Debug Trace Facility is on, all executed procedure steps and action blocks generated with the debug option are traced.

Starting DTF

To start the Debug Trace Facility, execute a CICS transaction. The suggested name of this transaction is DTF. However, this name is changed when installing CA Gen. Contact your system support staff to determine the transaction name at your site. The documentation assumes that it is DTF.

The DTF command has the following format:

DTF *command transaction* [*user ID*] [*terminal ID*]

The arguments on the DTF transaction are positional. The command and transaction arguments are required. Although user ID and terminal ID are optional, include the user ID to specify the terminal ID because they are positional.

DTF Commands

command can be any of the following:

ON

Turns on tracing for one or more transactions and creates a record in the debug control queue to cause tracing to begin when the selected transaction and user ID occur in a trace call.

OFF

Turns off tracing for one transaction or user, and removes a record from the debug control queue to prevent tracing. TDFI removes the record when you stop tracing at the debug terminal.

SHOW

Lists the records in the debug control queue.

PURGE

Deletes the records from the debug control queue and turns off tracing.

DTF Parameters

transaction Code

The name of the transaction to trace. Use an asterisk (*) as a wild card to trace all transactions for a specified user ID. When you specify an asterisk (*), specify an explicit, non-wild card user ID.

user ID

Restricts tracing to one user. For server modules, this is the ID passed in the Common Format Buffer (CFB) header. For block mode transactions, log in with CESN and provide an ID.

If no user ID is supplied, DTF stores an asterisk (*) as a wild card in the debug control to indicate that DTF is to trace all users of the specified transaction. If a wild card transaction code is specified, an explicit (non-wild card) user ID must be specified.

Default: Trace all users of the transaction.

Terminal ID

Specifies the debug terminal to use to display the trace information. Default: The terminal ID for the terminal on which you execute the DTF transaction.

Example DTF Command

This example starts the server trace for transaction GL1 and user N2A3 with the debug terminal at N23A:

```
DTF ON GL1 N2A3 N23A
```

This example traces all transactions for User ID N2A3:

```
DTF ON * N2A3
```

This example traces transaction GL1 from any User ID

```
DTF ON GL1 *
```

DTF Error Codes

DTF, TFDI, or Runtime TIRCRUNC issues the following errors or abend codes:

Error	Message
tf1	START Transaction TFDI failed
tf2	WAIT EVENT failed
tf4	READ TSQ DTFRxxxx failed
tf5	QIDERR or ITEMERR for TSQ COMPDTF
tf6	GETMAIN failed
tf7	WRITE TSQ DTFIterm failed
tf8	FREEMAIN failed
tf9	WRITE TSQ DTFXIterm failed
tf13	FREEMAIN failed
tf14	WRITE TSQ COMPDTF failed
tf23	READ TSQ COMPDTF failed

The following errors are issued for TSQ TIRTxxxx:

Error	Message
tf90	WRITE TSQ TIRTxxxx failed
tf91	REWRITE TSQ TIRTxxxx failed
tf92	REWRITE TSQ TIRTxxxx ITEMERR
tf93	WRITE TSQ TIRTxxxx (when recovering from ITEMERR) failed

The following errors are issued for TFDI:

Error	Message
DI0	GETMAIN failed
DI2	READ TSQ DTFIterm failed
DI4	WRITE TSQ DTFRxxxx failed
DI6	Error sending message ‘Press ENTER to remain in Debug mode’
DI7	Error sending message ‘To Exit press CLEAR’

Error	Message
DI0	GETMAIN failed
DC7	Error sending message to terminal
DI8	WRITE TSQ DTFIterm failed
DI9	DELETE TSQ DTFIterm failed
DI12	READ TSQ DTFIterm failed

Common Testing Errors

The following is a list of some of the common errors that are made during testing.

- All load module components that are needed for the test must be generated and installed. When testing the flow between two procedure steps, be sure that both procedure steps were installed.
- If you attempt to access a load module that was not generated, the test fails. If the attempt is from within the Application Test Facility, the error message is:

  `CSV0031 REQUESTED MODULE module_name NOT FOUND`
- Press End, as defined in the testing environment for your application, to return to a clear screen.
- To test any part of a multi-step procedure, always generate the first procedure step and the procedure steps you intend to test. This is because the first procedure step provides the access to all other procedure steps in the procedure.
- You can only use the Debug Trace Facility with those components for which you generated debug support. You can test components without using the Debug Trace Facility.
- All load module components must be generated before the load module can be installed.
- The install fails when any component was not generated. The link-edit step has unresolved references when the application was statically linked. The application fails execution with U4038 abends when the application uses the Dynamically Link option.

- A load module must be installed before it can be tested. Ensure that it is installed using the correct installation parameters.
  - For the Application Test Facility, be sure to specify IEFAE as your target environment, or on the Host, select the Target TSO test facility option.
  - For CICS, specify CICS as your target environment. When a component marked for Compatibility is unavailable at execution time, the application attempting to dynamically call the Compatibility module abends with U03999. CA Gen writes a message to the CEEMSG DD indicating the name of the missing module.
- When testing a load module within the Application Test Facility that uses an external action block that contains IMS/CICS environment calls or DL/I calls, be sure to stub out the IMS, CICS, or DL/I calls within the external action block or the external action block itself. Failure to do so, causes the test to abend with a System 0C1 or 0C4.
- For Application Test Facility testing, ensure your ISPF logon CLIST points to a table library with the TIAECMDS command table. Failure to do so causes the PF3 End command to fail.
- CA Gen includes the TIAECMDS. The library in which it is stored is determined during CA Gen installation.

  Note: For information about installation, see the *z/OS Installation Guide*.
- If you experience problems allocating the TRANMAP file, verify that it exists and check the contents. If the file seems corrupted, delete it and start the testing facility again. CA Gen generates a new TRANMAP file for each user ID when one does not exist.

Making Changes to an Application

After testing an application, change it. Always change the related action diagrams, views, and screens, regenerate the source, and reinstall the application.

Important! Do not modify the generated source code.

If you change a diagram that is the source of information for a generated load module, regenerate one or more of the load module components. The type and scope of the change determine which load module components are regenerated. Occasionally, regenerate the complete load module. More often, changes that affect specific components require that you regenerate those components.

Intelligent Regeneration automates the process of determining which components are regenerated based on recent changes in the model and regenerate only those portions of the model that is affected, but still lets you control the regeneration process. Examine the Intelligent Regeneration reports before manually regenerating components.

If you know what changes were made to the diagrams since the load module was generated and which components are affected by the change, some guidelines exist that you can use to decide what is regenerated. If you are unsure of the changes that are made or their impact, it is best to regenerate the entire load module.

More information:

Intelligent Regeneration (see page 245)

Debug Trace Facility in the HE Environment

The Debug Trace Facility operates the same under CICS and the Application Test Facility.

Note: To use the Debug Trace Facility, select debug support when you generate your application.

Trace Action Block Calls

The Action Block Call Trace panel shows the active procedure step and the action blocks that are called by the procedure step or its action blocks. The panel is not static. It is a dynamic listing that shows the sequence of calls as they are issued by the procedure step or its action blocks. The panel displays each time that you enter or return to a procedure step or an action block that was generated with debug support. It can also be requested from the Action Diagram Trace panel.

```
-------------------------- APPLICATION DEVELOPMENT --------------------------
                           Action Block Call Trace
 COMMAND===>                                                SCROLL===> HALF

                 Action Block Name                At Stmt #
                 -------------------------------  ---------
                 MENU                             DIALOG MGR
 ENTERING===>    ABBIL1                           35
                 ABBIL2                           47
```

For each call, the panel shows the name of the action block or procedure step and the statement number from which it was called. In the figure above the action block ABBIL1 was called by statement number 35 of the procedure step MENU. The action block ABBIL2 was called by statement number 47 of ABBIL1. Procedure step MENU was called by the Dialog Manager.

The Action Block Call Trace panel can be thought of as a stack that shows the hierarchy of action block calls within a procedure step. When a call is made, an entry is added to the stack and displayed on the screen. When you return from the call, the last entry is deleted from the stack and from the screen. When you transfer to another procedure, the stack is cleared and a new panel begins.

To begin the execution of the procedure step or action block, press the Enter key. If the procedure step or action block was generated with the Debug option, the Action Diagram Trace panel displays.

The TRACE command lets you turn the action diagram trace on and off. If TRACE is OFF, no action diagram trace panels are displayed. To resume the display of action diagrams, type TRACE ON or T ON on the COMMAND line and press Enter.

Action Diagram Trace Panel

The Action Diagram Trace panel lets you trace the logic of your application at runtime. The Debug Trace Facility shows the action diagram for the procedure step or action block executing. The panel displays only if the procedure step or action block was generated with debug support.

```
-------------------------- APPLICATION DEVELOPMENT ------------- ENTER OPTION
                             MENU Action Diagram Trace
COMMAND===>                                                  SCROLL===> HALF

                 Action Block Name                 At Stmt #
                 --------------------------------  ---------
                 MENU                              DIALOG MGR
 ENTERING===>    ABBIL1                            35
                 ABBIL2                            47
```

As your application executes, the corresponding action diagram statements are highlighted on the panel. The Debug Trace Facility highlights the statement about to be executed and pauses. Press Enter to continue the execution.

The Action Diagram Trace panel supports the following commands to navigate in the action diagram:

HELP, H, or ISPF Key PF1

Displays the online help

UP or ISPF Key PF7

Accepts an M parameter, or number of lines

DOWN or ISPF Key PF8

Accepts an M parameter, or number of lines

LOCATE or L

Accepts the statement number as a parameter

FIND or F

Accepts a string or enclosed in single quotes, as in *'string'*

RFIND, R, or ISPF key PF5

Repeats the previously requested FIND command.

SKIP or SK

Bypasses executing the highlighted statement.

EXIT or X

Immediately returns the currently executing action block to the program that called it.

TRACE or T

Toggles the trace ON or OFF <UNTIL <GT|GE> *stmt no*>

When TRACE is OFF, the action diagram trace panel for this action diagram is not displayed. To resume the action diagram display, specify TRACE ON from an Action Block Call Trace Panel.

You can turn off trace until you reach a specific statement in the action diagram using the Trace Off Until Statement_number command. For example, to turn the trace off until you reach the statement number 12, use the command:

```
TRACE OFF UNTIL 12
```

To turn off trace until you reach at least statement number 15, use the command:

```
T OFF U GE 15
```

This command is useful to turn off trace, perhaps during a READ EACH, but you are not certain that statement 15 is executed. The tracing resumes on the next statement number that is 15 or greater.

Use the following commands to control the panel appearance of the panel.

HILIGHT or HI

Accepts the parameters REVERSE, BLINK, NORMAL, UNDERLINE

Highlights the statement about to execute as blinking, reverse video, underlined, or normal.

Default: reverse video.

COLOR or C

RED, BLUE, GREEN, WHITE, TURQ, PINK, YELLOW

Controls the color of the highlighted statement.

COLOR TEXT or C TEXT

RED, BLUE, GREEN, WHITE, TURQ, PINK, YELLOW

Controls the color of all statements other than the highlighted statement.

Access to Other Debug Panels

The Action Diagram Trace Panel also includes commands that let you access the other debug panels:

DISPLAY CALL

Displays the Action Block Call Trace Panel

DISPLAY *vvvv*

Displays the requested Action Diagram View Display Panel (*vvvv* = IMPORT, EXPORT, LOCAL, or ENTITY)

DISPLAY SYSTEM

Displays the Action Diagram System View Display Panel

To return to the Action Diagram Trace Panel from these three panels, press Enter.

Display or Modify Views

The Action Diagram View Display lets you see the current import, export, local, or entity action views for an action diagram. The display shows the detailed information about the view and its predicates (attributes), including the current predicate values. The display also includes commands to expand repeating group views and modify predicate values.

Access the display from the Action Diagram Trace panel, using the command:

DISPLAY *vvvv*

where *vvvv* = IMPORT, EXPORT, LOCAL, or ENTITY.

The following illustration shows a sample EXPORT view display.

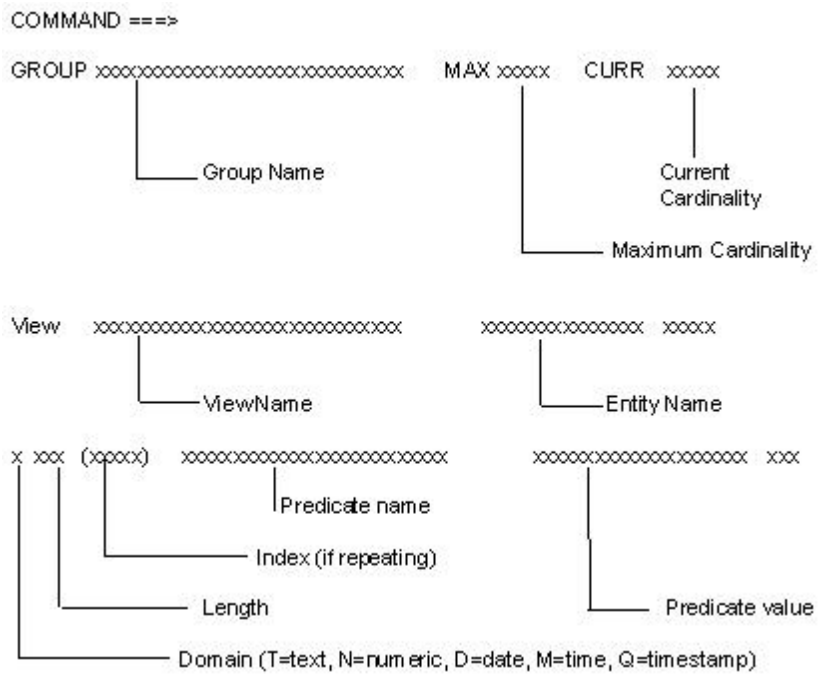
```
---------------------- APPLICATION DEVELOPMENT --------------------------
                    MENU Action Diagram EXPORT View Display

 COMMAND===>                                                 SCROLL===> HALF
 S                                          1...+....10...+....20...+....30..
   VIEW  TEMPOUT SELECTION
   N 001  NUMBER                        =   +0
   VIEW  SENT EMPLOYEE
   N 006  NUMBER                        =   +000000
   VIEW  SENT DEPARTMENT
   N 004  NUMBER                        =   +0000
   VIEW  SENT DIVISION
   N 003  NUMBER                        =   +000
   ******  BOTTOM OF VIEW  *********



 Sel codes:  E:Expand Repeating Group Views   M:Modify data value
```

For entity views that are not a part of a group, the Action Diagram View Display Panel displays the name of the view followed by its predicates and predicate values. The format of the display information is shown in the following illustration:



For non-repeating group views, the panel displays a line identifying the group, immediately followed by the entity views within the group and the predicates and predicate values for each entity view.

For repeating group views, the panel displays a single line with the name of the group and its maximum and current cardinality. To see the views within the group, expand the repeating group view. You cannot expand a repeating group whose current cardinality is zero. The following message displays at the top right of the panel:

```
Repeating Group is Empty
```

You cannot expand a non-repeating group.

To expand a repeating group, type E next to the group and press Enter. The repeating group expands to show the entity views defined within the group.

Expanding a nested repeating group view shows the entity views and the name of the next repeating group. Nested repeating group views let you expand the repeating group until you reach the inner-most nested group. Keep typing E next to the label GROUP and pressing Enter until you reach the inner-most level.

To expand an entity view within a repeating group, type E next to the entity view and press Enter. The repeating predicates of the entity view are displayed with an index identifying the predicate occurrence number and the predicate value. When you expand a repeating group that has only one entity view, the repeating predicates are immediately displayed.

The current value for each predicate (attribute) is displayed on the panel. You can change predicate values. To modify a predicate value, type M next to the predicate to be modified, use the Tab key to position the cursor to the predicate value and enter the new value. When you exit the panel, the new value is accepted. Processing continues using the new value.

Modifying a predicate value that exceeds the screen size is a multi-step process:

1. Modify value on screen as described previously.
2. Press PF11 to scroll to the right. The remainder of the predicate value is displayed.
3. Modify the remainder.

Several commands help navigate in the panel.

HELP, H, or ISPF Key PF1

Displays the online help.

UP or ISPF Key PF7

Moves the cursor up number of lines

Accepts a parameter of M or number of lines

DOWN or ISPF Key PF8

Moves the cursor down a number of lines

Accepts a parameter of M or number of lines

FIND or F

Locates a string.

Accepts a string or a string that is enclosed in single quotes, as in '*string*'

RFIND, R, or ISPF Key PF5

Repeats the previously requested FIND command.

RIGHT *nnn* or ISPF Key PF11

Scrolls the predicate value portion of the panel to the right. Use RIGHT to view the predicate values that exceed 32 characters.

Accepts a number to indicate the number of characters to scroll.

Default: 10 characters

LEFT *nnn* or ISPF Key PF10

Scrolls the predicate value portion of the panel to the left. Use LEFT to view the predicate values that exceed 32 characters.

Accepts a number to indicate the number of characters to scroll.

Default: 10 characters

Display System Variables

The Action Diagram System View Display panel lets you see the current runtime values of the system variables. This panel is accessed with the DISPLAY SYSTEM command. The variables are the Special Attributes available to you during Business System Design. The variables are generated and maintained by the CA Gen and can be used by your application system. This panel displays the current values of the system variables, whether you have chosen to use them in your application.

You can change the values of the system variables that are displayed on this panel except the current date and time.

```
 ------------------------- APPLICATION DEVELOPMENT --------------------------
                    MENU Action Diagram SYSTEM View Display

COMMAND===>

TRANCODE     ===> MENU         USER ID      ===> userID

TERMINAL ID  ===> termID       PRINTER ID   ===>

CURRENT DATE ===> mm/dd/yy     CURRENT TIME ===> hh:mm:ss

RETRY COUNT  ===> 0            RETRY LIMIT  ===> 10

SYSTEM COMMAND:
===> PROCESS

INFORMATION MESSAGE:
===>
```

TRANCODE

Transaction code being executed.

USER ID

Uniquely identifies the terminal operator for the transaction.

TERMINAL ID

Uniquely identifies the terminal from which the transaction was requested.

PRINTER ID

Uniquely identifies a printing device for the transaction.

CURRENT DATE

Current system date.

CURRENT TIME

Current system time.

SYSTEM COMMAND

Application system command, as defined in a COMMAND IS statement in an action diagram.

INFORMATION MESSAGE

This is the message area that displays errors, warnings, instructions, and messages to the user. Error messages are set in a procedure step or originate in the CA Gen software.

To change the value of a system variable, use the Tab key to move the cursor to the field and type in a new value. The new value takes effect when you exit the panel.

To exit the panel, press Enter.

Application Migration in HE

There are different application migration requirements depending on the CA Gen release that the application is migrating from and the type of linkage the application uses to invoke the various components of that application.

To aid migration from releases of Gen before release 7, use the Compatibility setting for the Dynamically Link Packaging property. Its purpose is to:

- Enable CA Gen 8.5 applications (which are built as DLLs) to call components that are built as non-DLL load modules.
- Optionally, build components that cannot be DLLs as non-DLLs that use the same z/OS runtime as CA Gen 8.5 DLL applications.

Note: Since the Compatibility feature was not available in AllFusion Gen 7 or AllFusion Gen 7.5, components that are previously migrated to these releases became DLLs. Migration information from these releases to CA Gen 8.5 does not mention Compatibility.

Note: For more information about migrating to CA Gen 8.5, see the *Release Notes*.

Migrating from CA Gen 8

The CA Gen 8.5 z/OS Runtime DLLs must be deployed to the target execution environment and must supersede the CA Gen 8 Runtimes. CA Gen 8.5 generated applications and runtime DLLs must reside in the PDSE libraries.

Migrating from AllFusion Gen 7.6

The CA Gen 8.5 z/OS Runtime DLLs must be deployed to the target execution environment and must supersede the AllFusion Gen 7.6 Runtimes. CA Gen 8.5 generated applications and runtime DLLs must reside in the PDSE libraries.

Migrating from AllFusion Gen 7 .5

All AllFusion Gen 7.5 application components were installed to resolve addressability to the runtimes residing in DLLs. Since all AllFusion Gen 7.5 applications are DLLs, no activities are required to migrate an application from AllFusion Gen 7.5 to CA Gen 8.5.

The CA Gen 8.5 z/OS Runtime DLLs must be deployed to the target execution environment and must supersede the runtimes that is provided with AllFusion Gen 7.5. CA Gen 8.5 generated applications and runtime DLLs must reside in the PDSE libraries.

Gen--Migrating from AllFusion Gen Release 7

Compiler options that are changed in AllFusion Gen 7 to enable the creation and use of DLLs. A small portion of the runtimes were built as DLLs in AllFusion Gen 7 with most of the remainder becoming DLLs in AllFusion Gen 7.5. The CA Gen 8.5 applications must be compiled using options that enable DLL support to use the runtime DLLs.

When migrating from AllFusion Gen 7 to CA Gen 8.5, evaluate how the AllFusion Gen 7 application was built, if all or some of the components were compiled in AllFusion Gen 7.

When all the application components were compiled in AllFusion Gen 7, reinstall all the existing code to re-link the application. The re-link step resolves the addressability of the runtimes located in the DLLs. Ensure that runtimes from previous releases are not inadvertently picked up from the SYSLIB concatenation through auto-call. The CA Gen 8.5 load library contains all the required runtimes. Load libraries from previous releases must not be included in the SYSLIB concatenation.

When only some application components were compiled in AllFusion Gen 7, recompile all the components being migrated that were not previously compiled at the AllFusion Gen 7 level and re-link all components. A recompile is required to ensure that the generated components are built with DLL support and a re-link is required to resolve the addressability of the runtimes that is located in the DLLs. Ensure that only CA Gen 8.5 runtimes are included from the SYSLIB concatenation through auto-call. The CA Gen 8.5 load library contains all the required runtimes. Load libraries from previous releases must not be included in the SYSLIB concatenation.

To re-link an application without recompiling it, select the Link-edit load module option or the Link-edit dynamically linked modules option. Set these options the same as used when building the original AllFusion Gen 7 application. CA Gen 8.5 generated applications and runtime DLLs must reside in the PDSE libraries.

Migrating from Release 6.5 and Earlier Releases

When migrating from releases earlier than AllFusion Gen 7 to CA Gen Release 8.5, evaluate the application components that can become DLLs and the application components that must remain as non-DLL load modules.

Note: It is possible that feature enhancements offered in future releases of CA Gen require that routines reside in a DLL to use those features.

Some of the application components that can become DLLs are regenerated and all the components are recompiled, re-linked, and re-installed regardless of the application type, (block mode or servers), or the type of linkage, (static or dynamic program call), used between application components.

The following list describes the components that must be regenerated or reinstalled-for each application type. Reinstallation implies recompile, re-link and if applicable bind.

RI Triggers

No components require regeneration. All components require re-installation.

Batch

Batch Manager requires regeneration. All components require re-installation.

TSO block mode, standard map

Dialog Manager requires regeneration. All components require re-installation.

TSO block mode, enhanced map

Dialog Manager and Map require regeneration. All components require re-installation.

IMS block mode, standard map

Dialog Manager requires regeneration. All components require re-installation.

IMS block mode, enhanced map

Dialog Manager and Map require regeneration. All components require re-installation.

CICS block mode, standard map

Dialog Manager requires regeneration. All components require re-installation.

CICS block mode, enhanced map

Dialog Manager and Map require regeneration. All components require re-installation.

IMS server

Server Manager requires regeneration. All components require re-installation.

CICS server

Server Manager requires regeneration. All components require re-installation.

Components that remain as non-DLL load modules must use the z/OS Dynamic Program Call Compatibility feature.

This feature allows a phased migration of applications that are built by releases of AllFusion Gen before Release 7, that use dynamically called procedure steps, screens, or action blocks (including EABs). The Compatibility setting causes the caller of a module marked for compatibility to always be regenerated and reinstalled, to include a call to the runtime routine that handles the call to the non-DLL load module. The option Process modules that are marked for Compatibility determine which components are rebuilt as CA Gen Release 8.5 non-DLL load modules. The option Process modules that are marked for Compatibility can be selected to regenerate and reinstall components or only to reinstall them. The reinstall is required to enable the non-DLL load modules to use the CA Gen Release 8.5 DLL runtimes.

RI triggers and action blocks that are statically called by modules that are marked for Compatibility must be built using the NODLL compiler option. If these RI triggers and action blocks are also used in CA Gen applications built as DLLs, they must be compiled using the DLL option.

Selecting the Process modules marked for Compatibility option causes the RI triggers and action blocks that are statically called by modules that are marked for Compatibility to be generated and precompiled once, but compiled twice. Separate the libraries are provided in the target environment to hold the separate NCAL modules resulting from the two compile steps.

Applications that run under TSOAE and contain modules that are marked for Compatibility cannot dynamically call modules that are built with a release earlier than AllFusion Gen 7.6.

Enhanced Map Block Mode applications containing screens that are marked for Compatibility cannot dynamically call screen managers that are built with a release before AllFusion Gen 7.6.

The option Force compile of all components can be used to recompile and relink all application components, except for EABs or user exits which must be done separately.

Important! Ensure that runtimes from previous releases are not inadvertently picked up from the SYSLIB concatenation through an auto-call. The CA Gen Release 8.5 load library contains all the required runtimes. Load libraries from previous releases must not be included in the SYSLIB concatenation.

EAB Migration

When an External Action Block has its associated Dynamically Linked Packaging property set to Yes, migrating to CA Gen 8.5 requires recompiling the EAB and reinstalling it as a DLL. If the EAB dynamically calls other user programs, those user programs must also be built as DLLs.

When an External Action Block has its associated Dynamically Linked Packaging property set to Compatibility, the option Process modules that are marked for Compatibility determine if the EAB must be rebuilt as a z/OS non-DLL load module or remain as built by the releases of CA Gen earlier to release 7. If the EAB dynamically calls other user programs, those user programs must also be non-DLLs.

How an External Action Block that has its associated Dynamically Linked Packaging property set to No, so it is statically linked into the calling application, must be compiled depends on the type of application that calls it. When the calling application is marked for Compatibility the EAB must be built using the NODLL compiler option and must be placed in a library in the Compatibility EAB load libraries panel. When the calling application is not marked for Compatibility the EAB must be built using the DLL compiler option and must be placed in a library in the EAB Load libraries panel. When called by both application types, two copies of the EAB must be built and each placed in the appropriate load library.

More information:

External Action Blocks (see page 207)

z/OS Runtime User Exits

In CA Gen Release 8.5, user exits reside in separate unique DLLs. Users migrating to CA Gen Release 8.5 must rebuild and redeploy those user exits they have customized in a release of CA Gen prior to Release 8.5 as in releases before AllFusion Gen 7.6 or into the Gen runtime DLLs as in AllFusion Gen 7.6. The user exits are no longer linked into each generated application. All the modified user exits must be rebuilt so their specific DLLs can be updated.

For CA Gen Release 8.5 all but the TCPIP Direct Connect User Exits reside in the CEHBSAMP data set. The User Exits used by the CICS Sockets Server Listener can be found in the CEG8SAMP data set while the User Exits for the TCPIP Direct Connect for IMS are in the CEG9SAMP dataset.

The JCL procedures that build the user exits and their specific DLLs are provided in relevant CE*SAMP data sets in members MKUEXITS, MKUECTCP, and MKUEITCP respectively. The JCL to rebuild the User Exits used by the C runtimes for Codepage translation reside in member MKCRUN in the CEHBSAMP data set.

Important! Ensure that the user exits used from previous releases are not inadvertently picked up from the SYSLIB concatenation through auto-call. The CA Gen Release 8 user exits must be updated and deployed in their respective DLLs.

