ASG-SmartTest™ Installation Guide

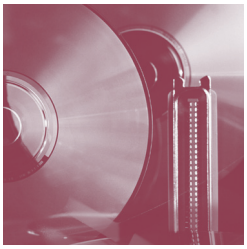
Version: 6.0

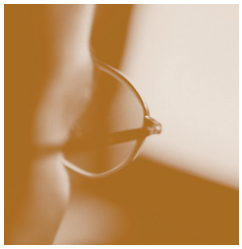
Publication Number: STX0300-60

Publication Date: February 2002

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

© 1989-2002 Allen Systems Group, Inc. All rights reserved.
All names and products contained herein are the trademarks or registered trademarks of their respective holders.





ASG Worldwide Headquarters Naples, Florida USA | asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 941.435.2200 Fax: 941.263.3692 Toll Free: 1.800.932.5536

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (941) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

© 2002 Allen Systems Group, Inc.
All names and products are trademarks or registered trademarks of their respective holders.

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)
- Verify whether you received an ASG Service Pack for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack.

If You Receive a Voice Mail Message:

1. Follow the instructions to report a production-down or critical problem.
2. Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
3. Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578	941.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.222	49.89.45716.400	support.de@asg.com
Singapore	65.332.2922	65.337.7228	support.sg@asg.com
All other countries:	1.941.435.2200		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578		
Asia	65.332.2922	**Japan/Telecom**	0041.800.9932.5536
Australia	0011.800.9932.5536	**Netherlands**	00.800.3354.3578
Denmark	00.800.9932.5536	**New Zealand**	00.800.9932.5536
France	00.800.3354.3578	**Singapore**	001.800.3354.3578
Germany	00.800.3354.3578	**South Korea**	001.800.9932.5536
Hong Kong	001.800.9932.5536	**Sweden/Telia**	009.800.9932.5536
Ireland	00.800.9932.5536	**Switzerland**	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536	**Thailand**	001.800.9932.5536
Japan/IDC	0061.800.9932.5536	**United Kingdom**	00.800.3354.3578
		All other countries	1.941.435.2200

ASG Web Site

Visit http://www.asg.com, ASG's World Wide Web site.

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (941) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

Contents

Preface

This *ASG-SmartTest Installation Guide* explains the installation and maintenance of ASG-SmartTest (herein called SmartTest). SmartTest is the Testing/Debugging component of the ASG Existing Systems Workbench (ESW). It automates the time consuming and error prone process of testing and debugging application programs.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

About this Publication

This publication consists of these chapters:

- Chapter 1, "Introduction," provides an overview of SmartTest and ASG's ESW products.
- Chapter 2, "Installing and Customizing SmartTest," describes the system resources needed and provides the instructions for installing and maintaining SmartTest.
- Chapter 3, "Customizing CICS," contains installation instructions and describes the system resources needed to install and maintain SmartTest-CICS.
- Chapter 4, "Customizing IMS," describes the procedures used to customize specific SmartTest-IMS libraries.
- Chapter 5, "Validating ASM," describes the validation of the SmartTest-ASM installation.
- Chapter 6, "Customizing APS," provides information for using SmartTest with APS programs.
- Chapter 7, "Validating PLI," describes how to validate the SmartTest-PLI installation.
- Chapter 8, "Customizing Stored Procedures," provides instructions for installing and customizing the SmartTest Stored Procedure option.

Related Publications

The documentation library for ASG-SmartTest consists of these publications (where *nn* represents the product version number):

- *ASG-Center Installation Guide* (CNX0300-*nn*) contains installation and maintenance information for ASG-Center, the common set of libraries shared by the ASG-ESW suite of products.
- *ASG-SmartTest CICS User's Guide* (STC0200-*nn*) contains specific commands and test session setup information for the CICS environments.
- *ASG-SmartTest for COBOL and Assembler User's Guide* (STA0200-*nn*) contains introductory and usage information for COBOL and Assembler. It also contains test session setup information for the TSO, ISPF, IMS/DB, DB/2, BTS, and Batch environments.
- *ASG-SmartTest IMS User's Guide* (STM0200-*nn*) contains specific commands and test session setup information for the IMS/DC environments.
- *ASG-SmartTest Installation Guide* (STX0300-*nn*) contains information for installing and maintaining ASG-SmartTest.
- *ASG-SmartTest PLI User's Guide* (STL0200-*nn*) contains introductory and usage information about how to use ASG-SmartTest with the PL/I language. It also contains test session setup information for the TSO, ISPF, IMS/DB, DB/2, BTS, and Batch environments.
- *ASG-SmartTest Quick Start for COBOL/ASM* (STA0900-*nn*) summarizes how to use ASG-SmartTest with the COBOL or Assembler language.
- *ASG-SmartTest Quick Start for PL/I* (STL0900-*nn*) summarizes how to use ASG-SmartTest with the PL/I language.
- *ASG-SmartTest Reference Guide* (STX0400-*nn*) contains detailed reference information about CUA pull-downs and pop-ups, ASG-SmartTest command syntax, and pseudo code.
- *ASG-SmartTest Reference Summary* (STX0600-*nn*) summarizes the syntax and usage of ASG-SmartTest commands.
- *ASG-SmartTest TCA User's Guide* (STT0200-*nn*) contains procedures for using the ASG-SmartTest-TCA (Test Coverage Analysis) option.

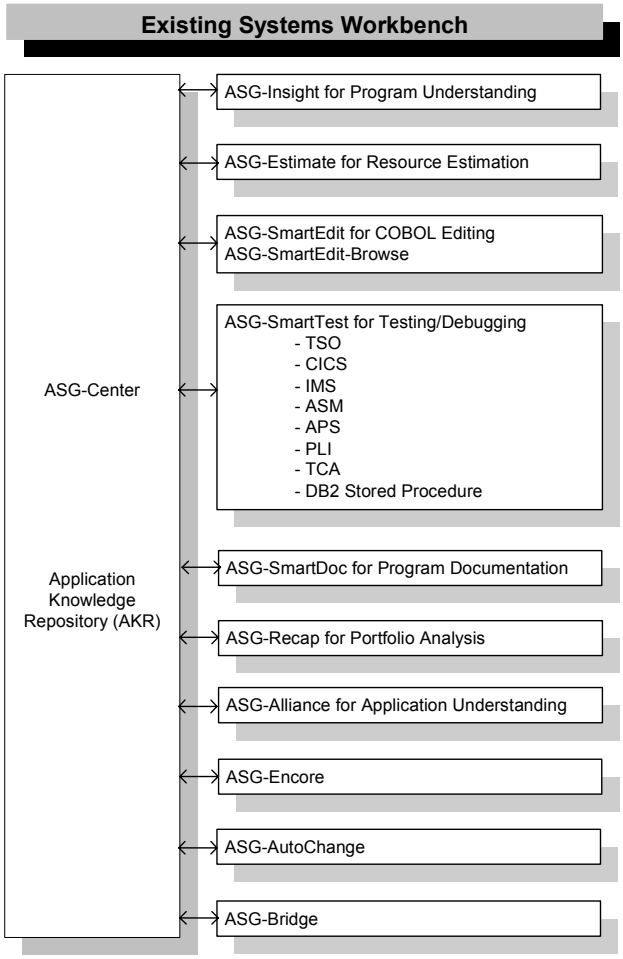
Note:

To obtain a specific version of a publication, contact the ASG Service Desk.

ASG-Existing Systems Workbench (ASG-ESW)

ASG-ESW (herein called ESW) is an integrated suite of components designed to assist organizations in enhancing, redeveloping, or re-engineering their existing systems. ESW products use the Application Knowledge Repository (AKR) to store source program analysis information generated by the Analytical Engine. Figure 1 represents the components of ESW.

Figure 1 • ASG Existing Systems Workbench



This table contains the name and description of each ESW component:

ESW Product	Herein Called	Description
ASG-Alliance	Alliance	The application understanding component that is used by IT professionals to conduct an analysis of every application in their environment. Alliance supports the analysis and assessment of the impact of change requests upon an entire application. Alliance allows the programmer/ analyst to accurately perform application analysis tasks in a fraction of the time it would take to perform these tasks without an automated analysis tool. The impact analysis from Alliance provides application management with additional information for use in determining the resources required for application changes.
ASG-AutoChange	AutoChange	The COBOL code change tool that makes conversion teams more productive by enabling quick and safe changes to be made to large quantities of code. AutoChange is an interactive tool that guides the user through the process of making source code changes.
ASG-Bridge	Bridge	The bridging product that enables field expansion for program source code, without being required to simultaneously expand the fields in files or databases. Because programs are converted in smaller groups, or on a one-by-one basis, and do not require file conversion, testing during the conversion process is simpler and more thorough.
ASG-Center	Center	The common platform for all ESW products. Center provides the common Analytical Engine to analyze the source program and store this information in the AKR. This common platform provides a homogeneous environment for all ESW products to work synergistically.

ESW Product	Herein Called	Description
ASG-Encore	Encore	The program re-engineering component for COBOL programs. Encore includes analysis facilities and allows you to extract code based on the most frequently used re-engineering criteria. The code generation facilities allow you to use the results of the extract to generate a standalone program, a callable module, a complement module, and a CICS server. Prior to code generation, you can view and modify the extracted Logic Segment using the COBOL editor.
ASG-Estimate	Estimate	The resource estimation tool that enables the user to define the scope, determine the impact, and estimate the cost of code conversion for COBOL, Assembler, and PL/I programs. Estimate locates selected data items across an application and determines how they are used (moves, arithmetic operations, and compares). Time and cost factors are applied to these counts, generating cost and personnel resource estimates.
ASG-Insight	Insight	The program understanding component for COBOL programs. Insight allows programmers to expose program structure, identify data flow, find program anomalies, and trace logic paths. It also has automated procedures to assist in debugging program abends, changing a computation, and resolving incorrect program output values.
ASG-Recap	Recap	The portfolio analysis component that evaluates COBOL applications. Recap reports provide function point analysis and metrics information, program quality assessments, intra-application and inter-application comparisons and summaries, and historical reporting of function point and metrics information. The portfolio analysis information can also be viewed interactively or exported to a database, spreadsheet, or graphics package.
ASG-SmartDoc	SmartDoc	The program documentation component for COBOL programs. SmartDoc reports contain control and data flow information, an annotated source listing, structure charts, program summary reports, exception reports for program anomalies, and software metrics.

ESW Product	Herein Called	Description
ASG-SmartEdit	SmartEdit	The COBOL editing component that can be activated automatically when the ISPF/PDF Editor is invoked. SmartEdit provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor (for example, the APS generator) during syntax checking. SmartEdit supports all versions of IBM COBOL, CICS, SQL, and CA-IDMS.
ASG-SmartTest	SmartTest	The testing/debugging component for COBOL, PL/I, Assembler, and APS programs in the TSO, MVS Batch, CICS (including file services), and IMS environments. SmartTest features include program analysis commands, execution control, intelligent breakpoints, test coverage, pseudo code with COBOL source update, batch connect, disassembled object code support, and full screen memory display.

Invoking ESW Products

The method you use to invoke an ESW product depends on your system setup. If you need assistance to activate a product, see your systems administrator. If your site starts a product directly, use the ISPF selection or CLIST as indicated by your systems administrator. If your site uses the ESW screen to start a product, initiate the ESW screen using the ISPF selection or CLIST as indicated by your systems administrator and then typing in the product command on the command line.

The product names can also vary depending on whether you access a product directly or through ESW. See "ESW Product Integration" on page xiii for more information about using ESW.

To initialize ESW products from the main ESW screen, select the appropriate option on the action bar pull-downs or type the product shortcut on the command line.

Product Name	Shortcut	ESW Pull-down Options
Alliance	AL	Understand ▸ Application
AutoChange	CC	Change ▸ Conversion Set
Bridge	BR	Change ▸ ASG-Bridge
Encore (Re-engineer)	EN	Re-engineer ▸ Program
Estimate	ES	Measure ▸ ASG-Estimate
Insight (Understand)	IN	Understand ▸ Program
Recap (Portfolio Analysis)	RC	Measure ▸ Portfolio
SmartDoc (Document)	DC	Document ▸ Program
SmartEdit	SE	Change ▸ Program **Or** Change ▸ Program with Options
SmartTest	ST	Test ▸ Module/Transaction

ESW Product Integration

Because ESW is an integrated suite of products, you are able to access individual ESW products directly or through the main ESW screen. As a result, you might see different fields, values, action bar options, and pull-down options on a screen or pop-up depending on how you accessed the screen or pop-up.

Certain ESW products also contain functionality that interfaces with other ESW products. Using SmartTest as an example, if Alliance is installed, SmartTest provides a dynamic link to Alliance that can be used to display program analysis information. If Insight is installed and specified during the analyze, the Insight program analysis functions are automatically available for viewing logic/data relationships and execution path. For example, the Scratchpad option is available on the Options pull-down if you have Insight installed. Access to these integrated products requires only that they be installed and executed in the same libraries.

Examples

Example 1. Figure 2 shows the Encore Primary screen that displays when you access Encore directly.

The Encore Primary screen contains these eight action bar menu items: File, View, Extract, Generate, Search, List, Options, and Help.

Figure 2 • Encore Primary Screen

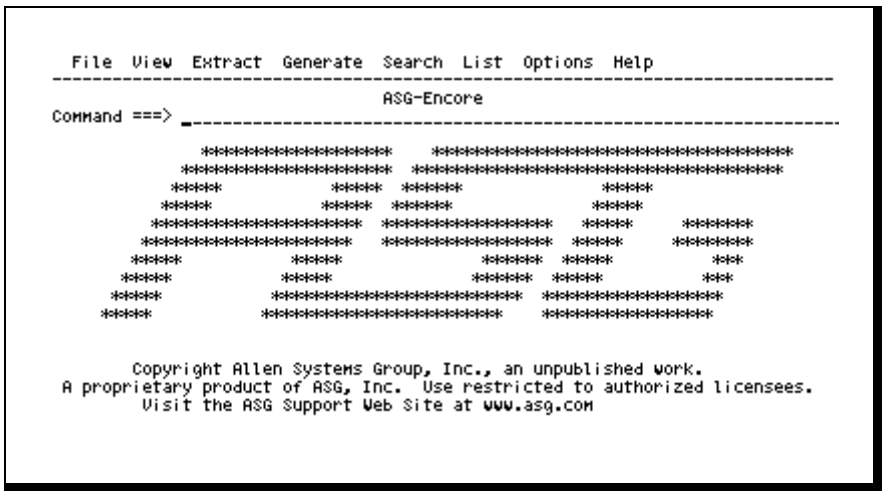
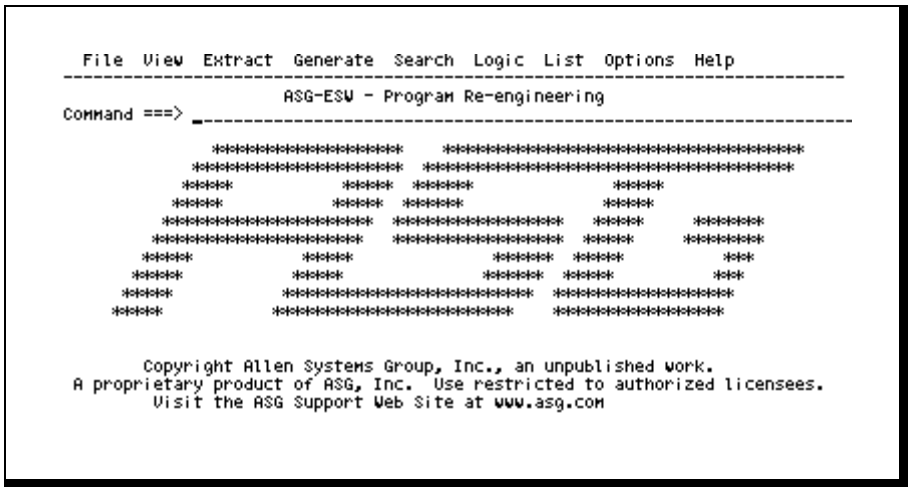


Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ▸ Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays if Insight is installed.

Figure 3 • ESW Encore Primary Screen



Example 2. Figure 4 shows the File - Analyze Submit pop-up that displays when you access SmartTest directly. Figure 5 shows the File - Analyze Submit pop-up that displays when you access SmartTest through ESW.

Notice that the Analyze features field in Figure 5 lists additional ESW products than shown on Figure 4. This field is automatically customized to contain the ESW products you have installed on your system.

The actions shown on these screens also vary. For example, the D action (ASG-SmartDoc Options) is available on the File - Analyze Submit screen if the SmartDoc product is installed on your system. In Figure 4, the ASG-SmartDoc Options action is not available.

Figure 4 • File - Analyze Submit Screen

```
                          File - Analyze Submit
Command ===> ____________________________________________________________

          E - Edit JCL                                S - Submit JCL

Compile and link JCL (PDS or sequential):
   Data set name  'USER12.REL.CNTL(VIAPCOBC)'

Analyze features (Y/N):
   ASG-SmartTest:  Y   Extended Analysis:  N

AKR data set name 'USER12.GENERAL.AKR'
AKR program name                      (if overriding PROGRAM-ID)

Analyze options:
   ____________________________________________________________________
   ____________________________________________________________________
   ____________________________________________________________________

Compile? (Y/N) . . . . . . . . . . Y     (Y if needed by features)
Link load module reusable? (Y/N) Y
```

Figure 5 • File - Analyze Submit Screen (Accessed through ESW)

```
                          File - Analyze Submit
Command ===> ____________________________________________________________

          E - Edit JCL     S - Submit JCL     D - ASG-SmartDoc Options

Compile and link JCL (PDS or sequential):
   Data set name  'USER12.REL.CNTL(HTEST)'

Analyze features (Y/N):
   ASG-Insight:  Y   ASG-SmartTest:  Y   Extended Analysis:  N
   ASG-SmartDoc:  N   ASG-Encore:  N
AKR data set name 'USER12.GENERAL.AKR'
AKR program name  __________          (if overriding PROGRAM-ID)

Analyze options:
   ____________________________________________________________________
   ____________________________________________________________________
   ____________________________________________________________________

Compile? (Y/N) . . . . . . . . . . Y     (Y if needed by features)
Link load module reusable? (Y/N) Y     (ASG-SmartTest
```

Publication Conventions

ASG uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
lowercase italic monospace	Information that you provide according to your particular situation. For example, you would replace *filename* with the actual name of the file.
Monospace	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax.
	Also used for denoting brief examples in a paragraph.
Vertical Separator Bar (|) with underline	Options available with the default value underlined (e.g., Y|N).

1 Introduction

This chapter provides an introduction to SmartTest-CICS and contains these sections:

Note:

ASG-Center installation and customization must be performed before customizing SmartTest. If Center has not been installed, see the *ASG-Center Installation Guide*.

ASG Service Pack

Verify whether you received an ASG Service Pack for this product. If so, read the instructions for installing the Service Pack before proceeding with the product installation. The installation instructions are located in a text file on the distribution media included with the Service Pack. If you have any problems with the Service Pack, contact the ASG Service Desk.

Overview

SmartTest is the Testing/Debugging component of ESW for COBOL, PL/I, Assembler, and APS programs in the TSO, MVS Batch, CICS (including file services), and IMS environments. SmartTest features include program analysis commands, execution control, intelligent breakpoints, test coverage, pseudo code with COBOL source update, batch connect, disassembled object code support, and full-screen memory display.

SmartTest provides a comprehensive testing and debugging tool that brings intelligence to testing. Knowledge of the program (syntax, logic relationships, and execution flows) is the foundation for providing a broad array of testing/debugging utilities.

SmartTest Features

SmartTest executes in the CICS, IMS/DC, and TSO/ISPF environments and provides full ISPF compatibility, which facilitates learning and usage through a standard IBM interface. It is an electronic window through which COBOL, PL/I, and Assembler programs are viewed, analyzed, tested, and debugged.

SmartTest provides these comprehensive testing and debugging features and functions:

Feature	Description
Comprehensive Program Knowledge	Provides the testing environment with complete knowledge about the program, including syntax, logic relationships, and execution flow. This knowledge is stored in the AKR.
Reduction of Setup Barriers	Eliminates test session setups and TSO resource consumption via a batch connection facility. Conversion utilities reduce the time spent setting up TSO/ISPF.
Storage Protection - CICS only	Eliminates a major deficiency of CICS, which is its lack of storage protection. Without SmartTest, a transaction can corrupt the storage of other programs or CICS management modules and storage, which can result in a system crash.
COBOL and PL/I Intelligence	Supports COBOL-compatible pseudo code that can be temporarily added to a program, saved permanently in the AKR, or automatically added to the existing program. Dynamic Breakpoints can be set based on COBOL or PL/I syntax statements or conditions.
Script Automation	Provides a predefined command sequence that can be included in any test session. These script files can be created automatically during a test session, then used for regression testing of a program.
Scrollable Window Displays	Provides easy access to related information using in-context windows for data display and logic.
Memory Facilities	Provides memory identification, display, and update facilities.

Feature	Description
Hot Key Branching	Provides easy access to related logic between paragraphs, sections, and CALLed routines.
Assembler Integration	Provides full-screen Assembler integration of programs, CALLed modules, or memory display of data elements.
Integrated Program Analysis	Provides access to program analysis functions for viewing logic/data relationships, execution paths, etc., through the AKR.
Center Integration	Supports the automation of the software maintenance cycle.

2 Installing and Customizing SmartTest

This chapter describes the installation and customization steps for SmartTest and contains these sections:

Prerequisite

Center installation and customization must be performed before customizing SmartTest. If Center has not been installed, see the *ASG-Center Installation Guide*.

Installation Requirements and Considerations

Operating Environment

SmartTest has these requirements:

- MVS/ESA or OS/390
- TSO ISPF Version 3.5 through 4.8
- CICS Version 4.1 or CICS/TS 1.1, 1.2, 1.3, and 2.1 (SmartTest-CICS only)
- IMS Version 5.1, 6.1, or 7.1
- VSAM for the AKR
- TSO logon region size of 2048K or larger
- Direct access storage

Note:
See the *ASG-Center Installation Guide* for quantities.

- 3270 type terminals; Models 2, 3, 4, or 5
- INTERSOLV, Inc.'s APS version 2.1 or later - for SmartTest-APS only

Supported Environments

SmartTest supports these environments:

- Languages
  - COBOL II (including releases 3 and 4)
  - COBOL/370
  - COBOL for MVS and VM
  - COBOL for OS/390 and VM
  - CASE-generated COBOL

- High-level Assembler through a.3.0
- OS PL/I Version 2.3
- PL/I for MVS & VM

- Environments
  - TSO
  - BTS
  - DLI
  - CICS Version 4.1
  - CICS Transaction Server 1.1, 1.2, 1.3, and 2.1
  - CICS command-level programs
  - IMS/DC
  - ISPF Dialog Manager
  - HOGAN
  - LANGUAGE environment
- Databases
  - VSAM
  - IMS/DB (DL/1)
  - DB2
  - IDMS/DB
  - SYSTEM 2000
  - DATACOM/DB
  - TOTAL/TIS
- Other
  - CA-Optimizer II (COBOL II)
  - INTERSOLV APS
  - JCL to CLIST conversion
  - JCL PROC expansion
  - COBOL syntax checking
  - Batch system connection facility

Installation Instructions

These are the installation steps in this chapter:

Step	Description
1	"Step 1 - Modifying and Executing CNTL Library Members" on page 9 to customize them for your environment. Modify the SmartTest Default Installation Option parameter file located in ASG.VIACEN*xx*.CNTL as member VIA$PRMP.
2	"Step 2 - Modifying Installed CLIST Libraries" on page 12.
4	"Step 3 - Adding SmartTest Modules to MLPA/PLPA" on page 14.
5	"Step 4 - Preparing ISPF to Invoke SmartTest" on page 15.
6	"Step 5 - Validating SmartTest (SmartTest-TSO Only)" on page 16.
7	"Step 6 - Reviewing User Tables and Exits (SmartTest-TSO Only)" on page 20. See Appendix C, "SmartTest User Tables and Exits," on page 135 for more information.
8	"Step 7 - Testing ISPF Dialog Manager (SmartTest-TSO Only)" on page 20.
9	"Step 8 - Testing Authorized Programs (SmartTest-TSO Only)" on page 20.
10	"Step 9 - Validating IDMS Support by Analyzing an IDMS Program" on page 21.
11	"Step 10 - Validating DB2 Support by Analyzing a DB2 Program" on page 21.
12	"Step 11 - Validating Update Facility (SmartTest-TSO Only)" on page 22.
13	"Step 12 - Operational Considerations (SmartTest-TSO Only)" on page 24.

If you are installing SmartTest for CICS, IMS/DC terminal emulation, Assembler, APS, PL/I, or DB2 Stored Procedures, you are ready to perform the customization and validation steps for those environments.

Step 1 - Modifying and Executing CNTL Library Members

Modifying JCL Parameters for Installation Members

In the CNTL library members described in this chapter, you will need to specify the correct values for these parameters:

Parameter	Description
VIASOFT	Specifies the high-level node where the ESW products are installed.
CENTER	Specifies the second-level node where ESW products are installed. If the ESW dataset names contain more than three nodes, then specify all nodes except the first and the last as CENTER. For example, the dataset name of SYS3.CEN*xx*.NEW.LOADLIB should have VIASOFT=SYS3 and CENTER=CEN*xx*.NEW.
SYSOUT	Specifies the correct SYSOUT character.
SYSDA	Specifies the appropriate UNIT for temporary datasets.
COMPILR	Specifies the COBOL compiler load module name.
COBLIB	Specifies the COBOL compiler run time subroutines.
COBCOMP	Specifies the COBOL compiler load library name.
USERLIB	Specifies the user load library.

Allocating the SmartTest Batch Submit Queue File

VIASBDAM is the JCL to allocate and format the batch execution queue file, VIAQUEUE. The name of the queue must match the name specified in the SMARTEST CLIST. Because VIASBDAM is used as input and output for SmartTest users, special consideration should be given when defining security for this dataset.

Note:

Update authority is required for the queue file.

To allocate the batch submit queue file

1. Review VIASBDAM and make the necessary modifications.
2. Submit the job.

Optional CNTL Members

Modify these CNTL library members, even though they are not executed during the installation process:

Member	Purpose
VIALAB	SmartTest VIAMAIN training program execution JCL.
VIALAB3	SmartTest VIAMAIN3 training laboratory program execution JCL.
VIALAC23	JCL to compile and link the VIAMAIN3 training program.
VIAMEJCL	SmartTest VIAMERGE demonstration program execution JCL.
VIAPASMA	JCL to assemble and link the VIAPASM demonstration program.
VIAPASMJ	SmartTest VIAPASM demonstration program execution JCL.
VIAPCII	COBOL II or later compile procedure. Copy this PROC to a user PROCLIB or embed it in compile JCL members.
VIAPCOBC	JCL to compile and link the VIAPCOB demonstration program.
VIAPCOBJ	JCL to execute the VIAPCOB demonstration program.
VIAPDLGC	JCL to compile and link the VIAPDLGM demonstration program.
VIAPEXIT	JCL to assemble and link-edit the SmartTest user exits and tables.
VIAPPLIC	JCL to compile and link the VIAPPLI demonstration program.
VIAPPLIJ	JCL to execute the VIAPPLI demonstration program.

Note:

Appendix F, "SmartTest CNTL and CLIST Members," on page 153 contains a complete list of all SmartTest CNTL members and their descriptions.

Customizing Installation Options

To override the default SmartTest installation options

Edit the ASG.VIACEN*xx*.CNTL member VIA$PRMP and modify the appropriate options, which are shown in this example:

```
ASMH-Deck=YES; Default=YES
ASMH-Object=NO; Default=NO
*
DB2-Procedure-Plan=VIAPPLAN; Default=VIAPPLAN
*
* IMS Time Stamp Representation U=UTC L=LOCAL
IMS-TSR=L; Default=L
```

This table contains a description of the SmartTest installation options:

Option	Description
ASMH-Deck	Specifies the default value of the DECK option for Analyze Assembler H (IEV90) support. This should duplicate the value in the actual Assembler H installation for your shop. The default is YES.
ASMH-Object	Specifies the default value of the OBJECT option for Analyze Assembler H (IEV90) support. This should duplicate the value in the actual Assembler H installation for your shop. The default is NO.
DB2-Procedure-Plan	Specifies the DB2 plan name to be used by the SmartTest-DB2-SP option to access SYSIBM.SYSPROCEDURES Table definitions. This must be the same name as that specified for the PLAN parameter of the VIAPBIND step during SmartTest customization. The default is VIAPPLAN.
IMS-TSR	Specifies the time stamp extension to the I/O PCB. The control region parameter, TSR=(U/L), specified in DFSPB*xxx* PROCLIB member, controls the representation of the time stamp with respect to local time versus Coordinated Universal Time (UTC). The default value is L.

Note:

See the installation option parameters in the *ASG-Center Installation Guide* for information on changing the installation parameters values.

Step 2 - Modifying Installed CLIST Libraries

If you plan to use SmartTest to test BTS, IMS, or ISPF dialogs, review these CLISTs and ensure that the listed libraries and work datasets are valid for your site. If you do not plan to use SmartTest to test BTS, IMS, or ISPF dialogs, you can skip this step.

Member	Purpose	
VIAPUBTS	Restores BTS system variables and parameters to the site defaults. Default dataset names are provided and should be modified to use site standards, which can be found in existing BTS test JCL or CLISTs. These items have been provided with standard IBM installation defaults:	
	BTS LOADLIB	'BTS.BTSLIB'
	BTS FORMAT	'IMS.FORMAT'
	QIOPCB	TEMPORARY DATASET
	QALTPCB	TEMPORARY DATASET
	QALTRAN	TEMPORARY DATASET
	BTSOUT	TERM
	BTSPUNCH	None
	BTSDEBUG	None
	BTSSNAP	None

Member	Purpose
VIAPUIMS	Restores IMS system variables and parameters to the site defaults. Default dataset names and parameters for DLI and BMP are provided and should be modified to use site standards, which can be found in existing IMS test JCL or CLISTs. These items have been provided with standard IBM installation defaults: IMS RESLIB 'IMS.RESLIB' IMS PROCLIB 'IMS.PROCLIB' DFSVSAMP 'IMS.PROCLIB(DFSVSMDB)' DB2 LOADLIB 'DSN220.DSNLOAD' IMSMON None IEFRDER TEMPORARY DATASET IMS PSB LIB None IMS DBD LIB None IMS ACB LIB None DLI PARMS BUF 8 SPIE 0 TEST 0 EXCPVR 0 RST 0 SRCH 0 MON N SWAP 0 IRLM N BKO N BMP PARMS OPT N SPIE 0 TEST 0 DIRCA 000 CPUTIME 0

Member	Purpose	
VIAPUSPF	Restores ISPF dialog system variables to the site standards.	
	ISPF PROGRAM LOAD LIBRARY	'ISP.SISPLOAD'(REQUIRED)
	ISPF PANEL LIBRARY	None (see Note below)
	ISPF LINK LIBRARY	None (see Note below)
	ISPF TABLE LIBRARY	None (see Note below)
	ISPF MESSAGE LIBRARY	None (see Note below)
	ISPF SKELETON LIBRARY	None (see Note below)
	ISPF LIST DATASET	SYSOUT
	ISPF LOG DATASET	SYSOUT

Note:

For those default libraries listed as None, your existing ISPF libraries are used if they are not supplied. Appendix F, "SmartTest CNTL and CLIST Members," on page 153 contains a complete list of all SmartTest CLIST members and their descriptions.

Step 3 - Adding SmartTest Modules to MLPA/PLPA

These SmartTest load modules are re-entrant and can be linked:

VIAPEMRP	VIAPOCRP	VIAPPGMP	VIAPRESP
VIAPESUP	VIAPODBP	VIAPPRBE	
VIAPFSPP	VIAPPCPP	VIAPPVRP	

Use this statement to add load modules to MLPA/PLPA:

```
AMODE(31),RMODE(ANY)
```

For the names of other modules that are eligible for location in the MLPA or PLPA, see the installation guide(s) for any other ESW product(s) you have installed.

These are some of the advantages of locating these modules in the MLPA or PLPA:

- A reduction of the memory requirement per user.
- An overall decrease in required swap space.
- Performance improvements.

Moving these modules to MLPA/PLPA is optional. ASG recommends that the original ESW load library (from the installation tape) be kept as a staging library so that any required PTFs can be easily applied. The re-entrant modules can be copied to LPA and the non-reentrant modules copied to a separate user library. These steps also require changes to user logons or product allocations. The CNTL library contains two members, VIASLPAJ and VIASLPXJ, which you can use to perform these copy steps.

Note:

Do not use the ISPF 3.3 copy feature to copy these modules because some of them have aliases.

Step 4 - Preparing ISPF to Invoke SmartTest

Note:

Skip this step if the ESW product menu has already been installed.

Use this panel definition card to add a SmartTest option to the ISPF Primary Menu or another dialog menu. This line describes the SmartTest option to the user:

```
%S + ASG-SmartTest - ASG Integrated Testing/Debugging System
```

Use this statement to invoke SmartTest based on the user selection of the letter S:

```
S, 'CMD(%SMARTEST) NEWAPPL(VIAP)'
```

Note:

After updating the ISPF environment, it might be necessary to re-enter ISPF before the facilities are available.

To invoke SmartTest with a CLIST, follow this step:

- Type `TSO VIATEST` on the command line in the ESW primary panel. The VIATEST CLIST invokes the SMARTEST CLIST while specifying the NEWAPPL(VIAP) parameter to set the correct application ID.

To invoke SmartTest as a command

Note:

Complete this step if you are using SmartTest to debug DB2 programs and you received error message ASG2367 when starting a SmartTest-DB2 session.

1 Edit the VIASGBL CLIST member.

2 Change the value from NO to YES in this statement:

```
SET &STCMD  = NO  /*INVOKE ASG-SMARTEST AS COMMAND?
```

Caution! This change disables SmartEdit access from SmartTest. If you are in a SmartTest session creating pseudo code and try to access SmartEdit, you receive the message `ASG0999E INTERNAL ERROR IN REENEDIT`.

Step 5 - Validating SmartTest (SmartTest-TSO Only)

Note:

The dataset names in the validation steps are the default installed names. If you have changed them, use the changed names where the default names are specified.

To test the logon library allocations

1 Start SmartTest by selecting the correct option for your site.

If you installed SmartTest as described in "Step 4 - Preparing ISPF to Invoke SmartTest" on page 15, use the appropriate ISPF menu selection or CLIST to start SmartTest. If you installed the ESW product menu as described in the *ASG-Center Installation Guide*, use the appropriate ISPF menu selection or the CLIST to display the ESW product primary screen. Select Test ▸ Module/Transaction to display the SmartTest primary screen.

If you start SmartTest using the ESW primary screen, the product name displays as ESW - Testing/Debugging.

2 Select Help ▸ About to verify the product releases and levels of SmartTest and Center that are installed. The product names, release numbers, maintenance levels, and the operating system display.

Press PF3/PF15 to return to the SmartTest primary screen.

