CA-Top Secret®

Installation and Maintenance Guide
Release 3.0
VSE

SP1

Second Edition, September 2000

Contents

About This Guide

Purpose

This guide outlines:

- CA-Top Secret installation steps (a copy of the required JCL for each installation step is provided under the subheading, "Required JCL")
- Start-up and shut-down considerations
- VSE considerations
- TSSMAINT utility
- CA-Top Secret maintenance procedures.

Note: The *Installation Guide* does not include customization steps for subsystems. Special installation and implementation considerations for subsystems—that is, CICS, BATCH, CA-IDMS—can be found in facility-specific *Implementation* Guides.

Systems programmers involved with the installation of CA-Top Secret and security administrators responsible for CA-Top Secret's implementation should read this guide and also be familiar with the material presented in the following guides:

User Guide
Implementation: CICS
Control Options Guide
Planning Guide.

Organization

Chapter	Description
1	Introduces the requirements and prerequisites necessary for installing CA-Top Secret.
2	Lists the system requirements and the corresponding steps.
3	Lists the steps of CA-Top Secret installation, and describes each one.
4	Presents special considerations for Multi-CPU environments, SMF, JES2, HSM, and online job submission.
5	Explains starting and shutting down CA-Top Secret.
6	SNTL2TSS Conversion Utility
7	ALRT2TSS Conversion Utility
8	XSMA2TSS Conversion Utility
A	Lists the Distribution Tape format.
B	Describes how to enlarge and to recover your Security File.
C	Provides a convenient checklist to help ensure that you take all necessary installation steps.
Index	Provides an efficient way to locate specific material.

CA-Top Secret Publications

The following publications are supplied with CA-Top Secret:

Title	Contents
Installation Guide	CA-Top Secret installation for VSE, including: installation and maintenance steps, startup and shutdown considerations, backup and recovery procedures, and Security File conversion.
Command Functions Guide	Comprehensive reference to the TSS command and CA-Top Secret resources.
Control Options Guide	Comprehensive reference of CA-Top Secret control options.
Implementation: BATCH, STC and APPC Guide	Provides for setting up security in Batch, STC and APPC environments.
Implementation: CICS Guide	Provides for setting up security in a CICS environment.
Messages and Codes Guide	Provides all CA-Top Secret messages and codes.
Planning Guide	General guide for planning a successful CA-Top Secret security implementation and describing the latest enhancements to CA-Top Secret.
Report and Tracking Guide	Details the use of TSSUTIL, TSSTRACK, TSSAUDIT, TSSCFILE and TSSCPR and provides sample reports.
Troubleshooting Guide	Includes CA-Top Secret debugging options and facilities, information to be prepared prior to contacting Technical Support, and an explanation of customer support.
User Guide	Conceptual overview of CA-Top Secret architecture and capablilities. Also includes implementation instructions that span all facilities, including: implementation how to's, customization, and the CA-Top Secret utilities.

All guides are updated as required. Instructions accompany each update package.

Related Publications

The common component services formerly known as CA90s have been enhanced and renamed CA Common Infrastructure Services, CA-CIS. All the documentation for the services exists in the following publications supplied by Computer Associates:

Name	Contents
CA-CIS Administrator Guide	A guide for system administrators.
CA-CIS CAICCI User Guide	Describes how to use the communication protocols needed for cross-system communication.
CA-CIS Installation Guide	Tells how to install the CA-CIS.
CA-CIS Message Guide	Lists messages and codes for CA-CIS.
CA-CIS Reference Guide	Describes how to access and use the VSE common components.
CA-CIS Systems Programmer Guide	Details the programming requirements for command, security, and signon exits.

The following publications relate to CA-Top Secret and are available from Computer Associates:

Title	Operating System
CA-EARL Reference Guide	MVS/VM/VSE
CA-EARL User Guide	MVS/VM/VSE
CA-EARL Systems Programmer Guide	VSE
CA-EARL Examples Guide	MVS/VM/VSE

Command Notation

Enter the following exactly as they appear in command descriptions:

UPPERCASE	identifies commands, keywords, and keyword values which must be coded exactly as shown.
MIXEDcase	identifies acceptable command abbreviations. The UPPER-CASE letters represent the minimum abbreviation possible. The lowercase letters of the command may be entered for further clarification **Note:** In some cases, the command processor may require a more detailed abbreviation due to user-defined resource being the same as a command abbreviation. In these cases, either enter the complete command or code a national or numeric character in one of the first four positions of the user-defined resource.
underlining	identifies default operands. These operands do not need to be entered to take effect.
Symbols	"" / * # () , must be coded exactly as shown.

The following items clarify command syntax; do not type when they appear:

lowercase	indicates keyword values which you must supply.
[ ]	identifies optional keywords.
|	separates alternative keywords or values; choose one.
{ }	indicates that you must enter one of the keywords.
...	means the preceding value may be repeated more than once.

The following table indicates the appropriate command format.

Sample Command	Explanation
TSS PER(acid) DSN(dsname)	You must supply a value for the ACID and for the data set name.
MODE(DORM|IMPL|WARN|FAIL)	You must choose **only** one of the keywords.
TSS {ADDto } (acid) MCSAUTH {(INFO) } {REMove } {(MASTER)} {REPlace} {(SYS) } {(IO) } {(CONS) } {(ALL) }	You must select a command function, enter the name of an ACID, enter the MCSAUTH keyword, and select an operand from the list.

Chapter 1. Introduction

This chapter describes the installation process in general terms. The remainder of this guide contains detailed information and instructions needed to accomplish a successful installation.

Installation procedures follow this arrangement:

- Installation procedures are divided into numbered units called steps. Each step will complete a unit of the installation process, such as "Complete the Installation Worksheet".
- Each numbered step is further divided into alphabetized tasks. A task can apply to either all possible configurations or a subset of them.
- Each step begins with a description, which states the purpose of the step and lists any components affected by tasks included within the step.