Note: For more information about User Exits and their associated JCL procedures, see the *User Exit Reference Guide*.

Chapter 10: External Action Blocks

External Action Blocks in HE

External Action Blocks access processing logic that is contained in subroutines that are not generated by CA Gen. You can create the external subroutine to fit a specific need, or it can exist previous to development of your application system. You create and compile the subroutine independent of CA Gen.

For example, you may want your application system to use a standard date manipulation or security routine that is defined specifically for your organization. Or the application may need to access databases that CA Gen toolsets do not support, such as DL/I,or that contain entities (tables) not defined in the model's Data Model.

External Action Blocks are part of a procedure step and have the following characteristics:

- Called from a procedure step or from an action block
- Allow the view matching
- Return to the calling procedure step or action block when completed

CA Gen External Action Blocks are created using the workstation Design Toolset. An External Action Block contains no logic of its own. It contains only an EXTERNAL statement. The External Action Block is called by a USE statement in a procedure step or action block that is also created with the Design Toolset.

When you generate the External Action Block using the Host Encyclopedia Construction Tools, a program skeleton that is called a stub is created that includes much of the working storage and linkage for a framework for the External Action Block. External action block stubs are generated once, then modified by the user, and later maintained by hand.

The External Action Block is a modified stub that accesses or contains an external subroutine. Modify the stub by adding a call to the external subroutine or the subroutine logic itself.

The External Subroutine is the non-CA Gen program logic or transaction you want to access from the CA Gen generated application.

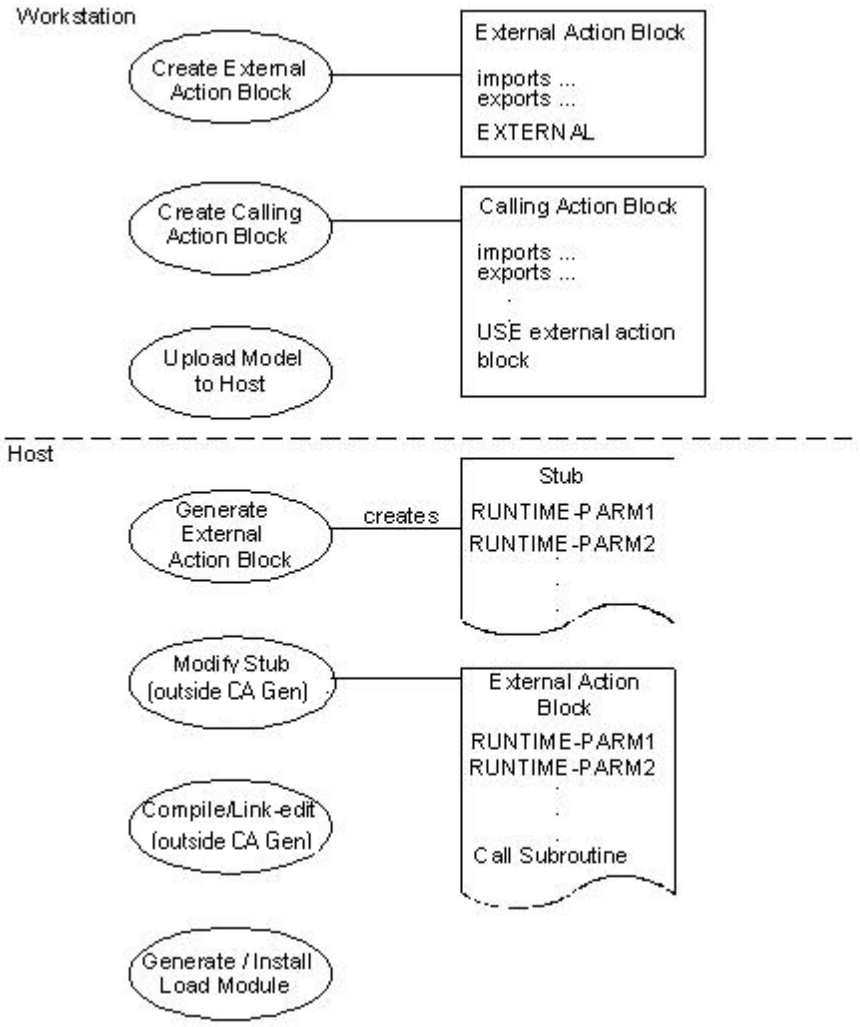
Prerequisites

Complete the following steps before you can use the Host Encyclopedia Construction Tools to generate External Action Blocks:

- Create the Data Model and action diagrams for the calling procedure step or action block and the CA Gen External Action Block.
- Complete the view matching between the calling and called action blocks. This is the information that is passed to the External Action Block.
- Upload the model to the Host Encyclopedia.

Implement External Action Blocks

The procedure you follow to create and to implement External Action Blocks includes steps on the workstation and the host. The following figure shows the overall process flow.



Workstation Tasks

Use the workstation toolsets to:

- Create the CA Gen External Action Block using the Action Diagramming tool in the Design Toolset.

- Create the CA Gen procedure step or action block that calls the External Action Block by name with a USE statement. Perform the view matching is associated with the USE statement.
- Use the Action Diagramming tool in the Design Toolset.
- Upload the model to the Host Encyclopedia.

Host Tasks

Use the Host Encyclopedia Construction tools to:

- Define the target environment and the construction libraries using the Host Encyclopedia Construction tool.
- Be sure to define the External Action Block libraries that contain the load module for the External Action Block and associated DBRMs.
- Define the CA Gen load module packaging.
- Provide CA Gen with a member name for the stub. This is the name that is used by the procedure step or action block to call the external subroutine.
- CA Gen automatically prompts you to supply member names the first time a system generation or testing facility function is requested for a load module. It also provides a function that allows you to change a member name.
- Generate the External Action Block stub. The stub defines the linkage between the CA Gen procedure step and the External Action Block. This includes the import and export views.
- Outside of CA Gen, modify the External Action Block stub.
- Copy the generated stub to a separate library before modifying it. This prevents an accidental overwrite of the stub. Add the logic to meet the specific requirements of your external subroutine. You use the modified stub to call an existing external subroutine or you include the subroutine logic in the stub.
- After modifying the stub, it becomes the source module for the External Action Block. Compile and link it before installing the load module.

  Note: If you insert the logic for the external subroutine into the stub, the CA Gen does not preserve the changes during regeneration. Copy changes into the new External Action Block stub that is created when the load module is regenerated. For this reason, it is recommended that you call the external subroutine from the stub rather than to insert the logic itself.
- Compile and link edit the external subroutine and the modified External Action Block stub (outside CA Gen). Use the appropriate compiler option for the application type.

- If the external subroutine uses DB2, identify the DB2 DBRMs associated with the external subroutine. Use the option to resolve an External Action Block in the Load Module Expansion panel to identify the DB2 DBRMs.
- Use the system generation tools to generate and install the applications system.
- Use the testing facility to test the application system's functionality.

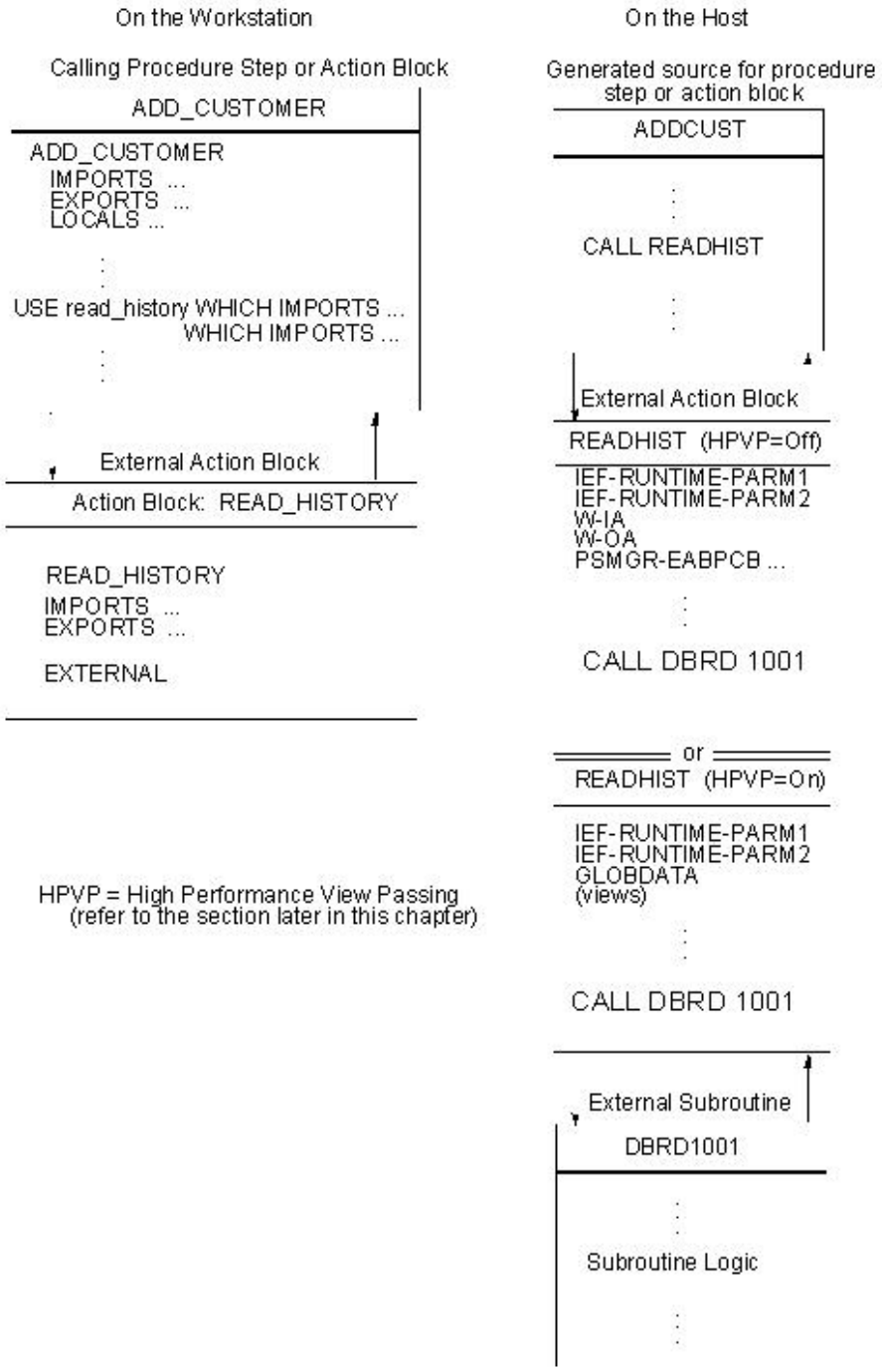
  Note: The CA Gen testing facility does not support external logic that uses DL/I, IMS or CICS calls. If your CA Gen application uses this type of logic, create the procedures steps with references to the external logic disabled. Then generate and test the CA Gen load module. After making changes to the load module, enable the USE logic to your action diagrams and regenerate the affected load modules. Alternatively, if your application target is CICS, you test your application using the Debug Trace Facility under CICS. This allows you to test your External Action Block logic.
- After you finish testing and make the changes, regenerate parts of the application and/or reinstall the application using a different TP monitor or other environment parameters.
- Move the load modules and DB2 DBRMs to production libraries and bind on the production DB2 subsystem. These activities are performed manually outside of CA Gen.

More information:

Load Module Packaging in HE (see page 107)
Specify Target Environment for HE (see page 77)
Testing Applications (see page 177)
Supplying the Member Name (see page 225)

External Action Block as Interface to Subroutine

The External Action Block can function as an interface to the external subroutine. Place a CALL statement in the External Action Block stub to access the external subroutine. The following illustration shows this scenario in more detail. The diagram applies to online and batch procedure steps.



In the example, the calling procedure step ADD_CUSTOMER access the customer history database using an external subroutine, DBRD1001. The External Action Block READ_HISTORY defines the views of the CA Gen data that is needed by DBRD1001 and matches them with the views from ADD_CUSTOMER.

The generated source for the procedure step is named ADDCUST in the example. It contains a statement that calls READHIST.

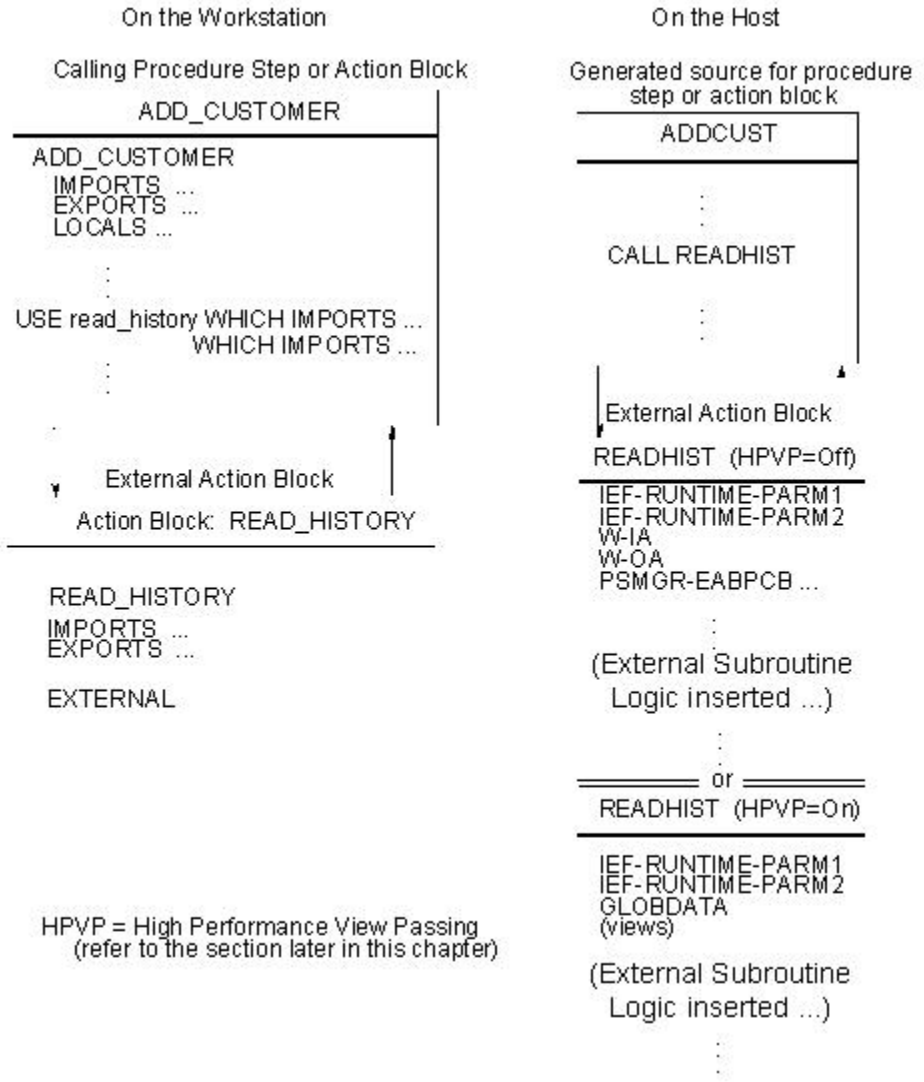
The External Action Block READHIST is the stub that is generated by CA Gen and modified by an application developer. READHIST uses the data that is passed from ADDCUST to call DBRD1001. READHIST reformats the data division components that are created from the CA Gen views to the structure needed by DBRD1001.

Note: The EXTERNAL statement identifies READ_HISTORY as an External Action Block from which CA Gen generates a stub. The modified stub uses the first eight characters of the External Action Block name, READHIST, as the default member name. The READHIST stub uses the import and export view that is defined for the READ_HISTORY External Action Block as its input and output parameters. The information in these parameters is used to create the calling interface to the DBRD1001 subroutine.

External Action Block with Embedded Subroutine

You can embed the entire logic of the subroutine in the External Action Block. This illustration to the right shows the processing flow for an External Action Block that contains the external subroutine logic.

In the following figure, the calling procedure step ADD_CUSTOMER access the customer history database using an external subroutine. The External Action Block READ_HISTORY defines the views of CA Gen data that is needed by the embedded subroutine, and matches them with the views from ADD_CUSTOMER. The External Action Block READHIST is the CA Gen generated stub that is modified to include all of the logic for the subroutine. READHIST uses the data that is passed from the generated procedure step ADDCUST to process.



Load Module Structure

Executable load modules for an application are created during Construction. The system generation tools create a COBOL source module for the procedure step or action block and then compile the source module. The resultant load module is stored in a library you specify. The name of this library match the library name that you gave to the internal libraries, that are the Internal Generated Source Code library and Internal Static non-DLL NCAL Modules library for action blocks that are statically called by Compatibility modules and Internal NCAL Load Module library for all others.

When you compile and link the modified External Action Block stub and external subroutine, together they become the External Action Block. When the EAB is statically called by a module marked for Compatibility place the resulting load module in the External Compatibility Action Block Load library or if otherwise place the resulting load module in the External Action Block Load library.

Important! Do not use the same libraries for External Action Block Load Libraries and External Compatibility Libraries concatenation. These libraries are mutually exclusive.

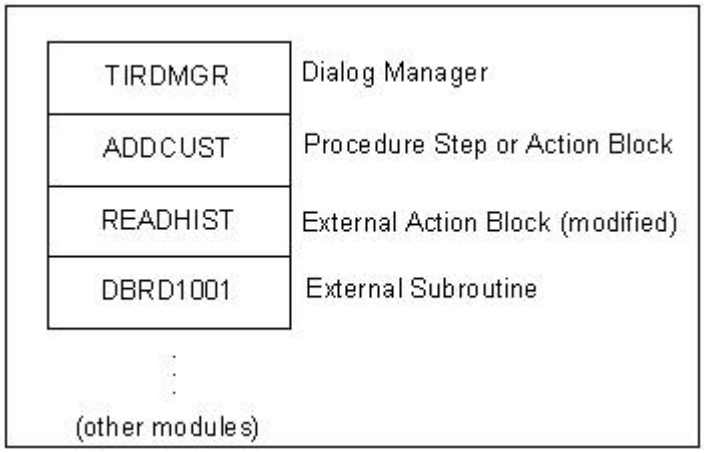
For DB2, place related DBRMs in the External Action Block DBRM library. The External Action Block load module is not executable.

When you generate and install, that is, compile, link-edit, and bind, a load module that contains a procedure step or action block that uses the External Action Block, the linkage editor resolves the reference to the External Action Block. How this reference is resolved depends on the type of call that is used to invoke the External Action Block and how it was compiled and linked. The Dynamically Link packaging option that is selected for the EAB determines if the EAB is statically included in the resulting executable (composite) load module, or invoked through a dynamic call.

When the EAB is statically included in the resulting executable or composite load module, this load module is stored in the Executable Load Module library. The composite load module contains the procedure step or action block, the modified External Action Block stub, and the external subroutines that the stub call.

When the EAB is invoked through a dynamic call, the EAB resides in an executable, separate from the load module, or composite executable. In this case, the executable containing the EAB must be built as a DLL or must be an action block that is marked for Compatibility. A Compatibility EAB is one that has its Dynamically Link Packaging property set to Compatibility and was built as a non-DLL module.

The following figure is a conceptual drawing of the composite load module IEF0000X. This load module contains one procedure step, ADDCUST, which uses the External Action Block READ_HISTORY. The name of the modified stub for the External Action Block is READHIST. It calls the external subroutine DBRD1001.



External Action Block Definition

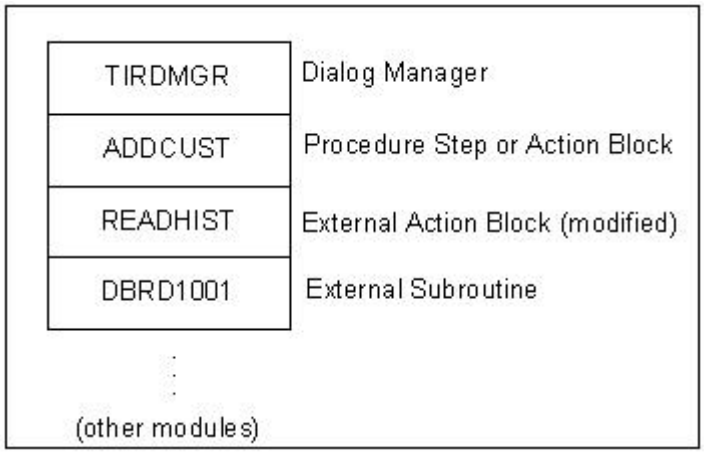
An External Action Block is defined using the Action Diagramming Tool, which is part of the Design Toolset on the workstation. An External Action Block is an action block that requires only the following components:

- Import Views
- Export Views
- EXTERNAL statement

Note: An External Action Block contains no action statements or action diagram logic of its own. The action block only contains the EXTERNAL statement.

The EXTERNAL designation tells CA Gen that a non-CA Gen generated program provides the data for the External Action Block's export views. The procedure step or action block that USEs the External Action Block supplies the import view data.

The following illustration is an example of an action diagram that defines the External Action Block READ_HISTORY.



The import views define the data that is passed from the CA Gen generated application to the External Action Block. The export views define the data that is returned to the CA Gen generated application from the External Action Block.

The External Action Block must be properly defined for CA Gen to be able to generate the code for the procedure step or action block that uses the External Action Block. Code generation does not automatically generate the code for the External Action Block. You can use the code generation tools to generate a stub for the External Action Block.

Create the Calling Procedure Step or Action Block

To create an action diagram for each procedure step and action block, use the Action Diagramming tool. To access an External Action Block, a procedure step or action block calls an External Action Block with the USE statement.

The data that is passed to and returned from the External Action Block is defined by view matching in the action diagram of the procedure step (or action block) that USEs the External Action Block. Define the External Action Block before you can perform the view matching. The view matching is done in the action diagram of the calling procedure step or action block. When you invoke a USE action, match the views of the External Action Block to the views of the procedure step or action block that is calling it.

Create External Action Logic

The following ways to create logic for an EAB:

- Add a call to an existing subroutine
- Write code for a new subroutine, and add a call to that subroutine in the EAB
- Add logic directly into a CA Gen generated stub

When you use the CA Gen generated stub, much of the work is already done in a format that is acceptable to the CA Gen software.

EAB code can be written in any language that follows LE linkage conventions. The specific requirements exist, however, for the action block name and the order of parameters that are passed to and from the CA Gen generated load module.

For COBOL, the PROGRAM-ID field must be the same as the name, or entry point, of the interface routine that is called in the stub.

The CA Gen generated load module passes the following parameters to the EAB interface routine in the order indicated as follows:

For COBOL with High Performance View Passing

IEF-RUNTIME-PARM1

IEF-RUNTIME-PARM2

PSMGR-EAB-DATA (null array)

w_ia (import view)

w_oa (export view)

IEF-RUNTIME-PARM1 and IEF-RUNTIME-PARM2 are the first two parameters that are passed from the CA Gen generated load modules. These parameters must be coded on the entry statement of the interface routine and must always be passed in this order to any subordinate routines that are called. However, if user code that contains IMS or CICS-specific code is placed directly in the EAB stub these IEF-RUNTIME-PARM parameters are changed. For IMS, the IEF-RUNTIME-PARM parameters are mapped to the IO-PCB and ALT-IOPCB, respectively. If the user code uses the IO-PCB, the ALT-IOPCB, or both, remove the corresponding IEF-RUNTIME-PARM from the LINKAGE SECTION and the PROCEDURE DIVISION USING statement.

For CICS, the IEF-RUNTIME-PARM parameters are mapped to the CICS EXEC Interface Block (EIB) and the COMMAREA, respectively. If the user code uses DFHEIBLK, DFHCOMMAREA, or both remove the corresponding IEF-RUNTIME-PARM from the LINKAGE SECTION and the PROCEDURE DIVISION USING statement. In addition, remove these parameters if the modified EAB is processed by the CICS translator, as the CICS translator automatically includes a reference to DFHEIBLK and DFHCOMMAREA.

Note: High Performance View Passing impacts the order of parameters that are transferred into an EAB. By default, High Performance View Passing is set ON.

For component development, High Performance View Passing must be set ON for the component and the consuming model. The import and export views for COBOL are the fourth and fifth parameters that are passed and are represented as W-IA and W-OA.

The interface routine must contain data structures that correspond exactly to the import and export views of the EAB. The fields in the data structures correspond to attributes in the import and export views of the EAB. Each attribute field in the data structures must be preceded by a one-byte field that is defined in COBOL as PIC X. This one-byte field contains a value that is not changed.

PSMGR-EAB-DATA is an array set to zeros (Null array). For the target system implementation, this array is passed but not used. The array is used for IMS and CICS applications. Modify the stub using an editor that can save files in the appropriate character set format. The modified stub cannot contain control codes or header information unique to the editor.

High Performance View Passing

High Performance View Passing makes passing views between action blocks more efficient. With High Performance View Passing enabled, individually matched views that are identical do not have to be copied into the calling action block at runtime. For frequently used action blocks, this can result in decreased CPU usage.

Enabling High Performance View Passing

Enabling the High Performance View Passing feature causes a change in the way that External Action Blocks are generated. The parameters that are passed in order include:

- IEF-RUNTIME-PARM1
- IEF-RUNTIME-PARM2
- GLOBDATA
- List of individually matched views (using the names that are provided in the action diagram), for example:
  - IMPORT-NON-REPEATING-GR-0005GV
  - EXPORT_CUSTOMER

The PSMGR-EAB-DATA (null array) is not passed directly as in normal view passing, because it is part of the GLOBDATA. There is a change in the order of the parameters passed. In addition, multiple views are passed with unique names rather than a single import/export view named W-IA or
W-OA.

The following two excerpts from External Action Block stubs illustrate some of the differences in the generated stubs when High Performance View Passing is in use.

External Action Block Stub Without High Performance View Passing

```
*
    PROCEDURE DIVISION USING IEF-RUNTIME-PARM1, IEF-RUNTIME-PARM2,
     W-IA, W-OA, PSMGR-EAB-DATA.

MAIN-0013697045.

    *
    * PERFORM PARA-0013697045-INIT THRU PARA-0013697045-INIT-EXIT
    * PERFORM PARA-0013697045 THRU PARA-0013697045-EXIT
    * GO BACK.

PARA-0013697045.
    MOVE 'N' TO FUNC-0013697045-ESC-FLAG

     * * * * * * * * * * * * * * * * * * * * * * *
     * USER-WRITTEN CODE SHOULD BE INSERTED HERE *
     * * * * * * * * * * * * * * * * * * * * * * *

PARA-0013697045-EXIT
EXIT.
```

External Action Block with High Performance View Passing

```
*
     PROCEDURE DIVISION USING IEF-RUNTIME-PARM1, IEF-RUNTIME-PARM2,
      GLOBDATA, IMPORT-0001EV IO-0002EV
      GROUP-IN-NONREPEATING-0001GV GROUP-IN-REPEATING-0002RG,
      EXPORT-0006EV OI-0007EV GROUP-OUT-NONREPEATING-0003GV
      GROUP-OUT-REPEATING-0004RG.

MAIN-0013631509.

    *
    * PERFORM PARA-0013631509-INIT THRU PARA-0013631509-INIT-EXIT
    * PERFORM PARA-0013631509 THRU PARA-0013631509-EXIT
    * GO BACK.

PARA-0013631509.
     MOVE 'N' TO FUNC-0013631509-ESC-FLAG

       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       * USER-WRITTEN CODE SHOULD BE INSERTED HERE             *
       * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

PARA-0013631509-EXIT.
EXIT.
```

Disabling High Performance View Passing

With High Performance View Passing disabled, the calling action block must (in most cases) copy the views that are matched to the called action block's imports and exports. The views must be copied to and from the caller's structure to two new structures that correspond to the called action blocks'
W-IA (import) and W-OA (export) structures. This must occur even if the individually matched group or entity views are the same, unless there is only one view in the import or export set and the generated structures are identical. This is the more familiar method of handling views for External Action Blocks.