3 Review and/or modify SmartTest options by selecting Options on the action bar.

a Select Options ▸ Product Parameters. The Options - Product Parameters pop-up displays. Review and/or modify parameter definitions and then press PF3/PF15.

b Select Options ▸ Log/list/punch. The Options - Log/List/Punch Definition pop-up displays. Review and/or modify the log, list, and punch file defaults.

c Enter the job statement information and press PF3/PF15.

This step is recommended before proceeding to ensure valid options for your user ID. See the online help or the *ASG-SmartTest Reference Guide* for more information about using these options.

To allocate an AKR

Note:

If you already created an AKR when validating another ESW product, skip this procedure.

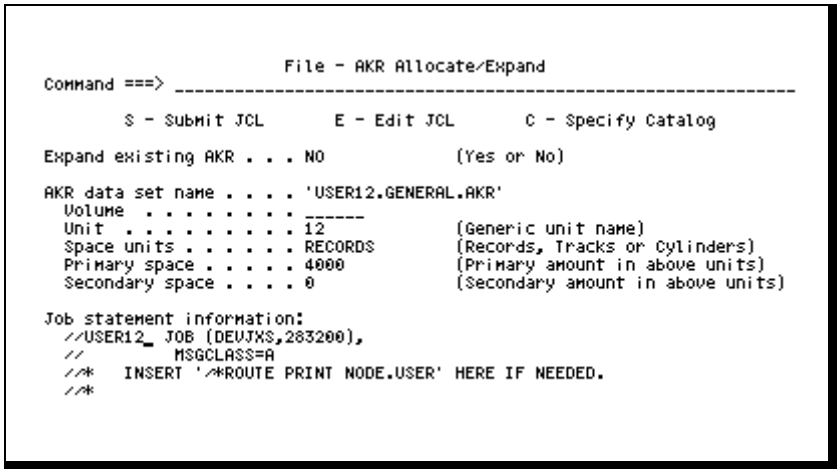
1 Select File ▸ AKR utility. The File - AKR Utility pop-up displays.

2 Complete the File - AKR Utility pop-up by entering the name of a AKR to be allocated. Type `A` in the command input area and press Enter.

The fields on the File - AKR Allocate/Expand pop-up vary depending on the values for the AKR-DSORG-VSAM and SMS parameters in the Center installation options file.

3 On the File - AKR Allocate/Expand pop-up (shown in Figure 6), verify the AKR name.

Figure 6 • File-AKR Allocate/Expand Pop-up



a Type the SMS classes or the Volume and space information for the permanent AKR.

b Type the appropriate Unique parameter for the selected volume if you are allocating a VSAM AKR.

c If this dataset must be cataloged in a user catalog, type `C` on the command line to display the AKR Catalog Information pop-up. Enter the Catalog DSN and Password as required and press PF3/14 to return to the File - AKR Allocate/Expand pop-up.

d Type the job statement information for your site and submit the job by typing `S` in the command input area.

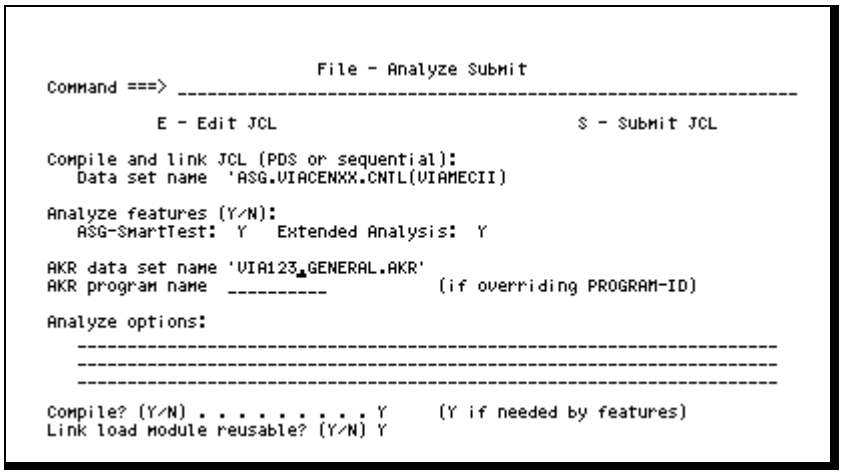
e Wait for the job to finish and verify that the AKR was successfully allocated and initialized.

f Press PF3/PF15 to exit.

To compile and analyze the demonstration programs

1 Select File ▸ Compile/Analyze. Type the appropriate information on the File - Analyze Submit pop-up shown in Figure 7.

Figure 7 • File - Analyze Submit Pop-up



a The name of the dataset that contains the compile and link JCL should be ASG.VIACEN*xx*.CNTL(VIAMECII). Set the ASG-SmartTest field to Y.

b Enter the dataset name of the AKR you created.

c Set the Compile and Link load module reusable fields to Y. Verify the JOB card and routing information by typing `E` in the command input area and pressing Enter. Check the Job information and type `SUB` in the command input area to submit the analyze job. Press PF3/PF15 to exit.

d Verify that the analyze job completed successfully.

2. Change the JCL member name and submit the new analyze job by typing `E` in the command input area and pressing Enter. Check the job information and type `SUB` in the command input area. Press PF3 to exit.

   a. The program being prepared, VIAMAIN, is used in the training classes and also to validate the UPDATE facility.

   b. Verify that the analyze job completed successfully.

3. Set up the VIAMERGE demonstration program by typing `ENV` (the abbreviation for the ENVIRONMENT command) in the command input area and pressing Enter. This displays the Environment Selection Menu.

   a. Verify the AKR and load library names shown.

   b. To select the TSO execution environment, type `1` in the command input area and press Enter. The TSO Session Setup screen displays.

   c. On the TSO Session Setup screen, specify the load module VIAMERGE. The Break on Entry field should have a value of YES.

   d. To view the Convert Batch JCL screen, type `C` in the command input area and press Enter.

   e. On the Convert Batch JCL screen, specify the default JCL library setup when the libraries were loaded from the installation tape and the member name VIAMEJCL.

      Enter your CLIST dataset name for the File Allocation CLIST. Type `C` in the command input area and press Enter to generate a CLIST from the JCL. A short message displays that indicates a successful JCL conversion.

      Press PF3/PF15 to return to the TSO Session Setup screen.

To test the VIAMERGE program

1. Press PF4/PF16 (RUN) to start the test and display the Program View screen.

   The status box displays at the bottom of the screen. This box shows a `BREAK AT START OF TEST SESSION` status message. The current statement line is highlighted and contains chevrons (`>>>>>>`) in the line prefix area.

2. Press PF4/PF16 (RUN) again. A system 0C7 error should occur on the ADD 1 TO END-FILE-COUNT statement. A window is automatically displayed noting that END-FILE-COUNT contains invalid data.

3. Replace the invalid data message with a zero and press the Erase EOF key to clear the rest of the field. Then, press PF4/PF16 once again to resume the test.

The test should now be complete and the status box should display a BREAK ON PROGRAM RETURN status. Press PF4/PF16 again to clear the status box from this test session.

If you do not trap the 0C7, check your compiler options. Make sure you are including the NUMPROC(PFD) option.

4 Type =X in the command input area to exit SmartTest.

Step 6 - Reviewing User Tables and Exits (SmartTest-TSO Only)

To complete the SmartTest installation, it might be necessary to modify user tables and exits, such as the Step Exclusion Table, the Module Exclusion Table, Backtrack Control Table, or the Batch Submit Exit. Review Appendix C, "SmartTest User Tables and Exits," on page 135 for information about modifying these tables and exits.

Step 7 - Testing ISPF Dialog Manager (SmartTest-TSO Only)

To test ISPF Dialog Manager programs, copy VPPISPFT to a user-accessible panel library (DDNAME=ISPPLIB).

Step 8 - Testing Authorized Programs (SmartTest-TSO Only)

SmartTest supports testing of authorized programs. However, because of the security issues involved, SmartTest's main load module is distributed non-authorized. If your site requires authorized program testing, ASG recommends creating a separate, secure load library, then relinking VIAPPRBE authorized.

Note:

VIAPPRBE must not be link authorized if it is to be copied into the MLPA/PLPA.

Step 9 - Validating IDMS Support by Analyzing an IDMS Program

Note:

If your site has tests that involve both IDMS and DB2, execute the VIAPEMOX member from the ESW CNTL library. This addresses IDMS U4005 abends, which can occur during execution.

To validate IDMS support

1. Select File ▸ Analyze/Analyze. The Analyze Submit pop-up displays.
2. Type the dataset name for the compile and link JCL for the IDMS program to be analyzed.
3. Type the AKR dataset name.
4. Set the SmartTest analyze flag to Y.
5. Type `S` in the command input area and press Enter to submit the analyze job.
6. Verify that the analyze job ran successfully.

Step 10 - Validating DB2 Support by Analyzing a DB2 Program

Note:

If you have DB2, verify that ASG.VIACEN*xx*.CNTL(VIASBIND) has been run (see "Step 1 - Modifying and Executing CNTL Library Members" on page 9). This installation step is required for sites having DB2. Failure to complete the DB2 installation step results in erroneous behavior by ESW products.

To validate DB2 support

1. Select File ▸ Compile/Analyze. The Analyze Submit pop-up displays.
2. Enter the dataset name for the compile and link JCL for the DB2 program to be analyzed.
3. Enter the AKR dataset name.

4 Set the SmartTest analyze flag to Y.

5 Type `S` in the command input area and press Enter to submit the analyze job.

6 Verify that the analyze job ran successfully.

Step 11 - Validating Update Facility (SmartTest-TSO Only)

If you completed the SmartTest validation, use this AKR member to verify the Update facility:

```
VIAMAIN  (for TSO)
```

To validate the Update Facility

1 Save pseudo code lines in the VIAMAIN laboratory program:

a On the primary screen, type `TEST` in the command input area and press Enter to display the Test Menu.

b Ensure the PROGRAM/CSECT field or the AKR MEMBER field contains VIAMAIN. Also, ensure the AKR name is correct on your Environment screen.

c Type `VIEW` in the command input area and press Enter to select the Program View screen. The VIAMAIN COBOL program displays.

d Scroll or use the ISPF FIND command to locate the WORKING STORAGE section of the program and insert a new line immediately after the line containing 77 PCTR PIC 99 COMP. This should be the new line:

```
77 NEW-PCTR  PIC 99 COMP.
```

e Scroll or use the ISPF FIND command to locate the program label P900-END. After this label, insert a new line:

```
MOVE ZEROES TO NEW-PCTR.
```

f Type `SAVE PSEUDO` in the command input area and press Enter to save the new pseudo code lines for VIAMAIN in the AKR.

g Type `QUALIFY CANCEL` and press Enter `to` release control of the AKR.

h Type `RETURN` in the command input area and press Enter to return to the Primary screen.

i Type `U` in the command input area and press Enter to proceed to the Update facility.

2 Use the Update facility to automatically update the VIAMAIN COBOL source code with the pseudo code lines from the VIAMAIN AKR member.

a In the Data Set Name field, type:

`'ASG.VIACEN`*`xx`*`.CNTL(VIAMAIN)'`

where *xx* is the Center version number. Press Enter to make the VIAMAIN COBOL source available for editing.

b Type `UPDATE ALL` in the command input area and press Enter to display the Pseudo Code Update screen.

c Ensure the AKR name is correct, specify VIAMAIN as the program name, and press Enter. This automatically adds the pseudo code lines to the VIAMAIN source code and displays a message indicating the number of pseudo code lines inserted.

d Verify the proper placement of the inserted lines. This validates the Update Facility.

To restore the VIAMAIN source to its original state

1 Type `CANCEL` in the command input area and press Enter.

Note:

Typing `SAVE` permanently alters the VIAMAIN source that could interfere with the use of VIAMAIN as a laboratory problem for user training.

2 Type `=X` in the command input area to exit the SmartTest Update Facility.

Step 12 - Operational Considerations (SmartTest-TSO Only)

Batch Session Testing

A SmartTest batch session simulates the SMF Job Wait Time. If this time is exceeded, a user abend (U0978) is issued.

Using the CANCEL command to terminate a Smart Test batch session results in a User 222 abend code. This simulates the operator's cancel command, and permits subsequent steps to execute, or be flushed, depending on the JCL COND= parameters.

3 Customizing CICS

This chapter describes how to customize SmartTest-CICS and contains these sections:

Note:

If you are a first time user, read "Installation Requirements and Considerations" on page 6 before you begin to install and customize SmartTest-CICS. If Center has not been installed, see the *ASG-Center Installation Guide*.

Prerequisites

Before beginning SmartTest-CICS customization, ensure that these prerequisites have been met:

- The SmartTest-CICS option has been downloaded from the product tape.
- Center and SmartTest product installation and customization are completed.
- For CICS/TS, you must have the Optional Source and DSECT libraries. To assemble the SmartTest installation table, you typically need:

Optional Source	DFHOPT01 and DFHOPT02
DSECT	DSECT01 and DSECT02, or just DSECTS

These are the members required from the above libraries:

DFHLDLDA	DFHLDLDM	DFHMPARS
DFHMSET	DFHPGDCD	DFHSMMCA
DFHSMMCM	DFHTSHD	DFHXMIQA
DFHXMIQI	DFHXMXND	

- For CICS/TS 1.3, IBM APAR PQ24735 or PQ34321 is required.

Installation Requirements and Considerations

This section can help you determine the system requirements for installing and activating SmartTest-CICS. These are some of the topics included:

- An overview of SmartTest under MVS and a description of the communication requirements.
- CICS Signon requirements.
- CICS security setup.
- Restrictions and performance considerations for SmartTest-CICS.

The steps to perform the SmartTest-CICS installation are described in "Installation Instructions" on page 41. If you are installing SmartTest-CICS for the first time, please review the information in the rest of this section before proceeding to the installation instructions.

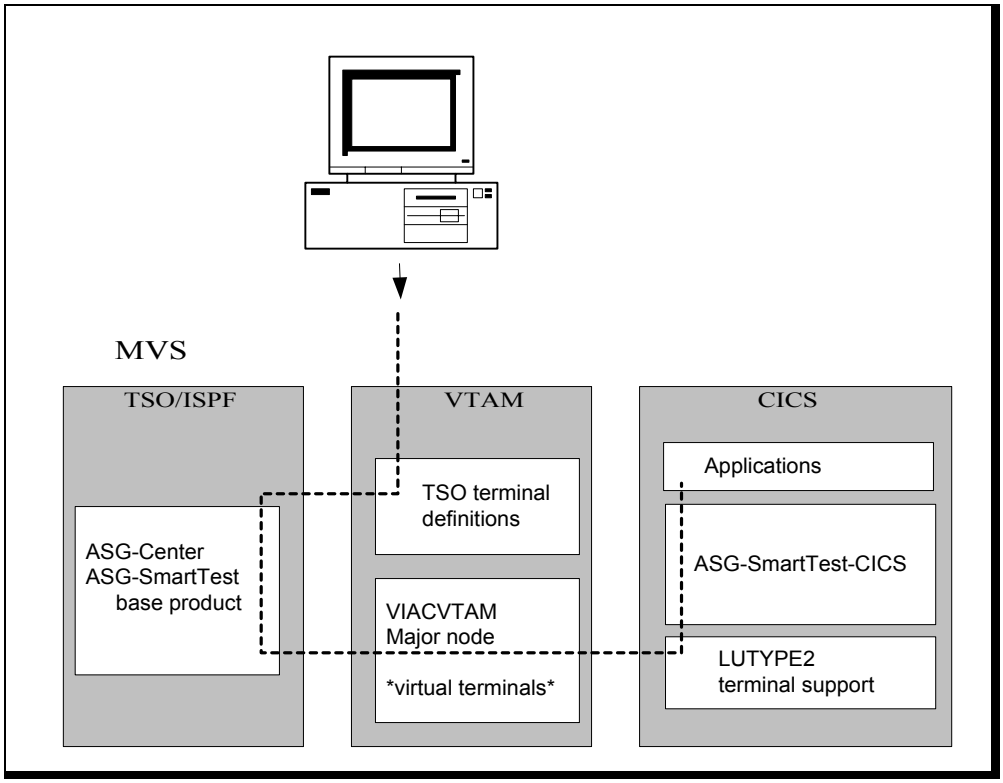
VTAM Definition Requirements

As part of the installation instructions, you or your VTAM systems programmer will add VTAM definitions for SmartTest-CICS. These VTAM definitions are used to enable virtual terminal communication between SmartTest-TSO and SmartTest-CICS.

After installation of SmartTest-CICS, programmers can log on to specified CICS regions from TSO/ISPF using the SmartTest interface. Requests and responses are transmitted through the virtual terminal to and from CICS. CICS application maps and other screens are transmitted through the virtual terminal to the TSO terminal. Input that is destined for CICS is entered through the TSO terminal and sent through the virtual terminal to CICS.

Figure 8 shows how MVS, TSO/ISPF, ACF/VTAM, CICS, and SmartTest work together.

Figure 8 • ASG-SmartTest-CICS with MVS, ACF/VTAM, TSO/ISPF, and CICS



Note:

SmartTest-CICS and SmartTest-TSO do not need to operate in the same VTAM domain.

CICS Terminal Definition Requirements

The virtual terminal support requires compatible terminal definitions in CICS. You or your CICS systems programmer can add or change the CICS terminal definitions as needed during the installation.

Depending on your installation procedures and SmartTest testing needs, you can define these terminals in one of several ways. “Standalone CICS Regions” and “Multi-region Operation (MRO) CICS Regions” outline the terminal definition methods. SmartTest-CICS terminal definitions should be modeled on your current LUTYPE2 SNA terminal definitions. The parameters required for SmartTest-CICS are discussed in "Step 3 - Installing CICS Terminal Definitions" on page 51.

Standalone CICS Regions

- Assemble macro level TCT definitions in a suffixed table.
- You can create and install combinations of TERMINAL definitions for each terminal and TYPETERM definitions using RDO (CEDA).
- You can enable autoinstall using combinations of Autoinstall Model TERMINAL definitions and TYPETERMS (also using CEDA). Additional checking and verification is often provided by the site's DFHZATDX program modifications.

Note:

Although hard-coded TCTs are supported, ASG recommends that you use the Autoinstall method.

Multi-region Operation (MRO) CICS Regions

All of the standalone region styles of terminal support are available in MRO regions. These are additional questions to consider:

- Can users log on directly to all of the MRO regions, or are they required to log on to a TOR?
- How is terminal activity between regions supported? Are terminals defined shippable or are they mirrored using local and remote definitions?

Identifying Your Current Terminal Definition Method

To determine how a particular terminal is defined to CICS

1 Log on to the single-image or terminal-owning region from a VTAM logon screen (not from a session manager).

2 After you have logged on, close and immediately reopen the Transient Data destination that CICS uses to log terminal access messages.

This is the command format:

```
CEMT SET TDQUE(CSSL) CLOSED
```

Closing the Transient Data destination is necessary to force CICS to write the buffer with the most recent log entries. Remember to reopen CSSL immediately.

Note:

If your site reroutes CSSL to another Transient Data destination through indirection or other changes in MSGUSR log handling, you should use your normal method of accessing recent log entries.

3 Scan the log (usually under the name MSGUSR for the active CICS Region) for the terminal ID you just logged on to.

If you see a log entry for your terminal similar to this:

```
DFH3461I X214 CSNE  tttttt  NODE REMTX214 SESSION STARTED
```

you have a TCT macro-generated table definition or a specific RDO (CEDA) terminal definition and can skip the "If You Use Autoinstall" section.

Note:

ASG recommends that you consider changing to the Autoinstall method for defining terminals.

If the log entry is similar to these examples:

```
mm/dd/yy tttttt INSTALL TERMINAL(X214);
DFH5935I INSTALL FOR TERMINAL: X214, NETNAME: REMTX214,
MODEL_NAME: LU20
```

Or

```
DFHZC5966I mm/dd/yy ... CICSTOR1 INSTALL started for TERMINAL
(X214) (Module....
DFHZC6935I .... CICSTOR1 Autoinstall for terminal X214,NETNAME
REMTX214 using model-name LU20 successful.
```

you are using Autoinstall and should read the next section.

If You Use Autoinstall

You need to determine the appropriate TYPETERM and MODEL definitions for defining SmartTest-CICS terminals. To determine the TYPETERM in use, use the CICS log that records terminal access (usually CSSL). For each terminal model that is used for SmartTest-CICS access to CICS, find a representative log entry in the terminal owning region, and determine the MODEL TERMINAL definition in use.

A CSSL log entry for an autoinstalled terminal can have this format:

```
DFH5935I INSTALL FOR TERMINAL: X214, NETNAME:REMTX214,
MODEL_NAME: LU20
```

A CEDA EXPAND GROUP(*) TERMINAL(LU20) request for the MODEL_NAME in the log might show:

```
OBJECT CHARACTERISTICS
CEDA View
     TErminal : LU20
     Group    : REMTCT1
     AUTINSTModel: Only
     AUTINSTName: LU20
TERMINAL IDENTIFIERS
     TYpeterm : LU2000
     Netname  : LU2000
```

Note the TYPETERM in use, and use CEDA to investigate the TYPETERM definitions:

```
CEDA EXPAND GROUP(*) TYPETERM(LU2000)
```

After selecting one of the resulting entries for viewing, page down through the display to locate the DEVICE and IOAREALEN definitions:

```
DEVice       : LUTYPE2
Ioarealen    : 01536, 16384
```

This TYPETERM definition is adequate for SmartTest-CICS terminals that require DEVICE (LUTYPE2) and IOAREALEN 01536, 16384. If you cannot locate an existing TYPETERM definition that meets the SmartTest-CICS requirements, you must create a new TYPETERM. See the TCT macro sample in "Step 3c - Specifying Terminal Definitions Using TCT Macros" on page 51 for a review of the important parameters. You can then create individual TERMINAL definitions (one per VIACVTAM label) specifying the new TYPETERM.

Note:

If the correct DLOGMODs have been chosen for VIACVTAM, CICS attempts to find support for a LUTYPE2 terminal. SmartTest-CICS passes data back and forth between the CICS region and TSO in the form of Terminal I/O Areas (TIOAs). The complexity of the user's request for information from CICS controls how large a given TIOA might be. SmartTest-CICS cannot function properly if the IOAREALEN is too small.

Autoinstall Programs

Your CICS Autoinstall program can implement verification routines or other site-specific requirements. If your Autoinstall processing verifies the NETNAME, ensure that the SmartTest-CICS virtual terminal VTAM labels are accepted. Autoinstall requirements might be another reason to make use of the ACBNAME exit, VIACEX04, documented in Appendix D, "SmartTest-CICS User Exits," on page 141.

If You Code Individual TERMINAL Definitions Using RDO

If your site uses individual terminal definitions under RDO (CEDA), you can use the IBM utility DFHCSDUP to migrate TCT definitions to your CSD file during the installation.

Defining the Signon/Logoff Process

You must inform SmartTest-CICS how your installation logs onto CICS, so that SmartTest-CICS does not interfere with normal or installation-specific logon procedures and screens.

Depending on your installation, entries might be required in the VIACAPPL CNTL library member. Entries can be required for every region that is to be logged on to through SmartTest-CICS. To verify if VIACAPPL entries are required, log on to CICS from a VTAM screen using the command:

```
LOGON xxxxxxx
```

where *xxxxxxx* is the character string used for accessing the CICS region.

Make a note of every screen that displays, every transid or keystroke you enter, and all messages returned to the screen, such as the CICS signon acknowledgement. If your signon sequence is standard (and it is permissible for users to specify the CICS region's actual VTAM APPLID on the SmartTest-CICS Session Setup screen), you might not need to customize the CNTL library member VIACAPPL.

You can use the default signon settings if your CICS logon procedure consists of these actions:

- Sending a VTAM logon to a CICS VTAM APPLID that results in a Good Morning screen.
- Clearing your screen and typing some version of CSSN/CESN.
- Entering an ID and password and receiving a message from CICS (or ACF2) indicating that signon is complete.

If you do not use any of these logon procedures, you must describe the signon process in "Step 4 - Specifying Your CICS Signon/Logoff" on page 53.

Note:

There are special requirements if you have a pseudo-conversational menu that displays at CICS signon. It might also be necessary to specify the SIGNOFF parameter for each Terminal Owning Region (TOR) in VIACAPPL. With this parameter, you can tell SmartTest-CICS to drive a particular transaction at disconnect time. See "Step 4 - Specifying Your CICS Signon/Logoff" on page 53 for more information.

Security Requirements

If your site implements Command Security or Resource Security in your CICS test regions, these considerations apply:

- All SmartTest Transaction definitions (other than the demo transactions) must specify CMDSEC(NO) and RESSEC(NO).
- The system initialization parameters CMDSEC and RESSEC must be set to ALWAYS.
- SmartTest-CICS uses CICS commands and resources to service the testing of user application transactions. If your site has imposed rigid security on your CICS test regions, it may be necessary for you to create specific rules to allow SmartTest to function properly. For example, if you enabled Command Security with the SIT parameter XCMD, you must specify UPDATE authority for the EXITPROGRAM resource for all SmartTest users.
- If you should receive a security error message related to SmartTest-CICS processing, determine the affected resource and the access authority required and arrange to permit the required access to all SmartTest users. For example, a RACF error message relating to insufficient access to the EXITPROGRAM resource would have these components:

```
ICH408I USER(userid ) GROUP(group  ) NAME(lastname, firstname)
   EXITPROGRAM CL(CCICSCMD)
   INSUFFICIENT ACCESS AUTHORITY
   FROM **.* (G)
   ACCESS INTENT(UPDATE )  ACCESS ALLOWED(READ   )
```

  This message indicates that the user required UPDATE access to resource EXITPROGAM (which is included in the CCISSCMD class or the related group class) and that the profile used was **.* (which is a generic profile).

Temporary Storage Requirements

SmartTest-CICS creates Temporary Storage queues as part of its processing. If you have a Temp Storage table defined for your CICS region, it is necessary to ensure that SmartTest-CICS queues will not be defined as Remote or Recoverable. SmartTest-CICS Temp Storage queues begin with the character string VIA and with the character specified in the SmartTest-CICS Installation table parameter HEXPREF. The default character is v (lower case).

CICS and External Security Software

Exclude all load modules that are names of security software from SmartTest-CICS monitoring. You can exclude these names generically using the TYPE=EXCLUDE option found in the VIACEMT1 Environment Table. In addition, you should initialize security software before starting SmartTest-CICS and terminate after SmartTest-CICS.

SmartTest-CICS operates under the security established for the operator and monitored transaction. SmartTest-CICS does not interfere with preestablished CICS security settings.

SmartTest-CICS Internal Security

SmartTest-CICS internal security provides the capability to restrict the use of certain SmartTest-CICS commands, resources, and features. By default, SmartTest-CICS is delivered with no restrictions active; all functions are available to everyone. You can review all protectable options and restrict access by operator id during the installation process.

The SmartTest-CICS diagnostic transactions VIAL and VIAD are intended for use by your Systems/Technical staff. You might want to restrict it.

Transaction VIAI is used to start and stop SmartTest-CICS and transaction VIAH is used to start and stop the Dynamic Call support from a native CICS session. Application programmers normally do not need access to VIAI because its functions are implicit in the first connection from TSO to CICS under SmartTest.

Transactions VIAA, VIAR, and VIAC are internal to SmartTest-CICS. ASG recommends that you do not restrict these transactions.

If Resource Security Level Checking (RSLC=YES) is specified for transactions running under SmartTest-CICS and the PPT entry DFHEMTA has Resource Security Level (RSL) set to zero, some functionality is not available to the SmartTest-CICS users (for example, NEWCOPY and File Services directories). RSL=PUBLIC is recommended for the PPT entry DFHEMTA, as well as for the ESW programs.

Restrictions

RENTPGM Restrictions

If you want to test programs that are link edited RENT and RMODE(ANY), you must specify the SIT override option RENTPGM=NOPROTECT. Otherwise, ERDSA errors can prevent testing.

Compiler Restrictions

As noted in *IBM's Application Programmer's Reference Manual*, these OS/VS COBOL compiler options are not supported by CICS or SmartTest-CICS:

```
COUNT, ENDJOB, FLOW, DYNAM, STATE, SYMDUMP, SYST, and TEST
```

IBM discourages the use of these options because they cause technical storage violations and entire region WAITs.

Data Compression Restrictions

SmartTest-CICS terminals must be excluded from data compression. SmartTest-CICS control blocks are passed in TIOAs and have positional parameters.

BMC Superoptimiser data compression is handled automatically. As of this writing, BMC's Input Suppression feature is not implemented in such a way that it can be recognized and bypassed. Since it is not possible to exclude SmartTest-CICS terminals from BMC Input Suppression, the feature must be turned off.

Products such as PI and McKinney System's Show and Tell can be configured to exclude SmartTest-CICS terminals from data compression. If UFO's compression exit is in use, it might be necessary to disable it for SmartTest-CICS. DATAPACKER and MEGAPACKER also have terminal data compression components. If you use a session manager with a data compression feature enabled, it might be necessary to log on to TSO using a session that does not have the data compression feature enabled. You may want to review an AUXTRACE for any region that SmartTest-CICS will be used in, to investigate any XZC... exits.

Uppercase Translation User Exits (XZCIN)

If your installation has an exit that provides uppercase translation for terminal input, it must include a check for READ TYPE=TEXT requests and ignore the translation (as IBM does) to allow SmartTest-CICS to operate. SmartTest-CICS issues DFHTC READ TYPE=TEXT requests and expects non-translated input.

This is the logic required to check for READ TEXT requests:

```
TM TCTTECS,TCTTENTR Q-TERMINAL READ TYPE=TEXT
BO BYPASSUC         Y-DON'T TRANSLATE
```

Terminal Control (XZCATT) Exit VIACZATT

SmartTest uses the terminal control exit VIACZATT to partially control communication between TSO and CICS during pseudo-conversational transactions and CRTE processing. This exit is automatically enabled when one of these occurs:

- The first user connects from TSO.
- SmartTest-CICS is initialized through the PLTPI.
- The VIAI START transaction.

If the TERMINAL definitions used for SmartTest-CICS terminals specify a transaction ID, you must either specify SmartTest in the startup PLT or enable the VIACZATT exit program during PLT processing using the command:

```
EXEC CICS ENABLE PROGRAM (VIACZATT) EXIT (XZCATT) START
```

CSECT Number Limitation

SmartTest-CICS currently supports the first 285 CSECTs of a load module for symbolic testing. If you want to set breaks in and otherwise closely monitor the 300th CSECT, it is necessary to relink the load module so that the CSECT is one of the first 285.

ESW Dataset Restrictions

It is recommended that all ESW product load modules be kept in one load module dataset. This procedure simplifies installation and subsequent application of maintenance.

Installations that choose to create separate ESW load module datasets must make specific SmartTest-CICS load modules available to SmartTest-TSO and vice versa.

SmartTest-TSO must have access to these SmartTest-CICS specific load modules:

VIACVTAM
VIACEMT1
VIACCPU (if one was created at installation)

SmartTest-CICS requires access to these load modules:

- All modules prefixed with VIAC (with the exception of VIACVTAM and VIACCPU).
- VIAPBKRP (required for Backtrack Facility).

SmartTest-CICS Temporary Storage Queues

SmartTest-CICS Temporary Storage queues cannot be shared between CICS regions.

Language Optimization Limitations

Note:

See Optimization Limitations in the *ASG-SmartTest for COBOL and Assembler User's Guide* or the *ASG-SmartTest PLI User's Guide*.

COBOL Limitations

See "COBOL Compiler Options" on page 129.

CICS/VTAM Connectivity Considerations

- The VIACCPU load module is available to TSO. It accurately reflects the SMF CPU ID for any CPU with a TSO that will be used to connect to SmartTest-CICS.
- There is a VIACVTAM member in SYS1.VTAMLST (or its equivalent) of every CPU with a TSO that will be used to connect to SmartTest-CICS. VIACVTAM has been VARYed active to VTAM.

- The VIACVTAM definitions for each terminal type use the same DLOGMODs specified for the real terminals that are used to access your CICS regions.
- The labels used in the VIACVTAM definitions are acceptable to your autoinstall program in CICS, or are specified in TCT definitions in the NETNAME parameter. The labels can be changed to create better terminal IDs, but the ACBNAMEs should not be changed. If your installation has ACB naming requirements, you might need to use EXIT4 (see Appendix D, "SmartTest-CICS User Exits" on page 141).
- Whether you use TCT macros, CEDA terminal definitions for each terminal, or autoinstall, SmartTest terminal support requires:

CEDA Typeterm	TCT macro
DEVICE(LUTYPE2)	TRMTYPE=LUTYPE2
IOAREALEN(1536,16384)	TIOL=(1536,16384)
USERAREALEN(the largest of any transaction that will be tested)	TCTUAL=(the largest of any transaction that will be tested)

  Otherwise, the definitions should support the same features as your other LUTYPE2 terminals.
- For an MRO installation, use your standard method of defining local and remote terminal support for your TORs and AORs.
- The terminal support in your CICS region can accept the SmartTest VTAM NETNAME created on the CPU being used for TSO.
- The signon procedure for your CICS region is standard:
  - A VTAM logon results in a Good Morning screen.
  - After clearing the screen, type in some version of `CSSN/CESN`.
  - Enter an ID and password, and a message is returned to the screen by CICS (or ACF2) to indicate that signon is complete.

  If this does not accurately describe a signon sequence, you must code TYPE=APPL entries in the VIACAPPL installation table. With these entries, you can tell SmartTest what to expect at signon, so that SmartTest's initialization can proceed properly. For information on APPL entries, see "Step 4 - Specifying Your CICS Signon/Logoff" on page 53. Contact the ASG Service Desk if you need assistance selecting the appropriate parameters.

Command-CICS

The product Command-CICS must be initialized before SmartTest-CICS to avoid storage violation warning messages during a test session.

Data Space Testing

SmartTest-CICS supports Assembler modules that access ESA data spaces during testing. If applications are using data spaces, you must add 0E0 and 01D to the System Recovery Table (SRT) to prevent CICS from abending when it encounters a data space program check that is caught by the CICS ESTAE.

Performance Considerations

SmartTest-CICS has a minimal impact on response time because of these reasons:

- Only those programs being monitored incur overhead. (Since SmartTest-CICS Monitors only the programs you specify, it is best to request monitoring only for the programs you want to test. If you specify monitoring at the transaction level, SmartTest monitors every program associated with the transaction.)

  Note:

  See the *ASG-SmartTest CICS User's Guide* for more information about how to specify which programs are to be monitored. Entries made on the Session Setup screen and directly into the Protection Tables determine what is monitored.

- Program code comprises only a small part of the transaction path length. Most task response time is actually spent waiting for resources or data transmission; SmartTest-CICS is not involved in these processes.

You generally do not need to change any parameters in the CICS System Initialization Table (SIT) to support SmartTest-CICS. However, a high value for the Interval Control Exit (ICV) time results in slow response during periods of light CICS activity.

Also, the Terminal Scan Delay value (ICVTSD) affects response time. A recommended value for the SIT ICVTSD parameter is 100. It is possible that you will need to increase your ICVR (Runaway Task Time) value if transactions to be tested are approaching your current limit without SmartTest-CICS.

