

# CA InterTest™ and CA SymDump® - 11.0 Configuring

Date: 06-Jun-2018





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

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2018 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Table of Contents

---

<b>Configuration Best Practices .....</b>	<b>9</b>
Symbolic Files .....	9
Dynamic Symbolic Support for CA Endeavor SCM .....	10
Starting and Stopping CA InterTest for CICS in a CICS Region .....	11
Performance Considerations for CICS Regions Using DB2 .....	11
CA SymDump Batch Installation with Other SVC 51 Intercepts .....	12
CA SymDump Batch PRTLIB Repositories Management .....	13
CA SymDump for CICS PROTDMP Files .....	13
<b>Configure Your Product With CA CSM .....</b>	<b>15</b>
Configure the Deployed Product .....	15
CA InterTest Batch Tasks .....	16
Activate Batch Link .....	17
CA SymDump Batch Tasks .....	17
Allow Concurrent Updates to Symbolic Files .....	18
Customize for Dynamic Symbolic Support .....	18
Add Support for CA File Master Plus .....	19
Review Language Environment (LE) Considerations .....	19
Add Runtime Load Library to LNKLST .....	19
APF-Authorize Runtime Load Library .....	20
Activate CA SymDump Batch .....	20
Establish Access to the TSO Facility .....	20
CA InterTest for CICS Tasks .....	21
CA SymDump for CICS Tasks .....	23
<b>Configure Your Product Without CA CSM .....</b>	<b>26</b>
Configuration Checklists .....	26
Configuring CA InterTest Batch .....	26
Configuring CA SymDump Batch .....	27
Configuring CA InterTest for CICS .....	28
Configuring CA SymDump for CICS .....	30
Options .....	31
Global Options .....	31

CA SymDump Batch Options .....	33
Execution-Time Options CAOUDFRX .....	33
Abend Handler and Reporter CAOETABL .....	37
Define the User Abend Control Table CAOCAB80 .....	43
Define the Abend Filter Table CAVHFMOD .....	50
Facilitate Multiple Default Execution Options .....	51
CA InterTest Batch Options .....	52
Configure ISPF Options .....	52
Configure Batch Link and Sysplex .....	54
CICS Options .....	55
Configure IN25OPTS Options .....	55
Refresh IN25OPTS Options .....	56
IN25OPTS Keywords .....	56
CA InterTest for CICS Options .....	81
Configure Abend Codes .....	81
Customize Help Abend Codes .....	82
CA SymDump for CICS Options .....	82
Procedure for Modifying Dump Capture Criteria .....	83
IN25CAPT Keywords .....	84
How Macro IN25CAPT Controls Dump Capture .....	85
External Security Option .....	85
Resource Protection Levels .....	85
External Security Using CA Top Secret .....	87
External Security Using CA ACF2 .....	88
External Security Using IBM RACF .....	89
User Exits .....	90
CA InterTest Batch User Exits .....	90
Define Excluded Programs .....	90
Define the Batch Link Selection User Exit .....	90
Define the Batch Link Scheduling User Exit .....	91
CA SymDump Batch User Exits .....	91
Control Exclusion Criteria .....	91
Control the Destination of Reports .....	92
Dynamic Allocation for the CAIPRINT DD .....	95
Control Access to Reports in the Repository .....	98
Management Reporting System (MRS) User Exits .....	100
CA InterTest for CICS User Exits .....	104
Unprotect Main Storage with User Exit IN25LETX .....	104
Password Protection .....	107
CA SymDump for CICS User Exits .....	113
Global User Exit XDUREQ .....	113
Configure SymDump for CICS Security .....	113

External Security Installation Option .....	114
Resource Protection Levels .....	115
External Security Using CA Top Secret .....	115
External Security Facility .....	117
Security Exit Facility .....	120
Allocate Files .....	122
Allocate PROTSYM .....	122
Concurrently Updating the PROTSYM File .....	123
Populate the Symbolic File .....	123
Update Your In-house Compile and Assemble Procedures .....	123
CA InterTest Batch Files .....	123
Allocate PROFLIB .....	123
Allocate ALIBs (Optional) .....	124
Allocate INCLIBs (Optional) .....	124
CA SymDump Batch .....	125
Allocate CAIPRINT Repository Library .....	125
CA InterTest for CICS and CA SymDump for CICS .....	126
Install the Help Facility .....	126
Create the User-Defined Abend Description File .....	126
CA InterTest for CICS .....	126
Create the Debug Session File .....	126
Create the Checkpoint File .....	126
CA SymDump for CICS .....	127
Create the PROTDMP File .....	127
Estimating the Size of the PROTDMP File .....	127
Interfaces and Compatibility .....	128
Configure Dynamic Symbolic Support .....	128
Configure the Started Task Procedure for CA Endeavor support .....	128
Shut Down CA Endeavor SCM Dynamic Symbolic Support Started Task .....	129
Configure Dynamic Symbolic Support for CA InterTest Batch .....	129
Configure Dynamic Symbolic Support for CA SymDump Batch .....	130
Configure Dynamic Symbolic Support for CA InterTest and SymDump for CICS .....	130
Batch Interfaces and Compatibility .....	130
CA InterTest Batch .....	131
CA SymDump Batch .....	143
CICS Interfaces and Compatibility .....	150
CA InterTest for CICS and CA SymDump for CICS .....	151
CA InterTest for CICS .....	154
CA SymDump for CICS .....	175
Activate Your Product .....	178
All Products .....	178
Authorization .....	178

CAIRIM .....	178
Query Activation Status .....	180
CA InterTest for CICS .....	180
Activate CA InterTest for CICS from a Program .....	180
Start CA InterTest for CICS from a PLT .....	181
Automatic Startup in CICSplex Regions .....	182
Activate CA InterTest for CICS .....	182
Deactivate CA InterTest for CICS .....	182
Restart CICS and Run VRPT Transaction .....	183
CA SymDump for CICS .....	184
Define and Schedule the Trace Format Region Job .....	184
Restart CICS and Run VRPT Transaction .....	184
Start and Stop Dump Capture Facility .....	185
Purge a Dump Data Set Automatically at Startup .....	185
Shut Down Automatically .....	185
Shut Down Trace Format Region .....	186
CA SymDump Batch .....	186
Deactivate CA SymDump Batch .....	186
Installation Verification .....	186
CA InterTest Batch .....	186
CA SymDump Batch .....	187
Review and Edit JCL Member CARXDEMO .....	187
Run the Demo and Examine the Results .....	187
Run the CA IDMS/DB, DB2, and IMS Demo Programs .....	187
Review Abend Reports Using the Viewer .....	188
CA InterTest for CICS .....	189
CA SymDump for CICS .....	191

# Configuring

---

This article contains information about the steps you need to take to configure CA InterTest and CA SymDump products. To configure your product, follow the instructions in either [Configure Your Product With CA CSM \(see page 15\)](#) or [Configure Your Product Without CA CSM \(see page 26\)](#).

# Configuration Best Practices

---

This article contains configuration best practices for your product.

- [Symbolic Files \(see page 9\)](#)
- [Dynamic Symbolic Support for CA Endeavor SCM \(see page 10\)](#)
- [Starting and Stopping CA InterTest for CICS in a CICS Region \(see page 11\)](#)
- [Performance Considerations for CICS Regions Using DB2 \(see page 11\)](#)
- [CA SymDump Batch Installation with Other SVC 51 Intercepts \(see page 12\)](#)
- [CA SymDump Batch PRTLIB Repositories Management \(see page 13\)](#)
- [CA SymDump for CICS PROTDMP Files \(see page 13\)](#)

## Symbolic Files

Before configuring your CA Technologies testing tool product, carefully consider how to best organize your symbolic information that is stored in your PROTSYM files. Well organized symbolic information allows the most effective and efficient access to your symbolics during testing.

There is no one correct way to organize symbolic information that works best for everyone. How you organize your symbolic information will always depend on your environments, your application systems, and the number of programs and programmers involved in your testing. Although there is no best organization that works for everyone, you should consider the following guidelines when planning for symbolic support:

- In a *production environment*, consider maintaining a separate PROTSYM file for each application system that can be shared by all of your programmers. Because only one copy of each program is typical in a production environment, the same symbolic information could be shared by any programmer debugging in that environment.  
In production environments, symbolic information is often limited to a single version of each program, and testing or debugging is typically less frequent and often limited to critical, real-time failures. Separate, shared PROTSYM files allow you to maintain a single set of symbolic information for each production application system that would be ready and available for access in the event of a failure.
- In a *test environment*, multiple versions of the same program are more likely to exist, and shared executable libraries are complimented by *sandbox* libraries that house test fixes or various works in progress. For this environment, a single shared symbolic repository is adequate for the shared executables. However, each programmer may want to own their own PROTSYM file, as it will provide them with the freedom to test using the correct symbolic information without impacting other members of the team.

### Business Value:

Well organized symbolic information lets your application programmers easily locate, access, and share symbolic information for their programs. As a result, you could see better performance for your CA Technologies testing tools, optimal use of your DASD space, and the best possible productivity for your application programmers.

**Additional Considerations:**

As a general rule, think of your PROTSYM files as nothing more than a match for your load libraries, remembering that for each unique version of a load module you may require symbolic information for testing or debugging. Like the load libraries that exist at each level or stage of your development environment, the corresponding symbolic files should vary in size depending on the maximum number of programs expected at each level, with the individual PROTSYM files being the smallest by far.

If you are also licensed for CA Endeavor Software Change Manager (CA Endeavor SCM), CA testing tools provide an additional level of automation through dynamic symbolic support. This feature lets you populate your PROTSYM file dynamically on demand with symbolic information for programs built using CA Endeavor SCM.

We recommend that you set the threshold for used space in the symbolic file to monitor the space utilization percentage. When the threshold is reached, a message is issued that indicates that the symbolic file is almost full. Set the threshold by specifying the USEDSPACEMSG parameter with the INITIALIZE or UPDATE functions of IN25UTIL.

**More Information:**

For more information about creating, populating, maintaining, and using your PROTSYM symbolic repositories to enhance your testing, see [Symbolic Support \(https://docops.ca.com/display/CAITSD11/Symbolic+Support\)](https://docops.ca.com/display/CAITSD11/Symbolic+Support), which is provided with each CA Technologies testing tool product.

For more information about dynamic symbolic support, see [Installing \(https://docops.ca.com/display/CAITSD11/Installing\)](https://docops.ca.com/display/CAITSD11/Installing).

## Dynamic Symbolic Support for CA Endeavor SCM

If you are licensed for CA Endeavor Software Change Manager (CA Endeavor SCM), CA Technologies recommends using the dynamic symbolic support feature provided by the CA Technologies testing tools. This feature allows you to populate your PROTSYM file dynamically on demand with symbolic information for programs built using CA Endeavor SCM.

Because there are many ways to maintain the content of your PROTSYM files, you may encounter the scenario where you have multiple versions of the same program for which you need to locate and select the correct symbolic information. Using dynamic symbolic support gives you the option to create your symbolic information only when it is needed, rather than always reloading your PROTSYM file after each compilation or assembly. This can save you time, CPU utilization, and even DASD space.

**Business Value:**

The dynamic symbolic support feature is available with many of CA Technologies testing tool products including CA InterTest Batch, CA InterTest for CICS, CA SymDump Batch, and CA SymDump for CICS. This feature simplifies the setup requirements for using symbolic support, while simultaneously eliminating the majority of potential setup errors associated with it. By improving the accuracy of the content of your symbolic file and eliminating common setup errors, dynamic symbolic support can greatly increase the productivity of your programmers.

**Additional Considerations:**

Dynamic symbolic support requires that you compile or assemble your applications using a CA Endeavor SCM process with the footprint feature active. When symbolic information is required by one of the CA Technologies testing tools, your PROTSYM files are inspected first for the matching symbolic information. When no matching information is found, the executable is inspected for a CA Endeavor SCM footprint. If this footprint is found, that footprint is used to locate the matching listing that is dynamically loaded into your specified PROTSYM file. The matching symbolic information is then used by your test session.

For specific instructions related to setting up and using dynamic symbolic support, see [Installing and Batch Debugging](#).

## Starting and Stopping CA InterTest for CICS in a CICS Region

CA recommends that customers start CA InterTest for CICS during CICS PLT processing to ensure that the CICS global user exits required by CA InterTest are the first to receive control.

**Business Value:**

This practice ensures that other global user exits will not interfere with CA InterTest for CICS when the product is making a decision to monitor or debug a CICS program.

**More Information:**

For more information on starting CA InterTest for CICS, during CICS PLT processing, see [Activate Your Product \(see page 178\)](#).

## Performance Considerations for CICS Regions Using DB2

CA recommends that customers starting CA InterTest or SymDump for CICS in a region with DB2 activity should review the XRMIO option and set it according to their needs.

When the XRMIO option is set to YES (the default), CA InterTest and SymDump for CICS can provide additional diagnostic information about the application through the use of the CORE=LASTSQL command. This command causes CA InterTest and SymDump for CICS to display the last SQL statement executed by the application. However, if you set XRMIO to YES and the applications that run in your CICS region heavily use EXEC SQL calls, performance in that region can be degraded.

**Business Value:**

Although using the XRMIOOUT global user exit from IBM can affect performance in regions that use EXEC SQL calls extensively, this exit can provide valuable diagnostic information for DB2 applications that contain EXEC SQL statements. This information can be useful in identifying or resolving problems and can improve time to resolution.

**Additional Considerations:**

You must carefully weigh the value of the information you receive against the performance considerations of using XRMIOOUT in regions where heavy DB2 activity exists. Especially in production regions where performance is most critical, consider carefully whether the additional diagnostic information provided by XRMIOOUT is valuable enough to risk the possible performance impact of using this option where extensive DB2 activity exists.

## CA SymDump Batch Installation with Other SVC 51 Intercepts

If you have multiple products that use an intercept for SVC 51, run CAIRIM to initialize CA SymDump Batch after all other products.

**Business Value:**

Initializing CA SymDump Batch after all other products have been initialized helps ensure that CA SymDump Batch will get control first. This enables the most accurate possible analysis and reporting before any possible interference can occur.

**Additional Considerations:**

When CA SymDump Batch is initialized, it replaces the address of SVC 51, the DUMP SVC. When the product is finished with its reporting, it will take one of the following actions, depending on the SymDump option.

- If the SymDump option is OFF, control will be returned to the operating system.
- If the SymDump option is ON, control will be given to the address that was replaced when CA SymDump Batch was initialized.

Therefore, if any other product, such as CA SymDump System, that intercepts SVC 51 was initialized before CA SymDump Batch, that product will not get control unless the SymDump option is set to ON.

# CA SymDump Batch PRTLIB Repositories Management

When configuring CA SymDump Batch, you must set your default PRTLIB. The PRTLIB repository is the VSAM data set that maintains your abend reports. You can have one repository for multiple z/OS systems or one per system. However, care must be taken to determine if your installation requires the use of more than one PRTLIB.

## Business Value:

Well managed repositories let your application programmers easily locate, access, and share abend reports for their applications. You can use multiple repositories per system with the use of multiple default tables or by using a user exit. As a result, you get better performance from your CA testing tools, optimal use of your DASD space, and the best possible productivity for your application programmers.

## Additional Considerations:

When CA SymDump Batch is configured, the default PRTLIB data set is defined with the PRTLIB option of CAOETABL. However, not all installations may prefer writing all abend reports to one repository. Specific installations may want a few groups to write to a specific repository. To provide this flexibility, you can override the PRTLIB value in several ways.

- Use the existing user exit, CAOCUPRT to return the PRTLIB to use based on JOBNAME, STEPNAME, abending program name, and USERID.
- Define the PRTLIB by specifying a PRTLIB value in your CAIOPTS DD within the job step JCL.
- Allocate the CAIOPTS DD dynamically for each user by following the naming convention userid.CAIOPTS.

## More information:

For specific instructions on setting up the user exit CAOCUPRT, the PRTLIB option, or dynamic allocation of your CAIOPTS data set, see [Installing \(https://docops.ca.com/display/CAITSD11/Installing\)](https://docops.ca.com/display/CAITSD11/Installing) and [Batch Abend Analysis \(https://docops.ca.com/display/CAITSD11/Batch+Abend+Analysis\)](https://docops.ca.com/display/CAITSD11/Batch+Abend+Analysis).

## CA SymDump for CICS PROTDMP Files

You can view transaction dumps from multiple CICS regions using a single dump viewing CICS region. Because the CA SymDump for CICS dump viewer runs as a CICS transaction, it is possible to view dumps from multiple CICS regions from a single dump viewing region. To do this, add remote CICS file definitions for the dump data sets from all CICS regions.



**Note:** The CICS releases for all related dump data sets must match the dump viewing region. You cannot mix dump data sets from different CICS releases in the same dump viewing region.

**Business Value:**

By providing access to dumps from multiple CICS regions through a single viewing region, CA SymDump for CICS simplifies the process of locating dumps and ensures that dumps remain available even when the CICS region, in which those dumps were produced, becomes unavailable. This improves programmer productivity, reduces time to resolution, and helps to control or eliminate costly down time.

**More Information:**

For more information on creating PROTDMP files and estimating their size, see [Allocate Files \(see page 122\)](#). For more information on maintaining and reporting on the contents of your PROTDMP files, see [Batch Utility \(https://docops.ca.com/display/CAITSD11/Batch+Utility\)](https://docops.ca.com/display/CAITSD11/Batch+Utility).

# Configure Your Product With CA CSM

---

This article describes the manual tasks you perform when configuring your product using CA CSM.

- [Configure the Deployed Product \(see page 15\)](#)
- [CA InterTest Batch Tasks \(see page 16\)](#)
- [CA SymDump Batch Tasks \(see page 17\)](#)
- [CA InterTest for CICS Tasks \(see page 21\)](#)
- [CA SymDump for CICS Tasks \(see page 23\)](#)

Data sets are created by CA CSM after successfully completing each step, that is the SMP/E installation, deployment, and configuration procedures.



**Note:** CA Technologies strongly advises that you perform the initial installation and configuration of your product and its components in a test environment as a precaution. This testing will let you detect any possible conflicts with other vendor products.

## Configure the Deployed Product

Configuration is a process of copying the deployed libraries to run-time libraries and customizes the product for your site to bring it to an executable state. You can configure CA Technologies products that you have already acquired, installed, and deployed using CA CSM. You cannot use CA CSM to configure a product unless you have already used CA CSM to deploy the product.

You perform the following high-level tasks to configure your products using CA CSM:

1. Select a configurable deployment on the Deployments tab to view details and products for that deployment.
2. Select a product in the deployment and start the Configuration wizard to create a configuration. Complete each of the steps in the wizard. The wizard has multiple levels of detailed instructions and guides you through choosing configuration settings for your site. At any point, you can save your work and can come back to it later. Configurations where you have partially completed the steps in the wizard are listed on the Configurations tab. The steps in the wizard include the following:
  - a. Define a configuration name and select a system for the configuration.
  - b. Select configuration functions and options.
  - c. Define system preferences.
  - d. Create target settings.

- e. Select and edit resources.
3. Build the configuration. The last step of the Configuration wizard lets you build the configuration. If needed, you can edit the configuration and can build the configuration again. Building the configuration closes the wizard and creates a configuration with all your settings.
4. (Optional) Validate the configuration. Validation verifies access to resources that are going to be used when you implement the configuration.
5. Implement the configuration. You implement a configuration to make your deployed software fully functional. Implementation executes on the destination system, applying the variables, resources, and operations that are defined in the configuration. CA CSM configures the product.

After the configuration process completes, the product is ready for you to use.

## CA InterTest Batch Tasks

Configure the external tasks for CA InterTest Batch.

### Follow these steps:

1. APF authorize the application runtime load library. For more information, see [Activate Batch Link \(see page 17\)](#).
2. Allocate a profile data set for each user. Required, if the USERPROFLIBS configuration options is selected. If the USERPROFLIBS option is not selected, a single PROFLIB will be created automatically. For more information, see [Allocate Files \(see page 122\)](#).
3. Allow users to concurrently update PROTSYM files. Required, if the NEWPROTSYM option is selected. For more information, see [Allocate Files \(see page 122\)](#).
4. Allocate an ALIB for each user. Required, if both the ALIB and USERALIBS configuration options are selected or you will be using the Foreground debugging feature. For more information, see [Allocate Files \(see page 122\)](#).
5. Allocate an INCLUDE library for each user. Required, if both the INCLIB and USERINLIBS options are selected. For more information, see [Allocate Files \(see page 122\)](#).
6. (Optional) Add the product to the ISPF main menu. For more information, see [Batch Interfaces and Compatibility \(see page 130\)](#).
7. Add CA InterTest Batch to your TSO logon procedure. For more information, see [Batch Interfaces and Compatibility \(see page 130\)](#).
8. (Optional) Customize CA Roscoe for use with the application. For more information, see [Batch Interfaces and Compatibility \(see page 130\)](#).
9. Customize the application started task for CA Endeavor SCM Dynamic Symbolic Support. For more information, see [Interfaces and Compatibility \(see page 128\)](#).



**Note:** Required, if the MULTNDVRSITES option is selected, indicating that the customer is using multiple Endeavor site IDs.

10. (Optional) Update the OS/VS COBOL procedure, COBINT and compile, bind, and post-process the OS/VS COBOL foreground demonstration program, CAMRCOB. For more information, see [Batch Interfaces and Compatibility \(see page 130\)](#) .



**Note:** Required, if the COBOL configuration option is selected and the customer still compiles with the OS/VS COBOL compiler.

## Activate Batch Link

This operation updates your CAS9 procedure and PARMLIB then executes the CAS9 procedure to activate Batch Link.

Use the following procedure to activate the Batch Link feature of CA InterTest Batch.

### Follow these steps:

1. Add and APF-authorize the CA InterTest Batch runtime load libraries (CAI.CAVHAUTH and CAI.CAVHLOAD) to the STEPLIB concatenation in your CAS9 procedure.



**Note:** All customized configuration elements must be available through STEPLIB.

2. Add the following control card to the member or data set pointed to by the PARMLIB DD in your CAS9 procedure:

```
PRODUCT(CA INTERTEST/SYMDUMP) VERSION(VHA0) INIT(CAVHINIT) PARM(220)
```

3. Execute CAIRIM using your CAS9 procedure.  
You have activated Batch link.

MSM can execute this step automatically during the implementation process when you change the default value of Manual operation to Automatic operation before releasing it. Otherwise, specify Confirm manual operation when you implement the configuration.

## CA SymDump Batch Tasks

Follow these steps to configure the external tasks for CA SymDump Batch:

1. Allow Concurrent Updates to Symbolic Files.

2. Customize for Dynamic Symbolic Support.
3. Add Support for CA Filemaster.
4. Add Runtime Load Library to LNKLST.
5. APF-Authorize Runtime Load Library.
6. Activate CA SymDump Batch.
7. Establish Access to the TSO Facility.

## Allow Concurrent Updates to Symbolic Files

The MVS RESERVE and DEQ macros are used to allow sharing of a PROTSYM file between regions or systems. The resource major name used in the RESERVE and DEQ macros is INTERTST. Place the major name INTERTST into an MVS service that converts the RESERVE into a cross-system ENQ if this service is available at your site.

## Customize for Dynamic Symbolic Support

If you are an existing CA Endeavor SCM user and the FOOTPRINT feature is used when applications programs are generated in CA Endeavor SCM, you can customize CA Endeavor SCM for dynamic symbolic support by editing the started task procedure.



**Note:** This step is not required if you have only one C1DEFLT5.

### Follow these steps:

1. Use the CAI.CAVHJCL member INTNDVVR, the started task procedure, to access DSS to dynamically load program symbolic information.  
This started task must be an APF-authorized task.
2. Modify this procedure to conform to your installation standards.
3. Copy the procedure to a system procedure library.  
The member name of the procedure must have the CA Endeavor SCM C1DEFLT5 table site ID appended to it. For example, if the site ID is 3, the procedure member name should be INTNDVVR3.  
The first seven-character prefix of the procedure member name you copied to the system procedure library must match the value defined in the NDVRPROC option in CAOUDFRX.



**Note:** Create a procedure for each unique site ID.

## Add Support for CA File Master Plus

If you are an existing CA File Master Plus user and want to use the FM viewer command, you must perform the following additional set up. Before you can use FM viewer command you must perform the following procedure.

**Follow these steps:**

1. Set up the CA File Master Plus to use LIBDEFs (LIBALLOC='Y').
2. Verify that the FM1 executable exists in a data set allocated to SYSPROC or SYSEXEC.
3. Allocate CAI.CAVHCLS0 to SYSPROC or SYSEXEC.

## Review Language Environment (LE) Considerations

Use the following procedure to add abnormal termination exit entry for CAOECEEX.

**Follow these steps:**

1. Edit CEE.SCEESAMP(CEEEXTAN).
2. Follow the instructions in this member to add an abnormal termination exit entry for CAOECEEX.
3. Apply the applicable SYSMOD to make these changes take effect.



**Note:** For more information about the CEEEXTAN exit, see the *IBM z/OS Language Environment Customization* guide.

## Add Runtime Load Library to LNKLST

Add the CA SymDump Batch runtime load library, CAI.CAVHLOAD, to the LNKLST concatenation. This can be done dynamically by creating and activating a new LNKLST set. For example, a new LNKLST set called LNKLST01 can be created by copying the existing set, LNKLST00, and adding CAI.CAVHLOAD to the end of the concatenation.

Follow these steps:

1. SETPROG LNKLST,DEFINE,NAME=LNKLST01,COPYFROM=LNKLST00
2. SETPROG LNKLST,ADD,NAME=LNKLST01,DSNAME=CAI.CAVHLOAD,ATBOTTOM
3. SETPROG LNKLST,ACTIVATE,NAME=LNKLST01

The PROGxx or LNKLSTxx member of SYS1.PARMLIB should be updated to permanently add CAI.CAVHLOAD to the IPL LNKLST set at the desired location in the concatenation.

## APF-Authorize Runtime Load Library

To APF-authorize the CA SymDump Batch runtime load library, add an entry for CAI.CAVHAUTH to member IEAAPFxx of SYS1.PARMLIB.



**Note:** Authorization does not take effect until the next IPL.

## Activate CA SymDump Batch

**Follow these steps:**

1. Add the CA SymDump Batch load library CAI.CAVHAUTH to the STEPLIB concatenation in your CAS9 procedure (SYS1.PROCLIB(CAS9)).
2. Add the following control card to the member or data set pointed by the PARMLIB DD (in SYS1.PROCLIB(CAS9)):
 

```
PRODUCT(CA INTERTEST/SYMDUMP) VERSION(VHA0) INIT(CAVHINIT) PARM(220)
```
3. Execute CAIRIM using the CAS9 procedure.

## Establish Access to the TSO Facility

You must include the CA SymDump Batch ISPF libraries in your session ISPF files concatenation to access the CA SymDump Batch panels. Use the following procedure to concatenate the ISPF file and tailor the Master panel.

**Follow these steps:**

1. Concatenate the following data sets into the indicated ddnames (if they were not previously concatenated):

Data Set Name	DDName	Description
CAI.CAVHPNLO	ISPLLIB	ISPF panels for CA SymDump Batch
CAI.CAVHKL0	ISPLLIB	ISPF skeletons for CA SymDump Batch
CAI.CAVHMSG0	ISPLLIB	ISPF messages for CA SymDump Batch
CAI.CAVHTBLO	ISPTLIB	ISPF tables for CA SymDump Batch
CAI.CAVHCLS0	SYSPROC	ISPF CLISTs for CA SymDump Batch
CAI.CAVHLOAD	ISPLLIB or STEPLIB	CA SymDump Batch Load Library

2. Perform the following steps if your site uses variable-length CLISTs:

- a. Execute `CAI.CAVHJCL(CARXCNVT)` to change the CLISTs from fixed to variable length format.
  - b. Point the SYSPROC DD to the `CAI.VCAICLIB` data set (instead of `CAI.CAVHCLS0`).
3. Allocate the following data sets into the indicated ddname (if they were not previously allocated):

Data Set Name	DDName	Description
CAI.CAVHTBL0	CAOEXTBL	ISPF tables for CA SymDump Batch
CAI.CAVHHELP	CAOIHELP	ISPF Help library

You have now placed the CA SymDump Batch ISPF libraries into your session ISPF files concatenation.

To tailor the ISPF Master panel, do one of the following

- Copy the `ISP@MSTR` panel supplied with CA SymDump Batch (named `CARXSBA0`)
- Add new lines to your existing `ISP@MSTR` panel.

Use the following procedure to add new lines to your existing panel.

**Follow these steps:**

1. Add the first line after the last option selection displayed on the panel.
2. Define this line as an option number for the user to select with a brief description of the function. For example:
 

```
% 6 +CA SYMDUMP BATCH - Options Panels
% 7 +CA SYMDUMP BATCH - Invoke the CAIPRINT Repository Viewer
```
3. Add the lines containing the commands required to execute the options to the list of `&SEL` variables defined at the end of the panel. For example:
 

```
6. 'PANEL(CA0EP001) NEWAPPL(CA$Z) '
7, 'PGM(CA0IF000) NEWAPPL(CA0I) '
```

## CA InterTest for CICS Tasks

We recommend that you configure your product using CA CSM.

When you configure your product using CA CSM, perform the manual tasks described in following procedure to configure the external tasks. Data sets are created by CA CSM after successfully completing each step that is the SMP/E installation, deployment, and configuration procedures.

**Follow these steps:**

1. Allow concurrent updates to symbolic files.  
The MVS RESERVE and DEQ macros are used to allow sharing of a PROTSYM file between regions or systems. The resource major name used in the RESERVE and DEQ macros is INTERTST. Place the major name INTERTST into an MVS service that converts the RESERVE into a cross-system ENQ if this service is available at your site.
2. Customize for Dynamic Symbolic Support.
  - a. Edit member INTNDVDR of the CA Testing and Fault Management common runtime procedure library (CAI.CAVHPROC), modify it to conform to your installation standards, then save it.
  - b. To create a started task for a CA Endeavor SCM site ID, copy INTNDVDR to a system procedure library as member INTNDVDRn, where n is the CA Endeavor SCM C1DEFLT5 table site ID. Create a procedure for each unique CA Endeavor SCM site ID.



**Note:** The INTNDVDRn procedures require read security access to any load library data sets in which application programs that were created by CA Endeavor SCM reside and which may require symbolic support for debugging with CA InterTest for CICS. They also require write security access to any PROTSYM files they will populate.

3. Add CCI spawn parameters.  
This step adds the CCI spawn parameters for the CA InterTest for CICS dynamic symbolic support feature to the JCL procedure used to start CA ENF. If you want this step to be executed automatically by CA CSM, use the 'Action' button to change it from a 'Manual' to an 'Automatic' operation before releasing it at implementation time.  
To perform this step manually, add the following DD statement to the SPNPARMS DD in the JCL procedure used to start CA ENF:
 

```
// DD DISP=SHR,DSN=avhproclib(INDVSPWN)
```
4. Refresh CA ENF.  
Schedule a refresh of CA ENF to activate the CCI spawn parameters for CA InterTest for CICS dynamic symbolic support of CA Endeavor SCM. The dynamic symbolic support feature will not be available until CA ENF has been refreshed.
5. Update CICS startup JCL and restart CICS.



**Note:** If your site has previously installed CA SymDump for CICS as a separate product, some DD statements may already exist.

If you use DD statements in your CICS startup JCL, your CA InterTest for CICS files, as defined in the CSD, require DD statements in the JCL for CICS. DISP=SHR must be specified for PROTSYM to allow batch compiles to run concurrently with CICS. Alternatively, you may directly add the data set names to the CEDA FILE definition, defined in the CSDINT member in CAI.CAVHJCL and remove the DD statements for those files.

Add the following DD statements to your CICS startup JCL:

```
//PROTCPF DD DSN=CAI.PROTCPF,DISP=OLD
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//PROTHLF DD DSN=CAI.CAVHHLF,DISP=SHR
//PROTUHF DD DSN=CAI.PROTUHF,DISP=SHR
//PROTMLLOG DD DSN=CAI.PROTMLLOG,DISP=SHR
```

If you are using the CA InterTest for CICS dynamic symbolic support feature for CA Endeavor SCM, add an NDVRSYM DD statement to specify the PROTSYM file to be used for dynamic symbolic loading. The NDVRSYM DD must be specified in the CICS startup JCL for CA InterTest for CICS to invoke the dynamic symbolic load option. A sample NDVRSYM DD statement follows:

```
//NDVRSYM DD DSN=CAI.PROTSYM,DISP=SHR
```

If you specified more than one symbolic file in the SYMFIL variable of the IN25OPTS variable group, include a DD for each SYMFIL specified.

Add both CA InterTest for CICS load library and the symbolic common component load library to the DFHRPL in the CICS startup JCL:

```
// DD DSN=CAI.CAVHLOAD,DISP=SHR
```

If you execute the demo programs, add the following DD to the DFHRPL in your CICS startup JCL to specify the same load library data set name that you used when you ran the demo program's compile and link JCL:

```
// DD DSN=CAI.demoload,DISP=SHR
```



**Note:** CA InterTest for CICS makes calls to IEWBIND when using the online COMPOSITE module command and IEWBIND requires about 750 KB of below the line storage. To ensure there is sufficient storage available (below the line for IEWBIND calls), you must ensure that your DSALIM is at least 750 KB less than your below the line private area size. Failure to do this could result in S0F4 and U900 abends in your CICS region during dump capture.

## CA SymDump for CICS Tasks

This section describes the manual tasks that you perform when configuring CA SymDump for CICS (SYMC) using CA CSM.

**Follow these steps:**

## 1. (Optional) Add CCI Spawn Parameters.

This step is only required if the ENDEVOR option is selected. This step adds the CCI spawn parameters for the CA InterTest for CICS dynamic symbolic support feature to the JCL procedure used to start CA ENF. If you want this step to be executed automatically by CA CSM, use the Action button to change it from a 'Manual' to an 'Automatic' operation before releasing it at implementation time.

To perform the step manually, add the following DD statement to the SPNPARMS DD in the JCL procedure used to start CA ENF:

```
// DD DISP=SHR,DSN=CAI.CAVHPROC(INDVSPWN)
```

## 2. (Optional) Refresh CA ENF.

This step is only required if the ENDEVOR option is selected. Schedule a refresh of CA ENF to activate the CCI spawn parameter for CA InterTest for CICS dynamic symbolic support of CA Endevor SCM. The dynamic symbolic support feature will not be available until CA ENF has been refreshed.

## 3. Install or Customize DFHPEP.

If your site does not have a custom DFHPEP module implemented, the DFHPEP included in the product load library must be installed in a library concatenated before the IBM CICS library suffixed with SDFHLOAD, so that it replaces the IBM supplied dummy version of DFHPEP. If your site already uses a customized DFHPEP it should be customized and relinked to include the following source statement:

```
EXEC CICS LINK PROGRAM('IN25PEP')
```

Using the supplied copy of DFHPEP or customizing your own DFHPEP to link to IN25PEP is not a hard requirement for installation of CA SymDump for CICS. The dump capture still functions without DFHPEP/IN25PEP processing. However the failure to configure DFHPEP as suggested previously prevents capture of the last screen display in an MRO environment.

## 4. Update CICS Startup JCL and Restart CICS.



**Note:** If your site has previously installed CA InterTest for CICS as a separate product, some DD statements may already exist.

If you use DD statements in your CICS startup JCL, your CA SymDump for CICS files, as defined in the CSD, require DD statements in the JCL for CICS. DISP=SHR must be specified for PROTSYM to allow batch compiles to run concurrently with CICS. Alternatively, you may add the dataset names directly to the CEDA FILE definition defined in the CSDSYM member in CAI.CAVHJCL and remove the DD statements for those files.

Add the following DD statements to your CICS startup JCL:

```
//PROTDMP DD DSN=CAI.PROTDMP,DISP=SHR
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//PROTHLF DD DSN=CAI.CAVHHLF,DISP=SHR
//PROTUHF DD DSN=CAI.PROTUHF,DISP=SHR
//PROTMLOG DD DSN=CAI.PROTMLOG,DISP=SHR
```

If you are using the CA InterTest for CICS dynamic symbolic support feature for CA Endeavor SCM, add an NDVRSYM DD statement to specify the PROTSYM file to be used for dynamic symbolic loading. The NDVRSYM DD must be specified in the CICS startup JCL for CA InterTest for CICS to invoke the dynamic symbolic load option. A sample NDVRSYM DD statement follows:

```
//NDVRSYM DD DSN=CAI.PROTSYM,DISP=SHR
```



**Note:** To view dumps from another concurrently active CICS region, each region must have its own CA SymDump for CICS file. In addition, the region used to view the dumps must have a separate FCT entry for each dump file.

If you specified more than one symbolic file in the SYMFIL variable of IN25OPTS, include a DD for each SYMFIL specified.

Add the CA SymDump for CICS load library to the DFHRPL in the CICS startup JCL.

```
// DD DSN=CAI.CAVHLOAD,DISP=SHR <productname>
```



**Note:** CA SymDump for CICS makes calls to IEWBIND when using the online COMPOSITE module command and IEWBIND requires about 750 KB of below the line storage. To make sure that there is sufficient storage available (below the line for IEWBIND calls), you must verify that your DSALIM is at least 750 KB less than your below the line private area size. Failure to do this may result in S0F4 and U900 abends in your CICS region during dump capture.

When you have completed the changes to your CICS startup JCL, restart the CICS region.

# Configure Your Product Without CA CSM

---

We recommend that you use CA CSM to configure your product, but you can configure your product manually. This section describes the tasks you need to complete to configure your product without CA CSM. Click on the link for the product that you are configuring to see the tasks for that product.

- [Configuration Checklists \(see page 26\)](#)
- [Options \(see page 31\)](#)
- [User Exits \(see page 90\)](#)
- [Allocate Files \(see page 122\)](#)
- [Interfaces and Compatibility \(see page 128\)](#)
- [Activate Your Product \(see page 178\)](#)
- [Installation Verification \(see page 186\)](#)

## Configuration Checklists

This section contains checklists to help you configure individual CA InterTest and CA SymDump products. You can install between one and four CA InterTest and CA SymDump products at one time. By default, all four products are installed.

If you want to configure all four products, be sure to complete all of the configuration steps in this section, beginning with [Options \(see page 31\)](#).

If you want to configure a single product or subset of the 4 products, use the checklists in this section to ensure you complete all of the necessary tasks.

- [Configuring CA InterTest Batch \(see page 26\)](#)
- [Configuring CA SymDump Batch \(see page 27\)](#)
- [Configuring CA InterTest for CICS \(see page 28\)](#)
- [Configuring CA SymDump for CICS \(see page 30\)](#)

## Configuring CA InterTest Batch

After you complete the installation process, configure CA InterTest Batch using CA CSM or using the manual process outlined in this article.

### More information:

[Installing \(https://docops.ca.com/display/CAITSD11/Installing\)](https://docops.ca.com/display/CAITSD11/Installing)

[Configure Your Product With CA CSM \(see page 15\)](#)

**Follow these steps to manually configure your installation of CA InterTest Batch:**

1. [Customize global options in CAVHCONF. \(see page 31\)](#)
2. [Configure ISPF options in INTISPF \(see page 52\).](#)
3. [Configure batch Link and SYSPLEX. \(see page 54\)](#)
4. [Configure external security options in CAVHCONF \(see page 85\).](#)
5. [Define the CA InterTest Batch user exits. \(see page 90\)](#)
6. [Allocate files. \(see page 122\)](#)
7. [Configure dynamic symbolic support. \(see page 128\)](#)
8. [Customize the following compatibility and interface options: \(see page 130\)](#)
  - CLISTs
  - Compile procedures
  - ISPF main panel
  - TSO logon procedure
  - CA Roscoe
  - BTS extended support
  - DB2 SP and IMS/DC Debugging
9. [Activate your product. \(see page 178\)](#)
10. [Verify your installation. \(see page 186\)](#)

## Configuring CA SymDump Batch

After you complete the installation process, configure CA SymDump Batch using CA CSM or using the manual process outlined in this article.



**More information:**

[Installing \(https://docops.ca.com/display/CAITSD11/Installing\)](https://docops.ca.com/display/CAITSD11/Installing)

[Configure Your Product With CA CSM \(see page 15\)](#)

**Follow these steps to manually configure your installation of CA SymDump Batch:**

1. [Customize global options in CAVHCONF. \(see page 31\)](#)
2. [Configure the CA SymDump Batch default options. \(see page 33\).](#)
3. [Configure external security options in CAVHCONF \(see page 85\).](#)
4. [Define the CA SymDump Batch user exits. \(see page 91\)](#)
5. [Allocate files. \(see page 122\)](#)
6. [Configure dynamic symbolic support. \(see page 128\)](#)
7. [Customize the following compatibility and interface options: \(see page 130\)](#)
  - DB2 support
  - Language Environment (LE)
  - CLISTs
  - ISPF main panel
  - Log and notify procedures
  - CA Netman
8. [Activate your product. \(see page 178\)](#)
9. [Verify your installation. \(see page 186\)](#)

## Configuring CA InterTest for CICS

After you complete the installation process, configure CA InterTest for CICS using CA CSM or using the manual process outlined in this article.



### More information:

[Installing \(https://docops.ca.com/display/CAITSD11/Installing\)](https://docops.ca.com/display/CAITSD11/Installing)

[Configure Your Product With CA CSM \(see page 15\)](#)

### Follow these steps to manually configure your installation of CA InterTest for CICS:

1. [Customize global options in CAVHCONF. \(see page 31\)](#)
2. [Configure the CA InterTest for CICS default options. \(see page 55\)](#)
3. [Configureabend codes. \(see page 81\)](#)

4. [Configure external security options in CAVHCONF \(see page 85\).](#)
5. [Define the CA \(see page 113\)InterTest \(see page 55\) for CICS user exits. \(see page 104\)](#)
6. [Allocate files. \(see page 122\)](#)
7. [Configure dynamic symbolic support. \(see page 128\)](#)
8. [Customize the following options: \(see page 150\)](#)
  - DB2 support
  - DSA limit parameters
  - CICS JCL and startup parameters
  - CICS resource definitions
  - Assembler DSECTs
9. [Customize the following interface and compatibility options for CA InterTest for CICS: \(see page \)](#)
  - PROMMAC macro
  - CNTL commands
  - Calls to software and macro support
  - DB2 support for CA InterTest for CICS
  - Wild branch handling
  - Add COBOL file structures to the symbolic file
  - User-defined CORE commands
  - HOGAN software support
  - EDF support
  - MRO support
10. [Review performance considerations. \(see page 175\)](#)
11. [Activate your product. \(see page 178\)](#)
12. [Verify your installation. \(see page 186\)](#)

## Configuring CA SymDump for CICS

After you complete the installation process, configure CA SymDump for CICS using CA CSM or using the manual process outlined in this article.

 **More information:**

[Installing \(https://docops.ca.com/display/CAITSD11/Installing\)](https://docops.ca.com/display/CAITSD11/Installing)

[Configure Your Product With CA CSM \(see page 15\)](#)

**Follow these steps to manually configure your installation of CA SymDump for CICS:**

1. [Customize global options in CAVHCONF. \(see page 31\)](#)
2. [Configure the CA SymDump for CICS default options. \(see page 55\)](#)
3. [Configure the dump capture options. \(see page 82\)](#)
4. [Configure external security options in CAVHCONF \(see page 85\).](#)
5. [Define the CA SymDump for CICS user exits. \(see page 113\)](#)
6. [Allocate files. \(see page 122\)](#)
7. [Configure dynamic symbolic support. \(see page 128\)](#)
8. [Customize the following options: \(see page 150\)](#)
  - DB2 support
  - Language Environment (LE)
  - DSA limit parameters
  - CICS JCL and startup parameters
  - CICS resource definitions
  - Assembler DSECTs
  - DFHPEP
9. [Review performance considerations. \(see page 175\)](#)
10. [Activate your product. \(see page 178\)](#)

11. [Verify your installation. \(see page 186\)](#)

## Options

This article describes options that apply to all CA InterTest and CA SymDump products. For all products, you must update CAVHCONF or the application will use the default values. After you review and optionally update CAVHCONF, set the options that apply to the specific products that you are installing. These options are described in the following pages, also located in this section of the documentation.

- [CA SymDump Batch Options \(see page 33\)](#)
- [CA InterTest Batch Options \(see page 52\)](#)
- [CICS Options \(see page 55\)](#)
- [CA InterTest for CICS Options \(see page 81\)](#)
- [CA SymDump for CICS Options \(see page 82\)](#)
- [External Security Option \(see page 85\)](#)

## Global Options

The options that control all products are stored in CAVHCONF. Modify the CAI.CAVHJCL(CAVHCONF) sample JCL to update the default values before you submit the modified job.

You can update any of the following keywords.



**Note:** SYMPSWD and EXTSEC=ALL|SYMBOLIC are mutually exclusive and if EXTSEC=ALL or EXTSEC=SYMBOLIC is specified then SYMPSWD will be ignored.

- **ABEND\_PGM\_FILTER**

Activates the abend program filtering option in CA SymDump Batch. The default is OFF. To activate this option, specify ON and create a list of abend program names in the CAVHFMOD module. The specified abend program names are then from the dump index and replaced with a meaningful program name. This name is obtained by going up one level on the active program call chain. Use ABEND\_PGM\_FILTER to avoid filling the dump index with multiple instances of a common error handling program name.



**Note:** This filtering applies *only* to the dump index. The abending program name and offset are not changed in the abend report.

**Syntax:**

ABEND\_PGM\_FILTER=(OFF|ON)

**More information:**[Define the Abend Filter Table CAVHFMOD \(see page 50\)](#)

- **EXTSEC**

Specifies the external security settings in IN25UTIL and InterTest Batch. Specify SYMBOLIC to activate external security for the symbolic file INITIALIZE, UPDATE, DELETE, PURGE, and RELOAD functions. Specify INTBAT to activate external security for mismatched symbolic members.