This result in increased CPU usage for frequently USEd action blocks, particularly if the views are not matched identically.

Matching Import Views

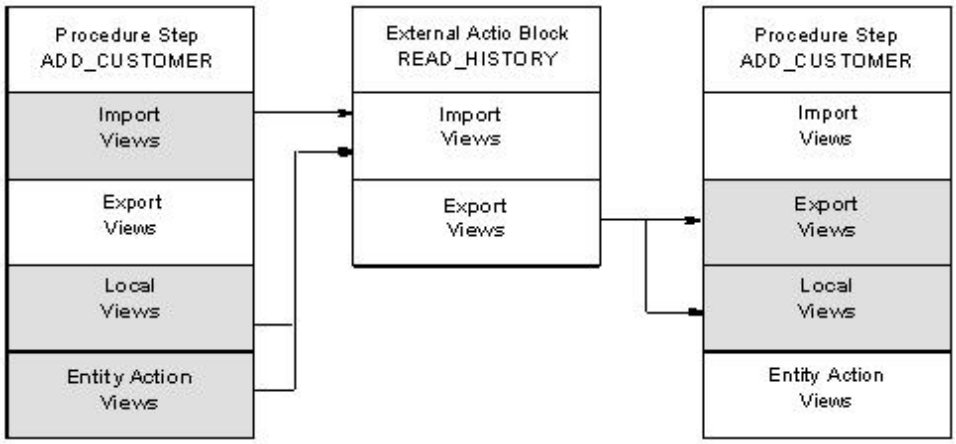
CA Gen maps the External Action Block import views to the supplying views from the procedure step or action block. The data is passed to the External Action Block. The supplying views in the procedure step is import views, entity action views, or local views.

Matching Export Views

CA Gen matches the External Action Block export views to the receiving views from the procedure step or action block as in the following illustration. (This is the data that is returned to the procedure step or action block.)The receiving views in the procedure step may be export views or local views.

External Action Block export views match to the receiving views, that are, the export and local views, in the procedure step or action block.

The following figure illustrates the possible supplying and receiving views for view matching in the Procedure Action Diagram. In this figure, Procedure Step ADD_CUSTOMER calls (uses) External Action Block READ_HISTORY.



When you have matched the import and export views of the External Action Block, the USE statement is complete. The procedure that is used to match the views is described in the *Block Mode Design Guide*. The USE statement displays in the action diagram of the procedure step or action block.

Note: For more information, see the *Block Mode Design Guide*.

The following code sample shows a portion of the action diagram for the procedure step ADD_CUSTOMER:

```
Procedure Step:  ADD_CUSTOMER
 ADD_CUSTOMER
  IMPORTS:
      Group View input customer_details
          Entity View input customer
  EXPORTS:
      Group View new customer_details
          Entity View existing customer
  LOCALS:
       Work View ief_supplied
  ENTITY ACTIONS:
       Entity View current cost_center
              .
              .
              .
  USE read_history WHICH IMPORTS: Entity View input customer
                   WHICH EXPORTS: Entity View existing customer
```

This procedure step uses the External Action Block READ_HISTORY.

Create External Action Blocks in HE

You are responsible for writing, compiling, and link-editing the external subroutine and making it available to the application. The external subroutines can be written in any language, but must follow specific conventions so that they interface properly with CA Gen generated programs.

To ensure that these requirements are met, CA Gen generates a COBOL program stub that you can use as a base for modification. The stub defines the runtime parameters that are passed from the calling procedure step or action block to the External Action Block. It also defines the import and export views of the External Action Block to match those of the calling procedure step or action block.

The requirements includes:

- Supplying a member name for the External Action Block
- Generating the External Action Block stub
- Modifying the stub to use the appropriate parameters and to call or to contain the external subroutine logic
- Resolving the External Action Block if it uses embedded SQL
- Generating and installing the procedure step that calls the External Action Block

Supplying the Member Name

The member name, entry point, of the External Action Block must be the same as its load module name. Identify it to CA Gen before you can generate the procedure step or action block that uses the External Action Block.

The first time that you request a system generation function for a load module, you can specify the member names to generate for the procedure steps, CA Gen action blocks, external action blocks, and screens. CA Gen includes a modify member name function to use to change a member name for a generated object.

Note: If you change the member name of a component that has already been generated, regenerate all affected load modules.

The Specify Member Names for *Load Module* panel is used to specify or change the name of an External Action Block stub. The panel looks different for online and batch load modules, but function the same.

More information:

Intelligent Regeneration (see page 245)

Member Name Requirements

Each name must be a valid PDSE member name and must be unique within the business system. (The name can have a maximum of eight alphanumeric characters. The first character must be a letter.) CA Gen detects invalid or duplicate names and prompts you to supply a new name.

If you do not specify a name, CA Gen assigns a default member name. The default name for an External Action Block stub is the first eight valid characters of the External Action Block name. For example, the default member name for the External Action Block READ_HISTORY is READHIST. If the first character is invalid, the default name is the Host Encyclopedia object ID preceded by the letter E.

Compile and link the External Action Block using the same name that you specified as the member name.

In the example from the figure in the External Action Block as Interface to Subroutine section in this chapter, if you do not specify a member name for the External Action Block READ_HISTORY, CA Gen assumes that the member name for the modified stub is READHIST (the default). If you modify the stub using the name READHIST as shown in earlier examples, CA Gen resolves the reference to the External Action Block during the final link edit of the load module. Use of another name would prevent CA Gen from successfully installing the External Action Block.

Generating the External Action Block Stub

You can use CA Gen to generate a stub of the COBOL program that implements an External Action Block. The generated stub is a partial COBOL program that includes much of the working-storage and linkage sections that are required to interface with CA Gen-generated programs. Use the stub to help write External Action Block code that interfaces correctly with CA Gen-generated code. It serves as a base to which you add the External Action Block logic.

The stub is stored in the sequential data set:

```
userid member_name.COBOL
```

member_name is the name that is specified on the Specify Member Name for *Load Module* panel. Copy this data set to a library before modifying it to prevent an accidental overwrite and the need to regenerate the stub.

You generate the stub during system generation by accessing the Load Module Expansion Panel for the load module that contains the External Action Block. From this panel, use the code-only option to generate the External Action Block stub. Generating the load module using the G option does not generate the stub.

Modifying the External Action Block Stub

The CA Gen generated procedure step or action block passes the following parameters to the External Action Block.

- IEF-RUNTIME-PARM1 (IO-PCB, DFHEIBLK)
- IEF-RUNTIME-PARM2 (ALT-IO-PCB, DFHCOMMAREA)
- W-IA (IMPORT-VIEW)
- W-OA (EXPORT-VIEW)
- PSMGR-EAB-DATA (PCB-DATA for IMS)

Depending upon the TP monitor in use, modify some of these parameters for proper functioning.

IEF-RUNTIME Parameters

IEF-RUNTIME-PARM1 and IEF-RUNTIME-PARM2 are the first two parameters that are passed from CA Gen generated programs. These must be coded on the entry statement of the stub and always be passed in this order to subordinate routines.

For the IMS applications, these parameters are the IO-PCB and alternate IO-PCB. In the CICS applications, these parameters are DFHEIBLK and DFHCOMMAREA. These parameters are not used for TSO and batch applications.

A special modification is needed for action blocks that are compiled with the CICS translator (precompiler). If you add the CICS commands or reference fields in DFHEIBLK, use the translator, and remove IEF-RUNTIME-PARM1 and IEF-RUNTIME-PARM2 from the linkage section and from the USING statement at the beginning of the PROCEDURE DIVISION. The translator inserts DFHEIBLK and DFHCOMMAREA to replace them. Failure to remove IEF-RUNTIME-PARM1 and IEF-RUNTIME-PARM2 result in an ASRA abend on the first move of data to the export views.

If the target TP monitor is CICS, but you added no CICS commands to the External Action Block and do not reference fields of the DFHEIBLK, you can compile without using the CICS translator. If you do so, do not remove IEF-RUNTIME-PARM1 and IEF-RUNTIME-PARM2.

W-IA and W-OA Parameters

W-IA and W-OA are the import and export views, respectively. The stub must contain data structures that correspond exactly to the import and export views of the External Action Block. The fields in the data structure correspond to attributes in the import and export views of the External Action Block. Each field in a data structure must be preceded by a field that is defined as PIC X.

Note: If the import or the export views have been modified and you want to replace the previous view declaration with the new one, then the stub must be regenerated. Regeneration is also required if the stub is used for reference and changes to the views were manually made. In this case, the generated code for the calling action diagram could be used instead. If the stub is regenerated, the data and procedure names could be different, as they are not stored on the encyclopedia but generated by an algorithm.

PSMGR-EAB-DATA Parameters

PSMGR-EAB-DATA (also named PCB-DATA) is an array of pointers that IMS applications uses to interface to DL/I databases. For the CICS and TSO applications, the array is set to zeros, a null array.

In the IMS applications, the array contains the PCB addresses as pointers and External Action Blocks uses to establish addressability to database PCBs. For each database PCB, code a PCB mask and set its address to the appropriate pointer. For example, assume that you have on DL/I database PCB, which is the third PCB in the PSB (following the I/O PCB and alternate I/O PCB). If you were to code a PCB mask in the LINKAGE SECTION named DB-PCB, you would establish addressability through the statement SET ADDRESS OF DB-PCB TO PSMGR-EABPCB-PTR(3).

Note: For more information, see IMS Documentation.

In the CICS applications, command level calls establish addressability by scheduling the required PCBs in the program. The scheduling does not require the addresses to be passed to each named module because they are implicitly available through the CICS DL/I calls.

Note: For more information about CICS DL/I services, see the appropriate CICS documentation.

In the TSO applications, the array is null because CA Gen does not support DL/I databases under TSO.

Using Varying Length Fields

CA Gen generates a two letter reference (DL) in the EAB to track the actual length of a varying length field. If you are using varying length fields in views to External Action Blocks, the DATA-FILPOST-0011DL field must be set.

Using Repeating Groups

If the export view of the External Action Block contains a repeating group view, indicate the number of occurrences that are returned by setting the maximum field. This is a numeric field that immediately precedes the repeating group, and which has a name that ends with the letters MA (for MAximum). The field is also redefined as alphabetic using a name that ends with the letters MX. If you use multiple repeating groups, you have multiple maximum fields, one for each group.

You are responsible to track how many occurrences are returned and placing that count in the maximum field. Failure to do so results in the calling procedure or action block behaving as if the External Action Block returned no data in the repeating group.

CA Gen generates the status of the repeating group and places it in the External Action Block stub as a two letter reference that is described in the following list:

AC

Active/Inactive flag for line item in group view. Used by CA Gen.

AS

Attribute status. Used by CA Gen.

FL

View full. Used by CA Gen in action blocks, not EABs.

PS

Subscript/pointer for repeating group view. Used by CA Gen.

RF

View referenced. Used by CA Gen in action blocks, not EABs.

Dynamically Link Options

Use the Target Environment panel to set the dynamically link options or use the Specify Member Names panel to override the options for a specific module.

The Dynamically Link property for individual EABs can be set to DEF, YES, NO, or CMP. DEF is the default.

DEF

For a business system, a default value for the Dynamically Link property can be established for action blocks. The action block Dynamic Link Property can be YES (enabled), NO (disabled), or CMP (Compatibility). If enabled, the resulting Dynamically Link property for an individual EAB that is set to Default is interpreted as having been explicitly set to Yes. If disabled, the resulting Dynamically Link property for an individual EAB that is set to Default is interpreted as having been explicitly set to No. If Compatibility, the resulting Dynamically Link property for an individual EAB that is set to Default is interpreted as having been explicitly set to Compatibility.

YES

Indicates that the associated EAB is invoked using a dynamic program call. The EAB must be built in its own fully resolved executable and must reside in a DLL.

NO

Indicates that the associated EAB is not to be the target of a dynamic program call, and as such isis statically linked into its identified owning load module.

CMP - Compatibility

Indicates to invoke the associated EAB using a dynamic program call. The EAB is expected to be a z/OS non-DLL load module that follows LE linkage conventions. To regenerate the EAB stub, select the generation option Process modules that are marked for Compatibility.

If regenerated, the EAB must then be rebuilt as a z/OS non-DLL load module. After rebuilding, put the NCAL EAB in a library in the External Compatibility Libraries panel. Modules containing calls to EABs that are marked for Compatibility require that they be generated so they can use runtime code to perform the DLL to non-DLL load module dynamic program call.

If the EAB marked for Compatibility or its callers contain SQL code, the DB2 BIND is redone. DBRMs for the generated caller module are automatically produced and used. However, DBRMs belonging to the EAB marked for Compatibility is resolved and made available to the BIND step.

Compile and Link-edit External Action Block

You are responsible for compiling and link-editing the External Action Block. The EAB can be statically linked, included in another module, or dynamically linked, its own module.

A static EAB can be called from a Compatibility module, from a static module or a dynamic DLL. When linking a static EAB, do not link DL/I, COBOL, DB2, CICS, or ISPF system modules into the load module for the External Action Block. The installation process links these modules into your final executable load module.

1. When the static EAB is called from a Compatibility module and the statically linked EAB is built, use the following compile and link options:
    a. Compile the EAB with the options OPT, NOSEQ, NODLL, NODYNAM, RENT, and NOEXPORTALL
    b. Link edit the EAB with the options RENT, REUS, NCAL, and DYNAM(NO).
    c. The NCAL linked EAB must be placed in a library in the External Compatibility Libraries panel.
2. When the static EAB is called from a static module or a dynamic DLL, the statically linked EAB must be built using the following compile and link options:
    a. Compile the EAB with the options OPT, NOSEQ, DLL, NODYNAM, RENT, and NOEXPORTALL
    b. Link edit the EAB with the options RENT, REUS, NCAL, and DYNAM(NO).
    c. The NCAL linked EAB must be placed in a library in the External Action Block Load Libraries panel.

When the EAB has been packaged with the Dynamically Link option of Yes, the EAB must be built as a DLL to be invoked by a generated application. Modules that are called by an EAB built as a DLL must also be a DLL. DLLs require that modules be bound as objects in Program Management Format 3 (PM3) and must reside in a PDSE library.

To build the EAB as a DLL, use the following compile and link-edit options:

Follow these steps:

1. Compile the EAB with the options OPT, NOSEQ, DLL, and EXPORTALL in addition to the default compiler options.

    Note: Selecting DLL forces the RENT and NODYNAM compiler options to also be used.

2. Link the EAB with the options RENT, REUS, and DYNAM(DLL).

    Note: Do not confuse the compiler option NODYNAM with the link-edit option DYNAM(DLL). The two are not the same as one is a compiler option while the other is link option.

3. Copy the EAB DLL to a data set that allow the DLL to be resolved in the target environment. For TSO and IMS this is a data set that is part of the STEPLIB concatenation. For CICS this is a data set that is part of the DFHRPL concatenation. Also, if the target environment is CICS, a CICS program definition must be added for the resulting EAB DLL.

Note: For information about DLLs, see IBM's Language Environment and the Program Management (Binder) documentation.

When the EAB has an associated Dynamically Link Packaging property set to Compatibility, the EAB must be built as a non-DLL. Modules that are marked for Compatibility must follow standard OS or LE linkage conventions and must operate in the same AMODE as the caller. These modules are not DLLs and must be fully resolved programs, eligible for being the target of an OS style dynamic program call.

To build the EAB as a non-DLL, use the following compile and link-edit options:

1. Compile the EAB with the options OPT, NOSEQ, NODLL, RENT in addition to the default compiler options.
2. Link edit the EAB with the options RENT, REUS.
3. Copy the non-DLL EAB to a data set that will allow the DLL to be resolved in the target environment. For TSO and IMS, this is a data set that is part of the STEPLIB concatenation. For CICS this is a data set that is part of the DFHRPL concatenation. Also, if the target environment is CICS, a CICS program definition must be added for the resulting non-DLL EAB.

Identify External Action Block Libraries

If the External Action Block is called by a module marked for Compatibility the library that contains the load module must be specified to CA Gen as an External Compatibility Library. If the External Action Block is called by an application that is not marked for Compatibility, then the library that contains the load module for the External Action Block must be specified to CA Gen as an External Action Block Load Library.

Identify DB2 DBRMs to Resolve External Action Block for DB2

If target DBMS is DB2 and the External Action Block uses embedded SQL, CA Gen must be given the names of database request modules, DBRMs, associated with the External Action Block. The External Action Block Resolution panel is used to identify DB2 DBRMs for an External Action Block. The panel can also be used to change the name of the External Action Block member. The member name is the name of the load module that contains the External Action Block.

An External Action Block that contains embedded SQL only precompiled once, even if it is built as DLL and non-DLL. The same External Action Block DBRM libraries are used for the DB2 Bind of EABs that are used by Compatibility and non-Compatibility applications.

Create Your Own External Action Block Source

You are not required to use the generated stub, as long as you follow these conventions:

- The name (entry point) of the External Action Block must be the same as its load module name and must be identified to CA Gen.
- The External Action Block must accept the five parameters that are passed to it from the CA Gen generated program.
- The External Action Block must contain data structures that correspond exactly to the import and export views of the External Action Block.

Use Existing Subroutines

If you use an External Action Block, as described previously, to access an existing subroutine, you do not need to changethe subroutine. However, the load library that contains the subroutine must be specified to CA Gen as an external load library. Also resolve the External Action Block by identifying the DB2 DBRMs.

Chapter 11: DB2 Utilities

Maintain Your Database

The database generation tools generate databases for a model and allow you to generate the JCL to run several DB2 utilities. These utilities help you maintain the integrity of your database through backup and restore capabilities. They also allow you to monitor the database efficiency, (database performance and space utilization), and tune your database to improve efficiency.

To use the DB2 utilities CA Gen supports, first generate the database and its subordinate objects.

Customers must have IBM's DB2 utilities or equivalent utilities from other vendors.

Important! CA Gen generates JCL for IBM's DB2 utilities. Using the utilities from non-IBM vendors requires customizing the generated JCL.

Note: For more information about DB2 utilities, see your vendor's documentation.

More information:

Database Generation in HE (see page 37)

DB2 Utilities Supported by CA Gen

To generate the JCL to run these DB2 utilities, use the database generation tools:

COPY

Create image copies of table spaces.

RECOVER

Recover table spaces and indexes.

REORG

Reorganize table spaces and indexes.

RUNSTATS

Gather RUNSTATS data for table spaces and indexes.

To select the DB2 utilities, use the DB2 Utilities panel:

```
                                DB2 Utilities
COMMAND ===>

 Model name . . . :  GEN SAMPLE MODEL
 Business system  :  CORPORATE_MANAGEMENT
 Data base  . . . :  GENDB

 Select one utility generation option, then press enter.

  . Create image copies of table spaces
  . Recover table spaces and indexes
  . Reorganize table spaces and indexes
  . Gather RUNSTATS data for table spaces and indexes

 If desired, type a creator ID.
   Creator ID   ????????


 F1=Help  F12=Cancel
```

COPY Utility

The DB2 COPY utility creates an image copy of a table space or a data set within a table space. It does not copy indexes. You can recover indexes from recovered table spaces.

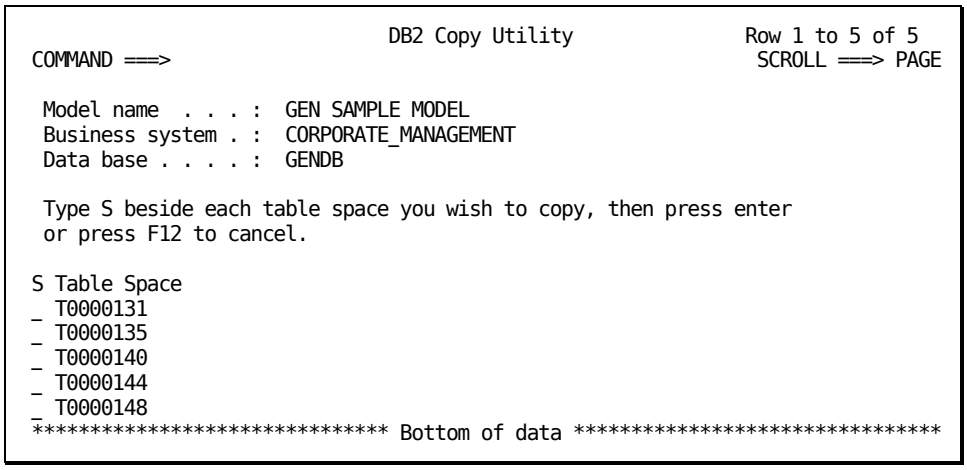
There are two types of image copies, a full image copy that copies all pages in a table space or data set and an incremental image copy that copies only the pages that are modified since the last COPY utility executed.

Note: CA Gen generates the JCL to support the full image copy only.

The COPY utility writes all pages from the table space to the output data set. If the table space includes multiple data sets, the COPY utility writes each data set to a separate output data set.

The generated JCL is for a Generation Data Group (GDG) tape copy. The high-level node in the GDG index is your TSO user ID. This can be changed in the generated JCL before the COPY utility job is submitted.

Use the DB2 Copy Utility to select the table spaces to copy.



The following example of the JCL generated for the COPY utility that is based on the selections that are made with the DB2 COPY utility.

```
EDIT ---- DAAAXXX.TICTEMP1.TIDUTJCL ----------------- COLUMNS 001 072
COMMAND ===>                                            SCROLL ===> CSR
****** ***************************** TOP OF DATA **************
000001 //DAAAXXX JOB(ACCOUNT INFO)
000002 //*TEST JCL SAVED
000003 //*
000004 //*
000005 //*
000006 //*
000007 //*
000008 //*
000009 //*
000010 //* ************************************************
000011 //* * DEFINE GDG MODEL FOR UTILITIES *
000012 //* ************************************************
000013 //STEPDCB EXEC PGM=IEFBR14
000014 //MODEL DD DSN=DAAAXXX.GDG,DISP=(NEW,CATLG,KEEP),UNIT=SPACE,
000015 // SPACE=(TRK,0)
000016 //* ************************************************
000017 //* * DEFINE GDGS FOR COPY UTILITY *
000018 //* ************************************************
000019 //STEP1 EXEC PGM=IDCAMS,COND=EVEN
000020 //SYSPRINT DD SYSOUT=*
000021 //SYSUDUMP DD DUMMY
000022 //SYSIN DD *
000023 DEF GDG (NAME(DAAAXXX.DB2T.COPY.T0000131) LIM(10))
000024 DEF GDG (NAME(DAAAXXX.DB2T.COPY.T0000135) LIM(10))
000025 SET MAXCC = 0
000026 /*
000027 //STEPCOPY EXEC PGM=DSNUTILB,REGION=1024K,
000028 // PARM='DB2T,CD000310',COND=EVEN
000029 //*
000030 //STEPLIB DD DSN=SYS1.DB2.LINKLIB,DISP=SHR
000031 //*
000032 //SYSPRINT DD SYSOUT=*
000033 //SYSUDUMP DD SYSOUT=*
000034 //T0000131 DD DSN=DAAAXXX.DB2T.COPY.T0000131(+1),
000035 // DISP=(NEW,CATLG,DELETE),
000036 // UNIT=TAPE,VOL=(,RETAIN),
000037 // LABEL=(1,SL),
000038 // DCB=(DAAAXXX.GDG)
000039 //T0000135 DD DSN=DAAAXXX.DB2T.COPY.T0000135(+1),
000040 // DISP=(NEW,CATLG,DELETE),
000041 // UNIT=AFF=T0000131,VOL=(,RETAIN,REF=*.T0000131),
000042 // LABEL=(2,SL),
000043 // DCB=(DAAAXXX.GDG)
000044 //SYSIN DD *
```

```
000045 COPY TABLESPACE GENDB.T0000131
000046 DSNUM ALL
000047 COPYDDN T0000131
000048 DEVT TAPE
000049 FULL YES
000050 SHRLEVEL REFERENCE
000051 COPY TABLESPACE GENDB.T0000135
000052 DSNUM ALL
000053 COPYDDN T0000135
000054 DEVT TAPE
000055 FULL YES
000056 SHRLEVEL REFERENCE
000057 /*
000058 //
****** ************************** BOTTOM OF DATA *************
```

RECOVER Utility

DB2 allows you to recover an entire table space, a data set, pages within an error range, or a single page. If a smaller data recovery unit (a page, for example) is required, you can use the generated JCL as an example from which to build the JCL you need.

Note: CA Gen generates JCL to recover an entire table space or an index.

The DB2 RECOVER utility recovers data to the current state or to a previous state. Data is recovered from image copies of a table space and database log change records. If the most recent full image copy data set is unusable, and previous image copy data sets exist in the system, RECOVER uses the previous image copy data sets.

You can recover multiple table spaces and indexes from the same database in a single execution of the RECOVER utility. DB2 locks the entire tablespace to allow an update of secondary indexes across partition boundaries. To recover an index, specifically request that the index be recovered. It is not recovered automatically when the table space is recovered.

To select the table spaces and indexes to recover, use the DB2 recover Utility panel.

```
                              DB2 RECOVER Utility            Row 1 to 13 of 23
COMMAND ===>                                                    SCROLL ===> PAGE

 Model name  . . . :  GEN SAMPLE MODEL
 Business system . :  CORPORATE_MANAGEMENT
 Data base . . . . :  GENDB

 Type S beside each table space and index you wish to recover
 then press enter or press F12 to cancel.

S Table Space Index
_ T0000131
_
_               PROJID
_               I0000078
_               I0000080
_ T0000135
_
_               TEAMID
_               I0000084
_ T0000140
_
_               DIVID
_ T0000144
```

The following example of the JCL generated for the RECOVER utility bthat is ased on the selections made using the DB2 Copy Utility.

```
EDIT ---- DAAAXXX.TICTEMP1.TIDUTJCL ------------ COLUMNS 001 072
COMMAND ===>                                      SCROLL ===> CSR
****** ***************************** TOP OF DATA ****************************
000001 //DAAAXXX JOB(ACCOUNT INFO)
000002 //*TEST JCL SAVED
000003 //*
000004 //*
000005 //*
000006 //*
000007 //*
000008 //*
000009 //*
000010 //STEPRECV EXEC PGM=DSNUTILB,REGION=1024K,
000011 // PARM='DB2T,RD000310',COND=EVEN
000012 //*
000013 //STEPLIB DD DSN=SYS1.DB2.LINKLIB,DISP=SHR
000014 //*
000015 //SYSPRINT DD SYSOUT=*
000016 //UTPRINT DD SYSOUT=*
000017 //SYSUDUMP DD SYSOUT=*
000018 //* ************************************************
000019 //* * WORK DATA SETS FOR SORTING INDEXES *
000020 //* ************************************************
000021 //SORTWK01 DD DSN=DAAAXXX.DB2T.RECV.SORTWK01,
000022 // DISP=(NEW,DELETE,CATLG),UNIT=SPACE,
000023 // SPACE=(TRK,(75,25))
000024 //SORTWK02 DD DSN=DAAAXXX.DB2T.RECV.SORTWK02,
000025 // DISP=(NEW,DELETE,CATLG),UNIT=SPACE,
000026 // SPACE=(TRK,(75,25))
000027 //SORTWK03 DD DSN=DAAAXXX.DB2T.RECV.SORTWK03,
000028 // DISP=(NEW,DELETE,CATLG),UNIT=SPACE,
000029 // SPACE=(TRK,(75,25))
000030 //SORTWK04 DD DSN=DAAAXXX.DB2T.RECV.SORTWK04,
000031 // DISP=(NEW,DELETE,CATLG),UNIT=SPACE,
000032 // SPACE=(TRK,(75,25))
000033 //* ************************************************
000034 //* * MESSAGES FROM DFSORT *
000035 //* ************************************************
000036 //UTPRINT DD SYSOUT=*
000037 //* ************************************************
000038 //* * WORK DATA SETS FOR INTERMEDIATE OUTPUT *
000039 //* ************************************************
000040 //SYSUT1 DD DSN=DAAAXXX.DB2T.RECV.SYSUT1,
000041 // DISP=(NEW,DELETE,CATLG),UNIT=SPACE,
000042 // SPACE=(CYL,(75,25))
000043 //SYSIN DD *
000044 RECOVER TABLESPACE GENEDB.T00001316
```

```
000045 RECOVER INDEX (PROJID)
000046 RECOVER INDEX (I0000078)
000047 000048 RECOVER INDEX (I0000080)
000049 /*
000050 //
****** ***************************** BOTTOM OF DATA ************************
```

REORG Utility

The REORG utility reorganizes a table space to improve access performance and reorganizes indexes so that they are more efficiently clustered. During the REORG execution, DB2 locks the entire tablespace to allow an update of secondary indexes across partition boundaries.

