

# **Advantage™** **CA-InterTest® Batch**

## **Getting Started**

**31**



Computer Associates®

MAN09145511E

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been 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 the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2003 Computer Associates International, Inc.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

## Chapter 1: Welcome

CA Technology Services: Enabling Solutions Through Experience .....	1-1
CA Education Services .....	1-2
Computer Associates: The Software That Manages eBusiness .....	1-2
For More Information .....	1-3
Debugging Made Easy .....	1-3
Testing and Fault Management .....	1-3
Features .....	1-4
Summary of Changes for This Release .....	1-5

## Chapter 2: System Requirements

Operating Environments .....	2-1
TSO LOGON Requirements for Foreground Debugging .....	2-1
Compiler Support .....	2-2
Supported Documentation Viewers .....	2-2
CA Common Services .....	2-3
CAIRIM .....	2-3
CA LMP .....	2-4
Installing CA Common Services .....	2-4
Hardware Requirements .....	2-5
Distribution Libraries .....	2-5
Target Libraries .....	2-6
Additional Files .....	2-7

---

## Chapter 3: Installation

Preinstallation Considerations	3-1
SMP/E Considerations	3-1
RECEIVE Processing	3-2
APPLY Processing	3-2
ACCEPT Processing	3-2
Package Contents	3-3
CA Common Services Prerequisite	3-3
Summary of Installation Steps	3-4
Upgrading from an Earlier Release	3-6
Step 1. Install CA Common Services	3-6
Step 2. Tailoring the LMP Key	3-7
Step 3. Review System Requirements	3-7
Step 4. Complete the Installation Worksheet	3-8
Installation Worksheet	3-8
Step 5. Load Installation SAMPJCL Library	3-10
Step 6. Unloading the Documentation	3-11
Step 6a. Unloading BookManager Files from the Product Cartridge	3-11
Changing the Prefix in the Bookshelf Definition	3-12
Downloading Files for Use on a PC	3-12
Step 6b. Unloading the Acrobat PDF File from the Product Cartridge	3-13
Printing the Quick Reference Card	3-13
Step 7. Allocate Target and Distribution Libraries	3-14
Step 8. Define SMP/E Libraries and ZONES	3-15
Step 8a. Allocate Private SMP/E Libraries (Optional)	3-15
Step 8b. Define Additional Product Target and Distribution Zones	3-16
Step 9. Customize the SMP Procedure	3-16
Step 10. Receive the Product	3-17
Step 11. Customization	3-18
Step 11a. Define Excluded Programs	3-18
Step 11b. Batch Link User Exit	3-18
Step 12. Apply the Product	3-19
Step 13. Activate the Batch Link Feature	3-19

---

Step 14. Allocate Additional Files .....	3-20
Step 14a. Allocate PROFLIB .....	3-20
Step 14b. Allocate PROTSYM .....	3-21
Concurrently Updating the PROTSYM File .....	3-21
Step 14c. Allocate ALIBs (Optional) .....	3-22
Step 14d. Allocate INCLIBs (Optional) .....	3-22
Step 15. Customize Options .....	3-23
Step 16. Customize CLISTs .....	3-25
CAMRDRVR .....	3-25
MR31CLST .....	3-26
MR31CIMS .....	3-27
MR31CBTS .....	3-28
MR31CDB2 .....	3-29
MR31STRT .....	3-29
MR31KEYS .....	3-30
CAMRCMD .....	3-30
Step 17. Customize Compile PROCs .....	3-30
COBINT .....	3-30
COB2INT .....	3-31
PLIINT .....	3-32
ASMINT .....	3-32
Step 18. Adding to ISPF Main Panel (Optional) .....	3-33
Step 19. Add ISPF Libraries to TSO LOGON Procedure (Optional) .....	3-33
Step 20. Customize Advantage CA-Roscoe (Optional) .....	3-34
MR31ETSO .....	3-34
MR31EPL .....	3-34
INTROS .....	3-35
Step 21. Run Demo (Optional) .....	3-37
Step 22. Review COBOL RTL Considerations .....	3-38
Step 23. Review and Customize for BTS .....	3-38
Screen Handling Solution for the Message BTS0067I .....	3-40
Applications Using Message Switching and DISPLAY or Figure Verbs .....	3-41
Step 24. Accept the Product .....	3-42
Step 25. Save Materials and Output .....	3-42

---

## Chapter 4: Applying Maintenance

Overview .....	4-1
SMP/E Libraries .....	4-1
APPLY and ACCEPT Commands .....	4-2
Maintenance Delivery .....	4-3
Informational APAR or PIB .....	4-3
Program Temporary Fixes .....	4-4
PTF Maintenance Tape Installation .....	4-4
Overview of Maintenance Steps .....	4-5
Step 1. Review and Follow the Cover Letter Instructions .....	4-6
Step 2. Load the Sample JCL from the Tape .....	4-6
Step 3. Unload the Documentation .....	4-7
Step 4. Customize the SMP/E Procedure .....	4-7
Step 5. Receive Maintenance .....	4-7
Step 6. APPLY CHECK Maintenance .....	4-8
Step 7. Restore Any Applicable SYSMODs .....	4-9
Step 8. Apply Maintenance .....	4-10
Step 9. Reapply Any Applicable SYSMODs .....	4-10
Step 10. Refresh Batch Link Feature .....	4-11
Step 11. Execute IVP to Validate Maintenance .....	4-11
Step 12. Accept Maintenance .....	4-11
Step 13. Save All Materials and Output .....	4-12

---

## Appendix A: Symbolic File

About the Symbolic File .....	A-1
Using the Post-Processors with Multiple Product Releases .....	A-2
COBOL Symbolic Support .....	A-2
Notes for Post-Processing COBOL Programs .....	A-3
IBM COBOL Compiler Options .....	A-4
Advantage CA-Optimizer and Advantage CA-Optimizer/II Options .....	A-5
Post-Processor Options .....	A-6
Controlling Printed Output with the CUTPRINT Option .....	A-7
Notes about the CUTPRINT option: .....	A-7
Saving Your Listing for Display with the LISTER Option .....	A-8
Notes about the LISTER Option .....	A-8
Setting Data as Nonpurgable .....	A-9
COBOL JCL Examples .....	A-10
COBOL/VS or Advantage CA-Optimizer Example .....	A-10
COBOL II and Above or Advantage CA-Optimizer/II Example .....	A-12
Assembler Symbolic Support .....	A-14
Assembler Options .....	A-15
Post-Processor Options .....	A-15
Controlling Printed Output with the CUTPRINT Option .....	A-16
Notes about the CUTPRINT option: .....	A-16
Saving Your Listing for Display with the LISTER Option .....	A-17
Notes about the LISTER Option .....	A-17
Setting Data as Nonpurgable .....	A-18
Assembler JCL Example .....	A-19
PL/I Symbolic Support .....	A-21
Compiler Options .....	A-22
Installation Options .....	A-22
Unsupported Features .....	A-23
Post-Processor Options .....	A-23
Controlling Printed Output with the CUTPRINT Option .....	A-24
Notes about the CUTPRINT option: .....	A-24
Saving Your Listing for Display with the LISTER Option .....	A-25
Notes about the LISTER Option .....	A-25
Setting Data as Nonpurgable .....	A-26

---

PL/I JCL Example .....	A-27
------------------------	------

## Appendix B: Symbolic File Maintenance

Control Statements .....	B-1
Examples .....	B-3
Initializing a Symbolic File .....	B-3
Purging Symbolic Information by Age .....	B-4
Deleting Symbolic Information by Program .....	B-4
Generating Reports and Purging Programs .....	B-4
Unloading Programs .....	B-5
Reloading Programs .....	B-5
Printing a Program Listing .....	B-6
Reorganizing the Symbolic File .....	B-7

## Index

# Welcome

---

This book is designed to introduce Advantage™ CA-InterTest® Batch to you, in an efficient and visual manner. By the time you have finished reading this guide, you will have an overview of the wide scope of the product and its usability will be familiar to you. It is important to us that you feel comfortable with Advantage CA-InterTest Batch before you begin to use it.

## CA Technology Services: Enabling Solutions Through Experience

When it comes to getting on the information fast track, CA Technology Services can recommend and install a full suite of portal and knowledge management solutions to keep your business moving. In addition, our associates offer the proprietary expertise on custom-fitting your enterprise for solutions ranging from life cycle management, data warehousing and next-level business intelligence. Our experts will leave you with the technology and knowledge tools to fully collect, exploit and leverage your data resources and applications.

## CA Education Services

Computer Associates Global Education Services (CA Education) offerings include instructor-led and computer-based training, product certification programs, third-party education programs, distance learning and software simulation. These services help to expand the knowledge base so companies are better able to utilize CA's products more efficiently, contributing to their greater success. CA Education is developed to assist today's technologists in everything from understanding product capabilities, to implementation and quality performance. Because the vast community of education seekers is varied, so too are CA's methods of instruction. CA Education is committed to provide a variety of alternatives to traditional instructor-led training, including synchronous and asynchronous distance learning, as well as Unicenter simulation.

For training that needs to be extended to a wider audience — for a fraction of the cost and logistical hassle of sending everybody away to a class — CA Education offers excellent distance learning options.

## Computer Associates: The Software That Manages eBusiness

The next generation of eBusiness promises unlimited opportunities by leveraging existing business infrastructures and adopting new technologies. At the same time, extremely complicated management presents challenges — from managing the computing devices, to integrating and managing the applications, data, and business processes within and across organizational boundaries — and look to CA for the answers. CA has the solutions available to help eBusinesses address these important issues. Through industry-leading eBusiness Process Management, eBusiness Information Management, and eBusiness Infrastructure Management offerings, CA delivers the only comprehensive, state-of-the-art solutions, serving all stakeholders in this extended global economy.

## For More Information

After walking through this *Getting Started* guide, you can refer to the numerous resources available to you for additional information. Your Advantage CA-InterTest Batch product tape contains useful instructional documents that showcase your software as well as detailed explanations about the product's comprehensive, feature-rich components.

## Debugging Made Easy

### Testing and Fault Management

Advantage CA-InterTest Batch is a comprehensive interactive tool for testing and debugging batch and IMS/DC applications through a TSO/ISPF or Advantage™ CA-Roscoe® Interactive Environment interface. With Advantage CA-InterTest Batch, programmers use interactive, online facilities with the actual source listing for debugging, rather than dumps or reports. This enables many errors to be resolved in a single test session. Advantage CA-InterTest Batch intercepts all applicationabend conditions and identifies the source code statement that caused each error. Interactive debugging functions include the ability to set breakpoints, trace program execution, place data items in a keep window on the source listing screen, examine and modify the values of data items and find a data string in working storage. A user-friendly, menu-driven allocation and JCL conversion facility simplifies the task of allocating program files for foreground execution or preparing JCL for batch execution.

## Features

You will enjoy the following easy to use features:

- Full screen source listing display
- Interactive debugging through a TSO/ISPF or Advantage CA-Roscoe interface
- All application abends are intercepted and the source statement that caused the abend is identified
- Batch Link Facility allows users to debug multi-step jobs running in batch
- Unconditional and conditional breakpoints can be set at specific labels or statements, at entry to or exit from a program or all programs, or when the value of a variable changes
- Automatic trace function displays the flow of the application execution in forward or reverse sequence
- Frequency counters identify how many times each statement has been executed
- Automatic Program Variable Display (AUTOKEEP) displays the contents of all program variables or data names referenced in an active statement of a program
- Data items in the Keep Window are modifiable.
- Sizable Keep Window becomes scrollable when it contains more data item entries than can be displayed in the space provided
- Data items can be symbolically displayed and modified
- Interactive display of file attributes
- Interactive control block information displays
- Application debugging is fully supported for many databases, including Advantage™ CA-IDMS®/DB Database, Advantage™ CA-Datcom®/DB Database, IMS, and DB2
- Built-in reports display execution summaries, histograms, untested code, and potential bottlenecks

- Automatic conversion of batch JCL into CLIST or ALIB statements with validation simplifies and reduces user setup time for application testing
- Supported JCL sources include AllFusion™ CA-Librarian® for z/OS and OS/390, AllFusion™ CA-Panvalet® for z/OS and OS/390, and PDS
- Easy-to-use ISPF interface and similarities to Advantage™ CA-InterTest® for CICS shorten the learning curve
- Split screen facilities allow a fully functional ISPF or Advantage CA-Roscoe second window, including the ability to edit source code while debugging
- Alias names can be assigned to any data item

## Summary of Changes for This Release

A summary of new features and enhancements to existing features for this release is provided in the Release Summary. The Release Summary can be found on the product cartridge along with the *User Guide*, the *Quick Reference* card, and online versions of this guide. To locate the documents see the section [Step 6. Unloading the Documentation](#) in the “Installation” chapter of this guide.



# System Requirements

---

## Operating Environments

Advantage CA-InterTest Batch 3.1 operates on any z/OS or OS/390 system with TSO and ISPF 4.1 or above, or on any z/OS or OS/390 system with Advantage CA-Roscoe 6.0 and above with ETSO.

## TSO LOGON Requirements for Foreground Debugging

The TSO LOGON requirements for Advantage CA-InterTest Batch are region size (when logging on) and the number of DD statements that can be dynamically allocated at any one time during the session (DYNAMBR).

Advantage CA-InterTest Batch uses about one megabyte of virtual storage, so users should LOGON with a region large enough for both Advantage CA-InterTest Batch and the program being debugged. A region of between 3.0 and 4.0 megabytes is sufficient for most processing.

The DYNAMBR number should be high for Advantage CA-InterTest Batch, especially when running IMS/BTS applications in foreground. Allocation checks to see if the number in DYNAMBR has been exceeded and has no other effect on the TSO session.

Because Advantage CA-InterTest Batch has a dynamic STEPLIB facility, it is not necessary to allocate libraries containing dynamically called modules at LOGON time.

## Compiler Support

Advantage CA-InterTest Batch can debug any non-authorized program compiled with any COBOL or PL/I compiler supported by IBM. Advantage CA-InterTest Batch can also debug any non-authorized assembler program assembled with Assembler F, Assembler H, or the High Level Assembler.

## Supported Documentation Viewers

Documentation for Advantage CA-InterTest Batch is provided in both PDF format and BookManager format on the product tape. Our intention is to provide you with the convenience of being able to view the documents on your PC in PDF format and to view the documents on your mainframe in BookManager format. However, if you want to view the BookManager formats on your PC, you should use at least version 2.4 of IBM Softcopy Reader. To view the guides in PDF format on your PC, you should use version 6.0 of Adobe Acrobat Reader. See the section, [Step 6. Unloading the Documentation](#), in the chapter “Installation” of this guide for instructions on unloading the documentation sets from the product tape.

## CA Common Services

Advantage CA-InterTest Batch exploits the installation of the following CA Common Services services:

- CAIRIM
- CA LMP

These CA Common Services services must be installed or maintained at the genlevel indicated on the cover letter for the product.

This section presents an overview of each of these services and explains how your product implements each service. Refer to the CA Common Services documentation for detailed information.