1.1 Product Distribution

The machine-readable program materials required for installation are distributed as a multifile installation tape in library BACKUP format suitable for installation via the IBM Maintain System History Program (MSHP). Chapter 3, "Installation Materials," provides a detailed description of the tape format and contents.

If CA-Top Secret was purchased from IBM, the product is shipped as a V2 optional program product and should be installed using the IUI panels provided by IBM.

1.2 Using MSHP

MSHP provides the ability to control installation and maintenance activities in a consistent manner. This format also provides an installation mechanism that system programmers use to maintain the VSE operating system.

In much the same way, MSHP is used to perform the installation and maintenance of CA-Top Secret and other VSE products. When utilizing MSHP to install a product, a MSHP History File is required to archive product information, such as product identification and library residence. The History File is subsequently used during maintenance application for product and library identification and for archival of maintenance information.

The CA-Top Secret distribution tape includes the libraries containing CA-Top Secret and a corresponding History File. The target History File used to install CA-Top Secret should be kept separate from the operating system's History File. It is also recommended that the CA-Top Secret target libraries be kept apart from the VSE system libraries or sublibraries.

The CA-Top Secret installation procedure also installs CA-SAF functions to support the features of the current and future releases of VSE/ESA. The following information should be available before beginning the installation:

- The approximate number of present and future zones, divisions, departments, users, and profiles that will be defined to CA-Top Secret.
- The number of present and future DASD volumes CA-Top Secret will manage.
- The "owner" of CA-Top Secret security and recovery files (usually the person designated as the Master Central Administrator - MSCA).
- The security file encryption key.

Chapter 2. System Requirements

CA-Top Secret is a software system designed to provide complete data security for the VSE installation.

CA-Top Secret security can be imposed automatically -- without any program modifications -- upon all resources which are accessed using standard access methods. Protected resources can include:

- programs
- transactions
- data files (tape and disk, all access methods)
- DL/1 segments and PSBs
- CICS transient data and temporary storage elements

In addition, comprehensive user interfaces are provided to allow the more sophisticated installation to include standard seurity checking in the design of applications, and to allow the user to designate any element of any data structure as a protected resource.

CA-Top Secret also provides complete control over the use of terminal and partition facilities and can be used to restrict their use to specific purposes, by specific users, by day, and even by time of day.

Definition and maintenance of the security standards may be done using a command-driven CICS transaction, or using a command-driven batch utility program. Access to the security database and to the log file, which records all attempted violations, is also provided both online, with full browse and selection capabilities, and via standard reports.

In summary, CA-Top Secret provides the complete answer to the problem of data and resource security for the VSE user. And since it was developed as an integral part of the CA family of software products, CA-Top Secret functions smoothly and efficiently to assure the maximum benefit for the minimum overhead.

2.1 Supervisor Requirements

For:	The minimum requirement is:
Batch Security	VSE/ESA version 1.4 and above. For IBM, purchased program VSE/ESA 2.4 and above.
online security	CICS/VS Release 2.3 or above
ICCF support	ICCF Release 2 or above.

CA-Top Secret requires no source or object modifications to the supervisor System Control Program or other IBM components. All interactions with CICS or batch processing are transparent to the user unless a security violation is detected for a resource which has been defined as protected.

2.1.1 For VSE/ESA Version 1.4 and Above

2.1.1.1 SYS Statement

CA-Top Secret requires the following value in the SYS statement during system initialization:

- JA=YES in order to guarantee automatic signoff at end-of-job.

2.1.2 Security at the Library, Sublibrary, and Member Level

CA-Top Secret provides full library, sublibrary, and member level security by interfacing with the VSE/ESA librarian component and other IBM components that access libraries. This is accomplished by intercepting the security checks that are issued by the librarian components. Since the IBM components issue security checks only when IBM security is activated, it is necessary to activate IBM security during IPL.

After CA-Top Secret has been activated, the IBM security checking routines will be disabled and all IBM issued security checks will be processed entirely by CA-Top Secret.

These three steps must be performed to successfully activate IBM security for use with CA-Top Secret:

1. Verify that the IBM security phases are cataloged in SYSRES.
2. Assemble and link edit a dummy IBM security table.
3. Update the ASI procedure to activate IBM security.

Each step is described in detail in the following section.

2.1.2.1 Step 1: Verify IBM security phases are cataloged in SYSRES.

The IBM security phases must reside in the system library IJSYSRS.SYSLIB. The IPL process will automatically load into the SVA the security phases that require SVA residency. The following is a list of the required phases:

```
DTSECJCL.PHASE
DTSECAPP.PHASE
DTSECSVC.PHASE
DTSLOGON.PHASE
$IJBXPCA
$IJBRCFA
$IJBRCFB
$IJBRCFC
```

Note: You need not be concerned if you cannot find all of the phases in this list, the availability of certain phases is dependent on the release level of VSE installed.

2.1.2.2 Step 2: Assemble and link edit a dummy IBM security table.

IBM security cannot be activated unless a security table has been assembled, link edited, and cataloged into the system sublibrary IJSYSRS.SYSLIB. This phase is loaded into the SVA during IPL and will control security until CA-Top Secret has been activated. The following sample job will assemble the minimum security table required to allow the system to be successfully IPLed and CA-Top Secret to be activated.