To select the table spaces and indexes to reorganize, use the DB2 Reorg Utility. CA Gen generates the REORG JCL based on the selections made using the DB2 REORG Utility.

RUNSTATS Utility

The RUNSTATS utility scans a table space or index to gather information about utilization of space and efficiency of indexes.

The information is recorded in the DB2 system catalog. It is used by the Installation function during the bind process to select access paths to data. The database administrator can use catalog queries to review the RUNSTATS information. The information is used to evaluate database design and determine when table spaces or indexes are reorganized.

To select the table spaces and indexes for which to gather RUNSTATS information, use the DB2 RUNSTATS Utility. CA Gen generates the RUNSTATS JCL based on selections that are made with the RUNSTATS Utility.

Creating Database Utility JCL

The database utility functions create the JCL to run a DB2 utility. The activities to generate the JCL for a DB2 utility are:

- Select the Database
- Select Utility and Specify DB2 Creator ID
- Select Objects and Generate JCL
- Specify Job Statement Information

- Edit Generated JCL
- Submit Generated JCL

Select the Database

The first step in generating the JCL for a DB2 utility is to select the required database from the Database List panel. The Database List shows all databases that are defined for the model.

Select Utility and Specify DB2 Creator ID

After selecting the database, select the utility to perform from the DB2 Utilities options.

The panel also allows you to specify the Creator ID. The Creator ID is the DB2 Authority ID of the DB2 user who created the database objects. The DB2 system catalog identifies the Creator ID for DB2 objects.

If specified, the Creator ID is included in the generated JCL to make index names that are fully qualified. Table space names are qualified by the database name. If a Creator ID is not supplied, the creator qualification is not included in the generated JCL.

Select Objects and Generate JCL

After you select the DB2 utility, select the objects to process. Each utility supports different objects:

- COPY supports Table Spaces
- RECOVER supports Table Spaces and Indexes
- REORG supports Table Spaces and Indexes
- RUNSTATS supports Table Spaces and Indexes

CA Gen generates the JCL to run the utility for the selected objects.

Specify Job Statement Information

CA Gen prompts for the job statement information to include in the generated JCL. If the job statement information is not supplied, CA Gen still generates the JCL, but the job is not executable until you add the job statement information.

CA Gen provides two ways to specify or revise job statement information: during database generation, or outside of database generation, using the Environment option on CA Gen Main Menu. Both options specify and update the same set of JCL.

The Environment option allows display of the Update JCL for Data definition Language panel. Generating the JCL from within the Database Definition panels lets you access the Specify DDL Job Statement Information panel. Both panels display eight lines of JCL for initial entry or update.

Your job statement information meets the requirements and standards of your installation. The time and region parameter values vary depending on the number and size of the DB2 objects that are involved in the utility.

Edit Generated JCL

When the JCL for the selected utility and objects are generated, CA Gen automatically displays it in an ISPF Edit session, allowing you to review the generated JCL and make changes.

Submit Generated JCL

If your environment permits, and you have been granted authority, you submit the job to run the selected utility.

DB2 security requires that all users be authorized, explicitly or implicitly, to do whatever they do. To execute a DB2 utility, you must have the proper authority. Usually, the Database Administrator (DBA) is responsible for granting and revoking DB2 authorities. Consult your DBA before executing a database utility.

Use the appropriate vendor documentation to find more information about DB2 security mechanisms.

If you do not intend to submit the job immediately, save it so that it can be submitted later. CA Gen does not save the job for you.

Generating JCL for a DB2 Utility

Follow these steps:

1. Access the Application System Construction panel for specifying the model and business system.
2. Type the Host Encyclopedia name of the model.

   You can also use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a model name from the list by typing any character except a period (.) or a space character next to one name and press Enter again.

3. Type the Host Encyclopedia name of the business system.

   You can also use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a business system from the list by typing any character except a period (.) or a space character next to one name and press Enter again.

4. Press Enter. The Application System Construction Menu displays.

   CA Gen automatically retrieves the construction library and target environment specifications that are stored in the Host Encyclopedia for the selected business system.

5. Select 4 for Generate Database Definition and press Enter.

   The Database List panel displays.

6. Type U next to the required database and press Enter.

   The DB2 Utilities panel displays.

7. Select the utility:

   - COPY
   - RECOVER
   - REORG
   - RUNSTATS

8. If a Creator ID is required to make index names in the generated JCL fully qualified, press the Tab key to position the cursor to the Creator ID field and type the DB2 Authority Code of the creator of the DB2 object.

   If you do not know whether the Creator ID is required for the utility you want to run, contact your DBA.

9. Press Enter.

   The object selection panel for the requested utility displays.

10. Type an S next to the name of each object you want processed by the selected utility. You can select more than one object at a time. CA Gen generates the required JCL for all selected objects.

11. Press Enter.

    The Specify DDL Job Statement Information panel displays. The information displays previously specified entries.

12. Type the changes to the job statement information for the JCL.
    a. To save changes and exit the panel, press End. CA Gen generates the JCL.

       The generated JCL is displayed in an ISPF Edit session (if the auto-submit option is set to No on the Generation Options panel). Review the JCL and edit it if necessary.

       If you do not intend to submit the job immediately, save it so that it can be submitted later. CA Gen does not save the job for you.

    b. To cancel changes to the job statement information and exit the panel, type Cancel on the COMMAND line and press Enter or press Cancel.
13. Type SUBMIT on the COMMAND line and press Enter to submit the job. The job runs the requested DB2 utility.

Chapter 12: Intelligent Regeneration

Automated Regeneration

When a CA Gen generated application requires changes, the application developer modifies the appropriate diagrams and regenerates the affected application components. For relatively small changes, it is not necessary nor is it desirable to regenerate the entire application. However, it is not always obvious which components the change affects. CA Gen Intelligent Regeneration tool automates the process of determining which components need regeneration.

You can still perform repeat generation of application components by expanding load modules and making selections in the normal Construction panels. In some cases, when it is clear which components need regeneration, you prefer this approach. However, if you are not sure what changed, or if you are uncertain which application components need regeneration, the Intelligent Regeneration tools can make the analysis for you.

You can mix the two methods of regeneration with no ill effects. When you upload changes to the Host Encyclopedia, the model is updated in a manner that supports subsequent identification of components requiring regeneration.

Intelligent Regeneration lets you regenerate only those parts of an application system that is affected by recent changes to the model. Intelligent Regeneration makes maintenance of existing systems quicker and easier to implement because CA Gen tracks the changes to the model and performs selected generation only of changed components.

In addition to automatic determination and application component regeneration, Intelligent Regeneration lets you examine the impact of proposed changes before making the changes. You can also review the impact of changes made before regenerating the affected components.

Terms

These terms help you to understand Intelligent Regeneration processing:

Generation object

An object type that can be generated and installed. Action blocks, screens, and load modules are examples of generation objects.

Aggregate object

An object type that identifies a key change in the model. It can be thought of as a trigger that causes other actions. The changing of an aggregate object cause a generation object to be marked for regeneration.

Generate

Create the source code.

Install

Compile, link-edit, and place in a library ready for execution.

Prerequisites

Intelligent Regeneration evaluates only packaged load modules. To use Intelligent Regeneration, complete the following steps:

- Created a model using the workstation toolsets
- Performed Transformation or Retransformation (in Design) and completed the Data Store List and Data Structure List
- Uploaded the model to the Host Encyclopedia
- Created a business system with at least one procedure step
- Packaged for at least one procedure

Model Conversion Considerations

Consider the following for models that have had code that is generated under earlier versions of Gen code generators:

- To install a load module, the Dialog Manager and all screens in the load module must be regenerated in the current encyclopedia.
- It is a more efficient use of resources to perform generation from the Applications System Construction panels before using Intelligent Regeneration.

Tracking Changes

CA Gen tracks changes in a model using these identifiers:

- Session Object
- OK Session Flag
- Generation object level

CA Gen tracks changes by timestamp, when the change was made, and by user ID, who changes, and tracks changes in aggregate for a high-level object, for example, an action diagram, and its lower-level object components, for example, a statement. Changes to certain low-level components do not result in flagging objects for regeneration. Included are changes to text descriptions and object positions within a display such as the Entity Relationship Diagram (ERD).

Note: The aggregate objects that are referenced for Intelligent Regeneration and the timestamp user ID that indicates the last change are the same as those used for Version Control, although some differences do appear in the list that is displayed for each.

Not every aggregate object that is used for Version Control affects code generation. For example, certain objects that are created with the Planning Toolset, subject areas, and functions. Some aggregate objects that affect code generation could be migrated only as a component of a higher-level aggregate object. For example, a subtype, import view set, or parameter delimiter.

Note: For more information, see the *Host Encyclopedia Version Control User* Guide.

Session Object

The actual user ID-timestamp is carried in a session object, which is referenced by all aggregate objects that are changed during one session. CA Gen creates a unique session object for the set of changes from one toolset session each time a model update occurs. Functions on the encyclopedia that affect model information, such as Packaging, Rename, or Delete, also incorporate this logic. The reference for an aggregate object changes each time a component of that aggregate object changes, so that it always points to the last change made.

Session objects have an ID number, which is assigned in sequence as sessions are added. Since these numbers are assigned chronologically, the greatest value indicates the latest change to the model.

OK Session Flag

All application components have a generation timestamp and an OK Session property value. The OK Session value is used as a flag to indicate whether the component requires generation. When you request regeneration, or preview the results of changes in a model, Intelligent Regeneration tracks those components that require regeneration. If you do not continue with regeneration, and the flag *Mark and record components requiring generation* is set to Yes, components that are identified but not regenerated are flagged to facilitate analysis in the next session.

In addition, if a component is determined *not* to require regeneration, an association is made to the latest session in the model, to indicate that the component generation is current with that session. That session date overrides the actual generation date for future Intelligent Regeneration sessions, to minimize change analysis that is required.

Automatic Regeneration Processing

When you change a model during a workstation session and upload the changes, CA Gen attaches a unique session object for that set of changes. To determine key changes, cascading occurs from lower-level objects to higher-level objects. For example, changing a permitted value flags the associated attribute. These higher-level or aggregate objects are attached to the session object that has timestamp properties that are assigned. As you change your model, aggregate objects are detached from an earlier session and reattached to the new session.

Effects on Cross-Copied Models

When a model is cross-copied between encyclopedias, the session object ID numbers are reset to zeroes/blanks. This preserves the chronological integrity of session numbers in the new encyclopedia.

Processing Logic

You can request Intelligent Regeneration for a model, a business system, or a load module. You can further restrict the scope of the request by specifying changes that are made during a specific time frame, by a specific user ID, or select to ignore changes that are made to the target environment and construction libraries. If the scope is restricted by these parameters, the results of the analysis are not saved for future work sessions. When the scope of the request is defined, the Intelligent Regeneration process:

- Determines all application components included in the scope of the request: a business system has load modules; load modules have procedure steps; procedure steps have action diagrams and screens; procedure step action diagrams have action blocks, derivation algorithms, referential integrity triggers, or both.
- Records the earliest timestamp that a component was generated.
- Identifies the sessions and the associated aggregate objects that occurred since the earliest component timestamp and, if requested, within the specified time frame and by the specified user ID.
- Reviews each generation object, according to the Intelligent Regeneration Rules, to mark the appropriate components as changed.
- Flags the components for regeneration if an aggregate object change affects that component, and if the aggregate object's session timestamp is later than the component's generation timestamp.

Regeneration occurs for the flagged components.

z/OS Compatibility Option Considerations

When dynamically link default options are modified at the Business System level, Intelligent Regeneration processes individual modules using Default value for the dynamically link option if Ignore Target Environment Parameter Changes option is set to No on Regeneration Options panel. If this option is set to Yes, Intelligent Regeneration ignores the modifications that are made at the Business System level.

When the dynamically link option is changed to CMP(compatibility), for an action block, Intelligent Regeneration marks the calling module as requiring regeneration because the call statement is different in the caller module. IIntelligent Regeneration will also list compatibility action blocks as required to be regenerated when their dynamic flag is changed to CMP, although they only require to be relinked. This is because change capture or Intelligent Regeneration does not know what was changed. This behavior is the same for action blocks with dynamically link option set to DEF (Default), when the Business System defaults change to CMP and Ignore Target Environment Parameter Changes option is set to No.

Intelligent Regeneration does not list screen modules as requiring regeneration when their dynamic flag is set to CMP. Instead it lists their associated action blocks as requiring regeneration because when a screen's dynamic flag is modified, change capture flags the action block as changed and not the screen.

After identifying the components, Intelligent Regeneration provides generation and installation options panels similar to those in Host Construction. These Intelligent Regeneration panels and dialogs include a *Process modules marked for Compatibility Yes/No* option to indicate that modules marked for Compatibility should or should not be processed, similar to Host Construction.

- When this option is set to No, Intelligent Regeneration behaves the same as in the AllFusion Gen 7.6 release without the Compatibility Priority Enhancement. CA Gen bypasses the generation, installation, remote installation processing, or both of modules that are marked for Compatibility. When Intelligent Regeneration runs again, the Compatibility modules are listed as requiring regeneration or reinstallation even though their processing was specifically bypassed because Intelligent Regeneration uses the generation or installation timestamp in decision making.
- When this option is set to Yes, modules marked for Compatibility are processed, that is generation, installation, inclusion for remote installation, or both occurs, only if they are included within the scope of those modules that are selected for processing. When Intelligent Regeneration runs again, Compatibility modules are not listed as requiring regeneration, or reinstallation if they were previously generated or installed, and the module was not modified since the last Intelligent Regeneration run, such as modifying the action diagram on the Toolset.

Selecting the Process modules marked for Compatibility option causes the RI Trigger modules and all Action Blocks statically called by the module marked for Compatibility to compile twice, once using the NODLL compiler option and again using the DLL compiler option.

The result of the NODLL compile is linked into the Compatibility module. The result of the DLL compile can be linked into DLL applications that statically call these components.

Intelligent Regeneration identifies the Compatibility modules if Process modules marked for Compatibility option is set to Yes.

Aggregate Objects List

The list of aggregate objects in the following list appear in the same order as the expansion panels. The objects in the list are a reference point for a group of objects. CA Gen identifies objects by their aggregates. The rules tables in this chapter help you to determine the components that a change affects. The following list shows Aggregate Objects:

- ACTION BLOCK
- ATTRIBUTE
- BATCH JOB
- BATCH JOB STEP
- BUSINESS SYS IMPL
- CLASSIFIER
- COMMAND
- COMMAND SYNONYM
- COMPONENT IMPLEMENT
- COMPONENT SPEC
- DATA COLUMN
- DATA TABLE
- DEFAULT EDIT PATTERN
- DEFAULT ERR FLD ATTR
- DEFAULT LITERAL ATTR
- DEFAULT NRM FLD ATTR
- DEFAULT PROMPT ATTR
- DENORMALIZED COLUMN
- DIALECT
- DIALOG FLOW
- ENTITY
- EXIT STATE
- EXPORT VIEW SET
- EXT. IMPLMENT. LOGIC
- FOREIGN KEY COLUMN
- IMPLEMENTATION LOGIC

- IMPLEMENT. SCREEN
- IMPORT VIEW SET
- INTERFACE TYPE
- LINK TABLE
- LOCAL PFKEY
- ONLINE LOAD MODULE
- PARAMETER DELIMITER
- PARAM. STR. DELIMITR
- PERMITTED VALUE
- PROCEDURE
- PROCEDURE STEP
- RELATIONSHIP
- SCREEN
- SERVER MANAGER
- SPECIFICATION TYPE
- SUBTYPE
- SYSTEM-WIDE PFKEY
- TECH DESIGN DEFAULT
- TEMPLATE
- WORK ATTRIBUTE
- WORK ATTRIBUTE SET
- Z/OS LIBRARY

The following list shows some of the higher level objects on selection panels and the corresponding lower level objects.

Entity

Entity Subtype, Attribute, Relationship Membership

Entity Subtype

Entity Subtype, Attribute, Relationship Membership

Work Attribute Set

Work Attribute

Procedure

Procedure Step

Procedure Step

Procedure Step Action Block, Screen, Dialog Flow, Local PFKEY

Batch Job

Batch Job Step

Data Record

Data Field, Denormalized Field, Foreign Key Field, Entry point

Intelligent Regeneration Rules

For each aggregate that you change, one or more components need regeneration. You make these changes in various diagrams within CA Gen Toolset. Aggregate Object identifies changes in CA Gen diagramming tool where the changes occur.

Changes from Dialog Flow Diagram

Aggregate Object	Change	Regeneration Rule
DIALOG FLOW	Add a dialog flow Change dialog flow type Change Data Sent/Returned Change Execute first/Display first property Change Flows on/Returns On exit state Assign Autoflow exit state	Regenerate the Dialog Manager associated with the source procedure step
BUSINESS PROCEDURE	Add a procedure Delete a procedure step in a multi-step procedure Modify procedure properties Add a procedure step	Regenerate the Dialog Managers for all steps in the procedure

Aggregate Object	Change	Regeneration Rule
BUSINESS PROCEDURE STEP	Add a procedure step Add/Delete a dialog flow Modify procedure step properties Mark a procedure step for unformatted input	Regenerate the Dialog Manager associated with the procedure step Regenerate the screen associated with the procedure step Regenerate the Dialog Managers for all load modules that contain a flow to the procedure step

Changes from the Screen Designer

Aggregate Object	Change	Regeneration Rule
SCREEN	All changes to the screen	Regenerate the screen
TEMPLATE	All changes to a template	Regenerate all screens that use the changed template

Changes from Business System Configuration

Aggregate Object	Change	Regeneration Rule
TECHNICAL SYSTEM	Change a technical system property Add usage of a library Delete usage of a library	Regenerate all screens in the Business System associated with this technical system Regenerate all Dialog Managers in the business system Regenerate all action blocks in the Business System

Changes from Load Module Packaging

Aggregate Object	Change	Regeneration Rule
ONLINE LOAD, BATCH JOBSTEP	Any Load Module Packaging change	Regenerate the Dialog Manager for the associated load module Regenerate Dialog Managers for all load modules that contain a flow to any repackaged procedure step in the newly packaged load module

Changes from Member Name Resolution

Aggregate Object	Change	Regeneration Rule
SCREEN IMPLEMENTATION	Modify the member name of a screen Modify the mid name of a screen Modify the mod name of a screen Modify the format name of a screen	Regenerate the screen
IMPLEMENTATION UNIT, EXTERNAL IMPLEMENTATION UNIT	Modify the member name of an action block Modify action block type, External or Internal	Regenerate the action block Regenerate the Dialog Manager associated with renamed procedure step Regenerate all action blocks that USE the renamed action block Regenerate all action blocks that read the derived attribute the renamed action block sets
PARAMETER DELIMITERS	Modify the set of parameter delimiters in a Business System	Regenerate all screens in the Business System where the associated procedure step has unformatted input

Aggregate Object	Change	Regeneration Rule
PARAMETER STRING DELIMITERS	Modify the set of parameter string delimiters in a Business System	Regenerate all screens in the Business System where the associated procedure step has unformatted input
PFKEY (system)	Associate a command with a PFkey Modify display property for PFkey	Regenerate all screens in the business system, except those in which the system-wide PFkey was overridden by a local PFkey
PFKEY (local)	Associate a command with a local PFkey for a given procedure step	Regenerate the screen associated with the procedure step
DEFAULT EDIT PATTERN	Add or change a default edit pattern	Regenerate screens containing a screen variable that uses the default edit pattern If screen variable is on a template, regenerate all screens that use that template
DEFAULT VIDEO ATTRIBUTE ...ERROR FIELD ...LITERAL ...NORMAL FIELD ...PROMPT	Change properties associated with the default attributes of that video object type	Regenerate screens that contain one or more of that field, where the default video attributes have not been overridden with custom settings Regenerate screens that USE a template that contains one or more of that type field, where the default video attributes have not been overridden with custom settings

Aggregate Object	Change	Regeneration Rule
COMMAND	Change the name of a command Add/Delete a command synonym Add a command	Regenerate all action blocks that reference the command in an expression Regenerate the Dialog Manager for a procedure step whose action diagram contains a reference to the command Regenerate the Dialog Manager for the source procedure step for autoflow caused by the command Regenerate the Dialog Manager for the source procedure step for flow that sets the command
COMMAND SYNONYM	Add a Command Synonym	Regenerate the Dialog Manager for all load modules in the Business System
EXIT STATE	Add an exit state Modify the exit state name Modify the message text of an exit state Modify the exit state type Modify the exit state message type	Regenerate all Action Blocks that contain a reference to the exit state Regenerate the Dialog Manager for the procedure step that references the exit state for an autoflow

Changes from Action Diagramming

Aggregate Object	Change	Regeneration Rule
ACTION BLOCK	Add or change an action block Add or change a derivation algorithm Modify the local view set of an action block Modify the entity action view set of an action block Modify the import or export view set of a procedure step Modify the import or export view set of an action block Modify the import or export view set of a screen	Regenerate the action block

Changes from View Maintenance

Aggregate Object	Change	Regeneration Rule
IMPORT VIEW EXPORT VIEW	Modify the import or export view set of a procedure step Modify the import or export view set of an action block Modify the import or export view set of a screen	Regenerate the screen associated with the procedure step Regenerate the Dialog Manager for the load module that is associated with the procedure step Regenerate the Dialog Manager for procedure steps that flow to the procedure step whose view changed Regenerate the action block Regenerate all action blocks that USE the action block whose view set was changed Regenerate all action blocks that read a derived attribute, when the derivation algorithm view set was changed Regenerate all derivation algorithms whose import view set contains a derived attribute whose derivation algorithm view set has changed

Changes from Data Modeling

Aggregate Object	Change	Regeneration Rule
ENTITY TYPE ENTITY SUBTYPE	Add a new entity type Add or change properties of an entity type Change the identifier set for an entity type Add/Delete an attribute, relationship, identifier or partitioning	Regenerate all action blocks that contain entity action views of the changed entity type If an entity subtype was changed, regenerate all action blocks that contain entity action views of an entity type or subtype in the hierarchy of the highest level entity type. All components, that is procedure steps and action blocks,must be regenerated if an attribute is added to a common entity type when using persistent views.
CLASSIFIER	Add/Change a classifier for a partitioning	Regenerate the action block associated with highest level entity type in current entity hierarchy Regenerate action blocks associated with highest level entity types that are involved in relationships with an entity in the current entity hierarchy (DELETE triggers only) Regenerate action blocks that reference a relationship in the current entity hierarchy through a DISASSOCIATE statement
RELATIONSHIP	Add a relationship membership Add or change the properties of a relationship membership (cardinality/optionality/name)	Regenerate all Design action blocks that reference the relationship in an Entity Action view Regenerate all Design action blocks that reference the relationship membership through a DISASSOCIATE statement, and all action blocks that call the DISASSOCIATE action block

Aggregate Object	Change	Regeneration Rule
ATTRIBUTE	Add an attribute Add or change properties of an attribute Add/Delete a permitted value	Regenerate all Dialog Managers for procedure steps that contain predicate views of the attribute Regenerate all action blocks that contain predicate views of the attribute Regenerate all screens that contain predicate views of the attribute
PERMITTED VALUES	Add or change a permitted value for a text attribute Add or change a permitted value range for a numeric attribute	Regenerate all action blocks that contain Entity Action views that contain the attribute, if the Entity Action view is used in a READ, UPDATE or CREATE statement Regenerate all screens that reference the associated attribute as a screen variable

Changes from Data Structure Diagram

Aggregate Object	Change	Regeneration Rule
DATA FIELD FOREIGN KEY FIELD DENORMALIZED FIELD	Add or change a Field in a Record Change Data Type of a Field Denormalization changes	Regenerate all action blocks that reference the record in which the field resides
RECORD LINK RECORD	Add/Delete a field in a record Rearrange Fields in a Record Add a record Modify record properties	Regenerate all action blocks that reference the record

Impact Analysis and Preview Reporting

Intelligent Regeneration provides the reports to preview the consequences of changes or proposed changes to the model.

Construction Change Impact Report

Use the Construction Change Impact report to view the application components that would require regeneration if a change occurs. When you select the objects that are proposed for change, the report shows the components that require regeneration if that change occurs. Every aggregate object that you select is in the report. Since no changes are made no application components are marked for regeneration.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 1 for Construction change impact report and press Enter.

   The Construction Change Impact panel displays.
3. Type the Host Encyclopedia name of the model.
4. Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices.
5. Select a model name from the list by typing any character except a period (.) or a space character next to the name and press Enter.
   a. Scope the business system by placing a slash character (/) opposite the option to enable. Place a period (.) or a space character opposite the option to disable.
   b. Type a business system name only when regenerating a single business system. To regenerate all business systems, do not type a name in this blank.
   c. Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices.
   d. Select a business system name from the list by typing any character except a period (.) or a space character next to the name and press Enter.
   e. Scope the load module by placing a slash character (/) opposite the option to enable. Place a period (.) or a space character opposite the option to disable.
   f. Select the Execution mode, online or batch.
6. Press Enter.

   The Select Aggregate Object Types panel displays.
7. Type a slash character (/) by the object types to expand. Use the scroll keys to see the entire list.
8. Press Enter.

   The system displays the message:

   ```
   Input has been accepted, press "enter" or enter "accept" command to proceed.
   ```

9. Press Enter.

   The Object Occurrences panel displays. The list contains all object occurrences of the first type selected.

10. Type S or E next to the objects you to select for reporting.

    - Type S to analyze individual aggregate object to select it for reporting
    - Type E to expand individual aggregate object

11. Press Enter.

    If you did not expand objects, the system displays the message:

    ```
    Input has been accepted, press "enter" or enter "accept" command to proceed.
    ```

    If you expanded objects, the system displays the Expanded Occurrences panel to select more objects to add to the report using S to add to the report and E for further expansion, if available. CA Gen displays an error message if the selected object cannot be expanded.

    When you complete expansion and have added the required objects to the list to be analyzed, press Enter. The Object Occurrences panel displays.

12. Press Enter.

    The Confirm Change Impact List panel displays.

13. Type a slash character (/) next to an aggregate object occurrence to remove from the list. The system responds with a message indicating that the selected objects were removed from the list.

14. Press Enter to begin processing or click Cancel to stop processing.

    If you cancelled processing, the Select Aggregate Object Types panel opens to allow selection of new analysis criteria.

    If you chose foreground processing, CA Gen processes the selection that you made.

    When complete, the Browse Report File panel displays.

    If you selected background processing, the Update JCL for Encyclopedia Functions panel displays.