### CAIRIM

CAIRIM, CAI Resource Initialization Manager, is the common driver for a collection of dynamic initialization routines that eliminate the need for user SVCs, SMF exits, subsystems, and other installation requirements commonly encountered when installing systems software. These routines are grouped under the Computer Associates z/OS and OS/390 dynamic service code, S910. Some of CAIRIM's features are:

- The ability to obtain SMF data
- Verification of proper software installation
- Installation of z/OS and OS/390 interfaces
- Automatic startup of CA and other vendor products
- Proper timing and order of initialization
- CAIRIM is required in order to run CA LMP

## CA LMP

The CA License Management Program provides a standardized and automated approach to the tracking of licensed software. It uses common realtime enforcement software to validate the user's configuration. CA LMP reports on activities related to the license, usage and financials of program solutions. The routines that accomplish this are integrated into the Computer Associates z/OS and OS/390 dynamic service code, S910 (the CAIRIM service). CA LMP features include:

- Sharing a common Key Data Set among many CPUs
- Using “check digits” to detect errors in transcribing key information
- Entering Execution Keys without affecting any CA software solution already running
- No special maintenance requirements
- CA LMP is required to run Advantage CA-InterTest Batch

## Installing CA Common Services

Advantage CA-InterTest Batch requires the CA Common Services components CAIRIM and CA LMP.

If these CA Common Services components are not already installed on your system, you must do so before proceeding with this installation. Refer to the CA Common Services *Getting Started* guide for detailed instructions.

## Hardware Requirements

Advantage CA-InterTest Batch runs on any IBM S/370, S/390 or compatible.

### Distribution Libraries

The following table estimates the minimum required disk space for the distribution libraries required to install Advantage CA-InterTest Batch.

<b>Name</b>	<b>Blksize</b>	<b>Blocks</b>	<b>Dir Blks</b>	<b>Description</b>
CAI.CMR31LLD	6144	800	110	Product distribution load library
CAI.CMR31MLD	3120	5	1	Product distribution macro library
CAI.CMR31SLD	3120	410	15	Product distribution source library
CAI.CVH31LLD	6144	210	5	Debugging Tools distribution load library
CAI.CVH31SLD	3120	15	5	Debugging Tools distribution source library
CAI.CZ270LLD	6144	260	25	Unicenter® CA-JCLCheck™ Utility distribution load library
CAI.CZ270MLD	3120	810	20	Unicenter CA-JCLCheck distribution source library

## Target Libraries

The following table estimates the minimum required disk space for the target libraries required to install Advantage CA-InterTest Batch. These libraries must be allocated if they do not already exist for some other Computer Associates product in your shop.

<b>Name</b>	<b>Blksize</b>	<b>Blocks</b>	<b>Dir Blks</b>	<b>Description</b>
CAI.CAICLIB	3120	35	5	common CLIST library
CAI.CAIISPM	3120	30	5	common ISPF msg library
CAI.CAIISPP	3120	160	20	common ISPF panel library
CAI.CAILIB	6144	1100	55	common load library
CAI.CAIMAC	3120	370	10	common macro library
CAI.CAIPROC	3120	20	5	common procedure library
CAI.CAISRC	3120	210	5	common source library
CAI.PPOPTION	3120	500	10	common options library

## Additional Files

The following table estimates the minimum required disk space for additional files that are not under SMP control.

<b>Name</b>	<b>Blksize</b>	<b>Blocks</b>	<b>Dir Blks</b>	<b>Description</b>
CALPNLLIB	3120	250	35	Advantage CA-InterTest Batch panel library
CALRPFLIB	3120	30	5	Advantage CA-InterTest Batch RPF library



This chapter explains how to install Advantage CA-InterTest Batch.

## Preinstallation Considerations

### SMP/E Considerations

Advantage CA-InterTest Batch is installed using SMP/E. This chapter describes installation using SMP/E. The machine-readable program materials required for installation are distributed as a multfile installation tape in SMP/E format. For a general description of the tape format, see the next section.

Use the IBM-supplied System Modification Program/Extended (SMP/E) to install, customize and maintain Advantage CA-InterTest Batch. SMP/E enables you to control these activities in a consistent manner.

There are three major operations performed by SMP/E when installing a product or performing maintenance. These operations manage a structure wherein a given product is present in two places: distribution libraries and target (operating) libraries. The distribution libraries are used for maintenance operations only. The product executes from the target libraries.

The operations performed are as follows:

- RECIEVE processing
- APPLY processing
- ACCEPT processing

### **RECEIVE Processing**

The installation tape is loaded by SMP/E into temporary data sets. If any error is detected, or the user wishes to stop the process at this point, a REJECT operation may be run, which undoes anything done during RECEIVE processing.

### **APPLY Processing**

During APPLY processing, SMP/E performs the operations dictated by the modification control statements (MCS), and updates the target libraries. The user may then test the modification. If the installation is to be aborted at this point, a RESTORE operation may be run, which restores the system libraries from the distribution libraries.

### **ACCEPT Processing**

When the ACCEPT operation is run, the modification is permanently placed in the distribution libraries. There is no direct way to undo the modification once ACCEPT has been run.

## Package Contents

Computer Associates provides the following materials for installation and use of Advantage CA-InterTest Batch.

- A standard label magnetic cartridge or standard label magnetic tape recorded at 6250 BPI (DCB=DEN=4) containing Advantage CA-InterTest Batch. The Volume Serial number is FCyymm. The tape contents are as follows:
  - DSN=CAI.SAMPJCL (File number 9): This is an unloaded PDS containing all of the sample JCL referred to in this document pertaining to installation and execution of Advantage CA-InterTest Batch. Load via IEBCOPY.
  - DSN=SMPMCS (File number 32): This file contains the SMP modification control statements (MCS) used by SMP to install Advantage CA-InterTest Batch.
  - Files beyond 32 are the SMP RELFILES that SMP download to disk during RECEIVE processing.
- Comprehensive documentation that explains how to use, install, customize and maintain Advantage CA-InterTest Batch.
- Be certain that you have taken into account any site-relevant instructions that may have been included in the release notes you received with Advantage CA-InterTest Batch.

## CA Common Services Prerequisite

Make sure you install the required CA Common Services components before proceeding with an SMP/E installation of Advantage CA-InterTest Batch.

## Summary of Installation Steps

The following list summarizes the steps involved in the Advantage CA-InterTest Batch installation process. Review this list before attempting to install Advantage CA-InterTest Batch, and then use it as a checklist when installing the product.

<b>Step</b>	<b>Description</b>
Step 1.	Install CA Common Services
Step 2.	Tailoring the LMP Key
Step 3.	Review System Requirements
Step 4.	Complete the Advantage CA-InterTest Batch Installation Worksheet
Step 5.	Load Installation SAMPJCL Library
Step 6.	Unloading the Documentation
	6a. Unloading BookManager File from the Product Cartridge
	6b. Unloading Acrobat PDF File from the Product Cartridge
Step 7.	Allocate Target and Distribution Libraries
Step 8.	Define SMP/E Libraries and ZONES
	8a. Allocate Private SMP/E Libraries (Optional)
	8b. Define Additional Product Target and Distribution Zones
Step 9.	Customize the SMP Procedure
Step 10.	Receive the Advantage CA-InterTest Batch Product
Step 11.	Customization
	11a. Define Excluded Programs
	11b. Btach Link User Exit
Step 12.	Apply the Advantage CA-InterTest Batch Product

---

<b>Step</b>	<b>Description</b>
Step 13.	Activate the Batch Link Feature
Step 14.	Allocate Additional Files
14a.	Allocate PROFLIB
14b.	Allocate PROTSYM
14c.	Allocate ALIBs (Optional)
14d.	Allocate INCLIBs (Optional)
Step 15.	Customize the Advantage CA-InterTest Batch Options
Step 16.	Customize the Advantage CA-InterTest Batch CLISTs
Step 17.	Customize the Advantage CA-InterTest Batch compile PROCs
Step 18.	Add Advantage CA-InterTest Batch Option to ISPF Main Panel (Optional)
Step 19.	Add Advantage CA-InterTest Batch ISPF Libraries to TSO LOGON Procedure (Optional)
Step 20.	Customize Advantage CA-Roscoe for Advantage CA-InterTest Batch (Optional)
Step 21.	Run Demo (Optional)
Step 22.	Review COBOL RTL Considerations
Step 23.	Review and Customize for BTS
Step 24.	Accept the Advantage CA-InterTest Batch Product
Step 25.	Save Materials and Output

---

## Upgrading from an Earlier Release

If you are upgrading from an earlier release, complete replacement of the Advantage CA-InterTest Batch product is recommended. However, retain your ALIB files and any JCL that you converted for Batch Link. Note that the PROFLIB for this release is not compatible with a PROFLIB used with release 2.0 and below.

Also, if you have existing PROTSYM datasets that you created from other products, such as Advantage CA-InterTest for CICS, that contain PL/I and Assembler programs, you must re-populate the PROTSYM. Also any COBOL programs in existing PROTSYMs must be re-populated if they were created with Advantage CA-InterTest for CICS release 6.0 or below.

### Step 1. Install CA Common Services

Advantage CA-InterTest Batch uses the CA Common Services component CAIRIM. If CA Common Services is not already installed, you must install it before continuing. See the CA Common Services *Getting Started* guide for complete instructions.

CAIRIM, the Resource Initialization Manager, is a started task used for initialization services by many Computer Associates solutions. It is a short-running job that sets up the environment required by the products, which use it, and then it terminates.

You must customize the CAIRIM procedure (CAS9). See the CA Common Services *Getting Started* guide for detailed information. Customize the procedure as follows:

**STEPLIB** -- The CAIRIM load library. If Advantage CA-InterTest Batch was installed into a different target load library than CAIRIM, you must add the Advantage CA-InterTest Batch CAILIB data set to the STEPLIB concatenation. The CAILIB data set must be APF authorized.

**RIMPARM** -- Add the following control card to the member or data set pointed to by the RIMPARM DD for initialization of Advantage CA-InterTest Batch:

```
PRODUCT(CA-InterTest Batch) VERSION(MR31) INIT(MR31INIT) PARM(220)
```

Where the PARM field indicates which user SVC is to be used for the Link feature of Advantage CA-InterTest Batch. The PARM must be a value between 200 and 255 and must be an SVC number that is not being used. See the CA Common Services *Reference Guide* for an explanation of the other parameters on this control card. Advantage CA-InterTest Batch allows you to specify the PARM parameter on this card.

## Step 2. Tailoring the LMP Key

For instructions on installing and tailoring the Advantage CA-InterTest Batch LMP execution key, see the CA Common Services *Getting Started* guide.

## Step 3. Review System Requirements

Be sure you review the system requirements to make certain that all the requirements are met before installing Advantage CA-InterTest Batch.

## Step 4. Complete the Installation Worksheet

The Installation Worksheet simplifies the task of tailoring the supplied JCL. Answer each question on the worksheet, filling in the blanks with the appropriate information. Default values are noted, so if the default value is acceptable, leave the item blank on the worksheet. However, you must supply appropriate volume serial numbers.

### Installation Worksheet

1. What is your installation generic unit name for permanent DASD volumes? Default: PERMDA=SYSDA PERMDA=\_\_\_\_\_
2. What is your installation generic unit name for temporary work DASD? Default: WORK=SYSDA WORK=\_\_\_\_\_
3. What is your installation generic unit name for the shipped installation tape? Default: TAPE=TAPE TAPE=\_\_\_\_\_
4. What is your installation standard SYSOUT class for CA product installs and SMP output? Default: SYSOUT=\* SYSOUT=\_\_\_\_\_
5. What is the DSname high-level qualifier for the Advantage CA-InterTest Batch installation libraries (both target and distribution libraries)? Default: PREFIX=CAI. PREFIX=\_\_\_\_\_
- a. What VOL=SER do you want for the target libraries? Default: TGTVOL= TGTVOL=\_\_\_\_\_
- b. What VOL=SER do you want for the distribution libraries? Default: DLIBVOL= DLIBVOL=\_\_\_\_\_

6. What is the DSname high-level qualifier for the SMP libraries to be used for the installation of CA products? Default: CAI=CAI. CAI=\_\_\_\_\_
- a. What VOL=SER do you want for the SMP libraries, if these libraries do not already exist? Default: VOLSER= VOLSER=\_\_\_\_\_
7. What VOL=SER do you want for the SMP TLIB data set? TLIB=\_\_\_\_\_
8. What is the data set name you will use for the PROTSYM? Default: PROTSYM=CAI.PROTSYM PROTSYM=\_\_\_\_\_
9. What User SVC do you wish to use for the Batch Link Feature? Note: This was determined in Step 1. SVC=\_\_\_\_\_

## Step 5. Load Installation SAMPJCL Library

The Advantage CA-InterTest Batch product installs using SMP/E. The product tape received with this package contains all the necessary data to install and execute Advantage CA-InterTest Batch. It is a standard label, 3480 cartridge. Before installing the product, you should load the sample JCL library from tape. This is the ninth file on the tape, DSN=CAI.SAMPJCL, and it is in IEBCOPY unloaded format. Use the following JCL as a model to load the sample JCL library to DASD.

```
//LOAD      EXEC PGM=IEBCOPY
//SYSPRINT  DD SYSOUT=A
//SYSUT1    DD DISP=(OLD,KEEP) ,
//           DSN=CAI.SAMPJCL ,
//           UNIT=TAPE,           <=== generic 3480 cartridge
//           VOL=SER=FCyymm
//           LABEL=(9,SL) ,
//           DCB=DEN=4
//SYSUT2    DD DISP=(NEW,CATLG,DELETE) ,
//           DSN=CAI.INTBATCH.SAMPJCL, <=== your DSN
//           UNIT=SYSDA,           <=== your generic DASD
//           VOL=SER=XXXXXX       <=== permanent DASD volser
//           SPACE=(3120,(40,20,10)), <=== minimum space required
//           DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSUT3    DD UNIT=SYSDA,
//           SPACE=(TRK,(5,5))
//SYSUT4    DD UNIT=SYSDA,
//           SPACE=(TRK,(5,5))
//SYSIN     DD DUMMY
```

Once this job has ended, your library contains all of the JCL needed to complete the installation of Advantage CA-InterTest Batch.

In order to satisfy your data center's needs, certain tailoring of JCL is necessary while executing the steps on the following pages. See the Installation Worksheet completed in Step 4 to obtain values for the various JCL parameters.

## Step 6. Unloading the Documentation

The updated documentation is provided on the product cartridge in both PDF and BookManager format.

If you want to print the manuals, it is recommended that you use the PDF files. You can use Adobe's Acrobat Reader software to view and print the PDF files on any PC platform. The Acrobat Reader is freely and easily available from [www.adobe.com](http://www.adobe.com). When downloading the Reader, check the "with Search" option, which lets you perform searches across multiple books if an index (.pdx file) is provided.

### Step 6a. Unloading BookManager Files from the Product Cartridge

In addition to the printed documentation provided by Computer Associates, updates to the Advantage CA-InterTest Batch documentation are provided in IBM BookManager format on the Advantage CA-InterTest Batch product cartridge in files 19 and 20. The BookManager READ files can be unloaded using the JCL found in SAMPJCL member MR31BOOK. Edit the JCL to conform to your installation standards and submit the job.