Additional entries for other libraries such as PRD1 and PRD2 might be needed, check the IBM sample DTSECTAB located in IJSYSRS.SYSLIB. Please notice, any definitions in the DTSECTAB **must** be defined with UACC=CON. Any other UACC right will prevent IBM security checks from being issued after TSS/VSE is activated.

```
// JOB DTSECTAB ASSEMBLE IBM SECURITY TABLE
// LIBDEF PHASE,CATALOG=IJSYSRS.SYSLIB
// OPTION CATAL
// EXEC ASSEMBLY,SIZE=300K
************************************************************************
*                SYSTEM ADMINISTRATOR USER ID                          *
************************************************************************
         DTSECTAB TYPE=USER,NAME=MSCASNTL,PASSWRD=MSCASNTL,          X
               ACC=(1-32,ALT),AUTH=YES,RIGHT=BTRANS,                 X
               SUBTYPE=INITIAL
************************************************************************
*                         SYSRES LIBRARY                               *
************************************************************************
         DTSECTAB TYPE=LIB,NAME=DOSRES.VSE.SYSRES.LIBRARY.IJSYSRS,   X
               ACC=(1-32),UACC=CON
         DTSECTAB TYPE=SUBLIB,NAME=IJSYSRS.SYSLIB,                   X
               ACC=(1-32),UACC=CON
         DTSECTAB TYPE=MEMBER,NAME=IJSYSRS.SYSLIB.*,                 X
               ACC=(1-32),UACC=CON
************************************************************************
*                       CA-TOP SECRET LIBRARY                          *
************************************************************************
         DTSECTAB TYPE=LIB,NAME=vvvvvv.fffffffff.lllllll,            X
               ACC=(1-32),UACC=CON
         DTSECTAB TYPE=SUBLIB,NAME=lllllll.ssssssss,                 X
               ACC=(1-32),UACC=CON
         DTSECTAB TYPE=MEMBER,NAME=lllllll.ssssssss.*,               X
               ACC=(1-32),UACC=CON,                                  X
               SUBTYPE=FINAL
         END
/*
// EXEC LNKEDT
/&
```

Notes on IBM security table:

If CA-Top Secret was purchased from IBM and installed into the recommended PRD2.TSSVSE library, then it is recommended that the sample source DTSECTAB.Z is punched from IJSYSRS.SYSLIB and used instead of the CA provided sample.

The system administrator User ID should also be defined as a CA-Top Secret user ID. Use the same user ID and password that were specified in the IBM security table.

Replace vvvvvv with the VOLID of the DISK containing the CA-Top Secret library. NOTE that if the library resides in VSAM controlled space, vvvvvv must be specified as a single asterisk.

Replace fffffffff with the FILEID of the library containing CA-Top Secret (the data set name as it appears between the quotes on the VSE DLBL statement in STANDARD LABELS).

Replace lllllll with the name of the library containing CA-Top Secret (the DTF name of the file as specified on the VSE DLBL statement in STANDARD LABELS).

Replace ssssssss with the name of the sublibrary containing CA-Top Secret.

Replace ccccc with the name of the sublibrary containing the CA-CIS module.

If a library is defined in the DTSECTAB, all the sublibraries in that library must be defined in the DTSECTAB as well.

The following are the DLBL and EXTENT statements for a NON-VSAM space library into which CA-Top Secret has been installed. Use this example as a model for your own statements. Replace the values shown as needed.

The example assumes the library contains three sublibraries, named CALIB01.DYNAM60, CALIB01.TSSVSE30, and CALIB01.CAI90.

```
// DLBL CALIB01,'CA.LIBRARY',99/365,SD
// EXTENT ,SAFE10,1,0,30,180
```

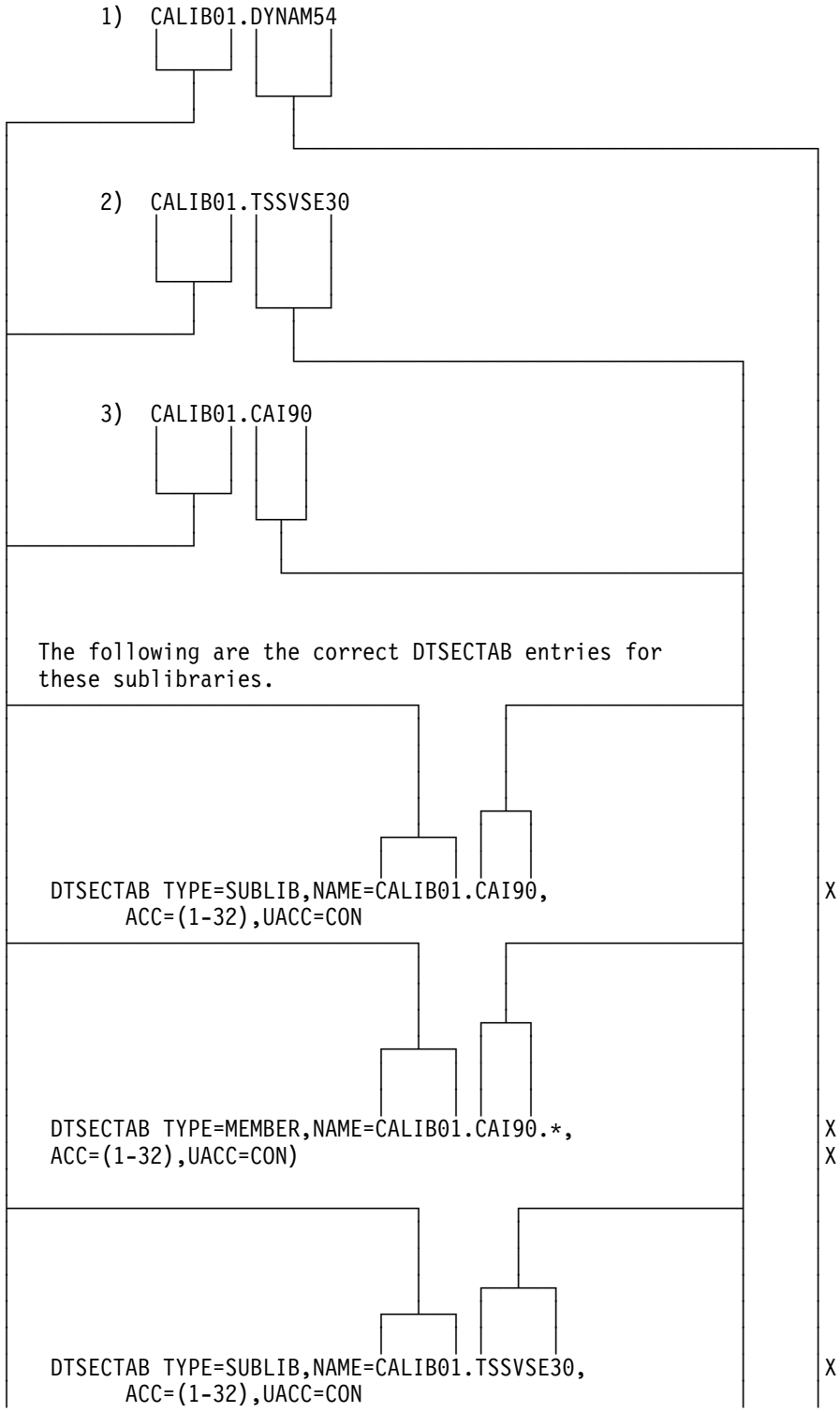
The following is the correct DTSECTAB entry for this library.

```
DTSECTAB TYPE=LIB,NAME=SAFE10.CA.LIBRARY.CALIB01,            X
      ACC=(1-32),UACC=CON
```