15. Type changes to your Job card and press Enter.

    The JCL is presented in ISPF Edit mode.

16. Type Submit on the COMMAND line and press Enter to submit the job for processing.

    The Application System Regeneration menu displays to perform other processing. Omit the remaining steps in this procedure.

17. Review the report in the Browse Report File panel and click End when finished.

    The Report Print Options panel displays.

18. Press the appropriate key or type a command on the COMMAND line and press Enter.

    You are only required to press Enter if the Auto-submit option on the Generation Options screen was not set to Yes.

    CA Gen returns to the Application System Regeneration menu.

Other messages that could occur in this report include those listed as follows. The term entity triggers refers to Referential Integrity triggers of ENTITY and RELATIONSHIP types.

- All components, including entity triggers, would be flagged for regeneration/installation
- All components, excluding entity triggers, would be flagged for regeneration/installation
- All entity triggers would be flagged for regeneration/installation
- No components would be flagged for regeneration/installation
- Modifying [aggregate] would flag these components for Regeneration
- No Business System components and no RI Triggers would be flagged for Regeneration
- No Business System components and no RI Triggers would be flagged for Installation

An example of a Construction Change Impact Report panel is shown in the following illustration:

```
2009-11-24                INTELLIGENT REGENERATION                PAGE    1
06:21                CONSTRUCTION CHANGE IMPACT REPORT

   MODEL NAME:              GEN SAMPLE MODEL
                         ALL OBJECTS CHANGED IN MODEL
                                BY ALL USERS

--------------------------------------------------------------------------------

Modifying Attribute NAME of Entity Type DEPARTMENT would flag these components
for Regeneration:
     Module Manager for Online Load Module MENU
     Module Manager for Server Manager P900
     Process Action Block ADD_DEPARTMENT
     Process Action Block ADD_EMPLOYEE
     Process Action Block DELETE_DEPARTMENT
     Procedure Step Action Block EMPLOYEE_DETAIL
     Process Action Block DELETE_DEPARTMENT
     Procedure Step Action Block EMPLOYEE_DETAIL
     Procedure Step Action Block EMPLOYEE_LIST
     Procedure Step Action Block MAINTAIN_DEPARTMENT
     Process Action Block MODIFY_DEPARTMENT
     Process Action Block MODIFY_EMPLOYEE
     Procedure Step Action Block SERVER_DETAIL_DEPARTMENT
     Procedure Step Action Block SERVER_DETAIL_EMPLOYEE
     Procedure Step Action Block SERVER_MAINTAIN_DEPARTMENT
     Screen for Procedure Step EMPLOYEE_DETAIL
     Screen for Procedure Step EMPLOYEE_LIST
     Screen for Procedure Step MAINTAIN_DEPARTMENT
  ------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------
Modifying Attribute NUMBER of Entity Type DEPARTMENT would flag these
components for Regeneration:
     Module Manager for Online Load Module MENU
     Module Manager for Server Manager P900
     Process Action Block ADD_DEPARTMENT
     Process Action Block ADD_EMPLOYEE
     Process Action Block DELETE_DEPARTMENT
     Process Action Block DELETE_DIVISION
     Procedure Step Action Block EMPLOYEE_DETAIL
     Procedure Step Action Block EMPLOYEE_LIST
     Procedure Step Action Block MAINTAIN_DEPARTMENT
     Procedure Step Action Block MENU
     Process Action Block MODIFY_DEPARTMENT
     Procedure Step Action Block MENU
     Process Action Block MODIFY_DEPARTMENT
     Process Action Block MODIFY_EMPLOYEE
     Procedure Step Action Block SERVER_DETAIL_DEPARTMENT
     Procedure Step Action Block SERVER_DETAIL_EMPLOYEE
     Procedure Step Action Block SERVER_MAINTAIN_DEPARTMENT
     Procedure Step Action Block SERVER_MAINTAIN_EMPLOYEE
     Screen for Procedure Step EMPLOYEE_DETAIL
     Screen for Procedure Step EMPLOYEE_LIST
     Screen for Procedure Step MAINTAIN_DEPARTMENT
     Screen for Procedure Step MENU
------------------------------------------------------------------------
```

Regeneration Impact Analysis Report

The report only includes aggregate object changes that cause CA Gen to flag the components for regeneration.

Instructions for generating JCL for a batch job are at the end of the procedure.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 3 for Regeneration impact analysis report. Press Enter.

   The Regeneration Impact Analysis Report panel displays.
3. Type the Host Encyclopedia name of the model.

   Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a model name from the list by typing any character except a period (.) or a space character next to the name, and press Enter.

   a. Scope the business system by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

   b. Type a business system name only when regenerating a single business system. To regenerate all business systems, leave this field blank.

Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a business system name from the list by typing any character except a period (.) or a space character next to the name and press Enter.

c. Scope the load module by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

d. Select the Execution mode, online, or batch.

4. Press Enter.

The Regeneration Options panel displays as shown in the following illustration:

```
                                Regeneration Options
COMMAND ===>

 To limit regeneration by date, type a beginning and/or ending date.
 To limit regeneration to a specific user, type the TSO user ID.
 To limit regeneration to a specific group, type the Group ID.
 To ignore Target Environment Parameter changes, select Yes.
 To mark and record components requiring regeneration, select Yes.

 Begin date  . . . . . .  0___ - 0_ - 0_ YYYY-MM-DD or 0
 End   date  . . . . . .  0___ - 0_ - 0_ YYYY-MM-DD or 0
 User ID . . . . . .  ________ TSO user ID
 Group ID  . . . . . .  ________ Group ID
 Ignore Target Environment Parameter Changes . . . . .  / Yes
                                                        . No
 Mark & record components requiring regeneration . . .  / Yes
                                                        . No

 Limiting regeneration to a particular user/group or by date may cause
 inconsistencies at generation/installation time.  Some components that
 are not within these limits may need to be regenerated.
```

5. Specify required limitations on the scope of the regeneration for beginning or ending dates, TSO user ID, Group ID, and rejection of the target environment changes and press Enter.

Limiting regeneration to a particular user/group or by date can cause inconsistencies during generation and installation. Some components that are not within the specified limitations may need regeneration.

6. To retain the results of the analysis, select Yes for the Mark & record components requiring regeneration option. Selecting Yes updates OK Session Flag and a subsequent regeneration does not list the aggregate changes that caused CA Gen to flag the component for regeneration. Selecting No to get the list of aggregates and the corresponding components requiring regeneration and their reasons for regeneration in subsequent sessions.

7. When you select No for the Ignore Target Environment Parameter Changes option, Intelligent Regeneration can detect changes to the business system target environment parameters and RI Trigger Target Environment parameters, Dynamically Link RI Triggers option, and RI Trigger Name. Intelligent Regeneration cannot detect other changes to the RI Trigger Target Environment parameters such as DBMS type for RI triggers and it cannot detect the changes to DBMS type for RI triggers. If DBMS type has changed for RI triggers, generate and install the RI triggers using the Application System Construction tools, that is, the regular generation, tools.

   If you chose foreground processing, CA Gen starts the report processing. When complete, the Browse Report File panel displays.

   If you selected background processing, the JCL for the report displays in ISPF Edit mode and you can change the JCL.

   If the Auto-submit option on the Generation Options screen was not set to Yes, type Submit on the COMMAND line and press Enter to submit the job for processing.

   The Application System Regeneration menu displays so you can perform other processing. Omit the remaining steps in this procedure.

8. Review the report in the Browse Report File panel and click End when finished.

   The Report Print Options panel displays.

9. Press the appropriate key or type a command on the COMMAND line and press Enter.

   CA Gen returns to the Application System Regeneration menu.

The following illustration is an example of the Regeneration Impact Analysis Report.

```
2009-11-24              INTELLIGENT REGENERATION              PAGE    1
  06:31             REGENERATION IMPACT ANALYSIS REPORT

     MODEL NAME:             GEN SAMPLE MODEL
                         ALL OBJECTS CHANGED IN MODEL
                                BY ALL USERS

  All Business System components and all RI Triggers were flagged for
  Regeneration.  All Business System components and all RI Triggers were flagged
  for Installation.  Review Construction Regeneration Report for additional
  information.
```

If no aggregate changes caused CA Gen to flag components, but components were flagged in an earlier session and not regenerated at that time, or components existed that had not been generated or installed, CA Gen displays one of the following mutually exclusive messages:

- All Business Systems components and all RI Triggers were flagged for Regeneration
- All Business System components and all RI Triggers were flagged for Installation

- All Business Systems components and some RI Triggers were flagged for Regeneration
- All Business System components and [some/all] RI Triggers were flagged for Installation
- All Business Systems components and no RI Triggers were flagged for Regeneration
- All Business System components and [no/some/all] RI Triggers were flagged for Installation
- All Business Systems components and all RI Triggers were flagged for Regeneration
- [Some/All] Business System components and all RI Triggers were flagged for Installation

The following messages could also occur in this report:

- Some Business System components and some RI Triggers were flagged for Regeneration
- [Some|All] Business System components and [some|all] RI Triggers were flagged for Installation
- Some Business System components and no RI Triggers were flagged for Regeneration
- [Some|All] Business System components and [no|some|all] RI Triggers were flagged for Installation
- All RI Triggers were flagged for Regeneration
- All RI Triggers were flagged for Installation
- Some RI Triggers were flagged for Regeneration
- [Some|All] RI Triggers were flagged for Installation
- No Business System components and no RI Triggers were flagged for Regeneration
- [No|Some|All] Business System components and [no|some|all] RI Triggers were flagged for Installation

Construction Regeneration Report

Follow these steps:

1. Open the Application System Regeneration menu.
2. Type 4 for Construction regeneration report and press Enter.

   The Construction Regeneration Report panel displays.

3. Type the Host Encyclopedia name of the model.

   Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a model name from the list by typing any character except a period (.) or a space character next to the name, and press Enter.

   a. Scope the business system by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

   b. Type a business system name to regenerate a single business system. To regenerate all business systems, leave this field blank.

      Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a business system name from the list by typing any character except a period (.) or a space character next to the name and press Enter.

   c. Scope the load module by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

   d. Select the Execution mode, online, or batch.

4. Press Enter.

   The Regeneration Options panel displays.

5. Specify the limitations on the scope of the regeneration for beginning or ending dates, TSO user ID, Group ID, and rejection of the target environment changes and press Enter.

   Limiting regeneration to a particular user/group or by date cause inconsistencies during generation and installation. Some components that are not within the specified limitations need regeneration.

6. To retain the results of the analysis, select Yes for the Mark & record components requiring regeneration option. This updates OK Session Flag. A subsequent regeneration fails to list the aggregate changes that caused CA Gen to flag the component for regeneration. Select No for the list of aggregates and the corresponding components requiring regeneration and their reasons for regeneration in subsequent sessions.

7. When you select No for the Ignore Target Environment Parameter Changes option, Intelligent Regeneration can detect changes to the business system target environment parameters and RI Trigger Target Environment parameters Dynamically Link RI Triggers option and RI Trigger Name. Intelligent Regeneration cannot detect other RI Trigger Target Environment parameter changes, such as DBMS type for RI triggers. If the DBMS type changed for RI triggers, generate and install the RI triggers using the Application System Construction tools, that are the regular generation tools.

   If you chose foreground processing, CA Gen starts the report processing. When complete, the Browse Report File panel displays.

   If you selected background processing, the JCL for the report displays in ISPF Edit mode to change the JCL.

If the Auto-submit option on the Generation Options screen has not been set to Yes, type Submit on the COMMAND line and press Enter to submit the job for processing.

The Application System Regeneration menu displays to select other processing. Omit the remaining steps in this procedure.

8. Review the report in the Browse Report File panel. Press End when finished.

   The Report Print Options panel displays.

9. Press the appropriate key or type a command on the COMMAND line and press Enter.

CA Gen returns to the Application System Regeneration menu.

The following list shows other messages this report could include. The term entity triggers refers to Referential Integrity triggers of ENTITY and RELATIONSHIP types.

- All components, including entity triggers, were flagged for regeneration/installation
- All components, excluding entity triggers, were flagged for regeneration/installation
- All entity triggers were flagged for regeneration/installation
- No components were flagged for regeneration/installation

Printing Reports Using Host Encyclopedia

The reports are generated exactly as if you display them through the Intelligent Regeneration tool.

To generate a report

1. Display the CA Gen Main menu.
2. Type 1 for Host Encyclopedia functions and press Enter.

   The Host Encyclopedia Functions menu displays.

3. Type 3 for Model Management and press Enter.

   The Model Management menu displays.

4. Type 14 for Model reports function and press Enter.

   The Model Reports menu displays.

5. Type the Host Encyclopedia name of the model in the space, and select a report number:

   - 3—Construction Change Impact Report
   - 4—Construction Regeneration Report
   - 10—Regeneration Impact Analysis Report

Press Enter to start reporting.

The appropriate panel displays.

6. Type the Host Encyclopedia name of the model.

   Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a model name from the list by typing any character except a period (.) or a space character next to the name, and press Enter.

   a. Scope the business system by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

   b. Type a business system name only if you are regenerating a single business system. To regenerate all business systems, leave this field blank.

      Use the Tab key to position the cursor at the field and press the Prompt key for a list of valid choices. Select a business system name from the list by typing any character except a period (.) or a space character next to the name and press Enter.

   c. Scope the load module by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

   d. Select the Execution mode, online, or batch.

7. Press Enter.

   The Regeneration Options panel displays.

8. Specify regeneration options.

   If you chose foreground processing, CA Gen starts the report processing. When complete, the Browse Report File panel displays.

   If you selected background processing, the JCL for the report displays in ISPF Edit mode and you can change the JCL.

9. Type Submit on the COMMAND line and press Enter to submit the job for processing.

   The Application System Regeneration menu displays. Omit the remaining steps in this procedure.

10. Review the report in the Browse Report File panel and press End when finished.

    The Report Print Options panel displays.

11. Press the appropriate key or type a command on the COMMAND line and press Enter.

    CA Gen returns to the Application System Regeneration menu.

Application System Regeneration

Regeneration is done at the level of detail you specify and can be performed from these levels:

- Business System
- Load Module
- Screen
- Action Block

You can start the selection process at a higher level and cascade down to lower levels. However, only business systems, load modules, and Referential Integrity triggers can be installed.

If Referential Integrity triggers require regeneration, they must be regenerated before other flagged components. All other flagged components at the level that is selected or lower are displayed with indications of regeneration requirements.

Display the Intelligent Regeneration functions from these panels:

- Business System Regeneration panel
- Load Module Regeneration panel
- Screen Regeneration panel
- Action Block Regeneration panel
- Referential Integrity Regeneration panel

Access the Regeneration Menu

Follow these steps:

1. Display the Application System Regeneration panel.

```
                        Application System Regeneration
COMMAND ===>

 Select one of the options below, then press enter.

 _ 1. Construction change impact report
   2. Regenerate application system
   3. Regeneration impact analysis report
   4. Construction regeneration report

  F1=Help  F3=Exit  F12=Cancel
```

2. Type 2 for Regenerate application system and press Enter.

```
                    Regenerate Application System
COMMAND ===>

Type a model name or request Prompt for list selection. Select the business
system scope. If the business system scope is one business system, type a
business system name or request Prompt for list selection. Select the load
module scope, then press enter.

  Model name . . . . . .   GEN SAMPLE MODEL_____________  +

  Business System scope    / All business systems
                           . One

  Business System name     _____________________________  +

  Load Module scope  . .   / All load modules
                           . Multiple


F1=Help  F3=Exit  F4=Prompt  F12=Cancel
```

3. Type the Host Encyclopedia name of the model or use the Tab key to move the cursor to the field and press the Prompt key for a list of valid choices. Select a model name from the list by typing any character except a period (.) or a space character next to the name, and press Enter.

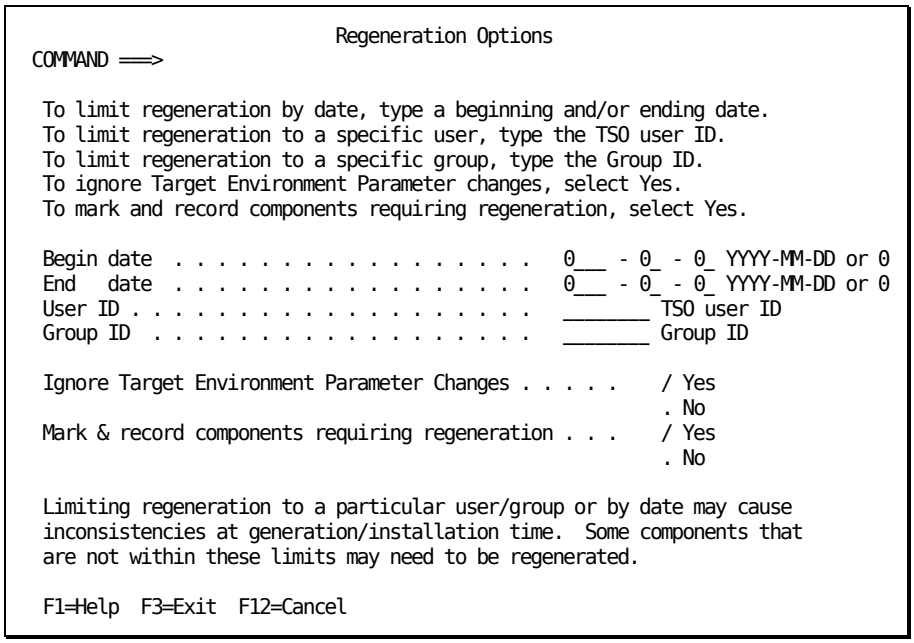
   a. Scope the business system by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option to disable.

   b. Type a business system name only if you are regenerating a single business system. To regenerate all business systems, do not type a name in this blank.

      You can also use the Tab key to move the cursor to the field and press the Prompt key for a list of valid choices. Select a business system name from the list by typing any character except a period (.) or a space character next to the name and press Enter.

   c. Scope the load module by placing a slash character (/) opposite the option to enable. Place a period character (.) opposite the option you disable.

4. Press Enter.



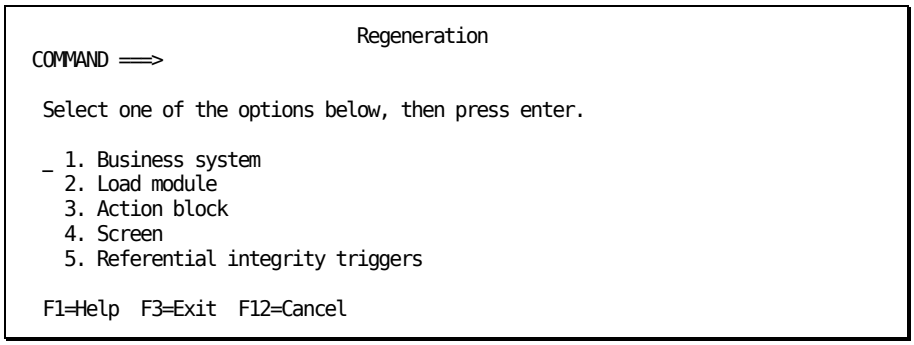
5. Specify required limitations, if any, on the scope of the regeneration about beginning or ending dates, TSO user ID, Group ID, and rejection of the target environment changes.

   Limiting regeneration to a particular user/group or by date cause inconsistencies during generation and installation. Some components that are not within the specified limitations require regeneration.

6. To retain the results of the analysis, select Yes for the Mark & record components requiring generation option. This option is useful if no or minimal changes had been made to a model since the last regeneration session but components had been marked for regeneration in that earlier session and the user chose not to regenerate them. If this option is set to No, the next intelligent regeneration operation must extract all of the aggregate changes that are made to the model since the earliest generation date for the set of components. This would degrade performance.

7. When you select No for the Ignore Target Environment Parameter Changes option, Intelligent Regeneration can detect changes to the business system target environment parameters and RI Trigger Target Environment parameters Dynamically Link RI Triggers option and RI Trigger Name. Intelligent Regeneration cannot detect other RI Trigger Target Environment parameter changes, such as DBMS type for RI triggers. If the DBMS type has changed for RI triggers, generate and install the RI triggers using the Application System Construction (regular generation) tools.

8. Press Enter.



9. Select the appropriate level of detail for the regeneration.
   - When Referential Integrity triggers need regeneration, CA Gen selects that option automatically. The Referential Integrity triggers must be generated first.
   - When you select Business system, you receive all the business systems within your model that is flagged for regeneration and you can expand components to their lower levels.
   - When you select Load module, you receive all the load modules within your model that is flagged for regeneration and can expand components to lower levels.
   - When you select Action block, you receive all action blocks within your model that is flagged for regeneration.
   - When you select Screen, you receive all the screens within your model that is flagged for regeneration.
   - When CA Gen determines that Referential Integrity triggers require regeneration, it displays the Referential Integrity Processing Options panel to select generation options for the RI triggers.

Business System Regeneration

Use the Business System Regeneration panel to access the highest level of regeneration available. From this panel, you can expand the view of components and can access the lower-level panels. For each business system, the panel shows you:

- Business system name
- Installation only status
- Number of components to regenerate
- Number of load modules to reinstall
- Procedure Step total

- Action Block total
- Screen total

```
                        Business System Regeneration          Row 1 to 1 of 1
COMMAND ===>                                                   SCROLL ===> CSR

 Model name :  GEN SAMPLE MODEL
 Type any action code next to one or more business system names, then press
 enter.

   E=List components   R=List reason for regeneration
   G=Generate          T=Set target environment (for non-server modules)
   I=Install

    INSTALL                                  Components      Load modules
Opt  ONLY      Business System              to regenerate     to install
 _     NO  CORPORATE_MANAGEMENT                    20               1
*************** TOTALS ****************        20               1
```

The panel also totals the counts of components and load modules at the bottom. The component totals include each action block once for the model. If you are using common action blocks, the action block shows in each load module that references it, but is counted for regeneration only once. CA Gen only shows components that are flagged for regeneration.

Use the Business System Regeneration panel to perform these tasks:

Specify Target Environment

Select T to display the Specify the Target Environment and Construction Libraries menu to set the target environment parameters.

Regenerate Business System

Select G to regenerate each component marked for regeneration in every load module marked for regeneration in the business system. This can include:

- Dialog Managers
- Batch Managers
- Server Managers
- Procedure steps
- Screens
- Action blocks

After testing, when you are ready to regenerate for the production environment, use this option with the appropriate changes in environment parameters.

Expand a Business System

Type E next to a business system name to display the Load Module Regeneration panel that lists all load modules that comprise the business system and were flagged for regeneration.

Use the Load Module Regeneration panel to expand other levels.

List Reason for Regeneration

Select R to produce a Construction Regeneration Report and display it in the Browse Report Form panel.

Install a Business System

Select I to install the regenerated load modules in a business system, locally or remotely, including all processing from compile to link edit and bind of the database plan.

If the business system contains load modules with components marked for Compatibility, that are built, ensure that the Process modules marked for Compatibility option is selected. Selecting Yes causes the action blocks statically called by the modules marked for Compatibility to be compiled twice, once using the NODLL compiler option and again using the DLL compiler option. The NCAL link edit of the NODLL compile is placed in the Compatibility Static non-DLL NCAL Modules library while the NCAL link edit of the DLL compile is placed in the Internal NCAL Load Modules library. Selecting No causes the action blocks to compile using the DLL compiler option and the result of the NCAL link is placed in the Internal NCAL Load Modules library only.

When the load modules are installed, CA Gen retrieves the action blocks from the appropriate library and installs them with the load module.

More information:

Specify Target Environment for HE (see page 77)

Specify the Target Environment

To set target environment and construction library parameters from within the Intelligent Regeneration dialog, use this procedure.

Follow these steps:

1. Open the Application System Regeneration menu.
2. Type 1 for Business system.

   The Business System Regeneration panel displays.
3. Type a T in the selection column next to a business system name and press Enter.

   The Specify Target Environment and Construction Libraries panel displays.

4. Type 1 for Specify Target Environment and press Enter.

   The Specify Target Environment Parameters panel displays.

5. Set the target environment parameters as required and press Accept or Save to store your changes.

   Note: Intelligent Regeneration to create code for more than one business system at a time. If you use the Accept command to override the Target Environment and Construction Libraries for a business system, generate code only for that business system. The overrides apply only to the first business system encountered during the generation process.

6. Type 2 for Construction libraries, if needed, and press Enter.

   The Specify Construction Libraries menu displays.

7. Type the number for the required library.

   If you selected internal libraries, the Specify Internal Libraries panel displays.

   If you selected internal compatibility libraries, the Specify Internal Compatibility Libraries panel displays.

   If you selected one of the external libraries, one of these panels displays:

   - Specify External Action Block Load Libraries
   - Specify External Action Block DBRM Libraries for DB2
   - Specify External System Load Libraries
   - Specify External Compatibility Libraries

8. Type the library names.
9. Press Save to save the changes or Cancel to ignore them.
10. Press End to return to the Business System Regeneration panel when all choices are complete.

More information:

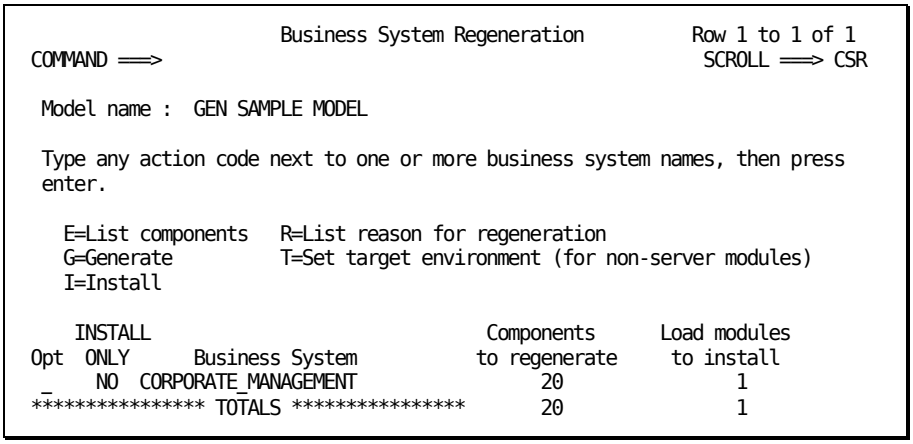
Specify Target Environment for HE (see page 77)

Regenerate a Business System

Use the Business System Regeneration panel to regenerate each component marked for regeneration in every load module marked for regeneration in the business system.

Follow these steps:

1. Open the Application System Regeneration menu.
2. Type 1 for Business system and press Enter.



```
                        Business System Regeneration          Row 1 to 1 of 1
COMMAND ===>                                                   SCROLL ===> CSR

 Model name :  GEN SAMPLE MODEL

 Type any action code next to one or more business system names, then press
 enter.

   E=List components   R=List reason for regeneration
   G=Generate          T=Set target environment (for non-server modules)
   I=Install

    INSTALL                                  Components      Load modules
Opt  ONLY      Business System             to regenerate     to install
 _     NO  CORPORATE_MANAGEMENT                   20               1
*************** TOTALS ***************        20               1
```

3. Type a G in the selection column next to the business system to regenerate and press Enter.

   The Regeneration Options panel displays.

4. Override the default values, if necessary. Press Enter to continue.

   The Confirm Regeneration of Business System panel displays.

5. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track regeneration progress.

6. Press Enter to exit the Generation Status panel when generation completes.

   If you chose background processing, the Update JCL for Background Code Generation panel displays.

7. Type changes to your Job card. Press Enter to submit the job.

Every panel includes an option to create a report that lists the reasons that a component requires regeneration.

To list reasons for regeneration

1. Type an R in the selection column next to the component and press Enter.

   The Browse Report File panel displays listing the Construction Regeneration Report for the components selected.

2. Press End to leave the Browse Report File panel.

   The Report Print Options panel displays.