The Advantage CA-InterTest Batch documentation set consists of six sequential data sets that contain the following:

- Bookshelf definition
- Search index
- The *Getting Started* guide
- The *User Guide*
- The *Quick Reference*
- The *Release Summary*

## Changing the Prefix in the Bookshelf Definition

The bookshelf definition (prefix BKSHELF) contains the data set names BookManager uses to reference the other books. The original bookshelf definition uses CAI.CAINB31O as the prefix in those data set names. To use BookManager Read on z/OS or OS/390, you must change the references to those data sets by replacing CAI.CAINB31O with the prefix used to unload the files.

## Downloading Files for Use on a PC

Perform the actions in this section only if you want to download the files to a PC. To use the PC versions of BookManager READ (use at least version 2.4 of IBM Softcopy Reader), be sure that the book files have been downloaded with a file extension of .BOO. The bookshelf must have an extension of .BKS and the index should have an extension of .BKI.

The table below describes the sequential files that are provided on the product tape and lists the commands you can issue to transfer them to a PC.

OS/390 Name/Description	OS/390 Format	File Transfer Command on PC
prefix.CAINB31O.BKSHELF Bookshelf definition	Variable block ASCII CRLF	RECEIVE <dir>\CAINB31O.BKS 'prefix.CAINB31O.BKSHELF'
prefix.CAINB31O.BKINDEX Search index	Fixed block LRECL 4096 Binary	RECEIVE <dir>\CAINB31O.BKI 'prefix.CAINB31O.BKINDEX'
prefix.CAINB31O.INB31OG0 Getting Started	Fixed block LRECL 4096 Binary	RECEIVE <dir>\INB31OG0.BOO 'prefix.CAINB31O.INB31OG0'
prefix.CAINB31O.INB31OU0 User Guide	Fixed block LRECL 4096 Binary	RECEIVE <dir>\INB31OU0.BOO 'prefix.CAINB31O.INB31OU0'
prefix.CAINB31O.INB31OQ0 Quick Reference	Fixed block LRECL 4096 Binary	RECEIVE <dir>\INB31OQ0.BOO 'prefix.CAINB31O.INB31OQ0'
prefix.CAINB31O.INB31OR0 Release Summary	Fixed block LRECL 4096 Binary	RECEIVE <dir>\INB31OR0.BOO 'prefix.CAINB31O.INB31OR0'

## Step 6b. Unloading the Acrobat PDF File from the Product Cartridge

Updates to the Advantage CA-InterTest Batch documentation are also provided in Adobe Acrobat PDF format on the Advantage CA-InterTest Batch installation and maintenance cartridge in file 16. The PDF file can be unloaded using the SAMPJCL member MR31PDF. Edit the JCL to conform to your installation standards and submit the job.

Download this file to a platform that supports Adobe Acrobat, such as a PC, with an extension of .TGZ. Decompress the file using one of the following methods:

- For Microsoft Windows NT/95/98 systems, use a tool that supports .TGZ files, such as WinZip 7.0 or above
- For UNIX systems, use either `gtar` or `gzip` and `tar`.

Start Adobe Acrobat to view the PDF files.

### Printing the Quick Reference Card

The *Quick Reference* card should be printed from the PDF on one sheet of legal sized paper to ensure a usable format.

Follow these instructions to correctly print the card:

1. From Acrobat Reader, press Ctrl+P (or select File, Print).
2. Select a printer that has duplexing capability.
3. Click Properties. Choose the following settings:
  - Paper Size: Legal (8.5 x 14)
  - Paper Source: Select the appropriate tray where legal paper is loaded or select Manual Feed to manually feed in a sheet of legal paper.
  - Orientation: Portrait
  - Duplex Printing: Short Side

After printing, fold the paper in half two times to achieve a quarter-fold reference card.

## Step 7. Allocate Target and Distribution Libraries

Member MR31ALC allocates all the Target and Distribution libraries required by Advantage CA-InterTest Batch during installation and maintenance. However, many Computer Associates products have common components and libraries that may have been already installed. Therefore, carefully analyze the installation steps to ensure that completed steps are not repeated.

Edit the JCL to conform to your installation standards and the previously completed worksheet. However, never change any of the ddnames or the last qualifier of the data set names. The parameters you specify in this JCL are:

DLIBVOL=,	The VOLSER of the Distribution Libraries.
TGTVOL=,	The VOLSER of the Target Libraries.
PREFIX='CAI.',	The product high-level qualifier. The default is CAI.
PERMDA=SYSDA	The generic UNIT for permanent DASD. The default is SYSDA.

All space allocations supplied in this member are in blocks to allow for compatibility between DASD types. The allocations given are the minimum required for installing Advantage CA-InterTest Batch. You may want to adjust these for your installation device types and to allow enough free space for maintenance (the more free space you allocate the less often they are compressed during maintenance). For common libraries already present, be sure there is sufficient space for Advantage CA-InterTest Batch.

Submit member MR31ALC and review the output.

## Step 8. Define SMP/E Libraries and ZONES

### Step 8a. Allocate Private SMP/E Libraries (Optional)

If SMP data sets have been created already during the installation of another Computer Associates product, then skip to step 8b.

Member CAINITE5 can be used to allocate and initialize a set of private SMP data sets for all Computer Associates products. You should keep Computer Associates products as entities distinct from other SMP data sets. These members also set up CAI global, target, and distribution zones for Computer Associates products.

The parameters you specify in CAINITE5 are:

VOLSER=,	The SMP library volume.
CAI='CAI.',	The high-level index for SMP. The default is CAI.
SYSOUT='*'	The SYSOUT class. The default is *.
PERMDA=SYSDA,	The generic permanent UNIT for SMP. The default is SYSDA.
WORK=SYSDA	The generic work UNIT for TEMP. The default is SYSDA.

Manual changes are required in this job stream to update SYSIN portions beyond the control log procedure's substitutional parameters. Both the IDCAMS and the ULCIN must be manually changed to reflect your chosen CSI name and volume. See the comments in the input overrides at the bottom of the job stream.

Edit the appropriate member and modify the JCL as indicated in the comments.

## Step 8b. Define Additional Product Target and Distribution Zones

Member MR31ZONE defines additional SMP Target and Distribution Zones and adds a DDDEF that is required for Advantage CA-InterTest Batch. Edit the JCL to conform to your installation standards and the previously edited worksheet.

Manual changes are required in this job stream to update SYSIN portions beyond the control of the procedure's substitutional parameters. The UCLIN must be manually changed to reflect your chosen CSI name, your SMPLTS data set name and your LE/370 linkedit data set name. See the comments in the input overrides at the bottom of the job stream.

## Step 9. Customize the SMP Procedure

Member MR31SMPE contains the model JCL procedure for using SMP/E. The procedure name is CAIMR31. Customize the SMP/E procedure as follows:

1. Create a new JCL procedure named CAIMR31 by copying the contents of MR31SMPE.
2. Modify member CAIMR31 to conform to your installation standards and the previously completed worksheet. This procedure can be placed into a system or user procedure library or executed in stream.

## Step 10. Receive the Product

Member MR31REC RECEIVES all the components (functional sysmods) of Advantage CA-InterTest Batch. It also unloads the required files that are shipped outside of SMP's control.

Edit the JCL to conform to your installation standards and the previously completed worksheet.

Two common components are provided on the tape. The first, CZ27000, is the Unicenter CA-JCLCheck common component. If you already have this sysmod applied, you do not need to apply it off of this tape. The second, CVH3100, is the Debugging Tools common component. If you already have this sysmod applied, you do not need to apply it off of this tape.

If other Computer Associates products have been installed, some of these functions may already have been received. If this is the case, SMP may re-RECEIVE the sysmods. Therefore, you should be aware that not all sysmods received later APPLY and ACCEPT. Failing sysmods should be removed and the job resubmitted.

Submit the job and review the output to verify that the RECEIVE processing ran successfully. If RECEIVE completed with an SMP return code other than zero, review the output, correct the problem, and resubmit.

## Step 11. Customization

### Step 11a. Define Excluded Programs

PPOPTION member CAMRXMOD defines the list of programs that will not be automatically monitored when wildcarding is used on the Monitor Control panel. This member was downloaded during the previous step. Edit the list to include any programs that you do not want to be automatically monitored. It is suggested that the supplied default list is left intact. If the need arises where you need to monitor a program whose name is present in this list, the program can still be monitored by explicitly adding the name of the program to the monitor control panel.

### Step 11b. Batch Link User Exit

When a selection is made off of the Batch Link Selection panel, a user exit, CAMRUSR1, is called. This allows for security checks to be performed to verify that the user that selected the job has the authority to monitor its execution. See the PPOPTION member CAMRUSR1 for more information on coding this user exit. The default member shipped will allow all users to select any available Batch Link job for debugging.

## Step 12. Apply the Product

Member MR31APP APPLYS all the components (functional sysmods) of Advantage CA-InterTest Batch to the target libraries. SMP does not require the Distribution libraries to be allocated during APPLY processing.

Edit the JCL to conform to your installation standards and the previously completed worksheet.

Uncomment and change any parameters that you wish to override.

If other Computer Associates products have been installed, some of these functions may have already been applied. If this is the case, a return code of 12 occurs. Remove any failing sysmods and resubmit the job. Optionally, you can specify REDO on the apply command statement to re-apply the sysmod.

Submit the job and review the output to verify that the APPLY processing ran successfully.

If APPLY completed with an SMP return code greater than 4, review the output, correct the problem, and resubmit.

## Step 13. Activate the Batch Link Feature

Activate the Batch Link Feature. If CAI.CAILIB is an APF Authorized library, you are ready to activate the Advantage CA-InterTest Batch Link Feature. Otherwise, use MR31RTL to copy the necessary activation modules to an APF authorized library of your choice. To activate the Batch Link feature, execute CAIRIM using the CAS9 procedure, which you modified in Step 1. Be sure that the modules to be refreshed are available to CAIRIM through STEPLIB or LINKLIST.

## Step 14. Allocate Additional Files

### Step 14a. Allocate PROFLIB

Member MR31PROF allocates the PROFLIBs for Advantage CA-InterTest Batch.

You must allocate a profile data set before executing Advantage CA-InterTest Batch. Member MR31PROF in the SAMPJCL library contains sample allocation JCL. 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 may access Advantage CA-InterTest Batch. There are three Advantage CA-InterTest Batch 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:

TGTVOL=,	VOLSER of the PROFLIB.
PREFIX='USER.',	The USERID high-level qualifier. The default is USER.
PERMDA=SYSDA	The generic UNIT for permanent DASD. The default is SYSDA.

**Note:** During execution, each Advantage CA-InterTest Batch user requires WRITE access to the PROFILE library only. All other required Advantage CA-InterTest Batch files can be secured with READ access only. Be sure to give similar attention to the optional files used by Advantage CA-InterTest Batch (for example, the session log).

## Step 14b. Allocate PROTSYM

Member VH31PROT creates the PROTSYM VSAM file, which contains the symbolic information and source listings for your application programs. Edit the JCL to conform to your installation standards and the previously completed worksheet. See the appendix "[Symbolic File](#)," for details.

Note the following items when creating the Symbolic File:

- Do not specify a secondary space allocation.
- The VH31PROT 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 appendix "[Symbolic File Maintenance](#)" explains how to delete members.
- The Symbolic file must not reside in the LSR pool.

### Concurrently Updating the PROTSYM File

The MVS RESERVE and DEQ macros are used by Advantage CA-InterTest Batch to allow sharing of the PROTSYM file between regions/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.

## Step 14c. Allocate ALIBs (Optional)

An ALIB is a data set used by Advantage CA-InterTest Batch to save allocation information. Advantage CA-InterTest Batch users can have any number of ALIBs at their disposal, but they must be created manually.

Member MR31ALIB creates and catalogs an ALIB. The parameters you need for this JCL are:

TGTVOL=,	The VOLSER of the target data sets.
PREFIX='USER.',	The USERID high-level qualifier. The default is USER.
PERMDA=SYSDA	The generic UNIT for permanent DASD.

## Step 14d. Allocate INCLIBs (Optional)

If you intend to use the Advantage CA-InterTest Batch INCLUDE command, you must allocate the Advantage CA-InterTest Batch INCLUDE library. Member MR31INCL allocates this library.

The parameters you need for this JCL are:

TGTVOL=,	VOLSER of the target data set.
PREFIX='USER.',	The SITE/USERID high-level qualifier. The default is USER.
PERMDA=SYSDA	The generic UNIT for permanent DASD. The default is SYSDA.

---

## Step 15. Customize Options

Member INTISPF in the PPOPTION library is used to customize the options that are passed to Advantage CA-InterTest Batch when it is invoked under ISPF.

The INTISPF member parameters are:

ISPF	Required in order to run Advantage CA-InterTest Batch 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. n may be an integer between 1 and 32767. The default is 1000.
EXTSTOR(nM   K)	<p>The extended storage parameter allows the user to set the amount of extended storage that may be accessed by Advantage CA-InterTest Batch under the MVS/XA operating system. This storage is used when formatting data above the 16-megabyte line.</p> <p>The default amount is 8M. The amount may be specified in Megabytes (nM) or Kilobytes (nK), and may be any integer from 1 to 100. It is not recommended to specify an amount greater than 20M or machine performance may be degraded.</p>
LINKPARAM(n)	Specifies the maximum number of linkage parameters that may be set by the installation. The default is 32. n may be 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 may be any integer between 1 and 4096. This might be helpful to TCAM users with buffering problems.
NOSTAX	A required Advantage CA-InterTest Batch startup parameter.
NOESTAE	A required Advantage CA-InterTest Batch startup parameter.
NOSTAE	A required Advantage CA-InterTest Batch startup parameter.
NOESTAI	A required Advantage CA-InterTest Batch startup parameter.
NOJSCB	A required Advantage CA-InterTest Batch startup parameter.

## Step 16. Customize CLISTS

A number of CLISTS provided in the CAICLIB library may need to be customized in order to run Advantage CA-InterTest Batch. These CLISTS are:

CAMRDRVR	MR31CIMS	MR31CDB2	MR31KEYS
MR31CLST	MR31CBTS	CAMRCMD	MR31STRT

### 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 wish 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.

## MR31CLST

Member MR31CLST is a sample CLIST showing necessary file allocations for Advantage CA-InterTest Batch. The following modifications should be made:

- 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.
- Choose per-site or per-user PROFLIB. In the sample CLIST, comment out the first statement after */\*Note3\*/* for per-user, or comment out the second statement after */\*Note3\*/* for per-site. Note that you may have to tailor these statements to meet your particular installation requirements.
- Under */\*Note4\*/*, uncomment the optional files you are using with Advantage CA-InterTest Batch. The optional files are:

**INT1CLIB** Defines the partitioned data set that contains CA-InterTest Batch commands to be used by the INCLUDE command. If it is not allocated, the INCLUDE command is disabled. For more information on the INCLUDE command, see the *User Guide*.

The default is CAI.PPOPTION, which includes the members needed for one of the demos (these members need to be copied if you define another name for INT1CLIB and you wish to run one of the demos).

**INT1CLOG** Defines a sequential file used for the Advantage CA-InterTest Batch session log. If it is not allocated, logging is disabled. For more information on the session log, see the *User Guide*.

**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 on the reports, see the HIST and XSUM Report Command descriptions in the *User Guide*.