Performance Techniques

These are some techniques for improved performance with long-running transactions:

- Consider putting the TSO SmartTest modules in the MLPA/PLPA.

  Note:

  See the *ASG-Center Installation Guide* for a list of MLPA/PLPA candidates.

- Monitor only the PPT modules that are to be tested, rather than all modules for the transaction. (Do not enter the TRAN ID on the Task Protection screen unless you really want to monitor everything running under that TRAN-ID.)
- Review the contents of the Global Task and Program Protection tables. Migrate as many entries as possible to the individual user task and program protection tables.

- If a single module contains multiple CSECTs, consider excluding some of the CSECTs.
- Turning off Storage Protection, Execution Tracking, and Counts provides significant performance improvement.
- Minimize the number of Breakpoints set to only those required.

Installing SmartTest in Different CICS Releases

The SmartTest-CICS Environment Table option T1CIREL (described on page 65) controls SmartTest-CICS' ability to simultaneously support multiple releases of CICS.

Region Names and Configurations Used in Installation Examples

For illustrative purposes, these names are used in the installation examples:

VTAM Applid	SYSIDNT	Description
CICSTEST	CICT	Standalone CICS TS 1.2 region
CICSTOR1	TCS1	CICS 4.1 Terminal Owning Region (TOR)
CICSAOR1	ACS1	CICS TS 1.3 Application Owning Region (AOR)
CICSAOR2	ACS2	CICS TS 1.3 Application Owning Region (AOR)
CICSAOR3	ACS3	A region in which SmartTest-CICS is not installed

User Exits

SmartTest-CICS provides user exits. Use of these exits is optional and most installations will not need to use them. You should review the descriptions to determine if any of the User Exits are needed in your site. See Appendix C, "SmartTest User Tables and Exits," on page 135 for more information.

Exit #	Description
1	Initialization exit used to define global shared storage locations to SmartTest-CICS.
2	Storage Validation exit for overriding default storage protection rules.
3	Removed as of SmartTest-CICS Version 2.1; it is no longer required.
4	Exit to provide VTAM ACB names to SmartTest-CICS.

Exit #	Description
5	Removed as of SmartTest-CICS Version 2.1; it is no longer required.
6	Exit to provide logon message data to CICS at SmartTest-CICS connect time.

Other Vendor Product Precautions

Co-existence With Other Vendor Products

This list of software products should have an entry on the Global Program Specification screen that specifies MONITOR=NO. SmartTest-CICS automatically adds the entries in the VIACTBLS file. This file includes (at a minimum) these products:

Product	Entry
IDMS	IDMSINTC
NATURAL	3NAT
EYEWITNESS	3DAS
UFO	3UFO
DB2	3DSN
TOP SECRET	3TSS

SmartTest-CICS co-exists with these and many other proprietary vendor software products. However, products that violate CICS coding standards can cause unpredictable results during a testing session. ASG makes a reasonable attempt to provide support for such products, but cannot guarantee success.

Co-existence With Other Software Debuggers

If you have another debugging product in the same region with SmartTest-CICS, do not allow it to monitor SmartTest-CICS programs (all ESW programs start with VIA*). It is often best to exclude from the other debugger any events that run on a SmartTest-CICS terminal. Also, it is important that the other debugger is not set up to monitor any part of an application that you are testing under SmartTest-CICS.

CA-PANSOPHIC Requirements

If your installation uses PANEXEC in the CICS environment, the PANEXEC PANEACCS module must be MVS loadable and the SmartTest VIAPEPAN module must be included in the system link list library or in the CICS STEPLIB.

HOGAN Requirements

For COBOL/370, COBOL II, and Assembler support:

- If your shop uses HOGAN only for COBOL/370, COBOL II, and Assembler modules, excluding the module CICSPEM from monitoring significantly improves performance. For information about excluding modules, see the TYPE=EXCLUDE topic in "Step 6b - Using the Type= Entries in VIACEMT1" on page 68.
- To ensure the best performance when the CICSPEM module is added to the Global Program Protection table as MONITOR=YES, users who test COBOL II and/or Assembler modules should either monitor at the program level or add CICSPEM to the User Program Protection table as MONITOR=NO.

SmartTest-CICS Abend Codes

These abend codes apply to SmartTest-CICS only. If you receive one of these abends, contact the ASG Service Desk:

Code	Description
VEXE	The SmartTest Command Level requester module VIACEXEC has encountered a logic error. A message that provides basic information about the request failure may have been written to the CSSL. There is also a Vacillates temp storage record that can be examined (see Appendix G, "Problem Determination," on page 163). Obtain the transaction dump and contact the ASG Service Desk.
VIA1	An internal error occurred while processing a temporary storage control record.
VIA2	An internal error occurred because SmartTest-CICS could not find the SmartTest control blocks on the user TCA chain.
VIA3	An internal error occurred due to unexpected module encountered in the Return processing logic. Expected DFHEIP module.
VIA4	An internal error occurred because SmartTest-CICS was unable to find SmartTest control blocks.
VIA5	An internal error occurred because the SmartTest-CICS monitor VIACEMAN is corrupted.
VIA6	An internal error occurred while processing the VIACTBLS file. Possible installation error or storage corruption by another task.

Code	Description
VIA7	An internal error occurred while processing a File Utility request.
VIA8	An error occurred because SmartTest modules could not be loaded, or they are incompatible with the TSO components. Ensure that the module VIACEMT1 reflects the current release for SmartTest and for the CICS region in which it is running.

Installation Instructions

This section contains a step-by-step description of the tasks you must complete to install SmartTest-CICS. If your site does not currently have SmartTest-CICS installed, review the steps in "New Installation of SmartTest-CICS" on page 41 before you proceed to the individual steps. If you are currently running SmartTest-CICS, review the summary of steps described in "Upgrading from a Previous Release of SmartTest-CICS" on page 42, then complete the steps that correspond to the upgrade process.

New Installation of SmartTest-CICS

If you are installing SmartTest-CICS for the first time, or if you are installing in a different release of CICS, perform these steps:

Steps
"Step 1 - Completing the Installation Worksheet" on page 43.
"Step 2 - Installing VTAM Definitions" on page 48.
"Step 3 - Installing CICS Terminal Definitions" on page 51.
"Step 4 - Specifying Your CICS Signon/Logoff" on page 53.
"Step 5 - Specifying Security" on page 56.
"Step 6 - Specifying SmartTest-CICS Initialization Options" on page 59.
"Step 7 - Specifying SmartTest-CICS Support for Multiple CICS Releases" on page 69.
"Step 8 - Running VIACASMJ" on page 70.
"Step 9 - Defining the SmartTest-CICS File, Programs, and Transactions" on page 71.
"Step 10 - Linking to VIACTSCL at Disconnect" on page 78.

Steps
"Step 11 - Customizing Optional SmartTest-CICS Features" on page 79.
"Step 12 - Enabling Global Remote Monitoring (Optional)" on page 82
"Step 13 - Specifying MRO Transaction Definitions (Optional)" on page 83.
"Step 14 - Specifying Optional Program List Tables - Initialization and Shutdown" on page 88.
"Step 15 - Editing your CICS Startup JCL" on page 88.
"Step 16 - Preparing the SmartTest-CICS Demonstration Programs" on page 89.
"Step 17 – Validating SmartTest-CICS Installation through Native CICS" on page 90.
"Step 18 - Validating SmartTest-CICS" on page 94.

Note:

If you use the optional SmartTest-CICS features PL/I, file services, or perform the necessary customization, the load modules for VIACVTAM, VIACEMT1, and the optional VIACCPU must be available to the TSO connection.

Upgrading from a Previous Release of SmartTest-CICS

If you have previously installed SmartTest-CICS, perform these tasks for the corresponding steps:.

Step	Description
1	Complete the Installation worksheet, using information from your prior installation. See "Step 1 - Completing the Installation Worksheet" on page 43.
2	Verify VIACVTAM in SYS1.VTAMLST and VIACCPU. See "Step 2 - Installing VTAM Definitions" on page 48.
3	Copy the existing TYPE=APPL entries from your current VIACAPPL table into the new VIACAPPL member. See "Step 4 - Specifying Your CICS Signon/Logoff" on page 53.
4	Copy the existing TYPE=OPERATOR entries from your current VIACAUTH member into the new VIACAUTH member. See "Step 5 - Specifying Security" on page 56.
5	Review the SmartTest-CICS initialization options. See "Step 6 - Specifying SmartTest-CICS Initialization Options" on page 59.

Step	Description
6	Run the VIACASMJ job. See "Step 8 - Running VIACASMJ" on page 70.
7	Review the SmartTest-CICS file, programs, and transactions for additional changes. Remember to add PPT definitions for any VIACEM*xx* installation tables you have created. See "Step 9 - Defining the SmartTest-CICS File, Programs, and Transactions" on page 71.
8	Add the link to VIACTSCL to your Autoinstall delete processing program. See "Step 10 - Linking to VIACTSCL at Disconnect" on page 78.
9	Customize the optional SmartTest-CICS features. See "Step 11 - Customizing Optional SmartTest-CICS Features" on page 79.
10	Edit your CICS startup JCL. See "Step 15 - Editing your CICS Startup JCL" on page 88.
11	Prepare the SmartTest-CICS demonstration programs. See "Step 16 - Preparing the SmartTest-CICS Demonstration Programs" on page 89.
12	Validate the installation through native CICS. See "Step 17 – Validating SmartTest-CICS Installation through Native CICS" on page 90.
13	Validate the installation through the SmartTest-CICS TSO connection. See "Step 18 - Validating SmartTest-CICS" on page 94.
14	If you use the optional SmartTest-CICS features PL/I, file services, or temporary storage cleanup perform the necessary customization.
15	The load modules for VIACVTAM, VIACEMT1, and the optional VIACCPU must be available to the TSO connection.

Step 1 - Completing the Installation Worksheet

This worksheet describes the parameters and values that are used in the installation procedures. Determine the appropriate values for your installation and record them for later reference. Space is provided to record different parameter values, if two releases of CICS are in use.

CICS Information

Parameter (default)	Install Value	Description	
	for CICS release____	for CICS release ____	
CICS CICS	_____	_____	High-level qualifier for CICS libraries. (Used in VIACASMJ, VIACJMAP, VIACJASM, VIACJCII, and VIAC------.)
MACLIB MACLIB	_____	_____	CICS MACLIB name. (Usually SDFHMAC) (Used in VIACASMJ, VIACJMAP, VIACJASM, VIACJCII, and VIAC------.)
MTSLIB MTS	_____	_____	CICS MTS (Macro Temporary Store) library. Used in VIACASMJ to ensure that the Installation/Environment table (VIACEMT1 and VIACEM*xx*) and any user exits are assembled with any CICS SMP maintenance in the APPLY state, if the CICS region is running with the MTS.
SRCLIB SOURCE	_____	_____	CICS SOURCE library. (Usually SDFHSRC for CICS 4.1 and DFHOPT01/DFHOPT02 for CICS/TS 1.1 and above.) (Used in VIACASMJ, because some releases of CICS have necessary DSECTS in the SOURCE library.)
CSDLIB USER.CSD	_____	_____	CSD library for RDO definitions. (Used in VIACCSDJ.)
GROUPLST GROUPLST	_____	_____	RDO GROUP LIST specified at CICS startup.

TERMINAL Definition Values

Used in "Step 3 - Installing CICS Terminal Definitions" on page 51 of installation.

Parameter (default)	Install Value	Description	
	for CICS release_____	for CICS release_____	
MOD2TERM none	_____	_____	Model TERMINAL definition for Model 2 terminals.
MOD3TERM none	_____	_____	Model TERMINAL definition for Model 3 terminals.

Parameter (default)	Install Value	Description
MOD4TERM none	_____	_____ Model TERMINAL definition for Model 4 terminals.
MOD5TERM none	_____	_____ Model TERMINAL definition for Model 5 terminals.
CPUID1 none	_____	CPU ID 1 (see member SMFPRM*xx* in SYS1.PARMLIB for field name SID). Used in VIACCPU to identify the CPU ID for the TSO from which SmartTest-CICS will connect to CICS.
CPUID2 none	_____	CPU ID 2 (Used only if there are two CPUs from which TSO users will be logging on to SmartTest-CICS.
CICSMOD2 10	_____	Number of model 2 terminals to be used concurrently for SmartTest-CICS processing.
DLOGMOD2 D4A32782	_____	LUTYPE2 SNA DLOGMOD for Model 2 terminal.
CICSMOD3 5	_____	Number of model 3 terminals to be used concurrently for SmartTest-CICS processing.
DLOGMOD3 D4A32783	_____	LUTYPE2 SNA DLOGMOD for Model 3 terminal.
CICSMOD4 10	_____	Number of model 4 terminals to be used concurrently for SmartTest-CICS processing.
DLOGMOD4 D4A32784	_____	LUTYPE2 SNA DLOGMOD for Model 4 terminal.
CICSMOD5 5	_____	Number of model 5 terminals to be used concurrently for SmartTest-CICS processing.
DLOGMOD5 D4A32785	_____	LUTYPE2 SNA DLOGMOD for Model 5 terminal.
TYPETERM	_____	RDO TYPETERM definition that meets SmartTest requirements. (See "Step 3 - Installing CICS Terminal Definitions" on page 51.)

Parameter (default)	Install Value	Description
GROUP	_____	RDO GROUP in which TYPETERM and TERMINAL definitions are (will be) placed.
TIOAL	_____	Terminal I/O Area Length (specify the largest used in user applications).

COBOL COMPILER Information - COBOL II and Later

Parameter (default)	Install Value	Description
	for CICS release_____	
COBCOMP SYS1.COB2COMP	_____	COBOL II and later compiler library. Used in VIACJCII to compile COBOL II demonstration program.
COBLIB SYS1.COB2LIB	_____	COBOL II and later run-time subroutine library. Used in VIACJCII.
COB2CIC SYS1.COB2CICS	_____	COBOL II and later CICS runtime subroutine library. Used in VIACJCII.
COMPILR IGYCRCTL	_____	COBOL II and later compiler program name. Used in VIACJCII.

Miscellaneous Parameters

Parameter (default)	Install Value	Description
	for CICS release_____	
VIASOFT ASG	_____	High-level qualifier(s) for Center permanent libraries (non-VSAM).
CICSTRN DFHECP1$	_____	CICS COBOL Command-level interface module name for translator. (Used in Center member VIA$PRMS, and in SmartTest-CICS CNTL library members VIACJASM and VIACJCII.)
TRNPARM none	_____	Run-time parameters for the CICS translator program.
ASMBLR IEV90	_____	IBM Assembler to be used (for installation tables and sample Assembler program).
CENTER VIACEN*xx*	_____	Second-level qualifier(s) for the Center permanent libraries (non-VSAM).
LOADLIB USER.LOADLIB	_____	User load library for demonstration programs. (For demonstration programs, usually the ESW installation library.)
PERMUNT SYSDA	_____	Disk unit for permanent SmartTest-CICS libraries.
PERMVOL none	_____	Disk volume for permanent SmartTest-CICS libraries.
SYSDA SYSDA	_____	Disk unit for temporary work space.
SYSOUT '*'	_____	SYSOUT class.
TAPE TAPE	_____	Unit type for tape devices.
VSAMVOL VSAMVOL	_____	Disk volume for SmartTest-CICS VSAM dataset.

Step 2 - Installing VTAM Definitions

The TSO to CICS communication feature requires that VTAM definitions be added. It is important that you complete these steps correctly. Otherwise, SmartTest-CICS users can experience difficulties or problems when using SmartTest-CICS virtual terminals.

The CNTL library members used in this step are VIACVTAM, VIACCPU, and VIACASMJ. VIACEX04 is optional.

Note:

Many of the values you use in this section have related entries on the Installation Worksheet.

Step 2a - Specifying Your Installation's CPU IDs

Edit the VIACCPU CNTL library member to specify your installation's CPU ID(s). VIACCPU is assembled during the installation ("Step 8 - Running VIACASMJ" on page 70). If you are unable to complete the instructions in VIACCPU, check with your MVS systems programmer. If this step is incomplete or incorrect, SmartTest-CICS uses the last character of the SmartTest-TSO user's CPU ID by default.

Step 2b - Specifying VTAM Definitions

Edit the VIACVTAM CNTL library member to specify SmartTest-CICS virtual terminals to VTAM. Each parameter on the VIACVTAM APPL definitions is defined in this table. If you are not sure of the correct VTAM attributes to specify or are unable to complete the instructions in VIACVTAM, check with your VTAM systems programmer.

Parameter	Description
IMPORTANT	SmartTest-CICS looks for and uses the queriable virtual terminal definitions before using the model-specific terminals (MOD 2, MOD 4, etc.). Because the queriable terminals are used first, the queriable terminal definitions in the VIACVTAM member must have a compatible DLOGMODE.

CAUTION	Review the queriable definitions in the VIACVTAM member to make sure the DLOGMODES are compatible.
	These are samples of queriable definitions that reflect a standard IBM DLOGMODE. Use these samples only as a guide. The CICS installation systems programmer and the VTAM systems programmer must review and edit these definitions before use:
	`VIAQA001 APPL ACBNAME=VIAQA001,EAS=1,MODETAB=ISTINCLM,DLOGMOD=D4A32XX3`
	`VIAQA002 APPL ACBNAME=VIAQA002,EAS=1,MODETAB=ISTINCLM,DLOGMOD=D4A32XX3`
	`VIAQA003 APPL ACBNAME=VIAQA003,EAS=1,MODETAB=ISTINCLM,DLOGMOD=D4A32XX3`
	`VIAQA004 APPL ACBNAME=VIAQA004,EAS=1,MODETAB=ISTINCLM,DLOGMOD=D4A32XX3`
	`VIAQA005 APPL ACBNAME=VIAQA005,EAS=1,MODETAB=ISTINCLM,DLOGMOD=D4A32XX3`
	`VIAQA006 APPL ACBNAME=VIAQA006,EAS=1,MODETAB=ISTINCLM,DLOGMOD=D4A32XX3`

Parameter	Description
LABEL	The LABEL of a VTAM APPL definition is the NETNAME presented to CICS. As delivered, the LABEL is the same as the ACBNAME. The LABELs in the SmartTest-CICS VIACVTAM should be changed to reflect the terminal requirements of your installation. (Typically, CICS autoinstall uses the last 4 characters of the LABEL, and not the ACBNAME, to create the terminal ID.) To ensure the uniqueness of terminal IDs when you are using more than one terminal model, you might define labels of VIA2V201, VIA2V202, VIA3V301, and so on.
APPL	For each terminal model, define enough APPL statements to support the largest number of concurrent SmartTest-CICS users. Determine, for instance, the number of SmartTest-CICS users that will be connecting to ANY CICS from a model 2 terminal.
ACBNAME	The ACBNAME is developed according to this formula: Positions 1-3 — VIA (required). Position 4 — 2, 3, 4, 5, or Q — determined dynamically from the terminal model of the TSO user. Position 5 — CPU identifier code. Specify the user-defined substitution character you coded in VIACCPU. Otherwise, you must specify the last character of the TSO user's CPU ID. Positions 6-8 — A sequential number. SmartTest-CICS automatically increments this counter as each additional user connects to CICS. A request is presented to VTAM for the ACBNAME developed in this process. Numbering starts at 001 for each terminal model. If you have ACBNAME coding standards or conventions, you can use SmartTest-CICS's EXIT 4 to customize your SmartTest-CICS VTAM environment. CNTL library member VIACEX04 contains skeleton code for this member, which can be assembled using CNTL library member VIACASMJ. VIACEX04 is listed in Appendix D, "SmartTest-CICS User Exits," on page 141.
EAS	Each APPL statement represents one VTAM session. The required EAS value is 1.
MODETAB	The default MODETAB is ISTINCLM, which is the IBM-supplied logon mode table for MVS. An alternate mode table can be chosen. Determine the MODETABs and DLOGMODs currently in use at your site for terminals (or sessions of a multi-session manager) used for access to CICS. For each terminal model that is used to access CICS through SmartTest-CICS, select a DLOGMOD that conforms to the SmartTest-CICS requirements. SmartTest-CICS virtual terminals must be described as LUTYPE2 SNA, even if the TSO user is accessing through a non-SNA terminal.

Parameter	Description
DLOGMOD	The DLOGMOD parameter specifies the logon mode table entry (contained in MODETAB) that describes the terminal capabilities needed for the SmartTest-CICS virtual terminal. The default entries found in the ISTINCLM MODETAB are compatible with SmartTest-CICS requirements. These default entries describe terminals: 3274 MODEL 1A with MODEL x SCREEN (LOCAL SNA) (LUTYPE2) These MODEENTs (DLOGMOD definitions) in ISTINCLM are compatible with SmartTest-CICS: D329001 and EMU3790 D4A3277*x*, D4A3278*x*, D4C3277*x*, D4C3278*x*, D632780*x*, SN*x*32702 where *x* is the terminal model number. Information about queriable devices is available in the VIACVTAM CNTL library member.

Note:

CICS data and attributes are presented to the TSO terminal after passing through these virtual terminals. The DLOGMODs you choose must support the appropriate attributes.

Step 2c - Activating the SmartTest-CICS VTAM Node

Add VIACVTAM to SYS1.VTAMLST (or its equivalent). Activate the SmartTest-CICS definition to VTAM by entering this command from the MVS console:

```
VARY  NET,ACT,ID=VIACVTAM
```

Note:

If your naming standards preclude the use of VIACVTAM as a major node name, you can define and activate a node name that conforms to your standards.

Also, add the VIACVTAM major node to the standard list of nodes to be activated during VTAM startup.

Step 3 - Installing CICS Terminal Definitions

Depending on your installation procedures and SmartTest Testing needs, you can define these terminals in one of several ways. Normally, you choose only one of these steps to define CICS terminals for SmartTest-CICS.

The optional CNTL library members used are VIACTCT and VIACCSDJ.

If necessary, review the instructions for determining the type of terminal support in use at your site in the CICS Terminal Definition Requirements section.

Step 3a - Specifying TERMINAL Definitions Using AUTOINSTALL

Specify the appropriate TYPETERM and MODEL definitions for SmartTest virtual terminal communication. If you choose to use an existing TYPETERM definition, make sure that it meets the IOAREALEN requirements for SmartTest-CICS.

If needed, enhance your Autoinstall program to accept SmartTest-CICS virtual terminals.

Step 3b - Specifying TERMINAL Definitions Using RDO

If your site uses individual terminal definitions under RDO (CEDA), you can use the IBM utility DFHCSDUP to migrate TCT definitions to your CSD file. After reviewing and modifying the TCT definitions provided in member VIACTCT, you can add TCT processing to the VIACCSDJ member in the CNTL library, which contains JCL to invoke DFHCSDUP. VIACCSDJ can be updated for TCT definitions at this time, and executed during the Define the SmartTest File, Programs, and Transactions step, if appropriate.

Note:

The IBM utility used in VIACCSDJ generates a standard TYPETERM entry (i.e., LU2000), which should be checked for possible duplication in your system. The group containing the SmartTest-compatible LU2000 TYPETERM must be the last one installed in your region. (Also, for MRO sites, the shippable attribute does not migrate, and has to be added, using CEDA ALTER.) If you choose to code the necessary TERMINAL and TYPETERM entries online, be sure that the SmartTest-CICS requirements noted in Step 3c - Specifying Terminal Definitions Using TCT Macros are honored. See the TYPETERM discussion in the Autoinstall Programs topic.

Step 3c - Specifying Terminal Definitions Using TCT Macros

Add TCT entries to your Terminal Control Table. The entries correspond directly with the VTAM APPL definitions you created in the VIACVTAM major node.

To specify terminal definitions

1 Determine the number of TCT entries required using these guidelines:

- Each VTAM APPL statement (in member VIACVTAM) should have a corresponding entry.
- The number of DFHTCT/APPL statements for each terminal model (2, 3, 4, 5, or Q) is the maximum number of concurrent users for that terminal model.
- Be sure that the value you specify in the NETNAME parameter is the same as the LABEL on the corresponding VIACVTAM APPL definition.

2 Modify the DFHTCT parameters in the CNTL library member VIACTCT to reflect installation standards (e.g., TCTUAL).

3 Add the TCT entries contained in VIACTCT to your terminal table.

4 Assemble the TCT using your standard procedures.

This is a sample TCT entry that can be used for your site. If your native CICS terminals support extended data streams, be sure to specify the same features for SmartTest-CICS terminals.

```
V202     DFHTCT TYPE=TERMINAL,                                          -
     GMMSG=YES,          <-- IF NORMAL FOR YOUR SITE        -
     TCTUAL=255,         <-- LARGEST TESTED APPLICATION     -
     RELREQ=(YES,YES),                                      -
     BRACKET=YES,                                           -
     DEFSCRN=(24,80),                                       -
     CHNASSY=YES,                                           -
     BUFFER=1024,                                           -
     RUSIZE=1024,                                           -
     TRMIDNT=V202,                                          -
     NETNAME=VIA2A002,                                      -
     TRMTYPE=LUTYPE2,          <-- REQUIRED                 -
     TRMMODL=2                                              -
     ACCMETH=VTAM                                           -
     TIOAL=(1536,16384),       <-- REQUIRED                 -
     FEATURE=(DCKYBD,UCTRAN),                               -
     ERRATT=INTENSIFY,                                      -
     CLASS=(CONV,VIDEO),                                    -
     TRMSTAT=TRANSCEIVE                                     -
```

Step 4 - Specifying Your CICS Signon/Logoff

This step is performed to describe your CICS signon and logoff to SmartTest-CICS. You should only do this step if one of these conditions applies:

- You have a signon procedure that differs from the one described in "Installation Requirements and Considerations" on page 26.
- You want to specify a logoff transaction to be driven upon disconnection from the SmartTest-CICS session.

The optional CNTL library member used is VIACAPPL.

Edit the VIACAPPL CNTL library member. (This member is automatically copied into the VIACEMT1 Installation/Environment table.)

TYPE=APPL Definitions

VIACAPPL consists entirely of TYPE=APPL definitions in sample form. Edit these entries to describe your signon procedure.

APPL entries in the Environment Table are used to specify certain characteristics of the CICS APPLID(s) referenced in CICS connect (logon) processing. A VIACENV TYPE=APPL entry is required only when one of these conditions is present:

- An alias name is used for the CICS.
- One (or more) of the CICS logon characteristics are non-standard as defined on the previous page.
- A particular signoff transaction must be driven.

This is an example of an APPL entry:

```
VIACENV TYPE=APPL,APPLID=CICSTOR1,ALIAS=TEST,SIGNOFF=CSSF,SREQD=NO
```

Note:

For MRO environments, SmartTest automatically provides access to the AORs if the necessary remote PCT definitions are created in the TOR. SmartTest-CICS dynamically builds a Remote Connections screen from your existing Connection definitions and remote PCT definitions you have added to the TOR for SmartTest.

To make use of the Remote Connections screen, specify the TOR on the SmartTest-CICS session setup screen. Figure 14 on page 86 is an example of the Remote Connections screen.

See the Optional PCT Entries for MRO section of "Step 13 - Specifying MRO Transaction Definitions (Optional)" on page 83 for more MRO information.

These are the available VIACENV TYPE=APPL options:

Option	Description
APPLID	Enter a valid CICS VTAM APPLID that can be connected to by SmartTest-CICS. For MRO, the APPLID should specify the TOR.
ALIAS	Optional. Specify an alternate name by which the CICS APPLID can be referenced. If coded, you must enter the ALIAS on the CICS Session Setup screen in the LOGON APPLID field or the associated TYPE=APPL entry will not be used. You can code user friendly names, or VTAM USSTAB short form names with this option. ALIAS defaults to the APPLID.
SREQD=YES|NO	Specify whether signon is required (using a CSSN/CESN or equivalent transaction) by the CICS APPLID before a user can invoke valid transactions. The default is YES.
GMSG=YES|NO	Specify whether a good morning message displays automatically on the terminal by CICS at logon/connect time. The default is YES. Specify this parameter in accordance with the terminal definitions used for SmartTest-CICS.
BMSG=YES|NO	Specify whether broadcast messages are always sent by CICS to the terminal at logon/connect time. The default is NO. If you have a pseudo-conversational menu that displays automatically at signon, you should specify GMSG=NO,SREQD=NO,BMSG=NO. This instructs SmartTest to reject the CICS VTAM BID until SmartTest has completed its initialization. The message `ASG2733 ASG-SmartTest is Active` displays briefly before your pseudo-conversational menu displays. This means that transactions VIAR, VIAA, and VIAC might run before the user has signed on. If necessary, make entries in your CICS security package to permit these transactions to run before a user is signed on.

Option	Description
SMSG='DFH3504I'	Provides the character string in single quotes that your signon transaction returns after a successful signon. SMSG is used in conjunction with SREQD=YES if the standard CICS signon facilities (CESN/CSSN) are not used. The maximum length is 16 characters. DFH3504I, DFHCE3549, and ACF2's ACFAE139 are handled automatically. It is only necessary to code an SMSG parameter if your signon acknowledgement message is not one of these three.
SIGNOFF=CSSF|CESF|NONE	Identifies the signoff TRANID used for this CICS. CICS/ESA and above can require CESF. If the normal logoff transaction is ACF2's LOGO, it should be specified here.

If the correct signoff transaction ID is not specified, SmartTest-CICS does not submit any transaction to CICS at disconnect time. As a result, the CSSL log might show `LOSTERM ERROR CODE 20` messages when the connection from TSO/ISPF to CICS is ended. Most systems handle the VTAM LOSTERM error with no negative effects on terminal processing. If a SmartTest-CICS terminal cannot be used more than once, this parameter can eliminate error handling problems.

CICS Session Setup Command==> C_____ Logon APPLID **CICSTOR1**	CICS Session Setup Command==> C_____ Logon APPLID **TEST**	CICS Session Setup Command==> C_____ Logon APPLID **CICSAOR1**
#1	#2	#3

If you defined only one TYPE=APPL definition, for example:

```
VIACENV TYPE=APPL,APPLID=CICSTOR2,ALIAS=TEST,SIGNOFF=CSSF,SREQD=NO
```

then entering Logon APPLIDs as illustrated on the SmartTest-CICS Session Setup screens would have these results:

#1 Typing `CICSTOR2` on the SmartTest-CICS Session Setup screen results in a logon to CICSTOR2, assuming CICSTOR2 is an existing VTAM APPLID. Because there is no TYPE=APPL definition that specifies CICSTOR2 as an ALIAS, SmartTest uses the default logon settings. SmartTest does not initialize until one of the three default Signon acknowledgement messages displays (see the SMSG parameter), and there is no transaction submitted to CICS when a user disconnects from the SmartTest-CICS session.

#2 Typing `TEST` on the Session Setup screen initiates a log on to CICSTOR1, because of the TYPE=APPL entry. SmartTest begins initialization immediately after the Good Morning Message displays on the screen (the default for GMSG is YES and SREQD is set to NO, indicating that users are not required to sign on, and no signon acknowledgement is to be expected). When users disconnect after logging on with TEST, SmartTest submits the CSSF transaction to CICS. (SIGNOFF=CSSF).

#3 Typing `CICSAORA` on the Session Setup screen attempts a direct logon to CICSAOR1. If CICSAOR1 has the appropriate terminal support, SmartTest attempts to connect using the default signon values.

Step 5 - Specifying Security

This step specifies operator access security to SmartTest-CICS features. You should only complete this step if you want to restrict the use of certain commands, resources, and features. By default, SmartTest-CICS is delivered with no restrictions active—all functions are available to everyone.

The optional CNTL library member used is VIACAUTH.

Edit the VIACAUTH CNTL library member. This member is automatically included in the VIACEMT1 Installation/Environment table.

VIACAUTH consists entirely of TYPE=OPERATOR definitions in sample form. Add entries to describe which operator IDs are permitted what kind of access. It is recommended that one or two people be set up with unrestricted access authority for troubleshooting purposes. The last entry should be an OPID=ALL type entry. It should be set up with the authorizations you want the majority of your users to have.

Optional OPERATOR entries in the Environment Table are used to define security authorizations for certain commands and tables. If no OPERATOR statements are entered, SmartTest-CICS places no security restrictions on these items; all users have access to all features. These are examples of OPERATOR entries.

```
VIACENV TYPE=OPERATOR,OPID=TSOID1,      LIMITED SETTING  +
      ENTRY=(2,3,7,13,17,21,25,29,33,34,35,36)

VIACENV TYPE=OPERATOR,OPID=TSOID2,      STD SETTING      +
    ENTRY=(2,3,8,14,17,18,19,20,21,22,23,24,25,26,27,28,29,3+
      0,31,32,33,34,35,36,37,38,39)

VIACENV TYPE=OPERATOR,OPID=TSOID3,       STD + GLOBAL TABLES  +
    ENTRY=(2,3,8,10,12,14,16,17,18,19,20,21,22,23,24,25,26,2+
      7,28,29,30,31,32,33,34,35)

VIACENV TYPE=OPERATOR,OPID=ALL,ENTRY=(999)  NO RESTRICTIONS
```

This table contains the syntax for each operand.

Operand	Description
OPID	Enter a valid Operator ID or ALL. The type of operator ID (either TSO user ID or CICS operator ID) to use is specified in the TYPE=OPTIONS section of VIACEMT1, under the OPIDTYP field. ALL is used to define security authorizations for all OPIDs not explicitly entered. If any OPIDs are explicitly entered, an OPID=ALL entry should also be included. ALL must be the last OPID specified. (OPIDTYP=TSO is recommended because it is always available to a SmartTest-CICS session.)
ENTRY	Enter the security level(s) in parentheses, separated by commas. You must enter one or more security values from this list. 1 FORCE command and command parameters allowed 2 AUTO STEP feature available 3 Display memory by absolute address 4 View the DFHSNT module (sign on table) 5 View SmartTest-CICS modules 6 Storage protection can be turned off for the USERID

Operand	Description
Note: Security levels 7 through 16 apply to the Global Protection tables only.	
	7 Allow ADD on Facility Specification screen
	8 Allow ADD/CHANGE/DELETE on Facility Specification screen
	9 Allow ADD on Task Specification screen
	10 Allow ADD/CHANGE/DELETE on Task Specification screen
	11 Allow ADD on Program Specification screen
	12 Allow ADD/CHANGE/DELETE on Program Specification screen
	13 Allow ADD on Storage Specification screen
	14 Allow ADD/CHANGE/DELETE on Storage Specification screen
	15 Allow ADD on Swap Specification screen
	16 Allow ADD/CHANGE/DELETE on Swap Specification screen
	17 Allow file READ access on File Services
	18 Allow file ADD access on File Services
	19 Allow file UPDATE access on File Services
	20 Allow file DELETE access on File Services
	21 Allow temporary storage READ access on File Services
	22 Allow temporary storage ADD access on File Services
	23 Allow temporary storage UPDATE access on File Services
	24 Allow temporary storage DELETE access on File Services
	25 Allow transient data READ access on File Services
	26 Allow transient data ADD access on File Services
	27 Allow transient data UPDATE access on File Services
	28 Allow transient data DELETE access on File Services
	29 Allow DL/I READ access on File Services

Operand	Description
	30 Allow DL/I ADD access on File Services
	31 Allow DL/I UPDATE access on File Services
	32 Allow DL/I DELETE access on File Services
	33 Allow remote monitoring facility access
	34 Allow LIST EIB UPDATE capability
	35 Allow the Backtrack facility to be activated
	36 Allow DB2 READ access on File Services
	37 Allow DB2 ADD access on File Services
	38 Allow DB2 UPDATE access on File Services
	39 Allow DB2 DELETE access on File Services
	999 All above features allowed

Step 6 - Specifying SmartTest-CICS Initialization Options

This step describes the SmartTest-CICS initialization options, which you can use to modify SmartTest-CICS attributes when you start SmartTest-CICS.