3. Type the letter or the required print option on the COMMAND line and press Enter.

CA Gen displays the previous panel.

Install a Business System

Use the Business System Regeneration panel to install the regenerated load modules in a business system, locally or remotely, including all processing from compile to link edit and bind of the database plan.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 1 for Business system.

   The Business System Regeneration panel displays.

3. Type I in the selection column next to a business system name and press Enter.

   The Installation Options panel displays.

4. Select the required installation conditions and press Enter.

   The Confirm Installation panel displays.

5. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track CA Gen's progress in regeneration.

6. Press Enter to exit the Generation Status panel when generation is complete.

   If you selected background processing, the Update JCL for Background Code Generation panel displays.

7. Type changes to your Job card. Press Enter.

Load Module Regeneration

Use the Load Module Regeneration panel to regenerate a load module or its components. From this panel, you can expand a load module to see the individual components that comprise a load module, such as the Dialog Manager, procedure steps, and action blocks.

For each load module, the panel displays:

- Install only status
- Load module name
- Load module type, such as online, batch, server, or z/OS Library
- Business system name
- Number of module managers, such as dialog, batch, and server managers, to regenerate
- Number of action blocks, including procedure steps, to regenerate
- Number of screen managers to regenerate

The panel also totals the counts of each component type at the bottom.

Use the Load Module Regeneration panel to perform these tasks:

Regenerate a Load Module

Select G to regenerate each component of the selected load module, including:

- Dialog Managers
- Batch Managers
- Server Managers
- Procedure steps
- Screens
- Action blocks

Expand a Load Module

Select E for a load module to display the Load Module Component Regeneration panel to list all the components that are flagged for regeneration in the load module, from which you can select individual components for regeneration.

List Reason for Regeneration

Select R to produce a Construction Regeneration Report and display it in the Browse Report Form panel.

Install a Load Module

Select I to compile, link edit, and bind the selected (regenerated) load modules, locally or remotely, including all processing from compile to link edit and bind of the database plan.

If the load module contains components that are marked for Compatibility, that are built, ensure that the Process modules marked for Compatibility option is selected.

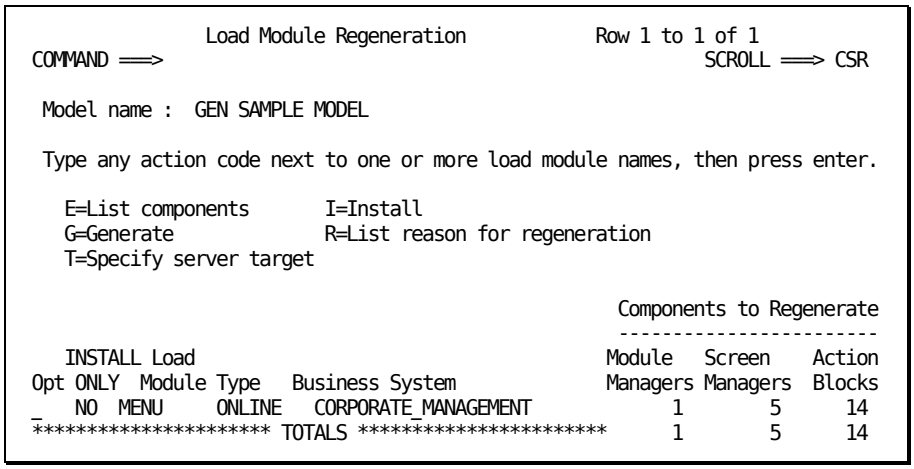
Selecting Yes causes the action blocks statically called by the modules marked for Compatibility to be compiled twice, once using the NODLL compiler option and again using the DLL compiler option. CA Gen places the NCAL link edit of the NODLL compile in the Compatibility Static non-DLL NCAL Modules library and the NCAL link edit of the DLL compile in the Internal NCAL Load Modules library.

Selecting No causes CA Gen to compile the action blocks using the DLL compiler option and CA Gen places the result of the NCAL link in the Internal NCAL Load Modules library only.

When the load modules are installed, CA Gen retrieves the action blocks from the appropriate library and installs them with the load module.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 2 for Load module and press Enter.



```
                Load Module Regeneration            Row 1 to 1 of 1
COMMAND ===>                                                SCROLL ===> CSR

 Model name :  GEN SAMPLE MODEL

 Type any action code next to one or more load module names, then press enter.

   E=List components       I=Install
   G=Generate              R=List reason for regeneration
   T=Specify server target

                                                    Components to Regenerate
                                                    ------------------------
   INSTALL Load                                    Module   Screen    Action
Opt ONLY  Module Type   Business System             Managers Managers  Blocks
_    NO  MENU     ONLINE   CORPORATE_MANAGEMENT          1        5      14
********************* TOTALS **********************     1        5      14
```

3. To specify or modify the target environment for a server load module, type T in the selection column next to the server load module name and press Enter.

   The Specify Server Target Environment panel displays. The panel shows the target operating system, source language, DBMS, and TP monitor for the server load module. Modify parameters if required and type a command key to exit.

4. Type a G in the selection column next to the load modules to regenerate and press Enter.

   The Regeneration Options panel displays.

5. Override the default values, if necessary and press Enter to continue.

   The Confirm Regeneration of Load Module panel displays.

6. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track CA Gen's progress in regeneration.

7. Press Enter to exit the Generation Status panel when generation is complete.

8. If you selected background processing, the Update JCL for Background Code Generation panel displays.

9. Type changes to your Job card and press Enter.

To install a load module

1. Open the Application System Regeneration menu.

2. Type 1 for Business system.

   The Business System Regeneration panel displays.

3. Type an E next to a business system name and press Enter.

   The Load Module Regeneration panel displays.

4. Type I in the selection column next to a load module name and press Enter.

   The Installation Options panel displays.

5. Select the required installation conditions and press Enter.

   The Confirm Installation panel displays.

6. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track CA Gen's progress in regeneration.

7. Press Enter to exit the Generation Status panel when generation is complete.

   If you selected background processing, the Update JCL for Background Code Generation panel displays.

8. Type changes to your Job card and press Enter.

More information:

Specify Server Target Environment (see page 141)

Load Module Component Regeneration

The Load Module Component Regeneration panel list all the components that are flagged for regeneration in the load module, from which you can select individual components for regeneration. To display the Load Module Component Regeneration panel, select E in the Load Module Regeneration panel.

Use the Load Module Component Regeneration panel to perform these tasks:

Regenerate a Load Module Component

When you select G for a load module component, CA Gen regenerates each component of the selected load module.

List Reason for Regeneration

Select R to produce a Construction Regeneration Report and display it in the Browse Report Form panel.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 2 for Load module and press Enter.

   The Load Module Regeneration panel displays.
3. Type an E in the selection column next to a load module name.

   The Load Module Component Regeneration panel displays.
4. Type a G in the selection column next to the load module components to regenerate and press Enter.

   The Regeneration Options panel displays.
5. Override the default values, if necessary, and press Enter to continue.

   The Confirm Regeneration of Load Module panel displays.
6. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track CA Gen's progress in regeneration.

7. Press Enter to exit the Generation Status panel when generation completes.

   If you selected background processing, the Update JCL for Background Code Generation panel displays.

8. Type changes to your Job card and press Enter.

Action Block Regeneration

Use the Action Block Regeneration panel to regenerate selected action blocks, including foreign action blocks. When an action owned by one business system is USEd by an action block in another business system, it is referred to as a foreign action block. When you select automatic installation, intelligent regeneration also compiles a foreign action block, if the selected business system was the owning business system or if all business systems were selected. Otherwise, use Application Construction to compile.

The Action Block Regeneration panel displays the action block name, the associated business system name, the action block member name, and the name and type of the load module in which it is used.

Use the Action Block Regeneration panel to perform these tasks:

Regenerate an Action Block

Select G for an action block to regenerate the selected action block. You can select more than one action block for regeneration at a time.

List Reason for Regeneration

Select R to produce a Construction Regeneration Report and display it in the Browse Report Form panel.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 3 for Action block and press Enter.

   The Action Block Regeneration panel displays.

3. Type a G in the selection column next to the action blocks to regenerate and press Enter.

   The Regeneration Options panel displays.

4. Override the default values, if necessary, and press Enter to continue.

   The Confirm Regeneration of Action Block panel displays.

5. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track regeneration progress.

6. Press Enter to exit the Generation Status panel when generation completes.

   If you selected background processing, the Update JCL for Background Code Generation panel displays.

7. Type changes to your Job card and press Enter.

Screen Regeneration

Use the Screen Regeneration panel to regenerate screens in a load module. The panel displays the screen name, the screen member name, and the associated business system name.

Use the Action Block Regeneration panel to perform these tasks:

Regenerate a Screen

Select G for a screen to regenerate the screen. You can select multiple screens to regenerate at a time.

List Reason for Regeneration

Select R to produce a Construction Regeneration Report and display it in the Browse Report Form panel.

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 4 for Screen and press Enter.

   The Screen Regeneration panel displays.

3. Type a G in the selection column next to the action blocks to regenerate and press Enter.

   The Regeneration Options panel displays.

4. Override the default values, if necessary.
5. Press Enter to continue.

   The Confirm Regeneration of Action Block panel displays.

6. Press Enter to begin processing or press Cancel to stop processing.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track regeneration progress.

7. Press Enter to exit the Generation Status panel when generation is complete.

   If you selected background processing, the Update JCL for Background Code Generation panel displays.

8. Type changes to your Job card and press Enter.

Regenerate Referential Integrity Triggers

When Referential Integrity triggers require regeneration, CA Gen automatically displays the Referential Integrity Processing Options panel. Use this panel to regenerate the referential integrity triggers for the model.

Note: you can select this option from the menu when you know the RI triggers requiring regeneration.

Use the Referential Integrity Processing Options panel to perform these tasks:

Install triggers

Choose to install the RI triggers after regeneration.

Process in foreground

Choose to regenerate in the foreground or to generate JCL for a batch job. If you select to generate in the background, CA Gen displays the Confirm Referential Integrity Trigger Regeneration panel. Press Enter to begin processing. CA Gen displays the JCL Job card for modifications. The Job card must be present for generation to complete. If you do not intend to submit the job immediately, save it. CA Gen does not automatically save the job.

Remote installation

Choose local or remote installation. If you select remote, the regenerated triggers become part of the RI trigger remote file.

Install RI Triggers

When CA Gen displays the Referential Integrity Processing Options panel, you have the opportunity to set trigger installation to Yes. If you set this option to Yes, CA Gen compiles the triggers according to the value selected for the option Process modules that are marked for Compatibility.

Selecting No causes the RI triggers to be precompiled and compiled once using the DLL compiler option and the result of the NCAL link is placed in the RI Trigger Compiled Load Modules library.

Selecting Yes causes the RI triggers to be precompiled once but compiled twice, once using the NODLL compiler option and again using the DLL compiler option. The NCAL link edit of the NODLL compile is placed in the RI Trigger Compatibility Compiled non-DLL Load Modules library while the NCAL link edit of the DLL compile is placed in the RI Trigger Compiled Load Modules library.

When the load module is installed, CA Gen retrieves the RI triggers modules from the appropriate library and installs them with the load module.

Regenerate RI Triggers

Follow these steps:

1. Display the Application System Regeneration menu.
2. Type 5 for Referential integrity triggers, if you know that the RI triggers require regeneration. Press Enter.

   The Referential Integrity Processing Options panel displays.

   If the RI triggers require regeneration, CA Gen automatically displays the Referential Integrity Processing Options panel. Regenerate RI triggers before regenerating a business system or components.

   Intelligent Regeneration cannot detect the changes to the DBMS type for RI triggers, even when you select No for the Ignore Target Environment Parameter Changes option. The RI trigger environment parameters are stored in the ISPF environment variables, not in the model. If the DBMS type changed for RI triggers, generate and install the RI triggers using the Application System Construction tools, that is, the regular generation tools.

   Intelligent Regeneration detects the changes to the Dynamically Link RI Triggers option and RI Trigger Name when you select No for Ignore Target Environment Parameter Changes option. However, Intelligent Regeneration marks all RI triggers as requiring regeneration and reinstallation, even though the RI target parameters do not require regeneration in this case. Intelligent Regeneration compares RI trigger module generation date and time with the RI target parameters change date and time when deciding to regenerate.

To avoid regenerating RI trigger modules, install RI triggers using the Application System Construction tools, that is, the regular installation tools, after modifying RI target parameters Dynamically Link RI Triggers option and RI Trigger Name.

3. Type a slash character (/) next to the options to enable and press Enter.

   The Referential Integrity Trigger Generation Confirmation panel displays.

4. Press Enter to begin regeneration. Press Cancel if you do not want to regenerate the RI triggers now.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track regeneration progress.

5. Press Enter to exit the Generation Status panel when generation is complete.

   If you selected background processing, the Update JCL for Background Code Generation panel displays.

6. Type changes to your Job card and press Enter.

More information:

Regenerate Referential Integrity Triggers (see page 287)

Install Regenerated RI Triggers

Follow these steps:

1. Display the Regeneration menu.
2. Type 5 for Referential integrity triggers, if you know that the RI triggers require regeneration. Press Enter.

   The Referential Integrity Processing Options panel displays.

   Note: Type the number for a processing level. If the RI triggers require regeneration, CA Gen automatically displays the Referential Integrity Processing Options panel. Regenerate RI triggers before regenerating a business system or components.

3. Type a slash character (/) next to the options to enable and press Enter.

   The Referential Integrity Trigger Generation Confirmation panel displays.

4. Press Enter to begin regeneration or press Cancel to skip regenerating RI triggers now.

   If you cancel processing, the Business System Regeneration panel displays.

   If you chose foreground processing, the Generation Status panel displays.

   The Generation Status panel displays messages that track regeneration progress.

5. Press Enter to exit the Generation Status panel when generation is complete.

   If you chose background processing, the Update JCL for Background Code Generation panel displays.

6. Type changes to your Job card and press Enter.

More information:

Regenerate Referential Integrity Triggers (see page 287)

Chapter 13: Testing Host Encyclopedia Construction

Introduction

Host Encyclopedia Construction is an optional CA Gen tool that generates the executable components of a block-mode system that targets z/OS. It also generates remote and local server procedures, and includes all programs, job control statements, screen formats, and transaction definitions.

This chapter describes how to use a small test model to verify that the Host Encyclopedia Construction installed correctly, before you make the tool available to users. You can use the CA GEN SAMPLE MODEL to test.

When DB2 is the target DBMS, execute the following tasks to test the Host Encyclopedia Construction:

- Verify the Runtime Profile (RPROF) table for DB2
- Select a test model
- Allocate construction libraries
- Verify target environment parameters
- Specify construction libraries
- Generate DB2 objects
- Set generation options
- Specify collection name and package bind defaults if using DB2 Package Bind
- Generate RI trigger modules
- Review load module packaging
- Generate and install load modules
- Test the DB2 application

Verify the Runtime Profile (RPROF) Table for DB2

The first task to verify the Host Encyclopedia Construction tool that is installed correctly, is to ensure you have a Runtime Profile (RPROF) table for each DB2 subsystem that is used with CA Gen-generated applications.

More information:

The Runtime Profile (RPROF) Table for DB2 (see page 309)

Select a Test Model

Use the Gen Sample Model as the model in the remaining tasks to build the test application. To use your own model, substitute the <*name*> of your model for Gen Sample Model.

To Use Your Own Model to Test

To use your own model, ensure that it fulfills the following requirements:

- Analysis is complete. The data model passes consistency check, and the elementary processes for the business systems have been scoped.
- Business System Design is complete. The design includes a dialog flow, procedures, action blocks, and screens that passed consistency check.
- The data model is transformed to a data structure, and the data structure passes consistency check.

If You Select the CA Gen Sample Model for Test

The CA Gen Sample Model generates a simple, the fully functional Human Resources application that passes consistency checks at all levels. After installing the CA Gen Sample Model, use it to test the Code Generation, Application Installation, and Database Generation.

Note: To install the CA Gen Sample Model for the IMS or CICS target environment, see the *z/OS Installation Guide*.

Install the CA Gen Sample Model

If you have not loaded the CA Gen Sample Model to the Encyclopedia, use one of the following methods to load it:

- Run CEJOB12 to install the CA Gen Sample Model, use member SAMPMODL. See the JCL for complete instructions about customizing and running CEJOB12.
- Copy member SAMPMODL from the TRAN library on the installation tape to (<*tiupref*>IEF.TRAN<*tiusufx*>). Type %IEFUP at the TSO Ready prompt, and press Enter.

Allocate Construction Libraries

The construction libraries are required to generate the source code and to install load modules. Additional data sets are required to generate MFS, batch procedures, or to keep listings.

Use the list of required DCB specifications and initial 3390 cylinder size recommendations when allocating internal libraries. All data sets are allocated as partitioned (DSORG=PO).

Partitioned Data set Description	LRECL	RECFM	Block Size	No. of Blocks	No. of Directory Blocks
Business System NCAL Load Modules See notes 1, 2, 3	0	U	19069	300	100
Business System Static non-DLL NCAL Modules See notes 1, 2, 3, 5	0	U	19069	300	100
COBOL Listings (optional)	133	FBA	3857	8250	100
Static non-DLL NCAL Listings (optional) See note 5	133	FBA	3857	8250	100
COBOL Source	80	FB	3120	4875	50
DB2 DBRM Modules (required for DB2 only)	80	FB	3120	975	50
Executable Load Modules See notes 1, 2, 3, 4	0	U	19069	300	100
Executable Batch Dynamic Action Block Library See notes 1,2,3	0	U	19069	300	100
Generated MFS Source (required for MFS only)	80	FB	3120	975	50
Generated Batch JCL (required for batch only)	80	FB	3120	975	50
Installation Control	80	FB	3120	975	50
Binder Control Cards	80	FB	3120	975	50

Partitioned Data set Description	LRECL	RECFM	Block Size	No. of Blocks	No. of Directory Blocks
Batch Binder Control Cards	80	FB	3120	975	50
RI Trigger COBOL Listings (optional)	133	FBA	3857	8250	100
RI Trigger Compatibility Compile Listings (optional) See Note 5	133	FBA	3857	8250	100
RI Trigger COBOL Source	80	FB	3120	4875	50
RI Trigger DBRM Modules - required for DB2 only	80	FB	3120	975	50
RI Trigger Installation Control	80	FB	3120	975	50
RI Trigger NCAL Modules See note 3	0	U	19069	300	100
RI Trigger Compatibility NCAL Modules See notes 3 and 5	0	U	19069	300	100
RI Executable Load Modules See note 6	0	U	19069	300	100
RI Executable Batch Load Modules See note 6	0	U	19069	300	100
RI Binder Control Cards See note 6	80	FB	3120	975	50
RI Batch Binder Control Cards See note 6	80	FB	3120	975	50

Note:

1. Blocksize for these data sets are the maximum that is allowed for the DASD device.
2. Do not use the same library for NCAL and executable load modules.
3. These data sets must be allocated as PDSEs.

4. If the Executable Load Module data set is included in the CICS or IMS data set concatenation, its block size must be the same or smaller than the block size of the other data sets in the concatenation list.
5. These data sets are only used if the Process modules marked for Compatibility option is selected. They contain the NCALs produced for the non-DLL compiled RI triggers or Action Blocks.
6. These data sets are only used if the Dynamically Link RI Triggers option is set to Yes.

The USERLIB job allocates these libraries that are required for the test:

NCAL

Business System NCAL Load Modules

LISTINGS

COBOL Compiled Listings (applications)

COBOL

COBOL Source (applications)

DBRM (for DB2 only)

DB2 DBRM Modules (applications)

LOAD

Executable Load Modules (applications)

BATCHLIB

Executable Batch Dynamic Action Block Library for dynamic batch modules.

MFS (for IMS only)

Generated MFS Source (applications)

JCL

Generated Batch JCL

INSTC

Installation Control (application modules)

BNDCTL

Binder control cards (application modules)

BATCHBND

Batch Binder control cards for dynamic batch modules.

RILIST

RI Trigger COBOL Compiled Listings

RICOBOL

RI Trigger COBOL Source

RIDBRM (for DB2 only)

RI Trigger DBRM Modules

RIINSTC

RI Trigger Installation Control

RIBNDCTL

RI trigger binder control cards

RIBATBND

RI trigger batch binder control cards

RINCAL

RI Trigger NCAL Modules

NCALC

Compatibility NCAL Load Modules

RIEXEC

RI Trigger Executable Load Modules (Dynamic RI - DLL)

RIBATLIB

RI Trigger Executable Batch Load Modules (Dynamic RI - DLL)

LISTINGC

Compatibility COBOL Compiled Listings

RINCALC

Compatibility RI Trigger NCAL Modules

RILISTC

Compatibility RI Trigger COBOL Compiled Listings

Edit USERLIB and submit the job to allocate the libraries in this table.

To edit USERLIB:

a. Compare the names and block sizes to your site's standards.

b. Remove unwanted DD statements.

Note: You can make a version of USERLIB available to others doing code generation to assist them in allocating their libraries.

Verify Target Environment Parameters

The target environment parameters are set for the CA Gen Sample Model.

Follow these steps:

1. Select option 4 on the CA Gen Main Menu and press Enter.

   The Application System Menu displays.

2. Select option 1 and press Enter.

   The Application System Construction screen displays.

3. Type the model and business system, GEN SAMPLE MODEL, and CORPORATE_MANAGEMENT and press Enter.

   The Application System Construction Menu displays.

4. Select option 5 and press Enter.

   The Specify Target Environment and Construction Libraries Menu displays.

5. Select option 1 and press Enter.

   The Specify Target Environment Parameters screen displays.

6. Enter the DB2 Subsystem that you are using, and verify the other parameters on the Specify Target Environment Parameters screen.

7. Press F6 to save your changes and return to the Specify Target Environment and Construction Libraries Menu.

Specify Construction Libraries

This procedure uses the library names from the USERLIB job.

Follow these steps:

1. Select option 2 on the Specify Target Environment and Construction Libraries Menu and press Enter.

   The Specify Construction Libraries Menu displays.

2. Select option 1 and press Enter.

   The Specify Internal Libraries screen displays.

3. Type the library names from the USERLIB job. The DB2 DBRM Modules library is only required when the target DBMS is DB2.

4. Press F6 to save your changes and exit the screen.

5. Continue pressing F3 until you return to the Application System Construction Menu.

Generate DB2 Objects

This procedure creates the DDL and JCL statements that allocate and construct DB2 objects.

Follow these steps:

1. Select option 4 on the Application System Functions Menu and press Enter.

   The Data Base List screen displays.

2. Type A in the Opt column, next to the database name GENDB, and press Enter.

   The DDL statements and VSAM definitions for the database are generated and the DDL Options screen displays.

   The database name is used to generate JCL to create DB2 objects. You can change the name in the generated DDL without adverse effects on the application.

3. Type Y or N for the options on the DDL Options screen and press Enter.

   The Generated DDL Text screen displays the DB2 objects for which DDL is generated.

4. Type GO on the command line and press Enter.

   The JCL Job Statement Information Input screen displays.

5. (Optional) Type a job card for the JCL containing DDL for the CA Gen Sample Model, or change the default job card.

6. Press Enter to create the DDL and JCL to generate DB2 objects.

   The generated JCL displays in ISPF edit mode.

7. (Optional) On the JCL, you can change the following entities:

   - DB2 VSAM names
   - Volumes
   - Tablespace names
   - Index names

   Note: Do not change the table or column names here. Instead, check out the model and change the names in the model at your workstation.

   The JCL for DDL resides in a data set allocated as:

   `<userid>.TICTEMP1.TIDDLJCL`

   or

   `<userid>.TICTEMP2.TIDDLJCL`

   If the database is preallocated, delete the lines in the JCL for creating the database.

8. Submit the job to create database objects.

9. Verify the job that is completed successfully.
10. Continue pressing F3 until you return to the Application System Menu.

Set Generation Options

Follow these steps:

1. Select option 4 on the Application System Menu and press Enter.

   The Generation Options screen displays.

2. Select the appropriate generation options.

   When generating an application that targets DB2, ensure that the Target TSO test facility option is set to Yes.

3. Press F2 to save your changes and exit the screen.

Using DB2 Package Binds

Use this procedure when the target DBMS is DB2 and turn on DB2 Package Binds.

Note: The collection name must be a valid name. If you do not know what this name is, see your DB2 administrator.

Follow these steps:

1. From the CA Gen Main Menu, select option 4 and press Enter.

   The Application System Functions Menu displays.

2. Select option 5 from the Application System Functions Menu.

   The Bind Options Menu displays.

3. Select option 1 (Model) from the Bind Options Menu.

   The Set Package Defaults for the Model screen displays.

4. Type a valid collection name.

   You may also enter other bind options.

   Press Prompt for context-sensitive Help for each entry.

5. Press Accept to verify your entries.
6. Press Save to save your entries.

   The Bind Options Menu displays.

7. Press End to return to the Application Systems Functions Menu.

Generate RI Trigger Modules

These procedures generate the source and executable referential integrity (RI) trigger modules that maintain the relationships between primary and foreign keys in the physical tables.

Follow these steps:

1. Select option 2 on the Specify RI Trigger Target Environment and Construction Libraries Menu and press Enter.

   The Specify RI Trigger Libraries screen displays.

2. Enter the names of the libraries that are allocated by the USERLIB job.
3. Press F6 to update the model and exit the screen.
4. Press F3 to return to the Referential Integrity Trigger Construction Menu.

To execute RI trigger generation

Follow these steps:

1. Select option 1 on the Referential Integrity Trigger Construction Menu and press Enter.

   The Referential Integrity Processing Options screen displays.

2. Select the processing options and press Enter.

   The Confirm Referential Integrity Trigger Generation screen displays.

3. Verify the generation options and press Enter.

   All RI triggers for the model are generated. The Generation Status screen shows the RI triggers being generated. If generation fails, correct problems and re-execute RI trigger generation.

4. When the RI trigger modules are generated and installed successfully, press Enter to exit the screen.
5. Continue pressing F3 until the Application System Menu displays.

Review Load Module Packaging

The CA Gen Sample Model is already packaged. Use this procedure only if you review or modify the packaging.

Follow these steps:

1. Select option 1 on the Application System Menu and press Enter.

   The Application System Construction screen displays.

2. Type or verify the model CA GEN SAMPLE MODEL and business system CORPORATE_MANAGEMENT, and press Enter.

   The Application System Construction Menu displays.

3. Select option 3 and press Enter.

   The Select Load Module Packaging Environment Menu displays.

4. Select option 1 and press Enter.

   The Package Online Load Module screen displays.

   Note: The load module name is used as the plan name.

5. Review the fields on the Package Online Load Module screen, and make changes, as needed.

6. If you changed the fields, press F6 to save the changes and exit the screen. If you did not change fields, press F12 to exit.

7. Press F3 to display the Application System Construction Menu.

Generate and Install Load Modules

Follow these steps:

1. Select option 1 on the Application System Construction Menu and press Enter.

   The Generate Business System screen displays.

2. Type G in front of each load module and press Enter.

   The Generation Options screen for the first module.

3. Select the generation options. Ensure Automatic Installation is Yes and press Enter.

   The Confirm Generation screen for the module displays.

4. Verify the generation options and press Enter.

   The Generation Status screen displays, showing the load module being generated and compiled.

The installation process links and binds the load modules that are based on the selected generation options. The load module is installed because the Automatic Installation option on the Generation Options screen is set. To install the generated load module, without this setting, type I next to the module on the Generate Business System screen and press Enter.

The Generation Status screen reports when generation completes successfully. If generation fails, modify your entries on the Specify Target Environment Parameters screen and re-execute load module generation.

5. When the load module is successfully generated, press Enter.

   The Generate Business System screen displays.