## MR31CIMS

MR31CIMS is a sample CLIST showing the necessary file allocations for Advantage CA-InterTest Batch testing of IMS/DB or DL/I programs. If you are licensed for the IMS option and wish 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 Advantage CA-InterTest Batch and IMS data sets.
- Under */\*Note2\*/*, uncomment those libraries that are not allocated in your TSO LOGON procedure or in another CLIST. Note that there are two sections marked */\*Note2\*/* in this CLIST.
- Choose per-site or per-user PROFLIB. In the sample CLIST, comment out the first statement after */\*Note3\*/* for per-user, or comment out the second statement after */\*Note3\*/* for per-site. Note that you may have to tailor these statements to meet your particular installation requirements.
- Under */\*Note4\*/*, uncomment the optional files you are using with Advantage CA-InterTest Batch. See the optional files listed under MR31CLST above 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.

## MR31CBTS

MR31CBTS is a sample CLIST showing the necessary file allocations for Advantage CA-InterTest Batch testing of IMS/DC programs under BTS. If you are licensed for the IMS option and wish 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 Advantage CA-InterTest Batch, IMS and BTS data sets.
- Under `/*Note2*/`, uncomment those libraries that are not allocated in your TSO LOGON procedure or in another CLIST. Note that there are two sections marked `/*Note2*/` in this CLIST.
- Choose per-site or per-user PROFLIB. In the sample CLIST, comment out the first statement after `/*Note3*/` for per-user, or comment out the second statement after `/*Note3*/` for per-site. Note that you may have to tailor these statements to meet your particular installation requirements.
- For the statements under `/*Note4*/`, uncomment the optional files you are using with Advantage CA-InterTest Batch. See the optional files listed under MR31CLST 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.

## MR31CDB2

MR31CDB2 is a sample CLIST showing necessary file allocations for Advantage CA-InterTest Batch testing of DB2 programs. If you wish 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. You may use ISPLLIB for the COBOL runtime library if it is different from the default (linklist). Note that there are two sections marked `/*Note2*/` in this CLIST.
- Choose per-site or per-user PROFLIB. In the sample CLIST, comment out the first statement after `/*Note3*/` for per-user, or comment out the second statement after `/*Note3*/` for per-site. Note that you may have to tailor these statements to meet your particular installation requirements.
- Under `/*Note4*/`, uncomment the optional files you are using with Advantage CA-InterTest Batch. See the optional files listed under MR31CLST for a description of the optional files.

This CLIST is only required if you will be debugging your DB2 applications in foreground.

## MR31STRT

MR31STRT 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 MR31KEYS.

## MR31KEYS

MR31KEYS optionally is called from the CLIST MR31STRT 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. See the *User Guide* for more information on the `/clist` command.

## Step 17. Customize Compile PROCs

Several sample JCL members are provided in library CAIPROC, which can be modified and used as Advantage CA-InterTest Batch compile/assemble PROCs. Which member you use depends upon whether you are using COBOL/VS, COBOL II and above, 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.

The user needs to supply the following parameters:

PROT	Specifies the data set name of the PROTSYM file.
COPT	Overrides the compiler options.

Several additional items must be specified in the JCL. Information that must be modified or supplied includes:

- COBOL compiler library
- COBOL copy library
- COBOL source library
- LKED SYSLIB

## COB2INT

COB2INT is a sample JCL stream showing a COBOL II, COBOL/370, 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.

The user needs to supply the following parameters:

PROTSYM	Specifies the data set name of the PROTSYM file.
COPT	Overrides the compiler options.

Several additional items must be specified in the JCL. Information that must be modified or supplied includes:

- COBOL compiler library
- COBOL copy library
- COBOL source library
- LKED SYSLIB

## PLIINT

PLIINT is a sample JCL stream showing a PL/I compile. This procedure will update the symbolic information in the PROTSYM file.

The user needs to supply the following parameters:

PROTSYM      Specifies the data set name of the PROTSYM file.

POPT            Overrides the PL/I compile options.

Several additional items must be specified in the JCL. Information that must be modified or supplied includes:

- PL/I source library
- PL/I copy library
- LKED SYSLIB

## ASMINT

ASMINT is a sample JCL stream showing an assemble. This assemble procedure will update the symbolic information in the PROTSYM file.

The user needs to supply the following parameters:

PROTSYM      Specifies the data set name of the PROTSYM file.

AOPT            Overrides the assembler options.

Several additional items must be specified in the JCL. Information that must be modified or supplied includes:

- Assembler library
- Assembler macro library
- LKED SYSLIB

## Step 18. Adding to ISPF Main Panel (Optional)

Member MR31ISPF in library PPOPTION is used to place Advantage CA-InterTest Batch on the ISPF main panel as an option called INTB. Use this model only if you want this option to appear in your ISPF main menu.

## Step 19. Add ISPF Libraries to TSO LOGON Procedure (Optional)

This step is needed only if LIBDEF statements were not uncommented in the Advantage CA-InterTest Batch CLISTs (see [Step 16. Customize CLISTs](#)).

The Advantage CA-InterTest Batch ISPF panel library must be concatenated to the IBM ISPF panel library, the Advantage CA-InterTest Batch message library must be concatenated to the IBM ISPF message library and the Advantage CA-InterTest Batch CLIST library must be concatenated to the IBM CLIST library.

Member MR31TSOL in the PPOPTION library adds Advantage CA-InterTest Batch ISPF libraries to your TSO LOGON procedure. A COBOL runtime library may be added if it is different from the site default.

## Step 20. Customize Advantage CA-Roscoe (Optional)

This step applies only to Advantage CA-Roscoe users.

Three sample members are provided in the PPOPTION library for customizing Advantage CA-Roscoe for Advantage CA-InterTest Batch: MR31ETSO, MR31EPL and INTROS.

### MR31ETSO

MR31ETSO is a sample JCL to add to your Advantage CA-Roscoe JCL. It should be concatenated to your Advantage CA-Roscoe ETSOLIB. The COBOL runtime library may be added if it is different from the site default.

### MR31EPL

MR31EPL contains sample definitions to add to your Advantage CA-Roscoe Eligible Program List for Advantage CA-InterTest Batch. Incorporate it into your Advantage CA-Roscoe EPL.

## INTROS

Member INTROS in the PPOPTION library is used to customize the options that are passed to Advantage CA-InterTest Batch when it is invoked under Advantage CA-Roscoe.

The INTROS member parameters are:

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. n may be an integer between 1 and 32767. The default is 1000.
EXTSTOR(nM   K)	<p>The extended storage parameter allows the user to set the amount of extended storage that may be accessed by Advantage CA-InterTest Batch under the MVS/XA operating system. This storage is used when formatting storage above the 16-megabyte line.</p> <p>The default is 8M. The amount may be specified in Megabytes (nM) or Kilobytes (nK), and may be any integer from 1 to 100. It is not recommended to specify an amount greater than 20M or machine performance may be degraded.</p>
LINKPARM(n)	Specifies the maximum number of linkage parameters that may be set by the installation. The default is 32. n may be 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 the executable statements when using the SLOWFAST command. The default is one quarter of a second (n=25). The value may be any integer between 1 and 4096. This might be helpful to TCAM users with buffering problems.
NOSTAX	A required Advantage CA-InterTest Batch startup parameter.
NOESTAE	A required Advantage CA-InterTest Batch startup parameter.
NOSTAE	A required Advantage CA-InterTest Batch startup parameter.
NOESTAI	A required Advantage CA-InterTest Batch startup parameter.
NOJSCB	A required Advantage CA-InterTest Batch startup parameter.

Other considerations for Advantage CA-Roscoe users:

- Concatenate the COBOL or PL/I runtime library if it is not in linklist and it is required to execute your application under Advantage CA-Roscoe.
- Increase the Advantage CA-Roscoe ETSALLOC parameter, if necessary.
- The CAI.RPFLIB library must be imported to an Advantage CA-Roscoe RPF library (see [Step 10. Receive the Product](#)).

## Step 21. Run Demo (Optional)

Running the demo ensures that the product has been installed correctly and provides an introduction to using Advantage CA-InterTest Batch. Using one of the PROCs customized in Step 17, compile/assemble and link one of the demo programs provided in CAI.CAISRC. See the chart below for determining which demo program is right for your shop. See the appendix, “Basic Foreground Demo Session” in the *User Guide* for detailed information on running the demo program.

CAMRASM      For Assembler programmers.

CAMRPLI      For PL/I programmers.

CAMRCOB      For OS/VS COBOL programmers

CAMRCOB2     For COBOL II programmers.

**Note:** The above programs use ISPF. Programmers on Advantage CA-Roscoe systems can use CAMRDMR and CAMRDMR2 for demonstration purposes.

## Step 22. Review COBOL RTL Considerations

When mixing COBOL/370, COBOL for MVS, and COBOL for OS/390 programs with COBOL II and OS/VS COBOL programs, you must be sure that your LE/370 RTL is ahead of your other COBOL RTLs in any concatenation. If mixing VS COBOL II with COBOL, be sure that your COBOL II RTL appears ahead of your COBOL RTL.

## Step 23. Review and Customize for BTS

Member INTBTSX is provided for ISPF users with BTS extended support. (This member is provided for clients who are licensed for the IMS option and wish to debug BTS applications in foreground.) Parameters relating to INTBTSX are:

ISPF	Required in order to run Advantage CA-InterTest Batch under the ISPF dialog manager.
SYMLDYN	This option 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. n may be an integer between 1 and 32767. The default is 1000.

EXTSTOR(nM   K)	The extended storage parameter allows the user to set the amount of extended storage that may be accessed by Advantage CA-InterTest Batch under the MVS/XA operating system. This storage is used when formatting storage above the 16-megabyte line. The default amount is 8M. The amount may be specified in Megabytes (nM) or Kilobytes (nK), and may be any integer from 1 to 100. It is not recommended to specify an amount greater than 20M or machine performance may be degraded.
LINKPARM(n)	Specifies the maximum number of linkage parameters that may be set by the installation. The default is 32. <i>n</i> may be 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 the executable statements when using the SLOWFAST command. The default is one quarter of a second (n=25). The value may be any integer between 1 and 4096. This might be helpful to TCAM users with buffering problems.
BTSNAME(*****)	The BTSTSOST user-defined name.
BTSEXT	Intercepted PA2 support.
BTSSYSO	BTS SYSOUT support.
NOSTAX	A required Advantage CA-InterTest Batch startup parameter.
NOESTAE	A required Advantage CA-InterTest Batch startup parameter.
NOESTAI	A required Advantage CA-InterTest Batch startup parameter.
NOJSCB	A required Advantage CA-InterTest Batch startup parameter.

## Screen Handling Solution for the Message BTS0067I

Advantage CA-InterTest Batch 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 above function, or your site has experienced any problems with PA2, parameter BTSEXT in member INTBTSX (see the previous parameter list) signals Advantage CA-InterTest Batch to use extended BTS support. The extended support causes BTS to ignore the PA2 key.

This modification has the following side effects on BTS:

1. 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 above.
2. The change also replaces the PF key simulation capability of BTS.  
This only affects users of Advantage CA-InterTest Batch and BTS using terminals with NO program function (PF) keys.

If either of the above causes a problem at your site, contact Advantage CA-InterTest Batch Customer Support for assistance.

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 in conjunction with another full-screen product such as Advantage CA-InterTest Batch.

## Applications Using Message Switching and DISPLAY or Figure Verbs

Applications that perform message switching and use DISPLAY and/or Figure verbs should use the BTSSYSO option.

Parameter BTSSYSO in member INTBTSX (see the previous parameter list) signals Advantage CA-InterTest Batch to use extended BTS SYSOUT support.

The BTSSYSO parameter is recommended if your site has programs that use DISPLAY and/or Figure in the BTS environment.

## Step 24. Accept the Product

Member MR31ACC ACCEPTs all the components (functional sysmods) of Advantage CA-InterTest Batch to the Distribution libraries.

Edit the JCL to conform to your installation standards and the previously completed worksheet.

If other Computer Associates products have been installed some of these functions may have already been accepted. If this is the case, a return code of 12 occurs. Remove the failing sysmods and resubmit the job. SMP/E users have the option of specifying the operand REDO on the ACCEPT command statement and should expect an SMP return code of 8, which, in this case, is permissible.

Submit the job and review the output to verify that the ACCEPT processing ran successfully. If ACCEPT completed with an SMP return code greater than 4, view the output, correct the problem, and resubmit the job.

You should ACCEPT the Advantage CA-InterTest Batch product immediately after you have successfully run the APPLY. This ACCEPT step provides a backup of the base product on the Distribution libraries. This backup is necessary for future SMP RESTOREs.

## Step 25. Save Materials and Output

Save all installation materials and installation-generated output so that it is available for your own reference and when contacting customer support.

# Applying Maintenance

---

## Overview

Advantage CA-InterTest Batch maintenance is delivered in SMP/E format. You should use SMP/E to apply all maintenance modifications.

**Caution:** If you attempt to apply maintenance without using SMP/E, the integrity of your Advantage CA-InterTest Batch libraries is at risk and the maintenance tracking capabilities of SMP/E are lost.

## SMP/E Libraries

SMP/E uses two sets of libraries:

- Target libraries are the product execution or runtime libraries.
- Distribution libraries are the product backup libraries.

When a product is installed by SMP/E, the target and distribution libraries are populated. The target libraries are populated during the SMP/E APPLY process and the distribution libraries are populated during the SMP/E ACCEPT process. At this point, the target and the distribution libraries contain identical versions of the product modules.

Maintenance through SMP/E is applied to the runtime system (the target libraries) first. After the maintenance has been running on the system for a period of time or when it appears that there are no problems with the maintenance, it is applied to the backup libraries (the distribution libraries).

When maintenance is installed on the running libraries, it is said to be applied. When maintenance is installed on the backup libraries, it is said to be accepted.

## APPLY and ACCEPT Commands

APPLY and ACCEPT are the names of two SMP/E commands. The APPLY process updates the target libraries and the ACCEPT process updates the distribution libraries.

The cycle of a product under SMP/E generally follows this schedule:

- APPLY and ACCEPT the base product installation material, creating target and distribution libraries.
- At this point, the target and distribution libraries are at the same maintenance level. Later, you APPLY maintenance in the form of an APAR or a PTF, bringing the target libraries to a new maintenance level.
- The target and distribution libraries are now at different maintenance levels. If there is a problem with the maintenance applied to your target libraries, the maintenance may be removed by the SMP/E RESTORE command.
- The RESTORE process replaces the modules in the target library that were affected by the maintenance with backup modules from the distribution libraries.
- After verifying that the maintenance does not cause any new problems on your runtime system, install the maintenance into your distribution libraries with the SMP/E ACCEPT process. This step brings your target and distribution libraries in sync at the same maintenance level.

Once you accept the maintenance into your distribution libraries, there is no SMP/E process to remove it.

## Maintenance Delivery

Maintenance to your Advantage CA-InterTest Batch system is packaged and delivered as SMP/E USERMODs, APARs, and PTFs. The various types of maintenance delivery are discussed below.

### Informational APAR or PIB

The informational APAR or Product Information Bulletin (PIB) is official information relative to a product.