**Syntax:**

```
EXTSEC=(NONE|ALL|SYMBOLIC,INTBAT)
```

- **IBSYSPLEX**

Specifies whether the sysplex support of InterTest Batch is enabled.

**Default:** N

- **PINT**

Specifies whether a product can be used on a specific LPAR. If permitted, then the normal LMP checks for that product are performed. If not permitted, a message appears stating that the product is unavailable on this LPAR. (This message appears in place of an LMP warning.) The default value is DYNAMIC. This value causes the product to determine which products have valid LMP keys on an LPAR. The product then sets the product intent values accordingly.

**Syntax:**

```
PINT=(INTBAT,INTCICS,SYMBAT,SYMCICS|ALL|DYNAMIC)
```

- **SCHTBSIZE**

Specifies the maximum number of entries in the Batch Link Scheduling Table for monitoring DB2 stored procedures and IMS/DC transactions.

**Default:** 20

- **SYMPSWD**

Specifies the PROTSYM password.

**Default:** 12345678**Limits:** Up to eight characters

- **WORKUNIT**

Specifies the unit name for temporary data sets.

**Default:** SYSDA**Limits:** Up to eight characters**Examples:**

```
CAVHCONF TYPE=CSECT,PINT=DYNAMIC,ABEND_PGM_FILTER=ON
```

```
CAVHCONF TYPE=CSECT,PINT=(SYMBAT,INTBAT)
```

```
CAVHCONF TYPE=CSECT,PINT=ALL
```

```
CAVHCONF TYPE=CSECT,PINT=DYNAMIC
```

```
CAVHCONF TYPE=CSECT,SYMPSWD=87654321
```

```
CAVHCONF TYPE=CSECT,PINT=(SYMBAT,INTBAT),EXTSEC=SYMBOLIC
```

CAVHCONF TYPE=CSECT,IBSYSPLEX=Y,SCHTBSIZE=30

## CA SymDump Batch Options

This article describes the following CA SymDump Batch options:

- [Execution-Time Options CAOUDFRX](#) (see page 33)
- [Abend Handler and Reporter CAOETABL](#) (see page 37)
- [Define the User Abend Control Table CAOCAB80](#) (see page 43)
- [Define the Abend Filter Table CAVHFMOD](#) (see page 50)
- [Facilitate Multiple Default Execution Options](#) (see page 51)

Data set CAI.CAVHJCL contains members CAOUDFRX, CAOETABL, and CAOCAB80. These members contain the JCL to assemble and link the product defaults.

Data set CAI.CAVHPROC contains the CAOUASML PROC to assemble and link the default table.

All CA SymDump Batch options default to unfixed, indicating that they can be overridden in the JCL. To fix an option, insert an asterisk (\*) between the equal sign (=) and the option value.

Optionally copy and update each member to conform to your installation standards. Submit each job to create a new set of defaults.



**Important:** Ensure that you change the following data set names to conform to your installation standards:

- The default symbolic libraries (member CAOUDFRX, macro CAOUSYM)
- The CAIPRINT repository library (member CAOETABL, option PRTLIB)

### Execution-Time Options CAOUDFRX

This section describes the options that CA SymDump Batch uses at execution time.



**Note:** Default options member CAOUDFRX contains references to options that are not documented here. These options are used by other CA Technologies products that share this member. These options have no effect on CA SymDump Batch.

- **CAOUSYM *dsname***  
Adds a symbolic file (a PROTSYM or CSL) to the search list for abend and Snap reports. Insert one CAOUSYM statement for each symbolic file used at your installation.

- **CAOUXDUP *modname***  
Specifies whether to add a control section name to the exclusion list for duplicate dump suppression. Control sections in the exclusion list are not subject to the duplication limit (DUPLIM) value. Exclude as many control sections as needed by including one CAOUXDUP statement for each name.
- **CAOUXMOD *modname***  
Specifies whether to add a control section name to the exclusion list for abend reports. Control sections in the exclusion list are not displayed in dump format as part of an abend report. Exclude as many control sections as needed by including one CAOUXMOD statement for each name.
- **DUPLIM=5|*nnn***  
Specifies the maximum number of duplicate reports to store in a single repository. Specify a number from 1 to 999, or specify 0 to suppress duplication checking. Duplicate reports have the same abending program name, program offset, and completion code. Lowering the value of DUPLIM does not delete any existing reports in the repository.
- **EXPDAYS=0|*nnn***  
Specifies the maximum number of days to retain a report in the repository, starting from the day it is added. Specify the number of days from 1 to 999, or specify 0 to suppress report expiration. Expired reports are automatically deleted during the auto-maintenance cycle. This cycle occurs immediately before any new report is added. During auto-maintenance, the age of each report is compared to the *current* EXPDAYS value. (Auto-maintenance does not use the EXPDAYS value when the reports were added.)
- **LANGMSG=ENGLISH|*xxxxxxxx***  
Specifies the language in which to write all execution-time messages. Only ENGLISH is supported currently.
- **LANGWTO=ENGLISH|*xxxxxxxx***  
Specifies the language in which to write all execution-time WTOs. Only ENGLISH is supported currently.
- **LIBDD=SYSLIB|*ddname***  
Specifies the ddname that MRS uses to identify the load library being reported on.
- **LINECNT=60|*nnn***  
Specifies the maximum number of lines to display on a page for all execution-time reports.
- **NDVRDD=CAINDVR|*ddname***  
Specifies the ddname that is used in the execution JCL to define the symbolic file. The symbolic file is populated dynamically when Dynamic Symbolic Support (NDVRDSS) is active for CA Endeavor SCM.
- **NDVRDSN=*dsname***  
Specifies the symbolic file that is populated dynamically when Dynamic Symbolic Support (NDVRVSS) is active for CA Endeavor SCM. Specify a PROTSYM file. When a data set is specified, it is included in the symbolic file search list automatically. If you do not want to specify a default dsname, leave this option blank.

- **NDVRPROC=*procname***  
Specifies the seven-character prefix of the procedure name. This procedure is started when Dynamic Symbolic Support (NDVRDSS) is active for CA Endeavor SCM and when multiple site IDs are used by the installation.
- **OPTSDD=CAIOPTS|*ddname***  
Specifies the ddname that references the execution-time options data set.
- **PANACCS=PANEACCS|*pgmname***  
Specifies the CA Panexec library access routine. This option is ignored at execution time if PANEXEC=OFF is specified.
- **PANEXEC=OFF|ON**  
Specifies whether CA Panexec is present on the system.
- **PRINTDD=CAIPRINT|*ddname***  
Specifies the ddname that references the report file.
- **PRTLBD=CAIPRTL|*ddname***  
Specifies the ddname that references the central VSAM repository.
- **REGMAX=(*nnnn*|128,*nnnn*|256)**  
Specifies the number of storage bytes to format before and after each register address for Assembler programs.
- **SAVEHEAP=ON|OFF**  
Specifies whether to save LE or PL/I heap storage for abending applications.
- **SHOWHEAP=ON|OFF**  
Specifies whether to format LE or PL/I heap storage. For LE enabled Assembler or PL/I programs, SHOWHEAP=ON requires SAVEHEAP=ON at execution time.
- **SHOWUNMRG=ALL|NONE|TBLS**  
Specifies whether to display storage for which symbolic names are not merged. This option applies only when merging options are used. This storage can result from *slack* bytes between fields. This storage can also result from table entries which are not merged because of limits imposed by the OCCURS option. The following options are valid:
  - **ALL**  
Formats all unmerged storage.
  - **NONE**  
Suppresses unmerged storage.
  - **TBLS**  
Formats only table storage which is unmerged because of the limits imposed by the OCCURS option.
- **SNAPS=0|*nnnn*|ON|OFF**  
Specifies the maximum number of Snap reports to produce during a job step. When this value is exceeded, all calls to CAODSNAP are ignored.  
The default value for *nnnn* is 32767.



**Note:** Each call to CAODSNAP generates a separate SNAP report in your central VSAM repository. If your program contains multiple SNAP calls or a SNAP call within a loop, you can flood the repository. To avoid this problem, you can write the output to the CAIPRINT DD alone. See the PRTREPT option.

- **SPBLKLN=0|nnnnn**  
Specifies the default block length to use for dynamic allocation.
- **SPPRIM=1|nnnnnnnn**  
Specifies the primary space allocation to use for dynamic allocation.
- **SPSECND=1|nnnnnnnn**  
Specifies the secondary space allocation to use for dynamic allocation.
- **SPUNITS=CYL|TRK|BLK**  
Specifies the default space units to use for dynamic allocation.
- **STORCLS=storclas**  
Specifies the default storage class to use for dynamic allocation. Currently, this option is used only for temporary files that the viewer creates.
- **SUPPRS=OFF|ON**  
Specifies whether to suppress the following items:
  - The CAIOPTS File Processing report and Report Summary for programs containing snap calls when the SNAP option is OFF.
  - The CAIOPTS File Processing report and Execution Monitor Summary for applications containing one or more programs that were optimized using the CA Optimizer DTECT option and which terminate normally.
- **SYMDD=CAISYM|ddname**  
Specifies the ddname that is used in your execution JCL to define a symbolic file override.
- **SYSOUT=\*|x**  
Specifies the default SYSOUT class to use when report files or dump files are allocated dynamically.
- **SYSOUTD=destination**  
Specifies the default destination for SYSOUT data sets that are allocated dynamically for report or dump files. The default is blank. When blank, no destination is used.
- **UNIT=SYSDA|xxxxx**  
Specifies the default UNIT name to use for dynamic allocation.
- **USERID='CA'|string**  
Specifies a text string that appears in the top left corner of each page of the execution reports.

## Abend Handler and Reporter CAOETABL



**Note:** If CA SymDump System is also installed, set the DUMP option to ON in the default member CAOETABL.

### Exclusion Options

CAOETABL includes the CAOETXCL macro, which lets you specify exclusion criteria for your abend reports.

You can use the following wildcard characters when specifying your criteria:

- **\***  
Denotes a prefix. For example, `ABC*` excludes all items that start with ABC.
- **?**  
Indicates a single character. For example, `AB??EFGH` excludes any items that start with "AB" and have "EFGH" in positions 5-8.
- **:**  
Delimits a range. For example, `100 : 200` excludes any account number from 100 to 200.



**Note:** This wildcard applies only to the `ACCOUNT=` keyword.

You can specify the following keywords in the CAOETXCL macro:

- **ABNDCDE=**  
Specifies a list of the system abend codes to exclude. Each entry is exactly 3 bytes.
- **ACCOUNT=**  
Specifies a list or range of account numbers to exclude. If an account number includes subfields, they are concatenated and treated as a single field. The maximum length is 25 bytes.
- **JOBNAME=**  
Specifies a list of job names to exclude. The maximum length is 8 bytes.
- **PGMNAME=**  
Specifies a list of program names to exclude. These names correspond to the name of the job step program. These names do not necessarily match the name of the abending program. The maximum length is 8 bytes.
- **SMFIDS=**  
Specifies the SMF ID of the running system to exclude. The maximum length is 4 bytes.

- **SUBSYS=**  
Specifies the subsystem type to exclude. Subsystem means a batch job (INIT), a started task (STC), a TSO session (TSO), IMS, DB2, or CA Datacom/DB.
- **USRABND=**  
Specifies a list of the user abend codes to exclude. Each entry is exactly 4 bytes.

## Reporting, Logging, and Security Options

Use the following keywords in macro CAOETGEN to set the reporting, logging, and security options:

- **ACB=OFF|ON**  
Specifies whether to print the access method control blocks for open VSAM files.
- **AMB=OFF|ON**  
Specifies whether to print the access method blocks for open VSAM files.
- **AMBL=OFF|ON**  
Specifies whether to print the access method block list for open VSAM files.
- **ASMINST=ON|OFF**  
Displays the abending Assembler instruction and associated operands on the abend page.
- **BLLMAX=512|nnnnn**  
Specifies the maximum number of bytes to format for an unknown BLL cell range. (The length of an 01-item in the LINKAGE SECTION cannot always be determined.) The maximum value is 4096.
- **BINDER=ON|OFF**  
Specifies whether to access user load modules through the binder API function calls or the standard I/O macros.
- **BINFRMT=DEC|HEX**  
Specifies whether to display the binary data values in decimal display or hexadecimal format.
- **BLOCKS=OFF|ON**  
Specifies whether to print all data management control blocks. This keyword overrides all other options that control the formatting of control blocks.
- **COBONLY=OFF|ON**  
Specifies whether to suppress an abend report when the ABEND does not occur in a COBOL program.
- **DB2ACTIV=OFF|ON**  
Specifies whether to display summary information for packages and DBRMs that are active when an ABEND occurs.
- **DCB=OFF|ON**  
Specifies whether to format the data control blocks for all open files.
- **DEB=OFF|ON**  
Specifies whether to format the data extent blocks for all open files.

- **DUMP=OFF|ON**  
Specifies whether to produce a system dump after the abend report is written.
- **FILES=ON|OFF**  
Specifies whether to report on all open files that did not appear in the File Section displays for any COBOL program in the abend report.
- **FST128=OFF|ON**  
Specifies whether to limit the display of file buffers to the first 128 bytes. This option affects the File Section displays and the Open Files report.
- **GRPADDR=ON|OFF**  
Specifies whether to display the base locator, displacement, and address of a group item on the merged Data Division displays.
- **IOB=OFF|ON**  
Specifies whether to format the input/output blocks for all open files.
- **LINECNT=60|nnn**  
Specifies the maximum number of lines to display on each page of an abend or Snap report.
- **LINKAGE=ON|OFF**  
Specifies whether to format the LINKAGE SECTION of any COBOL program being reported on.
- **LCLSTOR=ON|OFF**  
Specifies whether to format the LOCAL-STORAGE SECTION of any COBOL program being reported on.
- **LISTLINE=0|nn**  
Specifies the number of extra listing lines to merge into a report before and after the source statement at ABEND, snap, or transfer.
- **LOGNET=OFF|ON**  
Specifies whether to write a record to CA Netman to use for automated problem tracking. If ON is specified, also set the CA Netman files that are required for Machine-Generated Problem Tracking (MGPT).
- **LOGROS=ON|OFF**  
Specifies whether to send a message describing the ABEND to the CA Roscoe IE user who submitted the job. If ON is specified, also specify the CA Roscoe IE job name in the ROSCOE option.
- **LOGTSO=ON|OFF**  
Specifies whether to send a message describing the ABEND to the TSO user who submitted the job.
- **LOGUNI=ON|OFF**  
Specifies whether to send a message describing the ABEND to the CA NSM console.
- **MEMMAP=ON|OFF**  
Specifies whether to format the DSA, TGT, and PGT Memory Maps for any COBOL program being reported on.

- **MERGEDB=ON | OFF**  
Specifies whether to map areas in the DB2, CA IDMS/DB, and IMS database reports symbolically.
- **MARGAUTO=ON | OFF**  
Specifies whether to merge symbolic names onto automatic storage for PL/I procedures.
- **MRGBASED=ON | OFF**  
Specifies whether to merge symbolic names onto based variable storage for PL/I procedures.
- **MRCNTLD=ON | OFF**  
Specifies whether to merge symbolic names onto controlled storage for PL/I procedures.
- **MRGDATA=ON | OFF**  
Specifies whether to turn all merging options ON or OFF.
- **MRGDSECT=ON | OFF**  
Specifies whether to merge symbolic names onto Assembler program storage when DSECTs are used and USINGs are defined.
- **MGRFILES=ON | OFF**  
Specifies whether to map the FILE SECTION display symbolically.
- **MRGLINK=ON | OFF**  
Specifies whether to map the LINKAGE SECTION display symbolically.
- **MRGLOCAL=ON | OFF**  
Specifies whether to map the LOCAL-STORAGE SECTION display symbolically.
- **MRGPARMS=ON | OFF**  
Specifies whether to merge symbolic names onto parameter storage for PL/I procedures.
- **MRGSTAT=ON | OFF**  
Specifies whether to merge symbolic names onto static storage for PL/I programs.
- **MRGWORK=ON | OFF**  
Specifies whether to map the WORKING-STORAGE SECTION display symbolically.
- **NDVRASM=OFF | ON**  
Specifies whether the symbolic file is always populated dynamically for assembler programs which are not LE enabled. This option takes effect only when Dynamic Symbolic Support (NDVRDSS) is active for CA Endeavor SCM.
- **NDVRDSS=OFF | ON**  
Specifies whether Dynamic Symbolic Support is active for CA Endeavor SCM.
- **NTMGCTL=CAI.NTMMGCTL | OFF**  
Specifies the CA Netman NTMMGCTL file that is allocated dynamically when LOGNET=ON is specified.
- **NTMPI=CAI.NTMPI | *dsname***  
Specifies the CA Netman NTMPI file that is allocated dynamically when LOGNET=ON is specified.

- **NTMSC=CAI.NTMSC|*dsname***  
Specifies the CA Netman NTMSC file that is allocated dynamically when LOGNET=ON is specified.
- **OCCURS=1|*nnnnnnnn*|MAX**  
Specifies the maximum number of table occurrences to map symbolically when using the merging options. MAX sets the option to its maximum value of 16777215.
- **OPMEM=*xxxxxxxx***  
Specifies the member name that contains the execution options when the options file is a dynamically allocated PDS.
- **OPSUFFIX=CAIOPTS|*xxxxxxxx***  
Specifies the suffix to append to the user ID that submitted the job. This suffix is used when no options data set is included in the JCL. The suffix and ID create the options data set name for dynamic allocation.
- **OPTSDD=CAIOPTS|*ddname***  
Specifies the ddname for the options file that is dynamically allocated when it is not specified in the JCL. The abend Handler/Reporter performs the dynamic allocation, but the Execution Monitor reads the file. This name must match the name that is specified for the same variable in the CAOETGEN defaults member.
- **PLH=OFF|ON**  
Specifies whether to format the place holders for open VSAM files.
- **PRTLHCT=80|*nn***  
Specifies the percent full the PRTLIB dataset must reach to trigger a health check exception condition. If this percentage is reached the health check program will go into exception state and issue consol messages every 15 minutes until the problem is corrected.
- **PRTLIB=CAI.PRTLIB|*dsname***  
Specifies the central VSAM repository to write abend and Snap reports. If you do not want to use a repository, leave this option blank. For more information, see [Allocate CAIPRINT Repository Library \(see page 125\)](#).
- **PRTREPT=BOTH|REPOS|PRTDD**  
Specifies whether to write abend or Snap reports to the central VSAM repository (REPOS), the CAIPRINT DD (PRTDD), or both.
- **READLL=ON|OFF**  
Specifies whether CA SymDump Batch reads the LINKLIST libraries to obtain module information for abend and Snap reports. If READLL is OFF, CA SymDump Batch may not be able to report on programs that were loaded from LINKLIST.
- **REGMAX=(*nnnn*|128,*nnnn*|256)**  
Specifies the number of storage bytes to format before and after each register address for Assembler programs.
- **ROSCOE=*list of names***  
Specifies the CA Roscoe IE job names to use when LOGROS=ON is specified. You can include up to four names.

- **RPL=OFF|ON**  
Specifies whether to format the request parameter list for open VSAM files.
  - **RPTSZ80=OFF|ON**  
Specifies whether to force all data that is in dump format to appear in an 80-column format.
  - **SAVEHEAP=ON|OFF**  
Specifies whether to save LE or PL/I heap storage for abending applications.
  - **SHOWHEAP=ON|OFF**  
Specifies whether to format LE or PL/I heap storage. For LE enabled Assembler or PL/I programs, SHOWHEAP=ON requires SAVEHEAP=ON at execution time.
  - **SHOWUNMRG=ALL|NONE|TBLS**  
When merging options are used, this option specifies whether to display storage for which symbolic names are not merged. This storage can result from "slack" bytes between fields. This storage can also result from table entries that are not merged because of limits imposed by the OCCURS option. The following options are valid:
    - **ALL**  
Formats all unmerged storage.
    - **NONE**  
Suppresses unmerged storage.
    - **TBLS**  
Formats only table storage which is unmerged because of the limits imposed by the OCCURS option.
  - **SMF=OFF|ON**  
Specifies whether to write a record to the System Management Facility. If ON is specified, include the SMF record number in the SMFREC option.
  - **SMFREC=*nnn***  
Specifies the SMF record number for SMF recording when SMF=ON is specified. This value must be from 128 through 255, and it cannot conflict with an existing record number.
  - **SNAPS=ON|OFF**  
Specifies whether to support SNAP requests in a program.
-  **Note:** Each call to CAODSNAP generates a separate SNAP report in your central VSAM repository. If your program contains multiple SNAP calls or a SNAP call within a loop, you can flood the repository. To avoid this problem, write the output to the CAIPRINT DD alone. See the PRTREPT option.
- **STOPDD=CAOESTOP|*ddname***  
Specifies the ddname that is used in the execution JCL to suppress abend reports.
  - **SVAREA=ON|OFF**  
Specifies whether to produce a Save Area Trace report. This report shows the changed save areas, starting with the program that abended.

- **SYSOUT=\*|x**  
Specifies the default SYSOUT class to use at execution time when report files or dump files are allocated dynamically.
- **UCB=OFF|ON**  
Specifies whether to format the unit control blocks list for open files.
- **VSAMCAT=OFF|ON**  
Specifies whether to format the VSAM catalog information for each open VSAM file.
- **VSMIDX=OFF|ON**  
Specifies whether to format the VSAM indexed control blocks for the index portion of each open VSAM file with the control blocks for the data portion.
- **WRKSTOR=ON|OFF**  
Specifies whether to format the WORKING-STORAGE of each COBOL program in an abend report.
- **WTL=OFF|ON**  
Specifies whether to send a message describing the ABEND to the system log.
- **WTO=ON|OFF**  
Specifies whether to send a message describing the ABEND to the job log (using WTO with ROUTCDE=11).
- **WTOXCL=OFF|ON**  
Specifies whether to send a message which describes the reason that a report was not generated because of exclusion criteria.

## Define the User Abend Control Table CAOCAB80

The following tables control the abend processing:

- **Product Table**  
Contains predefined controls. The product table contains only the common system abend codes.
- **User Table**  
Contains user-defined abend control entries. These entries can define new controls, and can also override existing controls in the product table.



**Note:** The user abend control table must be link-edited with the load module name CAOCAB80.

Each entry in a control table defines the processing options for a specific abend code.

You can design diagnostic programs for the user abends and system abends.

### Follow these steps:

1. Add an entry to the user abend control table, CAOCAB80, for the specific abend using the DMOD parameter.

2. Use the CPXACODE macro (included with the product) to define user abend control table entries for CAOCAB80. The following example shows the syntax to use. The first positional parameter is required.

```
[Label] CPXACODE START
CPXACODEabendcode[,keyword=ON|OFF[,...]][,DMOD=modulename]CPXACODEEND
```

- **label**  
Identifies a specific code section.
- **START**  
Generates a CSECT statement. If a label is specified, the label is used as the CSECT name. Otherwise, the generated CSECT name is the default, CAOCAB80.



**Note:** The CSECT must be link-edited as a single CSECT load module. The module must be named CAOCAB80.

- **abendcode**  
Specifies the abend code. The following values are valid:
  - S0001-SFFF (for the system abend codes)
  - U0001-U4095 (for the user abend codes)
- **keyword**  
Specifies a keyword and whether to enable the abend reporting function:
  - **ON**  
Enables the abend reporting function for this keyword.
  - **OFF**  
Disables the abend reporting function for this keyword.

The keywords in this macro have the same function as the keywords in the CAIOPTS (SYSIN) data set. For more information about these keywords, see [CA SymDump Keyword Descriptions \(see page 45\)](#) and [Data Management Control Block Keywords \(see page 45\)](#).

- **DMOD**  
(Optional) Specifies the program name of an existing user diagnostic module. The CA SymDump Batch diagnostic modules are still called when the user CPXACODE macro definitions do not specify the DMOD operand.
- **END**  
Generates a trailer table entry that must be specified last.

#### Example:

The following syntax adds a new entry to the user control table:

```
CAOCAB80 CPXACODE START
CPXACODE U1005,BLOCKS=ON,DUMP=OFF
CPXACODE S322,DUMP=OFF,DMOD=TIMEOUT
```

```
CPXACODE END  
END
```

## CA SymDump Batch Keyword Descriptions

This section describes the CA SymDump Batch keywords:

- **DTECT**  
Turns abend reporting on and off.  
**Default:** ON
- **DUMP**  
Forces a SYSUDUMP.  
**Default:** OFF
- **F128**  
Prints only the first 128 bytes of a record.  
**Default:** OFF
- **MEMMAP**  
Prints the DSA, TGT, and PGT memory map.  
**Default:** ON
- **FILES**  
Prints open files report.  
**Default:** ON
- **SAVEAREA**  
Prints the save area trace report.  
**Default:** ON

## Data Management Control Block Keywords

This section describes the Data Management Control Block keywords:

- **BLOCKS**  
Sets all Data Management keywords to OFF.  
**Default:** OFF
- **ACB**  
Includes the VSAM Access Method Control Block.  
**Default:** OFF
- **AMB**  
Includes the VSAM Access Method Block.  
**Default:** OFF
- **AMBL**  
Includes the VSAM Access Method Block List.  
**Default:** OFF
- **DCB**  
Includes the Data Control Block.  
**Default:** OFF

- **DEB**  
Includes the Data Extent Block.  
**Default:** OFF
- **IOB**  
Includes the I/O Block.  
**Default:** OFF
- **PLH**  
Includes the VSAM Place Holder.  
**Default:** OFF
- **RPL**  
Includes the VSAM Request Parameter List.  
**Default:** OFF
- **UCB**  
Includes the Unit Control Block.  
**Default:** OFF
- **VSAMIDX**  
Includes the VSAM Indexed Control Blocks.  
**Default:** OFF



**Note:** Use ON to print the indicated control blocks; use OFF if you do not want the control block to print.

## User-Supplied Abend Diagnostic Modules

You can develop abend diagnostics modules to supplement CA SymDump Batch processing. These programs provide diagnostic material for the abend and diagnostic logic to determine the abend cause.

CA SymDump Batch provides a CPXDIAG macro to help you build diagnostic modules. This macro provides entry logic and exit logic, user workspace, and addressability to parameter information. This macro also provides the Abend Reporter interface for printing lines and altering abend control information.

The use of the CPXDIAG macro assumes the following register conventions:

Register	Description
R0, R1, R14, R15	Standard macro linkage and usage
R13	Provides addressability to register save area, parameter information, print line build area, and user workspace
R12	Program base register

See the following CPXDIAG function definitions:

- The following syntax establishes the CSECT name and the program and DSECT addressability, and saves the caller registers. Specify this function type first and code a label.

```
label CPXDIAG TYPE=START
```

- The following syntax restores the caller registers and returns. Specify this function at the end of your processing.

```
label CPXDIAG TYPE=END
```

- The following syntax writes a block of text to the abend report. The text is automatically centered on the report. Enclose each line of text in apostrophes. Verify that each line is less than or equal to 76 bytes. Generate blank lines by coding two or more contiguous commas.

```
label CPXDIAG TYPE=BLKTEXT, 'L1', 'L2', 'L3', ..., 'Ln'
```

- The following syntax writes a 133-byte line of data to the abend report. Provide the address of this line by using the AREA keyword. The first byte of the line is reserved for an ASA control character; you must place one there.

```
label CPXDIAG TYPE=PRINT, Rx, AREA= <label>
```

- The following syntax alters the abend control table options, but only for the current execution. You can specify one or more of the listed keywords. Assign a value of ON or OFF to each keyword. See the third point in the following example.

```
label CPXDIAG TYPE=ALTER, ACB=, AMB=, AML=,
BLOCKS=, DCB=, DEB=,
DTECT=, DUMP=, FILES=,
F128=, IOB=, MEMMAP=,
PLH=, RPL=, SAVAREA=,
UCB=,
```

### Example:

The following example shows a user diagnostic module:

```
SR1234 CPXDIAG TYPE=START  establish CSECT name and addressability
                                           column 72
1.      CPXDIAG  TYPE=BLKTEXT,                X
        'U1234 - PAYROLL SYSTEM ABEND.',      X
        'FICA TABLE IS NOT CURRENT'         X
        .
        .
        user diagnostic logic
        .
        .
2.      MVI CPXLINE,C'0'                      DOUBLE SPACE
        MVC CPXLINE+1(132),FICATABLE         LOAD 132 CHARACTER AREA
        CPXDIAG  TYPE=PRINT,AREA=CPXLINE
                                           column 72
3.      CPXDIAG  TYPE=ALTER,                    X
        DUMP=OFF,DTECT=OFF
        CPXDIAG  TYPE=END Notes:
```

Consider the following important points:

- When printing a text block, create a double space by specifying a NULL operand. (This operand consists of two apostrophes with no intervening spaces; for example, ' '). Enclose all text in single quotes. Each text block can contain up to 76 characters between the quotes.
- When using carriage control characters, consider the following information:
  - 'blank' represents a single space.
  - '0' represents a double space.
  - '-' represents a triple space.
- Instruct CA SymDump Batch not to perform any additional abend processing and to suppress the system dump.

The following information is available to the user diagnostic module automatically. The CPXDIAG TYPE=START macro automatically provides addressability using Register 13. You can build your own print lines in the CPXLINE space provided, or by using the TYPEPRINT macro. In addition, 120 bytes of workspace are provided at CPXUSPCE for your own routines.

CPXDSECT	DSECT		
CPXSAVE	DS	18F	REGISTER SAVE AREA
CPXIFACE	DS	A	ADDRESS OF MACRO INTERFACE TO DETECTOR
CPXMDGBL	DS	A	INTERFACE PARM
CPXIPRM	DS	5A	PARM LIST TO INTERFACE RTN
CPXTCB	DS	A	ADDRESS OF ABENDING TASK TCB
CPXTIOT	DS	A	ADDRESS OF ABENDING TASK TIOT
CPXCSECT	DS	CL8	ABENDING CSECT NAME
CPXENTRY	DS	A	ABENDING CSECT ENTRY POINT ADDRESS
CPXDATE	DS	CL8	CURRENT DATE "MM/DD/YY" GREGORIAN FORMAT
CPXTIM	DS	CL8	TIME OF ABEND "HH.MM.SS" FORMAT
CPXSFLAG	DS	X	STAE/ESTAE FLAGS
CPXREQ	EQU	X'80'	DUMP REPORT HAS BEEN TERMINATED, STEP OPTION SPECIFIED ON ABEND MACRO
CPXSTCC	EQU	X'10'	DON'T STORE COMPLETION CODE, NOT USED IN OS/V52
CPXRETC	DS	XL1	SYSTEM RETURN CODE FOR GIVEN ABEND
	DS	XL1	FUTURE USE
CPXATYPE	DS	CL1	ABEND TYPE
CPXUSER	EQU	C'U'	USER ABEND
CPXSYSTEM	EQU	C'S'	SYSTEM ABEND
CPXACODE	DS	CL4	ABEND CODE 0001-FFFF
CPXAPSW	DS	XL8	PSW AT ENTRY TO ABEND
CPXPPSW	DS	XL8	LAST PROBLEM PROGRAM PSW
CPXR0	DS	F	REGISTERS AT ABEND
.	.	.	.
.	.	.	.
.	.	.	.
CPXR15	DS	F	
	DS	8A	FUTURE USE
CPXLINE	DS	CL133	USER PRINT LINE WORK AREA
	DS	XL3	FUTURE USE
	DS	0F	
CPXUSPCE	DS	XL120	USER WORK SPACE
CPXLENGTH	EQU	*-CPXDSECT	LENGTH OF DIAGNOSTIC INFORMATION AREA

The following table contains the default product settings. You can update CAOCAB00 to change these settings.

Abend Code	FIRST128	MEMMAP	FILES	SAVEAREA	DUMP
001	ON		ON	ON	
002	ON		ON		
013	ON		ON		
031				ON	
038			ON		
03D					
0C1	ON	ON	ON	ON	
0C2	ON	ON	ON	ON	
0C4	ON	ON	ON	ON	
0C5	ON	ON	ON	ON	
0C6	ON	ON	ON	ON	
0C7		ON	ON		
0C8	ON	ON			
0C9	ON	ON			
0CA	ON	ON			
0CB		ON			
137	ON		ON		
213	ON		ON		
214	ON		ON		
237	ON		ON		
314	ON		ON		
322		ON	ON	ON	
413	ON				
513	ON		ON	ON	
522				ON	
613	ON		ON		
637	ON		ON		
706	ON				
713			ON		
714	ON		ON	ON	
722	ON		ON	ON	
804				ON	
806	ON	ON			
80A	ON			ON	
813					
913	ON		ON		

Abend Code	FIRST128	MEMMAP	FILES	SAVEAREA	DUMP
A13	ON		ON		
B37	ON		ON		
C13	ON		ON		
D37	ON		ON		
E37	ON		ON		
Default Entry (for any abend types not shown)		ON	ON	ON	ON

## Define the Abend Filter Table CAVHFMOD

The CAI.CAVHJCL(CAVHFMOD) member controls which abend program names are filtered from your dump index. This member is used by CA SymDump Batch.

Many of your applications may use a common abend handling program. When an abend occurs, the common program name and its offset appear in the dump index for all application failures. Multiple occurrences of a common name make it difficult to identify the abending application correctly. This problem also makes it difficult to implement automatic notification of the appropriate personnel when production failures occur.

However, you can use CAVHFMOD to specify one or more program names to filter. When an application abends, the specified names do not appear in the dump index. Instead, they are replaced with a meaningful program name and offset. These values are obtained by going up one level on the active program call chain.



### Notes:

- The program names that you specify here are not filtered from the dump index unless you activate the filtering option. Specify [ABEND\\_PGM\\_FILTER ON \(see page \)](#) in CAI.CAVHJCL(CAVHCONF).
- This filtering applies *only* to the dump index. The abending program name and offset are not changed in the abend report.

To configure the CAVHFMOD options, access the member and edit its JCL. This member contains sample JCL to assemble and link the CAVHFMOD filter table.

Use the following syntax to specify which program names to filter:

```
CAVHFMOD START
```

```
CAVHFMOD FILTER,PROGRAM=name
```

CAVHFMOD END

- **START**  
Indicates the start of the abend program list. Specify START once, before any FILTER syntax.
- **FILTER,PROGRAM=*name***  
Specifies a program name to filter. You can specify a fully qualified program name. You can also use the following wildcards:
  - **\***  
Denotes a prefix. For example, ABC\* selects all program names that start with ABC.
  - **?**  
Indicates a single character. For example, AB??EFGH selects all program names that start with "AB" and have "EFGH" in positions 5-8.
- **END**  
Indicates the end of the abend program list. Specify END once, after all FILTER syntax.

#### Example: Filter Four Program Names From the Dump Index

The following syntax sample specifies four abend program names to filter:

```
CAVHFMOD START

CAVHFMOD FILTER,PROGRAM=IEFBR14

CAVHFMOD FILTER,PROGRAM=CEE*

CAVHFMOD FILTER,PROGRAM=I?BABEND

CAVHFMOD FILTER,PROGRAM=USERDUMP

CAVHFMOD END

END ,
```

When this syntax is submitted, the specified program names are filtered out. These names are replaced with the program name from the preceding level of the active program call chain.

## Facilitate Multiple Default Execution Options

The CA SymDump Batch defaults are established during installation. Each application programming area that uses CA SymDump Batch can require different run-time defaults. In that case, the desired options are passed to the Execution Monitor by using the CAIOPTS data set.

#### Follow these steps:

1. Establish a library for the default options.

2. Create a library member for each programming area. Populate each member with the appropriate execution options for that area.
3. Verify that the CAIOPTS DD points to the correct library member.

**Examples:**

Use the following syntax when the program is executed directly:

```
//CAIOPTS DD DSN=EXEC.PARMLIB(MEMBERA),DISP=SHR
```

Use the following syntax to override a default option in the library member. After this syntax, add the syntax for those options that are different from, or not specified in, the MEMBERA member.

```
//CAIOPTS DD DSN=EXEC.PARMLIB(MEMBERA),DISP=SHR
//          DD *
```

Use the following syntax when a GO step in a cataloged procedure executes the program. Specify your member name using the symbolic parameter PARAMEM when invoking the procedure.

```
//CAIOPTS DD DSN=EXEC.PARMLIB(&PARAMEM),DISP=SHR
```

Use the following syntax to override a default option in the library member. After this syntax, add the syntax for those options that are different from, or not specified in, the MEMBERA member.

```
//GO.CAIOPTS DD DSN=EXEC.PARMLIB(MEMBERA),DISP=SHR
//          DD *
```



**Note:** If CA SymDump System is also installed, specify DUMP=ON in the default member CAOETABL.

## CA InterTest Batch Options

This section contains instructions how to customize CA InterTest Batch ISPF options, batch link options and enable the Cross-LPAR debugging feature.

- [Configure ISPF Options \(see page 52\)](#)
- [Configure Batch Link and Sysplex \(see page 54\)](#)

### Configure ISPF Options

When you are configuring CA InterTest Batch without CA CSM, use member INTISPF in the CAVHSAMP library to customize the options that are passed to the application when it is invoked under ISPF.

The INTISPF member parameters are as follows:

- **ISPF**  
This is required to run the application under the ISPF dialog manager.
- **SYMLDYN**  
This parameter is no longer used.

- **TRACE(*n*)**  
Specifies the number of entries to be saved in the trace table and displayed by the TRACE and TRACE SOURCE commands. The default is 1000 and *n* is an integer between 1 and 32760.
- **EXTSTOR(*nM*|*nK*)**  
The extended storage parameter allows you to set the amount of extended storage that is accessed by CA InterTest Batch under the z/OS operating system. This storage is used when formatting data preceding the 16 MB line. The default amount is 64M. Specify the amount in megabytes (M) or kilobytes (K), and *n* is any integer between 1 and 100.
- **LINKPARM(*n*)**  
Specifies the maximum number of linkage parameters that can be set by the installation. The default is 32 and *n* is any integer between 1 and 1000. Each linkage parameter requires a page of memory (4096 bytes) unless the default has been changed.
- **LINKSIZE(*n*)**  
Specifies the amount of memory to be getmained for each linkage parameter. The default is 4096. It is not recommended to specify less than 1024 or more than 32768. The parameters are always on page boundaries.
- **NOSLOW**  
Turns off the SLOW command at the installation. The users are notified with a message stating this when attempting to use the SLOW command. Comment this parameter out if you do not want the SLOW command disabled.
- **SLOWFAST(*n*)**  
Specifies the duration of the wait time in hundredths of a second between executable statements when using the SLOWFAST command. The default is one quarter of a second (*n*=25). The value is any integer between 1 and 4096. This might be helpful to TCAM users with buffering problems.
- **NOSTAX**  
This is a required startup parameter.
- **NOESTAE**  
This is a required startup parameter.
- **NOSTAE**  
This is a required startup parameter.
- **NOESTAI**  
This is a required startup parameter.
- **NOJSCB**  
This is a required startup parameter.

## Configure Batch Link and Sysplex

### Define the Batch Link Scheduling Table

The batch link schedule feature helps you to leverage your DB2 stored procedure and IMS/DC testing environment by allowing you to specify exactly which of these programs you want CA InterTest Batch to monitor. To enable the batch link schedule feature, customize the SCHATBSIZE option in CAVHCONF. For more information, see [Options \(see page 31\)](#).

The SCHATBSIZE parameter of macro CAVHCONF defines the number of entries of an in-core table you want the product to maintain in CSA.

Set SCHATBSIZE to a reasonable value (for example SCHATBSIZE=30) to stay in your testing threshold. Each entry uses 27 bytes of CSA storage.

The maximum value for SCHATBSIZE is 60. Setting SCHATBSIZE to zero (SCHATBSIZE=0) disables the schedule function. When the schedule function is disabled, all DB2 SP and IMS/DC transactions run unmonitored.

The contents of the table are maintained through the product interface. For more information about maintaining this table, see [Batch Link Facility \(https://docops.ca.com/display/CAITSD11/Batch+Link+Facility\)](https://docops.ca.com/display/CAITSD11/Batch+Link+Facility).

If you want the product to load a fixed schedule at every IPL, insert the following DD card to the CAS9 initialization job stream:

```
//INT1SKUT DD DISP=SHR,DSN=xxxx
```

xxxx is a fully qualified data set containing schedule entries to be imported. This data set can be a sequential data set or a partitioned data set. If it is a partitioned data set, provide a member name in parenthesis. Create this data set using the EXPORT function. For more information about the EXPORT function, see [DB2 and IMS Schedule Menu \(https://docops.ca.com/display/CAITSD11/DB2+and+IMS+Schedule+Menu\)](https://docops.ca.com/display/CAITSD11/DB2+and+IMS+Schedule+Menu).

### Enable the SYSPLEX (Cross-LPAR) Debugging Feature

You can view and debug jobs that are executing on different (foreign) LPARs than the one you are currently logged on to. For more information about debugging jobs executing on foreign LPARs, see [Batch Link Selection Panel \(https://docops.ca.com/display/CAITSD11/Batch+Link+Selection+Panel\)](https://docops.ca.com/display/CAITSD11/Batch+Link+Selection+Panel).

#### Follow these steps:

1. Ensure that TCP/IP is active and accessible across the SYSPLEX, and that the CA Testing Tools Server is running on all LPARs within the SYSPLEX.
2. Specify the port number that you want to use for CA InterTest Batch SYSPLEX support. Update BLSPOPT="XXXXX" in CAVHUOPT member which is located in CAVHOPTN target library data set.
3. Specify IBSYSPLEX=Y in your CAVHCONF configuration module.
4. Restart the CA Testing Tools server if the server is already running.

## CICS Options

This article describes how to change the default installation options for CA InterTest for CICS and SymDump for CICS. All users must create an IN25OPTS load module using the following sample JCL. If you want to make changes to IN25OPTS after you have already installed your product, you will also perform the procedure to Refresh IN25OPTS.

### Configure IN25OPTS Options

CA InterTest for CICS and CA SymDump for CICS installation options are contained in the IN25OPTS load module. Create an IN25OPTS load module that contains the modified options with a batch assemble-and-linkedit job.

### Sample JCL to Assemble the Options Program

The following example shows the JCL needed to assemble the CA InterTest for CICS and SymDump for CICS Options Program (IN25OPTS). Member CABAOPTJ in CAI.CAVHJCL contains sample JCL that can be used to assemble and link IN25OPTS. In the following example, the options LETSVC, EXTSEC and CWALET have been generated:

```
//IN25OPTS JOB
//*****
//*      IN25OPTS SAMPLE JCL      *
//*****
//ASM      EXEC PGM=ASMA90,REGION=1M,
//          PARM='LIST,XREF(SHORT),DECK,N00BJ'
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=&&OBJECT,UNIT=SYSDA,DISP=(NEW,PASS),
//          DCB=BLKSIZE=3200,SPACE=(TRK,(1,1))
//SYSLIB   DD DSN=CAI.CAVHMAC,DISP=SHR
//          DD DSN=CICS.MACLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN    DD *
*** CWA USER DSECT REQUIRED HERE IF USING THE CWALET OPTION ***
      USING CWADS,0          CA INTERTEST REQUIREMENT
CWADS     DSECT              USER CWA
CWAFLD1   DS    CL20
CWAFLD2   DS    CL15
CWAFLD3   DS    CL100
***** CA INTERTEST AND SYMDUMP OPTION SPECIFICATIONS*****
      IN25OPTS TYPE=CSECT,          X
          LETSVC=NO,                X
          EXTSEC=Y
      IN25OPTS TYPE=CSECT,          X
          CWALET=(CWAFLD1,CWAFLD2+15)
      IN25OPTS TYPE=FINAL
      END
/*
//LKED EXEC PGM=IEWL,PARM=(XREF,LIST,MAP),
// REGION=1M,COND=(4,LT,ASM)
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=USER.LOADLIB(IN25OPTS),DISP=SHR
//SYSLIN DD DSN=&&OBJECT,DISP=(OLD,DELETE)
//          DD *
          ENTRY IN25OPTS
          NAME IN25OPTS(R)
//
```



**Note:** The CWALET option requires you to place your user CWA DSECT before the first IN25OPTS specification in the assembly. The USING statement for this DSECT is required. The register value for the USING statement can be any value from 0 to 15.

## Refresh IN25OPTS Options

If you have already installed and configured the product, you can follow this procedure to change the existing IN25OPTS options. Create a new copy of the IN25OPTS load module, and then take the following steps to refresh the options.

### Follow these steps:

1. Terminate CA InterTest for CICS and SymDump for CICS.
2. Update the IN25OPTS load module that contains the modified options with a batch assemble-and-linkedit job.
3. Run the following commands.

```
CEMT SET  PROG(IN25*)  NEWCOPY
CEMT SET  PROG(IN##*)  NEWCOPY
```



**Notes:** Replace ## with your two-digit CICS release number (66 for CICS 4.1, 67 for CICS 4.2, 68 for CICS 5.1, 69 for CICS 5.2, 70 for CICS 5.3, and 71 for CICS 5.4).

Ignore any failures of IN##HOOK and IN25INIT to newcopy.

4. Restart CA InterTest for CICS and SymDump for CICS.

## IN25OPTS Keywords

The following list describes the IN25OPTS keywords. Some keywords apply to either CA InterTest for CICS or CA SymDump for CICS, and some apply to both products.

If you change a transaction ID, also change the CICS transaction definition.

Defaults are underlined>. Slashes (/) between parameters indicate you must make a choice.