You must assemble this table with the same level of CICS maintenance that your CICS region uses. If you apply CICS maintenance, be sure to reassemble your VIACEM*xx* table(s).

The CNTL library member used is VIACEMT1.

Step 6a - Editing VIACEMT1 Environment Options

Edit the VIACEMT1 CNTL library member. This member is assembled in "Step 8 - Running VIACASMJ" on page 70.

VIACEMT1 consists of several types of entries. Edit the TYPE=OPTIONS entry to define SmartTest-CICS system initialization attributes. A description of each TYPE=OPTIONS parameter is provided in this table:

Note:

See "Step 6b - Using the Type= Entries in VIACEMT1" on page 68 for a description of the other valid types of entries.

Entry	Description

TYPE=OPTIONS	Sets these initial parameters and values:
	`VIACENV TYPE=OPTIONS,  INDICATE GENERATION OF TABLES  +`
	`CICSVER=41,  CICS V 41,51,52,53, 61  +`
	`CWA=512,  CWA LENGTH (AS CICS S.I.T.)  +`
	`DB2CTLG=SYSIBM,  SPECIFY DB2 CATLG TABLE PREFIX  +`
	`DESTID=CSSL,  MESSAGE OUTPUT DESTINATION  +`
	`DL1SWAP=NO,  SPECIFY DL/1 PSB SWAPPING=NO  +`
	`DYNMOD#=3,  NON CICS LOADED MODULES TO TEST +`
	`EIB=YES,  ALLOW APPN. TO UPDATE THE EIB  +`
	`EXIT1=NO,  INITIALIZATION EXIT  +`
	`EXIT2=NO,  STORAGE VALIDATION EXIT  +`
	`FACSAVE=NO,  SAVE FACILITY TBL ON VIACTBLS  +`
	`GBLABND=NO,  GLOBAL ABEND INTERCEPT  +`
	`HEXPREF=A5,  HEX PREFIX FOR T/S KEY PREFIX  +`
	`ILBOIGN=NO,  IGNORE STOR VIOL BY COBOL ILBOS +`
	`MAXCALL=10000,  MAX CICS CALLS  (GBL MONITOR) +`
	`MAXFC=512,  MAX FILE I/O  (GBL MONITOR) +`
	`MAXTD=512,  MAX TD  I/O  (GBL MONITOR) +`
	`MAXTS=512,  MAX TS  I/O  (GBL MONITOR) +`
	`NUMBEXT=4194304,  EXTENDED STG LIMIT FOR TASK  +`
	`NUMMAX=204800,  TCA CHAINED STORAGE LIMIT  +`
	`OPIDTYP=CICS,  "CICS" OR "TSO" USED W/VIACAUTH +`
	`PLIMAIN=352, 60/112/352 DECIMAL OFFSET TO PLIMAIN ADCON +`
	`RMFSCOPE=LOCAL,  GLOBAL RMF TESTING SCOPE  +`
	`SHRTBL=25,  SHARED STR GETMAINS BY APPL.  +`
	`SVC=(1,11,19,20,35,47),  WAIT,TIME,OPEN,CLOSE,WTO,STIMER +`
	`SWAPADD=NO,  SWAP PROGRAM, AUTOMATIC PPT ADD +`
	`T1CIREL=NO  VIACEMT1 SUFFIX = CICS RLSE ##`

CICSVER=41|51|52|53|61	Specifies the CICS version and release number (41, 51, 52, 53, or 61). The default is 41. If you need to install SmartTest-CICS in two or more different releases of CICS, see the T1CIREL parameter in this table.
CWA=512	Specifies the length of the Common Work Area (CWA) of the CSA. The length specified must be the same as that specified in the CICS System Initialization Table (SIT). The default is 512.
DB2CTLG=SYSIBM	Specifies the default DB2 catalog prefix name for this CICS region. This option is required only if the File Services DB2 support is being installed. The user can override this value online. The default is SYSIBM.

Entry	Description
DESTID=CSSL	Specifies the name of the Transient Data Destination to route SmartTest print requests and informational messages. The default is the CSSL, the CICS Statistics destination.
DL1SWAP=YES|NO	Specifies whether DL/1 PSB swapping is to be performed. The default is NO.
DYNMOD#=3	Specifies the maximum number of modules to be interactively tested by SmartTest-CICS. These are modules that do not reside in the CICS PPT and are OS loaded.The default is 3.
EIB=YES|NO	Specifies whether to allow monitored programs to modify the Exec Interface Block (EIB). If you specify YES, any attempt by a monitored program to modify the EIB is allowed to take place. No storage violation warning message displays. The default is YES. It is highly recommended that NO be used to protect the EIB, as the EIB is normally READ ONLY. However, since many sites do modify the EIB for various reasons, the default has been set to YES to avoid installation problems.
EXIT1=YES|NO	Specifies whether to invoke the SmartTest-CICS Initialization User Exit during SmartTest-CICS initialization. The default is NO. This exit is coded and assembled only if your applications execute code or access data outside the application's storage chain. It is used to identify static areas where code or data have been permanently placed, and the addresses of such areas remain constant. See Appendix C, "SmartTest User Tables and Exits," on page 135 for more information.
EXIT2=YES|NO	Specifies whether to invoke the SmartTest-CICS Storage Validation User Exit during SmartTest-CICS initialization. The default is NO. This exit is coded and assembled only if your site requires special handling of storage areas acquired outside of the normal CICS task chain. See Appendix C, "SmartTest User Tables and Exits," on page 135 for more information.
FACSAVE=YES|NO	Supports Global Monitoring, although the preferred method for supporting Global Monitoring is to add a TYPE=FACILITY entry to the VIACEM*xx* installation table. This option defaults to NO and is no longer recommended. Do not set this option to YES. This entry is documented only for compatibility with older releases. Contact the ASG Service Desk if you are using Global Monitoring.

Entry	Description
GBLABND=YES|NO	Specifies the Global Abend Intercept option, which causes SmartTest-CICS to trap PCP (Program Control Program) ABEND requests for non-SmartTest-CICS monitored transactions (see Figure 9 on page 65). It produces a summary screen to the SmartTest-CICS print destination and to the terminal that owned the abending transaction. Abends generated internally by DFHPCP are not intercepted; e.g., APCT, APCN, APCR, APCP. In general, any other PCP abend request can be intercepted by SmartTest-CICS. See TYPE=EXABEND in "Step 6b - Using the Type= Entries in VIACEMT1" on page 68 for excluding specific abends from SmartTest-CICS interception. The global intercept feature only applies to facilities that are specified in the Global Facility Protection Table. This option does not enable monitoring for all events. See the *ASG-SmartTest CICS User's Guide* for more information about Global Facilities.
HEXPREF=A5	Specifies the hexadecimal value that you want SmartTest to use as the first position of its temporary storage key to avoid conflicts in your environment. SmartTest can generically delete temporary storage records based on this value.
ILBOIGN=YES|NO	Indicates use of certain COBOL verbs, such as INSPECT and DEBUG, that are not recommended by IBM for CICS due to the use of OS GETMAINs and possible unreclaimed storage. By default, if SmartTest-CICS monitors an application that uses these COBOL verbs, it traps certain storage modifications, by ILBO Routines, as storage violation errors. If you use these verbs, specify YES for this option. If you specify YES, SmartTest-CICS does not provide storage protection for some COBOL subroutines.
MAXCALL=0...10000...999999	Specifies the number of CICS calls that can be issued before a test session stops with a status message of `POSSIBLE CICS CALL LOOP` in the status box. The maximum value you can specify is 999999 and the default is 10000. Specify 0 (zero) to indicate no limit enforcement.
MAXFC=0...512...999999	Specifies the number of File requests that can be issued before a test session stops with a message of `FILE I/O REQUESTS EXCEED MAX`. The maximum value you can specify is 999999 and the default is 512. Specify 0 (zero) to indicate no limit enforcement.
MAXTD=0...512...999999	Specifies the number of Transient Data requests that can be issued before a test session stops with a message `TRANS DATA REQUESTS EXCEED MAX`. The maximum value you can specify is 999999 and the default is 512. Specify 0 (zero) to indicate no limit enforcement.

Entry	Description
MAXTS=0...512...999999	Specifies the number of Temporary Storage requests that can be issued before a test session stops with the message `TEMP STORAGE REQUESTS EXCEED MAX`. The maximum value you can specify is 999999 and the default is 512. Specify 0 (zero) to indicate no limit enforcement.
NUMBEXT=0...4194304... 99999999	Specifies the total amount of extended storage (above the 16MB line) that can be accumulated for the transaction before SmartTest-CICS stops the transaction with the status message `STORAGE ACCUMULATED EXCEEDS MAX`. The value excludes Program Storage, but includes all task-related storage above the 16MB line. The maximum value is 99999999 and the default is 4194304. Specify 0 (zero) to indicate no limit enforcement.
NUMMAX=0... 327680...99999999	Specifies the total storage amount that can be accumulated for the transaction before SmartTest-CICS stops the transaction with the status message `STORAGE ACCUMULATED EXCEEDS MAX`. The value excludes Program Storage, but includes all TCA chained storage and all tas-related storage above the 16MB line. The maximum value is 99999999 and the default is 500K. Specify 0 (zero) to indicate no limit enforcement. The maximum value was increased to accommodate LE/370's large control blocks. The above values (MAXCALL, MAXFC, MAXTD, MAXTS, NUMBEXT, and NUMMAX) can be overridden by a SmartTest-CICS user connected through TSO. You can change the values from option 1 on the Session Setup screen (Transaction Limits and Options) or from the List Limits display. If your site has chosen to enable Global Monitoring (documented in the *ASG-SmartTest CICS User's Guide*), you might want to revise these values upward. Otherwise, native CICS users (not connected through TSO) may have their sessions stopped when these limits are reached.
OPIDTYP=CICS|TSO	Restricts the use of SmartTest-CICS features. Enter the type of operator ID that is specified in CNTL library member VIACAUTH. These are the valid entries: CICS Use CICS if 3-character CICS operator IDs are specified. TSO Use TSO if 7-character TSO user IDs are specified. Use of TSO IDs is recommended. The TSO ID is always available in a SmartTest session, regardless of CICS signon requirements. The default is TSO.
PLIMAIN=60|112|352	Specifies the decimal offset from a PL/I program load point to the PLIMAIN CSECT. This number should be verified with a link edit map of any PL/I program. The default is 352, which corresponds to a hex offset of 160 in the link-edit map.

Entry	Description
RMFSCOPE=LOCAL|ALL	Specifies the testing scope for the Remote Monitoring Facility. LOCAL indicates that this CICS region is not participating in global remote monitoring. ALL indicates that this region is participating with all connected regions in global remote monitoring. The default is LOCAL.
SHRTBL=25	Indicates the number of spare entries provided to SmartTest-CICS to track the usage of CICS shared storage requests by the monitored programs. The default is 25. For SmartTest-CICS to permit transactions to update shared storage that they did not acquire, the transaction that made the original shared storage request must be monitored.
SVC=(1,11,19,20,35,47)	Specifies a list of acceptable SVC numbers to be executed under SmartTest-CICS control. If an SVC is issued that is not specified here, SmartTest-CICS stops the transaction and issues the `ILLEGAL SVC CALL` status message. These are the default SVCs: 1 WAIT 11 TIME 19 OPEN 20 CLOSE 35 WTO 47 STIMER

Entry	Description
SWAPADD=YES|NO	Works in conjunction with the Swap Program feature. If SWAPADD is set to NO, an attempt to swap a program to an undefined alternate program results in an APCT abend. If SWAPADD is set to YES, then SmartTest dynamically defines a temporary PPT entry in the CICS system for the alternate program.
T1CIREL=YES|NO	Provides support for multiple releases of CICS under SmartTest-CICS. Instructs SmartTest-CICS initialization logic to load an alternate environment table that has a defined PPT module name of VIACEM*xx*, where *xx* is the CICS release number (such as VIACEM41 or VIACEM51). When this option is enabled, SmartTest-CICS uses the VIACEMT1 module in CICS only long enough to see that the T1CIREL option is on. SmartTest then determines the release of CICS from the CSA and attempts to load a VIACEM*xx* module. Each suffixed VIACEM*xx* module must have been assembled with the correct MTS and/or MACLIB for the release of CICS and defined to CICS. The SmartTest-CICS CNTL library has prototype VIACEM*xx* source members, with the CICSVER option set appropriately. If you select this option, you must build the additional environment table modules and add PPT definitions for each new suffixed version. The module VIACEMT1 must always exist in the product library because it is loaded by both TSO and CICS. If you specify the T1CIREL option, it does not matter which version of CICS the module named VIACEMT1 is assembled with. The alternate release dependent module, such as VIACEM41, are used within CICS.

Figure 9 • ASRA Abend with GBLABND=YES and the Terminal Protected

```
ASG-SMARTTEST-CICS ERROR INTERCEPT SUMMARY    DATE: 27 JAN
TRAN-ID : VCOB                                          TIME: 15.35.304
TERMINAL: L046        DATA EXCEPTION (0C7)              OPID:

PSW    : 001386A0  AMODE: 24                  INSTRUCTION COUNT:       639

MODULE : VIACCII  +0036A0                      LINK-EDITED DATE: 13 JAN 00
PROGRAM: VIACCOB  +003658

  INSTRUCTION: AP    688(4,R8),1296(2,R12)

REGISTERS:
0-7  001383AC 00138696 50138668 00137C40 0039A6A0 00398410 003984D4 003994D4
8-15 0039A4D4 00139E6A 00135048 00137C08 0039AB44 0039A6A0 001384EC

*** PRESS ENTER TO CANCEL THIS TRANSACTION (NO DUMP) ***
*** PRESS PF1 TO CANCEL THIS TRANSACTION WITH A DUMP ***
```

TSO/ISPF Example

```
VIACEMT1 Installation Module

TYPE=OPTIONS
  CICSVER = unimportant
  OPIDTYP = TSO or CICS, if you are using TYPE=OPERATOR entries to restrict
  the use of SmartTest-CICS features
  T1CIREL = YES  other options are picked up   from VIACEMxx modules
TYPE=APPL,
  APPLID=CICSTEST,ALIAS=TEST
TYPE=APPL,   APPLID=CICSTOR1,ALIAS=CICSTOR1,
 SREQD=NO,GMSG=NO
 (any TYPE=APPL entry needed for any CICS region, no matter  what CICS release)
TYPE=OPERATOR, if required
```

CICSTEST TS 1.2 Example

```
DFHRPL-ASG LOADLIB
  VIACEMT1
  VIACEM41
  VIACEM52
  VIACEM53

PPT-VIACEMT1 (T1CIREL=YES)
    does not matter which release of CICS VIACEMT1 is assembled with when
    T1CIREL=YES, as long as it assembles successfully and is available
    to CICS

  VIACEM41 - OK to be present in PPT and DFHRPL, but won't be used in
             this region.
  VIACEM52 - MUST be present in PPT and DFHRPL. MUST have been
             assembled with the correct MTS/MACLIB
  VIACEM53 - OK to be present in PPT and DFHRPL, but won't be used in
             this region.
```

CICSTOR1 4.1 Example

```
DFHRPL-ASG LOADLIB
  VIACEMT1
  VIACEM41
  VIACEM52
  VIACEM53
PPT-VIACEMT1 (T1CIREL=YES)
  does not matter which release of CICS VIACEMT1 is assembled with when
  T1CIREL=YES, as long as it assembles successfully and is available
  to CICS

VIACEM41 - MUST be present in PPT and DFHRPL. MUST have been assembled
  with the correct MTS/MACLIB
VIACEM52 and VIACEM53
  OK to be present in PPT and DFHRPL, but won't be used in this region.
```

CICSAOR1 TS 1.3 Example

```
DFHRPL-ASG LOADLIB
  VIACEMT1
  VIACEM41
  VIACEM52
  VIACEM53

PPT-VIACEMT1 (T1CIREL=YES)
  does not matter which release of CICS VIACEMT1 is assembled with when
  T1CIREL=YES, as long as it assembles successfully and is available to CICS

VIACEM41 and VIACEM52
          OK to be present in PPT and DFHRPL, but won't be used in this region.
VIACEM53 - MUST be present in PPT and DFHRPL. MUST have been assembled with the
          correct MTS/MACLIB
```

If you need to install SmartTest-CICS in two or more different releases of CICS, specify YES and complete "Step 7 - Specifying SmartTest-CICS Support for Multiple CICS Releases" on page 69.

Step 6b - Using the Type= Entries in VIACEMT1

Edit the VIACEMT1 CNTL library member. This member is assembled in "Step 8 - Running VIACASMJ" on page 70.

VIACEMT1 consists of several types of entries. In addition to the TYPE=OPTIONS entry described above, you can define these SmartTest-CICS TYPE= monitoring options:

Entry	Description
TYPE=APPL	See "Step 4 - Specifying Your CICS Signon/Logoff" on page 53 for more information about the TYPE=APPL option.
TYPE=EXABEND	Use the optional Exclude Abend Intercept entries to prevent interception. SmartTest-CICS does not intercept any abend that is specified in this table.
TYPE=EXCLUDE	Use the TYPE=EXCLUDE entry to exclude any modules that should not be monitored. EXCLUDE entries in this table become effective every time SmartTest is initialized. If a user should delete one of these necessary entries using the online facilities, the entry redisplays the next time SmartTest is started.
TYPE=EXCLSTEP	Specify CSECTs that you want SmartTest to exclude from step requests. This is an application feature only. No values are preshipped or suggested.
TYPE=EXCSECT	Use the optional EXCSECT entries in the Environment table to list CSECT names to be excluded from monitoring. The entries specified represent link-edited CSECTS for any monitored load module. **Caution!** Monitoring for the transaction is suspended until control returns from the excluded CSECT. Enter names of CSECTS that are not to be monitored by SmartTest-CICS in parentheses, separated by commas.
TYPE=EXTASK	Use the TYPE=EXTASK entry to exclude any transactions that should not be monitored. EXTASK entries in this table become effective every time SmartTest is initialized. If you delete one of these necessary entries using the online facilities, the entry redisplays the next time SmartTest is started.

Entry	Description
TYPE=OPERATOR	See "Step 5 - Specifying Security" on page 56 for more information about this entry.
TYPE=TASK	Use for transaction names that are to be added to the online Protection Task table with the MONITOR=YES attribute. These transactions are monitored by SmartTest-CICS. Monitoring at the transaction level is not generally recommended except for the demonstration transactions.

Note:

If you are installing SmartTest-CICS for multiple releases of CICS, additional VIACEM*xx* definitions are required. See the T1CIREL environment table option in "Step 6 - Specifying SmartTest-CICS Initialization Options" on page 59, "Step 7 - Specifying SmartTest-CICS Support for Multiple CICS Releases" on page 69 and "Step 13 - Specifying MRO Transaction Definitions (Optional)" on page 83 for more information.

Step 7 - Specifying SmartTest-CICS Support for Multiple CICS Releases

You can modify the SmartTest-CICS attributes for different releases of CICS using initialization options. Perform only those steps that specify a CICS release in which SmartTest-CICS will be installed. See "Step 6a - Editing VIACEMT1 Environment Options" on page 60 for a description of the parameters contained in each member.

The optional CNTL library members used are VIACEM41, VIACEM51, VIACEM52, VIACEM53, and VIACEM61.

Note:

These tables must be assembled with the same level of CICS maintenance that your CICS region uses. If you apply CICS maintenance, you must reassemble your VIACEM*xx* table(s).

Step 7a - Editing VIACEM61 (CICS/TS 2.1 Only)

Edit the VIACEM61 CNTL library member. Uncomment the step that uses this member name in the VIACASMJ job (“Step 8 - Running VIACASMJ”).

Step 7b - Editing VIACEM53 (CICS/TS 1.3 Only)

Edit the VIACEM53 CNTL library member. Uncomment the step that uses this member name in the VIACASMJ job (“Step 8 - Running VIACASMJ”).

Step 7c - Editing VIACEM52 (CICS/TS 1.2 Only)

Edit the VIACEM52 CNTL library member. Uncomment the step that uses this member name in the VIACASMJ job (“Step 8 - Running VIACASMJ”).

Step 7d - Editing VIACEM51 (CICS/TS 1.1 Only)

Edit the VIACEM51 CNTL library member. Uncomment the step that uses this member name in the VIACASMJ job (“Step 8 - Running VIACASMJ”).

Step 7e - Editing VIACEM41 (CICS 4.1 Only)

Edit the VIACEM41 CNTL library member. Uncomment the step that uses this member name in the VIACASMJ job (“Step 8 - Running VIACASMJ”).

Step 7f - Edit VIACEMT1 (All Releases)

Edit the VIACEMT1 CNTL library member. Change the value on the T1CIREL= parameter to YES.

Step 8 - Running VIACASMJ

When you have finished editing all required CNTL library members in "Step 2 - Installing VTAM Definitions" on page 48 through "Step 7 - Specifying SmartTest-CICS Support for Multiple CICS Releases" on page 69, submit the VIACASMJ job. VIACASMJ creates load modules required by SmartTest-TSO and SmartTest-CICS.

The CNTL library member used is VIACASMJ.

Step 8a - Running the VIACASMJ Job

Edit the VIACASMJ CNTL library member to customize the default parameters. After you have edited the VIACASMJ job with the steps to run and the overrides for the high-level CICS prefixes for your CICS environment, save the edited results.

Step 8b - Checking the Output from the VIACASMJ Job

Check the output from the VIACASMJ job and, if needed, edit and submit VIACASMJ again.

The VIACASMJ job produces a listing and, if incorrect, an error code. If any error occurs when running the VIACASMJ job, an error code greater than 4 is returned.

High-level Assembler may return a condition code 4 and the message `ASMA303W **WARNING** Multiple address resolutions may result from this USING for the statement USING *,DFHEIBR`. This error is acceptable.

Note:

VIACASMJ has steps for the assembly and link-edit of several load modules. Run only the steps you need.

Step 9 - Defining the SmartTest-CICS File, Programs, and Transactions

In this step, you add the SmartTest-CICS resources to your CICS region(s). The resources include Programs, Transactions, and one file. If you are installing the SmartTest-CICS File Services optional feature, see "Step 11 - Customizing Optional SmartTest-CICS Features" on page 79, which documents the additional programs and RCT entry required. This step also documents the additional entries for the optional PL/I feature.

The CNTL library members used are VIACTBLJ and VIACSDJB. VIACPLT, VIACCSDJ, VIACFCT, VIACPCT, and VIACPPT are optional members.

Step 9a - Adding SmartTest-CICS VSAM File Definition

Use CNTL library member VIACTBLJ to define the small VSAM file that holds the SmartTest-CICS Global Protection Tables. This file contains a dozen records, at most. This file records certain program and transaction related entries you create in the Installation table and in subsequent updates to the Global Protection tables. These entries apply to all SmartTest-CICS users in a CICS region. The file is defined with shareoptions 3,3 so that you can use the same file in several regions. However, it is recommended that you define a separate file for each CICS region in which SmartTest-CICS will be used to test applications. Then, if one region has unique Global requirements, the entries are not propagated to other regions. VIACTBLJ uses CNTL library member VIACTBLD to load the file with one dummy record.

Step 9b - Adding SmartTest-CICS Resource Definitions

Use the ESW CNTL library member VIACSDJB (shown in Figure 10 on page 72) to add resource definitions directly to your CSD file.

Change the CICS load library, VIACTBLS file name (from Step 9a - Adding SmartTest-CICS VSAM File Definition), and the CSD file names to reflect your site requirements. Note or change the group name and startup list name.

If you elect to create the SmartTest definitions manually using RDO, do not change any of the parameter values in VIACSDJB and use the IBM default settings for definitions that are not specified in VIACSDJB .

Figure 10 • Sample VIACSDJB

```
//ASG      JOB (ASG),'ASG-SMARTTEST CSD DEFS'                          00010000
//*   INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.                 00020000
//*                                                                    00030000
//*   ****************************************************************
//*   * ASG, INC.          ASG-SMARTTEST                             * 00050000
//*   *                                                              * 00060000
//*   * ADD ASG-SMARTTEST-CICS RESOURCE DEFINITIONS TO YOUR          * 00070000
//*   * CICS SYSTEM DEFINITION (CSD) FILE.                           * 00080000
//*   *                                                              * 00090000
//*   * IF YOU ELECT TO ADD THE DEFINITIONS MANUALLY, USING RDO,     * 00100000
//*   * PLEASE DO NOT CHANGE ANY IBM DEFAULT PARAMETERS IF THE       * 00110000
//*   * CHANGE IS NOT SPECIFIED IN THE DEFINITIONS BELOW.            * 00120000
//*   *                                                              * 00130000
//*   * RUN THIS JOB AFTER YOU HAVE DEFINED YOUR VIACTBLS FILE       * 00140000
//*   * (CNTL LIBRARY MEMBER VIACTBLJ). MODIFY THE CSD DEFINITION    * 00150000
//*   * BELOW TO REFLECT THE FILE NAME YOU HAVE CHOSEN.              * 00160000
//*   *                                                              * 00170000
//*   * CHANGE THE CICS SDFHLOAD HIGH LEVEL QUALIFIER, THE "CSDLIB"  * 00180000
//*   * CSD FILE NAME, THE GROUP NAME, AND THE GROUP LIST            * 00190000
//*   * AT THE END OF THE CSD UPDATE INPUT.  CHANGE AS REQUIRED      * 00200000
//*   * FOR YOUR INSTALLATION.                                       * 00210000
//*   *                                                              * 00220000
//*   * YOU MAY WISH TO DELETE THE LAST STATEMENT THAT ADDS          * 00230000
//*   * THE SMART60 GROUP TO YOUR STARTUP LIST, AND ADD THE          * 00240000
//*   * GROUP ONLINE USING CEDA FOR YOUR INSTALLATION VERIFICATION.  * 00250000
//*   ****************************************************************
//*                                                                    00270000
//VIACSDJB PROC CICS='XXXX',        CICS HI  LEVEL NODES             00280000
//           CICSLIB='SDFHLOAD',   CICS LOW LEVEL LOADLIB / SDFHLOAD  00290000
//            SYSOUT='*',                                             00300000
//            CSDLIB='XXXXXXXX'                                       00310000
//*                                                                   00320000
//CSDUPD   EXEC PGM=DFHCSDUP                                          00330000
//STEPLIB  DD   DISP=SHR,DSN=&CICS..&CICSLIB                          00340000
//DFHCSD   DD   DISP=SHR,DSN=&CSDLIB                                  00350000
//SYSPRINT DD   SYSOUT=&SYSOUT                                        00360000
//SYSIN    DD   DUMMY                                                 00370000
//*                                                                   00380000
//         PEND                                                       00390000
//*                                                                   00400000
//* CREATE CSD DEFINITIONS                                            00410000
//*                                                                   00420000
//*                                                                   00430000
//STEP1    EXEC VIACSDJB,                                             00440000
//         CSDLIB='CSD.FILENAME',                                     00450000
//         CICS='CICSHLQ',                                            00460000
//         CICSLIB='CICS.SDFHLOAD'                                    00470000
//*                                                                   00480000
//CSDUPD.SYSIN    DD  *                                              00490000
  DELETE GROUP(SMART60)                                             00500000
                                                                    00510000
  DEFINE F(VIACTBLS) GROUP(SMART60)                                 00520000
         DSNAME(ASG.VIACENXX.VIACTBLS)                              00530000
         RECORDF(F) A(YES) BR(YES) DEL(YES) UPDATE(YES)             00540000
                                                                    00550000
  DEFINE PROGRAM(VIACEXEC) GROUP(SMART60)                           00560000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                          00570000
         DATALOC(BELOW)                                             00580000
```

```
  DEFINE PROGRAM(VIACEXRE) GROUP(SMART60)                               00590000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00600000
         DATALOC(BELOW)                                                 00610000
  DEFINE PROGRAM(VIACFILE) GROUP(SMART60)                               00620000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00630000
         DATALOC(BELOW)                                                 00640000
  DEFINE PROGRAM(VIACLIST) GROUP(SMART60)                               00650000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00660000
         DATALOC(BELOW)                                                 00670000
  DEFINE PROGRAM(VIACGLUE) GROUP(SMART60)                               00680000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00690000
         DATALOC(BELOW)                                                 00700000
*                                     VIACDB2- DB2 FILE SERVICES        00710000
  DEFINE PROGRAM(VIACDB2)  GROUP(SMART60)                               00720000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00730000
         DATALOC(ANY)                                                   00740000
*                                                                       00750000
*                                     VIACDLI- DLI FILE SERVICES        00760000
  DEFINE PROGRAM(VIACDLI) GROUP(SMART60)                                00770000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               00780000
         DATALOC(BELOW)                                                 00790000
*                                     VIADYSET,VIADYNHK, AND THE        00800000
*                                      ENTRY FOR DFHEPC ARE FOR         00810000
*                                      DYNAMIC CALL SUPPORT             00820000
  DEFINE PROGRAM(VIADYSET) GROUP(SMART60)                               00830000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00840000
         DATALOC(ANY)                                                   00850000
  DEFINE PROGRAM(VIADYNHK) GROUP(SMART60)                               00860000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00870000
         DATALOC(ANY)                                                   00880000
*                                     VIACTSCL IS TEMP STORAGE          00890000
*                                      CLEANUP                          00900000
  DEFINE PROGRAM(VIACTSCL) GROUP(SMART60)                               00910000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00920000
         DATALOC(ANY)                                                   00930000
*                                     VIACRMFT IS THE GLOBAL RMF        00940000
*                                      PROCESSOR                        00950000
  DEFINE PROGRAM(VIACRMFT) GROUP(SMART60)                               00960000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              00970000
         DATALOC(ANY)                                                   00980000
  DEFINE PROGRAM(VIACRMFD) GROUP(SMART60)                               00990000
         LANGUAGE(ASSEMBLER) EXECKEY(CICS)                              01000000
         DATALOC(ANY)                                                   01010000
*                                                                       01020000
  DEFINE PROGRAM(VIACEMAL) GROUP(SMART60)                               01030000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               01040000
         DATALOC(BELOW)                                                 01050000
  DEFINE PROGRAM(VIACEMAN) GROUP(SMART60)                               01060000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               01070000
         DATALOC(BELOW)                                                 01080000
  DEFINE PROGRAM(VIACEMCL) GROUP(SMART60)                               01090000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               01100000
         DATALOC(BELOW)                                                 01110000
  DEFINE PROGRAM(VIACEMFS) GROUP(SMART60)                               01120000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               01130000
         DATALOC(BELOW)                                                 01140000
  DEFINE PROGRAM(VIACEMGC) GROUP(SMART60)                               01150000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               01160000
         DATALOC(BELOW)                                                 01170000
  DEFINE PROGRAM(VIACEMSP) GROUP(SMART60)                               01180000
         LANGUAGE(ASSEMBLER) RESIDENT(NO)                               01190000
         DATALOC(BELOW)                                                 01200000
```

```
DEFINE PROGRAM(VIACEMT1) GROUP(SMART60)                          01210000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01220000
       DATALOC(BELOW)                                            01230000
DEFINE PROGRAM(VIACEM41) GROUP(SMART60)                          01240000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01250000
       DATALOC(BELOW)                                            01260000
DEFINE PROGRAM(VIACEM51) GROUP(SMART60)                          01270000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01280000
       DATALOC(BELOW)                                            01290000
DEFINE PROGRAM(VIACEM52) GROUP(SMART60)                          01300000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01310000
       DATALOC(BELOW)                                            01320000
DEFINE PROGRAM(VIACEM53) GROUP(SMART60)                          01330000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01340000
       DATALOC(BELOW)                                            01350000
DEFINE PROGRAM(VIACEM61) GROUP(SMART60)                          01360000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01370000
       DATALOC(BELOW)                                            01380000
DEFINE PROGRAM(VIACEM62) GROUP(SMART60)                          01390000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01400000
       DATALOC(BELOW)                                            01410000
DEFINE PROGRAM(VIACGKEY) GROUP(SMART60)                          01420000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01430000
       DATALOC(BELOW)                                            01440000
DEFINE PROGRAM(VIACLPGM) GROUP(SMART60)                          01450000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01460000
       DATALOC(BELOW)                                            01470000
DEFINE PROGRAM(VIACRUN) GROUP(SMART60)                           01480000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01490000
       DATALOC(BELOW)                                            01500000
DEFINE PROGRAM(VIACTBLS) GROUP(SMART60)                          01510000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01520000
       DATALOC(BELOW)                                            01530000
DEFINE PROGRAM(VIACUTBL) GROUP(SMART60)                          01540000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01550000
       DATALOC(BELOW)                                            01560000
DEFINE PROGRAM(VIACZATT) GROUP(SMART60)                          01570000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01580000
       DATALOC(BELOW)                                            01590000
DEFINE PROGRAM(VIAPBKRP) GROUP(SMART60)                          01600000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01610000
       DATALOC(BELOW)                                            01620000
                                                                 01630000
DEFINE PROGRAM(VIACPLI) GROUP(SMART60)                           01640000
       LANGUAGE(PLI) RESIDENT(NO)                                01650000
DEFINE PROGRAM(VIACDEMO) GROUP(SMART60)                          01660000
       LANGUAGE(COBOL) RESIDENT(NO)                              01670000
DEFINE PROGRAM(VIACASM) GROUP(SMART60)                           01680000
       LANGUAGE(ASSEMBLER) RESIDENT(NO)                          01690000
DEFINE PROGRAM(VIACCOB) GROUP(SMART60)                           01700000
       LANGUAGE(COBOL) RESIDENT(NO)                              01710000
DEFINE PROGRAM(VIACCOB2) GROUP(SMART60)                          01720000
       LANGUAGE(COBOL) RESIDENT(NO)                              01730000
DEFINE PROGRAM(VIACCOB3) GROUP(SMART60)                          01740000
       LANGUAGE(COBOL) RESIDENT(NO)                              01750000
DEFINE PROGRAM(VIACIVP) GROUP(SMART60)                           01760000
       LANGUAGE(COBOL) RESIDENT(NO)                              01770000
DEFINE MAPSET(VIACMAA) GROUP(SMART60)                            01780000
       RESIDENT(NO)                                              01790000
DEFINE MAPSET(VIACMAP) GROUP(SMART60)                            01800000
       RESIDENT(NO)                                              01810000
                                                                 01820000
```

```
 DEFINE TRANSACTION(VIAA) PROGRAM(VIACRUN) GROUP(SMART60) SPURGE(YES)
       DTIMOUT(0015) ISOLATE(NO) CMDSEC(NO) RESSEC(NO)              01840000
       TASKDATALOC(BELOW) TASKDATAKEY(CICS)                         01850000
 DEFINE TRANSACTION(VIAC) PROGRAM(VIACEXEC) GROUP(SMART60)          01860000
       ISOLATE(NO) CMDSEC(NO) RESSEC(NO)                            01870000
       TASKDATALOC(BELOW) TASKDATAKEY(CICS)                         01880000
 DEFINE TRANSACTION(VIAI) PROGRAM(VIACEXEC) GROUP(SMART60)          01890000
       ISOLATE(NO) CMDSEC(NO) RESSEC(NO)                            01900000
       TASKDATALOC(BELOW) TASKDATAKEY(CICS)                         01910000
 DEFINE TRANSACTION(VIAR) PROGRAM(VIACRUN)  GROUP(SMART60)          01920000
       ISOLATE(NO) CMDSEC(NO) RESSEC(NO)                            01930000
       TASKDATALOC(BELOW) TASKDATAKEY(CICS)                         01940000
 DEFINE TRANSACTION(VIVP) PROGRAM(VIACIVP)  GROUP(SMART60)          01950000
       ISOLATE(NO) TASKDATALOC(ANY) TASKDATAKEY(CICS)               01960000
       CMDSEC(NO) RESSEC(NO)                                        01970000
*                                                                   01980000
*                       VIAL IS FOR NATIVE CICS MEMORY DISPLAY      01990000
*                                                                   02000000
 DEFINE TRANSACTION(VIAL) PROGRAM(VIACLIST) GROUP(SMART60)          02010000
       ISOLATE(NO) CMDSEC(NO) RESSEC(NO)                            02020000
       TASKDATALOC(BELOW) TASKDATAKEY(CICS)                         02030000
*                                                                   02040000
*                        VIAH IS FOR DYNAMIC CALL SUPPORT           02050000
*                                                                   02060000
 DEFINE TRANSACTION(VIAH) PROGRAM(VIADYSET) GROUP(SMART60)          02070000
       ISOLATE(NO) TASKDATALOC(ANY) TASKDATAKEY(CICS)               02080000
       CMDSEC(NO) RESSEC(NO)                                        02090000
*                                                                   02100000
*                        VIAD IS FOR GLOBAL RMF MANAGEMENT          02110000
*                                                                   02120000
 DEFINE TRANSACTION(VIAD) PROGRAM(VIACRMFD) GROUP(SMART60)          02130000
       ISOLATE(NO) TASKDATALOC(ANY) TASKDATAKEY(CICS)               02140000
       CMDSEC(NO) RESSEC(NO)                                        02150000
*                                                                   02160000
*                       THE FOLLOWING TRANSIDS ARE FOR DEMOS        02170000
*                                                                   02180000
 DEFINE TRANSACTION(SMAR) PROGRAM(VIACDEMO) GROUP(SMART60)          02190000
       ISOLATE(NO)                                                  02200000
 DEFINE TRANSACTION(VCOB) PROGRAM(VIACCOB)  GROUP(SMART60)          02210000
       ISOLATE(NO)                                                  02220000
 DEFINE TRANSACTION(VCO3) PROGRAM(VIACCOB3) GROUP(SMART60)          02230000
       ISOLATE(NO)                                                  02240000
 DEFINE TRANSACTION(VASM) PROGRAM(VIACASM)  GROUP(SMART60)          02250000
       ISOLATE(NO)                                                  02260000
 DEFINE TRANSACTION(VPLI) PROGRAM(VIACPLI)  GROUP(SMART60)          02270000
       ISOLATE(NO)                                                  02280000
                                                                    02290000
 ADD   GROUP(SMART60) LIST(SMARLIST)                                02300000
/*                                                                  02310000
//*                                                                 02320000
```

Note:

If you are upgrading from a prior release of SmartTest-CICS, review your existing PPT or CEDA GROUP definitions carefully to ensure that any new programs are defined and have the correct options.