- PIBs are delivered by hard copy, CA's SupportConnect (<http://www.ca.com/supportconnect>), and on maintenance tapes.
- The naming convention for PIBs is *QIxxxxx*, where:
  - *QI* identifies an informational APAR
  - *xxxxx* is an identification number

## Program Temporary Fixes

A Program Temporary Fix or PTF is an official replacement module or modules that fixes one or several problems. A PTF is usually an accumulation of APARs.

- PTFs are installed as PTF SYSMODs, which you must ACCEPT before installing subsequent maintenance
- PTFs are always delivered on a periodic maintenance tape
- PTFs are named *Zpcrlxx*, where:
  - *pc* is the component code,
  - *rl* is the release, and
  - *xx* is a unique identifier.

## PTF Maintenance Tape Installation

A combined installation/maintenance tape is distributed periodically on a standard IBM cartridge that can be processed by SMP/E. It contains the newly published official PTF SYSMODs for Advantage CA-InterTest Batch. This cartridge is to be used only where SMP/E controls installation and maintenance of CA solutions. This tape is in the new combined format, which means that it can be used either to install Advantage CA-InterTest Batch at the genlevel specified on the external label, or to apply maintenance to bring an existing Advantage CA-InterTest Batch up to the genlevel specified on the external label.

A partitioned data set containing all necessary sample JCL is provided in the Load Sample JCL Library step. The sample JCL data set contains all JCL necessary for installation and maintenance. The maintenance-specific members are appended with an M. See the external label of the tape for the current volume serial number.

## Overview of Maintenance Steps

**Caution:** Before APPLYing any new maintenance, the base product and all prior maintenance to Advantage CA-InterTest Batch must be ACCEPTed.

The following list summarizes the steps involved in the Advantage CA-InterTest Batch maintenance process. Review this list before attempting to perform maintenance on Advantage CA-InterTest Batch.

Step	Description
Step 1.	Review and follow the cover letter instructions.
Step 2.	Load the sample JCL from the tape.
Step 3.	Unload the documentation.
Step 4.	Customize the SMP/E procedure.
Step 5.	Receive Advantage CA-InterTest Batch maintenance.
Step 6.	Run APPLY CHECK processing on Advantage CA-InterTest Batch maintenance.
Step 7.	Restore any applicable SYSMODs.
Step 8.	Apply Advantage CA-InterTest Batch maintenance.
Step 9.	Reapply any applicable SYSMODs.
Step 10.	Refresh Batch Link feature.
Step 11.	Execute Advantage CA-InterTest Batch Demo program to verify successful maintenance.
Step 12.	Accept Advantage CA-InterTest Batch maintenance.
Step 13.	Save all materials and output.

## Step 1. Review and Follow the Cover Letter Instructions

Before starting the maintenance steps below, you **MUST** first review and follow the cover letter instructions. This letter details any additional steps that may be required by your site.

Additionally, before starting this maintenance procedure, you should:

- Check the packlist for Advantage CA-InterTest Batch against the package contents.
- Review all PTFs (Program Temporary Fixes) and PIBs (Product Information Bulletins) provided.

***Important!** If any items are missing, or if you have any questions, call your local Computer Associates Technical Support center before attempting the maintenance procedure.*

## Step 2. Load the Sample JCL from the Tape

Use the following JCL as a model to load the sample JCL library to DASD:

```
//LOAD      EXEC PGM=IEBCOPY
//SYSPRINT  DD SYSOUT=A
//SYSUT1    DD DISP=(OLD,KEEP),
//          DSN=CAI.SAMPJCL,
//          UNIT=TAPE,           <=== generic 3480 cartridge
//          VOL=SER=FCyymm
//          LABEL=(9,SL),
//          DCB=DEN=4
//SYSUT2    DD DISP=(NEW,CATLG,DELETE),
//          DSN=CAI.INTBATCH.SAMPJCL, <=== your DSN
//          UNIT=SYSDA,          <=== your generic DASD
//          VOL=SER=XXXXXX      <=== permanent DASD volser
//          SPACE=(3120,(40,20,10)), <=== minimum space required
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSUT3    DD UNIT=SYSDA,
//          SPACE=(TRK,(5,5))
//SYSUT4    DD UNIT=SYSDA,
//          SPACE=(TRK,(5,5))
//SYSIN     DD DUMMY
```

Once this job has ended, your library contains all of the JCL needed to complete the maintenance for Advantage CA-InterTest Batch.

## Step 3. Unload the Documentation

See [Step 6. Unloading the Documentation](#), in the chapter "Installing Advantage CA-InterTest Batch," for an explanation of how to unload the documentation set.

## Step 4. Customize the SMP/E Procedure

During the installation of Advantage CA-InterTest Batch, an SMP/E procedure was selected and customized. If the SMP/E procedure used during the installation of Advantage CA-InterTest Batch is available, this step is not required.

If that procedure is not available, see [Step 9. Customize the SMP Procedure](#) in the chapter "Installing Advantage CA-InterTest Batch" for instructions on re-creating this procedure.

**Note:** This procedure is used in subsequent steps.

## Step 5. Receive Maintenance

Maintenance JCL member MR31RECM receives all of the PTFs corresponding to Advantage CA-InterTest Batch maintenance. Edit member MR31RECM to conform to your installation standards. This step refreshes your CAI.PNLLIB library. This step is required.

Submit the job and verify RECEIVE processing was successful. If the SMP/E RECEIVE completed with a return code greater than 4, review the output, correct the problem, and resubmit.

## Step 6. APPLY CHECK Maintenance

Maintenance JCL member MR31APCM APPLY CHECKs all PTFs corresponding to the components specified within the SMPCNTL DD statement. This step identifies any USERMODs and APARs that prevent PTF application, and any PTFs already applied.

Edit member MR31APCM to conform to your installation standards. Delete any DD statements within the SMPCNTL DD statement that correspond to Advantage CA-InterTest Batch components not currently installed.

Submit the job and verify APPLY CHECK processing was successful. If the SMP/E APPLY CHECK completed with a return code greater than 4, review the output, correct the problem, and resubmit.

**Note:** SMP/E APPLY CHECK processing performs preliminary validation on SYSMODs individually. Carefully review the SMP/E generated reports, noting any possible regression of SYSMODs. Computer Associates requires the removal of any SYSMOD preventing PTF application. To allow PTF application, perform SMP/E RESTORE processing on the SYSMODs identified during the SMP/E APPLY CHECK. The instructions to accomplish this are in Step 7.

## Step 7. Restore Any Applicable SYSMODs

Maintenance JCL member MR31RESM contains the control statements for an SMP/E RESTORE. This step restores SMP/E USERMODs and APARs identified by APPLY CHECK processing, to allow for PTF application. If you do not have any SYSMODs to RESTORE, you may continue to the next step.

Edit member MR31RESM to conform to your installation standards. Only the SYSMODs identified by APPLY CHECK processing need to be restored. Type these SYSMODs within the SMP\_CNTL DD statement.

After you complete editing, submit the job and verify that RESTORE processing ran successfully. If SMP/E RESTORE completed with a return code greater than 4, review the output, correct the problem, and resubmit.

The SMP/E APPLY CHECK should be executed again to verify that there are no additional SYSMODs that inhibit the application of maintenance.

## Step 8. Apply Maintenance

Maintenance JCL member MR31APPM APPLYS all PTFs corresponding to the components specified within the SMPCNTL DD statement.

Edit member MR31APPM to conform to your installation standards. Submit the job and verify that APPLY processing was successful. If the SMP/E APPLY completed with a return code greater than 4, review the output, correct the problem, and resubmit.

**Note:** If you are applying maintenance to the Unicenter CA-JCLCheck FMID, you must pass BYPASS (HOLDSYS) on the APPLY statement. Existing Unicenter CA-JCLCheck clients should not apply maintenance to this FMID with this product tape. Maintenance for this FMID should be applied through Unicenter CA-JCLCheck.

## Step 9. Reapply Any Applicable SYSMODs

Review the USERMODs and APARs restored by Step 7 (RESTORE applicable SYSMODs). If no SYSMODs were restored, skip to the next step.

SYSMODs identified by APPLY CHECK processing may be at a higher level than the PTFs contained on the Advantage CA-InterTest Batch maintenance tape.

Edit maintenance JCL member MR31APPM to conform to your installation standards. Submit the job and verify that APPLY processing ran successfully. If the SMP/E APPLY completed with a return code greater than 4, then review the output, correct the problem, and resubmit the job.

**Note:** SMP/E can handle only one update per element per APPLY select statement. In reapplying SYSMODs, it may be necessary to use multiple APPLY select statements.

## Step 10. Refresh Batch Link Feature

If CAI.CAILIB is an APF Authorized library, you are ready to activate the Advantage CA-InterTest Batch Link feature. Otherwise, use MR31RTL to copy the necessary activation modules to an APF authorized library of your choice. To refresh the Batch Link feature, execute CAIRIM using the CAS9 procedure that you modified during the installation. Be sure that the modules to be refreshed are available to CAIRIM through STEPLIB or LINKLIST.

## Step 11. Execute IVP to Validate Maintenance

Run the Advantage CA-InterTest Batch demo program to ensure that the product has been installed correctly. See the appendix, “Basuc Foreground Demo Session” in the *User Guide* for detailed information on running the demo program.

## Step 12. Accept Maintenance

After a reasonable period of testing the features of Advantage CA-InterTest Batch with the new maintenance level APPLYed, you should perform an SMP/E ACCEPT. This gives you a solid baseline to proceed from and makes it easier to APPLY/RESTORE any test USERMODs that you may be given by the Advantage CA-InterTest Batch support staff.

Maintenance JCL member MR31ACCM ACCEPTs all PTFs corresponding to the components specified within the SMPCNTL DD statement.

Edit member MR31ACCM to conform to your installation standards. Submit the job and verify ACCEPT processing was successful. If the SMP/E ACCEPT completed with a return code greater than 4, review the output, correct the problem, and resubmit.

## Step 13. Save All Materials and Output

Be sure to save all of your maintenance materials and all output from the maintenance process. This material is essential for future problem determination.

# Symbolic File

---

## About the Symbolic File

The CA common symbolic file (also known as the PROTSYM file) is used by various CA products to store symbolic information about your application programs. Information is typically stored into the symbolic file during a post-compile process, and is accessed later by one or more of your CA products.

The symbolic file can hold information about COBOL, PL/I or Assembler programs. Different CA products may provide different levels of support for these programming languages. However, your symbolic files may be shared by all of your CA products, if desired.

You can modify your compile and assemble procedures to include a post-processor step for loading your symbolic file. The post-processor step will not alter your program or load module in any way. It simply loads symbolic information from your program listing into your symbolic file.

Your symbolic file may be updated by as many concurrent jobs as necessary. Concurrent compiles or assemblies do not interfere with each other in updating the symbolic file.

## Using the Post-Processors with Multiple Product Releases

**WARNING!** Symbolic information stored using this version of the symbolic post-processors may not be downward compatible with earlier CA product releases.

- If you maintain multiple releases of a CA product and you require symbolic information for your application to be compatible with the older release, you should load your symbolic file using the post-processor provided with that release.
- If you require symbolic support for an application to be compatible with multiple CA products (such as Advantage™ CA-Realia® II Workbench Batch Option and Advantage CA-InterTest for CICS) you may need to maintain multiple symbolic files. Check the specific product documentation for special considerations of compatibility issues.

Information stored in the symbolic file using a post-processor from an earlier release will usually be upward compatible with newer product releases.

## COBOL Symbolic Support

The post-processor program that you use depends on your COBOL compiler. Use program IN25SYMC for OS/V5 COBOL, or IN25COB2 for all newer compilers. Sample JCL described later in this section explains which post-processor to use.

Here is how the symbolic post-processor works:

1. The symbolic post-processor operates **after** your compile step and collects information from the compiler's `//SYSPRINT DD` statement.
2. This temporary data set is passed to the post-processor step through the `//INPUT DD` statement.

3. The original listing from your compiler's //SYSPRINT DD is written to the //OUTPUT DD defined in the post-processor step. This DD should be defined in the same way as the //SYSPRINT DD was defined before the procedure was modified.
4. Messages from the post-processor are written to the //MESSAGE DD, typically a SYSOUT dataset to be printed.
5. The symbolic file is referenced by the //PROTSYM DD statement with DISP=SHR.
6. Old records in the PROTSYM file are replaced by new ones during each run of the symbolic post-processor.

See the appendix, "[Symbolic File Maintenance](#)" for instructions on removing old program data from the file.

## Notes for Post-Processing COBOL Programs

To use symbolic references in COBOL, you should also be aware of the following considerations:

- For CICS applications, the following compiler options must **not** be used: COUNT, ENDJOB, FLOW, DYNAM, STATE, SYMDMP, SYST, and TEST.
- At least one data item must be declared in Working Storage.
- If compiling a CICS program using the COBOL for MVS and VM compiler and above, the program must be defined to CICS as LE/370 and not COBOL or COBOL II. Failure to do so can cause unpredictable results.

## IBM COBOL Compiler Options

The following options must be used with the IBM COBOL compiler.

---

<b>COBOL</b>	<b>COBOL II, COBOL/370, IBM COBOL for MVS, IBM COBOL for OS/390, IBM COBOL for z/OS</b>	<b>Produces</b>
NONUM	NONUMBER	Suppress Compiler Generated Line Numbers
DMAP	MAP	Data Division Map
VERB	no option	Verb Names
SXREF	XREF	Cross-Reference of Data Names & Cross- Reference of Paragraph Names
CLIST or PMAP	OFFSET or LIST	Condensed Procedure Division Map  ASSEMBLER-like Procedure Division Map

---

---

## Advantage CA-Optimizer and Advantage CA-Optimizer/II Options

The options listed in the following table **must** be used with Advantage™ CA-Optimizer® and Advantage™ CA-Optimizer®/II, unless you bypassing optimization using the NOCOPT option.

If you are using the NOCOPT option, use the options for IBM COBOL compilers specified in the previous section. See the Advantage CA-Optimizer *User Guide* or Advantage CA-Optimizer/II *User Guide* for more information.

---

<b>Advantage CA-Optimizer</b>	<b>Advantage CA-Optimizer/II</b>	<b>Produces</b>
MLIST	MLIST or MOFFSET	Integrated Procedure Division Map
NONUM	NONUMBER	Suppress Compiler- Generated Line Numbers

---

DMAP	MAP or MMAP	Data Division Map
	INTERTST	Required for Advantage CA-InterTest for CICS and Advantage CA-InterTest Batch. Indicates that the program is being optimized for use with the debugger.
XREF	XREF	Cross-Reference of Data Names and Cross-Reference of Paragraph Names

## Post-Processor Options

Options are passed to the symbolic post-processor through a parameter statement in the CARDS file. The CARDS file can be an instream DD, a permanent file, or a temporary file generated by program IN25PARM prior to the post-processor step. In the example provided below, the CARDS file is generated by IN25PARM in the GENPARM step (indicated by ← 2).

- The only required field on the parameter statement is a program name field. For batch applications, this should be the PROGRAM-ID or CSECT name. For CICS applications, this should be the name in the CICS program definition or, for composite support, the monitor-name.
- If you pass options using a control card, the program name **must** start in column 1.