- [Common Keywords \(see page 58\)](#)
  - [BKPTERM \(see page 58\)](#)
  - [CORET \(see page 58\)](#)
  - [EXTSEC \(see page 59\)](#)
  - [HELPFIL \(see page 59\)](#)
  - [HELPT \(see page 59\)](#)
  - [IGLUT \(see page 59\)](#)
  - [LISTT \(see page 60\)](#)

- NDVRASML (see page 60)
- NDVRPROC (see page 60)
- SLBMARG (see page 60)
- SLBSCRL (see page 61)
- SLBTIT (see page 61)
- SYMDINT (see page 61)
- SYMFIL (see page 62)
- UCONLY (see page 62)
- USERHELP (see page 63)
- VRPTT (see page 63)
- XRMIO (see page 63)
- CA InterTest for CICS Keywords (see page 63)
  - AUDIT (see page 63)
  - AUTMON (see page 64)
  - BREAK (see page 64)
  - CICSPLEX (see page 64)
  - CKPTFIL (see page 65)
  - CNTLSEC (see page 65)
  - CNTLT (see page 65)
  - CNTOSEC (see page 65)
  - CNTSQLG (see page 66)
  - CORSEC (see page 66)
  - CPF11 (see page 67)
  - CPLXCKPI (see page 67)
  - CPLXFMID (see page 67)
  - CWALET (see page 67)
  - DFLTUSER (see page 68)
  - DL1SIZE (see page 69)
  - FDISP (see page 69)
  - FFORM (see page 69)
  - FILET (see page 70)
  - FILSEC (see page 70)
  - FSYMP (see page 70)
  - GLOG (see page 71)
  - ICMDBUF (see page 71)
  - ICMDEXE (see page 71)
  - ICMDMAX (see page 71)
  - IPINIT (see page 72)
  - IPLXT (see page 72)
  - IRCTT (see page 72)
  - ISERT (see page 73)
  - ITSTT (see page 73)
  - LETSVC (see page 73)

- MONOM (see page 73)
- MONOMSEC (see page 74)
- OSGETM (see page 74)
- PROMD (see page 74)
- READBUF (see page 75)
- RECNTMU (see page 75)
- RECNTNW (see page 75)
- SDISPLAY (see page 76)
- SLB (see page 76)
- SLBAKEEP (see page 76)
- STMTTRCE (see page 77)
- STUAR (see page 77)
- VERCOR (see page 77)
- VTATT (see page 78)
- CA SymDump for CICS Keywords (see page 78)
  - SYMDFIL (see page 78)
  - SYMDL (see page 78)
  - SYMDSVCD (see page 78)
  - SYMDT (see page 79)
  - SYMDWILD (see page 79)
  - TRCFDCLS (see page 79)
  - TRCFFMID (see page 79)
  - TRCFMCLS (see page 79)
  - TRCFMEGM (see page 79)
  - TRCFMEGT (see page 80)
  - TRCFSCLS (see page 80)
  - TRCFTHRD (see page 80)
  - TRCFTOUT (see page 80)
  - TRCFUNIT (see page 80)
  - TRCFVOL (see page 81)

## Common Keywords

### BKPTERM

**BKPTERM=no value /MOD2**

- **no value**  
The breakpoint display will not force a MOD2 terminal.
- **MOD2**  
The breakpoint display will always be treated as a MOD2 terminal.

### CORET

**CORET=CORE/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS CORE transaction. Be sure to adjust the CICS transaction definition if you do not use the default.



**Note:** The CICS restriction that no user transaction ID starts with the character C is not strict. You can use IDs such as CNTL and CORE if the CICS system does not use them.

## EXTSEC

### **EXTSEC=**YES/NO

Indicates whether External Security Manager is to be used.

- **YES**  
CA InterTest for CICS and CA SymDump for CICS call the External Security Manager to verify access to secured functions.
- **NO**  
CA InterTest for CICS and CA SymDump for CICS bypass the calls to the External Security Manager. When first installing CA InterTest for CICS and CA SymDump for CICS, this setting is useful until the product is tested and the security rules are written.

## HELPFIL

### **HELPFIL=**PROTHLF/filename

Specify the file name for the CA InterTest for CICS Help file. The name cannot exceed seven characters. Be sure to adjust the CICS file definition if you do not use the default.

If CICSplex=NO, the default, this task does not run.

## HELPT

### **HELPT=**HELP/transaction-id

Specify a four-character transaction ID for the CA InterTest for CICS HELP transaction. Be sure to adjust the CICS transaction definition if you do not use the default.

## IGLUT

### **IGLUT=**IGLU/transaction-id

Specify a four-character transaction ID for the CA InterTest for CICS IGLU transaction. This transaction displays the CA InterTest/CA SymDump for CICS global user exit list.



**Note:** Adjust the CICS transaction definition if you do not use the default.

## LISTT

### **LISTT=LIST/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS LIST transaction (viewing of the compile or assembly listing). Be sure to adjust the CICS transaction definition if you do not use the default.

## NDVRASML

### **NDVRASML=NO/YES**

Non LE assembler programs do not have a standard date/time stamp within the load module to compare against the date/time stamp of the program symbols in the PROTSYM file. Without a date /time stamp the symbols, verification fails and causes symbols for assembler programs to always be loaded by the CA Endeavor SCM dynamic symbolic load feature if the feature is active.

- **NO**  
Indicates you do not want symbols for non-LE assembler programs to be dynamically loaded on every request.
- **YES**  
Indicates you want symbols for non-LE assembler programs to be dynamically loaded on every request.

## NDVRPROC

### **NDVRPROC=INTNDVR/procedure name**

Specify the name of the dynamic symbolic support for CA Endeavor SCM procedure defined in the install procedure. This is the seven-character name prefix of the procedure, which starts when dynamic symbolic support is active and a dynamic symbols load is necessary.

## SLBMARG

### **SLBMARG=1/nn**

Specifies a decimal number, from 1 to 41, of the default margin value on the LIST facility screens for Assembler programs. Regardless of this setting, the Location field of each source listing line is shown. The margin value can be adjusted online at any time.

- **1**  
Presents the left-most portion of each source listing line.
- **41**  
Presents the right-most portion.
- **28**  
Suppresses the machine code.
- **34**  
Suppresses the machine code and statement numbers.

## SLBSCRL

### SLBSCRL=PAGE/HALF/STOP/nnnn

Specifies the default scroll value for the LIST facility (online viewing of compile and assembly listings) when the user presses the PF7 or PF8 keys. Regardless of this setting, the value can be adjusted online at any time.

- **PAGE**  
Scrolls the display one page.
- **HALF**  
Scrolls the display half a page.
- **STOP**  
Scrolls the display to the next declared breakpoint.
- **nnnn**  
Scrolls the display the specified number of lines (1 to 9999).

## SLBTIT

### SLBTIT=TITLES/NO/REGISTERS/KEEP/PROGRAM

Specifies the default view at the top of the display (info area) of the LIST facility (online viewing of compile and assembly listings). Regardless of this setting, the value can be adjusted online at any time.

- **TITLES**  
Displays the title and header lines, leaving less space for the source listing.
- **NO**  
Displays only the entry fields and a command line, leaving more space for the source listing.
- **REGISTERS**  
Displays the general registers and the condition code at a breakpoint.
- **KEEP**  
Displays your program data items at a breakpoint.
- **PROGRAM**  
Displays the load module name and the symbolic file information.

## SYMDINT

### SYMDINT=YES/NO

For CA SymDump for CICS users viewing a source listing, this option specifies whether the user can invoke CA InterTest for CICS functions for the abended program.

- **YES**  
CA SymDump for CICS users can invoke CA InterTest for CICS functions when viewing a source listing for an abended program. For example, users can set monitoring, invoke the FILE facility, and perform other CA InterTest for CICS functions directly from CA SymDump for CICS.
- **NO**  
CA SymDump for CICS users cannot invoke CA InterTest for CICS functions when viewing a source listing for an abended program. To invoke CA InterTest for CICS functions, users must access CA InterTest for CICS directly.

## SYMFIL

**SYMFIL=PROTSYM/file/(file1,file2,...)**

Specify the CICS file definition names for the CA InterTest for CICS symbolic files. The names cannot exceed seven characters. If more than one name is specified, the names must be separated by commas and the entire list enclosed in parentheses. Be sure to adjust the CICS file definition if you do not use the default.

This option tells CA InterTest for CICS the CICS file definitions of the CA InterTest for CICS symbolic files and their search order for program data. When monitoring is turned on for a program, CA InterTest for CICS searches the symbolic files in the sequence specified and for COBOL and PL/I programs finds the symbolic file member that matches the corresponding load module compile date /time. If a matching date/time cannot be found or a symbolic file member has a more recent date than the date in the load module, a symbolic file mismatch screen will appear and the customer must choose which symbolic file member to use for debugging. For Assembler programs, the symbolic file mismatch screen will always appear if a program is found on more than one symbolic file.

Once a matching date/time member is found or a member is selected from a mismatch screen, CA InterTest for CICS will continue to use the selected symbolic file member until all breakpoints and monitoring has been turned off for the program.

There is no upper limit to the number of symbolic files that you can specify. However, limit the number of files to reduce the overhead and the storage required for the VSAM buffers because each CICS file definition must be defined with LSRPOOL=NONE.

## UCONLY

**UCONLY=NO/YES**

Determines whether certain terminal output will be dynamically translated from lowercase to uppercase.

- **NO**  
Terminal output is not dynamically translated.
- **YES**  
Terminal output is dynamically translated to uppercase characters. If you previously installed the uppercase version of the CA InterTest for CICS load libraries because your terminals did not support the mixed case English character set, you should specify YES.

## USERHELP

### **USERHELP=PROTUHF/ filename**

Specify the file name for the optional help file of user-defined abend descriptions. The name cannot exceed seven characters. Be sure to adjust the CICS file definition if you do not use the default.

## VRPTT

### **VRPTT=VRPT/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS VRPT transaction used to verify the installation and display the installation options. Be sure to adjust the CICS transaction definition if you do not use the default.

## XRMIO

### **XRMIO=NO/YES**

For CA InterTest for CICS users, determines whether or not to use the CICS XRMIOOUT Global User Exit to keep track of the last SQL call issued by an application program.

- **NO**  
Allows capture of SQL information provided that an EXEC CICS command has not occurred between the SQL call and the abend.
- **YES**  
Provides the last SQL call to the applications programmer for the CORE=SQLCA and CORE=LASTSQL commands, but there is an overhead associated with the use of the XRMIOOUT exit that not all shops will want to incur.

## CA InterTest for CICS Keywords

### AUDIT

#### **AUDIT= NO/YES**

For auditing purposes, certain installations require that all CORE, FILE, and CNTL commands be recorded.

- **NO**  
Specify NO if you did not specify the GLOG= option or if you want to dynamically activate and deactivate global logging with CNTL=GLOG,ON and CNTL=GLOG,OFF commands.  
If you specify NO, the user can request logging of FILE commands by specifying FUNC=LOG on the FILE menu.
- **YES**  
Specify YES if you specified the GLOG= option and you need the global logging facility always active.  
With AUDIT=YES, neither the CNTL=GLOG,ON command nor the CNTL=GLOG,OFF command are accepted. The CNTL commands and CA InterTest for CICS responses are always recorded

in the transient data queue or data set that was specified with the GLOG= keyword. CORE changes to storage areas not owned by the task (usually password protected) are always recorded. The FILE transaction operates in the LOG=ON mode; that is, all user requests and resulting FILE screen images are recorded.



**Note:** If you do not specify the GLOG= keyword, the AUDIT=YES specification is ignored.

## AUTMON

### **AUTMON=**YES/NO

Indicates whether a new Monitor Table entry is created.

- **YES**  
Creates a new Monitor Table entry for a program when an automatic breakpoint is about to occur if the program is being monitored under a Monitor Table entry that does not name this program by its full name (for example, if the entry names a transaction). The new entry assumes all monitoring options currently in effect. This option makes it easier to debug the program.
- **NO**  
Does not create a separate Monitor Table entry.

## BREAK

### **BREAK=**YES/NO

Indicates whether CA InterTest for CICS is allowed to execute breakpoints.

- **YES**  
Allows CA InterTest for CICS to execute breakpoints; that is, to stop execution of the application program at certain points. Select this option if you intend to use CA InterTest for CICS as an interactive debugging tool.
- **NO**  
Does not allow CA InterTest for CICS to execute breakpoints. This option might be desirable in a separate version of CA InterTest for CICS used solely for monitoring production applications. CA InterTest for CICS generated with BREAK=NO monitors with less overhead, which is important in production.  
Instead of causing automatic breakpoints, CA InterTest for CICS abnormally terminates transactions with a transaction dump when it detects a standards violation. Theabend/dump code is INTE. The dump contains all of the usual diagnostic information.

## CICSplex

### **CICSplex=**YES/NO

CA InterTest for CICS provides special support for CICS regions participating in a CICSplex environment. Participating CICSplex regions have CA InterTest for CICS commands propagated across all regions in the CICSplex.

- **YES**  
Indicates that this CICS region is participating in at least one CICSplex family of regions, as named in the CPLXFMID keyword. Invokes CA InterTest for CICS CICSplex processing by using CAICCI. If CICSplex=YES, CAICCI processing must be installed on the z/OS operating system and available to CICS.
- **NO**  
Indicates that this CICS region is not participating in a CICSplex family.

## CKPTFIL

**CKPTFIL=PROTCPF/ filename**

Specify the file name for the CA InterTest for CICS checkpoint file. The name must not exceed seven characters. If you do not use the default, be sure to adjust the CICS file definition.

## CNTLSEC

**CNTLSEC=YES/NO**

Indicates whether certain CNTL commands can be protected with a password.

- **YES**  
Protects certain CNTL commands with a password and allows for password protection of certain CNTL options. Specify CNTLSEC=YES if the CNTL command functions should be available only to authorized personnel. With CNTLSEC=YES, the keyword CNTOSEC further specifies whether the subset of CNTL options should also be password protected.  
The passwords are defined in the IN25SEC2 macros. See [CA InterTest for CICS User Exits \(see page 104\)](#) for the list of CNTL commands and options subject to password protection and details on specifying passwords using IN25SEC2.  
With CNTLSEC=YES, you must use the IN25SEC2 macro to generate the CA InterTest for CICS program IN25SEC2 or use the IN25SEC2 delivered with CA InterTest for CICS.
- **NO**  
Does not protect certain CNTL commands with passwords and does not allow for password protection of certain CNTL options.

## CNTLT

**CNTLT= CNTL/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS CNTL transaction. Be sure to adjust the CICS transaction definition if you do not use the default.



**Note:** The CICS restriction that no user transaction ID begins with the character C is not strict. IDs such as CNTL and CORE can be used if the CICS system does not use them.

## CNTOSEC

**CNTOSEC=YES/NO**

When the value for the keyword CNTLSEC is YES, specify this keyword to include or exclude certain CNTL options from password protection. When CNTLSEC=NO, this keyword is ignored.

- **YES**  
Protects certain CNTL options with a password. The password is defined in an IN25SEC2 macro, as discussed in [Password Protection \(see page 107\)](#).
- **NO**  
Does not protect any CNTL options with passwords.

## CNTSQLG

### CNTSQLG=YES/NO

Indicates whether CNTL commands are logged.

- **YES**  
Logs CNTL commands executed from a sequential terminal and responses to them into the transient data destination defined in the GLOG option.
- **NO**  
Does not log CNTL commands.

## CORSEC

### CORSEC=YES/NO

Indicates whether a password is required to modify a main storage area protected by CA InterTest for CICS.

- **YES**  
Requires a password when a user of the CORE facility is about to modify a main storage area that is protected by CA InterTest for CICS, but does not require the password during normal debugging. Specify CORSEC=YES if you will use the CA InterTest IN25SEC2 macro to generate a password that would prevent the CORE facility from being used by unauthorized personnel. The IN25SEC2 macro must be used to generate the CA InterTest for CICS program IN25SEC2 or use the IN25SEC2 delivered with CA InterTest for CICS. The use of the IN25SEC2 macro is described in [Password Protection \(see page 107\)](#).  
Without the password, the CORE transaction lets you modify only those storage areas that belong to a task that is waiting in a breakpoint and are not protected by CA InterTest for CICS from being modified by the monitored program.
- **NO**  
Does not require a password before allowing modification of any field within the CICS region in main storage.  
See the description of the CORE transaction in [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging). The CORSEC specification also applies to data modification in the LIST Keep window.



**Note:** Regardless of the CORSEC= specification, the following main storage areas are not available for modification by the CORE transaction:

- CICS/VS Sign-on Table (DFHSNT)
- Main storage outside of the CICS region

## CPF11

### CPF11=7B/BF

Activates the PF11 key on the Detailed Breakpoint screen.

- **7B**  
Allows application programmers to use the PF11 key on the Detailed Breakpoint screen to tell CA InterTest for CICS to force the execution of any error condition that CA InterTest for CICS has detected even though the error condition that caused the automatic breakpoint still exists. In effect, the user tells CA InterTest for CICS that this is really not a violation and to execute regardless of the consequences.  
Before enabling this facility, be aware of the following considerations:
  - As a result of this execution, not only the tested transaction's data could be damaged but other concurrent transactions could also be affected and the entire CICS system may crash.
  - Using PF11 ignores not only the immediate error condition, but also all subsequent error conditions until the next breakpoint or CICS Service Request (whichever comes first).
- **BF**  
Specify CPF11=BF if you do not want to provide this facility.

## CPLXCKPI

### CPLXCKPI=0020/hhmm

If CICSplex=YES and the checkpoint facility is started by a CICSplex resynchronization request, defines the time interval in hours (HH) and minutes (MM) between checkpoints.

## CPLXFMID

### CPLXFMID=INTERTST/CICSplex-familyname

Defines an eight-character CICSplex family name associated with the CICS region. The same family name must be given to all CICS regions participating in that family to obtain CICSplex support throughout the family. CA InterTest for CICS supports multiple CICSplex families within or across multiple z/OS regions.

For example, all regions associated with CPLXFMID=REGIONA participate in one CICSplex family, while all regions associated with CPLXFMID=REGIONB participate in another.

## CWALET

### CWALET=<no value>/(label,label)

Specify the CWALET keyword only if you want to remove CA InterTest for CICS storage protection from certain CWA fields.



**Important!** If you choose to specify one or more CWALET areas, you must code a *separate* IN25OPTS TYPE=CSECT macro for each CWALET area (one area per macro).

IN25OPTS macros with CWALET must follow other IN25OPTS macros except for the TYPE=FINAL macro, which must be the last one.

- **no value**

CA InterTest for CICS protection will not be removed. Use the CSA= and LET= online options to let specific monitored programs modify additional areas of main storage. See [CICS Debugging](https://docops.ca.com/display/CAITSD11/CICS+Debugging) (<https://docops.ca.com/display/CAITSD11/CICS+Debugging>) for details about online options.

- **(label,label)**

Specify two Assembler labels, separated by a comma and enclosed in parentheses; for example, CWALET=(CSAUTA1,CSAUTA4). The first label defines the first byte of the area to be unprotected; the second label defines the first byte past the area.

When using labels, they must be defined in the DSECT for the CWA and inserted in the IN25OPTS compile; see the section Sample JCL to Assemble the Options Program for details.

If your CWA has an existing DSECT, use existing labels and offsets from the labels to define the unprotected area.

If you do not have an existing DSECT in your CWA, add the following two statements to the IN25OPTS module to create one:

```
***** CWA USER DSECT REQUIRED HERE *****
USING CWADS,0   CA INTEREST REQUIREMENT
CWADS          DSECT                USER CWA
***** CA INTEREST SPECIFICATIONS *****
```

You can then specify offsets from the beginning of the CWA. For example, the following macro unprotects a 20-byte area starting at the beginning of the CWA:

```
IN25OPTS TYPE=CSECT ,CWALET=(CWADS ,CWADS+20)
```

## DFLTUSER

### DFLTUSER=.ANY/SPECIFIC/BACKGROUND

DFLTUSER stands for default user and controls how CA InterTest for CICS monitoring functions by default. A user can override the default settings established by this parameter at any time.

- **.ANY**

When this setting is used, CA InterTest for CICS monitoring pays no attention to the CICS user ID of the session, even if the user is signed on to CICS. Monitoring is set for all users, and breakpoints, by default, are directed to the same terminal used to set them.

- **SPECIFIC**

This setting is only valid in a secure CA InterTest for CICS region where the users log on to CICS. When DFLTUSER=SPECIFIC is used, CA InterTest for CICS qualifies each monitoring entry with the user's CICS user ID. In addition, CA InterTest for CICS directs all options to monitoring to .ANY terminal. This setting allows many users to test the same program or transaction independently. This also allows users to move from terminal to terminal and have their monitoring entries and monitoring options follow them.

- **BACKGROUND**

This setting is only valid in a secure CA InterTest for CICS region where the users log on to CICS. Code this if your CICS system is used predominantly as an Application Server, where most transactions are not attached to terminals (that is, web interface transactions). This setting sets the FROM terminal to .ANY, and the breakpoint (TO) terminal to the current terminal that is setting the breakpoints.



**Important!** When DFLTUSER=SPECIFIC is used and a user does not sign on to the CICS region, CA InterTest for CICS uses the value of the CICS DFLTUSER for the monitoring entry. The CICS default user ID is optionally specified in the SIT parameters. If not specified in the SIT, its value is CICSUSER.

## DL1SIZE

**DL1SIZE=8192/nnnnn**

Specifies the number of bytes in the largest DL/I segment to be retrieved using the CA InterTest for CICS FILE facility. Storage corruption may occur if the size specified here is smaller than a DL/I segment that is retrieved using FILE.

## FDISP

**FDISP=FC/DL/TD/TS/DB**

Determines the default menu for the FILE transaction. The various menu types are described in [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging). The user can change the menu of the FILE facility online at any time.

- **FC**

The FILE transaction shows its first display in the DATATYPE=FC format; that is, ready to process CICS files defined in the CICS file definition (File Control Table).

- **DL**

The FILE transaction shows its first display in the DATATYPE=DL format; that is, ready to process DL/I segments.

- **TD**

The FILE transaction shows its first display in the DATATYPE=TD format; that is, ready to process CICS transient data queues.

- **TS**

The FILE transaction shows its first display in the DATATYPE=TS format; that is, ready to process CICS temporary storage.

- **DB**

The FILE transaction shows its first display in the DATATYPE=DB format; that is, ready to process DB2 data.

## FFORM

**FFORM=D/C/V/S**

Determines the default format in which the FILE transaction displays the data. The user can change the format of the FILE display online at any time. For illustrations of FILE displays, see [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging).

- **D**  
The FILE transaction shows its first display in dump format, similar to the nonstructured display of the CORE transaction. Shows both character and hexadecimal images.
- **C**  
The FILE transaction shows its first display in character format. This format displays the largest amount of data, but is useless for noncharacter data records.
- **V**  
The FILE transaction shows its first display in vertical format, similar to the DITTO utility. This format displays the least amount of data.
- **S**  
The FILE transaction shows its first display in structured format if symbolic information for the program containing the structures is stored in the CA InterTest for CICS symbolic file. COBOL and PL/I structures use 01 level names; Assembler structures use DSECT names. Shows both character and hexadecimal images. See the FSYPM option.

## FILET

**FILET=FILE/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS FILE transaction. Be sure to adjust the CICS transaction definition if you do not use the default.

## FILSEC

**FILSEC=YES/NO**

Indicates whether the CA InterTest IN25SEC2 macro generates passwords for security purposes.

- **YES**  
The CA InterTest IN25SEC2 macro generates passwords to prevent the FILE transaction from being used by unauthorized personnel. More comprehensive security enforcement has to be implemented on the CICS level or on the operating system level.  
With FILSEC=YES the IN25SEC2 macro must be used to generate the CA InterTest for CICS IN25SEC2 program, or use the IN25SEC2 delivered with CA InterTest for CICS. Failure to do so will make the FILE transaction useless. Using the IN25SEC2 macro is described in [Password Protection \(see page 107\)](#).
- **NO**  
Permits users of the FILE transaction to access any CICS file or DL/I data without a CA InterTest for CICS-enforced password.

## FSYMP

**FSYMP=PROTFILE/symbolic-programname**

Specify the name of the default dummy COBOL or PL/I program that contains all of the 01 level structures used at the site or the Assembler program that contains all the DSECTs. Symbolic information for this program must be stored in the CA InterTest for CICS symbolic file.

Users of the FILE transaction requesting records or DL/I segments in structured format must specify the symbolic name of the program that contains the structure. However, if a default program containing all of the structures has been compiled or assembled, users can omit the symbolic program name.

For a sample default COBOL program, see [Add COBOL File Structures to the Symbolic File \(see page \)](#). For more information on adding Assembler file structures, see [Add Assembler DSECTs to the Symbolic File \(see page \)](#).

## GLOG

### **GLOG=CSSL/DCT-destination-id**

Specify a four-character name of a transient data destination in the CICS Destination Control Table (DCT). The default, GLOG=CSSL, provides the statistics destination that is always present in the CICS system and is usually printed when the CICS system is terminated.

The destination must specify a direct or indirect extra partition transient data set that has a record size of at least 110 bytes (variable-length format, including the length field).

CA InterTest for CICS writes records to the specified transient data queue (destination) after the command CNTL=GLOG,ON is entered. See the description of the AUDIT keyword; also see the description of the CA InterTest for CICS global logging facility in [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging).

## ICMDBUF

### **ICMDBUF=1024/nnnnn**

Specifies the indirect command buffer size, which is the number of bytes used by CA InterTest for CICS when acquiring and allocating Indirect Command Input Work Buffers. These buffers are used by the Indirect Command Processing Program, which displays and executes indirect commands specified by the user for specific programs. The minimum buffer size is 80. The maximum buffer size is 32767.

## ICMDEXE

### **ICMDEXE=50/nnnnn**

Specifies the maximum indirect command execution count. When the maximum number of indirect commands executed exceeds the ICMDEXE count, message CAIN1266 is generated. The minimum count allowed is 1. The maximum count allowed is 1024.

## ICMDMAX

### **ICMDMAX=50/nnnnn**

Specifies the maximum indirect command statements count. When the number of indirect command statements exceeds the ICMDMAX count, message CAIN1259 is generated. The minimum count allowed is 1. The maximum count allowed is 1024.

## IPINIT

### IPINIT=YES/NO

Indicates whether support should be initialized for the common user interface.

- **YES**  
Indicates that the common user interface mainframe initialization should occur.
- **NO**  
Indicates that the common user interface will not be used and the CUI initialization should not be performed.

## IPLXT

### IPLXT=IPLX/transaction-id

Specifies a four-character transaction ID for the CA InterTest for CICSplex CCI. Specifies a four-character transaction ID for the CICSplex CCI (Common Communications Interface) to be used for CICSplex support by CA InterTest for CICS.



**Note:** Adjust the CICS transaction definition if you do not use the default.

## IRCTT

### IRCTT=VIRC/transaction-id

Specify a four-character transaction ID for the CA InterTest for CICS VIRC transaction. This is an internal CA InterTest for CICS transaction used only for handling breakpoints. Its use must not be protected by a specific signon. If started by mistake from a terminal, the transaction program does nothing.

In an MRO or ISC environment, the VIRC transaction must be installed as local in all Terminal Owning Regions and Application Owning Regions that use CA InterTest for CICS.

In a CICS environment that does not use MRO or ISC, the VIRC transaction must be installed to support automatic single-stepping.



**Note:** If you change the transaction ID, also change the CICS transaction definition.

## ISERT

**ISERT=ISER/transaction-id**

Specify a four-character transaction ID for the internal CA InterTest for CICS ISER transaction. Do not assign protection to a specific user.



**Note:** If you change the transaction ID, also change the CICS transaction definition.

## ITSTT

**ITSTT=ITST/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS ITST transaction. This transaction displays the CA InterTest for CICS Primary Option menu from a CICS prompt. Do not assign protection to a specific user.



**Note:** If you change the transaction ID, also change the CICS transaction definition.

## LETSVC

**LETSVC=YES/NO**

Indicates whether SVCs are permitted.

- **YES**  
Permits any SVCs in application programs except:

SVC 13 (hex 0D) -- ABEND macro

- **NO**  
Indicates that application programs do not use any SVCs except:

SVC 11 (hex 0B) -- TIME macro

These are the SVCs often used implicitly in applications, but any other SVCs should not be used in CICS programming.

Regardless of this keyword, you can use BYP= options during a CA InterTest for CICS test session to permit the monitored program to use SVCs.

## MONOM

**MONOM=NO/MENU/NOMENU**

Controls the activation of the segmented monitoring feature.

- **NO**  
Deactivates the segmented monitoring feature and removes any previous MON and NOM entries from the CNTL main menu.
- **MENU**  
Activates the feature and displays the MON and NOM options on the CNTL main menu.
- **NOMENU**  
Activates the segmented monitoring feature, but limits users to implementing segmented monitoring through CNTL commands or the LIST facility. NOMENU disables the use of the MON and NOM options on the Monitoring menus, and removes the MON and NOM entries from the CNTL main menu.

## MONOMSEC

### MONOMSEC=YES/NO

Controls password protection for the segmented monitoring options in the CNTL facility. If MONOM=NO, MONOMSEC is ignored.

- **YES**  
Implements password protection of the segmented monitoring feature. The password is controlled by a keyword in the IN25SEC2 macro. Instructions for specifying passwords for CNTL options are in [Password Protection \(see page 107\)](#).
- **NO**  
Permits the use of the segmented monitoring feature without password protection.

## OSGETM

### OSGETM=YES/NO

Indicates whether a GETMAIN SVC is allowed to be issued.

- **YES**  
Permits COBOL programs monitored by CA InterTest to issue the GETMAIN SVC. As suggested by the *CICS Application Programmer's Reference Manual*, any CICS COBOL program should be coded so that no COBOL facilities require a GETMAIN SVC. Such a GETMAIN SVC obtains storage from the OSCORE storage pool outside the storage controlled by the CICS Storage Management. There is no FREEMAIN if the COBOL program terminates. When, eventually, a GETMAIN cannot be satisfied, the CICS system will be abended by the operating system. OSGETM=YES is implied by the LETSVC=YES specification.
- **NO**  
Does not permit the GETMAIN SVC to be issued by COBOL programs. Unless you run software that issues the GETMAIN SVC, we recommend that you specify OSGETM=NO as GETMAINs should not be permitted. With this specification, an automatic breakpoint will occur when the GETMAIN SVC is detected. However, OSGETM=NO is overridden by LETSVC=YES.

## PROMD

### PROMD=00000020/name/number

The name or the number in this keyword is used for the default value of the PROM= parameter of the CNTL=START command.

- **name**  
Specify the CA InterTest IN25SEC2 macro generates passwords the name (up to eight characters) of the CA InterTest for CICS Startup Table that has to be generated with the PROMMAC macro. The use of the macro is described in [Activate Your Product \(see page 178\)](#).
- **number**  
Specify the number to be used as the default value of the PROM= parameter of the CNTL=START command, which specifies the initial number of entries in the Monitoring Table. This number must be eight characters long. For the initial period of use, the default value should suffice. The number of entries in the table should be no less than the number of names to be declared for monitoring. However, for efficiency, allow enough entries at the beginning by specifying the PROM= parameter here or in the CNTL=START command. A large number of unoccupied entries use more storage but do not increase the overhead because only occupied table entries are examined when the table is scanned.

## READBUF

### READBUF=YES/NO

Indicates whether the read buffer function is supported.

- **YES**  
Before sending a breakpoint to a terminal, CA InterTest for CICS issues a read buffer to save the last user screen from that terminal. If your CA InterTest for CICS demonstration session runs successfully, specify YES.
- **NO**  
You must specify NO if all terminals that would be used by CA InterTest for CICS users run under TCAM or do not support the read buffer function. If you do not specify NO, you must declare the NRB=ON online option whenever monitoring is declared.

## RECNTMU

### RECNTMU=YES/NO

Indicates whether CNTL menu generated commands are displayed before execution.

- **YES**  
Displays the CNTL commands generated by the CA InterTest for CICS menus just before the commands are executed. The user can learn these commands from the screen and, for efficiency, use them instead of menus.
- **NO**  
Suppresses display of CNTL commands.

## RECNTNW

### RECNTNW=YES/NO

Indicates whether CNTL RESTART or NEWCOPY commands are displayed before execution.

- **YES**  
Displays the CNTL commands that are automatically generated by CNTL=RESTART or CNTL=NEW, PROG= commands. These appear just before they are executed so that the user can see what is being done.
- **NO**  
Suppresses display of CNTL commands. Specify NO if users are not interested in these details.

## SDISPLAY

### **SDISPLAY=HEX/DATA**

Facility of the Structure display and AutoKeep display feature. Specifies the global default value for Structure Display Format. Regardless of this setting, the value can be adjusted online by program name.

- **HEX**  
Display data in hexadecimal character format
- **DATA**  
Display data in Structure Display Format

## SLB

### **SLB=YES/NO**

Indicates whether Source Listing Breakpoint displays are available.

- **YES**  
Uses Source Listing Breakpoint displays for the terminal from which the program was declared for monitoring if the compile (assembly) listing is available in the CA InterTest for CICS symbolic file.
- **NO**  
Uses Detailed Breakpoint displays when a program is declared for monitoring.

In either case, Source Listing Breakpoint displays can be turned on or off during the online test session.



**Note:** The CA InterTest for CICS LIST facility (the online viewing of compile and assembly listings before program execution) is not affected by this setting.

## SLBAKEEP

### **SLBAKEEP=YES/NO**

Specifies the default value for the LIST facility of the AutoKeep display feature. Regardless of this setting, the value can be adjusted online at any time.

- **YES**  
Displays items in the Keep window concerning the currently highlighted line and any data items you have requested to be displayed in the Keep Window.
- **NO**  
Displays only the data items you have requested to be displayed in the Keep Window.

## STMTRCE

**STMTRCE=500/nnnnn**

Specify the number of entries in the statement trace table. This table will be allocated to any transaction which executes one or more COBOL programs for which the TRACE option is active. The maximum value is 32767.

## STUAR

**STUAR=NO/YES**

Affects breakpoint processing in regard to the TCTTE user area (an area embedded in the TCTTE, whose address is in the field labeled TCTTECIA).

- **NO**  
The contents of the area are not saved for the duration of the breakpoint or restored when the task continues execution from the breakpoint. There is no need to preserve the data in the area if the breakpoint display is to be sent to another terminal, or if the area is to be modified (during the breakpoint) only by the person performing the test. Because command level (EXEC CICS) applications rarely use this area, STUAR=NO should be appropriate.  
Also specify STUAR=NO if the area may be affected during the breakpoint by a facility other than the tested application.
- **YES**  
The contents of the area are preserved for the duration of the breakpoint. Specify STUAR=YES if the logic of the tested application depends on saving this area while your task waits at a breakpoint, and then restoring it to its previous value, possibly with your modifications.  
STUAR=YES is desirable if the area can be used during the breakpoint and use of the area must not affect the tested program.

## VERCOR

**VERCOR=NO/YES**

Determines whether changes to main storage must be verified.

- **NO**  
Changes to main storage requested in a CORE transaction command can, but need not be, verified in the same command.  
If CORSEC=YES is specified, the user cannot modify any foreign piece of storage by mistake because the CORE transaction will ask for the password. As a result, you can specify VERCOR=NO because a careless user can harm only the tested program's data.

- **YES**

Any change to main storage requested in a CORE transaction command must be verified in the same command regardless of password protection (see the CORSEC= keyword). This compulsory verification could make a debugging session unnecessarily tedious, thus decreasing productivity. The system programmer will find this specification a hindrance, especially in situations when a certain value must be forced upon an area whose content continually changes.

For more information, see [Accessing Main Storage CORE \(https://docops.ca.com/display/CAITSD11/Accessing+Main+Storage+CORE\)](https://docops.ca.com/display/CAITSD11/Accessing+Main+Storage+CORE).

## VTATT

### **VTATT=VTAT/transaction-id**

Specify a four-character transaction ID for the CA InterTest for CICS VTAT transaction. This is an internal CA InterTest for CICS transaction used only for handling breakpoints. Since it is not used by any operator, its use must not be protected by a specific user. If started by mistake from a terminal, the program does nothing. Be sure to adjust the CICS transaction definition if you do not use the default.

## CA SymDump for CICS Keywords

### SYMDFIL

#### **SYMDFIL=PROTDMP/filename**

For CA SymDump for CICS users, specify the file name for the CA SymDump for CICS (the Symbolic Dump option) file. This is the name of the file from which dumps can be viewed; CA SymDump for CICS always writes to the file PROTDMP. The name cannot exceed seven characters. Be sure to adjust the CICS file definition if you do not use the default.

### SYMDL

#### **SYMDL=SYMD/transaction-id**

For CA SymDump for CICS users, specify a four-character transaction ID for the CA SymDump for CICS transaction for the capture for the last SQL statement in background during dump capture. Be sure to adjust the CICS transaction definition if you do not use the default. If no value is specified for this option, then no transaction will be scheduled.

### SYMDSVCD

#### **SYMDSVCD=YES/NO**

When unexpected events occur, such as task purging or catastrophic application failures, CA SymDump for CICS may have to abbreviate its dump capture. When this occurs, CA SymDump for CICS may produce an SVC diagnostic dump that could assist in researching whatever issues have transpired.

- **Yes**

(Default) CA SymDump for CICS may produce an SVC diagnostic dump if dump capture processing encounters unexpected conditions.

- **No**  
CA SymDump for CICS will not produce an SVC diagnostic dump.

## SYMDT

**SYMDT=SYMD/transaction-id**

For CA SymDump for CICS users, specify a four-character transaction ID for the CA SymDump for CICS transaction that allows dumps to be viewed online. Be sure to adjust the CICS transaction definition if you do not use the default.

## SYMDWILD

**SYMDWILD=**

Specify a wild card prefix for the Dump File ID that is specified on the DUMP/TRACE ANALYSIS screen. This limits what can be specified as a wildcard. For example, coding SYMDWILD=PROT lets you specify file names with a wildcard prefix of PROT in the Dump File ID input field of the CA SymDump Analysis screen. In this case, PROT\* would be valid input but PRO\* would be in error.

## TRCFDCLS

**TRCFDCLS=**

For CA SymDump for CICS users, specify an allocation DATA CLASS that the TRACE FORMAT REGION will use to allocate storage for trace formatting. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## TRCFFMID

**TRCFFMID=SYMDTRCF/8 character unique id**

For CA SymDump for CICS users, specify the unique TRACE FAMILY ID that a CICS region will use to locate and form an affinity with a TRACE FORMAT REGION that has an IN25OPTS with the same TRCFFMID. This parameter is used by both the CICS region and the TRACE FORMAT REGION.

## TRCFMCLS

**TRCFMCLS=**

For CA SymDump for CICS users, specify an allocation MANAGEMENT CLASS that the TRACE FORMAT REGION will use to allocate storage for trace formatting. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## TRCFMEGM

**TRCFMEGM =120/ total number of megs**

For CA SymDump for CICS users, specify the maximum size of extended private area storage in megabytes that the TRACE FORMAT REGION will allocate for use by all threads. This amount of extended private must be available to the TRACE FORMAT REGION when it starts, or the job will terminate. If this number is exceeded when a trace is selected for formatting, the user is invited to try again later, this allows performance tuning of the formatted trace facility. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## TRCFMEGT

**TRCFMEGT=15/megs per thread**

For CA SymDump for CICS users, specify the maximum size of extended private area storage in megabytes that the TRACE FORMAT REGION will allocate for use by one thread. If this is not available when a trace is selected for formatting, the user is invited to try again later. This will allow performance tuning of the formatted trace facility. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

For more information, see [Performance Considerations \(see page 175\)](#).

## TRCFSCLS

**TRCFSCLS=**

For CA SymDump for CICS users, specify an allocation STORAGE CLASS that the TRACE FORMAT REGION will use to allocate storage for trace formatting. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## TRCFTHRD

**TRCFTHRD=8/ number of threads**

For CA SymDump for CICS users, specify the number of simultaneous threads that will be supported for trace format requests. This parameter should be used as a rudimentary throttle to allow performance tuning of the formatted trace facility. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## TRCFOUT

**TRCFOUT=240/ trace timeout in seconds**

For CA SymDump for CICS users, specify trace timeout value in seconds that both the TRACE FORMAT REGION and the CICS task will wait on CCI calls used to communicate between the CICS task and the TRACE FORMAT REGION. When this value is exceeded, the CCI session is terminated, and the formatted trace resources are freed. The operator remains within the reloaded dump, and is given the option of reformatting the trace. This will allow performance tuning of the formatted trace facility. This parameter is used by both the CICS region and the TRACE FORMAT REGION.

## TRCFUNIT

**TRCFUNIT=**

For CA SymDump for CICS users, specify an allocation UNIT that the TRACE FORMAT REGION will use to allocate storage for trace formatting. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## TRCFVOL

### TRCFVOL=

For CA SymDump for CICS users, specify an allocation VOLUME that the TRACE FORMAT REGION will use to allocate storage for trace formatting. This parameter is used only by the TRACE FORMAT REGION and is ignored by a CICS region.

## CA InterTest for CICS Options

To complete the configuration for your product, follow the steps described in this article.

- [Configure Abend Codes \(see page 81\)](#)
- [Customize Help Abend Codes \(see page 82\)](#)

## Configure Abend Codes

This section describes how to limit the interception of abends and customize abend code help text.

### Specify Non-Intercepted Abend Codes

If certain abend codes should not be intercepted regardless of the CA InterTest for CICS Automatic Breakpoint (ABI) setting, use the IN25ABEN macro to specify the codes. CA InterTest for CICS provides a default IN25ABEN that includes the abend codes AKC3, ATNI, ATCV, and AEXY.



**Note:** AKC3 is needed to run CA InterTest for CICS with CA Verify for CICS.

To specify your own codes in the IN25ABEN macro, customize the sample member CCSAABEN in CAI.CAVHJCL.

The following example demonstrates the IN25ABEN macro usage:

```
//ASM.SYSIN DD *
  IN25ABEN TYPE=INITIAL
  IN25ABEN ABCODE=( ATNI , ATCV , AEXY )
  IN25ABEN ABCODE=AKC3
  IN25ABEN ABCODE=( AKCT , APLE )
  IN25ABEN ABCODE=( 1234 , '# Q$' , 1UMF )
  IN25ABEN TYPE=FINAL
END
/*
```

**IN25ABEN Coding Rules** -- Follow these rules when coding IN25ABEN:

- IN25ABEN TYPE=INITIAL must be coded first.

- The ABCODE= parameters must specify four-character abend codes. The abend codes ATNI, ATCV, and AEXY must be included. Include the code AKC3 if you are running CA InterTest with CA Verify for CICS.
- If ABCODE specifies a list of abend codes, then separate the codes with commas and enclose the list in parentheses.
- Codes that include spaces or special characters must be enclosed in single quotes.
- IN25ABEN TYPE=FINAL terminates the table.
- END terminates the assembly source.

After coding the macro, submit the JCL to assemble your options.

## Customize Help Abend Codes

You can customize the help text displayed at an automatic breakpoint for abend codes. A separate help file, PROTUFH, is available to hold your user-defined abend code descriptions.

After installation, the online Abend Code Maintenance menu (ITST Option 7.2) lets you add, inquire, update, and delete entries in the user-defined help file. Press PF1 from this menu to obtain help on using the facility.

```

CA InterTest for CICS ABEND CODE MAINTENANCE MENU

OPTION ==>

  Enter Option, Abend code, and Program name; then press enter.
  1 Add                - Add an Abend code/Program name
  2 Inquire            - Inquire on an Abend code/Program name
  3 Update             - Update an Abend code/Program name
  4 Delete             - Delete an Abend code/Program name

  Abend code . . ____ Program name . . _____

PF1 Help   2          3 End   4 Return  5          6
PF7        8          9      10      11        12

```

You can tie an abend code to a specific program by entering both the abend code and program name fields. You can create a generic abend code for all program names by entering the abend code only, with no program name.

During a user session when CA InterTest for CICS intercepts an abend, it first searches for a matching abend code entry on the user-defined help file. If a description for the abend code is found on the user-help file, CA InterTest for CICS displays its contents at the breakpoint and on related CA SymDump for CICS analysis displays.

If there is no user-defined description or if the user file does not exist, then the abend code description from the CA InterTest for CICS help file is used. If there is no abend code description in the CA InterTest for CICS help file, then a default NOTFABCODE is displayed.

## CA SymDump for CICS Options

This article describes CA SymDump for CICS options.

- [Procedure for Modifying Dump Capture Criteria \(see page 83\)](#)
- [IN25CAPT Keywords \(see page 84\)](#)
- [How Macro IN25CAPT Controls Dump Capture \(see page 85\)](#)

You can choose to selectively override the CA SymDump for CICS dump suppression criteria, by identifying dumps that must always be captured.

The IN25CAPT load module contains the criteria for forcing dump capture.

You can use the following steps to create a customized copy of IN25CAPT.

**Follow these steps:**

1. Modify member CABQCAPJ of CAVHJCL and specify the abend codes, programs and transactions that identify the dumps you intend to capture.
2. Assemble and link your customized code to create load module IN25CAPT.
3. Stop CA SymDump, if active.



**Note:** If CA Symdump is active when you issued the NEWCOPY, the module does not get replaced.

4. Issue the following CICS command to load the new copy of IN25CAPT  
CEMT SET PROG(IN25CAPT) NEWCOPY



**Note:** You can skip this step the first time you use IN25CAPT.

5. Restart CA SymDump for CICS.

## Procedure for Modifying Dump Capture Criteria

To create module IN25CAPT and specify the dump capture options, use macro IN25CAPT to specify the abend codes, programs and transactions, using any combination of exact and generic values.

Invoke the macro once to initialize the Dump Capture Table, once for each entry you require and once to terminate the table.

Following tasks are performed:

1. Each table entry identifies, either specifically or generically, the combination of abend code, program name and transaction name that describe abends that must always be captured. This task occurs regardless of the CA SymDump for CICS options that suppress duplicate dumps, or suppress dumps by abend code. Generic values are specified with a trailing asterisk (\*).