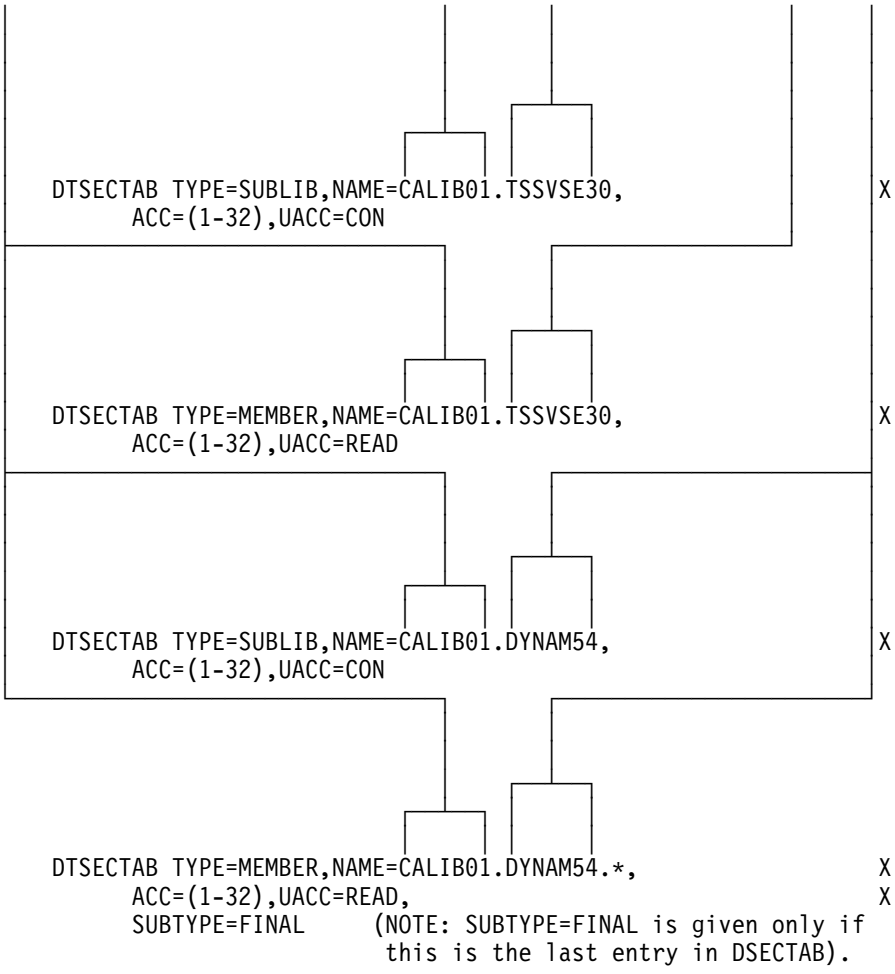
If this library were in VSAM space, the correct DTSECTAB entry would be:

```
DTSECTAB TYPE=LIB,NAME=*.CA.LIBRARY.CALIB01,                 X
```

This library contains 3 sublibraries:



Continued:


DTSECTAB TYPE=SUBLIB,NAME=CALIB01.TSSVSE30, X
ACC=(1-32),UACC=CON
DTSECTAB TYPE=MEMBER,NAME=CALIB01.TSSVSE30, X
ACC=(1-32),UACC=READ
DTSECTAB TYPE=SUBLIB,NAME=CALIB01.DYNAM54, X
ACC=(1-32),UACC=CON
DTSECTAB TYPE=MEMBER,NAME=CALIB01.DYNAM54.*, X
ACC=(1-32),UACC=READ, X
SUBTYPE=FINAL (NOTE: SUBTYPE=FINAL is given only if
this is the last entry in DSECTAB).

2.1.2.3 Step 3: Update the ASI procedure to activate IBM security.

Specify SEC=YES in the SYS IPL control statement. Before you IPL with IBM security activated, you should create an alternate ASI procedure with SEC=NO specified. Use this alternate ASI procedure for IPL to avoid IBM security cancellations.

When CA-Top Secret data sets have been initialized, all security options have been tailored, and the CA-Top Secret system has been completely installed, then IPL using the ASI procedure that has SEC=YES in the SYS IPL control statement. This activates library/sublibrary/member security.

Considerations And Restrictions

You must sign on using the system administrator user ID that was specified in the IBM security table when updating any sublibrary that was defined in the table if CA-Top Secret is not active.

If a // ID statement is required in the ASI procedure prior to CA-TOP SECRET activation (specifically for IBM native security checking), then omit the "//" on the ID statement.

Note: The user-id and password must be defined in the IBM security table (DTSECTAB).

2.2 Software Requirements

CA-Top Secret is fully integrated with the other Computer Associates file and job management software available for VSE users. CA-Top Secret is compatible with:

- CA-DYNAM Release 6.0 and above
- CA-SCHEDULER Release 7.3 and above

2.3 Installing Cumulative Maintenance

A cumulative maintenance tape is distributed periodically on a standard label cart by MSHP. See the CA-CIS maintenance tape.

If CA-Top Secret was purchased through IBM, then product maintenance will be provided by IBM as a part of VSE/ESA release updates. CA will still provide maintenance to the CA-CIS component on the regular cumulative maintenance tapes.