If you are not installing the CICS File Services option, VIACDLI and VIACDB2 are not required. If you are not installing the CICS PL/I option, the VIACPLI demonstration program is not required.

For CICS Releases Before CICS/ESA 3.2

Note:

Releases prior to CICS 4.1 are no longer supported.

To add resource definitions

1 Add the FCT entry contained in the CNTL member VIACFCT to your File Control Table and assemble the FCT using your standard procedures. Review the DSNAME parameter to ensure it matches your installation name.

Figure 11 shows the FCT entry required for SmartTest-CICS.

Figure 11 • FCT Entry Required for ASG-SmartTest-CICS

```
*********************************************************************
* ASG,INC.        ASG-SmartTest-CICS Rx.x                MMM, YYYY  *
*                                                                   *
* ASG-SMARTTEST FCT ENTRY FOR PROTECTION TABLES FILE (MONITORING     *
* RULES)                                                            *
********************************************************************* *
*
VIACTBLS DFHFCT TYPE=DATASET,                                        -
               DATASET=VIACTBLS,                                     -
               DSNAME=ASG.VIACENxx.VIACTBLS,                         -
               ACCMETH=VSAM,                                         -
               RSL=PUBLIC,                                           -
               LSRPOOL=NONE,                                         -
               SERVREQ=(ADD,UPDATE,BROWSE,DELETE,READ),              -
               FILSTAT=(ENABLED,CLOSED),                             -
               DISP=SHR,                                             -
               RECFORM=(FIXED,BLOCKED),                              -
               STRNO=2
```

2 Add the PPT definitions to your Processing Program Table. Assemble the PPT using your standard procedures. Use CNTL library member VIACCSDJ to add the necessary entries to your CSD file, or use RDO (CEDA).

The member VIACPPT is formatted to accommodate the assembler step of VIACCSDJ and not as a COPY member. To use VIACPPT as a COPY member, remove the Assembler header and trailer cards.

3 Add the PCT entries to your Program Control Table and assemble the PCT using:

- The CNTL library member VIACCSDJ to add the entries to your CSD file.
- RDO.

These PCT entries are contained in the CNTL member VIACPCT.

Note:

If you are upgrading from a prior release of SmartTest-CICS, review your existing PCT or CEDA GROUP definitions carefully to be sure that all necessary transactions are defined and have the correct options. For example, the transaction VIAA was new in SmartTest-CICS Version 3.1, and contained DTIMOUT requirements.

VIACPCT is formatted to accommodate the Assembler step of VIACCSDJ and not as a COPY member. To use VIACPCT as a COPY member, remove the Assembler header and trailer cards.

Step 9c - Adding COBOL Dynamic Call Support

Previous versions of SmartTest-CICS supported testing of programs dynamically CALLed by COBOL II programs by monitoring the calling program or the entire transaction. The Dynamic Call Support feature improves the performance of SmartTest-CICS sessions by allowing you to test desired programs without monitoring at the transaction level and without monitoring the COBOL II calling program. This also reduces resource use, particularly for Temporary Storage Queues.

To install the Dynamic Call Support feature

1 Ensure that this statement is included in the SIT override parameters:

```
RENTPGM=NOPROTECT
```

Note:

This statement is required for SmartTest and may already have been added to the SIT override parameters.

2 Ensure that these definitions are in the VIACSDJB CNTL library member:

```
PROGRAM(VIADYSET) with DATALOCATION ANY and EXECKey CICS
PROGRAM(VIADYNHK) with DATALOCATION ANY and EXECKey CICS
TRANSACTION(VIAH) with PROGRAM VIADYSET, TASKDATLoc ANY, TASKDATAKey
CICS
```

Note:

If you already have a SMARTTEST group defined, these definitions should be added.

3 Optional installation step: Add VIADYSET to the PLTPI list at any point and reassemble the PLTPI to enable Dynamic Call Support at initialization.

To enable the Dynamic Call Support feature, perform a cold start of the CICS region.

Note:

Do not add VIADYSET to your PLTs until you have successfully used the VIAH transaction. See "Step 18 - Validating SmartTest-CICS" on page 94.

Step 10 - Linking to VIACTSCL at Disconnect

SmartTest-CICS allocates work areas from CICS Shared Storage that exist for the duration of the testing session. To free this storage and to clean up the Temporary Storage Queues created for each user, you must enable the cleanup program from your terminal autoinstall exit program at Delete. When the SmartTest signon table fills up, subsequent users are not able to use SmartTest. This program also clears out the terminal entry from the SmartTest signon table.

To link to VIACTSCL

1. Modify your existing autoinstall program or DFHZATDX from the CICS.SDFHSAMP library for functions DELETE_CODE, DELETE_SHIPPED_TERM, DELETE_SHIPPED_RSE, DELETE_VIRTUAL_TERM, and CONSOLE_DELETE_TERM by inserting this statement:

```
EXEC CICS LINK PROGRAM('VIACTSCL')                    *
COMMAREA(DELETE_EXIT_COMMAREA)
```

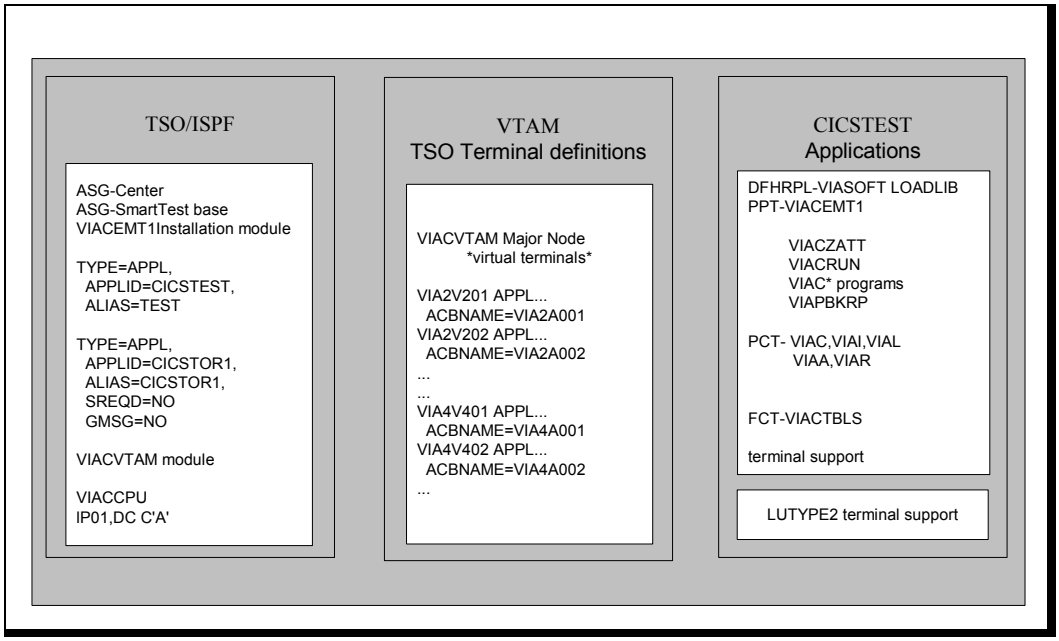
2. Change DELETE_EXIT_COMMAREA to the name of your COMMAREA in your autoinstall exit. DELETE_EXIT_COMMAREA is the COMMAREA that is passed to DFHZATDX by CICS, and is defined by macro DFHTCUDS.

Note:

See the *IBM CICS Customization Manual* for more information about writing a terminal autoinstall program.

You now have a complete installation for a single CICS region. Figure 12 shows the components.

Figure 12 • Components of CICS Standalone Region



Step 11 - Customizing Optional SmartTest-CICS Features

Enabling Dynamic Call Support

Start SmartTest in your CICS region and run the VIAH ON transaction in the same CICS region to install the Dynamic hook in that CICS region. If you successfully installed the option, you receive this message on your CICS screen:

```
ASG-SMARTTEST DYNAMIC PROGRAM HOOK IS NOW ACTIVE
```

Note:

You should only run this command one time in each CICS region to support all users.

Ending Dynamic Call Support

To end dynamic call support, run the VIAH OFF transaction. You should receive this message:

```
ASG-SMARTTEST DYNAMIC PROGRAM HOOK IS NOW INACTIVE
```

The VIAH OFF command stops the Dynamic Call Support for all users in the CICS region. If you added VIADYSET to the PLTSD, the Dynamic Call hook is removed when CICS is shut down.

PL/I Testing Support

The majority of the PL/I support is contained in Center and SmartTest product modules. For CICS, the only additions are CICS resource definitions for the demonstration transaction VPLI and program VIACPLI. VPLI uses the same map as the COBOL and Assembler demonstration programs - VIACMAP, which is assembled in "Step 16 - Preparing the SmartTest-CICS Demonstration Programs" on page 89.

Extended File Services for DB2 and DLI

SmartTest-CICS does not circumvent your existing file security. You can permit or deny the use of any of the File Services features by coding TYPE=OPERATOR entries, as described in "Step 5 - Specifying Security" on page 56.

The programs VIACDLI (for DL/1 File Services) and VIACDB2 (for DB2 File Services) must be added to your PPT definitions in CICS. For DL/1, program VIACDLI is all that is required to implement file services.

For DB2, modify and execute the VIACJBND member before you use the DB2 file utility. This member is used to bind the appropriate DBRM member to the associated plan. You must also complete the steps described in “VIACJBND” , "VIACJBND performs the BIND and grants PUBLIC access to the plan." on page 81, and "VIACRCT" on page 82.

Note:

You must run VIASBIND prior to running VIACJBND. See the *ASG-Center Installation Guide* for more information about VIASBIND.

VIACJBND

The JCL to bind the appropriate DBRM member (VIACDB13, VIACDB21, VIACDB22, VIACDB23 CNTL members) to the plan defined to the DB2 utility.

Note:

To submit this JCL, you must have EXECUTE BIND authority. Also, do not use the same plan name specified in the Center installation option DB2-Plan.

Be sure to specify these parameters correctly.

Parameter	Instruction
DB2LIB='DB2.LOADLIB'	Specifies the DSN associated with the PROC variable DB2LIB.
DBRM='23'	Specifies the suffix associated with the PROC variable DBRM. The valid suffixes are 13, 21, 22, and 23, and the default value is 23. Specify this same value on the MEMBER(VIACDB*??)* parameter in VIACJBND.
VIASOFT='ASG'	Specify the high-level node for ESW product datasets.
CENTER='VIACEN*xx*'	Specify the middle node for ESW product datasets.

These parameters are under //BIND.SYSTSIN DD * and //BIND SYSIN DD * statements at the bottom of the JCL.

Paremeter	Instruction
DSN SYSTEM(*????*)	Specify the correct DB2 system ID.
BIND PLAN(*xxxxxxxx*)	Specify the correct plan name. The BIND PLAN name must match the PLAN parameter in the RCT.
MEMBER(VIACDB*??*)	Replace *??* with the suffix that matches your DBRM CNTL member (VIACDB13, VIACDB21, VIACDB22, or VIACDB23). Valid suffixes for *xx* are 13, 21, 22, and 23. This suffix must also match the suffix on your DBRM= parameter.
LIB('*????*')	Specifies the name of the DB2 Load library where DSNTIAD is located.

Note:

VIACJBND performs the BIND and grants PUBLIC access to the plan.

VIACRCT

The RCT entry is used to communicate with the attachment facility. Modify the PLAN= parameter to match the BIND PLAN parameter in the CNTL member VIACJBND. Add this entry to your existing RCT for CICS.

Note:

Your DB2 administrator should review the security on any DB2 Plans or Tables that you want to access through SmartTest-CICS File Services. SmartTest does not override DB2 security.

Step 12 - Enabling Global Remote Monitoring (Optional)

The SmartTest-CICS remote monitoring has been extended to monitor remote and asynchronous transactions that run in any CICS region that is connected to the current SmartTest region and has Monitor set to YES on the Remote Connections screen. This feature uses a SmartTest Global remote monitoring facility (RMF) table (which is allocated and enabled in MVS CSA) and Global RMF testing to be activated in each participating CICS region.

To enable global RMF

1. Allocate a global RMF table.

   Run the VIAPRMFT CNTL member, which contains JCL to allocate the global storage for the RMF table. You must run this JCL before global monitoring can be used. VIAPRMFT only needs to be run once after each IPL.

   Before running this job for the first time, copy program VIAPRMFT from the installation load library to an APF authorized library. Update the JCL AUTHLIB parameter to reference the selected library or, if the library is in your system LINKLIST, remove (comment out) the STEPLIB in the JCL.

2. Change RMFSCOPE= in the VIACEMT1 CNTL member to activate global RMF in the participating CICS regions. Specifying RMFSCOPE=ALL activates global RMF in the CICS region. RMFSCOPE=LOCAL indicates that the region is not a participant in global remote monitoring. See "Step 6 - Specifying SmartTest-CICS Initialization Options" on page 59 for more information about VIACEMT1 parameters.

Managing Global Remote Monitoring with the VIAD Transaction

Use the VIAD transaction in a native CICS session to start and stop global remote monitoring in the local CICS region and at the global level. These are the valid transactions:

Transaction	Description
VIAD START	Initializes and activates global remote monitoring in the local CICS region. SmartTest-CICS must be initialized before running this transaction.
VIAD STOP	Deactivates global remote monitoring in the local CICS region.
VIAD RESET ENABLE	Initialize the global RMF table entries and sets the status of the table to active.
VIAD RESET DISABLE	Clears all entries in the global RMF table and sets the status of the table to inactive.
VIAD STATUS	Displays the current status of global remote monitoring.

Step 13 - Specifying MRO Transaction Definitions (Optional)

SmartTest-CICS provides a simple method to establish a testing environment in your existing AORs. Remote PCT definitions are added to the TOR for each AOR in which SmartTest-CICS is to be used. For example, using the sample region names in "Region Names and Configurations Used in Installation Examples" on page 38 assumes that these definitions have been created.

These examples are illustrated in Figure 13 on page 85.

CICSTOR1 - a CICS 4.1 TOR

```
Connection Definitions:ACS1, associated with NETNAME CICSAOR1
           ACS2, associated with NETNAME CICSAOR2
           ACS3, associated with NETNAME CICSARO3

PPT Definitions:VIACEM41 (installation table configured for CICS 4.1)
           VIACEMT1 (installation table with T1CIREL=YES)
           VIACZATT attach exit
           VIACRUN connection and RUN TRANSID support program
           VIAC* and VIAPBKRP load modules
PCT Definitions:VIAA and VIAR transactions associated with VIACRUN
           VIAC test function support transaction, associated with
           VIACEXEC
           VIAI internal operations support transaction associated with
           VIACEXEC
           VIAL diagnostic transaction, associated with VIACLIST
```

```
REMOTE PCT Definitions:VI1C, REMOTESystem=AOR1, REMOTEname=VIAC
           VI2C, REMOTESystem=AOR2, REMOTEname=VIAC

FCT Definition:VIACTBLS - Global protection table entries

LUTYPE2 SNA terminal support
```

CICSAOR1 - a CICS TS 1.3 AOR

```
Connection Definition:  TCS1, associated with NETNAME CICSTOR1

PPT Definitions:VIACEM53 (installation table configured for CICS TS 1.3)
           VIACEMT1 (installation table with T1CIREL=YES)
           VIACZATT attach exit
           VIACRUN connection and RUN TRANSID support program
           VIAC* and VIAPBKRP load modules

PCT Definitions:VIAA and VIAR transactions associated with VIACRUN
           VIAC test function support transaction, associated with
           VIACEXEC
           VIAI internal operations support transaction, associated with
           VIACEXEC
           VIAL diagnostic transaction, associated with VIACLIST

FCT Definition:VIACTBLS - Global protection table entries
```

CICSAOR2 - a CICS TS 1.3 AOR

```
Connection Definition: TCS1, associated with NETNAME CICSTOR1

PPT Definitions:VIACEM53 (installation table configured for CICS TS 1.3)
           VIACEMT1 (installation table with T1CIREL=YES)
           VIACZATT attach exit
           VIACRUN connection and RUN TRANSID support program
           VIAC* and VIAPBKRP load modules

PCT Definitions:same as in AOR1

FCT Definition:same as in AOR1
```

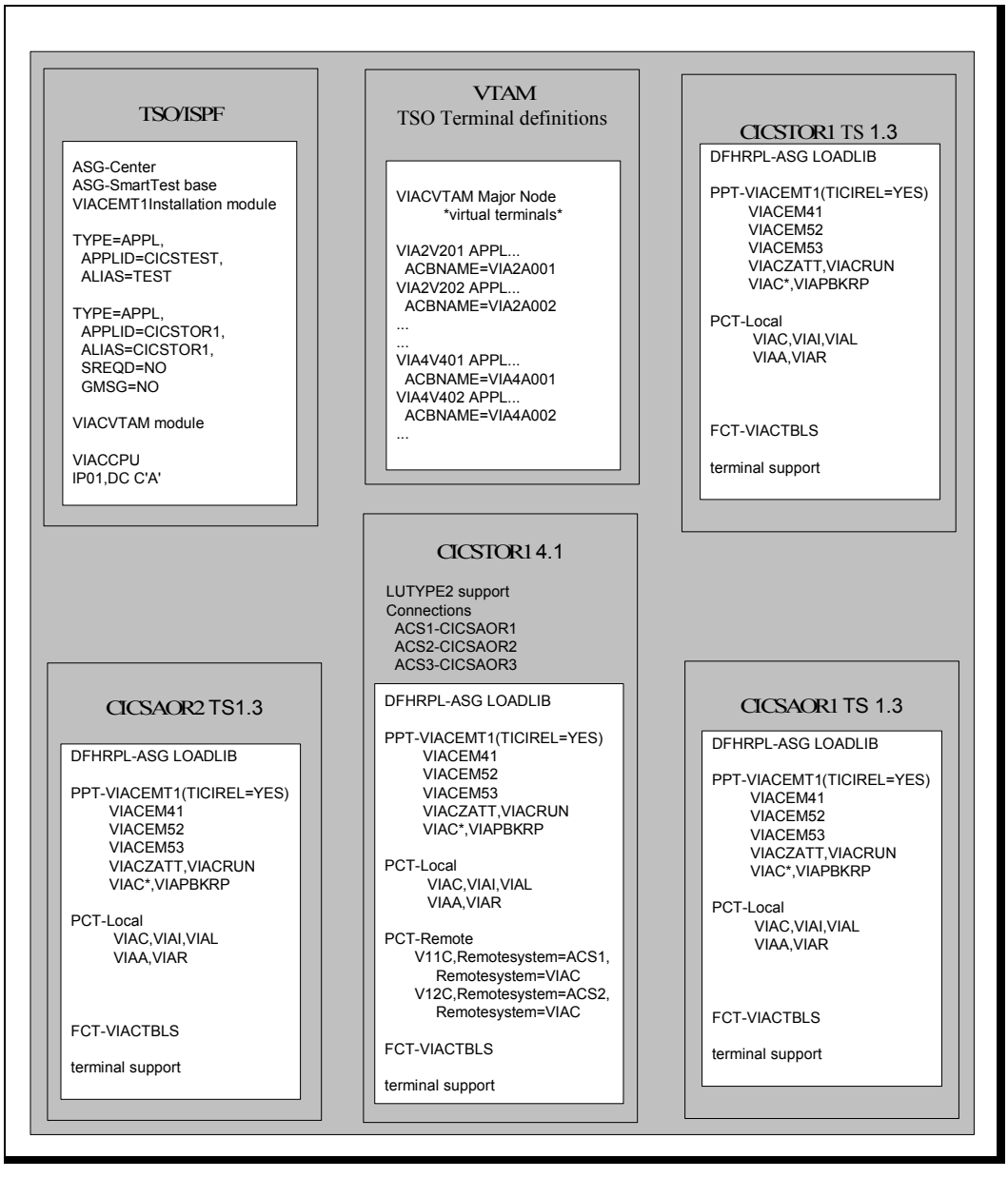
CICSAOR3 - a CICS TS 1.2 AOR

```
Connection Definition: TCS1, associated with NETNAME CICSTOR1

           ASG-SmartTest-CICS is not installed in this region
```

Figure 13 shows the components required to support all of these CICS regions simultaneously, plus CICSTEST, which is defined as a standalone TS 1.2 region. The only additions required for SmartTest-CICS's Remote Connection support for MRO are the Remote PCT definitions in CICSTOR1.

Figure 13 • MRO Components Illustration



These definitions also illustrate the use of the Installation table parameter T1CIREL to support multiple releases of CICS.

When a user connects to CICS through SmartTest-CICS, the program VIACRUN inquires against the TOR's connection definitions and remote PCT definitions. The first connection from SmartTest-CICS to CICSTOR1 results in the Remote Connections screen (accessed from Option 3 on the SmartTest Session Setup screen) shown in Figure 14.

Figure 14 • SmartTest-CICS Remote Connections Screen

```
                          Remote Connections
Command ===> ____________________________________________ Scroll ===> CSR

Specify SYSIDs which are to be used for ASG-SmartTest
monitoring.

Current SYSID CET1

SYSID  Monitor  NETNAME   Type  CICS Status    SmartTest-CICS Status TRANID
-----  -------  --------  ----  --------------  --------------------- ------
TOR1     YES    CICSTOR1  TOR   AVAILABLE       ACTIVE                VIAC
AORD     NO     CICSAORD  LU61  NOT AVAILABLE   NOT DEFINED
AOR1     YES    CICSAOR1  LU61  AVAILABLE       ACTIVE                VI1C
AOR2     YES    CICSAOR2  LU61  AVAILABLE       ACTIVE                VI2C
AOR3     NO     CICSAOR3  LU61  AVAILABLE       DEFINED               VI3C
TOR2     NO     CICSTOR2  LU61  NOT AVAILABLE   NOT DEFINED
```

When all necessary definitions are in place, you can set the Monitor column to YES for any AOR that indicates SmartTest DEFINED. This is generally all that is required if your application transactions are transaction routed from the TOR (automatically routed to a specified AOR).

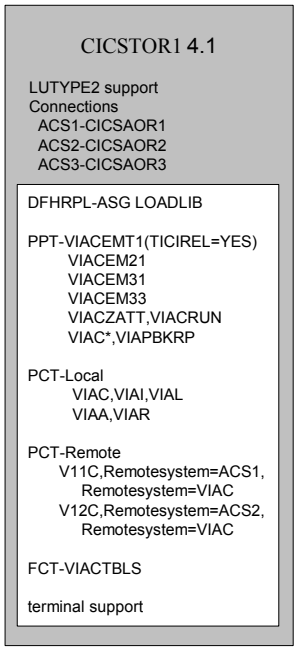
The Current SYSID at the top of the Remote Connections screen indicates the region from which SmartTest-CICS obtains resources when not in an active test session. This SYSID is also shown on the Session Setup screen as Active SYSID. SmartTest commands, such as NEWCOPY, LIst MEmory, and File Services (LIst FIles), always obtain their resources from the region indicated as the Active/Current SYSID. See Figure 15 on page 87.

For the VIAR transaction to perform properly in the TOR, these conditions must be met:

- The IBM group (DFHINQUI) must be installed. DFHINQUI is contained in the standard DFHLIST and DFHLIST2 lists.
- The SmartTest load library must be added to the TOR DFHRPL concatenation.

If you do not want to fully install SmartTest-CICS in your TOR, it is still necessary to define two programs and two transactions in the TOR.

Figure 15 • Example TOR Definition




```
PCT    Transaction     :  VIAA
       Program         :  VIACRUN
       Remotename      :  none
                          (Local definition)

PCT    Transaction     :  VIAR
       Program         :  VIACRUN
       Remotename      :  none
                          (Local definition)

PPT    Program         :  VIACZATT
       Language        :  Assembler

PPT    Program         :  VIACRUN
       Language        :  Assembler
```

The only reason to do a partial install in the TOR is if no user wants to test a program that executes in the TOR.

Note:

Program VIACLIST and TRANSID VIAL could also be added for troubleshooting. No other SmartTest-CICS transactions or programs should be defined to the TOR if you are attempting a partial installation.

Step 14 - Specifying Optional Program List Tables - Initialization and Shutdown

Optionally, add the PLT entry contained in the CNTL member VIACPLT to your Post-Initialization Program List Table and assemble the PLT using your standard procedures. A PLTPI entry for SmartTest-CICS might be required for co-existence with other vendor products or abend-handling exits. It is recommended that SmartTest-CICS not be added to the PLTPI until you complete "Step 17 – Validating SmartTest-CICS Installation through Native CICS" on page 90.

Note:

If initialization of SmartTest-CICS is not performed with a PLT Post-Initialization entry, processing for the first SmartTest-CICS user causes initialization of the product to be performed.

See the CNTL JCL member comments for instructions on where to place the SmartTest PLT entry in your PLT.

Optionally, add the PLT entry contained in the CNTL member VIACPLTX to your Shutdown Program List Table, and assemble the PLT using your standard procedures.

You can also initialize SmartTest-CICS with a sequential input device (card reader input) by including:

```
VIAI START\
```

Step 15 - Editing your CICS Startup JCL

Modify and add these JCL statements to the CICS startup JCL:

```
//DFHRPL   DD DSN=your CICS application libraries,DCB=BLKSIZE=largest
//          DD  DSN=ASG.VIACENxx.LOADLIB,DISP=SHR
//VIACTBLS  DD  DSN=ASG.VIACENxx.VIACTBLS,DISP=SHR
```

Note:

Ensure that the dataset with the largest blocksize is specified first for the DFHRPL libraries, or enter the DCB=BLKSIZE=*nnnn* parameter. Failure to do this causes unpredictable results during the execution of CICS.

SIT overrides might also be required. See "Step 9 - Defining the SmartTest-CICS File, Programs, and Transactions" on page 71.

Step 16 - Preparing the SmartTest-CICS Demonstration Programs

Customize JCL, compile and assemble programs, and assemble maps for the SmartTest-CICS demonstration programs. The demonstration programs and maps must execute successfully in a native CICS session before you can use them to demonstrate the testing and debugging capabilities under SmartTest-CICS.

CNTL library members used: VIACJMAP, VIACJCII, VIACJASM and VIACJPLI

The SmartTest-CICS demonstration programs are conventional CICS programs. They should be compiled/assembled as if they were developed at your site. The SmartTest-CICS CNTL library provides several JCL members that can be customized to reflect the compilers and assemblers in use at your site, or you can use your own procedures to compile and assemble the demonstration programs.

Note:

COBOL optimization rearranges and sometimes eliminates the object code that is generated for each COBOL statement. For this reason, the use of any type of COBOL optimization is not recommended.

JCL	Description
VIACJMAP	JCL to assemble the demonstration program mapsets. The COBOL and PL/I demonstration programs use mapset VIACMAP. The Assembler demonstration uses mapset VIACMAA. Customize the JCL to reflect the correct CICS and Assembler versions. ASG recommends that the demonstration programs be link-edited into the ESW LOADLIB that was concatenated into the CICS DFHRPL list in "Step 15 - Editing your CICS Startup JCL" on page 88.
VIACJCII	JCL to compile the demonstration programs in their COBOL II and later format. This JCL uses CNTL library members VIACDEM2, VIACCII, VIACCII2, VIACCII3, and VIACPYII. The link-edit requires the ESW LOADLIB for the inclusion of VIACBR14. Customize the JCL to reference the COBOL II and CICS releases you normally specify. Run the job. **Note:** Choose the JCL for the version of COBOL that is used most often for testing at your site. No matter which JCL member you choose, the program names are still VIACDEMO, VIACCOB, VIACCOB2, and VIACCOB3. These program names should have been defined to CICS as COBOL programs when you created the SmartTest-CICS PPT definitions in "Step 9 - Defining the SmartTest-CICS File, Programs, and Transactions" on page 71.

JCL	Description
VIACJASM	JCL to assemble the SmartTest-CICS command-level Assembler program. This program can be invoked from an option on the VCOB demonstration program menu, or as a separate transaction. Even if your site did not install the SmartTest-Assembler option, you can choose to assemble the VIACASM program so that it can be invoked from VCOB. Customize the JCL to reflect your usual Assembler and CICS releases and run the job.
VIACJPLI	JCL to compile and link the VIACPLI demonstration program. VIACPLI must be compiled, analyzed, and linked into a user load library that is part of the DFHRPL concatenation.

Step 17 – Validating SmartTest-CICS Installation through Native CICS

These tasks must be complete before the SmartTest validation process begins.

Task	Description
1._____	All relevant CNTL members must be updated and processed. **Note:** Ensure that the VIACASMJ CNTL member is updated and processed.
2._____	CICS PPT, PCT, FCT, and TCT entries must be updated and installed.
	_____ DFHRPL and VIACTBLS must be updated for CICS startup JCL.
	_____ VIACFCT DSNAME must be correct and enabled online. **Note:** Review your use of the RENTPGM=NOPROTECT SIT override. See "RENTPGM Restrictions" on page 33 for more detailed information.
3._____	The VTAM environment must be updated and varied active.
	_____ VIACCPU and VIACVTAM must be updated for all CPUs.
	_____ Supplied VTAM DLOGMODs must be compatible for CICS.
	_____ The system must be updated to enable the next IPL to pick up VTAM entries.
4._____	Shipped PTFs must be applied (if any).

Validating CICS

Log on to CICS (using your normal procedures) and perform these transactions to ensure that SmartTest-CICS operates in the CICS region.

To verify that all the required installation steps have been completed successfully

1. Log on to CICS (using your normal procedures).
2. Run the VIVP installation verification transaction. See "Using the VIVP Transaction" on page 93 for more information.

   Or

   Perform these steps:

   a. CEMT I DATASET(VIACTBLS) or CEMT I FILE(VIACTBLS)

      Change the status from closed to open. If you are unable to open this file, do not continue until this file can be opened.

   b. CEMT I TRAN(VIA*)

      Ensure that the SmartTest-CICS Transactions (VIAC, VIAI, and so forth) are defined.

   c. CEMT I PROGRAM(VIA*)

      Ensure that the SmartTest-CICS programs (VIACEXEC, VIACEMAL, and so forth) are defined. Make sure that each program definition shows the correct program language. Make sure that the programs specified in "Step 9b - Adding SmartTest-CICS Resource Definitions" on page 71 have EXECKEY=CICS (CEX on a CEMT display). Also, check for the demonstration programs.

3. VCOB (optionally VASM and VPLI)

   Select menu option 12 (Too Much Storage) to ensure that the programs VIACCOB and VIACOB2 were compiled successfully. Type `X` to exit.

4. Initialize SmartTest-CICS.

Initializing SmartTest-CICS

Enter these terminal transaction from a clear screen:

```
VIAI START
```

These are the expected messages:

```
ASG2760I ASG-SmartTest is Initializing
ASG2760I ASG-SmartTest Rx.y is active for CICS x.y
```

If these messages are not received, make a note of the complete text of any message that displays on the screen. It might be necessary to close and open the CSSL log in the CICS region, and check it for any messages regarding the SmartTest-CICS initialization attempt. If the message does not indicate that a VIAI STOP/PF1 is required, type this terminal transaction from a clear screen:

```
VIAI STOP
```

If the error message produced on the initialization attempt indicates that VIAI STOP/PF1 is required, or if SmartTest-CICS programs show a RES count after an attempt at a VIAI STOP, press PF1 instead of Enter after typing `VIAI STOP`. This should result in the message `SmartTest-CICS Has Been Stopped`, with perhaps a few additional messages about NEWCOPIES on SmartTest-CICS modules. Always check the RES counts for the SmartTest-CICS programs to make certain they are zero before attempting a restart. If you have any questions about what you are seeing when you attempt a native VIAI START, call the ASG Service Desk.