2. A dump gets captured when an abend matches at least 1 of the entries in IN25CAPT. To match a table entry, an abend must match all three specified criteria (Abend code, Program and Transaction). Even if one of the criteria is omitted, or a value of asterisk (e.g., PROG=\*) is specified then the status indicates that all abends match that criteria.

## IN25CAPT Keywords

This section lists the IN25CAPT keywords. Slashes (/) between parameters indicate you must make a choice.

CA SymDump for CICS does not validate the values specified for abend codes, transaction names or program names. Specifying undefined or invalid values results in no match, and hence, there is no change in dump handling.

- **TYPE=CSECT/ENTRY/FINAL**  
Describes the type of table entry.
  - **CSECT**  
Initializes the table. Must be the first entry in the table and can only occur once.
  - **ENTRY**  
Creates a single table entry, specifying one combination of abend codes, transactions and programs that identify abends that should be captured
  - **FINAL**  
Terminates the table. Must be the last entry in the table and can only occur once.
- **ABEND=code identifier**  
Identifies the abend code(s) described by this entry.
  - **code identifier**  
1-4 characters, specifying the abend code(s) to be captured. If the final character is asterisk (\*), abend codes that begin with the characters preceding the asterisk are considered a match. ABEND=\* indicates that all abend codes are considered a match.
- **PROG=program name**  
Identifies the program(s) described by this entry.
  - **program name**  
1-8 characters, specifying the abending programs to be captured. If the final character is asterisk (\*), programs that begin with the characters preceding the asterisk are considered a match. PROG=\* indicates that all program names are considered a match.
- **TRAN=transaction name**  
Identifies the transaction(s) described by this entry.
  - **transaction name**  
1-4 characters, specifying the transactions(s) to be captured. If the final character is asterisk (\*), transactions that begin with the characters preceding the asterisk are considered a match. TRAN=\* indicates that all transaction names are considered a match.

## How Macro IN25CAPT Controls Dump Capture

The section illustrates the effect of specifying certain values in the Capture option.

Assuming your customized IN25CAPT was assembled with the following options:

```
IN25CAPT TYPE=CSECT
IN25CAPT TYPE=ENTRY,TRAN=ABCD,PROG=PGM1
IN25CAPT TYPE=ENTRY,ABEND=ASRA,TRAN=CA*,PROG=*
IN25CAPT TYPE=ENTRY,TRAN=DDX,PROG=MYPROG*
IN25CAPT TYPE=ENTRY,TRAN=X12*,PROG=USERPG1
IN25CAPT TYPE=ENTRY,ABEND=AN*,PROG=USERP*
IN25CAPT TYPE=FINAL
```

The following table lists the abends and resulting actions for the above source code:

Abend Codes, abending transactions and programs	Result
Abend ASRA, program IEFBR14, transaction CAD	Capture, due to entry TRAN=CA*,PROG=*
Abend ASRA, program GENE, transaction DDX	No effect, no matching entries
Abend AICA, program MYPROG, transaction X123	No effect, no matching entries
Abend ANQA, program PGM123, transaction ABCD	No effect, no matching entries
Abend ANQA, program USERPG3, any transaction	Capture due to entry ABEND=AN*,PROG=USERP
Abend AD22, program USERPG2, transaction ABCD	No effect, no matching entries
Any abend, program MYPROG2, transaction DDX	Capture due to entry TRAN=DDX, PROG=MYPROG*
Abend ASRA, program USER1, transaction X12	No effect, no matching entries

## External Security Option

- [Resource Protection Levels \(see page 85\)](#)
- [External Security Using CA Top Secret \(see page 87\)](#)
- [External Security Using CA ACF2 \(see page 88\)](#)
- [External Security Using IBM RACF \(see page 89\)](#)

The EXTSEC installation option in CAVHCONF controls the external security feature for individual product components. The default is NONE. For more information about setting and modifying CAVHCONF options, see [Global Options \(see page 31\)](#).

## Resource Protection Levels

The resources protected in Symbolic Component under the External Security feature are as follows.



### Notes:

- Any systems that initialize, update options, delete, purge, or reload a PROTSYM also need the required access.
- The CAMRUSR1 and CAMRUSR2 user exits will be called only if the user has the necessary authorities for the requested action.

#### **VHALTER**

- **Class**  
CA@NTSYM
- **Entity**  
VHALTER
- **Access Level**  
Read
- **Function Type**  
Initialize, Update
- **Product Options/Menu Options**  
IN25UTIL INITIALIZE, UPDATE OPTIONS (MAXPGMVER, USEDSPACEMSG)

VHALTER lets you use the symbolic component IN25UTIL utility INITIALIZE and UPDATE functions.

#### **VHCONTR**

- **Class**  
CA@NTSYM
- **Entity**  
VHCONTR
- **Access Level**  
Read
- **Function Type**  
Delete, Purge, Reload Listing
- **Product Options/Menu Options**  
IN25UTIL DELETE, PURGE, or RELOAD

VHCONTR lets you use the symbolic component IN25UTIL utility DELETE , PURGE, and RELOAD functions.

The resources protected in InterTest Batch under the External Security feature are as follows:

#### **MRMISM**

- **Class**  
CA@NTSYM

- **Entity**  
MRMISM
- **Access Level**  
Read
- **Function Type**  
Mismatched Symbolic members
- **Product Options/Menu Options**  
Foreground or Batch Link debugging

MRMISM lets you select a mismatched symbolic member during debugging session.

## External Security Using CA Top Secret

The external security interface used in Symbolic Component is compatible with CA Top Secret 12.0 or above.

### Define the Resource Class CA@NTSYM

The Symbolic Component uses a unique resource class, CA@NTSYM, for its security controls.

### List a Resource Class

To determine if CA@NTSYM is currently defined to your CA Top Secret system, issue the following command:

```
TSS LIST(RDT) RESCLASS(CA@NTSYM)
```

This command lists the definition of the CA@NTSYM resource in the CA Top Secret Resource Definition Table (RDT).

- If CA@NTSYM is present and its definition matches the attributes below, no further action is required:  
ACCESS = ALL,UPDATE,READ DEFACC = READ
- If any of the attributes do not match, change the definition using the TSS REPLACE(RDT) command, specifying the correct parameters.
- If you get an error message indicating that the resource is missing, invalid, or reserved, you must define CA@NTSYM

If CA@NTSYM is not defined, you must add it to the RDT. Add new classes using the TSS command ADDTO function through any facility supporting the TSS command. Once added, these resources become a permanent part of the security file and can be immediately administered and reported on by name.

### Add a Resource Class

The syntax to add the CA@NTSYM resource is as follows.

```
TSS ADDTO(RDT) RESCODE(38) RESCLASS(CA@NTSYM) -  
ACLST(ALL,UPDATE,READ) DEFACC(READ)
```

The syntax uses a RESCODE of 38. This code value is up to each site and must be unique. Codes 01 to 3F are available.

Once you define the resource CA@NTSYM to your system, you can establish access controls to IN25UTIL functions. No IPL, re-initialization, or parameter changes are needed for these changes to become active and permanent.

## Use CA@NTSYM to Control IN25UTIL Functions

The CA@NTSYM resource controls IN25UTIL utility INITIALIZE, UPDATE, DELETE, PURGE, and RELOAD functions. CA@NTSYM is a standard short resource and is managed in the same manner as other access level resources.

Issue the following command:

```
TSS ADDT0(deptname) CA@NTSYM(VHALTER,VHCONTR)
```

- To allow user USER01 to have Read access to entity VHALTER, use the following command:

```
TSS PERMIT(USER01) CA@NTSYM(VHALTER) ACCESS(READ)
```

If you omit the ACCESS keyword, CA Top Secret defaults the PERMIT to Read access.

- To allow user USER01 to have Read access to entity VHCONTR, use the following command:

```
TSS PERMIT(USER01) CA@NTSYM(VHCONTR) ACCESS(READ)
```

If you omit the ACCESS keyword, CA Top Secret defaults the PERMIT to Read access.

## External Security Using CA ACF2

The external security interface in Common Component is compatible with CA ACF2 6.2 or higher.

### Define the Resource Class CA@NTSYM

Common Component uses a unique resource class, CA@NTSYM, for its internal security controls.

CA@NTSYM needs to be defined to CA ACF2 with its own three-character resource type. In the following example, CA@NTSYM is defined using the resource type SYM.

```
SET CONTROL(GSO) INSERT clasmap.CA@NTSYM -
RESOURCE(CA@NTSYM) RSRCTYPE(SYM) ENTITYLN(7)
```

After adding this clasmap definition, refresh the GSO clasmap.

To protect and allow access to the Common Component's protsym file, use the standard CA ACF2 rule writing.

```
SET RESOURCE(SYM) COMPILE * $KEY(VHALTER) TYPE(SYM) UID(uidstring/mask) SERVICE(READ)
ALLOW $KEY(VHCONTR) TYPE(SYM) UID(uidstring/mask) SERVICE(READ) ALLOW END STORE *
```

If the SERVICE keyword is not supplied, CA ACF2 defaults to permit access to all service levels.

## External Security Using IBM RACF

The external security interface in Common Component can be used with IBM RACF. It is compatible with IBM RACF v1.9 or higher.

### Define the Resource Class CA@NTSYM

Define a new class, CA@NTSYM, for Common Component security in the IBM RACF Class Descriptor Table (module ICHRRCDE). The following information should be included in the ICHERCDE macro entry:



**Note:** CA@NTSYM is referred to in other sections as CA@NTSYM.

```
REDEFINE CDT CA@NTSYM UACC(NONE) CDTINFO(POSIT(201) CASE(UPPER) FIRST(ALPHA,NATIONAL)
+
OTHER(ALPHA,NATIONAL) MAXLENGTH(7) MAXLENX(44) PROFILESALLOWED(YES))
```



**Note:** No recommendations are given for any DFTUACC; this is completely determined by each site.

### Router Table Entry for CA@NTSYM

Make an entry for the CA@NTSYM class in the IBM RACF Router Table (module ICHRFRTB). Specify ACTION=RACF. This entry is generated by the ICHRFRTB macro.

### Enabling the Class CA@NTSYM

Activate the new class CA@NTSYM just defined in the CDT and Router Table. To do this, issue the following command:

```
SETR0PTS RACLIST(CDT) REFRESH
```

Issue IBM RACF RDEFINES for Common Component Entities First, you must issue the IBM RACF RDEFINE command for every Common Component entity that you want to protect. For example, issue the following commands to protect the CA@NTSYM resources:

```
RDEFINE CA@NTSYM VHALTER UACC(NONE) RDEFINE CA@NTSYM VHCONTR UACC(NONE)
```

To protect and allow access to the Common Component's PROTSYM file, use the standard CA RACF rule writing.

#### Example:

Define a general user with access to VHALTER:

```
PERMIT VHALTER CLASS(CA@NTSYM) ID(userid) ACC(READ)
```

# User Exits

This article describes the user exits for each of the CA InterTest and CA SymDump products. The user exits are unique for each product. Click the page for your product to view the documentation on the user exits.

- [CA InterTest Batch User Exits \(see page 90\)](#)
- [CA SymDump Batch User Exits \(see page 91\)](#)
- [CA InterTest for CICS User Exits \(see page 104\)](#)
- [CA SymDump for CICS User Exits \(see page 113\)](#)

## CA InterTest Batch User Exits

This article describes CA InterTest Batch user exits.

- [Define Excluded Programs \(see page 90\)](#)
- [Define the Batch Link Selection User Exit \(see page 90\)](#)
- [Define the Batch Link Scheduling User Exit \(see page 91\)](#)

### Define Excluded Programs

CAVHSRC member CAMRXMOD defines the list of programs that will not be automatically monitored when wildcarding is used on the monitor control panel. Edit the list to include any programs that you do not want to be automatically monitored. We suggest that the supplied default list is left intact. If you must monitor a program whose name is present in this list, you can still monitor the program by explicitly adding the name of the program to the monitor control panel.

The JCL is included in the configuration PTF that assembles and links this program. The JCL member is CAMRXMOJ and it is found in the CAI.CAVHJCL data set. The JCL must be modified JCL to conform to their installation standards before submitting it.

### Define the Batch Link Selection User Exit

When you make a selection from the Batch Link Selection panel, a user exit, CAMRUSR1, is called. This allows for security checks to verify that the user who selected the job has the authority to monitor its execution. You can use the SYSID passed to CAMRUSR1 to enable further security checks, for instance when you have the [SYSPLEX option enabled \(see page \)](#) to allow debugging jobs on foreign LPARs.



**Note:** For more information about coding this user exit, see the CAVHJCL member CAMRUSR1. The shipped default member allows all users to select any available batch link job for debugging.

The JCL is included in the configuration PTF that assembles and links this program. The JCL member is CAMRUS1J and it is found in the CAI.CAVHJCL data set. The JCL must be modified JCL to conform to their installation standards before submitting it.

## Define the Batch Link Scheduling User Exit

When you make a selection from the Batch Link DB2 and IMS Schedule panel, a user exit, CAMRUSR2, is called. This allows for security checks to verify that the user has the authority to perform the scheduling functions.



**Note:** For more information about coding this user exit, see the CAVHJCL member CAMRUSR2. The shipped default member permits all users this authority.

The JCL is included in the configuration PTF that assembles and links this program. The JCL member is CAMRUS2J and it is found in the CAI.CAVHJCL data set. The JCL must be modified JCL to conform to their installation standards before submitting it.

## CA SymDump Batch User Exits

This article describes CA SymDump Batch user exits.

- [Control Exclusion Criteria \(see page 91\)](#)
- [Control the Destination of Reports \(see page 92\)](#)
- [Dynamic Allocation for the CAIPRINT DD \(see page 95\)](#)
- [Control Access to Reports in the Repository \(see page 98\)](#)
- [Management Reporting System \(MRS\) User Exits \(see page 100\)](#)

### Control Exclusion Criteria

CA SymDump Batch provides a user exit (CAOEUSR1) that permits more detailed exclusion or inclusion criteria. This user exit receives control following the examination of normal exclusion criteria (from CAOETABL).

CAOEUSR1 is entered with the following registers:

- **R1**  
Address of USRDATA parameter list
- **R13**  
Address of 72-byte save area
- **R14**  
Return address
- **R15**  
Address of CAOEUUSR1

The USRDATA parameter list contains the following information:

USRDATA	DSECT			
USRFLGS	DS	XL1	FLAG	BYTE

USRAC TM	EQU	B'10000000'	ACCOUNT CODE MATCH
USRSMFM	EQU	B'01000000'	SMF CODE MATCH
USRSABM	EQU	B'00100000'	SYSTEM ABEND CODE MATCH
USRUABM	EQU	B'00010000'	USER ABEND CODE MATCH
USRSUBM	EQU	B'00001000'	SUB SYSTEM MATCH
USRPROGM	EQU	B'00000100'	PROGRAM NAME MATCH
USRJOBM	EQU	B'00000010'	JOB NAME MATCH
USRACNT	DS	CL25	CURRENT JOB CARD ACCOUNT
USRSMF	DS	CL4	ACTIVE SMF ID
USRSABND	DS	CL3	USER ABEND CODE
USRUABND	DS	CL4	SYSTEM ABEND CODE
USRSUBS	DS	CL4	ACTIVE SUBSYSTEM
USRPROG	DS	CL8	PROGRAM NAME
USRJOB	DS	CL8	JOB NAME
	DS	0F	
USRWORK	DS	CL64	64-BYTE WORK AREA
USRL EN	EQU	*-USRDATA	

Use copybook CAOEP RM1 to map the USRDATA parm area.

CA SymDump Batch acts on the return code (in R15) from CAOEUR1 as follows:

- **0**  
The application uses its normal exclusion criteria from CAOETABL
- **4**  
The application produces a report
- **6**  
The application does not produce a report

After you customize the source for CAOEUR1 in CAI.CAVHJCL, submit the JCL to assemble and link the program. After linking the program, an IPL is required to load a new copy of the user exit into CSA.



When using the CAOEUR1 user exit, note the following:

- CAOEUR1 is provided as a stub that always returns with RC=0, so your exclusion criteria in CAOETABL is honored.
- CAOEUR1 executes AMODE=31 and RMODE=ANY and must be REENTRANT.

## Control the Destination of Reports

### CAIPRINT Repository Library

CA SymDump Batch provides a user exit, CAOCUPRT, which allows the CAIPRINT repository library to be determined based upon user-specified criteria (such as the userid or job name). The user exit receives control if the CAIPRTL B DD is not specified in the JCL.

To control the CAIPRINT repository library, code this user exit in Assembler language, and ensure that it conforms to the register conventions described next.

## Register Conventions

CAOCUPRT is entered with the following registers:

- **R1**  
Address of PRTPARM parameter list (as defined next)
- **R13**  
Address of 72-byte save area
- **R14**  
Return address
- **R15**  
Address of CAOCUPRT

## User Exit Parameter List

The PRTPARM parameter list contains the following information:

```

PRTPARM DSECT                                PARMLIST PASSED TO CAOCUPRT
PRTJOB  DS   CL8                              JOB NAME
PRTSTEP DS   CL8                              STEP NAME
PRTUSRID DS  CL8                              USERID
PRTTIME DS   CL8                              TIME OF REPORT IN EBCIDIC
*                                             (00HHMMSS)
PRTDATE DS   CL8                              DATE OF REPORT IN EBCIDIC
*                                             (YYYYMMDD)
PRTPGMID DS  CL8                              PROGRAM ID
PRTCODE  DS  CL7                              COMPLETION CODE (LEFT JUSTIFIED)
PRTDSN   DS  CL44                             DATA SET NAME
PRTPARML EQU *-PRTPARM
* NOTE: THE COMPLETION CODE WILL BE IN ONE OF THE FOLLOWING
* FORMATS:
*     SYSTEM ABEND CODE = SXXX
*     USER ABEND CODE  = UXXXX
*     MESSAGE ID       = MSGXXXX (I.E., CEE3501 OR IGZ0035)
*     RETURN CODE      = RCXXXX

```

All of the fields in the PRTPARM DSECT, including PRTDSN, are set before the call to CAOCUPRT. However, if the application did not abend, the program name will be blank.

Upon return from CAOCUPRT, if R15 contains a nonzero return code, the repository library defined in the default CSECT, CAOUDFRX, or specified by the option override in the CAIOPTS DD is used. If R15 contains a zero return code, the repository library returned by CAOCUPRT in the PRTDSN field of the PRTPARM DSECT is used.

## Sample User Exit Program

The following sample program returns a repository data set name based upon the assigned group of the user:

```

CAOCUPRT TITLE 'CAOCUPRT - CAIPRINT Repository User Exit'
CAOCUPRT CSECT
CAOCUPRT AMODE 31
CAOCUPRT RMODE ANY

```

```

        USING CAOCUPRT,12
        STM R14,R12,12(R13)      SAVE REGS
        LR  R12,R15              SET BASE REGISTER
        SPACE ,
        LR  R3,R1                -> INPUT PARM LIST
        USING PRTPARM,R3
        LA  R2,USRTBL           -> USERID TABLE
        USING USERDEF,R2
TBLLOOP DS  0H
        C    R2,=A(USRTEND)     END OF TABLE?
        BNL NODSN              YES, SET RC AND EXIT
        CLC USERID,PRTUSRID    FIND USERID IN TABLE?
        BE  GETDSN             YES, GET PRTLIB DSN
        LA  R2,USERDEFL(,R2)    GET NEXT ENTRY
        B   TBLLOOP            LOOP
        SPACE ,
GETDSN  DS  0H
        LH  R1,USERGRP         GET GROUP INDEX
        MH  R1,=H'44'          CALC INDEX INTO DSN LIST
        A   R1,=A(DSNLIST)
        MVC PRTDSN,0(R1)      SET PRTLIB DSN
        SR  R15,R15           SET RC=0
        B   EXIT              RETURN
        DROP R2,R3
        SPACE ,
NODSN   DS  0H
        LA  R15,4              SET RC=4
        SPACE ,
EXIT    DS  0H
        L   R14,12(R13)        RESTORE R14
        LM  R0,R12,20(R13)     RESTORE R0-R12
        BR  R14                RETURN
        SPACE ,
        TITLE 'GROUP EQUATES'
DEVL    EQU 0
QA      EQU 1
PROD    EQU 2
        SPACE ,
        TITLE 'USERID TABLE'
USRTBL  EQU *
        DC  CL8'USER1',Y(DEVL)
        DC  CL8'USER2',Y(DEVL)
        DC  CL8'USER3',Y(DEVL)
        DC  CL8'USER4',Y(QA)
        DC  CL8'USER5',Y(QA)
        DC  CL8'USER6',Y(QA)
        DC  CL8'USER7',Y(QA)
        DC  CL8'USER8',Y(PROD)
        DC  CL8'USER9',Y(PROD)
        DC  CL8'USER10',Y(PROD)
USRTEND EQU *
        SPACE ,
        TITLE 'PRTLIB DSN LIST'
DSNLIST EQU *
        DC  CL44'DEVL.PRTLIB'
        DC  CL44'QA.PRTLIB'
        DC  CL44'PROD.PRTLIB'
        LTORG ,
        EJECT
R0      EQU 0
R1      EQU 1
R2      EQU 2
R3      EQU 3
R4      EQU 4
R5      EQU 5
R6      EQU 6
R7      EQU 7
R8      EQU 8
R9      EQU 9
R10     EQU 10
R11     EQU 11
    
```

```

R12      EQU    12
R13      EQU    13
R14      EQU    14
R15      EQU    15
        SPACE 1
PRTPARM  DSECT                PARMLIST PASSED TO CAOCUPRT
PRTJOB   DS     CL8            JOB NAME
PRTSTEP  DS     CL8            STEP NAME
PRTUSRID DS     CL8            USERID
PRTTIME  DS     CL8            TIME OF REPORT IN EBCIDIC
*        (00HHMMSS)
PRTDATE  DS     CL8            DATE OF REPORT IN EBCIDIC
*        (YYYYMMDD)
PRTPGMID DS     CL8            PROGRAM ID
PRTCODE  DS     CL7            COMPLETION CODE (LEFT JUSTIFIED)
PRTDSN   DS     CL44           DSN RETURNED
PRTPARML EQU    *-PRTPARM
* NOTE:  THE COMPLETION CODE WILL BE IN ONE OF THE FOLLOWING
*        FORMATS:
*        SYSTEM ABEND CODE = SXXX
*        USER ABEND CODE   = UXXXX
*        MESSAGE ID        = MSGXXXX (I.E., CEE3501 OR IGZ0035)
*        RETURN CODE       = RCXXXX
        SPACE 1
USERDEF  DSECT
USERID   DS     CL8
USERGRP  DS     H
USERDEFL EQU    *-USERDEF
        SPACE 1
        END    CAOCUPRT

```

## User Exit JCL

JCL to assemble and link this user exit can be found in CAI.CAVHJCL(CAOCUPRT). The previous example is provided in that member.



When using the CAOCUPRT user exit to control the destination of abend reports, note the following information:

- CAOCUPRT is provided as a stub that always returns with RC=4. Therefore, the repository data set name does not change if a user exit is not provided.
- CAOCUPRT can execute AMODE=24 or AMODE=31, and must be REENTRANT.

## Dynamic Allocation for the CAIPRINT DD

CA SymDump Batch provides a user exit, CAOCUDSN that enables CAIPRINT to be written to a user-specified data set. The user exit receives control if the CAIPRINT DD is not specified in the JCL.

To control dynamic allocation for the CAIPRINT DD, code this user exit in Assembler language, and ensure that it conforms to the register conventions described next.

## Register Conventions

CAOCUDSN is entered with the following registers:

- **R1**  
Address of the USERDSN parameter list (as defined in User Exit Parameter List)
- **R13**  
Address of 72-byte save area
- **R14**  
Return address
- **R15**  
Address of CAOCUDSN

## User Exit Parameter List

The USRDSN parameter list contains the following information:

```

USERDSN DSECT      * PARM LIST PASSED TO CAOCUDSN
USRJOBID DS  CL8    JOB NUMBER (ASSIGNED BY THE SYSTEM)
USRJBNAM DS  CL8    JOB NAME (FROM THE JOB CARD)
USRJBSTP DS  CL8    STEP NAME
USRLENID DS  CL1    LENGTH OF USERID
USRUSRID DS  CL8    USERID (PERSON WHO SUBMITTED THE JOB)
USRTIME DS  CL8    TIME OF REPORT IN EBCIDIC (FORMAT THMMSS)
USRDATE DS  CL8    DATE OF REPORT IN EBCIDIC (FORMAT MMDDYYYY)
USRPRGNM DS  CL8    MAIN PROGRAM (LOAD MODULE ABEND OCCURRED IN)
USRABCODE DS  CL5   COMPLETION CODE (LEFT JUSTIFIED, Sxxx OR
*                      Unnnn
* LAST 3 FIELDS ARE FILLED IN BY THE USER EXIT CAOCUDSN
USRDSN DS  CL44    DSN RETURNED
USRUNIT DS  CL8    UNIT NAME RETURNED (OPTIONAL)
USRVOL DS  CL6    VOLSER FOR DSN RETURNED (OPTIONAL)
USRLEN EQU *-USRDSN

```

Upon return from CAOCUDSN, if R15 contains a nonzero return code or the USRDSN field is blank, CAIPRINT is dynamically allocated to SYSOUT. If R15 contains a zero return code, CAIPRINT is dynamically allocated to the data set defined in USRDSN. If the UNIT or volume fields were also specified, they are used by dynamic allocation.

### Example:

A sample user exit program appears next:

```

CAOCUDSN TITLE 'CAOCUDSN'
CAOCUDSN CSECT
CAOCUDSN AMODE 24
CAOCUDSN RMODE 24
*THIS SAMPLE PROGRAM WILL RETURN A DATA SET NAME OF
*USERID.TIME.JOBNAME
      USING CAOCUDSN,R12
      STM  R14,R12,12(R13)
      LR  R12,R15
      LR  R3,R1
      USING USERDSN,R3
      XR  R4,R4
      IC  R4,USRLENID          GET LENGTH OF USERID
      BCTR R4,0                EXECUTE LENGTH
      LA  R5,USRDSN           POINT TO OUTPUT FIELD
      EX  R4,MOVEID           MOVE IN USERID
      LA  R5,1(R4,R5)         BUMP POINTER
      MVI 0(R5),C'.'         MOVE IN PERIOD

```

```

        LA      R5,1(,R5)           BUMP POINTER
        MVC     0(8,R5),USRTIME    MOVE IN TIME
        LA      R5,8(,R5)           BUMP POINTER
        MVI     0(R5),C'. '        MOVE IN PERIOD
        LA      R5,1(,R5)           BUMP POINTER
        MVC     0(8,R5),USRJBNAM    MOVE IN JOB NAME
        LA      R5,8(,R5)           BUMP POINTER
        L       R14,12(R13)
        XR      R15,R15
        LM      R0,R12,20(R13)
        BR      R14
MOVEID  MVC     0(R4,R5),USRUSRID
        LTORG  ,
        EJECT
R0      EQU    0
R1      EQU    1
R2      EQU    2
R3      EQU    3
R4      EQU    4
R5      EQU    5
R6      EQU    6
R7      EQU    7
R8      EQU    8
R9      EQU    9
R10     EQU    10
R11     EQU    11
R12     EQU    12
R13     EQU    13
R14     EQU    14
R15     EQU    15
USERDSN DSECT
* PARM LIST PASSED TO CAOCUDSN
USRJOBID DS CL8   JOB NUMBER (ASSIGNED BY THE SYSTEM)
USRJBNAM DS CL8   JOB NAME (FROM THE JOB CARD)
USRJBSTP DS CL8   STEP NAME
USRLENID DS CL1   LENGTH OF USERID
USRUSRID DS CL8   USERID (PERSON WHO SUBMITTED THE JOB)
USRTIME  DS CL8   TIME OF REPORT IN EBCIDIC (FORMAT THMMSS)
USRDATE  DS CL8   DATE OF REPORT IN EBCIDIC (FORMAT MMDDYYYY)
USRPRGNM DS CL8   MAIN PROGRAM (LOAD MODULE ABEND OCCURRED IN)
USRABCDE DS CL5   COMPLETION CODE (LEFT JUSTIFIED, SXXX OR
* UNNNN)
* LAST 3 FIELDS ARE FILLED IN BY THE USER EXIT CAOCUDSN
USRDSN   DS CL44  DSN RETURNED
USRUNIT  DS CL8   UNIT NAME RETURNED (OPTIONAL)
USRVOL   DS CL6   VOLSER FOR DSN RETURNED (OPTIONAL)
USRLEN   EQU *-USRDSN
        END     CAOCUDSN

```

## User Exit JCL

JCL to assemble and link this user exit can be found in CAI.CAVHJCL(CAOIUSR1). The previous example is provided in that member.



When using the CAOCUDSN user exit to control the destination of abend reports, note the following information:

- CAOCUDSN is provided as a stub that always returns with RC=4. Therefore, the dynamic allocation of CAIPRINT does not change if a user exit is not provided.
- CAOCUDSN executes AMODE=24 and RMODE=24 and must be REENTRANT. It is linked with module CAOCIN00.

- When CAOUSR1 is updated, the product must be reinitialized for the user exit to take effect.

## Control Access to Reports in the Repository

You can customize and build the user exit program CAOUSR1 to secure the following types of access to the abend repository:

- Viewing reports
- Deleting reports
- Locking and unlocking reports
- Modifying formatting options
- Printing reports

This user exit receives control whenever any user attempts to access the repository for any of these operations.

To control access to reports in the repository, you must code this user exit in Assembler language, and it must conform to the conventions described next.

## Input Registers

CAOIPRM1 is entered with the following registers:

- **R1**  
Address of the input parm, mapped by DSECT CAOIPRM1. This area contains the ID of the user attempting the access, the request type, and information about the report being accessed.
- **R13**  
Address of a 72-byte register save area provided by the caller.
- **R14**  
Address where CAOUSR1 must return to its caller.
- **R15**  
Address of CAOUSR1.

## Output Registers

CAOIPRM1 is required to restore all registers to their original values, except R15, which must return one of the following values:

- **R15=0**  
Allow the access requested.
- **R15=4**  
Deny the access requested

Any other value in R15 may produce unpredictable results.

### User Exit Parameter List

The CAOIPRM1 parameter list contains the following information:

CAOIPRM1	DSECT		
PRM1REQ	DS	XL4	Request type
P1_OPEN	EQU	5	5 = Open a report for view
P1_DEL	EQU	9	9 = Delete a report
P1_LOCK	EQU	16	16 = Lock a report
P1_UNLK	EQU	17	17 = Unlock a report
P1_OPTS	EQU	21	21 = Update report options
P1_PRINT	EQU	22	22 = Print a report
PRM1USR	DS	CL8	Requestor's user id
PRM1RJOB	DS	CL8	Report Job Name
PRM1RUSR	DS	CL8	Report User Id
PRM1WORK	DS	XL256	256-BYTE WORK AREA
LPRM1	EQU	*-CAOIPRM1	

### Sample User Exit Program

The following sample version of CAOIUSR1 that unconditionally enables every requested access is provided in CA1.CAVHJCL:

```

CAOIUSR1 TITLE 'Sample Repository Viewer Security Exit'
**-----**
**
**  CAOIUSR1 - Sample Repository Viewer Security Exit
**
**  This user exit performs security validation prior to
**  various repository viewer functions.
**
**  Input Registers:
**
**  R1 contains the address of the parm area (CAOIPRM1)
**  R13 contains the address of a register save area
**  R14 contains the return address
**  R15 contains the address of CAOIUSR1
**
**  All other registers are unpredictable.
**
**  Output Registers:
**
**  R15 contains the return code (described next)
**
**  All other registers must be returned unchanged.
**
**  Return Codes
**
**  R15 = 0   Allow access.
**  R15 > 0  Disallow access.
**-----**

CAOIUSR1  SPACE ,
          CSECT
          USING CAOIUSR1,R12      Map the User Exit
          USING CAOIPRM1,R8      Map the Parameter Area
          USING LDATA,R13        Map Local Data
          SPACE ,
          STM  R14,R12,12(R13)    Save caller's registers
          LR   R12,R15           Set procedure base
          LR   R8,R1             Set parm base
          SPACE ,
          LA   R1,PRM1WORK       @ Work area provided
          XC   0(256,R1),0(R1)   Clear it out
    
```

```

        ST   R13,4(,R1)           Save backward chain
        ST   R1,8(,R13)          Save forward chain
        LR   R13,R1              Set our RSA address
        SPACE ,
* Perform security validation here.
        SPACE ,
RETURN  DS   0H
        L    R15,LDRC            Load the return code
        L    R13,4(,R13)         Restore @ caller's rsa
        L    R14,12(,R13)        Load return address
        LM   R0,R12,20(R13)      Restore registers
        BR   R14                 Return to caller
        TITLE 'LITERALS'
        LTORG ,
        TITLE 'DSECTS: Local Data'
LDATA  DSECT
LDSAVE DS   18F
LDRC   DS   F
        TITLE 'DSECTS: Parameter Area'
        COPY CA0IPRM1
        TITLE 'REGISTER EQUATES'
R0     EQU  0
R1     EQU  1
R2     EQU  2
R3     EQU  3
R4     EQU  4
R5     EQU  5
R6     EQU  6
R7     EQU  7
R8     EQU  8
R9     EQU  9
R10    EQU 10
R11    EQU 11
R12    EQU 12
R13    EQU 13
R14    EQU 14
R15    EQU 15
        END   CA0IUSR1

```

## User Exit JCL

After you customize the source for CA0IUSR1 in CAI.CAVHJCL, submit the JCL to assemble and link the program.



When using the CA0IUSR1 user exit to secure the various types of access to the abend repository, note the following information:

- CA0IUSR1 resides as a stand-alone single-CSECT load module.
- CA0IUSR1 receives control in 31-bit addressing mode.

## Management Reporting System (MRS) User Exits

The Management Reporting System (MRS) provides you with a facility to develop your own output report. Two user exits enable access to the data that the MRS processes. You can specify these exits separately or concurrently by using the COBOLEX or NOCOBEX execution options. (COBOLEX accesses Detail Report information; NOCOBEX accesses non-COBOL CSECTS.)

## Access Detail Report Information (COBOLEX)

COBOLEX provides access to the information that MRS uses to print the Detail Report. If you specify this exit, MRS loads a copy of the specified user program (see Invoke a User Exit JCL). Each time MRS processes a COBOL CSECT, a branch is made and data is passed to the user program using a parameter list pointed to by Register 1. The list contains the two parameters shown next:

```
XDATADDR DS A          address of MRS data
XNEWADDR DS AL1(128),AL3 address for a substitute print line
```

XDATADDR contains the address of the CSECT data being passed. The following DSECT shows the order in which the fields appear:

EXITBUF	DS	0CL102	USER EXIT BUFFER
XITMNAME	DS	CL8	MODULE NAME
XITMSIZE	DS	PL5	MODULE SIZE
XITCNAME	DS	CL8	CSECT NAME
XITCSIZE	DS	PL5	CSECT SIZE
XITDDLEN	DS	PL5	DATA SIZE
XITPDLEN	DS	PL5	PROCEDURE SIZE
XITCMPLR	DS	CL4	COMPILER
*****			
* POSSIBLE VALUES ARE			
*	DC	C'COBF'	COBOL-F
*	DC	C'ANS2'	ANS VERSION 2
*	DC	C'ANS3'	ANS VERSION 3
*	DC	C'ANS4'	ANS VERSION 4
*	DC	C'VSR1'	OS/V5 COBOL
*	DC	C' C2 '	COBOL/II
*	DC	C' C3 '	COBOL/370
*	DC	C'CMVS'	COBOL FOR MVS AND VM
*	DC	C'C390'	COBOL FOR OS/390
*****			
XITCMPID	DS	CL8	COMPILER ID
	DS	CL1	FILLER
XITCREL	DS	CL3	COMPILER RELEASE
	DS	CL3	FILLER
XITCDATE	DS	CL8	COMPILE DATE
	DS	CL3	FILLER
XITCTIME	DS	CL8	COMPILE TIME
XITOPTIM	DS	CL6	OPTIMIZER
	DS	CL1	FILLER
XITOPREL	DS	CL3	OPTIMIZER RELEASE
XITD	DS	CL1	DTECT
XITP	DS	CL1	PFLOW
XITC	DS	CL1	XCOUNT
XITX	DS	CL1	XTIME
XITF	DS	CL1	FDCHECK
XITS	DS	CL1	SUBRNGCK
XITR	DS	CL1	PARMCHK
XITB	DS	CL1	DBGSLLEEP
XITW	DS	CL1	WSINIT
XITI	DS	CL1	CICS
XITE	DS	CL1	EOS
XITL	DS	CL1	WSCLEAR
XITH	DS	CL1	PUSHPOP
XITO	DS	CL1	CASORT
	DS	CL4	FILLER

You can use XNEWADDR to pass back a substitute print line.

MRS expects the user program to pass a return code back in Register 15. When a return code 0 is passed back to MRS, processing continues as normal (the print line is output to SYSPRINT). When the return code is set to 8, MRS substitutes the print line whose address is passed back to MRS in the

return parameter list (XNEWADDR). When substituting a new print line, specify a carriage control character in column 1. The total line length, including the carriage control, must be 133 bytes. An error message replaces the normal print line for any other return code.

The MRS branches to the user program one final time after processing the entire library. The XDATADDR parameter addresses binary zeros in all of the data fields as a signal to the user program to close any open files. A system abend can occur if open files are not closed.

### Access Non-COBOL CSECTs (NOCOBEX)

NOCOBEX provides access to every non-COBOL CSECT in the library, except for the COBOL ILBOxxxx and IGZxxxx routines in COBOL load modules.

MRS loads a copy of the specified user program if you specify this exit in the JCL (see Invoke a User Exit JCL). Each time MRS processes a non-COBOL CSECT, a branch is made and the CSECT data is passed to the user program using a parameter list pointed to by Register 1. The list contains one address:

```
XDATADDR DS AL1(128),AL3
```

XDATADDR contains the address of the CSECT data being passed. This DSECT shows the order in which the data appears:

XDATA	DSECT		
XMNAME	DS	CL8	LOAD MODULE NAME
XMSIZE	DS	PL5	LOAD MODULE SIZE
XPNAME	DS	CL8	PROGRAM (CSECT) NAME
XPSIZE	DS	PL5	PROGRAM (CSECT) SIZE
XCOBONLY	DS	CL76	FOR FUTURE USE

MRS does not expect a return code from the NOCOBEX user program. MRS branches to the user program one final time after processing the entire library. The XDATADDR parameter addresses binary zeros in all of the data fields as a signal to the user program to close any open files. A system abend can occur if open files are not closed.

### Sample User Exit Program

The following Assembler program shows how you can use NOCOBEX to print your own detail report for non-COBOL CSECTs.

```

ASMEXIT TITLE 'MRS EXIT ROUTINE FOR NON COBOL CSECTS (NOCOBEX)'
ASMEXIT CSECT
        STM 14,12,12(13)
        LR 12,15
        USING ASMEEXIT,12
        LA 2,SAVE                                PERFORM SAVE AREA CHAINING
        ST 13,4(,2)                                BETWEEN MRS AND ASMEEXIT.
        ST 2,8(,13)
        LR 13,2
        L 10,0(,1)                                GET DATA POINTER AND SET UP
        USING INLINE,10                            ADDRESSABILITY FOR DATA AND
        LA 11,BUFFER                                OUTPUT BUFFERS.
        USING OUTLINE,11
        MVI OUTLINE,X'40'                            CLEAR OUTPUT LINE
        MVC OUTLINE+1(132),OUTLINE
        CLI INMOD,C'A'                            CHECK FOR VALID DATA BUFFER.
        BL ALLDONE                                - IF NOT VALID THEN DONE.
        TM FLAGS,PASS1                            CHECK FOR 1ST ENTRY TO ASMEEXIT.
        BZ GETCSECT                                - IF 1ST ENTRY OPEN SYSRPT.
    
```

```

OPEN (OUTDCB,(OUTPUT))
PUT OUTDCB,HEADLINE OUTPUT HEADER LINE.
NI FLAGS,X'FF'-PASS1 RESET 1ST PASS FLAG.
GETCSECT DS 0H
AP TOTCSIZE,INCSIZE ADD CSECT SIZE TO TOTALS.
MVC OUTCSECT,INCSECT MOVE CSECT NAME TO OUT BUFFER.
MVC EDITBUF,PATTERN EDIT CSECT SIZE INTO BUFFER.
ED EDITBUF,INCSIZE
MVC OUTCSIZE,EDITBUF+4
CLC INMOD,CURRMOD CHECK FOR NEW MODULE NAME.
BE WRITEREC - IF NEW THEN GET NEW DATA.
GETMOD DS 0H
AP TOTMSIZE,INMSIZE ADD NEW MODULE SIZE TO TOTALS.
MVC OUTMOD,INMOD MOVE MODULE NAME TO OUT BUFFER.
MVC EDITBUF,PATTERN EDIT MODULE SIZE INTO BUFFER.
ED EDITBUF,INMSIZE
MVC OUTMSIZE,EDITBUF+4
MVC CURRMOD,INMOD
WRITEREC DS 0H
PUT OUTDCB,OUTLINE OUTPUT THIS RECORD TO SYSRPT.
B RETURN RETURN TO MRS.
ALLDONE DS 0H
MVC BUFFER,TOTLINE MOVE SKELETON LINE TO BUFFER
MVC EDITBUF,PATTERN EDIT TOTAL MODULE SIZE INTO
ED EDITBUF,TOTMSIZE OUTPUT BUFFER.
MVC OUTMSIZE,EDITBUF+4
MVC EDITBUF,PATTERN EDIT TOTAL CSECT SI<E INTO
ED EDITBUF,TOTCSIZE OUTPUT BUFFER.
MVC OUTCSIZE,EDITBUF+4
PUT OUTDCB,OUTLINE SPACE AND OUTPUT TOTALS.
CLOSE (OUTDCB) CLOSE SYSRPT FILE.
RETURN DS 0H
L 13,4(,13) RESTORE SAVE AREA ADDRESS.
LM 14,12,12(13) RESTORE REGISTERS FOR MRS.
XR 15,15 SET ZERO RETURN CODE.
BR 14 RETURN TO MRS.
*
* LITERALS, CONSTANTS AND VARIABLES FOR ASMEXIT
*
LTORG
SAVE DC 18F'0' REGISTER SAVE AREA
CURRMOD DC CL8' ' CURRENT LOAD MODULE NAME
TOTMSIZE DC PL5'0' TOTAL MODULE SIZE (BYTES)
TOTCSIZE DC PL5'0' TOTAL CSECT SIZE (BYTES)
EDITBUF DC XL12'00'
PATTERN DC XL12'402020206B2020206B202020'
FLAGS DC AL1(PASS1)
PASS1 EQU X'80'
OUTDCB DCB DDNAME=SYSRPT,MACRF=PM,DSORG=PS,RECFM=F,LRECL=133, X
BLKSIZE=133
BUFFER DC CL133' '
HEADLINE DC CL133' MODULE LENGTH CSECT LENGTH'
TOTLINE DC CL133' TOTALS'
*
* DSECT FOR OUTPUT BUFFER (SYSRPT)
*
OUTLINE DSECT
OUTCTL DS CL1 CTL CHARACTER FOR OUTPUT
OUTMOD DS CL8 LOAD MODULE NAME
DS CL2
OUTMSIZE DS CL8 LOAD MODULE SIZE
DS CL8
OUTCSECT DS CL8 CSECT NAME
DS CL2
OUTCSIZE DS CL8 CSECT SIZE
DS CL2
OUTXTRA DS CL76 NOT USED
*
* INPUT DATA FROM MRS (CSECT/MODULE INFORMATION)
*

```

INLINE	DSECT		
INMOD	DS	CL8	MODULE NAME
INMSIZE	DS	PL5	MODULE SIZE
INCSECT	DS	CL8	CSECT NAME
INCSIZE	DS	PL5	CSECT SIZE
INXTRA	DS	CL76	NOT USED
	END	ASMEXIT	

## Invoke a User Exit JCL

To invoke a user exit, code the PARM parameter on the EXEC statement.

COBOLEX and NOCOBEX are keyword parameters that require the name of the designated user program as a value. Specify the user exits separately or concurrently.

If you are using a cataloged procedure, code the PARM parameter as follows:

```
//STEP10 EXEC CAIMRSII,LIBRARY='COBOL load library',
//          PARM=(' NOCOBEX=XXXXXXXX, COBOLEX=YYYYYYYY')
```

## CA InterTest for CICS User Exits

This article describes CA InterTest for CICS user exits.

- [Unprotect Main Storage with User Exit IN25LETX \(see page 104\)](#)
- [Password Protection \(see page 107\)](#)

### Unprotect Main Storage with User Exit IN25LETX

IN25LETX is an optional exit program coded by the user that specifies areas in main storage that should not be protected by CA InterTest for CICS. The areas to be unprotected are indicated by beginning and ending storage addresses.

This exit program provides a programmable function equivalent to the CA InterTest for CICS interactive LET option. This is a particularly useful option, because frequently an application program is designed to modify a storage area that is protected by CA InterTest for CICS. Ordinarily when that happens, CA InterTest for CICS initiates an automatic breakpoint and halts the program's execution.

IN25LETX must follow the rules for command-level programs. It executes within the task of the monitored application program, and must not do anything that would affect normal execution of the application. IN25LETX is invoked each time CA InterTest for CICS begins monitoring and also when it resumes monitoring, such as after a CICS command or after any kind of breakpoint.

The interface to IN25LETX is a CICS command level LINK with a COMMAREA. This allows a user's IN25LETX module to issue CICS SPI or API commands to locate key module addresses.



**Note:** The interface permits IN25LETX to return a variable-length table of pairs to CA InterTest for CICS in a single pass. Previous non-ESA versions of IN25LETX branched to CA InterTest for CICS for each address pair until all address pairs were passed.

## IN25LETX COMMAREA

The following table contains the IN25LETX COMMAREA fields. See the section Sample IN25LETX Program for the COMMAREA layout.