The following example saves symbolic references for the program named ORDEDIT using a control card:

```
COL = 1...5...θ  
      ORDEDIT
```

## Controlling Printed Output with the CUTPRINT Option

Since all COBOL source code and compiler information can be available online, you can request that the symbolic post-processor **not** print all or part of the listing produced by the compiler. To make this request, add one of the following keywords immediately after the program name on the parameter statement:

**,CUTPRINT=ALL** produces no COBOL listing

**,CUTPRINT=MAP** prints only the part that precedes the Data Division map

**,CUTPRINT=REF** prints only the parts that precede the data names Cross-Reference list

The following sample parameter statement saves symbolic information for the program ORDEDIT and prints only the source code portion of the compiler output.

```
ORDEDIT ,CUTPRINT=MAP
```

### Notes about the CUTPRINT option:

Specify the CUTPRINT parameter only when you do not want all or part of your listing printed. The entire listing will be printed if this parameter is missing from the parameter statement.

## Saving Your Listing for Display with the LISTER Option

You can also request that all or a portion of your source code be saved for online viewing. If saved, CA products will use the listing for symbolic displays, such as breakpoint displays from Advantage CA-InterTest for CICS.

To make this request you must add one of the following keywords to the parameter statement:

**,LISTER=ALL** saves the entire COBOL listing

**,LISTER=MAP** saves only the parts that precede the Data Division map (recommended for COBOL)

**,LISTER=REF** saves only the parts that precede the data names Cross-Reference list (recommended for COBOL II)

For example, to save the symbolic references for the program ORDEDIT, not print any portion of the listing, and save the source code portion of the program, specify the following:

```
ORDEDIT , CUTPRINT=ALL , LISTER=MAP
```

### Notes about the LISTER Option

- If the LISTER parameter is omitted, no listing is saved in the Symbolic File.
- You should use LISTER=MAP for COBOL and LISTER=REF for COBOL II for minimum overhead and disk space usage.
- If only the LISTER parameter is specified, it must follow the program name.

## Setting Data as Nonpurgable

You could also request that any saved symbolic data for this program be marked as nonpurgable. If a program's data is marked as nonpurgable, the data will not be removed from the Symbolic File when deleting programs using a purge interval batch run. However, the data could be deleted by program name. See the appendix, "[Symbolic File Maintenance](#)" for instructions on deleting data from the Symbolic File.

To mark data as nonpurgable, add the following keyword to the parameter statement as the last option:

**,NOPURGE**

In the following example, the parameter specification records the symbolic references for the program ORDEDIT, prints the listing, does not save any portion of the listing, and does not allow symbolic data to be removed from the symbolic file by a purge interval batch run:

**ORDEDIT,NOPURGE**

In the next example, the parameter specification records the symbolic references for the program ORDEDIT, does not print a copy of the listing, does not save any portion of the listing, and does not allow data to be removed from the Symbolic File by a purge interval batch run:

**ORDEDIT,CUTPRINT=ALL,NOPURGE**

In this example, the parameter specification records the symbolic references for the program ORDEDIT, does not print a copy of the listing, saves the source code portion of the listing for online display, and does not allow data to be removed from the Symbolic File by a purge interval batch run:

**ORDEDIT,CUTPRINT=ALL,LISTER=MAP,NOPURGE**

## COBOL JCL Examples

The following sections show JCL examples.

### COBOL/VS or Advantage CA-Optimizer Example

```

/** YOUR EXISTING CICS PREPROCESSOR JCL GOES HERE
/**      (IF DOING A COMMAND LEVEL COMPILE)

/** YOUR EXISTING COBOL OR CA-OPTIMIZER JCL GOES HERE WITH
/** THE FOLLOWING MODIFICATIONS:

For CA-Optimizer only:

//SYSPRINT DD DSN=&&INPUT,DISP=(MOD,PASS),UNIT=SYSDA,      ← 1
// SPACE=(TRK,(15,5)),                                     ← 1
// DCB=(DSORG=PS,LRECL=121,BLKSIZE=2420,RECFM=FBA)        ← 1

Otherwise:

//SYSPRINT DD DSN=&&INPUT,DISP=(,PASS),UNIT=SYSDA,          ← 1
// SPACE=(TRK,(15,5)),                                     ← 1
// DCB=(DSORG=PS,LRECL=121,BLKSIZE=2420,RECFM=FBA)        ← 1

For all symbolic users:

/** IF THE COMPILER GIVES A RETURN CODE NOT HIGHER THAN 4
/** RUN THE CA SYMBOLIC POST-PROCESSOR
/**
/** //STEPLIB AND THE //STEPCLIB MAY BE UNNECESSARY.
/**
/**GENPARG EXEC PGM=IN25PARG,PARM='PARAMETER STATEMENT'   ← 2
/**STEPLIB DD DSN=CAI.CAILIB,DISP=SHR                     ← 2
/**CARDS DD DSN=&&CARDS,DISP=(,PASS),UNIT=SYSDA,           ← 2
/** SPACE=(TRK,(1,1))
/**
/**POSTCOB EXEC PGM=IN25SYMC,COND=(4,LT,COBOL),REGION=512K ← 3
/**STEPLIB DD DSN=CAI.CAILIB,DISP=SHR                     ← 3
/**INPUT DD DSN=&&INPUT,DISP=(OLD,DELETE)                   ← 3
/**OUTPUT DD SYSOUT=A,                                     ← 3
/** DCB=(RECFM=FBA,LRECL=121,BLKSIZE=2420)                ← 3
/**MESSAGE DD SYSOUT=A                                     ← 3
/**PROTSYM DD DSN=PROTSYM.FILE,DISP=SHR                   ← 3
/**CARDS DD DSN=&&CARDS,DISP=(OLD,DELETE)                  ← 3
/**
/** IF THE COMPILER GIVES A RETURN CODE HIGHER THAN 4
/** THEN RUN AN IEBGENER TO PRINT THE COBOL LISTING
/**
/**PRINT EXEC PGM=IEBGENER,COND=(5,GT,COBOL)              ← 4
/**SYSPRINT DD DUMMY                                       ← 4
/**SYSIN DD DUMMY                                          ← 4

```

```
//SYSUT1 DD DSN=&&INPUT,DISP=(OLD,DELETE) ← 4  
//SYSUT2 DD SYSOUT=A ← 4  
//*  
//* YOUR EXISTING LINKAGE-EDITOR JCL GOES HERE
```

The following JCL modifications must be made to your existing compile procedure:

1. Modify //SYSPRINT data set in the COBOL/VS compiler or Advantage CA-Optimizer step so that the listing is written to a temporary file.
2. Insert a new job step to execute the symbolic post-processor program after the compiler or Advantage CA-Optimizer step.

This JCL example assumes that you will pass the parameter statement in the PARM parameter of the EXEC statement. If you prefer to pass the parameter statement using a control card:

1. Remove all of the statements in the example that are marked by a ← 2.
2. Change the CARDS DD statement marked by ← 3 from:  

```
//CARDS DD DSN=&&CARDS,DISP=(OLD,DELETE)  
to  
//CARDS DD *
```
3. Add the parameter statement instream in the CARDS DD, beginning with the program name, starting in column one.

**Note:** The symbolic post-processor is shown with a region size of 512 KB. Processing larger programs may require a larger region size.

## COBOL II and Above or Advantage CA-Optimizer/II Example

```

/* YOUR EXISTING CICS PREPROCESSOR JCL GOES HERE
/*      (IF DOING A COMMAND LEVEL COMPILE)

/* YOUR EXISTING COBOL OR CA-OPTIMIZE/II JCL GOES HERE WITH
/* THE FOLLOWING MODIFICATIONS:

For CA-Optimizer/II only:

//SYSPRINT DD DSN=&&INPUT,DISP=(MOD,PASS),UNIT=SYSDA,    ← 1
// SPACE=(TRK,(15,5)),                                  ← 1
// DCB=(DSORG=PS,LRECL=133,BLKSIZE=2660,RECFM=FBA)     ← 1

Otherwise:

//SYSPRINT DD DSN=&&INPUT,DISP=(,PASS),UNIT=SYSDA,       ← 1
// SPACE=(TRK,(15,5)),                                  ← 1
// DCB=(DSORG=PS,LRECL=133,BLKSIZE=2660,RECFM=FBA)     ← 1

For all symbolic users:

/* IF THE COMPILER GIVES A RETURN CODE NOT HIGHER THAN 4 -
/* RUN THE CA SYMBOLIC POST-PROCESSOR
/*
/* THE //STEPLIB AND THE //STEPCLIB MAY BE UNNECESSARY.
/*
//GENPARG EXEC PGM=IN25PARG,PARM='PARAMETER STATEMENT' ← 2
//STEPLIB  DD DSN=CAI.CAILIB,DISP=SHR                  ← 2
//CARDS   DD DSN=&&CARDS,DISP=(,PASS),UNIT=SYSDA,       ← 2
// SPACE=(TRK,(1,1))                                  ← 2
/*
//POSTCOB EXEC PGM=IN25COB2,COND=(4,LT,COBOL),REGION=512K ← 3
//STEPLIB  DD DSN=CAI.CAILIB,DISP=SHR                  ← 3
//INPUT    DD DSN=&&INPUT,DISP=(OLD,DELETE)             ← 3
//OUTPUT   DD SYSOUT=A,                                ← 3
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=2660)            ← 3
//MESSAGE  DD SYSOUT=A                                 ← 3
//PROTSYM  DD DSN=PROTSYM.DSNAME,DISP=SHR             ← 3
//CARDS    DD DSN=&&CARDS,DISP=(OLD,DELETE)            ← 3
/*
/* IF THE COMPILER GIVES A RETURN CODE HIGHER THAN 4 -
/* THEN RUN AN IEBCGENER TO PRINT THE COBOL LISTING -
/*
//PRINT EXEC PGM=IEBCGENER,COND=(5,GT,COBOL)          ← 4
//SYSPRINT DD DUMMY                                    ← 4
//SYSIN    DD DUMMY                                    ← 4
//SYSUT1   DD DSN=&&INPUT,DISP=(OLD,DELETE)            ← 4
//SYSUT2   DD SYSOUT=A                                 ← 4
/*
/* YOUR EXISTING LINKAGE-EDITOR JCL GOES HERE

```

The following JCL modifications must be made to your existing compile procedure:

1. Modify the //SYSPRINT data set in the COBOL compiler or Advantage CA-Optimizer/II step so that the listing is written to a temporary file.
2. Insert a new job step to execute IN25PARM. This program builds the CARDS file containing the parameters for the post-processor.
3. Insert a new job step to execute the post-processor, only if the return code from the compile step is acceptable.
4. Insert a new job step to execute IEBCGENER to print the compiler listing if the return code from the compile step is unacceptable.

If you prefer to code the CARDS DD statement instream:

1. Omit the statements in the example that are marked by ← 2.
2. Change the CARDS DD statement marked by ← 3 from:  

```
//CARDS DD DSN=&&CARDS,DISP=(OLD,DELETE)
        to
//CARDS DD *
```
3. Add the parameter statement instream in the CARDS DD, beginning with the program name, starting in column one.

**Note:** The CA symbolic post-processor is shown with a region size of 512 KB. Processing larger programs may require a larger region size.

## Assembler Symbolic Support

Use the Assembler language post-processor program IN25SYMA to load symbolic information for your assembler programs. Only Assembler F, Assembler H, and the High Level Assembler are currently supported.

Here is how the symbolic post-processor works:

1. The symbolic post-processor operates **after** your assembly step and collects information from the assembler's //SYSPRINT DD statement.
2. This temporary data set is passed to the post-processor step through the //INPUT DD statement.
3. The original listing from the assembler's //SYSPRINT DD is written to the //OUTPUT DD defined in the post-processor step. This DD should be defined in the same way as the //SYSPRINT DD was defined before the procedure was modified.
4. Messages from the post-processor are written to the //MESSAGE DD, typically a SYSOUT dataset to be printed.
5. The symbolic file is referenced by the //PROTSYM DD statement with DISP=SHR.
6. Old records in the PROTSYM file are replaced by new ones during each run of the symbolic post-processor.

See the appendix, "[Symbolic File Maintenance](#)" for instructions on removing old program data from the Symbolic File.

**Note:** For source listing support to work properly, all CSECTs must appear in the listing and each CSECT statement must have a label.

## Assembler Options

The entire listing produced by the Assembler is required. The following options must be specified:

Option	Produces
ESD	External Symbol Dictionary
DXREF	Dsect Cross-Reference
USING	Using Map
XREF(FULL) or XREF(SHORT)	Full Cross-Reference Short Cross-Reference

**Important!** In order to receive full symbolic support, be sure that all statements defining the start of a DSECT are printed.

**Note:** High Level Assembler Release 2.0 users must also specify the option LIST(121).

## Post-Processor Options

Options are passed to the symbolic post-processor through a parameter statement in the CARDS file. The CARDS file can be an instream DD, a permanent file, or a temporary file generated by program IN25PARM prior to the post-processor step. In the example provided below, the CARDS file is generated by IN25PARM in the GENPARM step (indicated by ← 2).

- The only required field on the parameter statement is a program name field. For CICS applications, this should be the name in the CICS program definition or, for composite support, the monitor-name. For Batch applications, this should be the name of the first CSECT that appears in the listing. Also the load module must have the CSECTs linked in the same order that they appear in the listing. Failure to preserve this order can result in unpredictable results in the debugger.

- If you pass options using a control card, the program name **must** start in column 1.

The following example saves symbolic references for the program named ORDEDIT using a control card:

```
COL = 1...5...0  
      ORDEDIT
```

## Controlling Printed Output with the CUTPRINT Option

Since the Assembler source listing can be available online, you can request that the symbolic post-processor **not** print all or part of the listing produced by the Assembler. To make this request, add one of the following keywords immediately after the program name on the parameter statement:

**,CUTPRINT=ALL** produces no Assembler listing

**,CUTPRINT=REF** prints only the parts that precede the Cross-Reference list

The following sample parameter statement saves symbolic information for the program ORDEDIT but does not produce a hardcopy listing.

```
ORDEDIT,CUTPRINT=ALL
```

### Notes about the CUTPRINT option:

Specify the CUTPRINT parameter only when you do not want all or part of your listing printed. The entire listing will be printed if this parameter is missing from the parameter statement.

## Saving Your Listing for Display with the LISTER Option

You can also request that all or a portion of your source code be saved for online viewing. If saved, CA products will use the listing for symbolic displays, such as breakpoint displays from Advantage CA-InterTest for CICS.

To make this request you must add one of the following keywords to the parameter statement:

**,LISTER=ALL** saves the entire Assembler listing

**,LISTER=REF** saves only the parts that precede the Cross-Reference

For example, to save the symbolic references for the program ORDEDIT, not print any portion of the listing, and save the entire listing of the program, specify the following:

```
ORDEDIT , CUTPRINT=ALL , LISTER=ALL
```

### Notes about the LISTER Option

- If the LISTER parameter is omitted, no listing is saved in the Symbolic File.
- If only the LISTER parameter is specified, it must follow the program name.