Terminating SmartTest-CICS

Normal product usage involves automatic termination of SmartTest-CICS when CICS is terminated. If necessary, you can terminate the SmartTest-CICS product by typing `VIAI STOP` on any blank screen.

This command is often necessary to facilitate new copy processing after applying SmartTest-CICS maintenance.

Note:

See Appendix B, "Interim Modifications" on page 133 for more information. If a stop request fails due to the message `ASG2764I ASG-SMARTTEST IS NOT ACTIVE`, and a VIAI START fails with the message `ASG2761I ASG-SMARTTEST IS ALREADY ACTIVE`, you can type `VIAI STOP` and press PF1 to force a stop.

To complete the installation process, modify other installation facilities such as IBM maintenance levels, security packages, and shared DASD handlers, if necessary.

Note:

Particular attention should be paid to the restrictions on DFHPCP and security software.

Using the VIVP Transaction

The VIVP installation verification transaction provides an on-screen exception report of specific SmartTest-CICS resources when the status or definition deviates from the installation option. The program does not initialize SmartTest-CICS and does not make any changes to the environment. Its basic intent is to report obvious errors and/or potential errors. The program verifies only basic installation and is not comprehensive. It runs in native CICS and does not require SmartTest for operation.

VIVP performs these functions:

- Reports the release of CICS.
- Verifies the existence of VIACEMT1 and displays the assembly date for VIACEMT1.
- Reports the existence (or absence) of a suffixed table (VIACEM*xx*) and the CICSVER that was used when it was assembled.
- Reports the absence of a SmartTest-CICS program (e.g., VIACEMAN).
- Reports the absence of a SmartTest-CICS PPT definition (e.g., VIACEMAN).
- Reports any SmartTest-CICS program that is NOT defined as ASSEMBLER.
- Reports any SmartTest-CICS program that is DISABLED.
- Reports the absence of a SmartTest-CICS transaction (e.g., VIAR, VIAA, VIAC, VIAI).
- Reports when a SmartTest-CICS transaction (e.g., VIAR, VIAA, VIAC, VIAI) is DISABLED.
- Reports when VIACTBLS does not exist or when the FCT DCB attributes are not defined correctly.
- Reports when a SmartTest-CICS transaction (e.g., VIAR, VIAA, VIAC, VIAI) is defined as ISOLATE=YES/. Transaction isolation must be enabled in the CICS region.
- Reports when SIT parameter RENTPGM is set to PROTECT.

Step 18 - Validating SmartTest-CICS

Note:

The dataset names mentioned in these validation steps are the default installed names. If you have changed them, use the changed names where the default names have been specified.

To validate the SmartTest-CICS installation

1 Start SmartTest-CICS by selecting the correct option for your site.

- If you installed SmartTest as described in "Step 4 - Preparing ISPF to Invoke SmartTest" on page 15, use the appropriate ISPF menu selection or the CLIST to start SmartTest.
- If you installed the ESW product menu as described in the *ASG-Center Installation Guide*, use the appropriate ISPF menu selection or the CLIST to display the ESW product primary screen.

  Select Test ▸ Module/Transaction to display the SmartTest primary screen.

  If you start SmartTest using the ESW primary screen, the product name displays as ESW - Testing/Debugging.

2 Select Help ▸ About to verify the product releases and levels of SmartTest-CICS and Center that are installed. The product names, release numbers, maintenance levels, and the operating system are indicated.

If SmartTest-CICS does not display in the Product name field, type `ENVIRONMENT` and choose the CICS environment (option 2).

3 Review and/or modify the SmartTest-CICS options by selecting Options on the action bar.

a Select Options ▸ Product Parameters. The Options - Product Parameters pop-up displays. Review and/or modify the parameter definitions and press PF3/PF15.

b Select Options ▸ Log/list/punch. The Options - Log/List/Punch Definition pop-up displays. Review and/or modify the Log, List, and Punch file defaults.

Enter the job statement information and press PF3/PF15.

This step is recommended before proceeding to ensure a valid operating environment for your user ID.

See the *ASG-SmartTest Reference Guide* or the online help facility for descriptions of these options.

4 Allocate an AKR by selecting File ▸ AKR utility. The File - AKR Utility pop-up displays.

Note:

If you have already created an AKR when validating another ESW product, proceed to step 5 on page 96.

a Complete the File - AKR Utility pop-up by typing the name of a ESW AKR to be allocated. Type `A` in the command input area and press Enter.

b On the File - AKR Allocate/Expand pop-up, the AKR name displays. Verify that it is correct, then:

- Enter the SMS values or the Volume and space parameters for the AKR.
- Enter an appropriate Unique parameter if this is a VSAM AKR.
- If this dataset must be cataloged in a user catalog, type `C` in the command area to display the AKR Catalog Information pop-up and enter the Catalog dataset name and password.
- Enter the job statement information for your site and submit the job by typing `S` in the command input area.
- Wait for the job to finish and verify that the AKR was successfully allocated and initialized.

Figure 16 shows the File - AKR Allocate/Expand pop-up using SMS and a VSAM AKR.

Figure 16 • File - AKR Allocate/Expand Pop-up

```
                        File - AKR Allocate/Expand
Command ===> ____________________________________________________________

       S - Submit JCL       E - Edit JCL       C - Specify Catalog

Expand existing AKR . . . NO             (Yes or No)

AKR data set name . . . . 'USER12.GENERAL.AKR'
  Volume  . . . . . . . . ______
  Unit  . . . . . . . . . 12             (Generic unit name)
  Space units . . . . . . RECORDS        (Records, Tracks or Cylinders)
  Primary space . . . . . 4000           (Primary amount in above units)
  Secondary space . . . . 0              (Secondary amount in above units)

Job statement information:
  //USER12_ JOB (DEUJXS,283200),
  //          MSGCLASS=A
  /*   INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
  /*
```

5 Analyze the programs by selecting File ▸ Compile/Analyze. Enter the appropriate information on the File - Analyze Submit pop-up.

a Use this statement for the compile and link JCL dataset:

```
ASG.VIACENxx.CNTL(VIACJCII) for COBOL II
```

b Enter the AKR dataset name that will contain SmartTest-CICS programs.

c Verify the JOB card and routing information by typing `E` in the command input area on the Analyze Submit screen. Type `SUB` in the command input area and press Enter to submit the analyze job. Press PF3 (End) to exit the Edit session.

d Verify that the analyze job completed successfully.

Note:

See the *ASG-SmartTest Reference Guide*, the Online Help facility, or the *ASG-Center Installation Guide* for information on the File - Analyze Submit pop-up.

6 Set up for the VCOB demonstration transaction by typing `ENV` in the command input area and pressing Enter to display the Environment Setup Menu.

Note:

SmartTest-CICS is shipped with VCOB in the Global Task Table. Make sure VCOB currently displays in the Global Task Table.

a Verify the AKR name shown.

b Type `2` in the command input area and press Enter to select CICS Session Setup.

c Verify the monitoring requests on the CICS Session Setup screen.

d Break at Start (Y/N) should be YES.

e Verify the CICS Logon Region APPLID on the CICS Session Setup screen. The Logon Region APPLID is the CICS Application ID or an alias defined in the VIACEMT1 table at installation time. The Toggle PF key should be PF12.

7 Connect (log on) to the CICS region specified in the CICS Logon Region APPLID field by typing `C` in the command input area and pressing Enter. If applicable, a standard CICS Logon screen displays for entry of the standard CICS Logon information. The connection process is complete when you receive this message:

```
ASG2733I ASG-SmartTest-CICS is active
```

This step validates that the VTAM and CICS TCT entries (if applicable) are installed and compatible.

These are the possible problems you might encounter at this point:

- The CICS region specified is not active.
- You are attempting to connect to a terminal that has not been defined to VTAM. The SmartTest-CICS Log file can be printed for further information.
- You are attempting to connect to a terminal that has not been defined to CICS. The SmartTest-CICS Log file can be printed for further information.
- SmartTest-CICS under CICS could not initialize. Verify that the FCT, PCT and PPT, entries are available and enabled. The SmartTest and CICS Log files can be printed for further information.

8 Execute the SmartTest-CICS NEWCOPY command for VIACCOB and VIACCOB2.

a Select Test ▸ CICS Newcopy or type `NEW VIACCOB` in the command line area.

If you have changed the transaction definition for VCOB to route it to an AOR, be certain that you confirm the ACTIVE SYSID.

For a discussion of MRO testing using the Remote Connections screen, see "Step 13 - Specifying MRO Transaction Definitions (Optional)" on page 83.

b Using the NEWCOPY command:

— Execute a CEMT NEWCOPY without toggling/cross-routing to CICS.

— Delete any old SmartTest-CICS program description control blocks.

— Ensure that the source image on the AKR and the load module are in sync.

9 Test the VIACCOB demonstration program by typing `VCOB`. This starts the test session and displays the Program View screen.

a The status box displays at the bottom of the screen. This box shows a `BREAK AT START OF TEST SESSION` status message. The current statement line is highlighted and contains chevrons (>>>>>>) in the line prefix area.

b Type `RUN` (PF4/PF16) to continue testing. The VCOB (VIACCOB) Demonstration Program Menu displays.

c Type `2` (Data Exception - S0C7) on the command line of the Demonstration Program Menu and press Enter. The Program View screen is redisplayed with a status box that indicates that a DATA EXCEPTION (S0C7) has occurred. The chevrons in the line prefix area (>>>>>>>) point to statement ADD +1 TO DATA-PACKED-DEC.

A screen displays noting that DATA-PACKED-DEC contains invalid data. Replace the * INVALID * data message with 0 (zero), then press PF4/PF16 (RUN) to continue testing. The test should now be complete and the VCOB Demonstration Program Menu should be redisplayed.

Type `X` (Exit This Demo) on the command line and press Enter. A message displays to indicate that the transaction has ended. Press PF12 to toggle to Program View.

The VIACCOB demonstration program is actually a detailed tutorial on SmartTest-CICS features. See the comments on the top of the source code for instructions.

d To view the comments, type `ISPF UP MAX` after you receive the `BREAK AT START OF TEST SESSION` message.

To reanalyze the demonstration programs after a test

1 Do not disconnect from CICS.

2 Type `Q CAN ALL` in the primary command line area to release the AKR members you have allocated during your test session.

3 Make the necessary corrections and submit the analyze.

4 Once the analyze has completed successfully, use the SmartTest-CICS NEWCOPY primary command for the reanalyzed load modules.

4 Customizing IMS

This chapter describes how to customize SmartTest-IMS and contains these sections:

Prerequisite

ASG-Center and SmartTest installation and customization must be performed before customizing SmartTest-IMS. If Center has not been installed, see the *ASG-Center Installation Guide* at this time. For SmartTest product installation, see "Installing and Customizing SmartTest" on page 5.

These are the steps for customizing SmartTest-IMS:

Steps
"Step 1 - Completing the Installation Worksheet" on page 100.
"Step 2 - Modifying and Executing CNTL Library Members" on page 101 (to customize for your environment).

Steps
"Step 3 - Modifying Installed CLIST Libraries" on page 102.
"Step 4 - Preparing User Definition Files" on page 103.
"Step 5 - Establishing Signon Validity Checking" on page 103.
"Step 6 - Validating SmartTest-IMS" on page 105.

Step 1 - Completing the Installation Worksheet

This table is an alphabetical list of the symbolic parameters that are used in the installation JCL procedures and CLISTs. Determine the appropriate values for your installation and record them for later reference.

Parameter	Default	Install Value	Description
VIASOFT	ASG	_____	High-level qualifier(s) for Center permanent libraries (non-VSAM).
CENTER	VIACEN*xx*	_____	Second level qualifier(s) for the Center permanent libraries (non-VSAM).
IMS	'IMS'	_____	IMS library high-level qualifier.
IOPCB LTERM	IOPCB	_____	Default LTERM name; found in VIAPUIDC CLIST.
MODBLKS	'IMS MODBLKS'	_____	IMS MODBLKS dataset name; found in VIAPUIDC CLIST.
MODBLKS SUFFIX	'A'	_____	IMS MODBLKS member name suffix; found in VIAPUIDC CLIST.
SYSDA	SYSDA	_____	Disk unit for temporary work space.
SYSOUT	'*'	_____	SYSOUT class.
VPVIDCLD	'ASG. VIACEN*xx*.CNTL	_____	LTERM and device type code file; found in VIAPUIDC CLIST.
VPVIDCLM	VIAPLTRM	_____	Member to contain LTERM and device type codes; found in VIAPUIDC CLIST.
VPVIDCTD	'ASG. VIACEN*xx*.CNTL'	_____	Transaction ID file; found in VIAPUIDC CLIST.
VPVIDCTM	VIAPTRAN	_____	Member to contain transaction IDs; found in VIAPUIDC CLIST.

Step 2 - Modifying and Executing CNTL Library Members

Note:

A complete list of all SmartTest CNTL members and their descriptions is available in Appendix F, "SmartTest CNTL and CLIST Members," on page 153.

These members should be reviewed for modification and executed.

Member	Description
VIAPOGMJ	JCL to create SmartTest-IMS environment-specific definitions. Any field-level or segment-level edit routines that are needed must be included in the link edit of VIAPOGM. These routines are normally included in the IMS nucleus and are named DFSME*xxx*, where *xxx* is the edit routine number. VIAPOGMJ links VIAPMFS modules that are IMS release-dependent. Use the specific IMS macro libraries and name the module VIAPMF*xx*, where *xx* is 31, 41, 51, 61, or 71. Assuming that your setup libraries and options are reflecting the correct version, this allows you to run the test with several versions of IMS. During execution, ASG-SmartTest determines the IMS version it is using and attempts to load the appropriate VIAPMFxx module. If this load fails, module VIAPMFS is used. VIAPMFS must be generated with the version of IMS that you are using. Rerun this jobstream when migrating to a new release of IMS or when maintenance is applied to the IMS LOAD library or the IMS GENLIBs.
VIAPIVPJ	JCL to create the demonstration facility.
VIAPMODB	JCL to create a list of transactions eligible for testing. This member can optionally be reviewed for modifications and executed.
VIAPCMD	Source for the SmartTest-IMS command authorized BMP. This member is used to install the SmartTest-IMS authorized BMP transaction. This transaction is optional, but is required by the SmartTest-IMS commands /START and /DISPLAY.
VIAPCMDJ	JCL to assemble and link the authorized BMP and to build the PSB. This member is used to install the SmartTest-IMS authorized BMP transaction. This transaction is optional, but is required by the SmartTest-IMS commands /START and /DISPLAY.

Member	Description
VIAPIVC	JCL to compile and link the VIAPIVP1 demonstration program. If you only have COBOL II and later compilers, change the PROC name from VIAPCVS to VIAPCII. **Note:** This member should be modified, but is not used until the validation step.
VIAPIMSA	JCL to allocate databases for the demonstration program. **Note:** This member should be modified, but is not used until the validation step.

Step 3 - Modifying Installed CLIST Libraries

Review the VIAPUIDC CLIST modifications.

VIAPUIDC	Used to restore IMS/DC system variables and parameters to the site defaults.

There are certain ISPF variables in the VIAPUIDC CLIST that, if unpopulated, view a user who enters the SmartTest-IMS/DC environment as a new user. In this case, the VIAPUIDC CLIST is redriven and default values are populated within the user's environment.

Although these variables are not used in all IMS environments, you can avoid the redriving of the CLIST by editing the VIAPUIMS CLIST in the ESW CLIST library. Populate the variables VPVPSB1 and VPVACBL1 with default PSB or ACBLIB libraries or with valid PSB or ACBLIB libraries that your applications will be using. If the CLIST is not updated, you can add a PSB/DBD library panel E.1.3.5 from the IMS/DC Session Setup screen or a library specified for the IMSACB value on panel E.2.3.5. If these panels are blank, the user is seen as a new user and VIAPUIMS CLIST resets the system defaults.

If running IMS/DC with public DBs, these libraries are not referenced. However, if they are unpopulated, it causes the redriving of the VIAPUIMS CLIST and overlays your library allocations.

Step 4 - Preparing User Definition Files

Two sample User Definition members containing transaction (VIAPTRAN) and LTERM names (VIAPLTRM) are supplied in ASG.VIACEN*xx*.CNTL. If another dataset name is used, the VIAPUIDC CLIST should reflect the dataset name change.

The optional transaction file contains the IMS Transaction IDs eligible for testing with SmartTest-IMS. When creating this member, list each Transaction ID on a separate line. For example:

```
PART
DSPALLI
ADDINV
```

Note:

In "Step 2 - Modifying and Executing CNTL Library Members" on page 101, the CNTL member VIAPMODB JCL created a transaction file containing transactions defined in your MODBLKS dataset. The generated transaction file, or a customized transaction file as described above, can be used for testing.

The LTERM file contains the IMS LTERM names and MFS device type codes defined to SmartTest-IMS for use with 3270 terminal emulation. When creating this file, each LTERM and device type should be listed on the same line. At least one blank must separate name from the type. If a device type is not specified, it defaults to 3270,2. For example:

```
IOPCB       3270,2
LTERM01     3270,4
LTERM02     3270-A2
```

Step 5 - Establishing Signon Validity Checking

The IMS command /SIGN is optional in the SmartTest-IMS 3270 emulation environment. This section suggests a way to provide some degree of signon validity checking and/or security verification within SmartTest-IMS. It is not intended to facilitate the use of the same DFSCSGN0 exit used by your IMS system.

When the application program receives control, the user ID field of the IOPCB contains the TSOID if the /SIGN command was not entered by the user. If the /SIGN command was entered and the IMS exit DFSCSGN0 is not present, the user ID field in the IOPCB contains the user ID from the /SIGN command. If the DFSCSGN0 exit is present, the user ID field of the IOPCB contains the TSOID.

Note:

A special PTF is available to cause signon to be required.

The use of the signon exit routine (DFSCSGN0) is supported by the SmartTest-IMS in the 3270 emulation environment. However, the exit executes in the user's TSO region and is quite limited in what IMS internal data structures and service routines it can access.

To be eligible for use with SmartTest, the exit must be link-edited as a separate load module and be in a library contained in the concatenation specified on the SmartTest-IMS environment screen. If the load module is present in the concatenation sequence, it is given control when the /SIGN command is entered on the emulated IMS terminal screen.

Note:

Because the DFSCSGN0 exit resides in the user's TSO region, it does not have access to most of the IMS data structures and service routines it would normally be able to access when executing in the IMS Control Region.

The exit routine, DFSCSGN0, can communicate with RACF-type security routines. However, the LTERM name associated with the IOPCB is meaningful only within the SmartTest-IMS session. As specified in the IBM reference manuals, the SCD and CTB are available to the DFSCSGN0 exit. However, these data areas are emulated by SmartTest-IMS and may not contain information useful to the exit routine.

The DFSCSGN0 exit routine is loaded by SmartTest-IMS during initialization of the test session. The same copy of the load module remains resident for the duration of the test session, that is, until an /RCL command is entered or until the user exits from the SmartTest IMS/DC environment. However, processing of the /SIGN command occurs under the application task. Consequently, any storage obtained using GETMAIN or any modules loaded by the DFSCSGN0 exit are not retained across transactions. Because of this, the exit must not save addresses of these storage areas within itself with the expectation that they will be valid for reference on subsequent executions of the /SIGN command.

The exit routine can obtain the user's TSOID from MVS with this Assembler instruction sequence:

```
L   R1,PSATOLD   GET CURRENT TCB ADDRESS
L   R1,TCBTIO-TCB(,R1)   GET TIOT ADDRESS
LA  R1,TIOCNJOB-TIOT(,R1)    POINT TO JOBSTEP NAME=TSOID
```

Step 6 - Validating SmartTest-IMS

To verify that the installation completed successfully

Note:

The dataset names used in these validation steps are the default installed names. If you have changed them, use the changed names where the default names are specified.

1 If you have not executed the CNTL members VIAPOGMJ and VIAPIVPJ, then do so now.

2 Enter SmartTest-IMS by selecting the correct option for your site.

- If you installed SmartTest as described in "Step 4 - Preparing ISPF to Invoke SmartTest" on page 15, use the appropriate ISPF menu selection or the CLIST to start SmartTest.
- If you installed the ESW product menu as described in the *ASG-Center Installation Guide*, use the appropriate ISPF menu selection or the CLIST to display the ESW product primary screen. Select Test ▸ Module/Transaction to display the SmartTest primary screen. If you start SmartTest using the ESW primary screen, the product name displays as ESW - Testing/Debugging.

3 Select Help ▸ About to verify the product releases and levels of SmartTest-IMS and Center that are installed. The product names, release numbers, maintenance levels, and the operating system are displayed.

If SmartTest-IMS does not display in the Product name field, press PF3/PF15 to return to the SmartTest primary screen. Type `ENVIRONMENT` and press Enter. Choose the IMS/DC environment option. You might have to press PF3/PF15 to return to a screen with the action bar containing the Help action.

Press PF3/PF15 to return to the SmartTest primary screen.

4 Review and/or modify the SmartTest options by selecting Options on the action bar.

a Select Options ▸ Product Parameters. The Options - Product Parameters pop-up displays. Review and/or modify the parameter definitions and press PF3/PF15.

b Select Options ▸ Log/list/punch to display the Options - Log/List/Punch Definition pop-up. Review and/or modify the Log, List, and Punch file defaults.

c Enter the job statement information and press PF3/PF15.

Note:

This step is recommended before proceeding to ensure a valid operating environment for your user ID. See the online help or the *ASG-SmartTest Reference Guide* for descriptions of these options.

5 To allocate an AKR, select File ▸ AKR utility. The File - AKR Utility pop-up displays.

Note:

If you have already created an AKR when validating another ESW product, proceed to step 6 on page 107.

a Complete the File - AKR Utility pop-up by entering the name of an AKR to be allocated. Type `A` in the command input area and press Enter.

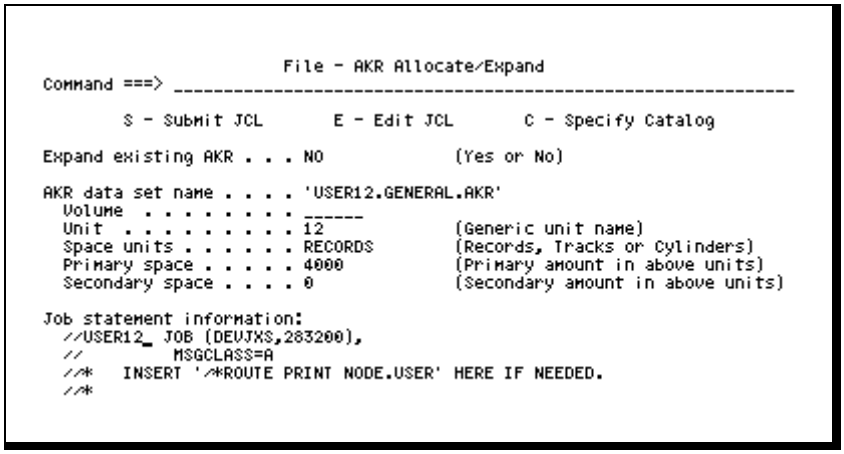
The fields on the File - AKR Allocate/Expand pop-up vary depending on the values for AKR-DSORG-VSAM and SMS in the Center installation options file.

b On the File - AKR Allocate/Expand pop-up:

- Verify the AKR name that displays.
- Type the SMS classes or the Volume and space information for the permanent AKR.
- Type an appropriate Unique parameter for the selected volume.
- If this dataset must be cataloged in a user catalog, type `C` on the command line and press Enter to display the AKR Catalog Information pop-up. Type the catalog DSN and password, if applicable.
- Type the job statement information for your site and submit the job by typing `S` in the command input area and pressing Enter.
- Verify that the AKR was successfully allocated and initialized.

Figure 17 is a sample of how the File - AKR Allocate/Expand pop-up looks using SMS and a VSAM AKR.

Figure 17 • File - AKR Allocate/Expand Pop-up



6 Analyze the demonstration program by selecting File ▸ Compile/Analyze. Enter the appropriate information on the File - Analyze Submit pop-up.

a The compile and link JCL dataset name field should be ASG.VIACEN*xx*.CNTL(VIAPIVC).

b Enter the AKR dataset name from step 5 on page 106.

c Verify the job statement and routing information by typing `E` in the command input area on the Analyze Submit screen and pressing Enter. Type `SUB` in the command input area and press Enter to submit the analyze job.

d Verify that the Analyze job completed successfully.

7 Set up the VIAPIVP1 demonstration program by typing `ENV` (the abbreviation for the ENVIRONMENT command) in the command input area and pressing Enter. This displays the Environment Selection Menu.

a Verify the AKR and load library names shown. Specify the application load libraries used in the VIAPIVC (compile/link) member.

b To select IMS/DC Session Setup, type `4` in the command input area and press Enter.

c To select the 3270 Terminal Emulation environment, type `E` in the command input area of the IMS/DC Session Setup screen and press Enter.

d To indicate that the databases will be allocated to the TSO region, type `1` in the command input area of the Setup 3270 Terminal Emulation pop-up and press Enter.

e To display the Execution Options pop-up, type `5` in the command input area of the Setup DBD/PSB pop-up and press Enter.

Verify that this data is as indicated:

IOPCB is set to IOPCB

INITIAL FORMAT is set to blanks

TOGGLE PFKEY is set to PF24

Press PF3/PF15 until the IMS/DC Session Setup screen displays.

f To display the GSAM/User File screen, type `A` in the command input area of the IMS/DC Session Setup screen and press Enter.

Ensure that the CLIST dataset and member name are valid.

To display the Convert Batch JCL screen, type `C` in the command input area and press Enter.

g Specify the default JCL library on the Convert Batch JCL screen as ASG.VIACEN*xx*.CNTL. Specify the Batch Execution JCL member name as VIAPIMSA.

Verify that the CLIST dataset and member name are valid.

To generate a CLIST from the JCL, type `C` in the command input area of the Convert Batch JCL screen and press Enter. A short message displays indicating successful JCL conversion. Press PF3/PF15 until the IMS/DC Session Setup screen displays.

h Type `T` in the command input area of the IMS/DC Session Setup screen and press Enter to display the Select Transactions pop-up.

Type `2` in the command input area of the Select Transactions pop-up and press Enter to display the Transaction pop-up.

Specify the Transaction Definition Dataset name as ASG.VIACEN*xx*.CNTL and the member name as VIAPTRAN. Ensure that the All MODBLKS Transactions field contains NO.

Press PF3/PF15 to return to the Select Transactions pop-up.

Type `1` in the command input area of the Select Transactions pop-up and press Enter to display the Transaction Definitions screen.

i Type `I` in the line command area on the Transaction Definitions screen and press Enter to insert a transaction.

Change the TRANCODE of the inserted transaction to VIAPIVP1 and press Enter.

The TRANCODE is highlighted and an S displays in the line command area to signify that this transaction has been selected for testing. Press PF3/PF15 until the IMS/DC Session Setup screen displays.

8 Test the VIAPIVP1 demonstration program by pressing PF4/PF16 (RUN).

This executes the CLIST allocations and displays the message `ASG2603I SmartTest-IMS is Active`.

Type `/FOR VIAPIVF1` to display the demonstration format screen.

a With the format screen being displayed, type `0C7` at the cursor and press Enter. This starts the test and displays the Program View screen.

b The status box displays at the bottom of the screen. This box shows a `BREAK AT START OF TEST SESSION` status message. The current statement line is highlighted and contains arrows (>>>>>>) in the line prefix area.

c Press PF4/PF16 (RUN). A data exception (0C7) should occur on the ADD +1 TO FORCED-S0C7-ABEND-COMP3 statement. A window is automatically displayed noting that FORCED-S0C7-ABEND-COMP3 contains invalid data.

The comments in the program describe four methods of fixing the abend. Choose method 1 by overtyping a valid number in the display window. Press PF4/PF16 (RUN) to resume the test.

The test should now be complete and the status box should display a BREAK ON PROGRAM RETURN status. Press PF4/PF16 again to display the original format screen.

d To verify that the database has been installed, enter any product code at the cursor and press Enter. This starts the test again and displays the Program View screen. Once again the status box shows a `BREAK AT START OF TEST SESSION` status message.

Press PF4/PF16 (RUN) to execute the program. The test should complete and the status box should display a `BREAK ON PROGRAM RETURN` status message. Press PF4/PF16 again to display the new FORMAT that contains the product information that you requested from the first FORMAT.

You can return to the original FORMAT by pressing Enter, or use the Clear key to clear the entire screen. If you chose to clear the screen, type `/RCL` to terminate the SmartTest-IMS session.

5 Validating ASM

This chapter describes how to customize SmartTest-ASM and contains these sections:

Prerequisite

ASG-Center and SmartTest product installation and customization must be performed before customizing SmartTest-ASM. If Center has not been installed, see the *ASG-Center Installation Guide* at this time. For SmartTest product installation, see "Installing and Customizing SmartTest" on page 5.

User Abend Codes

User abends are listed in online help and the Abend Codes appendix of the *ASG-Center Installation Guide*. If an abend that is not listed occurs, check the abend codes for the source manager that was running at the time of the abend.

Validating SmartTest-ASM

To verify that the installation completed successfully

Note:

The dataset names used in these validation steps are the default installed names. If you have changed them, use the changed names where the default names are specified.

1 Enter SmartTest by selecting the correct option from the menu. This tests the logon library allocations.

- If you installed SmartTest as described in "Step 4 - Preparing ISPF to Invoke SmartTest" on page 15, use the appropriate ISPF menu selection or the CLIST to start SmartTest.
- If you installed the ESW product menu as described in the *ASG-Center Installation Guide*, use the appropriate ISPF menu selection or the CLIST to display the ESW product primary screen.

  Select Test ▸ Module/Transaction to display the SmartTest primary screen.

  If you start SmartTest using the ESW primary screen, the product name displays as ESW - Testing/Debugging.

2 Select Help ▸ About to verify the product releases and levels of SmartTest-ASM and Center that are installed. The product names, release numbers, maintenance levels, and the operating system are displayed.

Press PF3/PF15 to return to the SmartTest primary screen.

3 Review and/or modify the SmartTest options by selecting Options on the action bar.

a Select Options ▸ Product Parameters to display the Options - Product Parameters pop-up. Review and/or modify the parameter definitions and press PF3/PF15.

b Select Options ▸ Log/list/punch to display the Options - Log/List/Punch Definition pop-up. Review and/or modify the Log, List, and Punch file defaults.

c Enter the job statement information and press PF3/PF15.

d Select Options ▸ PF keys to display the Options - PF Key Definition pop-up. See the online help or the *ASG-SmartTest Reference Guide* for more information about these options.

4 To allocate an AKR, select File ▸ AKR utility to display the File - AKR Utility pop-up.

Note:

If you have already created an AKR when validating another ESW product, proceed to step 5 on page 114.

a Complete the File - AKR Utility pop-up by entering the name of a AKR to be allocated. Type `A` in the command input area and press Enter.

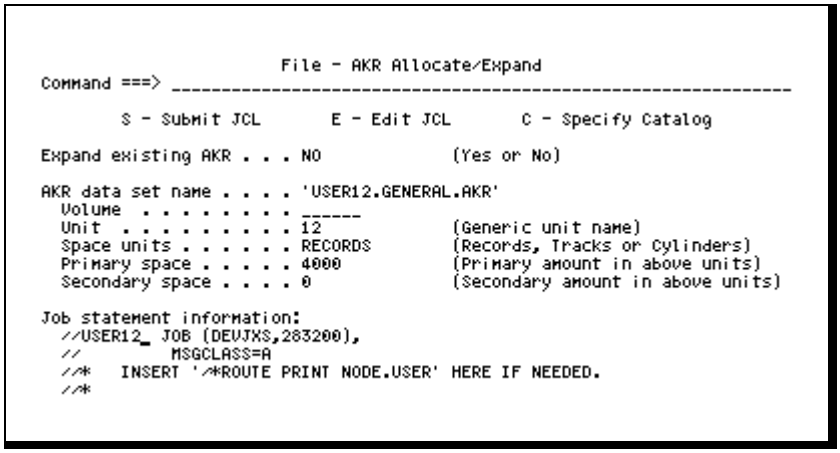
The fields on the File - AKR Allocate/Expand pop-up vary depending on the values for AKR-DSORG-VSAM and SMS in the Center installation options file.

b Complete these steps using the File - AKR Allocate/Expand pop-up:

- Verify the AKR name that displays.
- Type the SMS classes or the Volume and space information for the permanent AKR.
- Type an appropriate Unique parameter for the selected volume.
- If this dataset must be cataloged in a user catalog, type `C` on the command line and press Enter to display the AKR Catalog Information pop-up. Type the Catalog DSN and password, if applicable.
- Type the job statement information for your site and submit the job by typing `S` in the command input area and pressing Enter.
- Verify that the AKR was successfully allocated and initialized.

Figure 18 shows the File - AKR Allocate/Expand pop-up using SMS and a VSAM AKR.

Figure 18 • File - AKR Allocate/Expand Pop-up



5 Analyze the demonstration programs by selecting File ▸ Compile/Analyze. Enter the appropriate information on the File - Analyze Submit pop-up. This is the compile and link JCL dataset:

```
ASG.VIACENxx.CNTL(VIAPASMA)
```

a Enter the AKR dataset name from step 4 on page 113.

b Verify the JOB card and routing information by typing `E` in the command input area on the Analyze Submit screen. Type `SUB` in the command input area and press Enter to submit the analyze job.

c Verify that the analyze job completed successfully.

6 Set up the VIAPASM demonstration program by typing `SETUP` in the command input area and pressing Enter. The TSO Session Setup screen displays.

a Change the module to VIAPASM and type `ALL` in the Execution Parameters field.

Type `C` in the command input area and press Enter to select the Convert Batch JCL screen.

b Change the Member field in the JCL Library to VIAPASMJ. Also change the Member field in the CLIST Library to VIAPASMJ.

To generate a CLIST from the JCL, type `C` in the command input area and press Enter. A short message displays that indicates a successful JCL conversion.

c Press PF3/PF15 to return to the TSO Session Setup screen.

d To display the Test Session Tailoring screen, type `LIST TAILOR` in the command input area and press Enter.

Define an entry of VIA*.VIA* with COUNTS turned on. Define an entry of IEFBR14.IEFBR14 and set BREAK ON ENTRY to YES.

Press PF3/PF15 to return to the TSO Session Setup screen.

e To display the Load Module Intercept List pop-up, type `LIST INTERCEPTS` in the command input area and press Enter.

Insert an entry for IEFBR14.

Press PF3/PF15 to return to the TSO Session Setup screen.

7 Test VIAPASM by pressing PF4/16 (RUN). This starts the test and displays the Program View screen.

The status box displays at the bottom of the screen. This box shows a status message of `BREAK AT START OF TEST SESSION`. The current statement line is highlighted and contains chevrons (>>>>>>) in the line prefix area.

Follow the instructions for the validation as documented in the program source.

6 Customizing APS

This chapter describes how to customize SmartTest-APS and contains these sections:

Prerequisite

ASG-Center and SmartTest product installation and customization must be performed before customizing SmartTest-APS. If Center has not been installed, see the *ASG-Center Installation Guide* at this time. For SmartTest product installation, see "Installing and Customizing SmartTest" on page 5.