Chapter 3. VSE Installation Steps

If CA-Top Secret was purchased from IBM, then the product should be installed using the IUI panels provided from IBM for Optional Program products. After the product has been installed, proceed to Step 6 of these install procedures.

Computer Associates has developed standardized procedures for installing products using the MSHP utility. These standards for product installation have been developed to allow a common method of installation for all Computer Associates VSE products.

The installation process utilizes two types of history files and libraries: Production and Installation.

1. Production Libraries and History File

   The Production libraries and history file are created when the first CAI product tape is installed utilizing this standard. This library (or library set) and history file are designed to contain all Computer Associates VSE production products. Thus, allocate sufficient space for all VSE products you expect to install, even if only one product is being initially installed. Thereafter, when you install or reinstall a product, merge the new Production library set and history file only after the testing is complete.

2. Installation Libraries and History File

   The Installation libraries and history file are used for subsequent product installation. These libraries and history file are used for product installation, verification, and testing to avoid installation of a new product, or new release of an existing product, directly into the user's production environment. Each product tape that is installed will create separate libraries and a history file unique for that product tape. Once all testing has been completed, you can merge the product into the Production libraries and history file.

These two different library sets require two different installation procedures and two different sets of JCL: one procedure and JCL set is for the first time a CA product is installed and the second for installing any subsequent CA products.

When the first CA product tape is installed, one job creates the Production library (or library set) and history file, and installs the product(s) into same. When a subsequent CA product tape is installed, the Installation library (or library set) and history file are created and used for testing. Once tested, the product is migrated into the production environment and the Installation library set is then deleted.

The standard sequence for product installation is as follows:

1. Retrieving the initial install JCL. Use the supplied JCL example to extract CAINSTB1 from the install tape. The JCL example is shown in the "Install CAI Product Installation CAINSTB1" section of the Appendix.
2. Modifying the initial install JCL.

   Modify the variables in this JCL using an editor. A worksheet is provided defining the variables which must be provided (VOLSER, beginning block or track, etc.; a total of 19 variables for VSE Version 2).
3. Installing the product(s).

   Submit the modified MSHP job to create the Computer Associates production library(s), and install the product(s) from tape into them.
4. Tailoring and verifying the product(s).

   Proceed with product tailoring and verification.

3.1 Standard Installation JCL

Standard Installation JCL has been provided and must be used for installation of all Computer Associates VSE products. This JCL is described in detail in this manual's Appendix.

The following table lists the JCL members and information associated with each:

Job Name	VSE Version	Where Resides	Source Member	Description
CAINSTB1	Ver 2	tape file 8	N/A	Used to perform the initial CAI product installation utilizing this standard.
CAINSTB2	Ver 2	CA-CIS tape	CAINSTB2.Z	Used to perform all subsequent CAI product installations.
CAINSTB3	Ver 2	CA-CIS tape	CAINSTB3.Z	Used to merge products into the Production library(s) and history file when installed with CAINSTB2.

3.2 Installation Overview Checklist

Use the following checklist to track your progress through the installation process. These steps are described in detail in this chapter. Please refer to the step number in question if you find you must call Computer Associates Customer Support for assistance during the installation process.

- Step 1. Review System Requirements
- Step 2. Complete the Installation Worksheet
- Step 3. Install CA-CIS Services for VSE Tape
- Step 4. Restore Product Libraries
- Step 5. Install Distribution Tape VIA MSHP
- Step 6. Modify Source Books
- Step 7. Update ASI Procedures
- Step 8. IPL the VSE System
- Step 9. Allocate and Initialize Product Data Sets
- Step 10. CAIACMD Automatic Commands File
- Step 11. Tailor the Parameter File
- Step 12. Assign a Unique Customer Encryption Key
- Step 13. Create a Security File
- Step 14. Create the Backup Security File on DASD
- Step 15. Create the Recovery File
- Step 16. Create the Audit/Tracking File
- Step 17. Create an Alternate Audit/Tracking File (CAIAUD2)
- Step 18. Create the CPF Recovery File
- Step 19. Set Backup, Restore, and Recovery Features
- Step 20. Tailor Facility Specific Security

- Step 21. Tailor TSS.PROC and TSSB.PROC

3.3 Step 1: Review System Requirements

Before attempting to complete any of the following installation steps for CA-Top Secret, the VSE environment must be generated to meet the requirements of the CA-Top Secret product. To assure successful operation of CA-Top Secret after completion of this installation procedure, perform the following tasks in sequence.

3.3.1 Task 1A: Verify CA-Top Secret System Requirements.

Review Chapter 2, System Requirements, comparing the system specifications given there with those specified in your current VSE system generation. Note any variation from the system requirements given in Chapter 2.

3.3.2 Task 1B: Regenerate VSE if Required.

If any changes to your system generation are indicated in Task 1A, make those changes using your system software maintenance facility. If changes are required, you must IPL the new system during Step 7 before a successful result can be expected from this installation.

3.3.3 Task 1C: Review Installation Materials.

Review Chapter 3, Installation Materials, making sure you have received all of the required documentation and installation tape(s). If any discrepancies are noted, contact your Computer Associates representative to request the missing items before proceeding with this installation.

3.4 Step 2: Complete The Installation Worksheet

Several questions concerning the environment in which CA-Top Secret will be installed should be answered before proceeding with the installation process.

- Which DASD packs will be used to hold libraries and installation files?
- Which file IDs will be used for libraries and installation files?
- What extent information will be used for libraries and installation files?

The worksheet on the following pages should be completed before continuing any further with the installation procedure. The keywords on the worksheet are the same as the symbolic parameters used in the supplied installation JCL. These keywords will then be used to update the sample installation JCL for proper execution in your environment.

3.4.1 Task 2A: Common Component, Libraries, and MSHP Install

For installation of CA-Top Secret, the library requirements on a given device are as follows:

Disk Device	Number of Tracks
FBA	5000 Blocks
3375	100
3380	81
3390	76

Library Blocks - 2500

Note: Library requirements for additional Computer Associates products must be added to those above.