COMMAREA Field	Length	Primed By
Table Address	Fullword	User
Monitored Program	8 bytes	CA InterTest for CICS
Application Registers	Fullword	CA InterTest for CICS

- **Table Address**  
This field is set to zeros upon entry to the exit. CA InterTest for CICS assumes it was passed a valid table address if this field is modified. Leaving this field set to zeros causes CA InterTest for CICS to remove any address pairs previously set by this exit for this task.
- **Monitored Program**  
Contains the name of the program being monitored by CA InterTest for CICS at the time this exit was invoked.
- **Application Registers**  
The fullword address of the 16-fullword area containing the application's registers (0 to 15) that were saved just before the entry to this exit.

## LETX Table

For each area of main storage that is to be unprotected, IN25LETX stores a pair of addresses in a table (the LETX table). Each pair of addresses represents a *from* and a *to* area of storage that will not be protected by CA InterTest for CICS if those areas are modified by the application. For example, if the area to be unprotected resides at address X'11AA2E0' and is X'2000' bytes long, the address pair is 11AA2E0 and 11AC2E0.

The last entry in the table must contain two full words of X'FFFFFFFF'.

In many cases, the declared addresses may only need to be unprotected for a particular CICS task or program execution. Thus the logic of IN25LETX may need to consider the varying circumstances of execution, including the following tasks:

- Transaction identification
- Terminal identification
- Logical level of execution
- Program identification

IN25LETX must deliver a full table of address pairs each time it is invoked. IN25LETX does not have to be reentrant, meaning it can contain save areas and tables in which permanent addresses can be stored. There is no need to rebuild the LETX table if the addresses do not change. In this case, a one-time routine can find the areas, build the LETX table, and save the table in the IN25LETX module for the next time. IN25LETX can make itself resident by loading itself again (see the section Sample IN25LETX Program).

IN25LETX can reside above or below the 16 MB line and execute in either addressing mode. All table addresses, however, must have the X'80000000' bit turned off; that is, the address X'01234567' must not be passed as X'81234567'.

## Sample IN25LETX Program

The sample IN25LETX program that follows locates two areas, both permanently resident in storage, which are to be unprotected for all monitored programs running under transaction code ABCD. The first area is a table pointed to by an address in the user's TWA. The second area is a table loaded as a member of the CICS program definition named ATABLE. Both areas are established before CA InterTest for CICS becomes active and remain resident in main storage.

```

*
          TITLE 'IN25LETX -- CA INTERTEST LET EXIT -- USER CODED'
          DFHREGS                                REGISTER EQUATES
*
DFHEISTG DSECT                                WORKING STORAGE DSECT
XPROGLEN DS    F                                PROGRAM LENGTH
*****
*          I N 2 5 L E T X    C O M M A R E A          *
*****
COMMAREA DSECT                                COMMAREA
CTBLADDR DS    F                                ADDRESS OF USER LET TABLE
INTPROG  DS    CL8                             NAME OF MONITORED PROGRAM
INTREGS  DS    F                                ADDR OF APPL REGISTERS (0-15)
*
USERTWA  DSECT                                DSECT FOR USER TWA
UTBLADDR DS    F                                ADDRESS OF USER TABLE
UTBLEN  DS    H                                LENGTH OF USER TABLE
*
*****
IN25LETX DFHEIENT EIBREG=R11,CODEREG=R12,DATAREG=R13
*
          L      R10,DFHEICAP                    POINT TO COMMAREA
          USING COMMAREA,R10
*
          CLC   EIBTRNID,=C'ABCD'                IS THIS THE TRANSACTION?
          BNE   RETURN                            NO, RETURN, DO NOT SET ANY
*                                                    ADDRESSES FOR THIS TASK
*
PROCESS  CLC   LTWATBL,=D'0'                     HAVE THEY BEEN ALREADY SAVED?
          BE    GETAREAS                          NO, GO GET THEM
          LA   R1,LTABLE                          THEN, THE LET TABLE IS BUILT...
          ST   R1,CTBLADDR                        PUT LET TABLE ADDR IN THE COMMAREA
*
RETURN   EXEC CICS RETURN
*
GETAREAS DS    0H
*
          EXEC CICS ADDRESS TWA(R5)
          USING USERTWA,R5
*
          L      R1,UTBLADDR                      GET THE START OF THE TWA AREA
          N      R1,=X'7FFFFFFF'                 ENSURE HIGH-ORDER BIT IS TURNED
*                                                    OFF IN USER'S ADDRESS
          ST   R1,LTWATBL                         STORE THE AREA'S START ADDRESS
          AH   R1,UTBLEN                          ADD THE TABLE LENGTH
          ST   R1,LTWATBL+4                       STORE THE AREA'S END ADDRESS
*
          EXEC CICS INQUIRE PROGRAM('ATABLE') LOADPOINT(R0)          X
                   LENGTH(XPROGLEN)
*
          ST   R0,LPROGTBL                         STORE THE AREA'S START ADDRESS
          A    R0,XPROGLEN                         ADD THE PROGRAM'S LENGTH
          ST   R0,LPROGTBL+4                       STORE THE AREA'S END ADDRESS
*
*****
* LOAD THIS EXIT AGAIN SO IT WILL NOT HAVE TO REOBTAIN THE ADDRESSES *

```

```

* ON THE NEXT CALL TO THIS EXIT.
*****
EXEC CICS LOAD PROGRAM(' IN25LETX' ) HOLD
*
      LA      R1,LTABLE
      ST      R1,CTBLADDR      PUT ADDR OF LET TABLE IN COMMAREA
      B      RETURN
*
LTABLE DS    0H              LET TABLE
LTWATBL DC   2F'0'          ADDRESSES OF TWA TABLE
LPROGTBL DC   2F'0'          ADDRESSES OF PROGRAM TABLE
LEND    DC   X'FFFFFFFF'     END OF TABLE INDICATOR
      LTORG
      END
    
```

## Disable IN25LETX

CA InterTest for CICS uses IN25LETX during all monitoring; that is, for all entries of the CA InterTest for CICS monitoring table. To prevent CA InterTest for CICS from executing IN25LETX, you must disable it or remove it from the program library.

Once a link to IN25LETX fails (because the program is disabled), CA InterTest for CICS will not try the link again, thus avoiding much overhead. In such a case, the execution of IN25LETX will not resume until the next CA InterTest for CICS start.

## Password Protection

Online password protection is an optional CA InterTest for CICS feature. You can disable it selectively by redefining the password installation options CNTLSEC, CNTOSEC, CORSEC, and FILSEC. For instructions, see [IN25OPTS Keywords \(see page 56\)](#).

The IN25SEC2 macro is used to assemble a CA InterTest for CICS load module that contains the passwords for the CORE facility, FILE facility, and CNTL facility. The IN25SEC2 passwords are not stored anywhere in their unscrambled form. For secrecy, you can code PRINT OFF in your IN25SEC2 assembly. The default passwords are as follows:

- **MAIN**  
CORE
- **\$FUN**  
CNTL functions (commands)
- **\$OPT**  
CNTL options
- **\$MNO**  
CNTL options for segmented monitoring on and off (MON and NOM)
- **FILE**  
No default

Passwords can be changed at any time, even if someone is using the CORE or FILE facilities. Just run an IN25SEC2 assembly with the new passwords. If the CICS system is active, enter the following command to put the new passwords into effect:

```
CEMT SET PR0G(IN25SEC2) NEW
```

## Required Parameters

The IN25SEC2 macro is delivered in the CA InterTest for CICS source code file and must be unloaded from the distribution files before the assembly job.

```
IN25SEC2 TYPE=INITIAL,SCRNUM=(x,x,x,x)
```

The TYPE=INITIAL macro initiates the assembly and specifies the four numbers in the SCRNUM keyword that are to be used in your version of the scrambling algorithm. These numbers must be a value from 1 to 16, separated by commas, and enclosed in parentheses.

The TYPE=FINAL macro must be the last macro before the END statement. This macro must not contain any other parameters.

Between the TYPE=INITIAL and TYPE=FINAL macros, specify the IN25SEC2 macros to assign passwords to protected resources using the following keywords:

- **NAME**  
Specifies the name of the resource.
- **PASS**  
Specifies the four-character password for the resource. Only the characters A-Z, 0-9, \$, @, and # are allowed.

## Specify Passwords for the CORE Facility

The macro for the CORE password is as follows:

```
IN25SEC2 NAME=PROTCORE,PASS=xxxx
```

- **NAME**  
You must specify PROTCORE.
- **xxxx**  
Specifies the password.

When a user of the CORE facility receives the message ENTER SECURITY PASSWORD, the user must enter the four-character password. This password is then scrambled and compared with the scrambled password specified in the PASS= parameter. If the codes are equal, the CORE change is made and the password does not have to be entered again during the same session. If the password is entered incorrectly, the CORE facility asks for it again.



**Note:** Even if the password is entered correctly, areas outside CICS storage cannot be modified.

The following IN25SEC2 macro, delivered with the pregenerated version of CA InterTest for CICS, protects all storage not owned by a user's task:

```
IN25SEC2 NAME=PROTCORE,PASS=MAIN
```

- **PROTCORE**  
Specifies that this entry is for the CORE transaction.

- **MAIN**

The default password. To change or remove this password, update or remove the above macro and reassemble the IN25SEC2 program.

## Specify Passwords for Files and DL/I Databases

The macros for the FILE facility are:

```
IN25SEC2    NAME=yyyyyyyyy ,PASS=xxxx           WRITE only
IN25SEC2    NAME=yyyyyyyyy ,PASS=xxxx ,FPROT=YES  READ and WRITE
```

- **YYYYYYYY**

Specifies the name of the file ID (from the CICS file definition) or the DBD identification.

- **xxxx**

Specifies the password.

- **FPROT=YES**

Indicates that the file or DBD name should also be fetch protected; that is, data cannot be viewed or changed unless the password is correctly entered.

You must code one IN25SEC2 macro for every file or DBD name you want to protect; files or DBDs not named in a macro are not password protected against unauthorized use. These passwords apply only to access through the CA InterTest for CICS FILE facility.

When a user of the FILE facility receives the message ENTER SECURITY PASSWORD, the user must enter the four character password. This password is then scrambled and compared with the scrambled password specified in the PASS= parameter. If the codes are equal, the specified FILE or DBD is accessed. If the password is entered incorrectly, the FILE facility terminates.

You can code an IN25SEC2 macro to protect all files and DBD names that are not specifically protected in individual IN25SEC2 macros.

The following example password protects FILEA against both reads and writes with the password ASEC, password protects FILEB against writes only with the password BSEC, and password protects all other files against both reads and writes with the password FSEC.

```
IN25SEC2    NAME=PROTFILE ,PASS=FSEC ,FPROT=YES
IN25SEC2    NAME=FILEA ,PASS=ASEC ,FPROT=YES
IN25SEC2    NAME=FILEB ,PASS=BSEC
```



**Note:** Any security measures of your CICS installation are applied by CICS or the operating system after CA InterTest for CICS password protection is satisfied. The FILE facility functions as an application within CICS.XE "DL/I".

## Specify Passwords for CNTL Commands and Options

Password protection is available for certain CNTL commands and monitoring options.

To use CA InterTest for CICS to protect other commands and options or not protect the ones listed here, contact [CA Support \(http://support.ca.com/\)](http://support.ca.com/).

CNTL password protection does not apply to the following situations:

- Automatic startup of CA InterTest for CICS from the table assembled from CA InterTest for CICS PROMMAC macros.
- CNTL=RESTART command.
- CNTL=NEW command.
- CNTL=EXEC command.
- Startup of CA InterTest for CICS by the IN25PLT program in the PLT table.
- Execution of a CNTL command from a CRLP terminal; that is, a simulated terminal consisting of input and output sequential files.
- Execution of a CNTL command from the system console.

You do not need passwords in these situations because they either automatically start CA InterTest for CICS, which is controlled by the system programming staff, or reestablish functions that were password protected when they were originally requested.

### Password Protected CNTL Commands

The following commands are password protected by default if you use the pregenerated version of CA InterTest for CICS.

- **CNTL=CKPT**  
Start and end checkpoint
- **CNTL=END**  
Remove CA InterTest for CICS from CICS
- **CNTL=EXCL**  
Enter a program, transaction, or terminal in the Exclusion Table to prevent monitoring
- **CNTL=INCL**  
Remove a program, transaction, or terminal from the Exclusion Table
- **CNTL=ABP**  
Activate and deactivate the global automatic breakpoint facility
- **CNTL=PURGE**  
Start and end the purging of tasks that remained at a breakpoint longer than the specified time
- **CNTL=GLOG**  
Start and end logging of CNTL commands and CORE changes
- **CNTL=ON,PROG=.ALL**  
Declare monitoring for all application programs not specifically declared for monitoring
- **CNTL=OFF,PROG=.ALL**  
Remove monitoring of all application programs not specifically declared for monitoring

- **CNTL=ON,PROG=.ALL**  
Add monitoring options for all applications not specifically declared for monitoring
- **CNTL=ALL**  
Declare and remove monitoring for all applications not specifically declared for monitoring
- **CNTL=OFF**  
Removes monitoring options  
This command is password protected only to the extent that the option to be removed is password protected.

The following IN25SEC2 macro, delivered with the pregenerated version of CA InterTest for CICS, protects all of the previously listed CNTL commands:

```
IN25SEC2 NAME=PROT$FUN ,PASS=$FUN
```

- **PROT\$FUN**  
Identifies the CNTL commands listed previously. To change or remove this password, update or remove the above macro and reassemble the IN25SEC2 program.  
**Default:** \$FUN

## Password Protected CNTL Options

The following monitoring options are password protected by default; that is, if you use the pregenerated version of CA InterTest for CICS. You can disable password protection for these CNTL options by redefining the CA InterTest for CICS installation option CNTOSEC.

- **,BYP**  
Bypass storage protection or monitoring of BALR 14,15 and BALR 14,14 routines (for XA, BASSM, and BASR)
- **,CSA=(offset,length)**  
Do not protect specified areas of the CSA or CWA
- **,LET=name**  
Do not protect storage occupied by a specified program module
- **,LET=(address,length)**  
Do not protect a specified storage area
- **,MON=location**  
Set segmented monitoring on
- **,NOM=location**  
Set segmented monitoring off
- **,OVR=errcode**  
Disregard an error condition that would typically cause a breakpoint
- **,STR=ON**  
Preserve a copy of the CICS Trace Table for the duration of a breakpoint

- **,TON=\* or ,TON=terminal**  
Limit monitoring to a specific terminal
- **,USH=ON**  
Remove storage protection from storage areas in the shared subpool

### PROT\$OPT Name Option

The following IN25SEC2 macro, delivered with the pregenerated version of CA InterTest for CICS, protects all of the previously listed CNTL options except MON and NOM:

```
IN25SEC2 NAME=PROT$OPT, PASS=$OPT
```

- **PROT\$OPT**  
Identifies the password protected CNTL options.
- **\$OPT**  
The default password. To change or remove this password, update or remove the above macro and reassemble the IN25SEC2 program.

### PROT\$MNO Name Option

The following IN25SEC2 macro, delivered with the pregenerated version of CA InterTest for CICS, protects the segmented monitoring functions (MON and NOM) when MONOM=MENU or NOMENU, and MONOMSEC=YES:

```
IN25SEC2 NAME=PROT$MNO, PASS=$MNO
```

- **PROT\$MNO**  
Identifies the CNTL options MON and NOM.
- **\$MNO**  
The default password. To change or remove this password, update or remove the above macro and reassemble the IN25SEC2 program.

### Sample IN25SEC2 Assembly JCL

Sample JCL member CCSASEC2 is provided in CAI.CAVHJCL. Customize the member and submit it to assemble your password table.

The following example demonstrates the macro usage:

- The CORE password is changed to CORE.
- All files except FILE1 and FILE2 are WRITE protected with the password FSEC.
- FILE1 will be READ and WRITE protected with password WXYZ.
- FILE2 will be WRITE protected with password ABCD.
- CNTL commands and monitoring options are protected with the password CNTL.

```
//ASM.SYSIN DD *
IN25SEC2 TYPE=INITIAL, SCRNUM=(1, 1, 1, 1)
IN25SEC2 NAME=PROTCORE, PASS=CORE
IN25SEC2 NAME=PROTFILE, PASS=FSEC
```

```
IN25SEC2 NAME=FILE1, PASS=WXYZ, FPROT=YES
IN25SEC2 NAME=FILE2, PASS=ABCD
IN25SEC2 NAME=PROT$FUN, PASS=CNTRL
IN25SEC2 NAME=PROT$OPT, PASS=CNTRL
IN25SEC2 TYPE=FINAL
END
/*
```

## CA SymDump for CICS User Exits

This article describes CA SymDump for CICS user exits.

- [Global User Exit XDUREQ \(see page 113\)](#)
- [Configure SymDump for CICS Security \(see page 113\)](#)
- [External Security Installation Option \(see page 114\)](#)
- [Resource Protection Levels \(see page 115\)](#)
- [External Security Using CA Top Secret \(see page 115\)](#)
- [External Security Facility \(see page 117\)](#)
- [Security Exit Facility \(see page 120\)](#)

### Global User Exit XDUREQ

CA SymDump for CICS uses the CICS global user exit XDUREQ. If there are other products at your site that use this same exit, you will not have to make any modifications to CA SymDump for CICS.

However, be aware of the following:

- The exit modules get control in the order in which they are enabled. If another product uses the XDUREQ exit and is enabled before CA SymDump for CICS at PLT time, that product may change the storage associated with the abending transaction. CA SymDump for CICS does not modify task storage and is transparent to exit modules that are invoked after it has processed.
- The CA SymDump for CICS IN25INST program should always be placed after the CA InterTest for CICS IN25PLT program in the PLT.
- A return code from the XDUREQ global exit tells CICS whether to take the dump after the exit has processed it. If there are multiple exits and they pass different return codes, CICS ignores the return codes and processes the dump as if a normal return code were returned. This means that if CA SymDump for CICS and another product using the XDUREQ exit disagree about CICS taking its own dump, CICS takes the dump anyway.

### Configure SymDump for CICS Security

#### Secure CA SymDump for CICS

You can secure access to CA SymDump for CICS facilities by:

- Transaction-level security
- Menu-level security
- Dump-level security

## Transaction Level Security

You can apply transaction-level security to the CA SymDump for CICS transaction IDs using your site's external security package, such as CA Top Secret, CA ACF2, or IBM RACF. Transaction-level security applies when the user enters the transaction ID directly from CICS.



**Note:** Applying transaction-level security does not secure menu access to the equivalent functions. You may need to secure both.

The following table contains CA SymDump for CICS transaction IDs:

Transaction ID	Function	IN25OPTS Keyword
SYMD	CA SymDump Primary Option menu	SYMDET
SYME	Terminate (End) CA SymDump capture	
SYMI	Configure CA SymDump	
SYMS	Start CA SymDump capture facility	
SYMT	Capture a CICS internal trace	

Your site may have changed the transaction IDs using IN25OPTS. Use the VRPT transaction to display the current values online.

## MenuLevel Security

You can secure menu access to CA SymDump for CICS functions by using your site's external security package and the External Security Facility. Menu access applies whenever the user accesses the CA SymDump for CICS Primary Option menu from CICS (SYMD transaction) or from CA InterTest for CICS (ITST Option 5).

## DumpLevel Security

You can use a user-written security exit to control access to specific dumps and traces from either the online display or the batch print function. Dumps and traces meeting the criteria specified in the security exit are not displayed or printed.

## External Security Installation Option

The EXTSEC installation option in IN25OPTS turns the CA SymDump for CICS external security feature on or off. When CA SymDump for CICS is first installed, it is useful to leave EXTSEC set to No (the default) until the product is installed, tested, and the security rules written.

For more information about setting and modifying CA SymDump for CICS options, see [IN25OPTS Keywords \(see page 56\)](#).

## Resource Protection Levels

The resources protected in CA SymDump for CICS under the External Security feature are listed in the table below:

Class	Entity	Access Level	Function Type	CA SymDump Menu Options
CA@NTSYM	BQUSER	Read	User	1 Analysis (View dumps and traces)
CA@NTSYM	BQUSER	Update	Privileged user	1 Analysis (View plus Hold, Release, Delete)
CA@NTSYM	BQTRACE	Update	Trace authority	2 Trace capture
CA@NTSYM	BQADMIN	Update	Administrator	3 Configuration
CA@NTSYM	BQOPER	Update	Operator	4 Start 5 Stop

**BQUSER** with Read access lets the user access CA SymDump for CICS menu option 1, Dump Analysis. The user can view the list of dumps and traces, and then select individual dumps or traces to view. This user cannot hold, release, or delete dumps or traces.

**BQUSER** with Update access lets the user fully use CA SymDump for CICS menu option 1, Dump Analysis. This user can view the list of dumps and traces, select individual dumps or traces to view, and hold, release, and delete individual dumps or traces.

**BQTRACE** lets the user access CA SymDump for CICS menu option 2, Trace. This allows the user to request a CICS internal trace for later analysis. To view the trace, the user must also have BQUSER access with at least Read authority.

**BQADMIN** lets the user access CA SymDump for CICS menu option 3, Configuration. The administrator has access to the CA SymDump for CICS Configuration panel and the ability to set options for running CA SymDump for CICS.

**BQOPER** lets the user access CA SymDump for CICS menu options 4 and 5, which Start and Stop the CA SymDump for CICS capture facility, respectively.

To set up external security for your environment, continue by following the directions in the appropriate section:

## External Security Using CA Top Secret

The external security interface in CA SymDump for CICS is compatible with CA Top Secret 4.2 or above.

CA SymDump for CICS always performs its functions under the control of CICS. Therefore, no special definition of CA SymDump for CICS as a unique facility is required.

## Define the Resource Class CA@NTSYM

CA SymDump for CICS uses a unique resource class, CA@NTSYM, for its internal security controls.

### List a Resource Class

To determine if CA@NTSYM is currently defined to your CA Top Secret system, issue the following command:

```
TSS LIST(RDT) RESCLASS(CA@NTSYM)
```

This command lists the definition of the CA@NTSYM resource in the CA Top Secret Resource Definition Table (RDT).

- If CA@NTSYM is present and its definition matches the attributes below, no further action is required:
 

```
ACLST(UPDATE=6000,READ) DEFACC(READ)
```
- If any of the attributes do not match, you must change the definition using the TSS REPLACE(RDT) command, specifying the correct parameters.
- If you get an error message indicating that the resource is missing, invalid, or reserved, you must define it.

If CA@NTSYM is not defined, you must add it to the RDT. New classes are added using the TSS command ADDTO function through any facility supporting the TSS command. Once added, these resources become a permanent part of the security file and can be immediately administered and reported on by name.

#### Add a Resource Class

The syntax to add the CA@NTSYM resource is shown next. The syntax uses a RESCODE of 38. This code value is up to each site and must be unique. Codes 01 to 3F are available.

```
TSS ADDTO(RDT) RESCODE(38) RESCLASS(CA@NTSYM) -
  ACLST(UPDATE=6000,READ) DEFACC(READ)
```

Once you define the resource CA@NTSYM to your system, you can establish access controls to CA SymDump for CICS functions and information. No IPL, re-initialization, or parameter changes are needed for these changes to become active and permanent.

### Use CA@NTSYM to Control CA SymDump for CICS Functions

The CA@NTSYM resource controls all CA SymDump for CICS functions. CA@NTSYM is a standard short resource and is managed in the same manner as other access level resources. For a definition of the access levels by entity name, see Resource Protection Levels.

To define the CA SymDump for CICS functions to CA Top Secret, use the following command:

```
TSS ADDTO(deptname)
(BQUSER,BQTRACE,BQADMIN,BQOPER)
```



**Note:** Notice that all entity names start with BQ to uniquely identify them as CA SymDump for CICS commands. The rest of the entity name identifies its function, USER, TRACE, ADMIN, or OPER(ator).

#### Define User Access to BQ Functions

To allow user USER01 to have Read access to Option 1 of the CA SymDump for CICS menu to analyze a dump or trace, use the following command:

```
TSS PERMIT(USER01) CA@NTSYM(BQUSER) ACCESS(READ)
```

To allow user USER02 to have full (update) access to Options 1 and 2 of the CA SymDump for CICS menu to view, hold, release, and delete dumps and traces, and to capture a trace, use the following commands:

```
TSS PERMIT(USER02) CA@NTSYM(BQUSER) ACCESS(UPDATE)
TSS PERMIT(USER02) CA@NTSYM(BQTRACE) ACCESS(UPDATE)
To allow user USER99 to access all functions of <productname>:
TSS PERMIT(USER99) CA@NTSYM(BQUSER) ACCESS(UPDATE)
TSS PERMIT(USER99) CA@NTSYM(BQTRACE) ACCESS(UPDATE)
TSS PERMIT(USER99) CA@NTSYM(BQADMIN) ACCESS(UPDATE)
TSS PERMIT(USER99) CA@NTSYM(BQOPER) ACCESS(UPDATE)
```

If you omit the ACCESS keyword, CA Top Secret defaults the PERMIT to Read access.

## External Security Facility

CA SymDump for CICS external security provides menu-accessed resource protection and control through the use of your site's external security package, such as CA Top Secret, CA ACF2, or IBM RACF. The external security option provides security for the following:

- Protection for the configuring, starting, and stopping of CA SymDump for CICS through CA SymDump for CICS menu options 3, 4, and 5, respectively
- Protection for viewing, holding, releasing, and deleting dumps through CA SymDump for CICS menu option 1
- Control over whom is able to capture a CICS internal trace through CA SymDump for CICS menu option 2

CA SymDump for CICS checks authorization at the time of access, rather than checking authorization at signon. This enables your site to track security violations.

Protecting menu access to CA SymDump for CICS functions does not secure direct access to the same functions using a CICS transaction ID. To do this, you must secure the equivalent transaction IDs as discussed in *Securing CA SymDump for CICS*.

## External Security Using CA ACF2

The external security interface in CA SymDump for CICS is compatible with CA ACF2 6.2 or higher.

## Define the Resource Class CA NTSYM

CA SymDump for CICS uses a unique resource class, CA@NTSYM, for its internal security controls. CA@NTSYM needs to be defined to CA ACF2 with its own three-character resource type. In the following example, CA@NTSYM is defined using the resource type CAM.

```
SET CONTROL(GSO) INSERT CLASMAP.CA@NTSYM -
RESOURCE(CA@NTSYM) RSRCTYPE(CAM) ENTITYLN(8)
```

After adding this clasmap definition, you must refresh the GSO clasmap.

To protect and allow access to CA SymDump for CICS, use the standard CA ACF2 rule writing, as shown below:

```
SET RESOURCE(CAM)
COMPILE *

$KEY(entity) TYPE(CAM)
UID(uidstring/mask) SERVICE(READ/READ,UPDATE/UPDATE) -
    ALLOW/LOG/PREVENT
END
STORE *
```

If the SERVICE keyword is not supplied, CA ACF2 defaults to permit access to all service levels.

Once the resource CA@NTSYM is defined to your system, you can establish access controls to CA SymDump for CICS functions and information.

### Use CA@NTSYM to Control CA SymDump for CICS Functions

The CA@NTSYM resource controls all CA SymDump for CICS functions. For a definition of the access levels by entity name, see Resource Protection Levels.



**Note:** Notice that all entity names start with BQ to uniquely identify them as CA SymDump for CICS commands. The rest of the entity name identifies its function, USER, TRACE, ADMIN, or OPER(ator).

```
* Define access for CA@NTSYM
Set RESOURCE(CAM)
COMPILE *
* Define access for <productname> analysis
$KEY(BQUSER) TYPE(CAM)
* Set general user read-only access
UID(general user UID) SERVICE(READ) ALLOW
* Set privileged user read and update access
UID(privileged user UID) SERVICE(READ,UPDATE) ALLOW
* Define access for <productname> tracing
$KEY(BQTRACE) TYPE(CAM)
UID(tracing user UID) SERVICE(UPDATE) ALLOW
* Define access for <productname> Administration
$KEY(BQADMIN) TYPE(CAM)
UID(admin user UID) SERVICE(UPDATE) ALLOW
* Define access for <productname> operator
$KEY(BQOPER) TYPE(CAM)
UID(operator user UID) SERVICE(UPDATE) ALLOW
END
STORE *
```

### External Security Using IBM RACF

The external security interface in CA SymDump for CICS can be used with RACF. It is compatible with all supported versions of CA Common Services for z/OS.

### Required RACF Table Entries

Before implementing RACF security, you must define a new class for CA SymDump for CICS (CA@NTSYM) in the Class Descriptor Table, update the Router Table for the new class, and then enable the class. These steps are detailed next.

1. Class Descriptor Table (CDT) Entry for CA@NTSYM  
Define a new class, CA@NTSYM, for CA SymDump for CICS security in the RACF Class Descriptor Table (module ICHRRCDE). The following information should be included in the ICHERCDE macro entry:  
CA@NTSYM-This is referred to in other sections as CA@NTSYM.  
MAXLNTH=8, FIRST=ALPHANUM, OTHER=ALPHANUM.



**Note:** No recommendations are given for any DFTUACC; this is completely determined by each site.

2. Router Table Entry for CA@NTSYM  
Make an entry for the CA@NTSYM class in the RACF Router Table (module ICHRFR01). Specify ACTION=RACF. This entry is generated by the ICHRFRTB macro.
3. Enabling the Class CA@NTSYM  
Activate the new class CA@NTSYM just defined in the CDT and Router Table. To do this, issue the following command:  
SETROPTS CLASSACT(CA@NTSYM)

## Protect CA SymDump for CICS Entities

The CA@NTSYM resource controls all CA SymDump for CICS functions. For a definition of access levels by entity name, see Resource Protection Levels earlier in this section.



**Note:** Notice that all entity names begin with BQ to uniquely identify them as CA SymDump for CICS commands. The rest of the entity name identifies its function, USER, TRACE, ADMIN, or OPER(ator).

### Issue RACF RDEFINES for CA SymDump for CICS Entities

First, you must issue the RACF RDEFINE command for every CA SymDump for CICS entity that you want to protect. For example, issue the following commands to protect the CA@NTSYM resources:

```
RDEFINE CA@NTSYM BQUSER UACC(NONE)
RDEFINE CA@NTSYM BQTRACE UACC(NONE)
RDEFINE CA@NTSYM BQADMIN UACC(NONE)
RDEFINE CA@NTSYM BQOPER UACC(NONE)
```

Give all entities a universal access of NONE. Defining this security type denies access to anyone that does not have authorization.

## Permit the Use of CA@NTSYM Entities

Once you protect the entities, use the PERMIT command to give different levels of access to individual users for separate CA SymDump for CICS functions. For example:

- Define a general user with read-only access to the analysis panels by issuing:

```
PERMIT BQUSER CLASS(CA@NTSYM) ID(userid) ACCESS(READ)
```

- Define a privileged user with read and update access to the analysis panels by issuing:

```
PERMIT BQUSER CLASS(CA@NTSYM) ID(userid) ACCESS(READ,UPDATE)
```

- Define access for CA SymDump for CICS trace capture function by issuing:

```
PERMIT BQTRACE CLASS(CA@NTSYM) ID(userid) ACCESS(UPDATE)
```

- Define access for CA SymDump for CICS administration (configuration panel) by issuing:

```
PERMIT BQADMIN CLASS(CA@NTSYM) ID(userid) ACCESS(UPDATE)
```

- Define access for CA SymDump for CICS operator (start and stop functions) by issuing:

```
PERMIT BQOPER CLASS(CA@NTSYM) ID(userid) ACCESS(UPDATE)
```

## Security Exit Facility

CA SymDump for CICS provides an optional security exit facility. This is an alternative to using the External Security Facility. The security exit facility allows a user-written security exit to control access to dumps from the online dump display function and the batch print function.

The security exit receives control during online and batch processing when dumps that meet specified selection criteria are found. In effect, the security check is another selection criterion. No security violation message appears; instead, dumps for which the user is not authorized are not shown.

Both online and batch exits are optional and can be written in Assembler, COBOL, or PL/I. The CA SymDump for CICS SMP target libraries contain sample COBOL and Assembler exit programs in both source and load form. PL/I samples are not provided.

### Activate the Exit Facility

To activate the exit facility, you need only provide an active CICS exit module named IN25XC1 and an active batch exit module named IN25XB1.

Use the following example to activate the CICS Assembler sample exit.

#### Follow these steps:

1. Rename supplied load module IN25XC1A to IN25XC1.
2. Restart your CICS region or bring in new copies of IN##XCMR and IN25XC1, where ## specifies your two-digit CICS release number (for example, 66 for CICS 4.1, 67 for CICS 4.2, 68 for CICS 5.1, 69 for CICS 5.2, 70 for CICS 5.3, and 71 for CICS 5.4).

Use the following example to activate the batch Assembler sample exit.

#### Follow these steps:

1. Rename supplied load module IN25XB1 to IN25XB1D.
2. Rename supplied load module IN25XB1A to IN25XB1.

- Submit your execution JCL for the batch dump print program IN##PDMP where ## specifies your two-digit CICS release number (for example, 66 for CICS 4.1, 67 for CICS 4.2, 68 for CICS 5.1, 69 for CICS 5.2, 70 for CICS 5.3, and 71 for CICS 5.4). Sample JCL for execution of this program is member SYMPDMP in the CAVHJCL library.

If you do not want to install the exits immediately, you can leave the exit load modules. The user exit manager for CICS handles not finding the exit. The user exit manager for batch uses the dummy module provided.

The CA SymDump load library contains the following modules related to the security exit:

Module	Description
IN##XCMR	CA SymDump CICS user exit manager
IN25XBMR	CA SymDump batch user exit manager
IN25XC1A	Sample CA SymDump CICS security user exit (Assembler)
IN25XC1C	Sample CA SymDump CICS security user exit (COBOL)
IN25XB1A	Sample CA SymDump batch security user exit (Assembler)
IN25XB1C	Sample CA SymDump batch security user exit (COBOL)
IN25XB1	CA SymDump batch security user exit (dummy stub)

The CAVHSAMP target library also provides source code for the sample user exits.

## Test the Exit Facility

The samples are set up to allow access to dumps with the seconds in the time field less than 30, and to disallow the others. Try accessing dumps first without the exits, and then rename the sample exits to the proper names and try accessing dumps again (be sure to bring in new copies of IN##XCMR and IN25XC1 first). Fewer dumps are allowed to you. Trying this verifies the basic functioning of the exits.

## Online Dump Display

The CICS user exit is invoked separately for each dump. The exit gets control through a standard Command Level link and is passed a COMMAREA. This COMMAREA contains:

- A field containing the total length of the area
- A return code field
- Product and function identification fields
- All the identification data for the dump as shown on the Dump/Trace Analysis and Selection screens (such as CICS applid, CICS user ID, program, transaction ID, abend code, terminal ID, language, and date and time)

The copy member for the COMMAREA used by the exit is:

- SDMPXCMA for the Assembler exit
- SDMPXCMC for the COBOL exit

## Batch Print

The batch version of the user exit is linked to when dumps that meet the specified selection criteria are found. The user exit is invoked for both print and index functions so that the list of dumps available to a user is always identical.

The batch user exit is invoked separately for each dump. The exit gets control through a standard OS link and is passed a standard parameter list with one entry that is identical in format to the COMMAREA described in Online Dump Display in this section.

## Exit Processing

For both online and batch exits:

- A zero return code indicates that the user's request is authorized
- A nonzero return code indicates that access is denied

# Allocate Files

This article describes the required and optional file allocation steps needed to configure your product. All products must allocate the PROTSYM file. File allocations specific to each product are described after the PROTSYM information.

- [Allocate PROTSYM \(see page 122\)](#)
- [Populate the Symbolic File \(see page 123\)](#)
- [Update Your In-house Compile and Assemble Procedures \(see page 123\)](#)
- [CA InterTest Batch Files \(see page 123\)](#)
- [CA SymDump Batch \(see page 125\)](#)
- [CA InterTest for CICS and CA SymDump for CICS \(see page 126\)](#)
- [CA InterTest for CICS \(see page 126\)](#)
- [CA SymDump for CICS \(see page 127\)](#)

## Allocate PROTSYM

Member CAVHPROT creates the PROTSYM VSAM file, which contains the symbolic information and source listings for your application programs. Edit the JCL in the CAVHJCL library to conform to your installation standards and the previously completed worksheet.

Note the following items when creating the symbolic file:

- Do not specify a secondary space allocation.
- The CAVHPROT utility job (running the IN25UTIL program) could run for a long time depending on the size of the file. This occurs only the first time after the file is created by the IDCAMS job.
- Each symbolic file has an upper limit of about four million 2 KB records and holds approximately 147,000 programs.

- The symbolic file must not reside in the LSR pool.

## Concurrently Updating the PROTSYM File

The application uses the MVS RESERVE and DEQ macros to allow sharing of the PROTSYM file between regions or systems.

The resource major name used in the RESERVE and DEQ macros is INTERTST. Place the major name INTERTST into an MVS service that converts the RESERVE into a cross-system ENQ if this service is available at your site.

## Populate the Symbolic File

This section applies only to CA InterTest for CICS and CA SymDump for CICS users.

If your site previously loaded a PROTSYM file with the symbolic information for CICS DSECTs and CA InterTest for CICS and CA SymDump for CICS DSECTs, you may skip this step.

Member CABALSYM, found in the CAVHJCL library, populates the PROTSYM VSAM file with symbolic information for CICS DSECTs and CA InterTest and SymDump for CICS DSECTs. Edit the JCL to conform to your installation standards.



**Note:** Symbolic information and source listings for the demonstration programs will be loaded into the PROTSYM file when you prepare the programs for execution. For more information, see [Prepare the Demonstration Programs \(see page \)](#).

## Update Your In-house Compile and Assemble Procedures

To update the compile and assemble procedures and create symbolic information for your Assembler, PL/I, or COBOL programs, see [Symbolic Support](https://docops.ca.com/display/CAITSD11/Symbolic+Support). (<https://docops.ca.com/display/CAITSD11/Symbolic+Support>)

## CA InterTest Batch Files

### Allocate PROFLIB

You must allocate a profile data set before executing the application. CAVHPROC library member CAMRPROF contains the sample allocation for the profile data set. PROFLIBs can be allocated per user or per site. Each user requires one block and one directory entry in the PROFILE library to maintain session variables. Allocate a number of blocks equal to the number of programmers who will access the application. There are three users per directory block. The number of directory blocks should be changed if the number of users is greater than 90.

The parameters you need for this JCL are as follows:

- **TGTVOL=**,  
VOLSER of the PROFLIB
- **PREFIX='USER.'**,  
The USERID high-level qualifier (default = USER)
- **PERMDA=SYSDA**  
The generic UNIT for permanent DASD (default = SYSDA)



**Note:** During execution, each user requires WRITE access to the PROFILE library only. Secure all other required application files with READ access only. Be sure to give similar attention to the optional files that the application uses (for example, the session log).

## Allocate ALIBs (Optional)

The product uses an ALIB data set to save allocation information. Users can have any number of ALIBs at their disposal, but they must be created manually.

CAVHPROC library member CAMRALIB creates and catalogs an ALIB. The parameters you need for this JCL are as follows:

- **TGTVOL=**,  
The VOLSER of the target data sets
- **PREFIX='USER.'**,  
The USERID high-level qualifier (default = USER)
- **PERMDA=SYSDA**  
The generic UNIT for permanent DASD

## Allocate INCLIBs (Optional)

If you intend to use the INCLUDE command, you must allocate the INCLUDE library. CAVHPROC library member CAMRINCL allocates this library.

The parameters you need for this JCL are as follows:

- **TGTVOL=**,  
VOLSER of the target data set
- **PREFIX='USER.'**,  
The SITE/USERID high-level qualifier (default = USER)
- **PERMDA=SYSDA**  
The generic UNIT for permanent DASD (default = SYSDA)

# CA SymDump Batch

## Allocate CAIPRINT Repository Library

Make a copy of *CAI.CAVHJCL(CARXREPO)* and update it to conform to your installation standards.

Use *CARXREPO* to create and initialize a new CAIPRINT repository library. The data set name for your CAIPRINT repository is defined in the *PRTLIB* option of *CAOETABL*. This data set name will be defined in the next step, . Specify the same *dsname* when customizing *CARXREPO*.

*CARXREPO* consists of the following two steps:

Step	Program	Description
1	IDCAMS	Allocate the CAIPRINT repository.
2	CAOUPRTU	Initialize the CAIPRINT repository.



**Note:** RESERVE and DEQ macros are used when updating the CAIPRINT repository to enable sharing of files between regions and systems. The resource major name used in the RESERVE and DEQ macros is CAOUPRTA. If your installation uses a service that converts RESERVEs into cross-system ENQs, define the major name CAOUPRTA to the service.

### Estimated Size of the CAIPRINT Repository Library

An individual CAIPRINT report's size varies greatly depending on the number of active programs, the size of the data in each active program, the number of allocated files, and so on. Because of this, it is difficult to determine how much space to allocate for the repository.

As a guideline, using *CAOEDEMO* and a repository of one cylinder, you can store 17 reports in the repository. This report contains only one active COBOL program, but contains a large data division of approximately 500 KB.

Because of the number of factors that affect the determination of the size of your repository, we suggest that you determine the amount of space you want to dedicate to saving abend reports. You can migrate the reports to a larger repository later using the *CAOUPRTU* utility *COPY* function.



**Note:** Do not specify a secondary space allocation when creating the repository because the capacity of the repository is fixed by the allocation job and cannot be increased using secondary extents.



**Note:** The LE option TERMTHDACT(UADUMP) generates a U4039 abend when an unhandled condition occurs of severity 2 or greater. CA SymDump Batch will not report on this abend.

## CA InterTest for CICS and CA SymDump for CICS

### Install the Help Facility

The member DEFHELP, found in the CAVHJCL library, creates and loads the VSAM file that contains the CA InterTest and SymDump for CICS online help information. Edit the JCL to conform to your installation standards and the previously completed worksheet. This member and other referenced members are found in the CAI.CAVHJCL library after the APPLY step.

### Create the User-Defined Abend Description File

The member DEFUHLP, found in the CAVHJCL library creates the VSAM file that contains the user-defined abend description information. Edit the JCL to conform to your installation standards and the previously completed worksheet. This member and other referenced members are found in the CAI.CAVHJCL library after the APPLY step.

## CA InterTest for CICS

### Create the Debug Session File

The member DEFMLOG, found in the CAI.CAVHJCL library creates the file that is used to record breakpoints and keeps set when monitor logging is turned on. These records can later be loaded for another debug session without having to set breakpoints and keeps again. Edit the JCL to conform to your installation standards. If you decide to use the optional PROTMLLOG feature, then a unique MLOG dataset is required for each CICS region.



**Note:** For more information on the Monitor Logging facility, see [Monitoring Menu Options \(https://docops.ca.com/display/CAITSD11/Monitoring+Menu+Options\)](https://docops.ca.com/display/CAITSD11/Monitoring+Menu+Options).

### Create the Checkpoint File

The member DEFCKPT, found in the CAVHJCL library creates the checkpoint file that is used to periodically record the status of CA InterTest for CICS for a possible restart. This member and other referenced members are found in the CAI.CAVHJCL library. Edit the JCL to conform to your installation standards.



**Note:** For more information on the checkpoint recording facility, see [Monitoring Menu Options \(https://docops.ca.com/display/CAITSD11/Monitoring+Menu+Options\)](https://docops.ca.com/display/CAITSD11/Monitoring+Menu+Options).

## CA SymDump for CICS

### Create the PROTDMP File

The CA SymDump for CICS VSAM file, PROTDMP, is the CA SymDump for CICS dump data set. This file contains all of the dumps that will be viewed and maintained with CA SymDump for CICS. PROTDMP files that were initialized using a release of CA SymDump for CICS after r8.0 are compatible with this release. For information on compatibility with CICS releases, see [CICS Interfaces and Compatibility \(see page 150\)](#).

Use member SYMFILE in CAI.CAVHJCL to create and initialize the file with program IN25INIT. Be aware of the following:

- The maximum number of dumps the CA SymDump for CICS dump data set can hold depends on the CI (Control Interval) size. For more information, see the SYMFILE member in the CAVHJCL.
- Do not specify a secondary space allocation.
- The SYMFILE job may run for a long period of time depending on the size of the file.
- Two or more concurrently active CICS regions cannot share the same CA SymDump for CICS PROTDMP file for update.



**Note:** You cannot run IN25INIT on a previously initialized PROTDMP file. IN25INIT can only be run against a newly defined VSAM file.

### Estimating the Size of the PROTDMP File

The size of an individual dump varies greatly depending on the number of active programs, the size of the data in each active program, and so on. Because of these issues, it is difficult to determine how much space to allocate for the PROTDMP file.

Therefore, CA suggests that you determine the amount of space you are willing to dedicate to saving dumps. Dumps can always be migrated at a later date to a larger PROTDMP file using the COPY function of the IN25DMPU utility.



**Note:** Do not specify a secondary space allocation when creating the PROTDMP file because the capacity of the PROTDMP file is fixed by the allocation job and cannot be increased using secondary extents.

# Interfaces and Compatibility

This article describes considerations for using your CA InterTest and CA SymDump products with other products and interfaces. This article contains the considerations for all products. For more information on CA InterTest Batch and CA SymDump Batch, see [Batch Interfaces and Compatibility \(see page 130\)](#). For more information on CA InterTest for CICS and CA SymDump for CICS, see [CICS Interfaces and Compatibility \(see page 150\)](#).

- [Configure Dynamic Symbolic Support \(see page 128\)](#)
- [Configure Dynamic Symbolic Support for CA SymDump Batch \(see page 130\)](#)
- [Configure Dynamic Symbolic Support for CA InterTest and SymDump for CICS \(see page 130\)](#)

## Configure Dynamic Symbolic Support

The dynamic symbolic support feature improves your productivity by reducing the pre-testing setup time commonly involved in locating and loading a program's compiler or assembler listing. This feature automatically locates the listing and loads the symbolic file. This feature is fully integrated with CA Endeavor SCM.