Overview

Before generating an APS program that is to be debugged using SmartTest, set the APS DEBUG option on the APS Generator Options panel to YES. When you specify YES, APS can generate the proper macro-like statements in the COBOL source so that SmartTest can function properly.

Invoking the SmartTest Analyze

To invoke the proper SmartTest processing during the compile process, follow this step:

- Submit the final jobstream using the job submit utility VIASUBDS.

To integrate the SmartTest analyze submit facility into the APS product

1 Use the APS DEBUG field on the APS Generator Options screen (see Figure 19) is used to invoke the analyze submit facility. The APS Generator Options screen is provided by INTERSOLV, Inc.

Figure 19 • APS Generator Options Screen

```
--------------------------- APS Generator Options ----------------------------
OPTION ===>

 TARGET OS ===>                (MVS, OS2, PCDOS, OS400, VSE)
        DC ===>              (IMS, CICS, DLG, DDS, MVS, or ISPF (prototyper))
        DB ===>                (IMS, DLI, VSAM, or IDMS)
       SQL ===>                (OS2DM, XDB, SQLDS, DB2)

 JOB CLASS ===>                   JOB DEST ===>
 MSG CLASS ===>               CARDIN MEMBER ===>

   LISTGEN ===>       (Yes or No) COBOL-II ===>          (Yes or No)
     COBOL ===>                CICS RELEASE ===>          (1.7,  2.1 or 3.1)
    OBJECT ===>                 IMS RELEASE ===>          (2 or 3)
   MFS/BMS ===>                       SUPRA ===>          (Yes or No)
    GENSRC ===>                      EBCDIC ===>          (Yes or No)
APS DEBUG  ===>                     PC CICS ===>          (IBM, MFOCUS)

  APS Parm ===>
COBOL Parm ===>
```

2 Type `YES` in the APS DEBUG field and press Enter to process the APS compile JCL. If you enter NO in the APS DEBUG field, the compile JCL is submitted without performing the SmartTest analyze.

3 Modify the APS18SUB CLIST. The APS CLIST (APS18SUB) should be modified as indicated below to process the APS compile JCL. Figure 20 shows this CLIST.

Figure 20 • APS CLIST (APS18SUB)

```
PROC 0
CONTROL NOMSG
/* CLIST(APS18SUB) - APS1802 - 1/15/89 - INTERSOLV
ISPEXEC VGET (SSMMDB SSMIENV SSMIJRVL) ASIS
IF &SSMMDB = IDMS AND &SSMIENV = LOCAL THEN +
 DO
  IF &SSMIJRVL = DUMMY       THEN +
   DO
    ISPEXEC VGET(SSMIVNO) PROFILE
    SET &SSMIVNO = &SSMIVNO +1
    ISPEXEC VPUT(SSMIVNO) PROFILE
   END
 END
ISPEXEC VGET (ZTEMPF ) ASIS
CONTROL MSG
ISPEXEC VGET(SSMEXIT SSMEXPGM SSMEXTYP) PROFILE
IF &SSMEXIT = YES THEN ISPEXEC SELECT &SSMEXTYP(&SSMEXPGM)
SUBMIT '&ZTEMPF'
```

The last line of the APS CLIST, SUBMIT '&ZTEMPF', should be modified as shown in this example:

```
ISPEXEC VGET (APANIM) PROFILE
IF '&APANIM' = 'YES' THEN +
  VIASUBDS '&ZTEMPF'
ELSE +
  SUBMIT '&ZTEMPF'
```

7 Validating PLI

This chapter describes how to validate SmartTest-PLI and contains these sections:

Prerequisite

ASG-Center, SmartTest, and SmartTest-PLI must be installed and customized before attempting to validate SmartTest-PLI. If Center has not been installed, see the *ASG-Center Installation Guide* at this time. For SmartTest product installation, see "Installing and Customizing SmartTest" on page 5.

User Abend Codes

User abends are listed in the Abend Codes appendix in the *ASG-Center Installation Guide*. If an abend that is not listed occurs, check the abend codes for the source manager that was running at the time of the abend.

Validating SmartTest-PLI

To verify that the installation completed successfully

Note:

The dataset names mentioned in these validation steps are the default installed names. If you have changed them, use the changed names where the default names are specified.

1. Analyze the demonstration programs by selecting File ▸ Compile/Analyze. Enter the appropriate information on the File - Analyze Submit pop-up.

   a. The dataset that contains the compile and link JCL should be ASG.VIACEN*xx*.CNTL(VIAPPLIC).

   b. Enter the AKR dataset name from "Step 5 - Validating SmartTest (SmartTest-TSO Only)" on page 16.

   c. Verify the JOB card and routing information by typing `E` in the command input area on the Analyze Submit screen and press Enter. Type SUB in the command input area and press Enter to submit the analyze job.

   d. Verify that the analyze job completed successfully.

2. Set up the VIAPPLI demonstration program by typing `SETUP` in the command input area and pressing Enter. The TSO Session Setup screen displays.

   a. Change the module to VIAPPLI and type `ALL` in the Execution Parameters field. To select the Convert Batch JCL screen, type `C` in the command input area and press Enter.

   b. Change the Member field in the JCL Library to VIAPPLIJ. Also change the Member field in the CLIST Library to VIAPPLIJ.

   To generate a CLIST from the JCL, type `C` in the command input area and press Enter. A short message displays that indicates a successful JCL conversion. Press PF3/PF15 to return to the TSO Session Setup screen.

   c. To display the Test Session Tailoring screen, type `LIST TAILOR` in the command input area and press Enter.

   Define an entry of VIA*.VIA* with COUNTS turned on. Define an entry of IEFBR14.IEFBR14 and set BREAK ON ENTRY to YES.

   Press PF3/PF15 to return to the TSO Session Setup screen.

   d. To display the Load Module Intercept List pop-up, type `LIST INTERCEPTS` in the command area and press Enter. Insert an entry for IEFBR14 and press PF3/PF15 to return to the TSO Session Setup screen.

3 Test VIAPPLI by pressing PF4/16 (RUN). This starts the test and displays the Program View screen.

   a The status box displays at the bottom of the screen. This box shows a status message of `BREAK AT START OF TEST SESSION`. The current statement line is highlighted and contains chevrons (>>>>>>) in the line prefix area.

   b Follow the instructions for the validation as documented in the program source.

8 Customizing Stored Procedures

This chapter describes how to use SmartTest profiles and contains these sections:

Prerequisite

ASG-Center and SmartTest product installation and customization must be performed before customizing the SmartTest Stored Procedure option. If Center has not been installed, see the *ASG-Center Installation Guide* at this time. For SmartTest product installation, see "Installing and Customizing SmartTest" on page 5.

Step 1 - Modifying CNTL Library Member

To modify VIAPBIND

Note:

A complete list of all SmartTest CNTL members and their descriptions is available in Appendix F, "SmartTest CNTL and CLIST Members," on page 153.

1 Edit the CNTL member VIAPBIND and verify or change these fields:

Parameter	Description
VIASOFT='ASG'	Specify the high-level node for the ESW libraries.
CENTER='VIACEN*xx*'	Specify the middle-level node for the ESW libraries, where *xx* is the ESW version number.
DB2LIB=DSN410.DSNLOAD	Specify the name of the DB2 load library.

2 Verify or change the parameters following this statement:

```
//BIND.SYSTSIN DD *
```

DSN SYSTEM	Specify the name of the DB2 subsystem for the bind.
PLAN	Specify the name of the Plan to be used for ESW access of the DB2 catalog.

Note:

The PLAN name entered for this parameter is also entered in the VIA$PRMP member in the CNTL library for the installation option DB2-Procedure-Plan.

3 Verify or change the parameters following this statement:

```
//BIND.SYSIN DD *
```

PLAN	Specify the name of the plan to be used by DSNTIAD (i.e., modify DSNTIA*xx*, where *xx* is the DB2 release number).
LIB	Specify the name of the DB2 Load library where DSNTIAD is located.

Step 2 - Updating the DB2 Environment for Multiple DB2 Subsystem Support

To accommodate multiple DB2 subsystems

If your site has more than one DB2 subsystem, make these additional modifications to VIAPBIND:

1. Locate the BIND step in the instream PROC. Repeat the BIND step for each DB2 subsystem at your site. In each repeated BIND step, change BIND to BIND1, BIND2, and so forth. For example, if you have four subsystems, you should have four BIND steps: BIND, BIND1, BIND2, and BIND3.

2. Locate the //BIND.SYSTSIN DD and //BIND.SYSIN DD statements. For each BIND step added in "Step 1 - Modifying CNTL Library Member" on page 126, repeat the //BIND.SYSTSIN DD and //BIND.SYSIN DD statements. In each repeated set of control statements, change BIND to BIND1, BIND2, and so forth. In the SYSTEM(*????*) parameter for each step, change *????* to represent the appropriate DB2 subsystem.

Step 3 - Executing VIAPBIND

After making the necessary changes to the JCL parameters in VIAPBIND, execute the job to create the necessary plan(s).

Note:

VIAPBIND performs the BIND and grants PUBLIC access to the plan.

Appendix A
COBOL Compiler Options

Introduction

The tables in this appendix list the COBOL compiler options used by SmartTest. Each table contains the options that apply to a particular type of compiler.

Note:

A SmartTest analyze automatically forces these options to their required settings, so your standard compile/link JCL is usually acceptable. For the SYSPRINT listing, the SmartTest analyze tries to approximate the options provided by your site defaults and JCL. For example, if you specify NOSOURCE, the SmartTest analyze forces SOURCE for the compile, then strips the source statements out of the output listing before printing.

COBOL II Compiler Option

This table describes the COBOL II (and later) compiler options:

Required Compiler Option	Related Option	Comments
LIST	NOOFFSET	Required to establish the location of verbs and paragraph/section names.
MAP		Required to establish the location of data items in the user's load module.
NONUM		Required for compiler generated line numbers.
NOOFFSET	LIST	Required because OFFSET overrides LIST.
SOURCE		Required.
NOOPTIMIZER		Highly recommended for testing.

CA-OPTIMIZER II Compiler Option

This table describes the CA-OPTIMIZER compiler options:

Required Compiler Option	Related Option	Comments
LIST	NOOFFSET	Required to establish the location of verbs and paragraph/section names.
MDMAP		Required to establish the location of data items in the user's load module.
NONUM		Required for compiler generated sequence numbers.
NOOFFSET	LIST	Required because OFFSET overrides LIST.
SOURCE		Required.

Compiler Limitations

COPYLIBs With Debug Limitations

Currently, Analyze ignores (i.e., does not expand) copybooks that are flagged as DEBUG statements when the DEBUG option is not active. The COBOL compiler expands these entries, flagging each expansion line as DEBUG (i.e., a comment line since DEBUG is not active). The result is that the line numbers between the COBOL source and the Analyze source are different after the point of the COPYLIB insertion. Such programs produce sequence errors. These programs can be viewed by SmartTest; however, STEPping, BREAKing, and viewing disassembled code can produce unpredictable results.

Compiler Optimization Limitations

SmartTest processes COBOL II and later programs compiled with the OPTIMIZE compiler option, with these limitations:

- When testing with the ASM option ON, the disassembled code that displays for repeated code segments (i.e., embedded PERFORMs or subprograms) is shown only for the first occurrence of such code (i.e., the lowest address).
- When inserting pseudo code into repeated code segments (i.e., embedded PERFORMs or subprograms), you cannot insert user labels. This is because all pseudo code variables and labels are global, and insertion of multiple labels (one for each occurrence of the code) causes duplicate labels.

- When inserting pseudo code into repeated code segments (i.e., embedded PERFORMs or subprograms), avoid the use of the &COUNT internal variable. &COUNT maintains a separate instruction count for each separate occurrence of the repeated code. Instead, create a pseudo code variable, increment it on each pass through the pseudo code, and test the pseudo code variable rather than the &COUNT variable.
- When issuing the LIST COUNTS command, the default ordering is by execution address, not line number as stated on the screen. The result is that the same sequence of line numbers displays multiple times, once for each occurrence of repeated code segments. Each repeated line has its own independent execution count.
- Certain source lines might appear to be unexecutable from the Program View screen, and cause unexpected results for commands such as STEP, BREAK, and ZA (Zoom Assembler).

Compiler optimization also affects the reliability of certain SmartTest data display functions. These are some limitations that result from compiling with the OPTIMIZE compiler option:

- A variable value can reside in a register rather than in the storage where the variable has been assigned. SmartTest displays the contents of the storage to which the variable has been assigned.
- In the object code, code associated with one statement can appear to be part of a different statement.
- After optimization, code generated for a statement is dependent on register values loaded by code for preceding statements. Changing the path of flow in a program with the GO command can therefore deprive statements of necessary input.
- A breakpoint may not actually occur at the beginning of the code generated for the statement at which it was set.

For additional information about the effects of compiler optimization, see the *COBOL II Application Programming Guide*.

Any compiler optimization can result in relocation, combination, or elimination of underlying code generated for any COBOL statement. The STEP function and insertion of pseudo code in SmartTest can therefore be affected.

Appendix B
Interim Modifications

This appendix describes the procedures for requesting help and for applying PTFs.

Problems or Questions

If you have questions or problems:

1. Gather all applicable JCL listings, output listings, online log file listings, screen messages or screen prints, and diagnostic information.

2. Call the ASG Service Desk or your ASG Distributor.

Applying Interim Modifications

Periodic product updates are provided between major releases. Modifications are supplied in AMASPZAP format.

Place modifications in the ASG.VIACEN*xx*.PTF dataset that was included on the installation tape. This PDS contains these two members when installed:

- SPZAPJCL, which contains the JCL to apply AMASPZAP modifications.
- SAMPLPTF, which contains a dummy PTF.

You can add PTFs to the ASG.VIACEN*xx*.PTF dataset as needed.

SmartTest-CICS Interim Modifications

If you decide to apply SmartTest-CICS maintenance while your CICS test region is running, execute this sequence.

To activate the new product level in the CICS region

1. When no SmartTest sessions are active in the region, access the region as you normally would (not through SmartTest) and on a clear screen type:

   ```
   VIAI STOP
   ```

   If there are no users in session, you receive the message `ASG2762I ASG-SMARTTEST HAS BEEN STOPPED.`

2. Perform a CEMT NEWCOPY on all the PPT modules for SmartTest by issuing this command:

   ```
   CEMT SET PR(VIA*) NEWCOPY
   ```

3. On a clear screen type this command:

   ```
   VIAI START
   ```

4. After you receive the message, `ASG2760I ASG-SMARTTEST Rx.y IS ACTIVE FOR CICS x.y.`, SmartTest is fully updated in the CICS region.

   Note:

   If there are users in session, the message `ASG2766I SMARTTEST IS IN AN ACTIVE SESSION, THE "STOP" REQUEST WAS IGNORED` displays. You cannot continue with the NEWCOPY of the SmartTest modules in the CICS region at this time.

Appendix C
SmartTest User Tables and Exits

Introduction

This section describes the Step Exclusion table, Module Exclusion table, Backtrack Control table, and the Batch Submit Exit. These tables and exits are used to identify the modules and CSECTs to be:

- Excluded from the STEPping feature.
- Excluded from monitoring during a test session.
- Checked for authorization to do a SUBMIT command (SmartTest-TSO only).

These tables are located in the ASG.VIACEN*xx*.CNTL dataset.

Step Exclusion Table

The Step Exclusion table (VIAPEMXS) is used to identify the modules and CSECTs that are not to be STEPped through during execution time. These modules and CSECTs are ignored unless explicitly requested from the Session Tailoring screen.

Use the VIAPNOST macro to enter each exclude entry. This macro has two operands, MOD and PGM. If you omit either operand, the default value of * (asterisk) is used. The * indicates that the definition is generically specified. You can only specify an * as a suffix for a program or module name.

Examples

VIAPNOST MOD=ACME,PGM=TEST	Skips the TEST CSECT for only the ACME load module.
VIAPNOST PGM=ACME	Skips all CSECTS named ACME for any load module.
VIAPNOST MOD=PROD*	Skips all CSECTs found in load modules that begin with PROD.
VIAPNOST MOD=*,PGM=*	This type of entry is not permitted.

Caution! Use of generic names can cause conflicts with SmartTest if the entry is too generic in nature. For example, if you enter *.*, STEPping does not occur unless overridden by the user.

The JCL to assemble and link-edit the Step Exclusion table is contained in the VIAPEXIT member of ASG.VIACEN*xx*.CNTL.

Module Exclusion Table

Use the Module Exclusion table (VIAPEMEX) to identify modules that are to be excluded from monitoring during a test session. Control is passed to these modules and SmartTest is left in a suspended state. Pseudo code and other monitoring features are not available for excluded modules. Program checks and abends are still intercepted and reported; however, continuation from the point of the abend is not possible.

Modules to be excluded must follow these standard MVS linkage conventions:

- External modules are called through BALR, BAL, BAS, or BASSM.
- CSECTs within the monitored module are called through BALR 14,15.
- RETURN is the next instruction after the BRANCH.

Do not add generics for COBOL subroutines, such as ILBO or IGZ. Names entered in this table apply to load modules and CSECTs.

Module Exclusion Table Examples

```
DC      CL8'DATABASE'     Specific module name 'DATABASE'
DC      CL8'4DATA'        Any module beginning with 'DATA'
DC      8X'FF'            End of table delimiter
```

Generic entries must have the length included with the module name. This list of excluded module names are provided as defaults by SmartTest:

3DSN	DB2 modules
3IKJ	TSO modules
3IST	VTAM Modules
CPXCEM	Optimizer

Users of Datacom, IDMS, Supra, etc., should review the commented entries.

JCL to assemble and link-edit the Module Exclusion table is contained in the VIAPEXIT member of ASG.VIACEN*xx*.CNTL.

Excluded modules that issue the MVS STAE and SPIE macros interfere with the operation of SmartTest. Unpredictable results can occur.

Backtrack Control Table

Use the SET BACK size command to specify the size of the Backtrack execution history to be allocated. The size becomes the default buffer size for the current environment. SmartTest maintains two default buffer sizes, one for the CICS environment and one for all other environments.

The Backtrack execution history resides above the line in all environments. In the CICS environment, the Backtrack execution history is in the CICS address space for each user.

The current Backtrack buffer size is shown on the Test Session Options panel, which displays by typing `SET` with no operands.

A default size is determined from the user modifiable Backtrack Control table (VIAPBTTB).

The default size is used:

- If the SET BACK size command has never been executed.
- After the SET DEFAULTS command.
- After the SET BACK 0 command.

A table is supplied that sets default values for every user:

CICS - 100K Other environments - 1MB

If the table VIAPBTTB is not found or an entry for a USERID is not found, then the defaults of 100K for CICS and 1MB for other environments are used.

In addition to setting default sizes, the table is also used to establish maximum allowable sizes to be used for the Backtrack execution history.

For CICS, the table resides in the TSO/ISPF user's address space.

This table is assembled and link-edited as a part of module VIAPBTTB. It provides the ability to set default and maximum size values for the Backtrack execution history, for the TSO and CICS environments, by individual USERID.

A sample table is provided that sets default and maximum allowable values for a generic USERID. The table allows for wildcard characters to define patterns for the USERID. The first table entry that is found to match the USERID is used.

A default Backtrack execution history table is supplied as load module VIAPBTTB. The default size is 1MB for TSO and 100K for CICS. The maximum values allowed are 8MB for TSO and 1MB for the CICS environment. Reassemble and relink VIAPBTTB to change these values.

The sample table shown in Figure 21, as provided in control library member VIAPBTTB, contains:

Figure 21 • VIAPBTTB Sample Table

```
*     SAMPLE ENTRY FOR TSO USERID "SYSPROG"
*
* VIABKTRK USERID=SYSPROG,TSODEF=8M,TSOMAX=8M,CICSDEF=4M,CICSMAX=4M
*
*
*     SAMPLE ENTRIES THAT USE A PATTERN FOR USERID
*
* VIABKTRK USERID=USERID*,TSODEF=4M,TSOMAX=8M,CICSDEF=100K,CICSMAX=500K
* VIABKTRK USERID=???PAY,TSODEF=4M,TSOMAX=8M
*
*
*     ENTRY THAT PROVIDES VALUES FOR ALL USERIDS THAT WERE NOT
*     COVERED BY A PRIOR ENTRY IN THE TABLE.
*
         VIABKTRK                                                  *
               USERID=*,          USERID PATTERN                  *
               TSODEF=1M,         TSO DEFAULT SIZE                *
               TSOMAX=8M,         TSO MAXIMUM SIZE                *
               CICSDEF=100K,      CICS DEFAULT SIZE               *
                CICSMAX=1M         CICS MAXIMUM SIZE
```

The first pattern in the table that matches the TSO USERID is used to set the default values and maximum allowable values for that USERID.

The question mark (?) character in a pattern matches exactly one character in that position. The asterisk (*) character matches zero or more characters. The * character and the ? character can both be used in the same pattern.

Note:

Specifying a maximum buffer size of zero for TSOMAX or CICSMAX disables Backtrack for that environment.

To update the Backtrack Control table

1. Modify member VIAPBTTB in the control library
2. Submit the assemble and link job VIAPBTTJ.

Batch Submit Exit (SmartTest-TSO Only)

Use the Batch Submit Exit (VIAPEXSB) to check the TSO user's authority to perform a SUBMIT command. If the TSO user is not allowed to issue the SUBMIT command, a non-zero return code is passed back to the calling program. The calling program then displays a message indicating the SUBMIT was rejected by the user exit. See the SUBTABLE label in the user exit to add user IDs. The JCL to assemble and link-edit the Batch Submit Exit is contained in the VIAPEXIT member of ASG.VIACEN*xx*.CNTL.

Appendix D
SmartTest-CICS User Exits

User Exit 1 - Initialization Exit

User Exit 1, shown in Figure 22, is the Initialization Exit. This exit is optional and is used to define static storage areas to SmartTest-CICS that can be modified by monitored programs. To use this exit, specify EXIT1=YES on the Environment Table. For more information, see "Installing and Customizing SmartTest" on page 5.

Figure 22 • User Exit 1 - Initialization Exit

```
  MACRO                                                                 00010000
        VIACEX01 ,                                                      00020000
*********************************************************************
* ASG, INC.          ASG-SMARTTEST RX.X                    MMM YYYY * 00040000
*                                                                     * 00050000
*  OPTIONAL USER EXIT #1.  INITIALIZATION EXIT.                       * 00060000
*********************************************************************
* THIS IS A SAMPLE USER WRITTEN INITIALIZATION EXIT01 FOR SMARTTEST. * 00080000
*                                                                     * 00090000
* IF USED, IT IS INVOKED FROM PROGRAM VIACEMAL DURING SMARTTEST      * 00100000
* INITIALIZATION.                                                    * 00110000
*                                                                     * 00120000
* ITS PURPOSE IS TO PROVIDE A MEANS OF SPECIFYING USER DEFINED AREAS * 00130000
* FOR USE AS READ/WRITE STORAGE (OR EXECUTABLE CODE) THROUGHOUT THE  * 00140000
* LIFE OF THE CICS SESSION.                                          * 00150000
*                                                                     * 00160000
* THAT IS -                                                          * 00170000
*                                                                     * 00180000
*  1) FIXED MEMORY LOCATIONS THAT WILL BE USED AS READ/WRITE AREA    * 00190000
*       (FOR EXAMPLE AN MVS GETMAINED AREA)                           * 00200000
*                                                                     * 00210000
*  2) PROGRAM CODE BRANCHED TO DIRECTLY FROM APPLICATION PROGRAMS    * 00220000
*       (OTHER THAN CODE LOADED BY DFHPCP AS APPLICATION PROGRAM)     * 00230000
*                                                                     * 00240000
* ONLY AREAS THAT REMAIN STATIC AFTER INITIALIZATION SHOULD BE       * 00250000
* HANDLED BY THIS EXIT. AREAS THAT ARE OBTAINED, RELEASED AND        * 00260000
* RE-OBTAINED DURING THE CICS SESSION SHOULD BE HANDLED BY THE       * 00270000
* DYNAMIC STORAGE VALIDATION EXIT (VIACEX02).                        * 00280000
*                                                                     * 00290000
* THE AREAS SPECIFIED TO ASG-SMARTTEST MUST HAVE ASSOCIATED DUMMY PPT* 00300000
* ENTRIES IN BOTH CASES....                                          * 00310000
*                                                                     * 00320000
*        DFHPPT TYPE=ENTRY,PROGRAM=XXXXXXXX,RES=NO                    * 00330000
```

```
*                                                                     * 00340000
* IF SPECIFYING AN ALLOWABLE AREA, THE ONLINE PROTECTION FACILITIES  * 00350000
* SHOULD BE USED TO ADD AN "ALLOW" FOR THIS PROGRAM.                 * 00360000
*   EXAMPLE -                                                         * 00370000
*      AREA      NAME     OFFSET   LENGTH   PROTECT                  * 00380000
*      ----    --------   ------   ------   -------                  * 00390000
*      PGM     XXXXXXXX        0     9999      NO                    * 00400000
* IF SPECIFYING AN ALLOWABLE AREA, THE ONLINE PROTECTION FACILITIES  * 00350000
* SHOULD BE USED TO ADD AN "ALLOW" FOR THIS PROGRAM.                 * 00360000
*   EXAMPLE -                                                         * 00370000
*      AREA      NAME     OFFSET   LENGTH   PROTECT                  * 00380000
*      ----    --------   ------   ------   -------                  * 00390000
*      PGM     XXXXXXXX        0     9999      NO                    * 00400000
* TO USE THIS EXIT, SPECIFY "EXIT01 = YES' IN THE OPTIONS SECTION OF * 00420000
* THE ENVIRONMENT TABLE GENERATION CARD (VIACEMT1).                  * 00430000
*--------------------------------------------------------------------*
*  TO USE THIS EXIT,                                                 * 00450000
*--------------------------------------------------------------------*
*  PROVIDE YOUR OWN LOGIC TO :-                                      * 00470000
*                                                                    * 00480000
*          A) SET THE NAME OF THE CODE/MEMORY      (INTO 'NAME')     * 00490000
*          B) LOAD THE ADDRESS OF THE CODE/MEMORY  (INTO REG 1)      * 00500000
*          C) LOAD THE LENGTH  OF THE CODE/MEMORY  (INTO REG 0)      * 00510000
*          D) LINK TO LABEL 'STOW' AND RETURN                        * 00520000
*          E) REPEAT A-D UNTIL ALL SEPARATE AREAS ARE SPECIFIED      * 00530000
*                                                                    * 00540000
*  NOTES :                                                           * 00550000
*  =====                                                             * 00560000
*    1. ONLY REGISTERS 0,1,9 AND 'LINK' MAY BE USED AS SHOWN.        * 00570000
*    2. R6 IS THE BASE REGISTER FOR THE EXIT.                        * 00580000
*    3. WORK REGISTERS MAY BE USED BUT MUST BE SAVED AND RESTORED    * 00590000
*       BEFORE BRANCHING TO 'STOW' OR RETURNING ON R9.               * 00600000
*    4. A SAVE AREA IS PROVIDED IF NEEDED NAMED 'U2SAVE'             * 00610000
*    5. THE ACTUAL ADDRESS/LENGTH FOR EACH NAMED AREA MUST BE        * 00620000
*       CALCULATED USING YOUR OWN ALGORITHM.                         * 00630000
*    6. INCLUDE AN 'ALLOW' TABLE ENTRY USING THE "STORAGE PROTECTION"* 00640000
*       SCREEN. (SEE NOTE ABOVE).                                    * 00650000
*   ------------------------------------------------------------     * 00660000
*    - FOR EXAMPLE: IF THE SPECIAL CODE IS LOCATED VIA AN ADCON -    * 00670000
*    -              WITHIN THE C.I.C.S. CWA AREA               -    * 00680000
*    -      .                                                  -    * 00690000
*    -      MVC   NAME,=CL8'CWACODE'                           -    * 00700000
*    -      L     R1,512+20(R13)       EXTRACT CWA FIELD       -    * 00710000
*    -      L     R0,16(R1)  LOAD LENGTH (ASSUMING LENGTH IS    -    * 00720000
*    -      BAL   LINK,STOW              HELD IN FULLWORD AT   -    * 00730000
*    -      .                            OFFSET 16 WITHIN CODE)-    * 00740000
*    -      .     ...ETC                                       -    * 00750000
*   ------------------------------------------------------------     * 00760000
*                                                                      00770000
*       *---------------------------------*                            00780000
*       * INITIALIZATION EXIT STARTS HERE *                            00790000
*       *---------------------------------*                            00800000
EXIT01   CSECT                    MUST BE PRESENT                    00810000
         ORG  *-2                 ORG OVER EARLIER 'BR  R9' INSTR    00820000
         USING EXIT01,R6          TELL ASSEMBLER THE BASE REG.       00830000
*      *-------------------------------------------------------------* 00840000
*       * TELL SMARTTEST THE NAME, ADDR AND LEN OF SPECIAL AREA 1.  * 00850000
*      *-------------------------------------------------------------* 00860000
         MVC   NAME,=CL8'MYCODE1'   SET NAME OF MY FIRST CODE AREA A.  00870000
         L     R1,=A(ADDRESS OF SPECIAL CODE 1)  SEE NOTE 4.ABOVE  B.  00880000
         L     R0,=A(LENGTH OF SPECIAL CODE 1)   SEE NOTE 4.ABOVE  C.  00890000
         BAL   LINK,STOW          GO TO SAVE INFORMATION        D.  00900000
```

```
*      *-------------------------------------------------------------* 00910000
*       * TELL SMARTTEST THE NAME, ADDR AND LEN OF SPECIAL AREA 2.  * 00920000
*      *-------------------------------------------------------------* 00930000
        MVC   NAME,=CL8'MYCODE2'   SET NAME OF MY FIRST CODE AREA A.  00940000
        L     R1,=A(ADDRESS OF SPECIAL CODE 2)  SEE NOTE 4.ABOVE  B.  00950000
        L     R0,=A(LENGTH OF SPECIAL CODE 2)   SEE NOTE 4.ABOVE  C.  00960000
        BAL   LINK,STOW           GO TO SAVE INFORMATION        D.  00970000
*       *-------------------------------------------------*           00980000
*       * RETURN TO SMARTTEST'S INITIALIZATION  USING R9. *           00990000
*       *-------------------------------------------------*           01000000
        BR    R9                  RETURN                              01010000
        LTORG ,                   GENERATE LITERALS                   01020000
*       *------------------------*                                    01030000
*       * END OF USER EXIT 01    *                                    01040000
*       *------------------------*                                    01050000
        MEND                                                          01060000
```

User Exit 2 - Storage Validation Exit

User Exit 2, shown in Figure 23, is the Storage Validation Exit. This exit can optionally be used to dynamically override the SmartTest-CICS storage protection rules. To use this exit, specify EXIT2=YES on the Environment Table. For more information, see "Installing and Customizing SmartTest" on page 5.

Figure 23 • User Exit 2 - Storage Validation Exit

```
*********************************************************************
* ASG, INC.             ASG-SMARTTEST Rx.x                MMM YYYY * 00020000
*                                                                   * 00030000
*  OPTIONAL USER EXIT #2.  STORAGE VALIDATION EXIT.                 * 00040000
*********************************************************************
        MACRO                                                       00060000
        VIACEX02  ,                                                 00070000
.*******************************************************************
.* THIS IS A SKELETON 'USER WRITTEN' DYNAMIC EXIT(EXIT2) FOR SMARTTEST* 00090000
.*                                                                  * 00100000
.* IF USED, IT IS INVOKED FROM THE VALIDATION ROUTINE OF SMARTTEST TO * 00110000
.* PROVIDE FURTHER USER VALIDATION OF A TARGET ADDRESS AND LENGTH    * 00120000
.* BEFORE DECIDING TO GIVE A 'STORAGE VIOLATION' ERROR.             * 00130000
.*                                                                  * 00140000
.* TO USE THIS EXIT, CODE THIS MACRO PER EXAMPLE AND ASSEMBLE       * 00150000
.* SMARTTEST TABLES SPECIFYING 'EXIT2=YES'.                         * 00160000
.*                                                                  * 00170000
.* THIS ROUTINE IS EXTREMELY PERFORMANCE CRITICAL. MINIMIZE THE NUMBER* 00180000
.* OF TEST AND LOOPS PERFORMED, OTHERWISE A DEGRADATION OF          * 00190000
.* THE RESPONSE TIMES MAY BE NOTICED WITH SMARTTEST PROTECTED TASKS.  * 00200000
.*                                                                  * 00210000
.* PROGRAM CHECKS MUST NOT OCCUR IN THIS ROUTINE, SO PERFORM ANY    * 00220000
.* NECESSARY VALIDATION ON INTERMEDIATE ADDRESSES USED TO CALCULATE  * 00230000
.* ADDRESS/LENGTHS BEFORE USING IN INSTRUCTIONS WHICH ACCESS STORAGE. * 00240000
.*                                                                  * 00250000
.* IF A PROGRAM CHECK DOES OCCUR, PSW+ REGS ARE IN VIACEMSP PROGRAM  * 00260000
.* AT OFFSET +X'34' IMMEDIATELY AFTER THE PROGRAM CHECK.            * 00270000
```

```
.*                                                                    * 00280000
.*********************************************************************
*********************************************************************
* REGISTER USAGE:                                                     00310000
*                                                                     00320000
*   R0       - RESERVED ** DO NOT USE ***                             00330000
*   R1       - RESERVED ** DO NOT USE ***                             00340000
*                                                                     00350000
*   R2       - WORK REG                                               00360000
*   R3       - WORK REG                                               00370000
*                                                                     00380000
*   R4       - RESERVED ** DO NOT USE ***                             00390000
*   R5       - RESERVED ** DO NOT USE ***                             00400000
*                                                                     00410000
*   R6       - WORK REG                                               00420000
*   LINK(R7) - WORK REG USED FOR LINKAGE                              00430000
*   R8       - WORK REG                                               00440000
*   R9       - WORK REG                                               00450000
*   R10      - WORK REG                                               00460000
*   R11      - WORK REG                                               00470000
*                                                                     00480000
*                                                                     00480000
*   R12      - TCA                                                    00490000
*   R13      - CSA                                                    00500000
*   R14      - RESERVED ** DO NOT USE ***                             00510000
*                                                                     00520000
*   R15      - BASE REGISTER                                          00530000
*********************************************************************
* THIS EXAMPLE ILLUSTRATES THE FOLLOWING:                             00560000
*                                                                     00570000
* 1) FOR ONLY THE PROGRAM NAMED 'SPECIAL', ALLOW AN AREA IN THE CWA   00580000
*    FOR A LENGTH OF 128 BYTES TO BE UPDATED.                         00590000
*                                                                     00600000
* 2) FOR ALL PROGRAMS, ALLOW AN AREA WHICH IS INDIRECTLY ADDRESSED    00610000
*    FROM THE CWA, FOR A LENGTH CONTAINED IN THE 1ST TWO BYTES OF THE 00620000
*    AREA, TO BE UPDATED.                                             00630000
*********************************************************************
*        *--------*                                                   00650000
*        * ITEM 1 *                                                   00660000
*        *--------*                                                   00670000
        CLC   U2PGM,=CL8'SPECIAL'  OPTIONAL TEST OF CURRENT PROGRAM   00680000
         BNE   NOT1                NOT THE ONE                        00690000
*                ** ONLY FOR SPECIAL PGM ***                          00700000
         LA    R6,512(R13)   POINT TO CWA  (CSA+X'200')              00710000
         USING CWA,R6                                                 00720000
         L     R2,CWAFLD1    LOAD ADR OF AREA                       00730000
         LH    R3,=H'128'    LOAD LENGTH                            00740000
         BAL   LINK,VALADR   TRY THIS AREA                          00750000
NOT1     EQU   *                                                     00760000
*        *--------*                                                  00770000
*        * ITEM 2 *                                                  00780000
*        *--------*                                                  00790000
         L     R2,CWAFLD2    LOAD ADR OF AREA 2                     00800000
*       * SHOULD CHECK R2 IS VALID ADR BEFORE USING IN NEXT INSTR.    00810000
         L     R2,20(R2)     PICK UP ADDRESS FROM +20               00820000
*       * SHOULD CHECK R2 IS VALID ADR BEFORE USING IN NEXT INSTR.    00830000
         LH    R3,0(R2)      LOAD LENGTH FROM 1ST TWO BYTES         00840000
*       * EVEN IF GARBAGE ADDR IN R2 AT THIS POINT WILL NOT PGM CHK   00850000
         BAL   LINK,VALADR   TRY THIS AREA                          00860000
*        COPY IN MY DSECTS, AND THEN RESUME CSECT                    00870000
**       COPY DFHCSADS                                              00880000
CWA      EQU   *                                                    00890000
```

```
**       COPY  CSAWADS       COPY  USER CWA                          00900000
EXIT02   CSECT               RESUME CSECT                          00910000
*        ************************************                        00920000
*        * DROP THROUGH TO RETURN TO CALLER *                        00930000
*        ************************************                        00940000
         MEND                                                       00950000
```

User Exit 3 - Terminal I/O Optimization Exit

User Exit 3 has been removed as of SmartTest Version 2.1. It is no longer needed.