6. Press Enter and repeat the steps beginning with the Generation Options screen until all load modules are generated.

7. When all load modules are generated, press Enter to exit the Generate Business System screen.

8. Continue pressing F3 until the Application System Construction Menu displays.

Test the DB2 Application

Use these steps to test the application you only constructed. Before executing these steps, ensure you observe the following rules while using the TSO Test Facility:

- You must have at least 4 MB available in the TSO region for testing.
- Always use the END function key, usually defined as PF15, to clear the screen. DO NOT use the CLEAR key.
- Press the END function key twice to end a test and return to ISPF.

To specify the test parameters

1. Select option 2 on the Application System Construction Menu and press Enter.

   The Application Test Facility screen displays.

2. Verify the DB2 subsystem to which your application plans have bound and the application load library name where the generation application load modules reside.

3. (Optional) If DB2 or COBOL runtime routines are not on your system linklist, specify the required test libraries on the Application Test Facility screen.

To start the application

1. Press Enter on the Application Test Facility screen.

   A clear screen for transaction entry displays.

2. Enter the trancode MENU and press Enter.

   The test application's main menu displays.

To add several divisions to your application

1. Type 1 on the test application's Main Menu and press Enter.

   The Division Maintenance screen displays.

2. Type A (add) in the far left column, followed by descriptive information for each division and press Enter.

   Notes:

   a. Multiple action codes are processed simultaneously.

   b. Use the modify (M) and delete (D) action codes to refine the list of divisions.

   c. The PF7 and PF8 function keys scroll up and down, respectively, in the Division List.

3. After completing Division Maintenance, press PF3 to return to the Main Menu.

To add a department to the application and relate it to one of the divisions

1. Type 2 on the Main Menu and press Enter.

   The Department Maintenance screen displays.

2. Type A in the far left column, followed by descriptive department information, and press Enter.

   Notes:

   a. Use the modify (M) and delete (D) action codes to refine the list of departments.

   b. The PF7 and PF8 function keys scroll up and down, respectively, in the Department List.

3. After completing Department Maintenance, press PF3 to return to the Main Menu.

To add an employee to one of the division or department fields

1. Type 3 on the Main Menu and press Enter.

   The Employee List screen displays.

2. Type A in the far left column, followed by employee data, and press Enter.

   The Employee Detail screen displays.

3. Complete the Employee Detail screen and press Enter.

4. Press PF12 to return to the Employee List screen.
5. Press PF3 to return to the Main Menu.
6. When you are satisfied that Host Encyclopedia Construction is operating properly, press the END function key, defined as PF15, to exit the TSO Test Facility.

Next Steps

When developing applications that use online help, the next step in installing the Host Encyclopedia is to follow the steps in Installing the Help Model.

Note: For more information about Encyclopedia maintenance and administration, see the *Host Encyclopedia User Guide*.

Chapter 14: Installing the Help Model

Introduction

The Help Model provides a mechanism to store and retrieve text-based screen explanations, fields, and permitted values for an online help system to support a CA Gen-generated application.

The Role of the TIRHELP Exit

TIRHELP is a runtime exit, which is written in COBOL, containing code for the flow to the application Help business system. The same sample exit is included in z/OS Runtime DLLs, each used by the specific target environment, such as TSO, CICS, or IMS.

TIRHELP includes sample code as comments. Until you modify TIRHELP, it displays the message:

```
No help available.
```

After you remove the comment characters and execute the TIRHELP sample code, TIRHELP performs the following steps:

Follow these steps:

1. TIRHELP checks the flags and determines the procedure in the application help business system it flow to, that is the field description, screen description, or permitted values.
2. TIRHELP calls TIRMSG, which performs the transfer to the target procedure.
3. When the help procedures complete and it enters the exit, TIRHELP executes a SET NEXTTRAN TO CONCAT(TRIM(input tirhelp trancode) to return to the calling application.

The Role of Function Keys

Although you can select how to assign function keys, usually, PF1 invokes the HELP command, PF3 invokes the EXIT HELP command, and PF4 invokes the PROMPT command to display a list of permitted values for a field.

The HELP (PF1) and PROMPT (PF4) commands cause the dialog manager of the calling application to set appropriate Help flags. The dialog manager passes this information to the screen that calls TIRHELP.

PF1 (HELP Command)

The cursor position determines if the Help displays for a screen or a field. When the cursor is not on a field when the user presses PF1, the application displays Help for the entire screen. If the message NO HELP AVAILABLE FOR THIS SCREEN displays, a Help record is missing for the screen or the screen does not have a Help ID.

When the cursor is on a field when the user presses PF1, the application displays Help for the field. If the message NO HELP AVAILABLE FOR THIS FIELD displays, the Help record for the field is missing.

PF4 (PROMPT Command)

When the cursor is on a field that supports a list of permitted values when the user presses PF4, the help displays the list of values. If the message PROMPT IS ONLY VALID FOR ENTERABLE FIELDS displays, the list of permitted values for the field is not defined, or the field cannot accept entries.

Install the Help Model

The Help Model is a CA Gen-generated application. Upload it to your Host Encyclopedia, generate it, and install it. As an installed application, the Help Model has its own help system, identical in functionality to the help systems you are creating for your CA Gen-generated applications.

Follow these steps:

1. To install the Help Model under IMS or CICS, configure these environments for generating CA Gen applications.

   Note: For more information about configuring the IMS and CICS environment, see the *z/OS Installation Guide*.

2. Verify that the Help Model is in the Encyclopedia. If you created an Encyclopedia, the CEJOB13 job uploaded the Help Model. If the Help Model is not in the Encyclopedia, load it by executing one of the following steps:

   - Execute CEJOB13, member HELPMODL. See the JCL for complete instructions on customizing and running CEJOB13.
   - Copy member HELPMODL from the TRAN library on the CA Gen installation media to: <tiupref>IEF.TRAN<tiusufx> and type %IEFUP at the Ready Prompt and press Enter.

3. Specify target environment parameters for the two business systems in the Help Model: Help Maintenance and Application Help.

4. Specify the internal libraries using the same set of names for both business systems.

5. Identify the trancodes and load module names to use for the Help Model. There are eight trancodes for the Help Maintenance business system and three for Application Help.
6. If necessary, repackage the business systems using the new trancodes and load module names. These trancodes and load modues names must match the names in the TIRHELP user exit.
7. Copy TIRHELP from the CA Gen CEHBSAMP library to a source library.
8. Make the following changes to TIRHELP:
    a. Leave the program ID as TIRHELP.
    b. Uncomment the following statements that pertain to working storage:

        SCREEN-HELP-TRANCODE

        FIELD-HELP-TRANCODE

        FIELD-PROMPT-TRANCODE

        SCREEN-HELP-LOADMOD

        FIELD-HELP-LOADMOD

        FIELD-PROMPT-LOADMOD

        MSG-AREA

        DEST-LOADMOD

    c. Update the load module names to those used in the Application Help business system of the Help Model.
    d. Update the trancode names to those used in the Application Help business system of the Help Model.
    e. Modify paragraphs A1000, B1000, and D1000. Comment out the two lines with default code and remove the comments from the remaining code.
    f. Modify X0000 and uncomment the entire paragraph.
9. Use the MKUEXITS procedure that is supplied in the CEHBSAMP to build the TIRHELP and replace the procedure in the relevant DLL. You do not have to install the applications invoking the TIRHELP again because TIRHELP is included in the DLL.
10. Generate and install the database in your DBMS environment:
    a. Choose a Creator ID for the Help Model that differs from the Creator ID used to install the Host Encyclopedia. The Help Model uses table names that are also used by the Public Interface.
    b. You can change the technical design for the model for compatibility with your DBMS system. You cannot change the TABLE and COLUMN names.
11. Run code generation for both business systems.

12. Install the load modules.
13. (Optional) Load the Help data for the Help Model itself. Use JCL member:
    - LAPPHELP (<dataset prefix>.JCL(LAPPHELP)) if using IBM DB2 LOAD utility or
    - CAPPHELP (<dataset prefix>.JCL(CAPPHELP)) if using CA DB2 Fast Load utility.

Establish TSO Support

Follow these steps:

1. If the trancodes and load module names from the Help Model business systems were not automatically included as part of installing the Help Model applications, manually include them in the application model's TRANMAP data set.
2. In the Application Test Facility panel, specify the Application Load Library into which the CA Gen Application Help system was installed.

Start the Help Model

The Help Model trancode is in the packaging for the Main Menu procedure in the Help Maintenance business system. To start the Help Model, type the Help Model trancode, and press Enter.

```
<trancode>     Application Help Maintenance System     <date>
                          Main Menu

Screen Help ID xxxxxxxxxxxxxxxxxxxxxxxxxxxx(?=List)
Field Help ID xxxxxxxxxxxxxxxxxxxxxxxxxxxx(?=List)

Selection =>

Add/Delete Screen Help ID
Update Screen Description
Add/Delete Field Help ID
Update Field Description
Update Field Permitted Value

<error message line>
```

Note: For more information about how to enter screen IDs, field Help IDs, and text into the help Model database, see the *Block Mode Design Guide*. For more information about Encyclopedia maintenance and administration, see the *Host Encyclopedia User Guide*.

Chapter 15: The Runtime Profile (RPROF) Table for DB2

Introduction

The Runtime Profile, RPROF, manages transaction restart, scrolling, return from link, hidden data, and return from help, and stores profile information about an online dialog.

Each CA Gen online load module contains a Dialog Manager subroutine. The entry point to the Dialog Manager is TIRDMGR. TIRDMGR calls the Runtime Profile Manager subroutine, TIRPROFD, to access the RPROF.

The Runtime Profile can be a DB2 table or a CICS temporary storage queue. This chapter pertains to a DB2 table implementation.

Table Definition

The first four columns and the RPROF table column KEYFILL are used to define a unique index. When SUBPAGE equals 16, as in SUBPAGE = 16, KEYFILL causes the RPROF key to be large enough to allow one entry per SUBPAGE. The table has VARCHAR columns, but they are always filled to ensure that a record in RPROF never shares a 4K page with another RPROF record.

Note: If Type 2 Index is used, the KEYFILL field is omitted.

More information:

Sample RPROF Table Definition (see page 310)

Sample RPROF Table Definition

```
CREATE TABLE RPROF
(IETUID CHAR(8) NOT NULL,
IETAPPID CHAR(4) NOT NULL,
IETPSNUM SMALLINT NOT NULL,
SEQNUM SMALLINT NOT NULL,
CNTLOVHD VARCHAR(3072) NOT NULL,
PSTEPDTA VARCHAR(512) NOT NULL,
KEYFILL CHAR (86) NOT NULL WITH DEFAULT,
RPROF_DATE DATE NOT NULL WITH DEFAULT,
RPROF_TIME TIME NOT NULL WITH DEFAULT)
IN database.RPROF;
```

The values in the table have the following definitions:

IETUID

Contains the CICS, IMS, or TSO user ID. If the user ID is unavailable, the terminal ID is used.

IETAPPID

Contains a numeric ID representing the CA Gen business system.

IETSNUM

Contains a procedure step depth of stack number indicating the location of the procedure step in the dialog.

SEQNUM

Contains a sequence number to tie physical RPROF rows into a logical RPROF record. There can be as many as 11 physical RPROF rows per logical RPROF record, assuming no compression.

CNTLOVHD

Contains control information for the dialog flow, such as trancode and exit state.

KEYFILL

Provides the concurrency.

PSTEPDTA

Contains the procedure step export views.

RPROF_DATE

Provides the assistance in the cleanup of unused rows.

RPROF_TIME

Provides the assistance in the cleanup of unused rows.

Initially, the data that is stored in PSTEPDTA is compressed. Because of this compression, many RPROF records are stored in one RPROF row. The PSTEPDTA column is back-filled with spaces, after compression, to fill a 4K page for maximum concurrency.

How to Build RPROF Tables

The BLDRPROF job builds an RPROF table and indexes. CEINSTAL builds the BLDRPROF job when you set the Install Construction flag on the Set Encyclopedia Variables screen to Y.

One RPROF table can serve all applications on a DB2 subsystem, and possibly an entire business system. Although an RPROF can appear to be a bottleneck, a large, properly maintained RPROF can support over 400,000 transactions per day, without the contention.

The strategy that you deploy to recover the RPROF table or other table that is used by an application using RPROF determines how many RPROF tables you create. For example:

- If a CA Gen style transaction restart is important, synchronize the point of RPROF table recovery with the remaining application tables. This type of RPROF is considered a user table because other applications cannot share it, and they remain synchronized after a recovery.
- If a CA Gen style transaction restart is unimportant, many applications can share the RPROF table and it is considered a system table.

A damaged or destroyed RPROF table must be recovered because an empty RPROF causes contention problems.

When you build a new RPROF table, execute the CEJOB15R job to bind the runtime DBRM TIRPROFD to access the new RPROF table.

When you use an RPROF table that is created in a previous version of CA Gen, still execute the CEJOB15R job to bind the current version of the runtime DBRM TIRPROFD to use the existing RPROF table.

How to Improve Concurrency

The RPROF is designed for the maximum concurrency by ensuring there is no less than one logical RPROF record per DB2 data page, and one index record per DB2 index subpage.

The following techniques improve concurrency:

- Reorganize RPROF weekly to keep data and index in sequence.
- Embed FREEPAGEs in data and leave 90 percent of the index free.
- Do not make RPROF too small. A small RPROF can cause serious concurrency problems, due to too few entries in the index.
- For most applications, an RPROF table with at least 625 rows allows a satisfactory level of availability. In the applications with an high transaction volume, it is necessary to prepopulate an RPROF to add permanent rows that establish the index tree structure.
- Prepopulation is unnecessary when you use type 2 indexes.

One way to prepopulate an RPROF table is to load rows for each unique user ID, that is, IETUID. The prepopulated rows have an application ID, that is, an IETAPPID, outside the range of valid application ID numbers, so the application reset does not remove the row.

The permanent rows provide an index structure that eliminates most contention, because the high-level key entries already exist, and applications do not have to wait for index page splits or abend with a DB2 -911 error.

How to Establish RPROF Access

Multiple CA Gen applications, running on different DB2 databases, can access an RPROF table by using DB2 SYNONYMS. Each application table owner would execute:

```
CREATE SYNONYM "RPROF" FOR existing_auth_id RPROF
```

Unused Rows

Over time, an RPROF table accumulates unused rows. To detect unused rows, query the RPROF_DATE and RPROF_TIME columns in the RPROF table. You can delete unused rows but the RPROF table is more concurrent with more entries. Retain permanent rows that are used to prepopulate the RPROF, unless the table uses type 2 indexes.

How to Add Date and Time Columns

Older RPROF tables, those installed before Composer 4, do not have date and time columns. CA Gen does not require the columns, but they are helpful in maintaining the table. To add date and time columns to RPROF, use the following SQL statements:

```
ALTER TABLE RPROF
ADD RPROF_DATE DATE NOT NULL WITH DEFAULT;
ALTER TABLE RPROF
ADD RPROF_TIME TIME NOT NULL WITH DEFAULT;
```

If you add these columns, rows added before the ALTER statement contains the date and time that the ALTER statement executes. The new rows contain the correct date and time.

Chapter 16: Common User Access Standards

Common User Access for CA Gen

The CA Gen Toolset and Host Encyclopedia Construction Tools support Common User Access (CUA) standards using menu structures, standard symbols, Prompt service, and user-specified panel customization.

Using Menus

You can select an option on a CA Gen menu panel in the following ways:

- Position the cursor at the start of the COMMAND line, type the number corresponding to the option you want, and press Enter.
- Extended selection by combining commands (for example, 1.2.3) is also possible from the COMMAND line. To use extended selection, specify your default cursor position on menu panels as Command.
- Position the cursor at the choice entry field, type the number corresponding to the option, and press Enter. For an example, refer to the section on choice entry fields for menu selection. You can set your default cursor position on menu panels to Menu for automatic positioning.
- Use the Tab key or cursor keys to position the cursor and press Enter.

Standard Symbols

The CUA panels use standard graphic symbols that have particular meanings. Some symbols indicate the type of action to take. For example, a period (.) before a list item means that you can toggle that option on or off, or can select that item from a list with a slash character. Other symbols indicate that a special service is provided. For example, a plus sign (+) after an input field means that a list of selections is available for this field.

CA Gen host panels use the following standard symbols that have these meanings.

. . . .

Leader dots, to align a field literal with its data.

. . _?_?_

Data entry field, an input field into which you type data. It is padded with underscores to indicate the total length of the field.

. . :

Output field

+

Prompt indicator is the plus sign (+) following a data entry field. It indicates that a list of all permitted values for the field is available. The list of permitted values is named a selection list. From these values, you can select one entry.

To display the selection list, position the cursor at the start of the field and press PF4, or your designated Prompt function key, if you customized your function key definitions. Make your selection and press Enter. The system redisplays the panel with the selected value in the data entry field.

. Name1 . Name2

Single choice selection list of permitted values for an input field. Select one entry from the list.

The period (.) before each entry means select only one. Single choice selection lists are available when you request Prompt for a model, subset, and model family names. Single choice selection lists are also used for specifying batch or online execution of a report or function, and a range of objects for functions.

/

Default list selection indicator is the slash character (/). It is the default character for marking a selection on a selection list. Use the slash character (/) instead of S or X unless the panel content instructs otherwise. For example, when expanding an object for reporting in Intelligent Regeneration, select an object by typing S next to it but you expand the object by typing E next to it. Use the slash character (/) to mark a default selection. The selection lists that offer default values include online or batch execution mode, all objects or specific objects, and sort options for model, subset, and model family names selection lists. No action is necessary to use the default selection value. Use one of the following methods to change the default to another choice:

- Type a space character or period (.) next to the entry containing the / character
- Type / next to the required entry on the list

_ Item1 _ Item2

Multiple choice selection or action list. An underscore (_) in the leftmost column indicates a multiple choice action or selection list. You can select more than one item from the list if it is a selection list. If it is an action list, you can execute an activity in the list of activities on one or more entries in the list.

_ 1. 2.

Choice entry field for the menu selection. The underscore next to the first entry on a menu is a choice entry field. Type the number of the menu option to execute. Menus are a single choice selection list because you select only one option at a time.

More information:

Setting Defaults (see page 318)
Using Menus (see page 315)

Prompt Service

CA Gen host functions provide a number of selection lists that let you select an item from a list if you prefer not to type a value. Any input field ending with the Prompt symbol,+, supports selection list processing.

Changing Sort Options for Selection Lists

To change sort options for selection lists, use the Selection List Sort Options menu in the Environment Specification option in the Main Menu.

To change the default sort option, type a space character or a period (.) next to the entry containing the slash character (/). Type / next to the required option, and press Enter.

To use a selection list

1. Position the cursor at the beginning of the field.
2. Press the designated Prompt key.

   The default is PF4. The selection list displays containing all the items that you have authority to access.

Each selection list contains the items to which you have been given access. The period character next to each item means that you can select only one item from the list.

To select a name from the list

1. Locate the name or part of a name in the list.

   To use the ISPF Find (F) or Locate (L) command, type F or L on the COMMAND line, followed by the name or value to search for and press Enter. These commands do not require quotation marks.
2. Type a slash (/) or any character except period or a space character over the period to the left of the required item and press Enter.

   CA Gen inserts the selected value in the field on the panel from which you requested the Prompt service.

   To return to the previous panel without making a selection, press the designated Cancel function key, or type Cancel on the COMMAND line and press Enter. The selection lists do not support pressing End.

Setting Defaults

You can customize the panel colors, COMMAND line placement, function key definitions, and cursor positioning on menu panels, in the host environment or can use the defaults.

Changing COMMAND Line Placement

Follow these steps:

1. Select the Environment Specification option on the CA Gen Main menu and press Enter.
2. Select the Panel Specifications option and press Enter.

   The options for command-line placement are TOP and BOTTOM. TOP is the default.
3. Type BOTTOM in the command-line placement field and press Enter.

   The COMMAND line is at the bottom of all host panels.

Initial Cursor Placement on Menus

The cursor on host menus can initially be placed at the COMMAND line or at the choice entry field (the underscore next to the first entry on a menu). COMMAND line placement is the default.

Changing the initial placement of the cursor applies only to CA Gen menu panels. If you want to use the extended selection feature (1.2.3), use COMMAND line placement. Trying to use extended selection with choice entry cursor placement causes the path to be truncated to a single option number.

Follow these steps:

1. Select the Environment Specifications option on the CA Gen Main Menu and press Enter.
2. Select the Panel Specifications option and press Enter.
3. Type MENU in the cursor placement for menus field and press Enter.

Customizing Function Key Definitions

Follow these steps:

1. Type keys on the COMMAND line of CA Gen panels and press Enter.
2. The ISPF PF Key Definitions and Labels panel displays, showing the defaults.
3. Use the Tab key to position the cursor at the required key.
4. Type the new value and press End.

The default CA Gen function key settings and their meanings are:

PF1

Help

PF2

Accept

The Accept key is useful when selecting object occurrences in subsetting, version control, aggregate set management, and object cross-reference reports.

PF3

End

PF4

Prompt

PF5

Refresh

PF6

Save

PF7

Up

PF8

Down

PF9

Retrieve

PF10

Left

PF11

Right

PF12

Cancel

Note: Use the Accept key to move faster when selecting object occurrences in subsetting, version control, aggregate set management, and object cross-reference reports.

Displaying the Customized Function Key Legend

The minimum default CA Gen function keys that are shown are PF1=Help, PF3=Exit, and PF12=Cancel. To list all current function keys as a legend at the bottom of each panel, type PFSHOW on the COMMAND line on a CA Gen panel and press Enter.

To change the number of keys per line or the range of keys that are shown, 1-12, or 13-24. Type PFSHOW TAILOR and press Enter.

To return to the default function key legends for each panel, type PFSHOW OFF.

Changing Panel Colors

You can customize the colors used for text, input, and output fields.

Follow these steps:

1. Select the Environment Specification option on the CA Gen Main menu and press Enter.
2. Select the Panel Specifications option and press Enter.

   The Set Panel Specifications panel displays, showing the default color settings.
3. Use the Tab key to position the cursor at the field you want to change.
4. Type the new color value and press Enter.

   The bottom of the panel lists the available colors.

The default color for high intensity for text fields, input fields, and output fields is white. The default color for normal intensity text is green. Turquoise is the default color for normal intensity for input and output fields.

Cancel and Exit Keys

The basic dialog flow for CA Gen host functions conforms to CUA standards. CUA treats the Cancel and Exit commands as two distinct activities, and each having a standard or common function key that is dedicated to it.

Under CUA, the Cancel function, PF12 by default, lets you back up to the previously displayed panel. Repeated use of the Cancel function lets you back out of the function one panel at a time until you reach the menu from which you selected the option.

The Exit function, PF3 by default, stops the function that you are executing and returns to the menu from which you selected the option. Repeated use of the Exit function lets you back out of the CA Gen menu hierarchy one menu at a time until you reach the ISPF main menu.

Full-screen pop-up panels, which are used for selection lists and option-specific parameters for the Code Generation functions, do not support the Exit function because it applies to the application, not to the immediate panel being displayed. To return to the previously displayed panel without selecting an item, use the Cancel function. After cancelling the pop-up, you can Exit from the function.

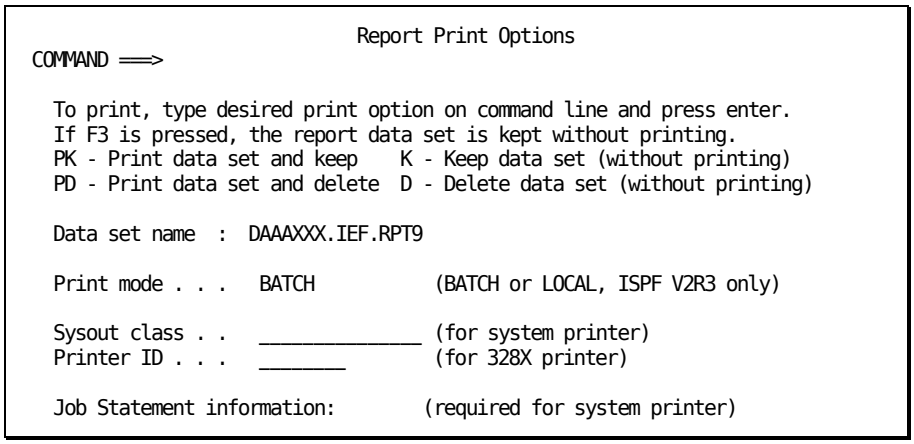
Printing User Reports

After viewing generated reports on the host, you can:

- print the reports and keep the data set
- print the reports and delete the data set
- Keep or delete the data set without printing

Follow these steps:

1. At the panel displaying the report, press the End key.



2. In the command field, type a print option:

   PK

   Print the report and keep the data set.

   PD

   Print the report and delete the data set.

K

Keep the data set without printing.

D

Delete the data set without printing.

3. To print the report, type a sysout class or print ID.
4. When using the system printer, type a Job Statement, JES statements are needed, and press Enter.

Chapter 17: Tips and Techniques

Generation Tips and Techniques

Database Generation

- Always generate and install the database before attempting to install programs that use the database. Ensure that the database is installed and active.
- Keep test data files small to avoid writing conversion routines, although this is done as part of the project. When possible, share test files with others using the same data to increase productivity.
- Use Generate and Install All DDL with DROP statements to replace a previous version of the installed table. Set the DROP statement in the DDL option to Yes on the Testing Environment Panel.
- When regenerating a database definition, specify DROP. Create a separate routine outside CA Gen to save data in a database before regenerating. Use the DBMS' import/export facility or write a conversion program.
- Detail DDL Generation enables you to generate and, optionally, install a table or an index at a time. If you use this option, be sure to include all changes.
- You employ this option to generate new indexes representing newly created entry points without dropping the base tables upon which they are built.
- The Generate and Install All DDL option reduces the number of difficult-to-locate integrity problems in the generated database definition.
- When you receive a message beginning with the word Errors:, the installation process failed. To determine the failure reason, review the results of the installation. Normally a problem is the result of an environmental error of some kind. For example, your DBMS was not properly started, or a table exists that has the same name as the one being generated.
- If you regenerate DDL, be sure to regenerate RI triggers.

System Generation and Installation

- It is not necessary to regenerate an entire application system when moving from one TP monitor to another. For example, only reinstallation is required when moving from TSO to CICS. Be careful not to install load modules containing action blocks that are generated with the debug option into IMS or CICS.

- Regeneration is required to move from one TP monitor to another when one of the TP monitors is IMS. The screen modules that are generated for BYPASS differ from those generated for MFS in the handling of literals and prompts. Therefore, screen modules must be generated when moving from TSO or CICS (BYPASS) to IMS (MFS) or vice versa.
- When installing RI Triggers or Action Blocks statically called by modules that are marked for Compatibility, use the option Process modules that are marked for Compatibility to ensure that the compile is done using the NODLL option.

Intelligent Regeneration

- n the workstation, use Ignore when you are viewing information with no intent to change.
- Many CA Gen Toolset windows have an option to Accept or Ignore changes. An Accept implies that a change has been made, and the content is not always examined. Accept can cause extra data to be transmitted during the upload and can cause changes to be flagged for Intelligent Regeneration and Version Control when no true changes occurred.
- Like Ignore and Accept on the workstation, End and Cancel terminate processing with different meanings. End exits the Intelligent Regeneration session without saving the temporary work for the session. Cancel dismisses the current screen and takes you back to the previous screen. In some cases, this allows you to make more selections and retain the use of temporary work.
- Regenerate frequently to minimize the work in a session.
- The Coordinate Intelligent Regeneration sessions with other encyclopedia tasks to minimize the contention. Intelligent Regeneration, like other Construction options, updates the model and can lock out other access.
- If multiple components require generation simultaneously, it is desirable to separate generation and installation, especially with batch processing. If one component fails (the library is out of space, linkage error, and so on), the entire process stops. This is especially true for batch processing.
- In some environments, the amount of output restrict batch processing. This require that you modify the batch JCL created by CA Gen, to create multiple jobs rather than one before submitting.
- If you know when the last Regeneration was done, use the timeframe to restrict the scope of information that CA Gen must analyze.
- Use Impact Analysis reports to coordinate implementation of changes that could affect multiple developers or projects.
- Package new Procedures and Procedure Steps before attempting to regenerate. Intelligent Regeneration only examines packaged load modules to find application components to process.