## Setting Data as Nonpurgable

You could also request that any saved symbolic data for this program be marked as nonpurgable. If a program's data is marked as nonpurgable, the data will not be removed from the Symbolic File when deleting programs using a purge interval batch run. However, the data could be deleted by program name. See the appendix, "[Symbolic File Maintenance](#)" for instructions on deleting data from the Symbolic File.

To mark data as nonpurgable, add the following keyword to the parameter statement as the last option:

**,NOPURGE**

In the following example, the parameter specification records the symbolic references for the program ORDEDIT, prints the listing, does not save any portion of the listing, and does not allow symbolic data to be removed from the symbolic file by a purge interval batch run.

**ORDEDIT,NOPURGE**

In the next example, the parameter specification records the symbolic references for the program ORDEDIT, does not print a copy of the listing, does not save any portion of the listing, and does not allow data to be removed from the Symbolic File by a purge interval batch run.

**ORDEDIT,CUTPRINT=ALL,NOPURGE**

In this example, the parameter specification records the symbolic references for the program ORDEDIT, does not print a copy of the listing, saves the entire listing for online display, and does not allow data to be removed from the Symbolic File by a purge interval batch run.

**ORDEDIT,CUTPRINT=ALL,LISTER=ALL,NOPURGE**

# Assembler JCL Example

```

/* YOUR EXISTING CICS PREPROCESSOR JCL GOES HERE
/* (IF DOING A COMMAND LEVEL ASSEMBLER PROGRAM)
/*
/* YOUR EXISTING ASSEMBLER JCL GOES HERE WITH
/* THE FOLLOWING MODIFICATIONS
/*
//SYSPRINT DD DSN=&&INPUT,DISP=(,PASS),UNIT=SYSDA,      ← 1
// SPACE=(TRK,(15,5)),                                ← 1
// DCB=(DSORG=PS,LRECL=121,BLKSIZE=2420,RECFM=FBA)    ← 1
/*
/* IF THE ASSEMBLER GIVES A RETURN CODE NOT HIGHER THAN 0
/* THEN RUN THE CA SYMBOLIC POST-PROCESSOR
/*
//GENPARM EXEC PGM=IN25PARM,PARM='PARAMETER STATEMENT' ← 2
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR                  ← 2
//CARDS DD DSN=&&CARDS,DISP=(,PASS),                  ← 2
// UNIT=SYSDA,SPACE=(TRK,(1,1))                      ← 2
/*
//POSTASM EXEC PGM=IN25SYMA,COND=(0,LT,ASM),REGION=512K ← 3
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR                  ← 3
//INPUT DD DSN=&&INPUT,DISP=(OLD,DELETE)               ← 3
//OUTPUT DD SYSOUT=A,                                 ← 3
// DCB=(RECFM=FBM,LRECL=121,BLKSIZE=2420)           ← 3
//MESSAGE DD SYSOUT=A                                 ← 3
//PROTSYM DD DSN=PROTSYM.DSNAME,DISP=SHR             ← 3
//CARDS DD DSN=&&CARDS,DISP=(OLD,DELETE)              ← 3
/*
/* IF THE ASSEMBLER GIVES A RETURN CODE HIGHER THAN 0
/* THEN RUN AN IEBCGENER TO PRINT THE ASSEMBLER LISTING
/*
//PRINT EXEC PGM=IEBCGENER,COND=(0,EQ,ASM)            ← 4
//SYSPRINT DD DUMMY                                    ← 4
//SYSIN DD DUMMY                                       ← 4
//SYSUT1 DD DSN=&&INPUT,DISP=(OLD,DELETE)              ← 4
//SYSUT2 DD SYSOUT=A                                  ← 4
/*
/* YOUR EXISTING LINKAGE-EDITOR JCL GOES HERE

```

The following JCL modifications must be made to your existing Assembler procedure to use the IN25SYMA program:

1. Modify the //SYSPRINT DD for the Assembler step so that it points to a temporary dataset with the following attributes:

```
// DCB=(RECFM=FBA,LRECL=121,BLKSIZE=12100)
```

For Assembler H, specify RECFM=FBM unless your assembler was generated to produce FBA listings.

2. Insert a new job step to execute IN25PARM. This program builds the CARDS file containing the parameters for the post-processor.
3. Insert a new job step to execute the post-processor, only if the return code from the Assembler step is acceptable.
4. Insert a new job step to execute IEBGENER to print the program listing if the return code from the Assembler step is unacceptable.

If you prefer to code the CARDS DD statement instream:

1. Omit the statements in the example that are marked by ← 2.

2. Change the //CARDS DD statement from:

```
//CARDS DD DSN=&&CARDS,DISP=(OLD,DELETE)
         to
//CARDS DD *
```

3. Add the parameter statement instream in the CARDS DD, beginning with the program name, starting in column one.

**Note:** The CA symbolic post-processor is shown with a region size of 512 KB. Processing larger programs may require a larger region size.

## PL/I Symbolic Support

Use the PL/I symbolic post-processor, IN25SYMP, to store information about your PL/I applications in the symbolic file.

Here is how the symbolic post-processor works:

1. The symbolic post-processor operates **after** your compile step and collects information from the compiler's //SYSPRINT DD statement.
2. This temporary data set is passed to the post-processor step through the //INPUT DD statement.
3. The original listing from the compiler's //SYSPRINT DD is written to the //OUTPUT DD defined in the post-processor step. This DD should be defined in the same way as the //SYSPRINT DD was defined before the procedure was modified.
4. Messages from the post-processor are written to both the //MSGSDD and the //MESSAGES DD, typically SYSOUT data sets to be printed.
5. The symbolic file is referenced by the //PROTSYMDD statement with DISP=SHR.
6. Old records in the PROTSYM file are replaced by new ones during each run of the symbolic post-processor.

See the appendix, "[Symbolic File Maintenance](#)" for instructions on removing old program data from the file.

## Compiler Options

Suggested parameters for the PL/I optimizing compiler provide enough information for successful symbolic debugging. The LIST option (Assembler-like listing of the generated code) is optional, but the tables of offsets must be provided.

### The following options must be used with the OS PL/I or PL/I for MVS and VM compiler:

AGGREGATE	OPTIONS
ATTRIBUTES (FULL)	SOURCE
MAP	STMT or GOSTMT
NEST	STORAGE
NOGONUM	XREF (FULL)
NOMARGINI	
NONUMBER	

### The following options must be used with the Visual Age or Enterprise PL/I compiler:

AGGREGATE	NEST
ATTRIBUTES (FULL)	NONUMBER
NOGONUM	OPTIONS
LIMITS (NAME (31))	SOURCE
LIST	NOSTMT
MAP	STORAGE
NOMARGINI	XREF (FULL)

**Note:** Because of special considerations, if you must use the %NOPRINT compiler option contact Computer Associates technical support.

**Note:** For the OS PL/I and the PL/I for MVS and VM compilers, the ESD option is required for programs that have controlled variables.

## Installation Options

For symbolic file compile date-time to load module checking on PL/I programs to function, your PL/I compiler must have been installed with the TSTAMP=YES option (the default).

## Unsupported Features

The following features of the PL/I compilers are not supported:

- For programs that are compiled with the Visual Age PL/I or the Enterprise PL/I compiler using the RENT option, variables that are defined as EXTERNAL cannot be referenced by Advantage CA-InterTest Batch commands.
- For programs that are compiled with the Visual Age PL/I or the Enterprise PL/I compilers, variables defined as WIDECHAR cannot be referenced by Advantage CA-InterTest Batch commands.
- For programs that are compiled with the OS PL/I or PL/I for MVS compiler, arrays with variable lengths cannot be referenced by Advantage CA-InterTest Batch commands.
- For all PL/I compilers, variables defined as AREA, OFFSET, or COMPLEX cannot be referenced by Advantage CA-InterTest Batch commands.
- For multi-file source input (the use of copy books) the program must first be pre-processed to resolve all macro definitions.

## Post-Processor Options

Options are passed to the symbolic post-processor through a parameter statement in the CARDS file. The CARDS file can be an instream DD, a permanent file, or a temporary file generated by program IN25PARM prior to the post-processor step. In the example provided below, the CARDS file is generated by IN25PARM in the GENPARM step (indicated by ← 2).

- The only required field on the parameter statement is a program name field. For CICS applications, this should be the name in the CICS program definition or, for composite support, the monitor-name.
- If you pass options using a control card, the program name **must** start in column 1.

The following example saves symbolic references for the program named ORDEDIT using a control card:

```
COL = 1...5...0  
      ORDEDIT
```

## Controlling Printed Output with the CUTPRINT Option

Since the PL/I source listing can be available online, you can request that the symbolic post-processor **not** print all or part of the listing produced by the compiler. To make this request, add one of the following keywords immediately after the program name on the parameter statement:

**,CUTPRINT=ALL** produces no PL/I listing

**,CUTPRINT=REF** prints only the parts that precede the Cross-Reference table

**,CUTPRINT=OFFSET** prints only the parts that precede the table of offsets

The following sample parameter statement saves symbolic information for the program ORDEDIT but terminates printing after the XREF table:

```
ORDEDIT,CUTPRINT=REF
```

### Notes about the CUTPRINT option:

Specify the CUTPRINT parameter only when you do not want all or part of your listing printed. The entire listing will be printed if this parameter is missing from the parameter statement.

## Saving Your Listing for Display with the LISTER Option

You can also request that all or a portion of your source code be saved for online viewing. If saved, CA products will use the listing for symbolic displays, such as breakpoint displays from Advantage CA-InterTest for CICS.

To make this request you must add one of the following keywords to the parameter statement:

**,LISTER=ALL** saves the entire PL/I listing

**,LISTER=REF** saves only the parts that precede the Cross-Reference table

**,LISTER=OFFSET** saves only the parts that precede the table of offsets

For example, to save the symbolic references for the program ORDEDIT, not print any portion of the listing, and save the entire listing of the program, specify the following:

```
ORDEDIT , CUTPRINT=ALL , LISTER=ALL
```

### Notes about the LISTER Option

- If the LISTER parameter is omitted, no listing is saved in the Symbolic File.
- If only the LISTER parameter is specified, it must follow the program name.

## Setting Data as Nonpurgable

You could also request that any saved symbolic data for this program be marked as nonpurgable. If a program's data is marked as nonpurgable, the data will not be removed from the Symbolic File when deleting programs using a purge interval batch run. However, the data could be deleted by program name. See the appendix, "[Symbolic File Maintenance](#)" for instructions on deleting data from the Symbolic File.

To mark data as nonpurgable, add the following keyword to the parameter statement as the last option:

**,NOPURGE**

In the following example, the parameter specification records the symbolic references for the program ORDEDIT, prints the listing, does not save any portion of the listing, and does not allow symbolic data to be removed from the symbolic file by a purge interval batch run.

**ORDEDIT,NOPURGE**

In the next example, the parameter specification records the symbolic references for the program ORDEDIT, does not print a copy of the listing, does not save any portion of the listing, and does not allow data to be removed from the Symbolic File by a purge interval batch run.

**ORDEDIT,CUTPRINT=ALL,NOPURGE**

In this example, the parameter specification records the symbolic references for the program ORDEDIT, does not print a copy of the listing, saves the entire listing for online display, and does not allow data to be removed from the Symbolic File by a purge interval batch run.

**ORDEDIT,CUTPRINT=ALL,LISTER=ALL,NOPURGE**

## PL/I JCL Example

```

/*
/*      YOUR EXISTING CICS PREPROCESSOR JCL GOES HERE
/*      WHEN DOING A COMMAND LEVEL COMPILE

/*      YOUR EXISTING PL/I COMPILER JCL GOES HERE
/*      WITH THE FOLLOWING MODIFICATIONS
/*
//SYSPRINT DD DSN=&&INPUT,DISP=(MOD,PASS),UNIT=SYSDA,    ← 1
//          SPACE=(CYL,(1,1)),DCB=BLKSIZE=5000
/*
/*      IF THE COMPILER GIVES A RETURN CODE LESS THAN 9,
/*      THEN RUN THE CA SYMBOLIC POST-PROCESSOR
/*
//GENPARM EXEC PGM=IN25PARM,PARM='PARAMETER STATEMENT' ← 2
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR                    ← 2
//CARDS   DD DSN=&&CARDS,DISP=(,PASS),                  ← 2
//        UNIT=SYSDA,SPACE=(TRK,(1,1))                 ← 2
/*
//POSTPLI EXEC PGM=IN25SYMP,REGION=2048K,COND=(9,LT,PLI),
// PARM='NOREPORT'
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR                    ← 3
//PROTSYM DD DSN=PROTSYM.DSNAME,DISP=SHR                ← 3
//INPUTT  DD DSN=&&INPUT,DISP=(OLD,PASS)                 ← 3
//MSG     DD SYSOUT=*                                   ← 3
//MESSAGE DD SYSOUT=*                                   ← 3
//SYSPRINT DD SYSOUT=*                                   ← 3
//PLIDUMP DD SYSOUT=*                                   ← 3
//CARDS   DD DSN=&&CARDS,DISP=(OLD,DELETE)              ← 3
/*
/*      IF THE COMPILER GIVES A RETURN CODE HIGHER THAN 9
/*      THEN RUN AN IEBGENER TO PRINT THE PL/I LISTING
/*
//GENP    EXEC PGM=IEBGENER,COND=(9,GT,PLI)             ← 4
//SYSPRINT DD SYSOUT=*                                   ← 4
//SYSIN   DD DUMMY                                       ← 4
//SYSUT1  DD DSN=&&INPUT,DISP=(OLD,DELETE)              ← 4
//SYSUT2  DD SYSOUT=*                                   ← 4
/*
/*      YOUR EXISTING PRE-LINKAGE-EDITOR JCL GOES HERE (if
necessary)
/*
/*      YOUR EXISTING LINKAGE-EDITOR JCL GOES HERE

```

The following JCL modifications must be made to your existing PL/I compile procedure to use the IN25SYMP program:

1. Modify the //SYSPRINT DD for the compile step so that it points to a temporary dataset with BLKSIZE > 129.
2. Insert a new job step to execute IN25PARM. This program builds the CARDS file containing the parameters for the post-processor.
3. Insert a new job step to execute the post-processor, only if the return code from the compile step is acceptable.
4. Insert a new job step to execute IEBGENER to print the compiler listing if the return code from the compile step is unacceptable.

If you prefer to code the CARDS DD statement instream:

1. Omit the statements in the example that are marked by ← 2.
2. Change the //CARDS DD statement from:  

```
//CARDS DD DSN=&&CARDS,DISP=(OLD,DELETE)
           to
//CARDS DD *
```
3. Add the parameter statement instream in the CARDS DD, beginning with the program name, starting in column one.

**Note:** The CA symbolic post-processor is shown with a region size of 2048KB. Processing larger programs may require a larger region size.

# Symbolic File Maintenance

---

The IN25UTIL batch utility program maintains and reports on the symbolic file. This program runs in batch, separate from the post-processors that are used to load symbolic information into the symbolic file.

IN25UTIL functions are requested using control statements in the CARDS file.

## Control Statements

All control statements in the CARDS file must begin in column 1.

PASSWORD=  
12345678/xxxxxxxx

Specifies a password that is required when maintaining the symbolic file. You only need to specify the password once, but it must precede the first update request.