The calculated file sizes should be used to complete the worksheet that follows.

Standard Product Installation Worksheet (1 of 2)

	DESCRIPTION:	**KEYWORD:**
1.	Supply the following information used to personalize the Computer Associates Production History File:	
	Customer Name	1. @CUSTNME=_____
	Customer Address	@CUSTADD=_____
	Customer Phone Number	@CUSTPHN=_____
	Programmer Name	@PROGNME=_____
2.	Supply the following information used for the Production History File EXTENT:	
	Volume ID of DASD pack	2. @HISTVOL=_____
	Beginning relative track or block	@HISTREL=_____
	Number of tracks or blocks	@HISTEXT=_____
3.	Supply the following information used for the Install History File EXTENT:	
	Volume ID of DASD pack	3. @INSTVOL=_____
	Beginning relative track or block	@INSTREL=_____
	Number of tracks or blocks	@INSTEXT=_____
4.	Supply the tape drive address where the installation tape will be mounted:	4. @TAPECUU=_____
5.	Supply the following information used for the Production Library EXTENT:	
	Volume ID of DASD pack	5. @DLIBVOL=_____
	Beginning relative track or block	@DLIBREL=_____
	Number of tracks or blocks	@DLIBEXT=_____

Standard Product Installation Worksheet (2 of 2)

DESCRIPTION: **KEYWORD:**

6. Supply the following information used for the Install Library:

Volume ID of DASD pack	6. @ILIBVOL=_____
Beginning relative track or block	@ILIBREL=_____
Number of tracks or blocks	@ILIBEXT=_____

7. Supply the product name and product code you are installing:

Product Name:	TOP SECRET	7. @PRODUCT=_____
Product Code:	KD30n *	@PRODCDE=_____

Note: The last digit of the product code may vary from one genlevel to another. You should use the last 5 digits of the genlevel from the tape you are installing.

3.5 Step 3: Install CA-CIS Services for VSE Tape

CA-Top Secret requires the installation of CA-CIS Services for VSE tape at a minimum Release of 1.4. If it is already installed, proceed to the next step.

Consult the CA-CIS documentation for details on performing this step.

Note: To avoid "down leveling" any services, remember the following:

For a given release of CA-CIS, never install a genlevel lower than the one already installed. For example, if Release 1.2, genlevel 9207 is already installed, do not install Release 1.2, genlevel 9204.

Never install a release lower than the one previously installed. For example, if Release 1.3 is already installed, do not install Release 1.2.

3.6 Step 4: Restore Product Libraries

Note: This step applies to initial installations only.

This step will create and load the CA-Top Secret product libraries from tape using MSHP. Both tasks are required only for the first CA product installed at your site.

3.6.1 Task 4A: Extract Initial Install JCL

The initial product installation JCL creates the Production libraries and history file and installs the product into these newly created libraries and history file. This JCL is used only for the first CA product installed at a site.

- For VSE/ESA Version 1.4 and above

The following JCL is to be used to extract CAINSTB1 from the installation tape and catalog it to an existing user library for VSE Version 2 systems:

```
// JOB     CAINSTB0          CATAL INSTALL JCL TO LIBRARY - VSE VERSION 2
// SETPARM LIBNAME=@LIBNAME  ** replace @LIBNAME w/ library name
// SETPARM SUBNAME=@SUBNAME  ** replace @SUBNAME w/ sublibrary name
// SETPARM TAPECUU=@TAPECUU  ** replace @TAPECUU w/ install tape address
   MTC     FSF,&TAPECUU,7
// ASSGN   SYSIPT,&TAPECUU
// EXEC    MSHP,SIZE=256K,PARM='ACCESS SUBLIB=&LIBNAME..&SUBNAME'
// RESET   SYSIPT
/&
```

3.6.2 Task 4B: Install Distribution Tape Via MSHP

In this task, the installation JCL member CAINSTB1 should be extracted from the "Z" sublibrary of the target library specified in Task 4A. The JCL should be edited and modified using the worksheet items located in the Appendix for the appropriate JCL member. Please adhere to the following guidelines when editing JCL into execution JCL:

1. Be sure to limit the scope of editor changes to columns 1 to 71 of the skeleton JCL.

After all modifications are complete, mount the installation tape on the specified tape drive and submit the JCL for execution.

3.6.3 Task 4C: Install CA-CIS Tape Via MSHP

The CA-CIS tape consists of components common to all Computer Associates VSE Products. If you have already installed the latest CA-CIS Services tape as part of another product's installation step, you may skip this task.

Installation of the CA-CIS tape can be done with JCL member CAINSTB2.Z or CAINSTQ2.Z.

For more information, refer to the *CA-CIS Services Installation Guide.*

3.7 Step 5: Install Distribution Tape Via MSHP

Note: This step applies to subsequent installations only.

In this task, the installation JCL member CAINSTB2 for VSE Version 2 users should be extracted from the "Z" sublibrary of the CAI Product Library that contains previously installed CAI products. The JCL should be edited and modified using the worksheet items located in the Appendix for the appropriate JCL member. Please adhere to the following guidelines when editing JCL into execution JCL:

1. Be sure to limit the scope of editor changes to columns 1 to 71 of the skeleton JCL.

After all modifications are complete, mount the installation tape on the specified tape drive and submit the JCL for execution.

3.8 Step 6: Modify Source Books

This step will punch the skeleton sample JCL required to complete the installation and customize the skeleton JCL into executable JCL according to the specifications supplied in the worksheet completed in Step 3.

3.8.1 Task 6A: Punch Sample JCL

Your CA sample JCL found in the source sublibrary Z includes all job streams needed in order to proceed with the CA-Top Secret installation. The tables on the next two pages show each member name, an indication of whether the member is required or optional, and a brief description. All steps apply to VSE/ESA.