- Remove unused load modules from the packaging. Unused load modules can skew the base date and cause unneeded processing.
- Check your Construction libraries to assure that the space allocations are sufficient.
- Try to accomplish several objectives in a single Intelligent Regeneration session.
- When you initially request Intelligent Regeneration processing, CA Gen flags all changed modules and builds work files. The longer changed components are left for regeneration, the more computing power is used building these flags. When you swap out of Intelligent Regeneration using the End key, the flags and other temporary work are deleted and must be rebuilt again when needed. Use Cancel to leave the screens without erasing the flags.
- If an attribute is changed, select the attribute as an aggregate object for the Impact Analysis and Preview Reports to find all action blocks that contain the attribute in a view. However, the Impact Analysis and Preview Reports does not find action blocks where the attribute is only a host variable and the attribute is not explicitly included in a view. To find action blocks that, do not include the attribute in a view but include the attribute as a host variable, select the Data Column as an aggregate object.

Chapter 18: Debug and Trace Facilities

Capturing Information

There are several different facilities and methods that are used to capture information about the processing of a CA Gen component. These facilities include:

- Diagnostic log files that receive diagnostic and status messages
- Messages that display on screen
- The CLIST DEBUG parameter that can be used to set various debugging and tracing options for a single CLIST
- The TIUDEBUG facility that can be used to set various debugging and tracing options for all CLISTs
- The Event Trace Facility (ETF) that captures real-time event data from key areas of the generator
- The auxiliary trace file that captures binary data and the CA Gen ETFFMTX utility that formats it

Diagnostic Log Files

When a CA Gen component such as a generator is started, certain files are allocated to receive diagnostic and status messages. These files are named using a standard naming convention, which utilizes the data set prefix and suffix that are defined during CEINSTAL and stored in the TIUGLOB CLIST. For more information, see the *z/OS Installation Guide*.

Two files are of interest in debugging a CA Gen system or application, LOGFILE and SYSPRINT. These files are named:

- tiupref.IEF.LOGFILE.tiusufx
- tiupref.IEF.SYSPRINT.tiusufx

For example, if you specified that the tiupref variable contains the value TSOUSER and left the tiusufx variable undefined or null, then the resulting data set names are TSOUSER.IEF.LOGFILE and TSOUSER.IEF.SYSPRINT.

The IEF.LOGFILE is used to record status messages during the process, while the IEF.SYSPRINT file is used to contain error messages, system dumps, and so forth. This vary slightly from one process to another. However, both files are allocated and are significant.

Note: When the CA Gen processes are started, the LOGFILE and SYSPRINT files are allocated to redirect their standard output and error messages. Once the process ends and another starts, the diagnostic data in this file for the previous process is *overwritten*.

On-Screen Messages

Some diagnostics are provided by the operating environment that is not produced by or even known to the CA Gen system. These diagnostics are equally important in understanding and resolving a problem. Often these diagnostics are provided as a message or messages that are written to the terminal. In many cases, this is the only place that the messages are displayed, and they must be captured by printing the screen or writing the messages down. To obtain these diagnostics, your TSO profile specify WTPMSG, explicitly or by default.

When you call Technical Support, have the contents of the IEF.SYSPRINT and IEF.LOGFILE files, and messages that are written to your terminal ready. To assist you, the Technical Support technician ask you to enter more commands, rerun certain parts of your procedure, or to obtain more data. The facilities the support technician request are documented in the following sections.

CLIST DEBUG Parameter

Many of the host services are called from or directed by command lists (CLISTs). Some CLISTs contain an option that is called DEBUG that can be used to set various debugging and tracing options.

The disadvantage of using the CLIST DEBUG parameter is that it applies to only one CLIST and is not available for certain areas of the product. The TIUDEBUG utility, which is described previously, performs the same functions as the DEBUG parameter but operates over all CLISTs.

The DEBUG( ) parameter accepts a single value, which is a string of single characters that define certain types of debugging activities to be performed. The string is composed of the following character values that are concatenated together:

C

Specify to display on your terminal the execution of commands for the CLIST.

S

Specify to display on your terminal the SQL statement that was being executed when an SQL error occurred.

P

Specify to trace the flow of control from one program procedure to another.

D

Specify to display certain key data variables in addition to the Procedure Trace. If used, the Procedure Trace is also specified.

Examples of the use of the DEBUG parameter string are:

```
EX install.pds.clist(iefup) DEBUG(CSDP)
EX install.pds.clist(iefup) DEBUG(PD)
```

TIUDEBUG Facility

While the DEBUG parameter on each CLIST provides a method of tracing the flow and operation of individual CLISTs, it is cumbersome if many are traced. An alternative is provided by the TIUDEBUG facility. TIUDEBUG effectively sets the debug option as previously for all CLISTS without the manual execution of each CLIST with a DEBUG parameter.

You can set up TIUDEBUG by selecting options on a screen or entering arguments on the TSO command line.

Select Screen Options

Execute the TIUDEBUG CLIST by typing TSO %TIUDEBUG on the command line of a CA Gen panel.

The Set Debug Options Screen includes the following options:

CLIST Trace

Specifies to display on your terminal the execution of commands for all CLISTS.

SQL Trace

Specifies to display on your terminal the SQL statement that was being executed when an SQL error occurred.

Procedure Trace

Specifies to trace the flow of control from one program procedure to another.

DPUT Trace

Specifies to display certain key data variables in addition to the Procedure Trace. If used, the Procedure Trace is also specified.

STAT Trace

Specifies if you want a report on the CPU, memory, and I/O service units that are consumed in the processing of the CA Gen component. Use this data to determine execution costs, resource consumption, and so on, and for performance tuning.

DSN Trace

Uses this option only when directed to do so by a Technical Support technician.

Abend option – None

Specifies to suppress a memory dump CA Gen abend. This also frees currently allocated dump data sets.

Abend option – SYSUDUMP

Specifies to produce a memory dump of the current task in the address space CA Gen abend. Only the problem program data is dumped.

Abend option – SYSABEND

Specifies to produce a *formatted* memory dump of all system information that is associated with the current address space CA Gen abend. This argument is slower and more costly than the SYSMDUMP argument.

Abend option – SYSMDUMP

Specifies to produce an *unformatted* memory dump of all system information that is associated with the current address space CA Gen abend. This argument is faster and less costly than the SYSABEND argument.

The Abend options allow you to specify if a memory dump is taken CA Gen abend, and if taken, to control the type of dump produced:

- The None option suppresses a memory dump. This is the default.
- A SYSUDUMP typically produces the least amount of output but limits the output to the current task in the address space, and only to problem program data. In some cases, this will not contain enough information to solve a problem.
- The SYSABEND option is not limited to problem program data, and dump all system information that is associated with the current address space. The memory areas are formatted and printed to a *report* using the dump formatter. This can take quite a long time, especially for large regions with many activities. SYSABEND formats the output file.
- The SYSABEND dump is needed only on the rare occasions when SYSUDUMP does not reveal the underlying cause.
- The SYSMDUMP option also dumps all system areas and problem program areas.
- As with SYSABEND, the memory areas are formatted and printed to a report using the dump formatter, producing a great deal of output. However, SYSMDUMP transfers the memory in pages to the output file without formatting them. The data must be formatted to be readable. Formatting can be accomplished by a utility within z/OS known as AMDPRDMP or print dump.
- Advantages of using SYSMDUMP are:
  - Unformatted SYSMDUMP data can be transmitted to Technical Support, where it can then be formatted for display. This reduces the requirements for disk space, communications time, and costs.
  - SYSMDUMP is fast and probably the least expensive of the dump options.

More information:

Selecting an Options Screen (see page 333)

Enter Arguments on the Command Line

TIUDEBUG allows you to enter arguments to the CLIST through TSO. When this format is used, no screen displays. Enter the arguments for the same options as shown on the screen, using the following convention:

C

Specifies to display on your terminal the execution of commands for all CLISTS.

S

Specifies to display on your terminal the SQL statement that was being executed when an SQL error occurred.

P

Specifies to trace the flow of control from one program procedure to another.

D

Specifies to display certain key data variables in addition to the Procedure Trace. If used, the Procedure Trace is also specified.

R

Specifies if you want statistics on system resource consumption, costs, and other performance- related data.

RESET

Specifies to reset all debugging options to their initial (default) state, which is usually OFF.

ABEND(RESET)

Specifies to suppress a memory dump CA Gen abend. This also frees currently allocated dump data sets.

ABEND(SYSUDUMP)

Specifies to produce a memory dump of the current task in the address space CA Gen abend. Only the problem program data is dumped.

ABEND(SYSABEND)

Specifies to produce a formatted memory dump of all system information that is associated with the current address space CA Gen abend. This argument is slower and more costly than the SYSMDUMP argument.

ABEND(SYSMDUMP)

Specifies to produce an unformatted memory dump of all system information that is associated with the current address space CA Gen abend. This argument is faster and less costly than the SYSABEND argument.

DSNTRACE()

Uses this argument only when directed to do so by a Technical Support technician.

Example of Command using Arguments

```
TSO %TIUDEBUG C S D P ABEND(RESET)
```

More information:

Select Screen Options (see page 330)

Event Trace Facility (ETF)

The Event Trace Facility (ETF) provides a means of gathering real-time event data from key areas of the generator. Events that can be traced include function and method entry and exit, acquisition, and release of memory resources, and file access, and other types of events.

The ETF trace can be disabled, enabled in a memory-resident mode only, or written to a disk file. When disabled, the trace does not impose noticeable overhead or performance costs on the generators. When enabled in memory mode, the resulting overhead is slight.

The memory-resident, or primary trace, mode of the ETF provides a fixed-length trace table, which is overwritten in such a way that the most recent event records are retained. This table is allocated in main, or primary, memory of the process being traced. For example, if the table is defined as containing 50 events, then only the most recent 50 are retained. If after 150 events the table is formatted, the last 50 are displayed.

If the ETF has been enabled, the primary mode table is automatically formatted whenever an ABEND condition is detected. This report is automatically written to the IEF.LOGFILE file, described previously. Technical Support request this information to identify what particular circumstances caused the failure and where in CA Gen the component that the failure occurred.

In addition, the ETF can write trace records to a disk file. This allows all events that are traced to be recorded for later analysis. This mode of operation is called an *auxiliary trace* because the media used to store the trace data is auxiliary storage (disk or tape). The data format is written in a binary, machine-readable format to reduce overhead. Formatting must be performed using the ETFFMTX utility program.

Note: For a moderate size screen, dialog manager, or action block, the data that is produced in an auxiliary trace are many megabytes, typically from 25 through 35 MB.

You can set up ETFTRACE by selecting options on a screen or entering arguments on the TSO command line.

Selecting an Options Screen

Execute the ETFTRACE CLIST by typing TSO %ETFTRACE on the command line on CA Gen panels.

Use the following fields on the Set CA Gen Event Trace Facility Environment Variable Screen to specify various options to control the execution of the trace facility:

Activate Trace

Specifies Y to turn on the trace or N (the default) to turn it off.

Size of Trace Table

Specifies the size, in number of events that can be recorded, of the primary trace table, or the buffer size for auxiliary tracing.

The value must be a positive integer, zero, or from 32 through 99,999 inclusive. If the value is zero, the ETF facility is disabled. A non-zero value causes the trace facility to allocate a table large enough to contain the specified number of events.

A non-zero value less than 32 is rounded up to 32. If a non-zero value results in a table that exceeds a segment of memory on the executing platform, the number is adjusted down to the largest number that does not exceed a segment (on Windows or MS/DOS a segment is 64K; on z/OS, 1M).

Auxiliary Trace

Specifies Y to create an auxiliary trace or N (the default) to deactivate this option.

ETFINCL (see note)

Specifies the events to be recorded to the ETF. By default, all events are recorded. The actual operation of this option depends on the value for ETFEXCL.

ETFEXCL (see note)

Specifies the events that are NOT to be written to the trace table, or the auxiliary trace file, depending upon the conditions. By default, no events are excluded. The actual operation of this option depends on the value for the ETFINCL option.

ETFOPTS (see note)

Used to specify keyword runtime options to control execution of the ETF. DO NOT specify options unless directed to do so by a Technical Support technician. The technician indicate which options to set to avoid adverse effects on performance, storage requirements, and costs.

Separate the keywords with commas. You also prefix keywords with NO to negate their effect. The supported keywords are:

- SIGNAL (default) and NOSIGNAL
- EXIT and NOEXIT (default)
- GEN and NOGEN (default)
- SYMBOL and NOSYMBOL (default)
- FLUSH and NOFLUSH (default)

Note: For these options to be activated, the Activate Trace field must be set to Y, and the Size of Trace Table field must contain a value for an auxiliary trace buffer area.

When the ETFINCL, ETFEXCL, or both options are defined, their actual behavior depends on which ones are specified:

- When both are undefined, the default is to include all defined events.
- When only the include variable is defined, only the events that are listed on the include variable are registered. All others are implicitly excluded.
- When only the exclude variable is defined, all events are implicitly registered except those listed on the exclude variable.
- When both are defined, the intersection of the two sets that are defined by these variables is used to determine which events are registered. All others are implicitly excluded.

The syntax of the ETFINCL and ETFEXCL values is the same, and allows for a single event, list of events, a range of events, or a combination of lists and ranges:

- When a single event is specified, only the event ID need be entered (and no parentheses are required). For example, ETFEXCL=3.
- When a list of discrete IDs are being entered, separated them with commas and enclosed in parentheses. For example, ETFEXCL=(1,3,5).
- A range of IDs is specified by using a dash or hyphen, indicating a from-to ordering. For example, ETFEXCL=(1-5) is valid, representing event IDs 1 through 5, inclusive.
- Lists and ranges can be combined, where each range occupies a member of the list. For example, ETFEXCL=(1,3-5,7,9-13,15).

Note: The ID numbers that are provided in the preceding description are merely examples. Event IDs and their description are not documented in this guide. The events, what they represent, and the data in them are considered to be proprietary and confidential. A Technical Support technician inform you of the event IDs that included or excluded, if any.

Entering Arguments on the Command Line

ETFTRACE also allows you to enter arguments to the CLIST through TSO. When this format is used, no screen is displayed. You can enter the following arguments:

ON

Specifies to activate the ETF. When you omit the AUX parameter, the trace is activated in primary mode. The default is OFF.

OFF

Specifies to deactivate the ETF, the default.

AUX

Specifies to enable tracing and place it in auxiliary trace mode. This causes the DDNAME ETFAUXF to be allocated to a disk file, named *tiupref*.IEF.AUXTRACE.*tiusufx*. All registered events are written to this file for subsequent formatting. This parameter implies the ON parameter. If not specified, the default is to disable auxiliary tracing.

ETFINCL()

Specifies the events to be included in the trace data. The description and syntax of this option are the same as for entering it on screen.

ETFEXCL()

Specifies the events to be excluded in the trace data. The description and syntax of this option are the same as for entering it on screen.

ETFOPTS()

Specifies an option string to control the execution behavior of the trace facility. The description and syntax of this option are the same as for entering it on screen.

ETFTBLSZ()

Specifies the table size, in number of event entries. When omitted, and tracing is active, the default is 1024.

RESET

Resets all parameters to their default (OFF) state.

Example

```
TSO %ETFTRACE AUX ETFTBLSZ(128) ETFINCL('(1-5,7,9)')
```

Formatting the Auxiliary Trace File

When the auxiliary trace is activated, it writes binary data to the auxiliary trace file. This provides the fastest way to collect the data without impacting the application being traced.

The data is in a raw format, which is unreadable. The data must be formatted by the CA Gen ETFFMTX utility to convert the data to a usable, readable format.

A CLIST is provided to access the ETFFMTX utility. No panel interface is provided. The CLIST accepts two arguments that are used to define the auxiliary trace file and formatter parameters:

AUXFILE()

The auxiliary trace data is written to a file named *tiupref*.IEF.AUXTRACE.*tiusufx*, that can be formatted for diagnosis.

If you execute more than one auxiliary trace, rename the file to save it. The formatter can accept input from a file other than the standard file name. If the file is renamed, then the file name can be passed to the formatter through this argument. Otherwise, the formatter assumes the standard name.

When the auxiliary trace file was created from a background execution of the generator, the data set name format changes to:
tiupref.BAT.D*jhhmmss*.IEF.AUXTRACE.*tiusufx*

j is the last digit of the Julian day number

hhmmss is the time that the file was created, in 2400-hour format

If the background execution, the data set name using this operand MUST be passed to the formatter.

CMDLINE()

The ETFFMTX utility accepts a wide range of arguments that are used to control the selection of events from the auxiliary trace file. These arguments can define selection filters, limits, and other mechanisms to limit the output to analyze a specific problem. The CMDLINE parameter collects these arguments as a string and passes them to the ETFFMTX utility.

Chapter 19: CBLTERM Field Definitions

Fields of the CBLTERM Copy Member

Describes the fields of the CBLTERM copy member included in the online, TIRTERMA, and batch, TIRTERMB, termination exits.

TERM-STATUS-CODE

When TIRFAIL calls the TIRTERM(A/B), TERM-STATUS-CODE controls what TIRFAIL does next using the following values:

' ' (space) or 0 (zero)

TIRFAIL displays the message and redisplays the previous screen with TERM-DEFAULT-MSG in the error message field.

1

Indicates that TIRTIRM(A/B) handled the messages and not display them. It redisplay the previous screen with TERM-DEFAULT-MSG in the error message field.

2

Indicates that TIRTIRM(A/B) has handled everything. TIRFAIL does not display the messages and does not redisplay the previous screen.

TERM-FAIL-TYPE

The following table contains a description of TERM-FAIL-TYPE errors:

Error	Value	Description
TERM-FAIL-DB2	P	A DB2 error occurred while accessing the RPROF (profile) table.
TERM-FAIL-IEC	I	A CA Gen error occurred in the Dialog Manager.
TERM-FAIL-EXEC	E	A database error occurred in an action block or procedure.
TERM-FAIL-DIALOG	D	A non-database error occurred in the Dialog Manager.
TERM-FAIL-TSQ	Q	An error occurred while accessing the CICS temporary storage queue profile table.

Remaining Fields

The list of remaining CBLTERM fields is as follows:

TERM-ERROR-ACTION-NAME

Contains the name of the action block.

TERM-DEFAULT-MSG

This is an output field that by default contains the following message:

```
TIRM000E: SYSTEM ERROR OCCURRED - CONTACT SUPPORT
```

The message can be changed in the termination exit to anything meaningful to the user. For online procedures with a screen, the message is visible in the error message field when the screen is redisplayed.

TERM-SYSTEM-PRINTER

TERM-SYSTEM-PRINTER is valued with the printer TERMID if the action block executed a PRINTER TERMINAL IS statement.

TERM-ERROR-ENCOUNTERED-SW

Indicates the message:

```
TIRM037E: ** A FATAL ERROR HAS BEEN ENCOUNTERED **
```

TERM-VIEW-OVERFLOW-SW

Indicates the message:

```
TIRM037E: ** FATAL VIEW OVERFLOW HAS BEEN ENCOUNTERED **
```

TERM-ACTION-ID

Is appended to the message:

```
TIRM034E: LAST OR CURRENT DATABASE STATEMENT = ...
```

TERM-ATTRIBUTE-ID

Is appended to the message:

```
TIRM040E: PERMITTED VALUES MISMATCH, FIELD = F ...
```

TERM-STATUS-FLAG

Produces the message:

```
TIRM038E: ** FATAL DATABASE ERROR HAS BEEN ENCOUNTERED **
```

TERM-LAST-STATUS

Is appended to the message:

```
TIRM039E: DB LAST STATUS = ...
```

TERM-TRACE-PTR

This field is documented in online help under the error message TIRM039E.

TERM-LAST-STATEMENT-NUM

Is appended to the message:

```
TIRM035E: CURRENT STATEMENT BEING PROCESSED = ...
```

TERM-CURR-AB-ID

Is appended to the message:

```
TIRM032E: LAST OR CURRENT ACTION BLOCK ID = ...
```

TERM-CURR-AB-NAME

Is appended to the message:

```
TIRM033E: LAST OR CURRENT ACTION BLOCK NAME = ...
```

TERM-EABPCB-CNT, TERM-EABPCB-ENTRY,TERM-EABPCB-PTR

These fields describe PCB pointers. The first is the IO-PCB, the second is the ALTERNATE-IO-PCB; the last is a database pointer.

TERM-SQLCA-PTR

The following is a pointer to the SQLCA. The address fields of the SQLCA, first define it in linkage. Use the following example:

```
MY-SQLCA
FILLER
MY-SQL-CODE
FILLER
```

Add a SET statement at the beginning of the procedure division:

```
SET ADDRESS OF MY-SQLCA TO TERM-SQLCA-PTR
```

TERM-IEF-COMMAND

The special field of COMMAND.

TERM-IEF-TRANCODE

The special field of TRANCODE.

TERM-EXIT-STATE

The exit state number.

TERM-EXIT-INFOMSG

The exit state message.

TERM-USER-ID

The special field of USERID.

TERM-TRMINAL-ID

The special field of TERMID.

TERM-PRINTER-ID

Represents the ID of the system printer.

TERM-DIALOG-MESSAGE-NUM

The message number is the FAIL-MSG-NO set be the Dialog Manager. See the *Message Guide* for the message that is represented by the error code displayed.

TERM-OUTPUT-MESSAGE

Before calling the termination exit, TIRFAIL, prepares a table of messages that it display on return from the exit if the TERM-STATUS-CODE is a space or a zero. These messages are available to the exit. The last line with a message is followed by a line of all spaces.

TERM-DIALECT-NAME

The current dialect

TERM-FAILURE-MESSAGE-TEXT

The text of the failure message. This is moved to TERM-DEFAULT-MSG if you want it displayed on the application screen instead of the message:

```
TIRM000E: SYSTEM ERROR OCCURRED - CONTACT SUPPORT
```

Chapter 20: Retrieve DB2 SQLCA Data

Action Diagramming Statements

CA Gen includes two action diagramming statements, WHEN DATABASE DEADLOCK OR TIMEOUT and WHEN DATABASE ERROR, that allows your application to continue processing when a deadlock, timeout, or other database exception occurs. If your application uses these database exception handling statements, it must provide its own logic to roll back previous updates and to ensure data integrity if processing continues.

Important! The WHEN DATABASE DEADLOCK OR TIMEOUT statement is inoperative for applications running under IMS because IMS automatically cancels and rolls back transactions that encounter this type of error, without returning control to the application program.

External Action Blocks

CA Gen also provides the source code for an External Action Block (EAB) that allows applications to retrieve the SQLCODE and other data from the SQLCA after one of these WHEN DATABASE statements executes. This source code is located in the same CA Gen CEHBSAMP used for user exits. However, because it functions as an External Action Block, two more steps are required to install it:

- Define and package the External Action Block in your application model.
- The code must be precompiled to ensure that the correct SQLCA definition is included.

Use External Action Blocks

Follow these steps:

1. Open your model using the CA Gen Toolset.
2. Select Design, then Work Set List.
3. Add a work set named DB2_MVS_OR_XDB_SQLCA. The CA Gen toolset and generators consider the XDB product equivalent to DB2 for z/OS. CA Gen includes the equivalent EAB for XDB on the platforms it supports.

4. Add the following attributes to this work set, in this order:

   SQLCODE

   8 byte numeric

   MESSAGE_LENGTH

   8 byte numeric

   ERROR_MSG

   Mixed case text of varying length.

   Limits: 256 bytes

   SQLWARN

   11 bytes of text, fixed length

   SQLSTATE

   8 bytes of text, fixed length

   DBMS_TYPE

   8 bytes of case-sensitive text, fixed length

5. Select Design, then Action Diagram.
6. Create a action block that is named GET_DB2_MVS_OR_XDB_SQLCA with the default action block property settings, to ensure the High Performance View Passing, and Generate Missing Flags in Code options are enabled.
7. Add an export view to this action block containing the DB2_MVS_OR_XDB_SQLCA work set, and its attributes, in the same order.
8. Add EXTERNAL as the only statement in the action block. You also add notes.
9. Open each existing action diagram that retrieve the SQLCA data.
10. Add a USE GET_DB2_MVS_OR_XDB_SQLCA statement after the appropriate the WHEN DATABASE statements, and match a local or export view to the EAB's DB2_MVS_OR_XDB_SQLCA export view.

    The USE statement does not have to be subordinate to the WHEN DATABASE statement, but the EAB fails to work properly unless a WHEN DATABASE statement has actually executed first.

11. Select Construction, then Packaging. Open a construction environment (Online, Batch, or Cooperative), that contains an action diagram that USES the new EAB.
12. Expand a load module that USES the new EAB.
13. Select GET_DB2_MVS_OR_XDB_SQLCA, Detail, Properties.
14. Specify GETXDBSQ as the source member name.
15. Save your model.

16. You do not need to generate the External Action Block stub.
17. Copy source member GETXDBSQ from your CA Gen CEHBSAMP to a separate library.
18. Precompile, compile, and link edit GETXDBSQ. Use the compile and link options as defined in the chapter External Action Blocks.
19. On the Host Encyclopedia, ensure the load library name that contains your load module is in your External Action Block Load Library or External System Load Library specification. For more information, see the chapter Target Environment and Construction Libraries.
20. On the Host Encyclopedia, expand a load module that uses GET_DB2_MVS_OR_XDB_SQLCA. Find this EAB in the component list, type X next to it, and press Enter.
21. In the first DBRM_NAME field, type GETXDBSQ, and press F6 to save your model.
22. Regenerate and reinstall load modules that you modified to USE this EAB.

More information:

Specify Target Environment for HE (see page 77)

Modify External Action Block Sample Code

You modify the EAB sample code to exclude SQLCA data that you do not need, to return more data, or to extend the length of the error message. If you modify the EAB sample code, also modify the view definitions in your model to agree with code changes. If you rename the EAB source member, also rename the packaging source name and the external DBRM.

Do not call this EAB except after executing a WHEN DATABASE statement. The EAB uses the SAVE-SQLCA area in GLOBDATA to obtain the SQLCA data it returns. The generated application only saves the SQLCA in this area under the following conditions:

- A WHEN DATABASE statement executes and the application continues processing.
- An exception occurs after a database call, without a WHEN statement for that exception. Without a WHEN statement for the exception, the application terminates with a fatal error before calling the EAB.

The SQLCA is not saved when a logical exception statement executes, or when a database call succeeds. Therefore, you cannot use GET_DB2_MVS_OR_XDB_SQLCA to retrieve SQLCA data that is associated with WHEN SUCCESSFUL, WHEN NOT FOUND, WHEN ALREADY EXISTS, WHEN NOT UNIQUE, or WHEN PERMITTED VALUE VIOLATION statements.

However, after a WHEN DATABASE statement executes, the saved SQLCA for that statement is available, unless replaced by the SQLCA for a subsequent WHEN DATABASE statement. You can use this EAB to return the SQLCA for the last database exception, regardless of how many intervening database calls were made since that WHEN DATABASE statement executed.

Index

A

B

C

D

E

F

G

H

I

J

L

M

O

P

R

S

T

U

V

W