You can implement the dynamic symbolic support feature if the CA Endeavor SCM footprint (FOOTPRNT) feature is used when applications programs are generated in CA Endeavor SCM.

### Configure the Started Task Procedure for CA Endeavor support

This configuration procedure is required in the following situations:

- If you want to use Dynamic Symbolic Support for CA InterTest CICS or SymDump CICS.
- If you are not using any of the CICS products and you use multiple CA Endeavor SCM C1DEFLT5 which are not in your STEPLIB(s). For more information, see [Batch Link Facility \(https://docops.ca.com/display/CAITSD11/Batch+Link+Facility\)](https://docops.ca.com/display/CAITSD11/Batch+Link+Facility), or [Symbolic Utilities \(https://docops.ca.com/display/CAITSD11/Symbolic+Utilities\)](https://docops.ca.com/display/CAITSD11/Symbolic+Utilities).

#### Follow these steps:

1. Install CAICCI as discussed in the section CA Common Services for z/OS and Requirements.
2. Copy the CAVHSAMP member INDVSPWN to the CA ENF parameter library and add the spawn parameter to the CA ENF spawn parameter list.
3. Use the CAVHPROC member INTNDVR, the started task procedure, to access CA Endeavor SCM to dynamically load program symbolic information.
4. Modify this procedure to conform to your installation standards.
5. Copy it to a system procedure library as member INTNDVRn. The member name of the procedure should have the CA Endeavor SCM C1DEFLT5 table site ID appended to it. For example, if the site ID is 3, the procedure member name should be INTNDVR3.
6. Repeat step 5 for each unique site ID.

**Notes:**

- This started task must be an APF-authorized task.
- The INTNDVRn procedures require read security access to any load library data sets in which application programs that were created by CA Endeavor SCM reside. They also require write security access to any PROTSYM files they will populate.
- CA ENF will need to be refreshed to activate the new spawn parameters added for this feature.

## Shut Down CA Endeavor SCM Dynamic Symbolic Support Started Task

The CA SymDump for CICS Dynamic Symbolic Support started task is simply canceled with the CANCEL operator command when shutdown is required.

## Configure Dynamic Symbolic Support for CA InterTest Batch

If you changed the procedure name from the default INTNDVR, you must also change and reapply USERMOD for the IN25SITE table. Edit CAVHJCL member IN25SITE.

Following is a sample IN25SITE program:

```
*****
* THIS TABLE IS USED TO SUPPORT CUSTOMER'S ENVIRONMENT *
* WHERE MULTIPLE C1DEFLT ARE USED. *
* *
* THE TABLE CONTAINS AN ENDEVOR SITE ID AND THE ASSOCIATED *
* PROCNAME TO START LISTING SERVER AS A STARTED TASK TO *
* RETRIEVE LISTINGS STORED UNDER ENDEVOR CONTROL. *
* *
* THIS TABLE MUST BE MODIFIED VIA SMP/E USRMOD IF THE *
* CUSTOMER HAS MORE THAN ONE C1DEFLT. A SINGLE C1DEFLT *
* SITE, DOES NOT NEED TO MODIFY THIS TABLE. *
* *
* THIS MODULE MUST BE ASSEMBLED AND LINKAGE EDITED INTO *
* CA INTERTEST BATCH LOAD MODULE LIBRARY. *
* *
* MACRO @NDVRSTE IS USED TO GENERATE THIS TABLE. *
* *
* MACRO FORMAT: *
* @NDVRSTE PROC= *
* PROC = A MAXIMUM OF 7 CHARACTER PROC NAME. *
* PROC = BLANKS FOR SINGLE C1DEFLT SITE. *
* *
* NOTE: CA INTERTEST WILL APPEND THE SITE ID OBTAINED FROM *
* ENDEVOR FOOTPRINT AS THE 8TH CHARACTER. THUS THE *
* PROCNAME IN YOUR PROCLIB IS EIGHT CHARACTERS. *
*****
IN25SITE CSECT ,
IN25SITE AMODE 31
IN25SITE RMODE ANY
*
@NDVRSTE PROC=
* @NDVRSTE PROC=INTNDVR
* END ,
```

The table is generated using macro @NDVRSTE. The macro has one parameter PROC, which is explained as follows:

A maximum seven-character name of a cataloged procedure used to start the listing server started task (STC). The eighth character is reserved as a suffix to be appended by the product based on your site ID. The name defined in the PROC keyword parameter must match the procedure name described above.

The JCL is included in the configuration PTF that assembles and links this program. The JCL member is IN25SITE and it is found in the CAVHJCL data set. The JCL must be modified JCL to conform to their installation standards before submitting it.

## Configure Dynamic Symbolic Support for CA SymDump Batch

The first seven-character prefix of the procedure member name you copied to the system procedure library must match the value defined in the NDVRPROC option in CAOUDFRX.

## Configure Dynamic Symbolic Support for CA InterTest and SymDump for CICS

- The seven-character prefix for this procedure name is specified and can be changed by editing the NDVRPROC parameter in the IN25OPTS macro.
- Include an NDVRSYM ddname in the IN25OPTS SYMFIL= parameter to indicate the PROTSYM file to be used for dynamic symbolic loading. The presence of an NDVRSYM PROTSYM ddname in the CICS startup JCL indicates to CA InterTest and SymDump for CICS to invoke the dynamic symbolic load option.

```
SYMFIL=(PROTSYM,xxxxxxx,xxxxxxx,NDVRSYM)
```

- If you use multiple CA Endeavor SCM C1DEFLT5 in your environment.

## Batch Interfaces and Compatibility

This article contains interface and compatibility considerations for CA InterTest Batch and CA SymDump Batch.

- [CA InterTest Batch \(see page 131\)](#)
  - [Customize CLISTs \(see page 131\)](#)
  - [Customize Compile Procedures \(see page 135\)](#)
  - [Add to ISPF Main Panel \(Optional\) \(see page 137\)](#)
  - [Add ISPF Libraries to TSO Logon Procedure \(Optional\) \(see page 137\)](#)
  - [Customize CA Roscoe \(Optional\) \(see page 137\)](#)
  - [Review and Customize for BTS \(see page 139\)](#)
  - [Customize for DB2 SP and IMS/DC Debugging \(see page 141\)](#)
- [CA SymDump Batch \(see page 143\)](#)

- [Bind DB2 Support Member \(see page 143\)](#)
- [Abend Reporting with Language Environment \(LE\) \(see page 143\)](#)
- [Review Language Environment \(LE\) Considerations \(see page 143\)](#)
- [Establish Access to the TSO Facility \(see page 144\)](#)
- [Log and Notify Facilities \(see page 146\)](#)
- [Use CA Netman Interface \(see page 149\)](#)

## CA InterTest Batch

### Customize CLISTS

You may need to customize a number of CLISTS provided in the CAVHCLS0 library to run your product. These CLISTS are as follows:

- CAMRDRVR
- CAMRCLST
- CAMRCIMS
- CAMRCBTS
- CAMRCDB2
- CAMRSTRT
- CAMRKEYS
- CAMRCMD

### CAMRDRVR

Member CAMRDRVR is a sample CLIST that acts as a driver for the four startup CLISTS. The following modifications should be made if you want to use this driver:

- Modify the PREFIX to match the prefix of the installed data sets.
- Under */\*Note2\*/*, uncomment the library allocations that are *not* allocated in your TSO logon procedure or other CLIST.

### CAMRCLST

Member CAMRCLST is a sample CLIST showing necessary file allocations for the application. The following modifications should be made:

- Modify the PREFIX to match the prefix of the installed data sets.
- Under */\*Note2\*/*, uncomment the option that pertains to your installation. Use ISPLLIB for the LE run-time library if it is different from the default (linklist).

Select per-site or per-user PROFLIB. In the sample CLIST, the first allocation statement after `/*Note3*/` is for those that want to allocate one PROFILE data set per site. The second allocation statement after `/*Note3*/` is for those that want to allocate a PROFILE data set per-user. The third statement after `/* Note3 */` is a CLIST that can create a per-user PROFILE data set if one does not exist and then allocate it. Note that you may have to tailor these statements or the CLIST to meet your specific installation requirements.

- Under `/*Note4*/`, uncomment the optional files you are using with the CA InterTest Batch. The optional files are as follows:
  - **INT1CLIB**  
Defines the partitioned data set that contains CA InterTest Batch commands used by the INCLUDE command. If it is not allocated, the INCLUDE command is disabled. For more information about the INCLUDE command, see [Batch Debugging \(https://docops.ca.com/display/CAITSD11/Batch+Debugging\)](https://docops.ca.com/display/CAITSD11/Batch+Debugging). CAI.CAVHSAMP is the data set that includes the members needed for executing the demos (these members must be copied if you define another name for INT1CLIB and you want to run one of the demos). Uncomment the ALLOC statement for INT1CLIB to allocate the default CAI.CAVHSAMP data set. Alternatively, you can execute the CLIST INT1CLIB, the next comment in the CLIST, to create a personal include library and allocate it to INT1CLIB.
  - **INT1CLOG**  
Defines a sequential file used for the CA InterTest Batch session log. If it is not allocated, logging is disabled. For more information about the session log, see [Batch Debugging \(https://docops.ca.com/display/CAITSD11/Batch+Debugging\)](https://docops.ca.com/display/CAITSD11/Batch+Debugging). Uncomment the first ALLOC DD (INT1CLOG) statement if you want to write your LOG data to a pre-allocated data set. You must create the data set yourself. Uncomment the second ALLOC statement if you want your LOG data written to SYSOUT. Uncomment the CLIST INT1CLOG to create a personal LOG data set and allocate INT1CLOG to it.
  - **INT1REPT**  
Defines a sequential file used to store the HISTogram and XSUM reports. If it is not allocated, the HIST and XSUM commands are disabled. For more information about the reports, see the HIST and XSUM Report Command descriptions in [Batch Debugging \(https://docops.ca.com/display/CAITSD11/Batch+Debugging\)](https://docops.ca.com/display/CAITSD11/Batch+Debugging). Uncomment the first ALLOC DD(INT1REPT) statement if you want to write your histogram data to a pre-allocated data set. You must create the data set yourself. Uncomment the second ALLOC statement if you want your histogram data written to SYSOUT. Uncomment the CLIST INT1REPT to create a personal histogram data set and allocate INT1REPT to it.
  - **INT1ALIB**  
Defines the partitioned data set that contains allocations saved under option 3, ALLOCATION. If it is not allocated, the save and retrieve features are disabled. For more information on allocating data sets and saving the allocations, see [Allocations Facility for ISPF \(https://docops.ca.com/display/CAITSD11/Allocations+Facility+for+ISPF\)](https://docops.ca.com/display/CAITSD11/Allocations+Facility+for+ISPF). Uncomment the CLIST INT1ALIB if you want to dynamically create and allocate a personal allocation library.

## CAMRCIMS

CAMRCIMS is a sample CLIST showing the necessary file allocations for the foreground testing of IMS /DB or DL/I programs. If you are licensed for the IMS option and want to debug these types of applications in foreground, the following modifications should be made:

- Modify the PREFIX and IMS to match the prefix of the installed application and IMS data sets.
- Under `/*Note2*/`, uncomment the option that pertains to your installation. Use ISPLLIB for the LE run-time library if it is different from the default (linklist).



**Note:** There are two sections marked `/*Note2*/` in this CLIST.

Select per-site or per-user PROFLIB. In the sample CLIST, the first allocation statement after `/*Note3*/` is for those that want to allocate one PROFILE data set per site. The second allocation statement after `/*Note3*/` is for those that want to allocate a PROFILE data set per-user. The third statement after `/* Note3 */` is a CLIST that can create a per-user PROFILE data set if one does not exist and then allocate it. Note that you may have to tailor these statements or the CLIST to meet your specific installation requirements.

- Under `/*Note4*/`, uncomment the optional files you are using with your product. See the optional files listed under CAMRCLST for a description of the optional files.
- Under `/*Note5*/`, add any other IMS files needed for your application.

This CLIST is only required if you will be debugging your IMS applications in foreground.

## CAMRCBTS

CAMRCBTS is a sample CLIST showing the necessary file allocations for the foreground testing of IMS /DC programs under BTS. If you are licensed for the IMS option and want to debug these types of applications in foreground, the following modifications should be made:

- Modify the PREFIX, IMS and BTS to match the prefix of the installed application, IMS, and BTS data sets.

Under `/*Note2*/`, uncomment the option that pertains to your installation. Use ISPLLIB for the LE run-time library if it is different from the default (linklist).



**Note:** There are two sections marked `/*Note2*/` in this CLIST.

Select per-site or per-user PROFLIB. In the sample CLIST, the first allocation statement after `/*Note3*/` is for those that want to allocate one PROFILE data set per site. The second allocation statement after `/*Note3*/` is for those that want to allocate a PROFILE data set per-user. The third statement after `/* Note3 */` is a CLIST that can create a per-user PROFILE data set if one does not exist and then allocate it. Note that you may have to tailor these statements or the CLIST to meet your specific installation requirements.

- For the statements under `/*Note4*/`, uncomment the optional files you are using with your product. See the optional files listed under CAMRCLST for a description of the optional files.
- Under `/*Note5*/`, add any other IMS files needed for your application.

- Under `/*Note6*/`, modify your BTSIN and other BTS-related files according to your installation standards.
- Modify the BTSAL parameter (in the first line of the CLIST) to NEW if the BTS files are to be recreated for every debugging session.

This CLIST is only required if you will be debugging your BTS applications in foreground.

## CAMRCDB2

CAMRCDB2 is a sample CLIST showing necessary file allocations for the foreground testing of DB2 programs. If you want to debug these types of applications in foreground, the following modifications should be made:

- Modify the PREFIX to match the prefix of the installed data sets.
- Under `/*Note2*/`, uncomment those libraries that are not allocated in your TSO logon procedure or in another CLIST. Use ISPLLIB for the COBOL run-time library if it is different from the default (linklist). If the DB2 libraries to run the application are not available through linklist or your TSO logon PROC, you must allocate them before issuing the RUN command in CLIST CAMRCDBX. The allocations can be made in CAMRCDB2 or CAMRCDBX, where the DB2 subsystem name is provided to the product.



**Note:** There are two sections marked `/*Note2*/` in this CLIST.

Select per-site or per-user PROFLIB. In the sample CLIST, the first allocation statement after `/*Note3*/` is for those that want to allocate one PROFILE data set per site. The second allocation statement after `/*Note3*/` is for those that want to allocate a PROFILE data set per-user. The third statement after `/* Note3 */` is a CLIST that can create a per-user PROFILE data set if one does not exist and then allocate it. Note that you may have to tailor these statements or the CLIST to meet your specific installation requirements.

- Under `/*Note4*/`, uncomment the optional files you are using with the application. See the optional files listed under CAMRCLST for a description of the optional files.

This CLIST is only required if you will be debugging your DB2 applications in foreground.

## CAMRCDBX

You might need to update CAMRCDBX to allocate the DB2 libraries to run the application, if they are not available through linklist or your TSO logon PROC. See the description for CAMRCDB2.

- Modify the PROC to include a query of the user supplied DB2 subsystem, &MRSSN, and then allocate the DB2 subsystem's appropriate loadlibs, both DSN and loadlib. Add a LIBDEF statement for ISPLLIB to allocate these loadlibs.
- If your InterTest Batch loadlib is not in your linklist then you also need to include that loadlib in the LIBDEF statement of ISPLLIB concatenation.

- If your application load library that contains the program you are debugging also has a program with the name “CALL”, then that “CALL” program will be executed in place of the INTBATCH program in the TSO CLIST “CALL 'CAI.CAVHLOAD(INTBATCH)' 'MEMBER(INTISPF)’” statement, which will cause unpredictable results or failures. If this situation exists, then you should add the TSO “TEST” command to the CAMRCDBX CLIST in between the ‘RUN’ and ‘CALL’ commands.

## CAMRSTRT

CAMRSTRT is called under an ISPF NEWAPPL. If you would like to modify the initial PF key settings for this APPL, uncomment the call to the CLIST CAMRKEYS.

## CAMRKEYS

CAMRKEYS is called optionally from the CLIST CAMRSTRT to set up initial PF key settings for this APPL. By default, all of the PF keys are set to an initial value. By default, the PF keys will only be set once. To have the PF keys reset, modify the variable CHANGED.

## CAMRCMD

Member CAMRCMD is the CLIST that is called from the /clist command during a debugging session. For more information on the /clist command, see [Batch Debugging \(https://docops.ca.com/display/CAITSD11/Batch+Debugging\)](https://docops.ca.com/display/CAITSD11/Batch+Debugging).

## Customize Compile Procedures

Several sample JCL members are provided in library CAVHPROC, which can be modified and used as the application compile/assemble procedures. Which member you use depends whether you are using COBOL/VS, COBOL II and higher, PL/I, or Assembler.

## COBINT

COBINT is a sample JCL stream showing a COBOL/VS compile. The difference between this compile procedure and the standard procedure supplied by IBM is that this procedure updates the symbolic information in the PROTSYM file.

You must supply the following parameters:

- **PROT**  
Specifies the data set name of the PROTSYM file.
- **COPT**  
Overrides the compiler options.

You must specify several additional items in the JCL. Information that must be modified or supplied includes the following:

- COBOL compiler library
- COBOL copy library
- COBOL source library

- LKED SYSLIB

## COB2INT

COB2INT is a sample JCL stream showing a COBOL II or IBM COBOL compile. The difference between this compile procedure and the standard procedure supplied by IBM is that this procedure updates the symbolic information in the PROTSYM file.

You must supply the following parameters:

- **PROTSYM**  
Specifies the data set name of the PROTSYM file.
- **COPT**  
Overrides the compiler options.

You must specify several additional items in the JCL. Information that must be modified or supplied includes the following:

- COBOL compiler library
- COBOL copy library
- COBOL source library
- LKED SYSLIB

## ASMINT

ASMINT is a sample JCL stream showing an assemble procedure. This procedure updates the symbolic information in the PROTSYM file.

You must supply the following parameters:

- **PROTSYM**  
Specifies the data set name of the PROTSYM file.
- **AOPT**  
Overrides the assembler options.

You must specify several additional items in the JCL. Information that must be modified or supplied includes the following:

- Assembler library
- Assembler macro library
- LKED SYSLIB

## PLIINT

PLIINT is a sample JCL stream showing a PL/I compile. This procedure updates the symbolic information in the PROTSYM file.

You must supply the following parameters:

- **PROTSYM**  
Specifies the data set name of the PROTSYM file.
- **POPT**  
Overrides the PL/I compile options.

You must specify several additional items in the JCL. Information that must be modified or supplied includes the following:

- PL/I source library
- PL/I copy library
- LKED SYSLIB

### Add to ISPF Main Panel (Optional)

To add CA InterTest Batch to your ISPF main menu, use member CAMRISPF in the CA InterTest Batch runtime source library (CAVHSAMP) as a model to add option INTB to your ISPF main menu.

### Add ISPF Libraries to TSO Logon Procedure (Optional)

To add the CA InterTest Batch ISPF libraries to your TSO logon procedure (instead of using LIBDEFs in the startup CLISTs), make the following changes to your TSO logon procedure:

1. Add the CA InterTest Batch ISPF panel library (CAVHPNL0) to the ISPLLIB concatenation.
2. Add the CA InterTest Batch ISPF message library (CAVHMSG0) to the ISPLMLIB concatenation.
3. Add the CA InterTest Batch CLIST library (CAVHCLS0) to the SYSPROC concatenation.
4. If the COBOL runtime library is different than the site default, add it to the ISPLLIB concatenation.

Refer to member CAMRTSOL in the CA InterTest Batch runtime source library (CAVHSAMP) for an example.

### Customize CA Roscoe (Optional)

This step applies only to CA Roscoe users.

The CAVHSAMP library provides three sample members for customizing CA Roscoe for the application: CAMRETSO, CAMREPL, and INTROS.

### CAMRETSO

CAMRETSO is a sample JCL to add to your CA Roscoe JCL. It should be concatenated to your CA Roscoe ETSOLIB. Add the COBOL runtime library if it is different from the site default.

## CAMREPL

CAMREPL contains sample definitions to add to your CA Roscoe Eligible Program List for the application. Incorporate it into your CA Roscoe EPL.

## INTROS

Use member INTROS in the CAVHSAMP library to customize the options that are passed to the application when it is invoked under CA Roscoe.

The INTROS member parameters include the following:

- **SYMLDYN**  
This parameter is no longer used.
- **TRACE(*n*)**  
Specifies the number of entries saved in the trace table and displayed by the TRACE and TRACE SOURCE commands. The default is 1000 and *n* is an integer between 1 and 32760.
- **EXTSTOR(*nM* | *nK*)**  
The extended storage parameter allows you to set the amount of extended storage that is accessed by the application under the z/OS operating system. This storage is used when formatting storage above the 16 MB line. The default is 64M. Specify the amount in megabytes (*nM*) or kilobytes (*nK*), and *n* is any integer between 1 and 100.
- **LINKPARM(*n*)**  
Specifies the maximum number of linkage parameters that are set by the installation. The default is 32 and *n* is any integer between 1 and 1000. Each linkage parameter requires a page of memory (4096 bytes) unless the default has been changed.
- **LINKSIZE(*n*)**  
Specifies the amount of memory to be getmained for each linkage parameter. The default is 4096. It is not recommended to specify less than 1024 or more than 32768. The parameters are always on page boundaries.
- **NOSLOW**  
Turns off the SLOW command at the installation. You are notified with a message stating this when attempting to use the SLOW command. Comment this parameter out if you do not want the SLOW command disabled.
- **SLOWFAST(*n*)**  
Specifies the duration of the wait time in hundredths of a second between the executable statements when using the SLOWFAST command. The default is one quarter of a second (*n*=25). The value is any integer between 1 and 4096. This might be helpful to TCAM users with buffering problems.
- **NOSTAX**  
This is a required startup parameter.
- **NOESTAE**  
This is a required startup parameter.

- **NOSTAE**  
This is a required startup parameter.
- **NOESTAI**  
This is a required startup parameter.
- **NOJSCB**  
This is a required startup parameter.

Other considerations for CA Roscoe users include:

- Concatenate the COBOL or PL/I runtime library if it is not in linklist and it is required to execute your application under CA Roscoe.
- Increase the CA Roscoe ETSALLOC parameter, if necessary.
- The CAI.CAVHRPF library must be imported to a CA Roscoe RPF library.

## Review and Customize for BTS

Member INTBTSX is provided for ISPF users with Business Transaction Services (BTS) extended support. (This member is provided for clients who are licensed for the IMS option and want to debug BTS applications in foreground.) Parameters relating to INTBTSX are listed following:

- **ISPF**  
This is required to run the application under the ISPF dialog manager.
- **SYMLDYN**  
This option is no longer used.
- **TRACE(*n*)**  
Specifies the number of entries saved in the trace table and displayed by the TRACE and TRACE SOURCE commands. The default is 1000 and *n* is an integer between 1 and 32760.
- **EXTSTOR(*nM* | *nK*)**  
The extended storage parameter allows you to set the amount of extended storage that is accessed by the application under the z/OS operating system. This storage is used when formatting storage above the 16 MB line. The default is 64M. Specify the amount in megabytes (*nM*) or kilobytes (*nK*), and *n* is any integer between 1 and 100.
- **LINKPARM(*n*)**  
Specifies the maximum number of linkage parameters that are set by the installation. The default is 32. *n* is any integer between 1 and 1000. Each linkage parameter requires a page of memory (4096 bytes) unless the default has been changed.
- **LINKSIZE(*n*)**  
Specifies the amount of memory to be getmained for each linkage parameter. The default is 4096. It is not recommended to specify less than 1024 or more than 32768. The parameters are always on page boundaries.

- **NOSLOW**  
Turns off the SLOW command at the installation. You are notified with a message stating this when attempting to use the SLOW command. Comment this parameter out if you do not want the SLOW command disabled.
- **SLOWFAST(*n*)**  
Specifies the duration of the wait time in hundredths of a second between the executable statements when using the SLOWFAST command. The default is one quarter of a second ( $n=25$ ). The value is any integer between 1 and 4096. This might be helpful to TCAM users with buffering problems.
- **BTSNAME(\*\*\*\*\*)**  
This is a BTSTSOST user-defined name.
- **BTSEXT**  
This is intercepted PA2 support.
- **BTSSYSO**  
This is BTS SYSOUT support.
- **NOSTAX**  
This is a required startup parameter.
- **NOESTAE**  
This is a required startup parameter.
- **NOESTAI**  
This is a required startup parameter.
- **NOJSCB**  
This is a required startup parameter.

### Screen Handling Solution for the Message BTS0067I

This product has a great deal of code that deals with screen handling in the BTS environment. While this code handles the majority of environments, there is a screen-handling problem that occurs when a transaction reinserts itself without issuing a COBOL GOBACK verb.

Symptom: BTS does not display the next IMS screen because BTS fails to realize that the PA2 key from VTAM is actually a RESHOW indicator, and has not been entered by the terminal operator. If this occurs, the following message is displayed:

```
BTS0067I NO MORE MESSAGES. SCREEN CLEARED.
```

If your site has programs that perform the previous function, or your site has experienced any problems with PA2, parameter BTSEXT in member INTBTSX (see the previous parameter list) signals the application to use extended BTS support. The extended support causes BTS to ignore the PA2 key.

This modification has the following side effects on BTS:

- With the PA2 key disabled, it is not functional for paging.  
This only affects applications that use multiple page or segment output messages and use the transaction insertion technique described previously.

- The change also replaces the PF key simulation capability of BTS.  
This only affects users and BTS using terminals with NO program function (PF) keys.

Contact [CA Support \(http://www.ca.com/support\)](http://www.ca.com/support). If either of these effects causes a problem at your site.

The problem arises because BTS does not enter and exit full-screen mode properly. While it functions in a stand-alone environment, a failure occurs when BTS is run with another full-screen product such as this product.

## Applications Using Message Switching and DISPLAY or Figure Verbs

Applications that perform message switching and use DISPLAY, Figure, or both verbs should use the BTSSYSO option.

Parameter BTSSYSO in member INTBTSX (see the previous parameter list) signals the application to use extended BTS SYSOUT support.

The BTSSYSO parameter is recommended if your site has programs that use DISPLAY, Figure, or both in the BTS environment.

## Customize for DB2 SP and IMS/DC Debugging

Special support is provided for debugging DB2 stored procedures and for IMS/DC transactions without the use of BTS. For more information about using batch link, see [Batch Link Facility \(https://docops.ca.com/display/CAITSD11/Batch+Link+Facility\)](https://docops.ca.com/display/CAITSD11/Batch+Link+Facility).



**Note:** The batch link scheduling feature must be active and the DB2 SP or IMS/DC transaction name to be debugged must have already been defined to the schedule table before job submission. Otherwise the submitted DB2 SP or IMS/DC job will run unmonitored.

## Using Batch Link to Debug Your DB2 Stored Procedures

You can use batch link to debug your DB2 stored procedures. Manually convert the JCL used to start the WLM address space in which the DB2 stored procedure executes. For more information about making the necessary changes to the procedure, see JCL Requirements in [Batch Link Facility \(https://docops.ca.com/display/CAITSD11/Batch+Link+Facility\)](https://docops.ca.com/display/CAITSD11/Batch+Link+Facility).

Two further modifications are necessary:

1. Since this is a JCL procedure, the INT1OPTS DD cannot be defined in-stream. Therefore, you must copy the contents of the INT1OPTS DD to a separate, permanent data set. Then change the INT1OPTS DD statement in the JCL procedure to point to that separate data set.
2. Update the new INT1OPTS data set by adding the DB2SP option at the end of the control statement.

The following panel is an example of the converted JCL:

```
//D810WLM PROC RGN=7M,APPLENV=D810WLM,DB2SSN=D810,NUMTCB=1
//IEFPROC EXEC PGM=CAMRBL01,REGION=&RGN,TIME=NOLIMIT,
// PARM='&DB2SSN,&NUMTCB,&APPLENV'
//STEPLIB DD DISP=SHR,DSN=CAI.CAVHLOAD
// DD DISP=SHR,DSN=DB2V8104.RUNLIB.LOAD
// DD DISP=SHR,DSN=CEE.SCEERUN
// DD DISP=SHR,DSN=DB2.DB2810.SDSNLOAD
//INT1LOAD DD DISP=SHR,DSN=CAI.CAVHLOAD
//INT1PNLL DD DISP=SHR,DSN=CAI.CAVHPNL1
//INT1MSGL DD DISP=SHR,DSN=CAI.CAVHMSG0
//INT1PARG DD DISP=SHR,DSN=CAI.CAVHSAMP
//INT1PROF DD DISP=SHR,DSN=CAI.PROFLIB
//INT1CLOG DD SYSOUT=*
//INT1OPTS DD DISP=SHR,DSN=CAI.CAVHSAMP(PGM)
```

Following is an example of the contents of CAI.CAVHSAMP(PGM):

```
EXEC=DSNX9WLM,DB2SP
```

After modifying the JCL that WLM uses to run the stored procedure, invoke the stored procedure. Start the application, select option 5 for batch link, and then option 2. Your stored procedure will be waiting to be debugged.

## Using Batch Link to Debug Your Online IMS Programs

Use batch link to debug your online IMS applications without the use of BTS.

1. Convert the JCL to start the MPP region for use with batch link.
2. Add the IMSDC option to the INT1OPTS DD. The following is an example of the converted JCL with the option added:

```
//IMSZMPP4 JOB (07300000), 'MPP REGION', REGION=4096K,
// CLASS=3,MSGCLASS=X,MSGLEVEL=(1,1)
//REGION EXEC PGM=CAMRBL01,REGION=512K,
// TIME=1440,DPRTY=(12,0),
// PARM=(MSG,001000000000,
// N000000000,
// , , , ,10,
// 5,IMSZ, , , ,
// , ,N,
// ', , ,)
//INT1OPTS DD *
EXEC=DFSRR00,PROFILE=TESTMPP,IMSDC
/*
//INT1PARG DD DISP=SHR,DSN=CAI.CAVHSAMP
//INT1LOAD DD DISP=SHR,DSN=CAI.CAVHLOAD
//INT1PNLL DD DISP=SHR,DSN=CAI.CAVHPNL1
//INT1MSGL DD DISP=SHR,DSN=CAI.CAVHMSG0
//INT1PROF DD DISP=SHR,DSN=CAI.PROFLIB
//INT1CLOG DD SYSOUT=*
//INT1REPT DD SYSOUT=*
//STEPLIB DD DSN=CAI.CAVHLOAD,DISP=SHR
// DD DSN=IMS.USER.LOAD,DISP=SHR
// DD DSN=IMS.IMSZ.PGMLIB,DISP=SHR
// DD DSN=IMS.IMSZ.RESLIB,DISP=SHR
//PROCLIB DD DSN=IMS.IMSZ.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=A,
// DCB=(LRECL=121,BLKSIZE=3129,RECFM=VBA),
// SPACE=(125,(2500,100),RLSE, ,ROUND)
```

3. Start the region by submitting the modified JCL.

4. Define the transaction that you want to debug that is scheduled to run in the 'test mpp' region. Run the transaction.
5. Start the application, select option 5, Batch Link, and then option 2. Your transaction will be waiting to be debugged.



**Note:** For information about using the scheduling feature, see [Batch Link Facility \(https://docops.ca.com/display/CAITSD11/Batch+Link+Facility\)](https://docops.ca.com/display/CAITSD11/Batch+Link+Facility). Using this feature, you can select which applications you would like to debug and which should be executed unmonitored.

## CA SymDump Batch

### Bind DB2 Support Member



**Note:** If your site executes programs in a DB2 environment, you must perform this step.

Member CARXBIND in CA/CAVHJCL binds DBRM CAODDBB0 into package CAODPACK, the package into plan CAODPLAN, and grants EXECUTE authority of the plan to PUBLIC.

Using the instructions provided in the JCL, copy and edit the JCL as needed for your DB2 subsystem.

Submit the job and review the output to verify that BIND processing ran successfully. If BIND processing completed with a return code greater than 4, review the output, correct the problem, and resubmit.



**Note:** If your site uses multiple DB2 subsystems, you must execute this job for each subsystem.

## Abend Reporting with Language Environment (LE)

If your application is executing under IBM's Language Environment (LE), it is possible to receive abend reports even when an ABEND and system dump do not occur.

### Review Language Environment (LE) Considerations

Applications that terminate abnormally under IBM's Language Environment (LE) may not result in an abend or a system dump. The two types of abnormal terminations that your applications might experience under LE are listed following:

- User abends, system abends, or program check interrupts

- For abends or program check interrupts in a user program with the LE TRAP option set to TRAP (ON), the LE Condition Manager intercepts the abend. The application is usually terminated without an abend or system dump.
- For abends or program check interrupts in a user program with the LE TRAP option set to TRAP (OFF), the application is terminated by the abend and a system dump is produced.
- For abends issued by LE, the application is terminated by the abend and a system dump is produced.
- Unhandled LE conditions  
If a condition is raised (by LE or a user program) with severity 2, 3, or 4, and is not handled by a condition handler, the application is terminated without an abend or system dump.

There are abnormal terminations under LE, which do not ordinarily result in an Abend report, because CA SymDump Batch remains dormant until a system dump is requested. To guarantee that CA SymDump Batch reports are produced for every abnormal termination under LE, CA SymDump Batch provides CAOECEEX to be used as the LE abnormal termination exit.



**Note:** There are several ways in which an LE-enabled application can handle its own abend, such as using USRHDLR or CEEHDLR. When an application defines a condition handler and a SymDump Batch abend report is also needed, it is up to that program to percolate the abend.

## CAOECEEX - Add Abnormal Termination Exit Entry

Use the following procedure to add abnormal termination exit entry for CAOECEEX.

### Follow these steps:

1. Edit CEE.SCEESAMP(CEEEXTAN).
2. Follow the instructions in this member to add an abnormal termination exit entry for CAOECEEX.
3. Apply the applicable SYSMOD to make these changes take effect.



**Note:** For more information about the CEEEXTAN exit, see the *IBM z/OS Language Environment Customization* guide.

## Establish Access to the TSO Facility

If your site does *not* have CLIST libraries with variable length records, proceed to Concatenate ISPF File and Tailor Master Panel.

## Convert CLIST Library from Fixed to Variable

The CA SymDump Batch ISPF panels are shipped with several CLISTs. These CLISTs are in CAI.CAVHCLS0 after APPLY processing completes. For CA Technologies to ship corrective maintenance to these CLISTs, they must be in a fixed-length, card-image library because SMP/E invokes IEBUPDTE to make source changes.

Many sites use CLISTs that are in variable-length libraries. If you are one of these sites, CA Technologies provides a small, stand-alone batch utility that converts our CLISTs from the fixed-length form we ship into a variable form you can concatenate with the rest of your CLIST libraries.



**Note:** This utility program can be used to convert any fixed CLIST library to a variable library. It does not check whether the library you are converting is from CA Technologies. This utility does not support selective conversion; it converts the entire library.

The JCL to invoke the utility is in CAI.CAVHJCL(CARXCNVT). The REPLACE parameter is optional; you can specify if you want the CLISTs in the output data set replaced by similarly named CLISTs from the input data set. *Use this parameter with caution.*

Check the following information when you review this JCL:

- If the CA SymDump Batch load libraries are added to the linklist, the STEPLIB is not necessary.
- All messages from CAOHCNVT are written to DDNAME SYSPRINT. The program does not execute if you do not allocate SYSPRINT. It is a fixed-length, 121-byte record with ASA carriage control.
- The input (fixed-length) CLIST library must be allocated to SYSUT1, and the output (variable-length) CLIST library must be allocated to SYSUT2.



**Note:** You *must* specify a BLKSIZE if the output data set is being created. The BLKSIZE must be an even multiple of 255 plus 4 (for the Block Descriptor Word).

## Concatenate ISPF File and Tailor the Master Panel

You must include the CA SymDump Batch ISPF libraries in your session ISPF files concatenation to access the CA SymDump Batch panels. Use the following procedure to concatenate the ISPF file and tailor the Master panel.

### Follow these steps:

1. Concatenate the following data sets into the indicated ddnames (if they were not previously concatenated):

Data Set Name	DDName	Description
CAI.CAVHPNLO	ISPLLIB	ISPF panels for CA SymDump Batch
CAI.CAVHSKLO	ISPLLIB	ISPF skeletons for CA SymDump Batch

Data Set Name	DDName	Description
CAI.CAVHMSG0	ISPMLIB	ISPF messages for CA SymDump Batch
CAI.CAVHTBLO	ISPTLIB	ISPF tables for CA SymDump Batch
CAI.CAVHCLS0	SYSPROC	ISPF CLISTs for CA SymDump Batch
CAI.CAVHLOAD	ISPLLIB or STEPLIB	CA SymDump Batch Load Library

2. Perform the following steps if your site uses variable-length CLISTs:
  - a. Execute CAI.CAVHJCL(CARXCNVT) to change the CLISTs from fixed to variable length format.
  - b. Point the SYSPROC DD to the CAI.VCAICLIB data set (instead of CAI.CAVHCLS0).
3. Allocate the following data sets into the indicated ddname (if they were not previously allocated):

Data Set Name	DDName	Description
CAI.CAVHTBLO	CAOEXTBL	ISPF tables for CA SymDump Batch
CAI.CAVHHELP	CAOIHELP	ISPF Help library

You have now placed the CA SymDump Batch ISPF libraries into your session ISPF files concatenation.

To tailor the ISPF Master panel, do one of the following

- Copy the ISP@MSTR panel supplied with CA SymDump Batch (named CARXSBA0)
- Add new lines to your existing ISP@MSTR panel.

Use the following procedure to add new lines to your existing panel.

**Follow these steps:**

1. Add the first line after the last option selection displayed on the panel.
2. Define this line as an option number for the user to select with a brief description of the function. For example:
 

```
% 6 +CA SYMDUMP BATCH - Options Panels
% 7 +CA SYMDUMP BATCH - Invoke the CAIPRINT Repository Viewer
```
3. Add the lines containing the commands required to execute the options to the list of &SEL variables defined at the end of the panel. For example:
 

```
6. 'PANEL(CAOEP001) NEWAPPL(CA$Z) '
7. 'PGM(CAOIF000) NEWAPPL(CAOI) '
```

## Log and Notify Facilities

The following LOG and NOTIFY facilities are available with CA SymDump Batch. For more information, see [CA SymDump Batch Options \(see page 33\)](#).

- **WTO**  
Writes a description of everyabend to the job log using WTO with ROUTCDE=11

- **WTL**  
Writes a description of every abend to the system log
- **LOGROS**  
Sends a description of every abend to the CA Roscoe IE user who submitted the job
- **LOGTSO**  
Sends a description of every abend to the TSO user who submitted the job
- **LOGNET**  
Writes a description of every abend to CA Netman
- **LOGUNI**  
Writes a description of every abend to the CA NSM console
- **SMF**  
Writes a description of every abend to the SMF log

For WTO, WTL, LOGROS, LOGTSO, and LOGUNI, one of the following messages is provided depending on the amount of information available:

**CAPE022I jzzzzzzj ssssssss ABENDED tnnnn IN PROGRAM pppppppp AT +xxxxxx**

**CAPE023I jzzzzzzj ssssssss ABENDED tnnnn AT cccccccccccccccccccccccc**

- **jzzzzzzj**  
Identifies the jobname.
- **ssssssss**  
Identifies the stepname.
- **t**  
Identifies the abend type (S=System or U=User).
- **nnnn**  
Identifies the abend code.
- **pppppppp**  
Identifies the program name.
- **xxxxxx**  
Identifies the program offset.
- **cccccccc**  
Identifies the source statement from the symbolic file.



**Note:** The source statement is only available in the log message if symbolic processing is done at execution time (that is, the Abend report is written to the CAIPRINT DD).

The SMF records are formatted as follows:

---

CSMFREC	EQU	*	MAP FOR SMF RECORD
CSMFRDW	DS	XL4	RDW FOR SMF RECORD
*..... STANDARD SMF RECORD HEADER			
CSMFIND	DS	XL1	
CSMFRECC	DS	XL1	SMF RECORD TYPE
CSMFTIME	DS	XL4	
CSMFDATE	DS	XL4	
CSMFSYSD	DS	XL4	
*..... END OF SMF RECORD HEADER			
CSMFJOB	DS	CL8	JOBNAME
CSMFSTEP	DS	CL8	STEPNAME
CSMFJBID	DS	CL8	JOB NUMBER
CSMFACCT	DS	CL25	JOB ACCOUNT NUMBER
CSMFPGID	DS	CL21	PROGRAMMER NAME
CSMFTSOD	DS	CL8	TSOID
CSMFPROG	DS	CL8	NAME OF ABENDING PROGRAM
CSMFCSEC	DS	CL8	NAME OF ABENDING CSECT
CSMFOFFS	DS	XL3	OFFSET INTO PROGRAM
CSMFABTP	DS	CL1	TYPE ABEND (SYSTEM OR USER)
CSMFABCD	DS	CL4	ABEND CODE
CSMFREAS	DS	XL1	REASON CODE
CSMFPSW1	DS	XL4	FIRST WORD OF ABENDING PSW
CSMFPSW2	DS	XL4	SECOND WORD OF ABENDING PSW
CSMFFLG1	DS	XL1	FLAG BYTE
CSMFOPT	EQU	B'10000000'	PROGRAM IS CA-OPTIMIZED
CSMFCSL	EQU	B'00001000'	CSL IMAGE IS PRESENT
CSMFLANG	DS	XL1	LANGUAGE CODE
CSMFCOBV	EQU	C'S'	VS COBOL
CSMFCOB2	EQU	C'X'	COBOL II
CSMFCOB	EQU	X'00'	NOT COBOL
CSMFCBV4	EQU	C'4'	COBOL VERSION 4
CSMFCBV3	EQU	C'3'	COBOL VERSION 3
CSMFCBV2	EQU	C'2'	COBOL VERSION 2
CSMFOREL	DS	CL6	OPTIMIZER RELEASE
CSMFCTIM	DS	CL8	TIME PROGRAM COMPILED
CSMFCDAT	DS	CL11	DATE PROGRAM COMPILED
CSMFSEQ	DS	CL4	EDS SEQUENCE NUMBER
	DS	CL4	** RESERVED **
CSMFCSLI	DS	CL133	CSL IMAGE- INCLUDED ONLY
*			IF CSL INDICATOR ON.
CSMFEND	EQU	*	

A utility program (CAOESMF) is provided to select and report on CA SymDump Batch SMF records. You can use this utility to design your own customized report using the DSECT provided.

```

*****
                SEQUENCE NUMBER = 0000005
JOBNAME       = MIKEDA           STEPNAME      = TSOPROC
DATE         = mm/dd/yy         TIME          = hh.mm (http://hh.mm/).ss
ABEND        = S0C7             SMFID         = XE86
JOB NUMBER   = TSU06037         ACCOUNT #     = 40500000
LMOD         = CA02DEMO        PROGRAM        = CA02DEMO
REL INTERRUPT = 000006BC       ABS INTERRUPT = 000066BC
LANGUAGE     = COBOL II CA-OPTIMIZED
CSL NOT AVAILABLE
*****
                SEQUENCE NUMBER = 0000006
JOBNAME       = MIKEDB           STEPNAME      = TSOPROC
DATE         = mm/dd/yy         TIME          = hh.mm (http://hh.mm/).ss
ABEND        = S0C4             SMFID         = XE86
JOB NUMBER   = TSU08811         ACCOUNT #     = 40500000
LMOD         = MAINPROG        PROGRAM        = MAINPROG
REL INTERRUPT = 00000202       ABS INTERRUPT = 0000EA02
LANGUAGE     = NOT COBOL
CSL NOT AVAILABLE
*****
                SEQUENCE NUMBER = 0000007
JOBNAME       = MIKEDC           STEPNAME      = TSOPROC
DATE         = mm/dd/yy         TIME          = hh.mm (http://hh.mm/).ss

```

```

ABEND          = S0C7          SMFID           = XE86
JOB NUMBER     = TSU08983     ACCOUNT #    = 40500000
LMOD          = TESTPGM      PROGRAM      = TESTPGM
REL INTERRUPT  = 000001FC     ABS INTERRUPT = 000061FC
LANGUAGE       = NOT COBOL
CSL NOT AVAILABLE
*****
Total number of records processed = 3

```

### Use CA Netman Interface

The LOGNET option activates the CA Netman interface. CA SymDump Batch uses the Machine-Generated Problem Tracking (MGPT) facility to open a problem record with CA Netman. Before this facility can be used, you must complete the following steps.

**Follow these steps:**

1. Install and activate CA Netman on the same system on which CA SymDump Batch will be active.
2. Select the MGPT feature during CA Netman system generation.
3. Identify the NTMMGCTL, NTMSC, and NTMPI files in CAOETABL when the CA SymDump Batch options are customized.
4. Add two MGPT table entry records to the CA Netman database. One record must specify CA SymDump Batch as the source product and SYS-ABEND as the input category. The other record must specify CA SymDump Batch as the source product and USER-ABEND as the input category.
5. Copy the CA Netman MGPT interface modules NTMENU, NTMGGUP, and NTMECLS into a linklist library so that they are available for CA SymDump Batch at abend time.
6. Ensure that the CA Netman started task NTMBRD is up and running if the problem records are to be transferred to the problem tracking database (NTMPI file).