Member (in Z sublib)	Required or Optional	Description
TSSLABEL	Required	Defines label areas to be defined in VSE standard label area
TSSPARMS	Required	DEfines CA-Top Secret security options
TSSMAIND	Required	Calculates space needed for security database
TSSMAINS	Required	Defines CA-Top Secret security database
TSSMAINB	Optional	Defines CA-Top Secret backup security file
TSSMAINR	Optional	Defines CA-Top Secret recovery file
TSSKEY	Required	Defines CA-Top Secret unique customer encryption key
TSSMAINA	Optional	Defines CA-Top Secret primary and secondary audit files
TSSMAINC	Optional	Defines CA-Top Secret CPF recovery file
TSSINIT	Optional	Sample TSS definitions
TSSAUTO.Z	Optional	Sample JCL used to define CA-Top Secret auto commands
TSS.PROC	Required	TSS startup proc
TSSB.PROC	Optional	TSS emergency startup proc
TSSAUDIT	Optional	Sample JCL used to run TSS AUDIT report

Member (in Z sublib)	Required or Optional	Description
TSSUTIL	Optional	Sample JCL used to run TSSUTIL security violation report

Use the following JCL to punch the sample JCL for VSE Version 2 installation:

```
// JOB    PUNCH TOP SECRET INSTALL JCL
*
*  INCLUDE APPROPRIATE DLBL AND EXTENT INFORMATION FOR THE LIBRARY
*  THAT THE PRODUCT WAS INSTALLED TO HERE.
*
// EXEC   LIBR
*  INCLUDE APPROPRIATE ACCESS LIBRARIAN COMMAND HERE.

 PUNCH  TSSLABEL.Z
 PUNCH  TSSPARMS.Z
 PUNCH  TSSMAIND.Z
 PUNCH  TSSMAINS.Z
 PUNCH  TSSKEY.Z
 PUNCH  TSSMAINA.Z
 PUNCH  TSSMNGR.Z
 PUNCH  TSS.PROC
 PUNCH  TSSB.PROC

/*
/&
```

3.8.2 Task 6B: Update Skeleton JCL

Customize the skeleton JCL that was punched in the previous task according to the installation requirements determined in Step 3, the Installation Worksheet.

Please adhere to the following guidelines when editing JCL into execution JCL:

1. Edit the output from Task 6A as a single file that will be used to perform the changes determined by the worksheet.
2. Limit the scope of editor changes to columns 1 to 71 of the skeleton JCL.

3.9 Step 7: Update ASI Procedures

3.9.1 Task 7A: Define CA-LMP Execution Key or IBM Security Keys

Customers who purchased CA-Top Secret from Computer Associates should obtain and define a CA-LMP key for the licensed product. Please see the *CA-CIS Installation Guide for Details*.

Customers who purchased CA-Top Secret from IBM as a part of the VSE/ESA 2.4 product solution, should obtain and define the IBM product key and IBM customer ID in the TSSPARMS copybook provided. The following lines should be inserted in the provided sample copybook for TSSPARMS:

- IBMCUST(..IBM Customer Number)
- IBMKEY(..IBM Product Specific Key)

CA-Top Secret VSE will initiate in a Global Warning Mode unless either a valid CA-LMP or IBM key has been detected.

3.9.2 Task 7B: Storage Requirements

CA-Top Secret for VSE 3.0 requires the following storage to run. The values include storage needed for CA-CIS services to execute.

SVA CA-Top Secret for VSE with CA-CIS uses approximately: 380k of 24-Bit SVA storage and 880K of 31-Bit SVA storage plus 1204K of 31-Bit SVA storage after TSSMNGR has executed and CA-Top Secret is completely activated.

CA-Top Secret partition
The CA-Top Secret partition uses approximately 1400k of partition getvis storage (combined of 24-Bit and 31-Bit storage, and requires an execute size of 128k.

The storage values vary depending upon number of user's defined, number of resources protected and could be higher or lower at various times.

User address space
Requirements vary depending upon the size of a user's Security Record(s). For a single-user address space, typically 1K is required per user. The minimum is 464 bytes. In a VSE/ESA environment the user key storage is allocated in GETVIS above the 16 megabyte line.

For multi-user address spaces (such as CICS), the average is 250 bytes for each user. User information is freed at signoff. Profile information is freed when the last user of the profile signs off at address space termination.

3.9.3 Task 7C: Update Standard Labels

CA-Top Secret requires that the DLBL and EXTENT information for the security files to be placed in standard labels. The logical unit(s) that are chosen for the CA-Top Secret data sets **must be permanently assigned** in all VSE partitions.

Use TSSLABEL.Z to insert in STDLABEL proc.

The following CA-Top Secret labels are shown as examples only. They must be tailored to meet your own site requirements. For information on tailoring these labels, refer to the notes that follow.

```
// DLBL CAIPRM1,'CAI.TOP.SECRET.PARM.FILE',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,5
// DLBL CAIACMD,'CAI.TOP.SECRET.AUTOCMD.FILE',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,5
// DLBL CAIAUD1,'CAI.TOP.SECRET.AUDIT1',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,105
// DLBL CAIAUD2,'CAI.TOP.SECRET.AUDIT2',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,105
// DLBL CAIRCVF,'CAI.TOP.SECRET.RECFIL',99/365
// EXTENT SYS0XX,YYYYYY1,0,Z,30
// DLBL CAICPFR,'CAI.TOP.SECRET.CPFFIL',99/365
// EXTENT SYS0XX,YYYYYY1,0,Z,30
```

Note: Alter entries as follows:

SYSOXX Your correct programmer logical unit.

YYYYYY The correct DASD volumes.

Z The starting number of the tracks or blocks.

3.9.4 Task 7D: Update LIBDEF Information

CA-Top Secret requires that the library into which it has been installed be added to the **permanent default LIBDEF SEARCH group** for ***all*** partitions.

Update the ASI LIBDEF information so that the Library.sublibrary for CA-Top Secret and the sublibrary for CA-CIS is available for all partitions.

3.9.5 Task 7E: Update SVA Statement

Update the SVA statement so that sufficient System GETVIS exists for the CA-Top Secret VSE system and the CA-CIS Services to function.