User Exit 4 - ACB Naming Exit

User Exit 4, shown in Figure 24, is the ACB Naming exit used to specify a list of ACB names that the VIACVTAM module uses to establish the TSO-CICS connection. This user exit allows you to customize the specified ACB names in the CNTL member VIACVTAM. To use this exit, link edit VIACEX04 to a library that can be loaded from STEPLIB, JOBLIB, LINKLIST, or LPA.

Figure 24 • User Exit 4 - ACB Naming Exit

```
*********************************************************************
* ASG, INC.         ASG-SMARTTEST Rx.x                    MMM YYYY * 00020000
*                                                                   * 00030000
*  OPTIONAL USER EXIT #4.  PROVIDE VTAM ACB NAMES TO ASG-SMARTTEST   * 00040000
*********************************************************************
*                                                                     00060000
*   VIACEX04 CAN BE USED TO SPECIFY A LIST OF ACB NAMES THAT          00070000
*          VIACVTAM WILL USE TO ESTABLISH THE TSO-CICS CONNECTION.    00080000
*                                                                     00090000
*   ON ENTRY TO VIACEX04:                                             00100000
*                                                                     00110000
*      REGISTER 13 CONTAINS THE ADDRESS OF A SAVE AREA.               00120000
*      REGISTER 14 CONTAINS THE RETURN ADDRESS, AND                   00130000
*      REGISTER 15 CONTAINS THE ENTRY POINT OF VIACEX04.              00140000
*                                                                     00150000
*  R9->VTAM-TSO Communication block and provides addressability       00160000
*    to the following (3) data items:                                 00170000
CSALS    EQU   X'D8',8          +D8 - CICS alias name, char(8)       00180000
CSACB    EQU   X'3C',8          +3C - CICS ACB name, char(8)         00190000
TSOAPL   EQU   X'34',8          +34 - TSO applic. id, char(8)        00200000
*                                                                     00210000
*                                                                     00220000
*   THE USER MUST SAVE THE REGISTERS ON ENTRY AND RESTORE THEM ON EXIT. 00230000
*                                                                     00240000
*                                                                     00250000
*   ON RETURN:                                                        00260000
*                                                                     00270000
```

```
*     REGISTER 15: SHOULD CONTAIN A COMPLETION CODE: A PLUS VALUE     00280000
*                  WILL CAUSE THE CONNECTION ATTEMPT TO BE ABANDONED.  00290000
*                                                                     00300000
*                  IF ZERO, THEN REGISTER 0 SHOULD CONTAIN THE ADDRESS 00310000
*                   OF A LIST OF ACB NAMES THAT CAN BE USED FOR THIS   00320000
*                    SESSION.                                         00330000
*                                                                     00340000
*                  A NEGATIVE VALUE WILL CAUSE THE ACB NAME SELECTION 00350000
*                  MECHANISM TO CONTINUE AS IF NO EXIT WERE PRESENT.  00360000
*                                                                     00370000
*   EACH ACB NAME IN THE LIST MUST BE 8 CHARACTERS (LEFT              00380000
*   JUSTIFIED AND PADDED WITH BLANKS) AND EACH LIST SHOULD BE         00390000
*   DELIMITED BY X'FF'.                                               00400000
*                                                                     00410000
*   FOLLOWING IS A SAMPLE EXIT ROUTINE.                               00420000
*                                                                     00430000

*********************************************************************
         PRINT NOGEN                                                  00450000
VIACEX04 CSECT                                                        00460000
         B     START-VIACEX04(R15) BRANCH AROUND CONSTANTS            00470000
         DC    CL8'VIACEX04'      MODULE ID EMBEDDED IN CODE.         00480000
         DC    CL8'&SYSDATE'      DATE AND TIME OF ASSEMBLY...        00490000
         DC    CL8'&SYSTIME'      ...IMBEDDED IN CODE.                00500000
START    STM   R14,R12,12(R13)    SAVE CALLERS REGS IN CALLERS SA.    00510000
         LR    R12,R15            GET EP IN R12.                      00520000
         USING VIACEX04,R12       ADDRESSABILITY FOR MODULE           00530000
         ST    R13,SAVE+4         SAVE CALLERS SA POINTER IN MY SA.   00540000
         LA    R9,SAVE            ADDR OF MY SA IN R9.                00550000
         ST    R9,8(,R13)         STORE THAT INTO CALLERS SA          00560000
         LR    R13,R9             ADDR OF MY SA IN R13.               00570000
*  You can call your own (authorized) program to obtain the LU        00600000
*  name of the TSO terminal. The chain is                             00610000
*     PSA -> ASCB -> TSB -> TSBX -> TVWA -> NIB                       00620000
*  You might want to select by CPUID.  The following instructions     00630000
*  just get the SID.                                                  00640000
*                                                                     00650000
         L     R4,16              LOAD THE CVT ADDRESS                00660000
         USING CVT,R4             ESTABLISH ADDRESSABILITY            00670000
         L     R4,CVTSMCA         EXTRACT THE SMF ADDRESS FROM CVT    00680000
         USING SMCABASE,R4        ESTABLISH ADDRESSABILITY            00690000
         MVC   SID,SMCASID         SAVE SID                           00700000
*                                                                     00710000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00720000
*  If you want to use your TSO terminal ID and you have RACF, this    00730000
*  can be obtained from the ACEE as follows:                          00740000
*                                                                     00750000
         USING PSA,0                                                  00760000
         L     R4,PSAAOLD         THIS USER'S ASCB                    00770000
         L     R4,X'6C'(,R4)      THE ASCB EXTENSION                  00780000
         L     R4,X'C8'(,R4)      THE ACEE                            00790000
         USING ACEE,R4                                                00800000
         MVC   TERMID,ACEETRID                                        00810000
*                                                                     00820000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00830000
*  Suppose you need the TSO user ID. That also is available from      00840000
*  the ASCB:                                                          00850000
*                                                                     00860000
         USING PSA,0                                                  00870000
         L     R4,PSAAOLD         THIS USER'S ASCB                    00880000
         L     R4,X'B0'(,R4)      TSO USERID (JOB NAME)               00890000
         MVC   USERID,0(R4)                                           00900000
```

```
*                                                                       00910000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00920000
*  The following instructions can be used to obtain the model number   00930000
*   of the terminal.                                                    00940000
*                                                                       00950000
         MVI   MODL,C'4'                                                00960000
         GTSIZE                                                         00970000
         CH    R0,=H'43'           IS IT A MOD 4?                       00980000
         BNL   MODCDEF             YES                                  00990000
         MVI   MODL,C'3'                                                01000000
         CH    R0,=H'32'           IS IT A MOD 3?                       01010000
         BNL   MODCDEF             YES                                  01020000
         MVI   MODL,C'5'                                                01030000
         CH    R0,=H'27'           IS IT A MOD 5?                       01040000
         BNL   MODCDEF             YES                                  01050000
         MVI   MODL,C'2'           N - MUST BE MODEL 2                  01060000
MODCDEF  DS    0H                                                       01070000

*                                                                       01080000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    01090000
*  If you want to examine the CICS ACB name, the CICS logon applid     01100000
*  that the user entered, or the TSO appl name, retrieve saved          01110000
*  register 9, and use the offsets listed above:                        01120000
         L     R1,SAVE+4                                                01130000
         L     R9,12+(R9+2)*4(R1)  RETRIEVE SAVED R9                    01140000
         MVC   CICSID,CSACB(R9)    CICS ACB NAME                        01150000
*        . . .                                                          01160000
*  Insert here code to select the proper list. If an error exists,      01190000
*  put non-zero value in 'retcode'.                                     01200000
*                                                                       01210000
*        MVC   RETCODE,=F'16'                                           01220000
         L     R0,CP1MOD2                                               01230000
*                                                                       01240000
RETURN   L     R13,SAVE+4          GET SAVE AREA ADDRESS                01250000
         L     R15,RETCODE         SET RETURN CODE                      01260000
         L     R14,12(R13)         RETURN REGISTER                      01270000
         LM    R1,R12,24(R13)      RESTORE CALLING ENVIRONMENT          01280000
         BR    R14                 RETURN                               01290000
*                                                                       01300000
         LTORG                                                          01310000
*                                                                       01320000
SAVE     DS    9D                                                       01330000
*                                                                       01340000
RETCODE  DC    F'0'                                                     01350000
CP1MOD2  DC    A(C1M2)                                                  01360000
CP2MOD4  DC    A(C2M4)                                                  01370000
CP2MOD5  DC    A(C2M5)                                                  01380000
*                                                                       01390000
MODL     DC    C' '                                                     01400000
SID      DC    CL4' '                                                   01410000
TERMID   DC    CL8' '              TSO TERMINAL ID                      01420000
USERID   DC    CL8' '              TSO USER ID                          01430000
CICSID   DC    CL8' '              CICS ACB NAME                        01440000
```

User Exit 5 - Dynamically Loaded Modules Exit

User Exit 5 has been removed as of SmartTest Version 2.1. It is no longer needed.

User Exit 6 - Providing LOGONMSG Data to CICS Exit

User Exit 6, shown in Figure 25, is used to provide logon message data to CICS.

Figure 25 • User Exit 6 - Provide LOGONMSG Data to CICS

```
*********************************************************************
* ASG, INC.          ASG-SMARTTEST Rx.x                    MMM YYYY * 00020000
*                                                                     * 00030000
*  OPTIONAL USER EXIT #6.  PROVIDE LOGONMSG DATA TO CICS.             * 00040000
*                                                                     * 00050000
*********************************************************************
*                                                                     * 00070000
*   VIACEX06 CAN BE USED TO SPECIFY LOGONMSG DATA WHICH CAN BE        * 00080000
*           RETRIEVED BY CICS WITH 'EXEC CICS EXTRACT LOGONMSG'       * 00090000
*            (NOT AVAILABLE ON CICS RELEASES PRIOR TO 1.7).           * 00100000
*                                                                     * 00110000
*   ON ENTRY TO VIACEX06:                                             * 00120000
*                                                                     * 00130000
*     REGISTER  3 CONTAINS THE ADDRESS THE TSO USERID (ISPF ZUSER)    * 00140000
*     REGISTER 13 CONTAINS THE ADDRESS OF A SAVE AREA.                * 00150000
*     REGISTER 14 CONTAINS THE RETURN ADDRESS, AND                    * 00160000
*     REGISTER 15 CONTAINS THE ENTRY POINT OF VIACEX06.               * 00170000
*                                                                     * 00180000
*  R9->VTAM-TSO Communication block and provides addressability       * 00190000
*    to the following (3) data items:                                 * 00200000
CSALS    EQU   X'D8',8           +D8 - CICS alias name, char(8)       * 00210000
CSACB    EQU   X'3C',8           +3C - CICS ACB name, char(8)         * 00220000
TSOAPL   EQU   X'34',8           +34 - TSO applic. id, char(8)        * 00230000
*                                                                     * 00240000
*                                                                     * 00250000
*   THE USER MUST SAVE THE REGISTERS ON ENTRY AND RESTORE THEM ON EXIT* 00260000
*                                                                     * 00270000
*                                                                     * 00280000
*   ON RETURN:                                                        * 00290000
*                                                                     * 00300000
*     REGISTER 15: SHOULD CONTAIN THE ADDRESS OF THE LOGONMSG         * 00310000
*                  in the FORMAT:                                     * 00320000
*                     Llogon message data                             * 00330000
*                WHERE 'L' IS A ONE BYTE LENGTH OF DATA TO FOLLOW.    * 00340000
*                                                                     * 00350000
*   LGNMSG=YES MUST BE SPECIFIED IN THE SIT (OR AS A SIT OVERRIDE).   * 00360000
*   DATA CAN BE EXTRACTED ONLY ONCE -- REFER TO THE EXTRACT COMMAND   * 00370000
*    IN THE COMMAND LEVEL APRM.                                       * 00380000
*                                                                     * 00390000
*   FOLLOWING IS A SAMPLE EXIT ROUTINE.                               * 00400000
*                                                                     * 00410000
*********************************************************************
```

```
PRINT NOGEN
VIACEX06 CSECT                                                          00440000
         B     START-VIACEX06(R15) BRANCH AROUND CONSTANTS              00450000
         DC    CL8'VIACEX06'      MODULE ID EMBEDDED IN CODE.           00460000
         DC    CL8'&SYSDATE'      DATE AND TIME OF ASSEMBLY...          00470000
         DC    CL8'&SYSTIME'       ...IMBEDDED IN CODE.                 00480000
START    STM   R14,R12,12(R13)    SAVE CALLERS REGS IN CALLERS SA.      00490000
         LR    R12,R15            GET EP IN R12.                        00500000
         USING VIACEX06,R12       ADDRESSABILITY FOR MODULE             00510000
         ST    R13,SAVE+4         SAVE CALLERS SA POINTER IN MY SA.     00520000
         LA    R9,SAVE            ADDR OF MY SA IN R9.                  00530000
         ST    R9,8(,R13)         STORE THAT INTO CALLERS SA            00540000
         LR    R13,R9             ADDR OF MY SA IN R13.                 00550000
** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00560000
*  Build the installation specific information to be extracted    *     00570000
*  by CICS.                                                       *     00580000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00590000
*                                                                       00600000
*  Retrieve saved register 9.                                           00610000
         L     R1,SAVE+4                                                00620000
         L     R9,12+(R9+2)*4(R1)  RETRIEVE SAVED R9                    00630000
*  At this point you can examine the CICS ACB name, the CICS alias,     00640000
*  and the TSO SLU name at the offsets listed above.                    00650000
*                                                                       00660000
         LA    R15,LGNMSG         DATA TO BE EXTRACTED.                 00670000
*                                                                       00680000
RETURN   L     R13,SAVE+4         GET SAVE AREA ADDRESS                 00690000
         L     R14,12(R13)        RETURN REGISTER                       00700000
         LM    R0,R12,20(R13)     RESTORE CALLING ENVIRONMENT           00710000
         BR    R14                RETURN                                00720000
*                                                                       00730000
         LTORG                                                          00740000
*                                                                       00750000
SAVE     DS    9D                                                       00760000
*                                                                       00770000
LGNMSG   DC    AL1(LMSG)                                                00780000
MSGO     DC    C'This data will be extracted by CICS.'                  00790000
LMSG     EQU   *-MSGO                                                   00800000
*                                                                       00810000
R0       EQU   0                                                        00820000
R1       EQU   1                                                        00830000
R2       EQU   2                                                        00840000
R3       EQU   3                                                        00850000
R4       EQU   4                                                        00860000
R5       EQU   5                                                        00870000
R6       EQU   6                                                        00880000
R7       EQU   7                                                        00890000
R8       EQU   8                                                        00900000
R9       EQU   9                                                        00910000
R10      EQU   10                                                       00920000
R11      EQU   11                                                       00930000
R12      EQU   12                                                       00940000
R13      EQU   13                                                       00950000
R14      EQU   14                                                       00960000
R15      EQU   15                                                       00970000
         END                                                            00980000
```

Appendix E

SmartTest-CICS MRO Installation and Setup

This appendix describes installation and setup requirements for the MROLIST parameter value and the additional installation definitions for supporting Cross-Route (CRTE).

Introduction

Running SmartTest in an MRO environment might require special installation requirements or connection procedures, based on the MRO testing needs. When using the MROLIST installation parameter (specified in the VIACEMT1 table), you need to specify how SmartTest is to process in an MRO environment.

Note:

See "Installing and Customizing SmartTest" on page 5 for specific instructions for installing SmartTest.

Installation Procedure

Define remote PCT definitions for the VIAC transactions in the TOR for each AOR where SmartTest is installed. See "Step 13 - Specifying MRO Transaction Definitions (Optional)" on page 83 for sample definitions.

If SmartTest is not installed in the TOR, then one additional local PCT (VIAR) and two PPT entries (VIACRUN and VIACZATT) must be defined.

User Setup

To define a remote connection

1. From the SmartTest-CICS Session Setup screen, connect to the TOR. Type `TOGGLE` and press Enter to return to TSO and select option 3 (Remote Connections screen).

2. Select the CICS SYSIDs that qualify for SmartTest testing and choose only the ones in which testing will occur. This minimizes unnecessary region communications.

3. Set the CURRENT SYSID field value to the CICS SYSID value of the AOR in which you want SmartTest to process when not in an active test session.

   SmartTest uses the CURRENT SYSID to determine from which region resources, such as files, memory areas, and load modules, are acquired. The CURRENT SYSID specification is an informal cross-routing technique for every SmartTest function that is not part of an active test.

4. Begin using SmartTest for testing and/or reviewing resources.

Note:

Because step 1 through step 3 are normally first-time setup tasks, they are not required for future sessions.

Appendix F
SmartTest CNTL and CLIST Members

This appendix lists the SmartTest CNTL and CLIST members.

SmartTest CNTL Members

Member	Description
STFINDAT	Script used to find date-sensitive code; contains a list of date identifiers; used as a parameter to the EXECUTE command.
VIA$PRMP	SmartTest default installation option parameters.
VIABKTRK	Macro used by Backtrack VIAPBTTJ JCL.
VIACALC	SmartTest COBOL training laboratory problem program.
VIADATA	SmartTest COBOL training laboratory data.
VIALAB	SmartTest VIAMAIN training laboratory program execution JCL.
VIALAB3	SmartTest VIAMAIN3 training laboratory program execution JCL.
VIALAC23	JCL to compile and link the VIAMAIN3 training laboratory program.
VIAMAIN	SmartTest COBOL training laboratory problem program.
VIAMAIN3	SmartTest COBOL II (and later) training program, using nested programs and other features.
VIAMEJCL	SmartTest VIAMERGE demonstration program execution JCL.
VIAMERGE	SmartTest COBOL demonstration program source.
VIAMIN01 VIAMIN02 VIAMIN03	Input files for the VIAMERGE demonstration program.

Member	Description
VIAMSGS	SmartTest COBOL demonstration program copybook.
VIAPALSC	Script used to execute an Alliance application level query.
VIAPASM	SmartTest Assembler demonstration program source code.
VIAPASMA	JCL to assemble and link the VIAPASM demonstration program.
VIAPASMJ	SmartTest VIAPASM demonstration program execution JCL.
VIAPBTSI	SmartTest BTSIN data cards for BTS installation verification demonstration.
VIAPBTSJ	SmartTest BTS execution JCL for BTS installation verification demonstration.
VIAPBTTB	SmartTest Backtrack allocation exit table source.
VIAPBTTJ	JCL to assemble and link SmartTest Backtrack allocation exit.
VIAPCII	COBOL II and later compile procedure. This PROC should be copied to a user PROCLIB, or it must be embedded in compile JCL members.
VIAPCOB	SmartTest COBOL demonstration program, using SmartTest features and online debugging techniques.
VIAPCOBC	JCL to compile and link the VIAPCOB demonstration program.
VIAPCOBJ	JCL to execute the VIAPCOB demonstration program.
VIAPDLGC	JCL to compile and link VIAPDLGM demonstration program.
VIAPDLGM	SmartTest COBOL dialog manager demonstration program.
VIAPEMEX	Module exclusion table for excluding modules from monitoring at test time.
VIAPEMOX	JCL to assemble and link VIAPEMOX member to the ESW load library. VIAPEMOX is used for testing IDMS programs which also access DB2.
VIAPEMXS	Step exclusion table for excluding modules from the SmartTest STEP facility.
VIAPEXIT	JCL to assemble and link edit the SmartTest user exits and tables.
VIAPEXSB	Batch submit user exit.
VIAPMSGB	SmartTest batch connect message source.
VIAPMSGJ	JCL to assemble and link the SmartTest batch connect messages.

Member	Description
VIAPNOST	Macro used by the VIAPEMXS JCL.
VIAPPLI	SmartTest-PLI demonstration program source.
VIAPPLIC	JCL to compile and link the VIAPPLI demonstration program.
VIAPPLIJ	SmartTest-PLI VIAPPLI demonstration program execution JCL.
VIAPQBTH	Alliance application level query for MVS batch environments; lists the data items found in the application; displays the associated program name, source library, and member for each data item.
VIAPQDB2	Alliance application level query for DB2 environments; lists the data items found in the application; displays the associated program name, source library, and member for each data item.
VIAPQDBP	Alliance application level query for environments using DB2 stored procedures; lists the data items found in the application; displays the associated program name, source library, and member for each data item.
VIAPQISP	Alliance application level query for ISPF batch environments; lists the data items found in the application; displays the associated program name, source library, and member for each data item.
VIAPQTSO	Alliance application level query for TSO environments; lists the program and all load modules where it resides, and lists all JCL members that execute the load modules.
VIAPRT	SmartTest COBOL training laboratory problem program.
VIAPSNAP	Load module to perform a snap dump.
VIAPSTUB	Object deck used by SmartTest link JCL.

SmartTest-CICS CNTL Members

Member	Description
VIACASM	SmartTest-CICS Assembler demonstration program source.
VIACASMJ	JCL to assemble the Environment Table (VIACEMT1) and CPU Table (VIACCPU). Modification instructions are in the VIACASMJ member.
VIACCII	SmartTest-CICS COBOL II or later demonstration program source.

Member	Description
VIACCII2	SmartTest-CICS COBOL II or later demonstration program source.
VIACCII3	SmartTest-CICS COBOL II or later demonstration program source.
VIACCPU	CPU Identification Table. Add all CPUs that will have access to SmartTest-CICS.
VIACCP21	Macro for assembling VIACEMT1.
VIACCP31	Macro for assembling VIACEMT1.
VIACCP32	Macro for assembling VIACEMT1.
VIACCP33	Macro for assembling VIACEMT1.
VIACCP41	Macro for assembling VIACEMT1.
VIACDBRM	DB2 DBRM data for binding VIACDB2.
VIACDEM2	SmartTest-CICS COBOL II or later demonstration program source.
VIACEMT1	Environment Options. This appendix lists the available options to customize SmartTest-CICS to your installation.
VIACEM21	Optional Environment Table member.
VIACEM31	Optional Environment Table member.
VIACEM32	Optional Environment Table member.
VIACEM33	Optional Environment Table member.
VIACEM41	Optional Environment Table member.
VIACEM51	Optional Environment Table member.
VIACENV	Macro for assembling VIACEMT1 module.
VIACEX01	User exit number 1 - Initialization exit.
VIACEX02	User exit number 2 - Storage Validation exit.
VIACEX04	User exit number 4 - ACB Naming exit.
VIACEX06	User exit number 6 - Exit to Provide LOGONMSG Data to CICS exit.
VIACFCT	CICS File Control Table (FCT) entries. Modify the DSNAME parameter of the FCT.
VIACJASM	JCL to assemble and link edit the Assembler demonstration program (VIACASM).

Member	Description
VIACJBND	JCL to bind the VIACDB2 DBRM CNTL member (VIACDBRM).
VIACJCII	JCL to compile and link edit the COBOL II demonstration system (VIACDEM2, VIACCII, VIACCII2, and VIACCII3).
VIACJES3	JCL to define a VSAM CSD for JES 3 sites.
VIACJMAP	JCL to assemble and link edit the demonstration program mapsets (VIACMAP and VIACMAA).
VIACJPLI	JCL to assemble and link edit the SmartTest-CICS PL/I demonstration program source (VIACPLI).
VIACMAA	BMS Assembler source for command level Assembler demonstration.
VIACMAP	BMS Assembler source for demonstration program map.
VIACPLI	SmartTest-CICS PL/I demonstration program source.
VIACPLT	CICS PLT Post-initialization entry.
VIACPLTX	CICS PLT Shutdown entry.
VIACPRAJ	JCL to format and print the CICS trace dataset.
VIACPRDJ	JCL to format and print the CICS dump dataset.
VIACPRVJ	JCL to format and print the VTAM trace dataset.
VIACRCT	DB2 RCT control cards.
VIACSDJB	For CICS 4.1 and above, the JCL to update the CICS CSD file for SmartTest-CICS PPT (VIACPPT) and PCT (VIACPCT). The TCT (VIACTCT) entries can be added to the CSD file using a modified version of VIACSD41. Modification instructions are in the VIACSDJB member.
VIACTBLD	Initialization member for the VIACTBLS file.
VIACTBLJ	JCL to define and initialize the SmartTest-CICS Monitoring Rules Table (Protection Table) -- VIACTBLS.
VIACTCT	CICS Terminal Control Table (TCT) entries. If not using the CICS Auto-Install facility for SmartTest-CICS, then add/delete DFHTCT entries to correspond to the number of terminals that can concurrently use SmartTest-CICS.
VIACVTAM	VTAM terminal entries model.

Member	Description
VIAPQCIC	Alliance application-level query for CICS environments; lists the transaction that invokes the specified program.
VIAPRMFT	RMF table allocation program.

SmartTest-IMS CNTL Members

Member	Description
VIAPCMD	SmartTest-IMS authorized command processor (BMP program source input).
VIAPCMDJ	JCL to assemble and link SmartTest-IMS authorized command processor (BMP program).
VIAPDTL0	SmartTest-IMS database input data cards.
VIAPIMSA	JCL to allocate databases for the demonstration program.
VIAPIVA1	SmartTest-IMS COBOL demonstration program.
VIAPIVC	JCL to compile and link the VIAPIVA1 demonstration program.
VIAPIVD	SmartTest-IMS DBD source input.
VIAPIVF1	SmartTest-IMS Format source input.
VIAPIVF2	SmartTest-IMS Format source input.
VIAPIVPJ	SmartTest-IMS JCL to create the demonstration facility.
VIAPIVPL	SmartTest-IMS PSB source input for database load.
VIAPIVP1	SmartTest-IMS PSB source input for database read.
VIAPMODB	SmartTest-IMS JCL to create transactions eligible for testing.
VIAPMOFF	Macro used by VIAPOGMJ.
VIAPOGMJ	JCL to create SmartTest-IMS environment-specific definitions. See "Step 2 - Modifying and Executing CNTL Library Members" on page 101 for more information.
VIAPMTRJ	JCL to print SmartTest-IMS internal trace file.
VIAPLTRM	LTERM names and MFS device types defined to IMS/DC.

Member	Description
VIAPQBTS	Alliance application-level query for BTS environments; lists the transaction that invokes the specified program.
VIAPQIMB	Alliance application-level query for IMS batch environments; lists the data items found in the application; displays the associated program name, source library, and member for each data item.
VIAPQIMS	Alliance application-level query for IMS environments; lists the transaction that invokes the specified program.
VIAPTRAN	IMS/DC transaction names eligible for testing.

SmartTest CLIST Members

Member	Description
SMARTEST	Used by the VIATEST CLIST to invoke the SmartTest product from native TSO.
VIAEDUSR	User exit to support source managers other than the ISPF options of Librarian and Panvalet.
VIAPJCLC	Invoked to execute the batch JCL conversion process.
VIAPJCLE	Displays errors during JCL conversion process.
VIAPSBAT	Invoked to edit generated JCL for the batch job submit function.
VIAPSUB	Invoked to submit a batch job.
VIAPTSOT	Invokes the SmartTest program under TSO test for diagnostic purposes only.
VIAPUBTS	Used to restore BTS system variables and parameters to the site standards. Default dataset names are provided and should be modified to use site standards, which can be found in existing BTS test JCL and CLISTs. These items have been provided with standard IBM installation defaults: BTS LOADLIB 'BTS.BTSLIB' BTS FORMAT 'IMS.FORMAT' QIOPCB TEMPORARY DATASET QALTPCB TEMPORARY DATASET QALTRAN TEMPORARY DATASET BTSOUT TERM BTSPUNCH none BTSDEBUG none BTSSNAP none

Member	Description
VIAPUIDC	IMS/DC only. Used to restore IMS/DC system variables and parameters to the site standards. Default dataset names, suffix for MODBLKS, and a default IOPCB LTERM name are provided and should be modified to use site standards, which can be found in existing IMS/DC JCL. These items have been provided with standard IBM installation defaults: IMS/DC FORMAT 'IMS.FORMAT' IMS/DC RECON none DFSSTAT none FPTRACE none IOPCB LTERM IOPCB IMS/DC MODBLKS 'IMS.MODBLKS' IMS/DC MODBLKS SUFFIX A IMS/DC TRANSACTION DEFINITIONS 'ASG.VIACEN*xx*.CNTL' MEMBER VIAPTRAN IMS/DC LTERM DEFINITIONS 'ASG.VIACEN*xx*.CNTL' MEMBER VIAPLTRM

Member	Description
VIAPUIMS	Used to restore IMS system variables and parameters to the site standards. Default dataset names and parameters for DLI and BMP are provided and should be modified to use site standards, which can be found in existing IMS test JCL or CLISTs. These items have been provided with standard IBM installation defaults: IMS RESLIB 'IMS.RESLIB' IMS PROCLIB 'IMS.PROCLIB' DFSVSAMP 'IMS.PROCLIB(DFSVSMDB)' DB2 LOADLIB 'DSN220.DSNLOAD' IMSMON none IEFRDER temporary dataset IMS PSB LIB none IMS DBD LIB none IMS ACB LIB none DLI PARMS BUF 8 SPIE 0 TEST 0 EXCPVR 0 RST 0 SRCH 0 MON N SWAP 0 IRLM N BKO N BMP PARMS OPT N SPIE 0 TEST 0 DIRCA 000 CPUTIME 0

Member	Description
VIAPUSPF	Restores ISPF system variables to the site standards: ISPF PROGRAM LOAD LIBRARY 'ISP.V3R1M0.ISPLOAD' (REQUIRED) ISPF PANEL LIBRARY none (see Note below) ISPF LINK LIBRARY none (see Note below) ISPF TABLE LIBRARY none (see Note below) ISPF MESSAGE LIBRARY none (see Note below) ISPF SKELETON LIBRARY none (see Note below) ISPF LIST DATASET SYSOUT ISPF LOG DATASET SYSOUT **Note:** For those default libraries listed above as None, your existing ISPF libraries are used if no other is supplied.
VIATEST	Invokes the SmartTest program from a CLIST using the correct application ID (VIAP).

Appendix G
Problem Determination

CICS: VEXE Transaction Abend

SmartTest-CICS acquires CICS resources through Command Level requests. Should a request receive an unexpected response, SmartTest abends the transaction with a VEXE abend code. Within the VEXE transaction dump, a piece of transaction storage contains information about the type of request and the CICS response codes. The transaction storage is normally the first allocated piece of storage and contains VIACEXEC at offset +8.

This table lists critical fields that the ASG Service Desk will need to help determine the cause of the abend:

Field	Description
+40 through +43	Command level function code.
+44 through +47	EIBRESP code.
+48 through +4B	EIBRESP2 code.
+B0 through +B7	The SmartTest CSECT name in control.
+B8 through +BB	CSECT internal reference number.

In addition to the VEXE transaction dump, the VIACEXEC control block is written to Temporary Storage under the key VIACEXEC. To access this information online, use the SmartTest-CICS File Services facility or enter this CECI command:

```
CECI READQ TS QUEUE('VIACEXEC') ITEM(1)
```

To expand this record, Tab to the INTO area and press Enter.

CICS: VIAL Memory List Transaction

For debugging purposes, the transaction VIAL is provided to view and update memory. Type *VIAL* to receive help information:

```
VIAL
                                                          CICS x.x

HELP: ADDRESS, PGM, MOD, CSA, TCA, TCT, TSM,       +/-/@OFFSET,
D=X"XX", D=C"C"
```

where:

Parameter	Description
ADDRESS	Identifies any absolute address.
PGM	Specifies the PPT module entry point.
MOD	Specifies the PPT module load point.
CSA	Specifies the CICS CSA control block.
TCA	Specifies the CICS TCA control block.
TCT	Specifies the CICS TCT control block.
TSM	Specifies the CICS TSM control block.
+*nnn*	Specifies the positive offset.
-*nnn*	Specifies the negative offset.
@*nnn*	Specifies the offset from the base address that points to an indirect address. The memory at the indirect address displays.
D=X"*xx*"	Changes the data at the specified address with the hexadecimal value(s) "*xx*".
D=C"*x*"	Changes the data at the specified address with the character value(s) "*x*".

This transaction is intended for the systems programmer and ASG personnel, and is not documented in the *ASG-SmartTest CICS User's Guide*.

Appendix H

SmartTest-CICS Resource Requirements

SmartTest-CICS requires varying amounts of CICS program and transaction storage, depending on the features used and the number of users. This table provides storage estimates for different types of storage.

Type of Storage	Space Requirement
Resident Program Storage	51K
Non-Resident Program RMODE(24)	
High Usage	114K
Low Usage	55K
Non-Resident Program RMODE(31)	
High Usage	95K
Low Usage	8K
Temporary	
Each active SmartTest terminal	2K
User-testing three COBOL II modules	1K
User-24 Transaction	
Average amount per testing defaults	15K
User-31 Transaction	
Average amount per testing defaults	8K
Counts feature for 3 programs	12K
Backtrack feature (default size)	100K

All of the resources used by SmartTest can be reclaimed by stopping SmartTest with the VIAI STOP transaction. In addition, all of the temporary storage records can be purged without stopping SmartTest by typing `VIAI PURGE`.

Index

D

E

X