The CA Netman problem record layout is as follows:

NETMREC	DSECT			
NRECTO	DC	CL8 'NETMAN'		PRODUCT TO
NRECFROM	DC	CL8 'CA-EDS'		PRODUCT FROM
NRECLNAM	DS	CL44		LOGICAL NAME
		ORG	NRECLNAM+16	
NRECSER#	DS	CL10	BLANK	SERIAL NUMBER
NRECRES1	DS	CL18		RESERVED
NRECDATE	DS	CL6	(CURRENT)	INCIDENT DATE - CYYDDD
NRECTIME	DS	CL8	(CURRENT)	INCIDENT TIME - HHMMSSSTH
NRECLOC	DS	CL8	BLANK	LOCATION
NRECRES2	DS	CL2		RESERVED
NRECSYS	DS	CL4	SMFID	SYSTEM CODE
NRECEQMT	DS	CL10	BLANK	EQUIPMENT TYPE
NRECUNIT	DS	CL5	BLANK	UNIT ID
		SPACE 1		
NRECACT	DS	CL1	(1)	ACTION NEEDED
NRECACTO	EQU	C '1'		OPEN A PROBLEM
NRECACTU	EQU	C '2'		UPDATE A PROBLEM
NRECACTC	EQU	C '4'		CLOSE A PROBLEM
		SPACE 1		
NRECUPD	DS	CL1	(N)	SOURCE TO UPDATE
NRECUPDY	EQU	C 'Y'		MGPT WILL UPDATE OR CLOSE

NRECU	PDN	EQU	C 'N'	MGPT WILL ONLY OPEN
		SPACE	1	
NRECDEV	DS	CL1	(N)	SPECIFIC DEVICE
NRECDEVY	EQU	C 'Y'		SPECIFIC DEVICE ASSOCIATED
NRECDEVN	EQU	C 'N'		NO SPECIFIC DEVICE ASSOCIATED
		SPACE	1	
NRECCAT	DS	CL10	SYS-ABEND	PROBLEM CATEGORY
*			OR USER-ABEND	
NRECDESC	DS	CL40	SHORT MSG TEXT	PROBLEM SHORT DESCRIPTION
*				
*			SHORT MESSAGE TEXT IS 'PROGRAM GOT SABC AT +00000000 RC=NN	
*				
		SPACE	1	
NRECTEXT	EQU	*		ADDITIONAL DESCRIPTION
NREJLIT	DC	C 'JOB='		
NREJNAM	DS	CL8		AREA FOR JOBNAME
	DS	CL1		
NRECSLIT	DC	C 'STEP='		
NRECSNAM	DS	CL8		AREA FOR STEPNAME
	DS	CL1		
NREJLIT	DC	C 'JOBID='		
NREJBID	DS	CL8		AREA FOR JOB ID
	DS	CL1		
NRECALIT	DC	C 'ACCOUNT='		
NREACCT	DS	CL25		ACCOUNTING CODE
	DS	CL1		
NRENLIT	DC	C 'NAME='		
NREPGID	DS	CL21		PROGRAMMER NAME
	DS	CL1		
NRETLIT	DC	C 'LANGUAGE='		
NRECLANG	DS	CL9		COMPILED LANGUAGE
*				
*		EITHER	'NOT COBOL'	*
*			'VS COBOL'	*
*			'COBOL II'	*
*			'COBOL V N'	*
*				*
		DS	CL1	
NRECSL	DS	CL80		AREA FOR CSL IMAGE IF AVAILABLE
NRECTUSE	EQU	*-NRECTEXT		
NRECFILL	DS	CL(205-NRECTUSE)		REMAINDER OF RECORD
		SPACE	1	
NRECE	ND	EQU	*	END OF EVENT RECORD
NRECS	IZE	EQU	NRECE	ND-NRECTO
				SIZE OF ONE RECORD
*				

Sufficient fields and identifying data are included in the record to permit filtering on a variety of criteria.

For more information about the uses for MGPT record details, see the CA Netman documentation.

## CICS Interfaces and Compatibility

This article contains interface and compatibility considerations for CA InterTest for CICS and CA SymDump for CICS.

- [CA InterTest for CICS and CA SymDump for CICS \(see page 151\)](#)
  - [Install DB2 Support \(see page 151\)](#)
  - [CICS Startup DSA Limit Parameter \(see page 152\)](#)
  - [Update the CICS JCL and Startup Parameters \(see page 152\)](#)
  - [Add the Required CICS Resource Definitions \(see page 153\)](#)

- Add Assembler DSECTs to the Symbolic File (see page 153)
- CA InterTest for CICS (see page 154)
  - Use the PROMMAC Macro (see page 154)
  - Sample PROMMAC Macros (see page 155)
  - Enter CNTL Commands from a CRLP-Type Terminal (see page 156)
  - Calls to Software and User Macro Support (see page 156)
  - DB2 Support (see page 164)
  - Handle Wild Branches (see page 165)
  - Add COBOL File Structures to the Symbolic File (see page 166)
  - Declare User-Defined CORE Commands (see page 167)
  - Special Considerations for HOGAN Systems (see page 170)
  - IBM's EXEC Debugging Facility (EDF) Support (see page 171)
  - Special Considerations for MRO Support (see page 172)
- CA SymDump for CICS (see page 175)
  - Install or Customize DFHPEP (see page 175)
  - Performance Considerations (see page 175)
  - Language Environment (LE) Considerations (see page 177)

## CA InterTest for CICS and CA SymDump for CICS

The following interface and compatibility information pertains to both CA InterTest for CICS and CA SymDump for CICS.

### Install DB2 Support

Use the following procedure to install DB2 support.

**Follow these steps:**

1. The CSDDDB266, CSDDDB267, CSDDDB268, CSDDDB269, CSDDDB270, and CSDDDB271 members on CAI.CAVHJCL provide RDO entries.
2. Bind the DBRM members (INB0FIDB and INB0AIDB) provided in the CAI.CAVHDBRM into an InterTest DB2 plan (the recommended name is INB0PLAN). The SAMPBIND member on CAI.CAVHJCL library provides a sample of DB2 BIND statements for the InterTest DB2 plan. For CA SymDump for CICS, you only need to bind the DBRM member INB0AIDB provided on the CAI.CAVHDBRM into the DB2 plan (the recommended name is INB0PLAN).
3. The DBA must grant bind and execute authority to the plan created in the previous step.
4. For DB2 subsystems running in new function mode (release 10.1 and above), ensure that the library containing the customized DSNHDECP (either SDSNEXIT or SDSNLOAD) is in the DFHRPL DD concatenation *after* the CICS program libraries.

- In order to support the CORE=LASTSQL command for CA InterTest and SymDump for CICS, the global exit XRMIOUT is started (by default) when you start the product. This exit then collects DB2-related call information for every DB2 call done in the system. This can have significant performance implications in some DB2 shops. The parameter XRMIO=YES/NO IN25OPTS controls use of this exit. The default is XRMIO=YES. We recommend that you carefully review this prior to using the default settings.

## CICS Startup DSA Limit Parameter

CA InterTest for CICS and SymDump for CICS make calls to IEWBIND when using the new online COMPOSITE module command and IEWBIND requires about 750 KB of below the line storage. To ensure there is sufficient storage available (below the line for IEWBIND calls), you must insure that your DSALIM is at least 750 KB less than your below the line private area size. Failure to do this could result in SOF4 and U900 abends in your CICS region during dump capture.

## Update the CICS JCL and Startup Parameters

If your site has previously installed CA InterTest or SymDump for CICS as a separate product, some DD statements may already exist.

If you use DD statements in your CICS startup JCL, your CA InterTest and SymDump for CICS files, as defined in the CSD, require DD statements in the JCL for CICS. DISP=SHR must be specified for PROTSYM to allow batch compiles to run concurrently with CICS. Alternatively, you may directly add the data set names to the CEDA FILE definition, defined in the CSDSYM member in CAI.CAVHJCL and remove the DD statements for those files.

## CA InterTest and SymDump Common DD Statements

Add the product load library to the DFHRPL in the CICS startup JCL.

```
// DD DSN=CAI.CAVHLOAD,DISP=SHR
```



**Note:** If you modified the IN25OPTS options during installation, make sure the load library that contains the modified IN25OPTS is in your DFHRPL concatenation before the CAI.CAVHLOAD library.

If you execute the demo programs, add the following DD to the DFHRPL in your CICS startup JCL. Specify the name of the load library which contains the demo programs:

```
// DD DSN=CAI.demoLoad,DISP=SHR
```

## CICS DD statements

```
//PROTCPF DD DSN=CAI.PROTCPF,DISP=OLD (CA InterTest for CICS only)
//PROTDMP DD DSN=CAI.PROTDMP,DISP=SHR (CA SymDump for CICS only)
//SYSMDUMP DD DSN=your.dataset.name (http://your.dataset.name/), (CA SymDump for CICS only)
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//PROTHLF DD DSN=CAI.CAVHHLF,DISP=SHR
//PROTUHF DD DSN=CAI.PROTUHF,DISP=SHR
```



**Note:** Pre-allocate your SYSDUMP data set with RECFM=FB,LRECL=4160,BLKSIZE=4160 and with enough space to capture an SVC dump for your CICS region.

## Add the Required CICS Resource Definitions

Member, CSDINT in CAI.CAVHJCL library adds the CICS resource definitions for the CA InterTest for CICS programs, files, and transactions. The CSDSYM member in CAI.CAVHJCL adds the CICS resource definitions for the CA SymDump for CICS programs, files, and transactions. Modify the JCL according to the instructions in these members.

Add the corresponding RDO group to your CICS startup list (the SIT parameter GRPLIST=) as listed in the following table:

CICS Level	CA InterTest for CICS RDO Group	CA SymDump for CICS RDO Group
CTS 4.1	INT11066	SYM11066
CTS 4.2	INT11067	SYM11067
CTS 5.1	INT11068	SYM11068
CTS 5.2	INT11069	SYM11069
CTS 5.3	INT11070	SYM11070
CTS 5.4	INT11071	SYM11071

The use of the CA InterTest for CICS transactions ISER, VIRG, and VTAT must not be protected to a specific sign on, and for CA ACF2 and RACF users must meet the special considerations described in [Installation Considerations \(https://docops.ca.com/display/CAITSD11/Installation+Considerations\)](https://docops.ca.com/display/CAITSD11/Installation+Considerations).

Do not change any of the options specified for the CA InterTest or SymDump for CICS entries. If you believe there is an error in any of the entries, contact CA Technical Support.

For CICS regions that use DB2, you must install the DB2CONN group before the CA InterTest or SymDump for CICS RDO entries.

## Add Assembler DSECTs to the Symbolic File

CA InterTest and SymDump for CICS let you display the major CICS areas in Assembler DSECT format. You can also display your own user areas in DSECT format. One way to do this is to add your DSECTs to the ones supplied by CA InterTest and SymDump for CICS. However, if you do this, your entries will disappear when you install a new release of CA InterTest and SymDump for CICS. A better way is to create one or more symbolic file members to contain your DSECTs.

Saving all of the DSECTs used at your site in one symbolic file member allows users of the FILE transaction to omit the symbolic program name when requesting records or DL/I segments in structured format. The name of this member must be referenced in the FSYMP installation option. The default name is PROTFILE.

Complete the following steps to create your own symbolic file members:

1. Run a job to add the members containing the DSECTs to the symbolic file.

2. Run a job to define the CORE keywords needed to display the DSECTs.

### Sample JCL for Adding Members to the Symbolic File

This sample JCL adds member USERDSEC to the symbolic file. This member contains the DSECT USERTWAF.

```
//USERDSEC      JOB    **your JOB card goes here**
//ASM           EXEC   PGM=ASMA90,REGION=1024K
//SYSLIB       DD     DSN=USER.SOURCE,DISP=SHR
//SYSUT1       DD     DSN=&SYSUT1,SPACE=(1024,(120,120),,,ROUND),
//              UNIT=VIO,
//              DCB=BUFNO=1
//SYSPUNCH     DD     SYSOUT=B
//SYSTEM       DD     SYSOUT=*
//SYSPRINT     DD     DSN=&&TEMP,DISP=(,PASS),UNIT=SYSDA,
//              SPACE=(CYL,(3,2)),
//              DCB=(RECFM=FBM,LRECL=121,BLKSIZE=2420)
//SYSIN        DD     *
USERTWAF       DSECT
USERDSEC       CSECT
               COPY USERTWAF
               END
//*
//PARM         EXEC   PGM=IN25PARM,REGION=512K,
//              PARM='USERDSEC,LISTER=REF,NOPURGE'
//STEPLIB      DD     DSN=CAI.CAVHLOAD,DISP=SHR
//CARDS        DD     DSN=&&CARDS,DISP=(,PASS),UNIT=SYSDA,
//              SPACE=(80,1)
//*
//SA           EXEC   PGM=IN25SYMA,REGION=512K
//STEPLIB      DD     DSN=CAI.CAVHLOAD,DISP=SHR
//INPUT        DD     DSN=&&TEMP,DISP=(OLD,DELETE)
//OUTPUT       DD     SYSOUT=*,
//              DCB=(RECFM=FBM,LRECL=121,BLKSIZE=2420)
//MESSAGE      DD     SYSOUT=*
//PROTSYM      DD     DSN=INTRTST.PROTSYM,DISP=SHR
//CARDS        DD     DSN=&&CARDS,DISP=(OLD,DELETE)
/*
```

For instructions on the STEPLIB DD, see the section Add COBOL File Structures to the Symbolic File.

For an explanation of the IN25PARM and IN25SYMA programs, see [Adding Symbolic Information \(https://docops.ca.com/display/CAITSD11/Adding+Symbolic+Information\)](https://docops.ca.com/display/CAITSD11/Adding+Symbolic+Information).



**Note:** This job does not require a link-edit step because it only updates the symbolic file. No load module is produced.

## CA InterTest for CICS

### Use the PROMMAC Macro

You can use PROMMAC macros in an assembly to create a load module defined in the CICS program definition that contains a number of CNTL commands. These commands are executed consecutively in the order of their appearance in the macros when either of the following commands is submitted:

```
CNTL=START,PROM=tablname
CNTL=EXEC,MODULE=tablname
```

Where *tablname* is the name of the load module generated by the assembly of the PROMMAC macros.

You can generate several alternate modules with different names, each with a CICS program definition, if you want alternate ways to start CA InterTest for CICS or execute CNTL commands.

If the **PROMD=tablname** option is specified in the IN25OPTS macro, you can achieve the same result by specifying:

```
CNTL=START
```

If the CNTL=START command is issued from a CRT terminal or CRLP-type terminal (a terminal whose input and output are two sequential files; see the IBM *CICS System Programmer's Guide*), the commands from the load module are displayed back. For a CRT terminal, there is a small time delay before each command is executed so the user can see what is being done.

The first positional parameter of the PROMMAC macro contains the CNTL command to be executed. The command must be enclosed in apostrophes.

If the table is to be used by the CNTL=START command, the first PROMMAC macro must contain this command:

```
CNTL=START , PROM=nn , PROX=nn '
```

Where *nn* sets the sizes of the internal CA InterTest for CICS tables.

The second positional parameter of the PROMMAC macro must contain END. It *must* be coded in the last PROMMAC macro before the END statement.

Any CNTL command can be specified except CNTL=LIST and CNTL=INQ.

## Sample PROMMAC Macros

The following JCL is suitable for z/OS installations. These examples are shown for illustration purposes only. For more information, see [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging)

```
.
//PROMAC JOB (NTSM,473) , 'JOHN BROWN' , CLASS=D , MSGCLASS=A
//ASSEM EXEC PGM=ASMA90 , REGION=512K ,
// PARM='XREF(SHORT) , DECK,NUM,LIST,ALIGN,NOOBJ '
//*
//SYSLIB DD DSN=CAI.CAVHMAC , DISP=SHR
//SYSUT1 DD SPACE=(CYL,(1,1)) , UNIT=SYSDA
//SYSUT2 DD SPACE=(CYL,(1,1)) , UNIT=SYSDA
//SYSUT3 DD SPACE=(CYL,(1,1)) , UNIT=SYSDA
//SYSPUNCH DD DSN=&OBJ , SPACE=(TRK,(1,1)) , UNIT=SYSDA ,
// DCB=(BLKSIZE=800 , LRECL=80 , RECFM=FB) , DISP=( , PASS)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
PROMMAC 'CNTL=START , PROM=20 , PROX=5 '
PROMMAC 'CNTL=ABP , ON , T321 '
PROMMAC 'CNTL=ON , PROG=(CFIL , TERMIO , FILEIO , ERRORS , SCAN , LLASRCH) ,
USR=. ANY '
PROMMAC 'CNTL=EXCL , PROG=(TERMIO , FILEIO) '
PROMMAC 'CNTL=ON , PROG=PBMASTER , FOL=ON , USR=. ANY '
PROMMAC 'CNTL=PURGE , INTRVAL=0100 ' , END
END
/*
//LINK EXEC PGM=IEWL , PARM='LIST,XREF,MAP' , REGION=512K
//SYSLMOD DD DSN=CAI.CAVHLOAD(PROMAC) , DISP=SHR
//SYSUT1 DD UNIT=SYSDA , SPACE=(TRK,(10,10))
```

```
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSN=&&OBJ,DISP=(OLD,DELETE)
//
```



**Note:** A CICS program definition named PROMAC is required for the generated table.

## Enter CNTL Commands from a CRLP-Type Terminal

If you have at least one CRLP-type terminal defined in your TCT, its input sequential file can contain CNTL commands. These commands are executed in sequence just as if they were entered manually from a terminal. Responses to them are written in the terminal's output sequential file.

To reduce the amount of time required to process large numbers of CNTL commands, perform the following steps:

1. Create a load module that contains the CNTL commands to be processed using the PROMMAC macro.
2. Add a PPT entry for the load module.
3. Set the following CA InterTest for CICS options:

```
RECNTMU=NO
RECNTNW=NO
```

4. Replace the CNTL commands in the sequential input with the following command:

```
CNTL=START , PROM=name
```

Where *name* is the name of the module that was created in Step 1.



**Note:** You can add the USR= option to any embedded CNTL=ON commands to assign the entries to a specific user ID. See [CNTL Commands and Menus \(https://docops.ca.com/display/CAITSD11/CNTL+Commands+and+Menus\)](https://docops.ca.com/display/CAITSD11/CNTL+Commands+and+Menus).

## Calls to Software and User Macro Support

This section explains how to support software (which must not be monitored by CA InterTest for CICS) that is invoked by CALLS from your application programs or your own macros. For example:

- To support global routines whose addresses are found in the CWA and that are given control from monitored programs by a BALR 14,14 or BALR 14,15 instruction. When such routines are given control, they are not monitored by CA InterTest for CICS until they return to the next byte after the BALR in the calling program.
- To make CA InterTest for CICS recognize a call (a COBOL or PL/I CALL verb or the Assembler CALL macro) to specific interfaces (such as database systems) and execute the call without monitoring the interface code.

In both cases, optional routines can be coded to do the following:

- Check the parameters before the global routine is given control
- Display CA InterTest for CICS automatic breakpoints with error codes defined by you



**Important!** CA InterTest for CICS assumes that the global routine of the CALL interface always returns control to the point of the CALL. If it does not, the user-written routine is mandatory.

## How Support Is Provided

To provide support, code a few lines of Assembler source code using the IN25UEX macros delivered in the CAI.CAVHMAC library. (The next section explains how to code these macros.) Then run assembly and link-edit steps to create the IN25UEXI module.

The macros create a table that, in addition to identification names and other information, contains references to the same routines that get link-edited with application programs because of CALLS issued by application programs. These references are resolved at link-edit time.

The IN25UEXI module consists of the table and the same modules that are usually link-edited with application programs. In the IN25UEXI, however, these modules are never executed. They are there only to compare a piece of their code with the code in the application load module.

CA InterTest for CICS uses the IN25UEXI module to compare the two pieces of code: the one link-edited with the IN25UEXI and the one that is about to receive control from the application program. The compared pieces of code are at the offset (from the entry point) and the length specified in the IN25UEX macros. If a large piece of code is involved, there is no need to include all of it with the IN25UEXI; include just enough to make the comparison. The comparison is made when a BALR 14,15 or BALR 14,14 instruction (for XA, BASSM or BASR instruction) is about to be executed.

If the code matches and there is no associated routine (defined in the IN25UEXI), CA InterTest for CICS drops monitoring and produces a standard CA InterTest for CICS entry in the CICS Trace Table (USER 144 code) with the identifier found in the corresponding entry in the IN25UEXI module. Monitoring resumes only upon return from the called module to the CALL statement.

If there is a user-defined routine, the routine is performed first. The routine can check the CALL's parameters and decide whether a particular interface should be given control or an automatic breakpoint should occur (if the CALL is incorrect), and whether monitoring should continue when the interface returns to the calling program.

IN25UEXI routines receive control in:

- AMODE31
- EXECKEY (CICS)
- BASESPACE MODE

## Code IN25UEX Macros for Called Software

The entry points are specified by using one of the following formats:

```
IN25UEX CALL=entry1,ROUT=routine,LENGTH=xx,DISP=xx,ID=xx
"      "      "
"      "      "
or
IN25UEX CALL=(entry1,.....,entryn)
"      "      "
"      "      "
IN25UEX TYPE=FINAL (REQUIRED)
place optional routines here
```

### CALL=entry1

This required parameter specifies the name of an entry point. Specify the same name used in your COBOL or PL/I CALL statement or Assembler CALL macro, or specify an arbitrary name.

A name specified in this CALL parameter is displayed on the CA InterTest for CICS Request Breakpoint menus and on the CNTL=INQ reports.

Multiple entry point names can be specified, as shown in the second format; however, the optional parameters listed next cannot be used.



**Important!** Do not specify entry points used in a CALL that is the result of an EXEC CICS.

### ROUT=routine

This optional parameter specifies the name of your special routine. Omit this parameter if there is no routine.

### LENGTH=xx

This optional parameter specifies the length (in decimal) of the comparison. The maximum length is 64 and the default length is 16. The default is sufficient in most cases.

### DISP=xx

This optional parameter specifies the offset (in decimal) from the entry point of the code to be compared. The maximum offset (displacement) is 512; the default offset is 0.

### ID=xx

This optional parameter specifies the identification value for the CA InterTest for CICS entry in the CICS Trace Table. Like all such entries produced by CA InterTest for CICS, the first byte contains the character C (USER 144), field A contains the displacement, and field B contains the characters US followed by the two characters specified in this parameter. If the ID= parameter is not specified, CA InterTest for CICS generates a value.

After IN25UEXI is link-edited, verify the total size of the result as shown in the link-edit listing. There is no need to include entire interface modules in the IN25UEXI, as the comparison occurs only for the specified (or default) length at the specified (or default) offset. If the IN25UEXI module is too large, you can code little Assembler CSECTs that identically replace the ones used by applications in IN25UEXI. This technique has been used in the pregenerated version of IN25UEXI. Its source code is provided in the CAI.CAVHJCL source library in member UEXIDB2. This technique has reduced the size of IN25UEXI from approximately 30 KB to just 192 bytes.

## Support Your Site's Global Routines

This section does not apply to most CA InterTest for CICS sites. It applies to global routines that are given control by the following two machine instructions:

```
L      14,CWALABEL
BALR  14,14
```

Where CWALABEL is the label of a field in the CWA that contains the address of the global routine.

The entry points are specified by using one of the following formats:

```
***** CWA USER DSECT REQUIRED HERE *****
      USING CWADS,0 CA INTERTEST REQUIREMENT
      CWADS DSECT      USER CWA
      CWAFLD1 DS CL20
      CWAENT1 DS F      USER ROUTINE
      CWAFLD3 DS CL100
***** CA INTERTEST SPECIFICATIONS *****
      IN25UEX CWAD=label1,ROUT=routine,ID=xx
      "      "      "
      "      "      "
      IN25UEX CWAD=labeln
      IN25UEX TYPE=FINAL (REQUIRED)
      place optional routines here
or
      IN25UEX CWAD=(label1,.....,labeln)
      IN25UEX TYPE=FINAL (REQUIRED)
      place optional routines here
```

### CWAD=cwadr

This required parameter specifies a label that is defined in the CWA and contains the address of the entry point of a piece of commonly used code, such as a date conversion routine. For example, if the routine address is 20 bytes from the beginning of the CWA, you can code:

```
IN25UEX CWAD=(CWADS+20)
```

Multiple entry point names can be specified as in the second format; however, if specified in this manner, the optional parameters listed next cannot be used.

### ROUT=routine

This optional parameter is used as in the CALL= form.

### ID=xx

This optional parameter is used as in the CALL= form.

## Code the ROUT= Routine

The ROUT= parameter coded in the IN25UEX macro specifies the name of a routine that is performed by CA InterTest for CICS prior to the execution of the IN25UEXI-supported CWA routine or CALL. The routines are coded in Assembler after the IN25UEX TYPE=FINAL control card. CICS commands are not allowed.

Each routine must be a CSECT whose name is specified by the ROUT= keyword. When CA InterTest for CICS passes control to the routine, the registers are set as follows:

R0 = 0

R1 = address of the BALR instruction

R2 = address of the called routine that is about to receive control by the BALR (or, for XA, by BASSM or BASR)

R3 = address of the CA InterTest for CICS work area

R4 = undetermined

R5 = undetermined

R6 = undetermined

R7 = undetermined

R8 = undetermined

R9 = address of an eight-byte field that contains the name specified in the CALL= or CWAD= parameter

R10 = undetermined

R11 = undetermined

R12 = user's TWA address

R13 = EIB address

R14 = return address for this routine

R15 = entry point address for this routine



### Notes:

- In XA or ESA systems, this routine receives control in the same AMODE (addressing mode 24 or 31) that the branch instruction had prior to its execution.

- Registers 3 and 14 must not be changed by the routine. If these registers are used, they must be saved and restored by the routine.
- The application program registers 0 to 15 (prior to the execution of the branch instruction) are at 96 (X'60') bytes past the address in register 3.

In addition to checking application-related parameters, the routine must determine if CA InterTest for CICS should continue monitoring. Monitoring must not continue if the code that is called by the monitored program does not return control to the next byte after the BALR instruction.

Before the routine returns to the address in register 14, it must set register 0 to one of the following values:

0

Execute the CALL without monitoring.

X'D0' X'FF'

CA InterTest for CICS should issue an automatic breakpoint with this error code.

-1

The routine does not return control to the next byte after the BALR; therefore monitoring must be dropped.

Any other negative value

CA InterTest for CICS should monitor the called piece of code.

## Sample JCL for Generating IN25UEXI Programs

This example shows a call to the global routine CWAENT1, which has an associated routine called MYCHECK.

```
//IN25UEXI JOB (123,45),USERID,MSGCLASS=A,TIME=(,09)
//ASM      EXEC PGM=ASMA90,REGION=1024K,
//          PARM='DECK,LIST,XREF(SHORT),ALIGN'
//SYSPRINT DD  SYSOUT=A
//SYSPUNCH DD  DSN=&&LOADSET,DISP=(NEW,PASS),UNIT=SYSDA,
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),SPACE=(400,(100,100,1))
//SYSLIB   DD  DSN=CAI.CAVHMAC,DISP=SHR
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSIN    DD  *
***** CWA USER DSECT REQUIRED HERE *****
          USING CWADS,0  CA INTERTEST REQUIREMENT
CWADS     DSECT          USER CWA
CWAFLD1   DS CL20
CWAENT1   DS F          USER ROUTINE
CWAFLD3   DS CL100
***** CA INTERTEST SPECIFICATIONS *****
*
*   INSERT YOUR IN25UEX STATEMENTS HERE
*
          IN25UEX CWAD=CWAENT1,ROUT=MYCHECK
          IN25UEX TYPE=FINAL
*
```

```

*   INSERT USER WRITTEN ROUTINES HERE
*
*   TITLE 'ROUTINE TO CHECK CWAENT1'
MYCHECK CSECT
  USING MYCHECK,15      ESTABLISH ADDRESSABILITY
  L      4,X'60'+4(,3)  OBTAIN APPLICATION'S REGISTER 1
  LTR    4,4            IS REGISTER 1 ZERO ?
  BZ     RETOKAY       YES, NOTHING MORE TO CHECK.
  L      4,0(,4)       PICK UP 1ST ADDR FROM PARM LIST
  LTR    4,4            IS THE HIGH ORDER BIT ON ?
  BM     ONLYONE       YES, ONLY ONE PARAMETER.
  CLC    =C'FINAL',0(4) 1ST PARAMETER SAYS 'FINAL' ?
  BNE    NOFINAL       NO, TREAT SEPARATELY.
  LH     0,=H'-1'      YES, INDICATE 'DROP MONITORING'
  BR     14             AND RETURN TO CA INTERTEST.
NOFINAL CLI 0(4),C'0'   1ST CHARACTER NUMERIC ?
  BNL    RETOKAY       YES, GO CONTINUE NORMALLY.
  LH     0,=H'-2'      NO, KEEP MONITORING THE CALLED-
  BR     14             - ROUTINE.
ONLYONE CLI 0(4),C'X'   1ST CHARACTER EQUAL TO "X"
  BE     RETOKAY       YES, GO CONTINUE NORMALLY.
  LA     0,X'E5'       NO,DO A BREAKPOINT WITH CODE E5
  BR     14             AND RETURN TO CA INTERTEST.
RETOKAY SR 0,0          INDICATE 'CONTINUE NORMALLY'
*                   (DO NOT MONITOR THE CALL)
  BR     14             AND RETURN TO CA INTERTEST.
  LTRG   14             TERMINATES THE ROUTINE.
*
*   CODE ANY ADDITIONAL ROUTINES HERE
*
*   END                TERMINATES THE ASSEMBLY OF IN25UEXI
/*
//LKED EXEC PGM=IEWL,REGION=512K,PARM=(XREF,LIST,MAP)
//SYSLMOD DD DSN= yourlib,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,DCB=BLKSIZE=1024,SPACE=(1024,(200,200))
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSN=&&LOADSET,DISP=(OLD,DELETE)
// DD *
  ENTRY IN25UEXI
  NAME IN25UEXI(R)
//

```

The IN25UEXI module created by the previous jobstream is used by the monitor program of CA InterTest for CICS. To install this module, perform the following steps:

1. Terminate CA InterTest for CICS (you can issue the CA InterTest for CICS checkpoint command before terminating CA InterTest for CICS).
2. Issue a CEMT SET PROG(IN25UEXI) NEWCOPY command for the new copy.
3. Issue a CEMT SET PROG(IN##PGM2) NEWCOPY command for the new copy.
4. Start or restart CA InterTest for CICS.



**Notes:** Replace ## with your two-digit CICS release number (66 for CICS 4.1, 67 for CICS 4.2, 68 for CICS 5.1, 69 for CICS 5.2, 70 for CICS 5.3, and 71 for CICS 5.4).

## IN25UEXI Instructions for Additional Vendor Products

This section lists IN25UEXI instructions for the following vendor products:

- Calls for COMPUTATIONS users
- Calls for SHRINK users
- Calls for Patient Care System (PCS) users
- Calls for HOGAN users
- Calls for CA Gen users

### Calls for COMPUTATIONS Users

To handle a call to the COMPUTATIONS package, code the IN25UEX control statement as follows:

```
IN25UEX CALL=MGCALL ,DISP=12 ,LENGTH=8
```

In addition, add an INCLUDE MSSECALL in your link-edit step.

You should exclude the COMPUTATIONS program from monitoring. To do this, issue the CNTL=EXCL, PROG=PS\* and CNTL=EXCL,PROG=PE\* commands.

### Calls for SHRINK Users

To handle calls to SHRINK, code the IN25UEX control statements as follows:

```
IN25UEX CALL=SHRINK  
IN25UEX CALL=EXPAND  
IN25UEX CALL=PUFFUP  
IN25UEX CALL=PUFFDOWN  
IN25UEX CALL=CLOSE
```

Where SHRINK, EXPAND, PUFFUP, PUFFDOWN, and CLOSE are the entry points.

### Calls for Patient Care System (PCS) Users

To handle calls to PCS, uncomment the COPY statement provided in the IN25UEXI member in CAI.CAVHSAMP:

```
***          COPY PCSUEXI
```

### Calls for HOGAN Users

To handle calls to HOGAN, add this COPY statement to the IN25UEXI member in CAI.CAVHSAMP:

```
COPY UEXIH0GN
```

Add the HOGAN.MACLIB to the ASSEMBLY step's SYSLIB.

Add the HOGAN.LOADLIB to the LKED step's SYSLIB.

### Calls for CA Gen Users

To handle calls to CA Gen, add this copy statement to the IN25UEXI member in CAI.CAVHSAMP:

```
COPY UEXIVG
```

## DB2 Support

### DB2 Application Program Support

The installation procedure is explained next. For information on how to test and debug DB2 application programs with CA InterTest for CICS, see [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging).

**Using the Pregenerated Version** -- A pregenerated IN25UEXI with DB2 support is provided. If you have no other special software situations that the IN25UEXI program will handle, there are no further installation steps you have to perform for DB2 support.

**Creating the IN25UEXI Module for DB2** -- To monitor application programs that issue SQL calls, a special program named IN25UEXI must exist in your CA InterTest for CICS load library. A pregenerated version of IN25UEXI, assembled for DB2 Release 10.1 and above, is provided.

You can also use the IN25UEXI program to support calls to programs that are not to be monitored by CA InterTest for CICS. If you have programs that issue calls, or require special handling, review the section Calls to Software and User Macro Support. In this case, you must combine the two uses in one IN25UEXI.

The source code for the preassembled version of IN25UEXI with DB2 support is provided in the member UEXIDB2 in the CAI.CAVHMAC library.

The following JCL example creates the IN25UEXI module for combined DB2 and special uses. Modify this example to meet your system requirements.

```
//IN25UEXI JOB ...
//ASM      EXEC PGM=ASMA90,REGION=1024K,
//          PARM='DECK,LIST,XREF(SHORT),ALIGN'
//SYSPRINT DD  SYSOUT=A
//SYSPUNCH DD  DSN=&&LOADSET,DISP=(NEW,PASS),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),SPACE=(400,(100,100,1))
//SYSLIB   DD  DSN=CAI.CAVHMAC,DISP=SHR
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSIN    DD  *
            COPY UEXIIDMS      CA IDMS
            COPY UEXIDATA      CA DATACOM
            COPY UEXITELN      CA TELON
            COPY UEXISORT      CA SORT
            COPY UEXIMAST      CA MASTERPIECE
            COPY UEXIDB2       DB2
            COPY UEXICPSM      CICSplex SM
            COPY UEXISOKT      TCP/IP SOCKETS
*
*   INSERT YOUR IN25UEX STATEMENTS FOR SPECIAL CALLS HERE
*
*   IN25UEX TYPE=FINAL
*
*   INSERT ANY USER WRITTEN ROUTINE HERE
*
*   END                TERMINATES THE ASSEMBLY OF IN25UEXI
/*
//LKED EXEC PGM=IEWL,REGION=512K,PARM=(XREF,LIST,MAP)
*
*   INSERT ANY //SYSLIB STATEMENTS FOR SPECIAL LOADLIBS HERE
*
//SYSLMOD DD DSN=yourLib,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,DCB=BLKSIZE=1024,SPACE=(1024,(200,200))
```

```
//SYSPRINT DD SYSOUT=A
//SYSLIN DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//      DD *
//      ENTRY IN25UEXI
//      NAME IN25UEXI(R)
//
```

The IN25UEXI module created by the previous jobstream is used by the monitor program of CA InterTest for CICS. To install this module, you must perform the following steps:

1. Terminate CA InterTest for CICS (you can issue the CA InterTest for CICS checkpoint command before terminating CA InterTest for CICS).
2. Issue a CEMT SET PROG(IN25UEXI) NEW command for the new copy.
3. Issue a CEMT SET PROG(IN##PGM2) NEW command for the new copy.
4. Start or restart CA InterTest for CICS.



**Notes:** Replace ## with your two-digit CICS release number (66 for CICS 4.1, 67 for CICS 4.2, 68 for CICS 5.1, 69 for CICS 5.2, 70 for CICS 5.3, and 71 for CICS 5.4).

## Monitor DB2 Applications

A pregenerated version of IN25UEXI, assembled for DB2 Releases 10.1 and above, is provided in your CA InterTest for CICS load library.

If you do not have any other special software situations that will be handled by the IN25UEXI program, you need not perform any additional installation steps for DB2 support. However, if you have programs that issue calls or require special handling, see Calls to Software and User Macro Support.

## Support DB2 Calls in the FILE and CORE Facilities

The FILE facility supports dynamic SQL calls to DB2. This feature lets users view, alter, add, or delete data in DB2 tables *without* leaving CICS. The CORE facility lets you view the last SQL statement executed (CORE=LASTSQL).

The pregenerated versions of IN25AIDB and IN25FIDB are preassembled with DB2 Release 10.1.

## Handle Wild Branches

When a monitored program passes control to another program directly by a branch instruction, bypassing the CICS services of an XCTL or a LINK macro or command, CA InterTest for CICS treats this as a wild branch (branching outside a module) and causes an automatic breakpoint. Such direct passing of control, although not advised by CICS coding standards, is used frequently in some applications.

You cannot monitor just the program receiving control by a direct branch from another program. To monitor a receiving program, you must also monitor the program passing control to it. Monitoring can begin only with the program that originally received control from CICS.

Most often, the program receives control by a direct branch caused either by a CALL statement or by a special macro. Usually, such code should not be monitored by CA InterTest for CICS. See the section [Calls to Software and User Macro Support \(see page 156\)](#) for an explanation of how to make CA InterTest for CICS drop monitoring in such cases.

## Monitor a Wild Branch

If the program that receives control should be monitored and you want to debug it with CA InterTest for CICS, there are three possible situations:

- The receiving program resides in the same load module as the program that passes control. In this situation, use the composite support facility of CA InterTest for CICS. This facility lets you debug a subprogram as if it were a separate program and supports all language combinations.
- The receiving program resides in another load module that has a CICS program definition. In this situation, use the FOL=CICS-program-definition-name online option and, if needed, composite support. This approach makes all CA InterTest for CICS debugging features available for the branched-to program.



**Note:** For COBOL II dynamically called programs, the FOL= option is not needed. Simply set breakpoints in the dynamically called program as you would for any other CICS program.

- The receiving program resides elsewhere and has no CICS program definition. In this situation, use the FOL=ON online option. In this case, breakpoints can be set for addresses, not offsets, and symbolic CA InterTest for CICS support is not available.

## Use the FOL=ON Option

We strongly advise that the FOL=ON online option be applied only at the program level; that is, monitoring declared with a CNTL=ON,PROG= command as opposed to CNTL=ON,TRAN= or CNTL=ON,TERM= commands. This allows online options to be used in the most convenient way, with different options such as breakpoints declared for different programs. If necessary, the FOL=ON online option can also be specified at the terminal or transaction level.

## Add COBOL File Structures to the Symbolic File

The COBOL program shown next is an example of a dummy default program that contains 01 level structures. Saving all of the 01 level structures used at a site in one file allows users of the FILE transaction to omit the symbolic program name when requesting records or DL/I segments in structured format. Symbolic information for the default program must be saved in the symbolic file, and its name must be referenced in the FSYMP installation option. The default name is PROTFILE.

```
//PROTFILE JOB (NTSM,473) , ' JOHN BROWN ' , CLASS=D , MSGCLASS=A
//*
//COB      EXEC PGM=IGYCRCTL , REGION=1024K ,
//  PARM=' OBJECT , APOST , FLAG ( I , W ) , LIST , XREF , MAP , NOOPT , VBREF '
//STEPLIB DD DSN=SYS1 . COB2COMP , DISP=SHR
//SYSLIB  DD DSN=CICS . COBLIB , DISP=SHR
//SYSLIN  DD DSN=&&LOADSET , DISP=( MOD , PASS ) ,
//          UNIT=SYSDA , SPACE=( 80 , ( 250 , 100 ) )
//SYSUT1  DD UNIT=SYSDA , SPACE=( 460 , ( 350 , 100 ) )
//SYSUT2  DD UNIT=SYSDA , SPACE=( 460 , ( 350 , 100 ) )
//SYSUT3  DD UNIT=SYSDA , SPACE=( 460 , ( 350 , 100 ) )
```

```

//SYSUT4 DD UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT5 DD UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT6 DD UNIT=SYSDA,SPACE=(460,(350,100))
//SYSUT7 DD UNIT=SYSDA,SPACE=(460,(350,100))
//SYSPRINT DD DSN=&&TEMPIN,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(15,5)),
// DCB=(DSORG=PS,LRECL=133,BLKSIZE=1330,RECFM=FBA)
//SYSIN DD *
ID DIVISION.
PROGRAM-ID. PROTFILE.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 PROGRAM-NAME PIC X(8) VALUE 'PROTFILE'.

LINKAGE SECTION.

** CAR SEGMENT **
01 CAR.
05 CAR-SEGMENT-KEY-FIELD.
15 CAR-SEGMENT-MAKE PIC X(12).
05 CAR-SEGMENT-MODEL PIC X(12).
05 CAR-SEGMENT-TYPE PIC X(10).
05 CAR-SEGMENT-WEIGHT PIC S9(5) COMP-3.
05 CAR-SEGMENT-CYLINDRS PIC S9(3) COMP-3.
** DEALER SEGMENT **
01 DEALER.
05 CAR-DEALER-SEGMENT-KEY.
15 CAR-DEALER-SHORT-NAME PIC X(9).
15 CAR-DEALER-TIEB PIC S999 COMP-3.
15 CAR-DEALER-NBRWD PIC S9 COMP-3.
05 CAR-DEALER-FIRST-NAME PIC X(24).

PROCEDURE DIVISION.
MOVE 'PROTFILE' TO PROGRAM-NAME.
GOBACK.

/**
//SYMSTEP EXEC PGM=IN25COB2,REGION=1024K
//STEPLIB DD DSN=CAI.CAVHLOAD,DISP=SHR
//INPUT DD DSN=&&TEMPIN,DISP=(OLD,DELETE)
//OUTPUT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//MESSAGE DD SYSOUT=A
//PROTSYM DD DSN=** your protsym file **,DISP=SHR
//CARDS DD *
PROTFILE,LISTER=ALL,NOPURGE
/**

```

In the STEPLIB DD statement on the symstep specify the name of the library that contains the Testing and Fault Management Symbolic component.



**Note:** A similar job can be used for COBOL/VS.

## Declare User-Defined CORE Commands

This section explains how to define your own CORE keywords. User-defined CORE keywords provide a shorthand method for entering a complex CORE command by specifying a simple command that contains one or more of your own keywords. The new keyword is automatically replaced with a string of command elements that could otherwise be difficult to remember.

Among the new commands can be some to display your own areas in main storage, such as crucial control tables for running an application system. If your replacement command includes the USE= element, the storage areas can be displayed as COBOL or Assembler structures, with names and offsets attached to each data field.

For details, see the CA InterTest for CICS Help facility. Select the CORE Facility, then Structured Data Displays, and then Additional Features.

## CORUCOM Macro

The CORUCOM macro lets you define any number of new CORE command keywords, each an abbreviation for a series of CORE command elements. For instance, there is a user table that is pointed to by an address located eight bytes into the CWA. There is a counter 12 bytes into this table that contains the number of times a service request transaction was issued. Add the following CORUCOM entry and assemble the CORUCOM table:

```
CORUCOM CODE=SERVCNT ,COMND=CWA@8+C
```

To display the service request count online, issue:

```
CORE=SERVCNT
```

Code one CORUCOM macro for each new CORE keyword. Submit the resulting source code to assemble a new CA InterTest for CICS load module named IN25UCOM. The module is used by the CORE facility as a table to convert your own keywords into strings of CORE keywords.

Since the CORUCOM macro generates a separate CICS load module, you can add new or modified commands at any time. To do this, code the commands, assemble the module, and then do a NEWCOPY for the load module IN25UCOM by issuing:

```
CEMT SET PROG(IN25UCOM) NEWCOPY
```

## Code the CORUCOM Macros

The CORUCOM macro is delivered in the CAI.CAVHMAC library. The first CORUCOM statement must be coded as follows:

```
1 1
  COL= 1...5....0....5
        CORUCOM TYPE=INITIAL
```

The last CORUCOM statement must be coded as follows:

```
COL= 1...5....0....5
        CORUCOM TYPE=FINAL
```

Each user-defined CORUCOM statement specifies one user-defined element of the CORE command and one string of CORE commands that will replace that user-defined element.

The user-defined element is specified by the CODE= keyword. Each element keyword must be less than eight characters long, must consist only of alphanumeric characters and, to prevent confusion with a hexadecimal number, must contain at least one of these characters:

GHIJKLMNOPQRSTUVWXYZ. When two or more keywords have the same prefix (for example, ICE2, ICE), the **longer** keyword must be specified first, as illustrated in the following example.

The string of CORE command elements is specified by the COMND= keyword. The string must be enclosed in apostrophes and, within the string, any apostrophe must be coded as two apostrophes. All elements of the CORE command are permitted, including other user-defined elements. In the following example:

- The new keyword ICE2 (second macro) uses another new keyword ICE (in the third macro). Notice that the longer keyword is specified first.
- The fourth macro defines the keyword MYTABLE, which displays a storage area pointed to by the address in the CWA, and produces a display formatted as a structure named MYTABLE in the CA InterTest for CICS symbolic file records identified by the name MYSYMDEF. For more details, see the section Adding Assembler DSECTs to the symbolic file.
- The \* in column 72 means continuation.

1 1 7

```
COL= 1...5...0.....6.....2
CORUCOM TYPE=INITIAL (required)
CORUCOM CODE=ICE2, COMND='ICE@4'
CORUCOM CODE=ICE, COMND='=CSA@54'
CORUCOM CODE=MYTABLE, COMND='=CWA@8, USE=MYSYMDEF.MYT*
        ABLE'
CORUCOM TYPE=FINAL (required)
```

### Sample JCL for Assembling IN25UCOM

The following IN25UCOM JCL example uses the standard IBM procedure to assemble and link-edit macro level programs.

```
//UCOM JOB ...
/* (COMMENT: A STANDARD IBM PROC FOR ASSEMBLER MACRO LEVEL)
//STEP EXEC ASMFCL
//ASM.SYSLIB DD DSN=CAI.CAVHMAC, DISP=SHR
//ASM.SYSIN DD *
CORUCOM TYPE=INITIAL
* YOUR FIRST CORUCOM STATEMENT GOES HERE
CORUCOM TYPE=FINAL
END
/*
//LKED.SYSLMOD DD DSN=CAI.CAVHLOAD, DISP=SHR
//LKED.SYSIN DD *
NAME IN25UCOM(R)
/*
//
```



**Note:** The IN25UCOM member of the CA InterTest for CICS source library contains sample source for assembling the IN25UCOM load module.