The value specified must match the value of the SYMPSWD= keyword in the IN25SOPT macro.

INITIALIZE

Initializes the symbolic file. This function **must always** be run after the symbolic file is created using VSAM Access Method Services.

For a newly defined file, the IN25UTIL program preformats all records. If you perform this function for an existing file, all symbolic data is removed.

The PASSWORD control statement must precede the INITIALIZE control statement.

PURGE=nnn

Removes symbolic data for any program that has not been compiled or assembled within the number of days specified by nnn, a decimal number from 1 to 365.

The PASSWORD control statement must precede the PURGE control statement.

Data for programs loaded using the NOPURGE post-processor option cannot be purged.

DELETE=xxxxxxxx

Deletes all symbolic data for the program specified by xxxxxxxx.

The PASSWORD control statement must precede the DELETE control statement.

REPORT

Produces a Symbolic File report that contains statistics and a detailed report on each program.

PRINT=xxxxxxxx

Prints the saved source listing for program xxxxxxxx.

UNLOAD=ALL/  
xxxxxxxx

Writes all symbolic data or just the symbolic data for program xxxxxxxx to the unload device.

```
RELOAD=ALL/  
xxxxxxx
```

Reloads all symbolic data or just the symbolic data for program xxxxxxxx from the device specified by the //RELOAD DD statement. Programs are reloaded only if they do not exist on the Symbolic File.

The PASSWORD control statement must precede the RELOAD control statement.

## Examples

The following examples contain sample JCL for performing common symbolic file maintenance tasks.

### Initializing a Symbolic File

The following example initializes a symbolic file.

```
//UTILITY JOB  
//STEP1 EXEC PGM=IN25UTIL  
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR  
//MESSAGE DD SYSOUT=*  
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR  
//CARDS DD *  
PASSWORD=12345678  
INITIALIZE  
/*
```

## Purging Symbolic Information by Age

The following example purges all programs that have not been compiled or assembled within the last 20 days.

```
//UTILITY JOB
//STEP1 EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//MESSAGE DD SYSOUT=*
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD *
PASSWORD=12345678
PURGE=20
/*
```

## Deleting Symbolic Information by Program

The following example deletes all symbolic data for programs ORDEDIT and TEST1.

```
//UTILITY JOB
//STEP1 EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//MESSAGE DD SYSOUT=*
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD *
PASSWORD=12345678
DELETE=ORDEDIT
DELETE=TEST1
/*
```

## Generating Reports and Purging Programs

The following example generates a system report, purges all programs that have not been compiled or assembled within 90 days, and generates another system report.

```
//UTILITY JOB
//STEP1 EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//MESSAGE DD SYSOUT=*
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD *
PASSWORD=12345678
REPORT
PURGE=90
REPORT
/*
```

---

## Unloading Programs

The following example unloads all programs on the symbolic file to tape.

```
//UTILITY JOB
//STEP1 EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//MESSAGE DD SYSOUT=*
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//UNLOAD DD UNIT=TAPE,VOL=SER=UNLDTP,
// LABEL=(,NL),DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=2042,BLKSIZE=20420)
//CARDS DD *
UNLOAD=ALL
/*
```

## Reloading Programs

The following example reloads all programs on tape to the symbolic file.

```
//UTILITY JOB
//STEP1 EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//MESSAGE DD SYSOUT=*
//RELOAD DD UNIT=TAPE,
// VOL=SER=UNLDTP,
// LABEL=(,NL),
// DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=2042,BLKSIZE=20420)
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD
PASSWORD=12345678
RELOAD=ALL
/*
```

## Printing a Program Listing

The following example prints the saved listing for program ORDEDIT.

```
//UTILITY JOB
//STEP1 EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//MESSAGE DD SYSOUT=*
//OUTPUT DD SYSOUT=A,DCB=(LRECL=133,BLKSIZE=3990)
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD *
PRINT=ORDEDIT
/*
```

## Reorganizing the Symbolic File

The following example reorganizes or changes the size of the symbolic file. This job unloads all programs, deletes and defines the symbolic file, initializes the symbolic file, reloads all programs, and generates a system report.

```
//UTILITY JOB
//UNLOAD EXEC PGM=IN25UTIL
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//MESSAGE DD SYSOUT=*
//UNLOAD DD DISP=(,PASS),
// UNIT=TAPE,
// VOL=SER=RELDTAP,
// LABEL=(,NL),
// DCB=(RECFM=FB,LRECL=2042,BLKSIZE=20420)
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD *
UNLOAD=ALL
/*
//IDCAMS EXEC PGM=IDCAMS,COND=(0,NE,UNLOAD)
//SYSUT1 DD UNIT=SYSDA,VOL=SER=SYMVOL,DISP=SHR
DELETE 'CAI.PROTSYM'
DEFINE CLUSTER (NAME(CAI.PROTSYM) -
                VOLUME(SYMVOL) -
                FILE(SYSUT1) -
                CYLINDERS(20) -
                CISZ(2048) -
                RECSZ(2040 2040) -
                SHR(4 4) -
                NUMBERED) -
DATA (NAME(CAI.PROTSYM.DATA))
/*
//RELOAD EXEC PGM=IN25UTIL,COND=(0,NE,UNLOAD)
//STEPLIB DD DSN=CAI.CAILIB,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//MESSAGE DD SYSOUT=*
//RELOAD DD DISP=(OLD,KEEP),
// UNIT=TAPE,
// VOL=SER=RELDTAP,
// LABEL=(,NL),
// DCB=(RECFM=FB,LRECL=2042,BLKSIZE=20420)
//PROTSYM DD DSN=CAI.PROTSYM,DISP=SHR
//CARDS DD *
PASSWORD=12345678
INITIALIZE
RELOAD=ALL
REPORT

/*
```



# Index

## A

---

ACCEPT command, 4-2

ACCEPT processing, 3-2  
maintenance, 4-11  
MR31ACC, 3-42

accepting maintenance, 4-11

Advantage CA-Optimizer II options, A-5

Advantage CA-Optimizer options, A-5

Advantage CA-Roscoe, customizing, 3-34

ALIB, allocating, 3-22

APAR, 4-3

APPLY CHECKing maintenance, 4-8

APPLY command, 4-2

APPLY processing  
MR31APP, 3-19  
SMP/E, 3-2

applying maintenance, 4-10

ASMINT, 3-32

Assembler programs  
Assembler options, A-15  
JCL example, A-19  
non-purgable data, A-18  
online listing display, Assembler, A-17  
post-processors for, A-14  
printed output, controlling, A-16

symbolic support, A-14

## B

---

Batch Link  
customize user exit, 3-18  
feature, 3-19  
feature refresh, 4-11

BookManager files  
downloading to a PC, 3-12  
unloading from cartridge, 3-11

BTS considerations  
ISPF users, 3-38  
message switching, 3-41  
screen handling in BTS, 3-40  
testing of IMS/DC programs, 3-28

BTS, reviewing and customizing for, 3-38

BTS0067I message, screen handling solution,  
3-40

BTSSYO option, 3-41

## C

---

CA Common Services  
CAIRIM, 3-6  
installation step, 3-6  
installing, 2-4  
LMP key, 3-7  
system requirements, 2-3

---

CA Technical Support, 4-3  
CAI Resource Initialization Manager. *See*  
CAIRIM  
CAI.RPFLIB, 3-37  
CAI.SAMPJCL, 3-10  
CAIMR31, 3-16  
CAINITE5, 3-15  
CAIRIM  
    batch link feature, 3-19  
    installing, 3-6  
    system requirements, 2-3  
CAMRASM, 3-37  
CAMRCMD, 3-30  
CAMRCOB, 3-37  
CAMRCOB2, 3-37  
CAMRDRVR, 3-25  
CAMRPLI, 3-37  
CAMRUSR1, 3-18  
CAMRXMOD, 3-18  
CA-Optimizer. *See* Advantage CA-Optimizer  
CA-Roscoe. *See* Advantage CA-Roscoe  
CLISTs, customizing, 3-25  
COB2INT, 3-31  
COBINT, 3-30  
COBOL programs  
    Advantage CA-Optimizer options, A-5  
    compiler options, A-5  
    JCL example, A-10  
    non-purgable data, A-9  
    online listing display, A-8  
    post-processors for, A-2  
    printed output, controlling, A-7  
    symbolic support, A-2  
compile PROCs, customizing, 3-30

cover letter instructions, reviewing, 4-6  
customization  
    Batch Link user exit, 3-18  
    define excluded programs, 3-18  
Customization  
    Step 11, 3-18  
CUTPRINT  
    Assembler, A-16  
    COBOL, A-7  
    PL/I, A-24  
CVH3100, 3-17  
CZ27000, 3-17

---

## D

define excluded programs, 3-18  
demo, running, 3-37  
distribution libraries  
    allocating, 3-14  
    described, 4-1  
    hardware requirements, 2-5  
    SMP/E, 4-1  
documentation viewers, 2-2  
DYNAMBR, 2-1

---

## E

Enterprise PL/I compiler options, A-22  
excluded programs, define, 3-18  
exit, user, Batch Link, 3-18

---

## F

features unsupported, PL/I compilers, A-23

---

## H

---

hardware requirements, 2-5  
    additional files, 2-7  
    distribution libraries, 2-5  
    target libraries, 2-6

## I

---

IMS/DB or DL/I programs, files for testing,  
3-27

IN25UTIL, B-1  
    PASSWORD, B-1

INCLIBs, allocating, 3-22

informational APAR or PIB, 4-3

installation  
    materials, 3-3  
    steps, checklist, 3-4  
    worksheet, completing, 3-8

INT1CLIB, 3-26

INT1CLOG, 3-26

INT1REPT, 3-27

INTBTSX  
    BTSSYSO, 3-41  
    customizing for BTS, 3-38

INTISPF, 3-23

INTROS, 3-35

ISPF Main Panel, 3-33

## J

---

JCL example  
    Assembler, A-19  
    COBOL, A-10  
    loading sample from tape, 4-6

PL/I, A-27

## L

---

libraries  
    distribution, 4-1  
    SMP/E, 4-1  
    target, 4-1

License Management Program, 2-4

Link, Batch, user exit, 3-18

LISTER  
    Assembler, A-17  
    COBOL, A-8  
    PL/I, A-25

LMP key, 3-7

## M

---

maintenance  
    accepting, 4-11  
    APPLY CHECKing, 4-8  
    applying, 4-10  
    customizing SMP/E procedure, 4-7  
    delivery, 4-3  
    receiving, 4-7  
    saving, 4-12  
    symbolic file, B-1  
    validating, 4-11

message switching and DISPLAY or figure  
verbs, 3-41

MR310ISPF, 3-33

MR31ACC, 3-42

MR31ACCM, accepting maintenance, 4-11

MR31ALC, 3-14

MR31ALIB, 3-22

MR31APCM APPLY CHECKs, 4-8

---

MR31APP, 3-19  
MR31APPM, reapplying SYSMODs, 4-10  
MR31CBTS, 3-28  
MR31CDB2, 3-29  
MR31CIMS, 3-27  
MR31CLST, 3-26  
MR31EPL, 3-34  
MR31ETSO, 3-34  
MR31INCL, 3-22  
MR31KEYS, 3-30  
MR31PROF, 3-20  
MR31REC, 3-17  
MR31RECM, receiving maintenance, 4-7  
MR31RESM, restoring SYSMODs, 4-9  
MR31RTL, 3-19  
    refresh Batch Link feature, 4-11  
MR31SMPE, 3-16  
MR31STRT, 3-29  
MR31TSOL, 3-33  
MR31ZONE, 3-16

## N

---

naming conventions for PTFs, 4-4  
non-purgable data  
    Assembler, A-18  
    COBOL, A-9  
    PL/I, A-26  
NOPURGE  
    Assembler, A-18  
    COBOL, A-9  
    PL/I, A-26

## O

---

operating environments, 2-1  
    compiler support, 2-2  
    documentation viewers, 2-2  
    TSO LOGON requirements, 2-1  
options, customizing, 3-23  
ORDEDIT,NOPURGE  
    Assembler, A-18  
    COBOL, A-9  
    PL/I, A-26  
OS PL/I compiler options, A-22

## P

---

PASSWORD, B-1  
PDF files, 3-13  
PL/I compilers, unsupported features, A-23  
PL/I for MVS and VM compiler options, A-22  
PL/I programs  
    compiler options, A-22  
    date-time checking, A-22  
    installation options, A-22  
    JCL example, A-27  
    non-purgable data, A-26  
    online listing display, PL/I, A-25  
    post-processor options, A-23  
    post-processors for, A-21  
    printed output, controlling, A-24  
    symbolic support, A-21  
PLIINT, 3-32  
post-processors  
    Assembler, A-14  
    COBOL, A-2  
    PL/I, A-21  
PPOPTION, 3-18  
    library, 3-23

---

PPOPTION member CAMRUSR1, 3-18

printed output

- controlling Assembler programs, A-16
- controlling COBOL programs, A-7
- controlling PL/I programs, A-24

Product Information Bulletins (PIBs), 4-3

PROFLIBs, allocating, 3-20

programs

- Assembler, A-14
- COBOL compiler options, A-4
- define excluded, 3-18
- PL/I, A-21
- printing, B-6
- purging, B-4
- reloading, B-5
- unloading, B-5

PROTSYM, 3-21

- allocating, 3-21
- initializing, B-2
- maintenance, B-1
- printing, B-2
- unload, B-2

PROTSYM, upgrading from an earlier release, 3-6

PTFs, 4-4

purging, programs, B-4

## R

---

reapplying SYSMODs, 4-10

RECEIVE processing

- preinstallation considerations, 3-2
- receive product step, 3-17

receiving maintenance, 4-7

replacement module, 4-4

RESERVE and DEQ macros, 3-21

RESTORE command, 4-2

restoring SYSMODs, 4-9

## S

---

SAMPJCL, loading, 3-10

sample JCL library, loading, 3-10

saving maintenance, 4-12

SMP procedure, customizing, 3-16

SMP SELECT control statement, 3-42

SMP/E, 3-1

- ACCEPT processing, 3-2
- APPLY processing, 3-2
- customize procedure, 4-7
- customize the SMP procedure, 3-16
- libraries, allocating, 3-16
- libraries, applying maintenance, 4-1
- libraries, define product target, 3-16
- libraries, defining and allocating, 3-15
- RECEIVE processing, 3-2
- SMP modification control statements, 3-3
- SMPMCS, 3-3

SMPMCS, 3-3

Symbolic File

- maintenance, B-1
- password protection, B-1

SYSMODs

- reapplying, 4-10
- restoring, 4-9

system requirements, 2-1

- installation step, 3-7

## T

---

target libraries

- allocating, 3-14
- hardware requirements, 2-6

---

overview, 4-1  
TSO LOGON, adding ISPF libraries, 3-33

## U

---

Unicenter Common Services. *See* CA  
Common Services

unloading documentation, 3-11  
    BookManager, 3-11  
    PDF, 3-13

unsupported features, PL/I compilers, A-23

upgrading from an earlier release, 3-6

user exit, Batch Link, 3-18  
USERMOD, 4-3

## V

---

VH31PROT, 3-21  
Visual Age PL/I compiler options, A-22

## W

---

worksheet, installation, 3-8