### Sample JCL for Defining CORE Keywords

After you have added the necessary members to the symbolic file, run a job to define the commands needed to access your DSECTs. The sample job below uses CORUCOM macros to define the user CORE keyword TWAF. For more information, see Declare User-Defined CORE Commands.

```

//UCOM JOB ...
//* (COMMENT: A STANDARD IBM PROC FOR ASSEMBLER MACRO LEVEL)
//STEP EXEC ASMFCL
//ASM.SYSLIB DD DSN=CAI.CAVHMAC,DISP=SHR
//ASM.SYSIN DD *
        COPY CORUCOM
        CORUCOM TYPE=INITIAL
        CORUCOM CODE=TWAF,COMND='TWA,USE=USERDSEC.USERTWAF'
        CORUCOM TYPE=FINAL
        END
/*
//LKED.SYSLMOD DD DSN=CAI.CAVHLOAD,DISP=SHR
//LKED.SYSIN DD *
        NAME IN25UCOM(R)
/*
//

```

## Special Considerations for HOGAN Systems

CA InterTest for CICS features are available for debugging HOGAN applications, including symbolic support if the CA InterTest for CICS post-compile step was executed.

## Use Considerations

**Monitoring the PEM Module** -- The PEM module does not need to be debugged and should not be monitored by CA InterTest for CICS. Monitor only your HOGAN application programs using the segmented monitoring option, as discussed next.

**Monitoring Application Programs that use HOGAN** -- Application programs that use HOGAN should be monitored only by using the CA InterTest for CICS segmented monitoring options. Do not monitor such programs by transaction or terminal name, or by global monitoring. [CICS Debugging \(https://docops.ca.com/display/CAITSD11/CICS+Debugging\)](https://docops.ca.com/display/CAITSD11/CICS+Debugging) explains how to use segmented monitoring.

The USH=ON monitoring option must also be used to prevent unnecessary automatic breakpoints from occurring.

All of the CA InterTest for CICS online options are available. However, the FOL= option should not be used. It is advisable to learn how to use the BYP= option and the PF11 override option.

## Installation Procedure

To install CA InterTest for CICS for HOGAN support, first complete all of the required steps given in *Configure Your Product with CA CSM and Installing Your Product Using Pax ESD or DVD*. Then complete your installation by performing the following steps.

1. Create the IN25UEXI module for HOGAN. Instructions are given in the section *IN25UEXI Instructions for Additional Vendor Products*.



**Note:** This routine is customized for each release of HOGAN. To ensure compatibility, please contact CA Support.

2. Enable the CA InterTest for CICS segmented monitoring options. To do this, you must set the MONOM installation option in the IN25OPTS module to either MENU or NOMENU. Optionally, you can enable password security for segmented monitoring by setting the MONOMSEC installation option to YES.

This completes the installation for HOGAN support in a CTS environment.

## IBM's EXEC Debugging Facility (EDF) Support

The EDF facility, which is activated by the CEDF transaction and described in the IBM *CICS/VS Application Programmer's Guide Command Level*, does not interfere with the CA InterTest for CICS monitoring and interactive testing. CA InterTest for CICS also does not interfere with EDF, except when EDF presents the EDF breakpoint display of the program that is being monitored by CA InterTest for CICS, EDF incorrectly indicates the location of the command since commands are issued by CA InterTest for CICS, not by the program.

## Advantages of CA InterTest for CICS over EDF

CA InterTest for CICS offers many advantages over the EDF facility, including the following advantages:

- The ability to set breakpoints anywhere in the program (not just at EXEC CICS commands).
- Data display and modification by symbolic names. This means the programmer does not need the most recent listing of the tested program, if the data names or paragraph names (labels in Assembler) remain the same.
- Monitoring, such as the ability to detect any illegal action of the program between CICS commands. CICS abends are intercepted by EDF, but damage may have occurred on the way to the abend and EDF may not give any specifics on the problem.
- The ability to declare an unconditional or conditional breakpoint at a specific location -- the EXEC CICS call at the point when the EXEC CICS call parameters have already been formatted. At that time you can change the parameters (for example, by issuing the CORE=ARGnn command) before you let the command execute.
- The ability to set request breakpoints for a particular type of CICS command regardless of where in the program it occurs, for all EXEC CICS commands, or for all except some EXEC CICS commands. For example, with one specification you can set breakpoints at all File Control commands or at all READ or WRITE commands.

## Use CA InterTest for CICS with EDF

If you want to use CA InterTest for CICS with EDF, you can turn on EDF before you begin monitoring a program with CA InterTest for CICS. Moreover, when a monitored program is stopped at a request breakpoint for an EXEC CICS command, you can activate EDF by entering any character in the field marked EDF in the lower-right corner of the Detailed Breakpoint display.

When CA InterTest for CICS and EDF are being used on the same task, be aware of the following:

- If a CA InterTest for CICS breakpoint is set at an EXEC CICS command, the CA InterTest for CICS breakpoint occurs before the command is passed to CICS.

- At the breakpoint, the programmer can review and change any parameters of the command before telling CA InterTest for CICS to continue with the task.
- CA InterTest for CICS checks the parameters and, if necessary, halts the task at an automatic breakpoint.
- If that does not happen, the command is passed by CA InterTest for CICS to CICS for execution; that is, to the EXEC interface.
- Subsequently, EDF presents the EDF breakpoint display of the program that is being monitored by CA InterTest for CICS, before a command breakpoint. The EDF facility, however, is in control at that time and any changes by the user during the EDF breakpoint cannot be checked by CA InterTest for CICS.
- Only after the command is executed and a command EDF breakpoint display appears and EDF is told to continue with the task can CA InterTest for CICS resume control and continue monitoring.

### Special Considerations for MRO Support

To use CA InterTest for CICS in an MRO environment, you must have the following items:

- All CA InterTest for CICS required CICS definitions in any CICS application-owning region (AOR) that will use CA InterTest for CICS for testing.
- A local program definition for IN25VIRC in the terminal-owning region (TOR).
- A local transaction definition for VIRC in the TOR; this transaction ID must be identical in the TOR and all AORs.
- One remote transaction definition for VTAT in the TOR for each AOR that will use CA InterTest for CICS. Of course, the TOR's local VTAT transaction IDs must be unique (for example, VTA1, VTA2, VTA3, and so on). However, the remote transaction IDs can be the same for all AORs (for example, VTAT), so you do not need to maintain a unique IN25OPTS for each AOR. If you choose to specify an alternate VTAT transaction ID in one or more AORs, you must be sure to specify the same alternate transaction ID in the TOR's remote VTAT transaction definition.

In addition, for all CICS regions that will use CA InterTest for CICS -- both AOR and TOR -- the transaction definition for VIRC must specify the same transaction code.

A sample set of definitions is supplied in member CSDINTTO in CAI.CAVHJCL library. This sample set contains all of the RDO definitions required for the CICS TOR region. All CA InterTest for CICS transaction names are the same as those specified in the AOR and can be modified by your site.



**Note:** We recommend that the CA InterTest for CICS CORE transaction be installed in the TOR as **local** to help system support staff resolve problems in that region.

Remote transaction definitions can be defined for all transactions except VIRC, which can be started from the TOR to run in the AOR. Using these transaction definitions is usually preferable to starting routing sessions with the CRTE SYSID= transaction.

You can add a transaction definition in each TOR.

```
DEFINE TRANSACTION(LNTL)
```

Where LNTL -- is a four-character transaction name for the CA InterTest for CICS CNTL transaction in the TOR:

```
REMOTENAME(RNTL)
```

Where RNTL -- is a four-character transaction name for the CA InterTest for CICS CNTL transaction in the AOR:

```
REMOTESYSTEM(TROW)
```

Where TROW -- the name under which the AOR is known to the TOR.



**Note:** RNTL and LNTL can be the same name.

## Monitor Considerations for DFLTUSER

CA InterTest for CICS qualifies all monitoring entries with a CICS user ID. All monitoring entries use the form:

```
promid.userid
```

The ATTACHSEC option of the CONNECTION entry in the CSD is used with the DFLTUSER= option setting in IN25OPTS to provide the granularity of monitoring required. Only the combinations of DFLTUSER and ATTACHSEC given in the following table are supported:

IN25OPTS DFLTUSER=	CSD CONNECTION, ATTACHSEC=	Typical Use	Additional Consideration
ANY	LOCAL	Nonsecure MRO environment	AOR and TOR need the same default CICS user ID (set in the SIT)
SPECIFIC	IDENTIFY	Secure MRO environment	TOR's SIT uses SEC=YES or MIGRATE option

The following sections discuss each combination in detail.

### Monitor in a Nonsecure MRO Environment

The following combination says that monitoring and monitoring options should be set by terminal ID and not by CICS user ID.

```
DFLTUSER=.ANY  
ATTACHSEC=LOCAL
```

This setting is typically used in a nonsecure MRO environment where the TOR's SIT uses the SEC=NO option. The user ID passed from the TOR is the TOR's default user ID as set in the SIT (typically CICSUSER).

CA InterTest for CICS compares this user ID to the AOR's default user ID and, if they match, sets the user ID to be monitored by .ANY user. This entry causes all users of the program to be monitored (as in earlier CA InterTest for CICS releases).



**Important!** Should the default user IDs of the AOR and TOR not match, the results are unpredictable. Therefore, you should check that the default user IDs in the AOR and TOR are the same.

## Monitor in a Secure MRO Environment

The following combination is allowed in a secure MRO environment where the TOR's SIT uses the SEC=YES or MIGRATE option.

```
DFLTUSER=SPECIFIC
ATTACHSEC=IDENTIFY
```

In this case, the application's unique user ID is passed from the TOR to the AOR only if the ATTACHSEC option of the CONNECTION entry is IDENTIFY. This requirement and specifying DFLTUSER=SPECIFIC in IN25OPTS allows CA InterTest for CICS to assign the user ID to each monitoring command as a default. This saves the user from having to type it in or specify it, and allows CA InterTest for CICS to properly monitor user activity in the TOR.

## Additional Monitoring Considerations

The CRT terminal that is to receive the CA InterTest for CICS breakpoint display is called the receiving terminal. After receiving the display, the terminal remains ready to execute any CA InterTest for CICS command. However, until the user disconnects from CA InterTest for CICS, it will accept only CA InterTest for CICS commands.

The receiving terminal must meet these specifications:

- IBM 3270-type CRT or compatible:
  - The terminal must have ATI (automatic task initiation) capability.
  - If its screen is larger than model 2, its default screen size (as defined in the TCT) must be 24 lines x 80 characters (the model 2 size)
- For two-terminal testing:
  - The receiving terminal must be logged on to the TOR at the time the breakpoint is about to be displayed.
  - At the time the first breakpoint display is about to appear, the receiving terminal must not be occupied by any task (transaction). This restriction also applies to CRTE.
  - The next transaction ID must not be primed in the TCTTE of the receiving terminal. If this occurs, CA InterTest for CICS cannot write the breakpoint screen because CICS prevents automatic task initiation.

- The receiving terminal must not be occupied by an explicit routing session. (Explicit routing sessions are started by entering CRTE SYSID= and are normally ended by entering Cancel.) When necessary, the receiving terminal will have a routing session started for it by CA InterTest for CICS so the terminal can receive a breakpoint display.
- The terminal from which the tested transaction is entered or the terminal that the tested transaction owns is called the sending terminal. Since a user typically sets breakpoints without naming the receiving or sending terminal, the receiving and sending terminal will be the terminal from which the CNTL command was entered when the user ID monitoring option=. ANY.
- If the sending terminal is not the same as the receiving terminal, the sending terminal will be unavailable and tied to the tested transaction during the breakpoint. This restriction does not apply if the receiving terminal is the same as the sending one because the user can disconnect from CA InterTest for CICS.

## Remote FILE Support

All CICS regions participating in a remote FILE transaction session must be at CA InterTest for CICS Version 4.2 or above. Also, the FILE transaction and its associated program entry IN25FLE must be defined to the file-owning region. The transaction name assigned to the FILE transaction must be the same in all regions.

## CA SymDump for CICS

### Install or Customize DFHPEP

If your site does not have a custom DFHPEP module implemented, then the DFHPEP included in the CAI.CAVHLOAD library must be installed in a library concatenated before the IBM CICS library suffixed with SDFHLOAD, so that it replaces the IBM supplied dummy version of DFHPEP.

If your site already uses a customized DFHPEP it should be customized and relinked to include the following source statement:

```
EXEC CICS LINK PROGRAM( 'IN25PEP' )
```

Using the supplied copy of DFHPEP or customizing your own DFHPEP to link to IN25PEP is not a hard requirement for installation of CA SymDump for CICS. The dump capture still functions without DFHPEP/IN25PEP processing. However the failure to configure DFHPEP as suggested previously prevents capture of the last screen display in an MRO environment.

## Performance Considerations

### ▪ CICS Virtual Storage

The CICS virtual storage requirement of the dump-writing part of CA SymDump for CICS is 20 KB for the permanent resident program IN25COLD. The CPU and I/O resources used by CA SymDump for CICS to produce a dump will, in many cases, be less than those required by a standard dump and are unlikely to exceed those required by a standard dump. The CICS virtual storage required to view a dump online is equal to the storage the dumping task originally required (including program storage for the program active at dump time). To this, add the size of the CSA, CWA, TCTTE, TUA, and the formatted trace table of the dumping task.



**Important!** The formatted trace table is built only when the user explicitly asks to see it.

Because the CA SymDump for CICS task is conversational, these CICS areas are held while a particular dump is analyzed. When the next dump is selected, these areas are freed. Most virtual storage requests, including the trace table, are moved above the 16-megabyte line, easing the virtual storage constraint.

- **CICS Internal Trace Table**

CA SymDump for CICS dump capture and analysis relies on the availability of the CICS internal trace table for many of its diagnostic functions. A trace table of 512 KB is usually large enough to hold a typical task's processing flow. You can increase or decrease the trace table size as required by your applications. Failure to activate the internal trace will result in a loss of product functionality.

EDSA utilization increases approximately 600 KB for each concurrent CA SymDump for CICS user. If capturing the CICS Internal Trace Table using the SYMT transaction, increase the EDSA size approximately two times the size of the CICS Internal Trace Table for each concurrent user of the SYMT transaction. Adjust your SIT EDSA specification accordingly.

- **PROTDMP File**

PROTDMP files that were initialized using a release of CA SymDump for CICS after r8.0 are compatible with this release.

A CA SymDump viewing region using the SYMD transaction to view dumps must be the same CICS release as the captured dumps that are being viewed. In this case we recommend that you have a separate PROTDMP for each release of CICS to minimize confusion. Clients who use the GUI to view dumps do not have this limitation, and are free to share a single PROTDMP file to contain any version of CICS dumps.

- **Trace Formatting Region**

The TRACE FORMAT REGION requires a maximum DASD workspace of approximately 40 KB for every 4 KB block of raw unformatted CICS trace as specified in the TRCFMEGT IN25OPTS parameter. This storage is dynamically obtained and released during the format process in a serial fashion for each CA SymDump for CICS, CICS trace that is selected, for example, only one format subtask has this DASD workspace at a time. Once a trace is formatted, the DASD workspace is released before the trace actually being displayed to the user. The next trace format request will re-obtain the needed DASD workspace. For parameters controlling where to get this dynamic DASD allocation, refer to your IN25OPTS definition.

The TRACE FORMAT REGION also requires 50 KB of extended private area main storage for every 4 KB block of raw unformatted CICS trace that is being formatted. This is where the formatted trace is stored once formatted by DFHTUXX0 for review and filtering by the user. This is a cumulative amount for every trace that is simultaneously being formatted and reviewed. It is obtained during the format request, and is released once the user completely leaves the CA SymDump for CICS selection screen on the CICS region that the dump was selected from.

For a transaction dump, CA SymDump for CICS captures up to 180 4 KB blocks of raw trace data. If the maximum 180 blocks were captured and the trace was selected for formatting by three users simultaneously, the TRACE FORMAT REGION would in a serial fashion obtain and release approximately 7 MB of DASD workspace, and also get 27 MB of total extended private area, that is, 9 MB per trace.

If one of the users exits the trace, then the TRACE FORMAT REGION would be using 18 MB of extended private area for the two remaining traces. Since the traces have already been formatted by DFHTUXX0, and now reside in the extended private area of the TRACE FORMAT REGION, there is no DASD workspace being used until a new format request is received.

The new CA SymDump for CICS TRACE FORMAT REGION allows trace formatting to be offloaded from the CICS region that the trace format request originates from. This results in improved

performance within the CICS region, as the actual formatting of the CICS trace is performed by DFHTUXX0 in the TRACE FORMAT REGION. The response times for trace formatting are dependent on the size of the traces, the number of simultaneous requests being performed by a TRACE FORMAT REGION, the amount of machine resource you make available to the TRACE FORMAT REGION. Careful review of your IN25OPTS parameters that control trace formatting insures optimal performance.

You are not limited to having one TRACE FORMAT REGION in your shop. By having a different TRCFFMID parameter specification in your IN25OPTS definition, you can have different TRACE FORMAT REGIONS each with different performance characteristics. For example, you may want to have a separate TRACE FORMAT REGION with much larger thread allocation sizes (TRCFMEGT) with only a few threads (TRCFTHRD) that you would make available to your users that want to view large traces captured with the SYMT trace capture facility.

CA SymDump for CICS uses the CICS INTERNAL TRACE TABLE as its source for capturing trace information. A rule of thumb for the number of trace entries for a 4 KB trace block in the CICS internal trace table is 40. This may vary significantly with the application or system activity being traced at the client site.

With regard to the TRCFMEGM (TRACE TOTAL MEGS, MAX IS 2000) and TRCFMEGT (TRACE THREAD MEGS MIN IS 9) settings each MB of storage set aside by these parameters support or 'back' approximately 20 raw 4 KB trace blocks.

CA SymDump for CICS currently captures up to 180 4 KB trace blocks for a transaction dump, and will capture the entire CICS internal trace table for an SYMT dump. The minimum recommended thread storage limit is therefore estimated to be 9 MB, and the theoretical max thread limit for a 2 GB max storage specification is 222.

Based on these guidelines, a general approximation for the SYMT internal trace capture is  $TRCFMEGT = ((TRTABSZ/4)/20)$ . The rule for the TRCFMEGM is also an approximation, as you cannot assume that all threads will always use maximum storage allocations. For a FORMAT REGION that would only handle SYMT traces, multiply threads \* TRCFMEGT to get TRCFMEGM. Large traces, such as those captured by SYMT will significantly affect the performance of an individual TRACE FORMAT REGION. The new CA SymDump for CICS TRACE FORMAT REGION uses the IBM DFHTUXX0 (where XX corresponds to your release of CICS) module that is provided with the base CICS product. If IBM makes changes to this module due to routine maintenance, you will need to recycle your TRACE FORMAT REGION to pick up these changes. To recycle the TRACE FORMAT REGION, you would issue a cancel or purge on the address space and then restart it.

## Language Environment (LE) Considerations

If your application is executing under IBM's Language Environment (LE), region level run-time options could prevent the application from intercepting some problems. One option is TERMTHDACT, which controls what happens when an error condition occurs and can prevent control from returning to InterTest.

For example, you could see the message 'IGZ0063S An invalid sign was detected in a numeric edited sending field' in your CEE message log, but the application does not stop at that instruction. The cause of the message might be that the value of TERMTHACT is set to TRACE, which would not produce an abend and return control to InterTest. Change the value to UADUMP to allow storage to be dumped and InterTest can get control and stop at the invalid instruction.

The CLER transaction in CICS can be used to dynamically change options while the region is running. Any changes made with CLER will be reset when the region is recycled. If you want to permanently change a run-time option but do not want to globally change the LE options at your site, assemble the CEEROPT CSECT and link it into a library. Place it into the RPL to override the options for a single CICS region. The input into the assembly step would look as follows:

**Example:**

```

CEEROPT CSECT
CEEROPT AMODE ANY
CEEROPT RMODE ANY
CEEXOPT TERMTHDACT=( ( UADUMP , CESE , 96 ) , OVR )
END

```

## Activate Your Product

This article describes the steps you need to take to activate your product. Product-specific are after the instructions for all products. There are no product-specific instructions for CA InterTest Batch.

- [All Products \(see page 178\)](#)
  - [Authorization \(see page 178\)](#)
  - [CAIRIM \(see page 178\)](#)
  - [Query Activation Status \(see page 180\)](#)
- [CA InterTest for CICS \(see page 180\)](#)
  - [Activate CA InterTest for CICS from a Program \(see page 180\)](#)
  - [Start CA InterTest for CICS from a PLT \(see page 181\)](#)
  - [Automatic Startup in CICSplex Regions \(see page 182\)](#)
  - [Activate CA InterTest for CICS \(see page 182\)](#)
  - [Deactivate CA InterTest for CICS \(see page 182\)](#)
  - [Restart CICS and Run VRPT Transaction \(see page 183\)](#)
- [CA SymDump for CICS \(see page 184\)](#)
  - [Define and Schedule the Trace Format Region Job \(see page 184\)](#)
  - [Restart CICS and Run VRPT Transaction \(see page 184\)](#)
  - [Start and Stop Dump Capture Facility \(see page 185\)](#)
  - [Purge a Dump Data Set Automatically at Startup \(see page 185\)](#)
  - [Shut Down Automatically \(see page 185\)](#)
  - [Shut Down Trace Format Region \(see page 186\)](#)
- [CA SymDump Batch \(see page 186\)](#)
  - [Deactivate CA SymDump Batch \(see page 186\)](#)

## All Products

### Authorization

Before you activate your product, you must make the executables in CAI.CAVHAUTH available through STEPLIB or LINKLIST. CAI.CAVHAUTH must be APF authorized.

### CAIRIM

CAIRIM must be active on each system that runs CA InterTest or CA SymDump products. You can specify the following parameters to customize CAIRIM before activating your products.

- **STEPLIB**

The following loadlibs are required if they are not in the linklist:

- The loadlib containing CAIRIM
- All customized configuration elements, which must be APF authorized.
- CAI.CAVHAUTH
- CAI.CAVHLOAD

- **RIMPARM**

Add the following control card to the member or data set pointed to by the RIMPARM DD to initialize the CA InterTest and CA SymDump products.

```
PRODUCT(CA INTERTEST/SYMDUMP) VERSION(VHB0) INIT(CAVHINIT) PARM(220)
```

You can specify the INIT and PARM parameters as follows:

**INIT**

Specifies the name of the initialization routine. Specify one of the following values:

- **CAVHINIT** (Required)  
Initializes or refreshes the CA InterTest and CA SymDump products based on the settings in CAVHCONF.
- **CAVHTERM**  
Terminates CA SymDump Batch.

**PARM**

You can specify the following values:

- **SVC number** (Required for InterTest Batch)  
Specify an SVC value between 200 and 255. The value must be an unused SVC number that has no locks assigned to it.
- **RTLONLY**  
Activate or deactivate the CA Optimizer/II RTL but not SymDump Batch.
- **SBONLY**  
Activate or deactivate SymDump Batch but not the CA Optimizer/II RTL. You must have CA Optimizer/II RTL activated to execute CA Optimizer programs.



**Note:** CAIRIM inserts a hook into the SVC table (specifically SVC51). If you have concerns about compatibility with other dump formatting products, see [Installation Considerations \(https://docops.ca.com/display/CAITSD11/Installation+Considerations\)](https://docops.ca.com/display/CAITSD11/Installation+Considerations). The existing address for SVC51 will be saved. Control will be passed to this address depending on the value of the DUMP option of CAOETGEN.





**Note:** If you upgrade from a previous product release, an IPL is required to refresh all modules from the previous release.

## Query Activation Status

### JCL

A procedure called CAVHLPQAQ is provided to invoke the Initialization Summary program. To obtain an initialization summary for your data center, specify the following:

```
// EXEC   CAVHLPQAQ
```

The initialization summary program, CAVHLPQAQ, uses the following DD statements:

- **STEPLIB**  
Defines the library containing CA InterTest and CA SymDump Initialization Summary program.
- **SYSPRINT**  
Defines the report file.

### Report

The Initialization Summary report displays the following information:

- Module name
- Address where module resides
- Release of each module
- Date module when the module was assembled
- Time when the module was assembled
- Size of module
- Number of CA InterTest CA SymDump modules found
- Contents of module CAOESQAT (for support purposes)
- Contents of CAOETABL

## CA InterTest for CICS

### Activate CA InterTest for CICS from a Program

In one of your programs you can code a routine to activate CA InterTest for CICS by the CICS command:

```
EXEC CICS START TRANSID('CNTL') FROM(data-area)...
```

The *data-area* must contain the command:

```
CNTL=START , PROM=tablname
```

Where *tablname* is the name of the load module assembled through PROMMAC macros. The table name must be followed by a space.

If the CNTL task to be automatically initiated is to own a terminal, be sure that the terminal is in service and has the ATI (automatic task initiation) capability when the CNTL task is started.

### Issuing CNTL Commands from a Program

You can issue any CNTL command using the EXEC CICS START command. However, for the CNTL=START command, the only permitted format is:

```
CNTL=START , PROM=tablname
```

If the CNTL task is activated at a terminal, response messages are displayed there and the command is executed as if someone had entered it from that terminal. For example, if the CNTL command sets a breakpoint, the from and to terminals are the terminal at which the task was activated.

After CA InterTest for CICS starts, a program can issue an EXEC CICS START command with the data record containing:

```
CNTL=EXEC , MODULE=tablname
```

This command executes the table of commands contained in *tablname*.

## Start CA InterTest for CICS from a PLT

If the program IN25PLT is included in the PLT, it starts CA InterTest for CICS as specified in the IN25OPTS member. IN25PLT is delivered in the distribution load library.

The following CA InterTest for CICS PLTPI entry must be added after DFHDELIM so that it is part of stage 2 processing:

```
DFHPLT TYPE=ENTRY , PROGRAM=IN25PLT
```

The IN25PLT program can be called by another user-written PLT program that passes a valid CA InterTest for CICS start command in a COMMAREA to IN25PLT. An example is given next:

```
INTSTART DC C 'CNTL=START , PROM=PROMTABL ... '
.
.
.
EXEC CICS LINK
      PROGRAM( ' IN25PLT ' )
      COMMAREA( INTSTART )
      LENGTH(=Y(L' INSTART))
```

Note the following points:

- The transient data destination specified in the GLOG= *CA InterTest for CICSoption* must be defined to CICS and must be opened.
- The group DFHINQUI must be defined in the CICS startup list for IN25PLT to execute.
- The CA InterTest for CICS Help file, PROTHLF, must be defined and available.

- If installing CA SymDump with CA InterTest, the CA SymDump for CICS IN25INST program should always be placed after the CA InterTest for CICS IN25PLT program in the PLT.
- You can guarantee that the CA InterTest for CICS exits are first to receive control by adding the program entry IN25PLTX to the CICS startup PLT. IN25PLTX is required as the first program entry in Phase 1 (prior to the DFHDELIM entry) of the PLT.

## Automatic Startup in CICSplex Regions

If CA InterTest for CICS is automatically started in a CICSplex-participating region (that is, CICSplex=YES in IN25OPTS), it resynchronizes its monitoring options with any active CICSplex members upon startup.



**Note:** Resynchronization occurs whenever the CNTL=START or CNTL=RESTART command is issued in a CICSplex-participating region.

## Activate CA InterTest for CICS

After you have completed the installation steps, you can activate CA InterTest for CICS in your CICS system at any time. To activate CA InterTest for CICS, issue the following command:

```
CNTL=START
```

For CA InterTest for CICS regions installed with CICSplex support (CICSplex=YES in IN25OPTS), the CNTL=START and CNTL=RESTART commands automatically have CA InterTest for CICS resynchronize monitoring with any active regions in the CICSplex.

At this time, you can compile and assemble your programs with the CA InterTest for CICS symbolic post-processor programs as described in [Symbolic Support \(https://docops.ca.com/display/CAITSD11/Symbolic+Support\)](https://docops.ca.com/display/CAITSD11/Symbolic+Support). These programs let you use symbolic references during testing.

Remember, however, that you do *not* have to compile your programs with one of the CA InterTest for CICS symbolic post-processor programs. All of the CA InterTest for CICS monitoring facilities are available for current programs without recompiling or reassembling them. We do recommend that you recompile or reassemble current programs if CA InterTest for CICS detects errors because symbolic support greatly facilitates debugging.

To activate CA InterTest for CICS at CICS startup, perform the following procedures:

1. Manually activate CA InterTest for CICS by issuing the CNTL=START command.
2. Automatically activate CA InterTest for CICS using the PROMMAC macro or the IN25PLT PLTPI program.

## Deactivate CA InterTest for CICS

CA InterTest for CICS can be deactivated at any time by issuing the following command:

```
CNTL=END
```

For regions using the CICSplex support option, you can issue the following command to concurrently deactivate CA InterTest for CICS in all connected CICSplex regions:

```
CNTL=END,SCOPE=GLOBAL
```

CA InterTest for CICS maintains a permanent service task that runs under the transaction ID ISER. The program is IN25SERV. CICS cannot complete a normal shutdown until this task ends. It is important that you deactivate CA InterTest for CICS before CICS shutdown to terminate this task.

To deactivate CA InterTest for CICS at CICS shutdown, perform one of the following procedures:

- Manually deactivate CA InterTest for CICS by issuing the CNTL=END command.
- Automatically deactivate CA InterTest for CICS using the IN25PLTE program. If you add this CA InterTest for CICS-supplied shutdown PLT program to your shutdown program list, CA InterTest for CICS will be deactivated automatically at normal CICS termination.

Add the following CA InterTest for CICS PLTSD entry before DFHDELIM so that it is part of Stage 1 processing:

```
DFHPLT TYPE=ENTRY,PROGRAM=IN25PLTE
```

## Restart CICS and Run VRPT Transaction

After CA InterTest for CICS has been installed, restart CICS and run the CA InterTest for CICS check-out program IN25VRPT. This program checks for common mistakes that may have been made during the installation of CA InterTest for CICS.



**Important!** You are strongly advised to perform this step so errors can be detected quickly.

Use the following procedure to check out the installation of CA InterTest for CICS.

### Follow these steps:

1. Enter the transaction code VRPT. CA InterTest for CICS displays the selection menu.
2. Enter 1 in the Option field and press Enter. CA InterTest for CICS displays the first screen of programs.
3. Continue to press Enter to display the next screen.
4. The CA InterTest for CICS check-out procedure is completed when the selection menu is displayed with the option field set to 4 or beyond.



**Note:** If you are upgrading from a previous release, urge your users to use ITST Option 8 to review "What's New" in this release.

# CA SymDump for CICS

## Define and Schedule the Trace Format Region Job

The TRACE FORMAT REGION must be running within your CCI network in order for you to format CICS traces within CA SymDump for CICS.

Member TRCMSAMP contains the sample JCL for the CA SymDump for CICS TRACE FORMAT REGION. This job may be run as a normally scheduled job, or as a started task at your discretion. The TRACE FORMAT REGION can accommodate trace format requests from multiple CICS regions that have CA SymDump for CICS installed within an existing CCI NETWORK. These CICS regions may be at different releases of CICS.



**Note:** The association between a given CICS region and the TRACE FORMAT REGION is made through CCI services based on parameter TRCFFMID in your IN25OPTS definition.

Although the TRACE FORMAT REGION will probably work with the standard defaults for the relevant IN25OPTS parameters, you are encouraged to familiarize yourself with the new IN25OPTS parameters that control trace formatting. This helps to ensure that you are successful in implementing this powerful feature of CA SymDump for CICS.

Edit the JCL to conform to your installation standards and the previously completed worksheet. However, never change any of the ddnames.

You must now include a STEPLIB DD statement pointing to your CAVHLOAD load library in addition to the STEPLIB DD statement pointing to your CICS load library. This is already in the TRCMSAMP sample JCL member.

Submit this job before attempting to format CICS traces from CA SymDump for CICS.

## Restart CICS and Run VRPT Transaction

After CA SymDump for CICS is installed, restart CICS and run the CA SymDump for CICS check-out program IN25VRPT. This program checks for common mistakes that may have been made during the installation of CA SymDump for CICS.



**Important!** You are strongly advised to perform this step so errors can be detected quickly.

### Follow these steps:

1. Type the transaction code VRPT. CA SymDump for CICS displays the selection menu.
2. Type 4 in the Option field and press Enter. CA SymDump for CICS displays the first program screen.

3. Continue to press Enter to display the next screen.
4. When the selection menu appears with the option field set to 1, the CA SymDump for CICS check-out procedure is complete.

## Start and Stop Dump Capture Facility

After installing CA SymDump for CICS, you can use one of the following methods to start the dump capture facility automatically with your CICS system.

- Enter the SYMS command from a CRLP-type terminal. If you have at least one CRLP-type terminal defined in your TCT, its input sequential file can contain the SYMS command. The command executes just as if it was entered manually from a CRT, and responses to it are written in the terminal's output sequential file.
- Start CA SymDump for CICS from a PLT table. If the program IN25INST is included in the PLT table, it starts CA SymDump for CICS. IN25INST is delivered in the target load library.

## Purge a Dump Data Set Automatically at Startup

If you are starting CA SymDump for CICS from the PLT table using IN25INST, you can automatically purge old dumps at that time. If the automatic purge of dumps configuration parameter is set to Y, IN25INST purges any dumps equal to or older than the number of days specified in the automatic purge hold days parameter. The default for this parameter is one day. For more information, see [CICS Abend Analysis \(https://docops.ca.com/display/CAITSD11/CICS+Abend+Analysis\)](https://docops.ca.com/display/CAITSD11/CICS+Abend+Analysis).



**Important!** If you do not want your dumps to be automatically deleted, set the automatic purge of dumps parameter to N and refer to the dynamic purge of oldest dump parameter. This allows CA SymDump for CICS to dynamically purge non-held dumps during dump capture on a first-in, first-out basis to obtain space for the new dump.

## Shut Down Automatically

You must deactivate CA SymDump for CICS prior to shutting down your CICS region, or an abend may occur.

You can manually deactivate CA SymDump for CICS using the SYME transaction from a 3270 terminal.

You can automatically deactivate CA SymDump for CICS using the IN25ENPL program. If you add this CA SymDump for CICS-supplied shutdown PLT program to your shutdown program list, CA SymDump for CICS is deactivated automatically at normal CICS termination.

To take advantage of this feature, add the following CA SymDump for CICS PLTSD entry before DFHDELIM so it becomes part of Stage 1 processing:

```
DFHPLT TYPE=ENTRY , PROGRAM=IN25ENPL
```

## Shut Down Trace Format Region

The CA SymDump for CICS Trace Format Region can be shut down in an orderly fashion by replying shutdown to the outstanding write to operator message (CAIN5951) that is on the operator's console.

## CA SymDump Batch

### Deactivate CA SymDump Batch

To deactivate CA SymDump Batch, use the same procedure that is used to activate the product, but instead of the RIMPARM card specified in that procedure, use the following:

```
PRODUCT(CA SYMDUMP/INTERTEST) VERSION(VHB0) INIT(CAVHTERM)
```

## Installation Verification

This article describes how to run the demo programs for each application. Running the demo programs validates the installation and provides an introduction to using your product.

- [CA InterTest Batch \(see page 186\)](#)
- [CA SymDump Batch \(see page 187\)](#)
  - [Review and Edit JCL Member CARXDEMO \(see page 187\)](#)
  - [Run the Demo and Examine the Results \(see page 187\)](#)
  - [Run the CA IDMS/DB, DB2, and IMS Demo Programs \(see page 187\)](#)
  - [Review Abend Reports Using the Viewer \(see page 188\)](#)
- [CA InterTest for CICS \(see page 189\)](#)
- [CA SymDump for CICS \(see page 191\)](#)

## CA InterTest Batch

Use one of the procedures customized in the [Customize Compile Procedures \(see page 135\)](#) to compile, assemble, and link one of the demo programs provided in CAI.CAVHSAMP. The following list shows the demo program names for each compiler. For detailed information about running the demo program, see [CA InterTest Batch Demo Sessions \(https://docops.ca.com/display/CAITSD11/CA+InterTest+Batch+Demo+Sessions\)](https://docops.ca.com/display/CAITSD11/CA+InterTest+Batch+Demo+Sessions).

- **CAMRASM**  
For Assembler programmers
- **CAMRPLI**  
For PL/I programmers
- **CAMRCOB**  
For OS/VS COBOL programmers

- **CAMRCOB2**  
For COBOL II programmers



**Note:** These programs use ISPF. Programmers on CA Roscoe systems can use CAMRDMR and CAMRDMR2 for demonstration purposes.

## CA SymDump Batch

Validate the CA SymDump Batch installation by completing the following steps:

1. Review and edit JCL member CARXDEMO.
2. Run the demo and examine the results.
3. Run the database demo programs.
4. Review abend reports using the Viewer.

### Review and Edit JCL Member CARXDEMO

JCL member CAI.CAVHJCL(CARXDEMO) is a compile, postprocess, link, and run of the demo program, CAOEDEMO. Edit the JCL to conform to your installation standards.

### Run the Demo and Examine the Results

Submit CARXDEMO and examine the results. Successful execution results in a SOC7 from the RUN step.

By default, the RUN step always produces the following CA SymDum Batch reports:

- CAIOPTS File Processing report
- Abend report

### Run the CA IDMS/DB, DB2, and IMS Demo Programs

If you use the CA IDMS/DB, DB2, and IMS abend reporting features, run the demo programs that apply to your site. If your site is not using these features, skip this step.

The sample JCL members for the demo programs are stored in CAI.CAVHJCL. Review and edit the following members that apply to your shop.

- **CARXDEM1**—for CA IDMS/DB
- **CARXDEM2**—for DB2
- **CARXDEM3**—for IMS

Run the job and examine the results. A CA SymDump Batch report is produced when each job abends.

## Review Abend Reports Using the Viewer

You have an option of reviewing abend reports for validating the installation.

### Follow these steps:

1. Select the appropriate option from your ISPF main menu (updated in Concatenate ISPF File and Tailor Master Panel) to start the CAIPRINT repository viewer.  
An index of all the abend reports that were created during this installation step is displayed.

```

CA SymDump Batch ----- Report Index ----- Line 1 of 1
Command ==>                                         Scroll ==> CSR
CAPI100I Repository contains 1 report(s) -----
Repository Dsname: CAI.PRTLIB
3509 Data Records      36 Used      1.1% Full
Filter: Jobname *      Step/Tsk *      Program *      Offset *
      Comp *      UserID *      Lock *      System
----- Lvl 1
Cmd  JobName  Step/Tsk  Program  Offset  Comp  Date      Time  UserId
.   USER01X  RUN      CAOEDMO  0003B64 S=0C7  2007/03/17 16.47  USER01
-----
    
```

2. Select a report for viewing by entering an **s** in the Cmd column next to the report entry, and press Enter.  
The Report Tree panel appears.

```

CA SymDump Batch ----- Report Tree ----- Line 1 of 13
Command ==>                                         Scroll ==> CSR
CAPI052I Profile restored as of 17:58:49 on 2011/07/20 -----
JobName  USER01A  Step/Tsk  RUN      UserId  USER01  Date  2011/07/20
Program  CAOEDMO     Offset    00041B6  Comp    S=0C7   Time  17.57
----- Lvl 2
Cmd  LineNo.  Description
.    1 - Job=USER01A
.    1 | -Input Options Display
    
```

If the report tree is displayed, you have successfully validated your installation.



**Note:** For a detailed description of the CAIPRINT repository viewer and its functionality, see [Batch Abend Analysis \(https://docops.ca.com/display/CAITSD11/Batch+Abend+Analysis\)](https://docops.ca.com/display/CAITSD11/Batch+Abend+Analysis).

## CA InterTest for CICS

Demonstration programs are distributed for Assembler, COBOL, and PL/I in the source format. You can assemble or compile them using the jobs found in CAVHJCL. The PROCs executed by these sample jobs are defined in CAVHPROC.

The sample jobs provide symbolic parameter overrides to the respective PROC members. You can modify the appropriate PROC members to define these symbolic parameters as defaults. You need to modify the PROC members to define certain default symbolic parameters to comply with your installation standards. For detailed information about running the demo program, see [CA InterTest for CICS Primers \(https://docops.ca.com/display/CAITSD11/CA+InterTest+for+CICS+Primers\)](https://docops.ca.com/display/CAITSD11/CA+InterTest+for+CICS+Primers).

The following list details the four members in CAI.CAVHJCL.

- **CABADMAJ**

The ASMDEMO job prepares the Assembler language demonstration program for execution. For more information about using the ASMDEMO program, see [Assembler Primer \(https://docops.ca.com/display/CAITSD11/Assembler+Primer\)](https://docops.ca.com/display/CAITSD11/Assembler+Primer). This job contains the following steps:

- **Step ALOCLIBS**

Allocates an object library and a load library. This step can be deleted if you are using existing libraries.

- **Step ASMDEMO**

Executes PROC CABAASMP to translate, assemble, and postprocess the Assembler language demonstration program ASMDEMO.

- **Step LKEDEMO**

Executes program IEWL to bind the object module created by Step ASMDEMO.

- **CABADM CJ**

The CABADM CJ job prepares the COBOL language demonstration program (COBDEMO) for execution. For more information about using the COBDEMO program, see [COBOL Primer \(https://docops.ca.com/display/CAITSD11/COBOL+Primer\)](https://docops.ca.com/display/CAITSD11/COBOL+Primer). This job contains the following steps:

- **Step ALOCLIBS**

Allocates an object library and a load library. This step can be deleted if you are using existing libraries.

- **Step COBDEMO**

Executes PROC CABACOBP to compile and postprocess the COBOL language demonstration program COBDEMO.

- **Step LKEDEMO**

Executes program IEWL to bind the object module from the previous step.

- **Step CSBIN25**

Executes PROC CABACOBP to compile and postprocess the COBOL language demonstration subprogram CSBIN25.

- **Step ASBIN25**

Executes PROC CABAASMP to translate, assemble, and postprocess the Assembler language demonstration subprogram ASBIN25.

- **Step COBDEML**

Executes PROC CABACOBP to compile and postprocess the COBOL language demonstration program COBDEML.

- **Step LKEDEML**

Executes program IEWL to bind the object modules from the previous steps into a composite load module.

**Step GLNKDEML**

Executes program IN25LINK to prepare COBDEML for composite mapping demonstration.

▪ **CABADMDJ**

The CABADMDJ job prepares the COBOL DB2 demonstration program (DB2DEMO) for execution. For more information about using the DB2DEMO program, see [COBOL Primer \(https://docops.ca.com/display/CAITSD11/COBOL+Primer\)](https://docops.ca.com/display/CAITSD11/COBOL+Primer). This job contains the following steps:

**Step ALOCLIBS**

Allocates an object library and a load library. This step can be deleted if you are using existing libraries.

**Step DB2TRAN**

Executes the DB2 Translator for the demonstration program DB2DEMO.

**Step COBDEMO**

Executes PROC CABACOBP to compile and postprocess the COBOL language demonstration program DB2DEMO.

**Step LKEDDEMO**

Executes program IEWL to bind the object module from the previous step.

**Step BINDPLAN**

Executes program IKJEFT01 to bind the plan for DB2DEMO program.

▪ **CABADMPJ**

The CABADMPJ job prepares the PL/I language demonstration programs. For more information about using the PL1DEMO program, see [PL/I Primer \(https://docops.ca.com/pages/viewpage.action?pageId=386435436\)](https://docops.ca.com/pages/viewpage.action?pageId=386435436). This job contains the following steps:

**Step ALOCLIBS**

Allocates an object library and a load library. This step can be deleted if you are using existing libraries.

**Step PLIDEMO**

Executes PROC CABAPLIP to compile and postprocess the PL/I language demonstration program PL1DEMO.

**Step LKEDDEMO**

Executes program IEWL to bind the object modules from the previous step.

**Step PSBIN25**

Executes PROC CABAPLIP to compile and postprocess the PL/I language demonstration subprogram PSBIN25.

**Step ASBIN25**

Executes PROC CABAASMP to translate, assemble, and postprocess the Assembler language demonstration subprogram ASBIN25.

**Step PLIDEML**

Executes PROC CABAPLIP compile and postprocess the PL/I language demonstration program PL1DEML.

**Step LINKDEML**

Executes program IEWL to bind the object modules from the previous steps into a composite load module.

**Step GLNKDEML**

Executes program IN25LINK to prepare PL1DEML for composite mapping demonstration.

## CA SymDump for CICS

After you install CA SymDump for CICS and start CICS, you can run the demonstration programs to force and then view a transaction dump.

**Follow these steps:**

1. Entering the transaction code SYMD to verify that CA SymDump for CICS was started.
2. Type 4 in the Option field and press Enter.  
CA SymDump for CICS will start or will tell you that it was already started.
3. Type X to terminate SYMD.
4. Type one or more of the following transaction codes. After the menu appears, press Enter to force an ASRA abend and the creation of a CICS transaction dump that should be captured and displayable by CA SymDump for CICS.  

DEMA	Assembler program ASRA (0C7)
DEMC	COBOL program ASRA (0C7)
DEMD	COBOL DB2 program ASRA (0C7)
DEMP	PL1 program ASRA (0C7)
5. Type the transaction code SYMD.
6. Type 1 on the Option field and follow the menu instructions to view the dumps created in the previous Step 4.