3.9.5.1 SVA and GETVIS Components

The following tables give the storage requirements for the various components that are installed with CA-Top Secret.

System Component	Partition GETVIS	System GETVIS
CA-Top Secret	50K	200K

- The partition GETVIS requirement may increase based on the selection of certain product options such as Security Checking Performance.
- Your site may require additional system GETVIS for internal tables. This is a minimum requirement for program storage.

3.9.6 Task 7F: Insert EXECUTE For CASAUTIL

There are two methods for activating CA-Top Secret. You can choose to activate only CA-Top Secret, or to activate all CA VSE products that run under System Adapter.

To activate only CA-Top Secret, add the following statement to the ASI procedure:

```
// EXEC CASAUTIL
   START TOPSECRET ESI/VSE ENF/VSE
   START INITIAL
/*
```

If you want to activate all CA VSE products that run under System Adapter, add the following statement to the ASI procedure:

```
// UPSI 00000000
// EXEC CASAUTIL
/*
```

The placement of CASAUTIL in the IPL procedure depends upon which release of VSE/ESA is being used. See *CA-CIS Systems Programs Guide* for details.

Running VSE/ESA Release 1.4 up to VSE/ESA Release 2.3: CASAUTIL should be placed into the IPL procedure immediately prior to the start of power.

```
// LIBDEF PHASE,SEARCH=(CAI2.CA90S,CAI2.TSSVSE30)
// ASSGN SYSLST,IGN
// EXEC CASAUTIL,SIZE=800K
/*
ASSGN SYSLST,UA
ASSGN SYSPCH,UA
START F1
STOP
.
.
```

Running VSE/ESA release 2.4 and above: CASAUTIL should be inserted into the IPL procedure prior to the first execute of BSSINIT.

```
// GOTO PWRSTRT
// LIBDEF PHASE,SEARCH=(CAI2.CA90S,CAI2.TSSVSE30)
// ASSGN SYS026,DISK,VOL=TSSVOL,SHR (permanent assign to DASD holding TSS catalog)
// EXEC CASAUTIL,SIZE=800K
// EXEC BSSINIT
/*
```

Note: For more information on using CASAUTIL, refer to the "Utilities" chapter in the *Systems Programmer Guide.*

3.10 Step 8: IPL the VSE System

IPL the VSE system. This enables the CA-Top Secret VSE components that are required to complete the installation process.

3.11 Step 9: Allocate and Initialize Product Data Sets

In this step you will define and initialize the CA-Top Secret data sets, perform any conversions from previous releases, and activate the security system.

3.11.1 Task 9A: Define CA-Top Secret Data Sets to System

CA-Top Secret consists of multiple files:

DTF Filename:	Function:
CAISECF	Security database
CAIBKUP	Security backup database
CAIPRM1	Security parameter file
CAIACMD	Security auto command file
CAIAUD1	Security primary audit file
CAIAUD2	Security secondary audit file
CAIRCVF	Security recovery file
CAICPFR	Security CPF recover file

These files may be accessed using any desired logical unit, but whichever you choose they must be permanently assigned to the correct DASD device at all times.

Specify labels for the CA-Top Secret data sets. The labels for these data sets must contain their actual extent information. These data sets must be aligned on cylinder boundaries, but they must not be allocated by a dynamic disk space management system.

The system logical unit, SYSRES, may not be used for the CA-Top Secret catalog files. The files may reside on the same physical volume as the system residence area, but another logical unit must be used to access them. This is an undocumented IBM restriction which may be removed in a later release (or put level) of VSE.

To provide for faster access, the locations of the catalog files are saved when they are accessed for the first time following IPL. Only one set of files may be used in a particular system, and they must not be moved except through the use of the appropriate TSSXTEND functions which cause the saved location to be updated.

3.11.1.1 Recommendations

Place file on different DASD, but always use the same type DASD.

If programmer logical units are used, they should be reserved for CA-Top Secret use only and never be unassigned or reassigned.

DLBL and EXTENT information should be cataloged in the system standard label area.

3.11.2 Task 9B: Determine Where to Place the Catalog Files

During normal operation, the security database (CAISECF) is accessed only during signon and signoff and when a violation occurs. Since the access to this file should be relatively small, the placement of this file is not critical to performance.

The primary security file and optional backup security file should be placed on different DASD, but always use the same type of DASD and and identical allocation size in tracks, blocks and cylinders for both files. It is recommended that the optioanl security recovery files are defined on a DASD different from the above.

The locating of the parameter, audit, CPF recovery, and auto command files can be allocated on the same or different DASD. Again, it is recommended that all files should be allocated on the same type of DASD.

3.12 Step 10: CAIACMD Automatic Commands File

To ensure that security is completely initialized before any jobs start processing, CA-Top Secret should be the first task to execute after CASAUTIL initialization. This can be accomplished through the use of the Automatic Commands File.

The Automatic Commands File, CAIACMD, will be used to automatically start commands once CA-Top Secret has been initialized.

The installation of CA-Top Secret created a Z copybook called TSSAUTO.Z in the Installation Library. Edit TSSAUTO.Z to include only the START commands to be executed after power initiation.

CA-Top Secret will execute the commands upon the first invocation of CA-Top Secret on the CPU. If CA-Top Secret is restarted, the commands file will be ignored. If CA-Top Secret abends during initialization and the automatic commands have not yet been issued, then CA-Top Secret will attempt to issue them before terminating.

Note: CA-Top Secret initialization must be complete before online systems such as CICS can be started. Complete initialization is indicated by the CA-Top Secret message number TSS9000I.

If CA-Top Secret is not the first task started after CASAUTIL, all tasks that reference "secured" data sets are abended by native security if selected.

File Characteristics: The Automatic Commands File is a standard sequential data set with the following attributes:

LRECL=80
RECFM=F[B]

The Automatic Commands File will hold any O/S commands that may be started as part of the normal IPL procedure. The file can also contain comment cards which must be identified by an asterisk **(*)** in column 1. The following example shows the content of a typical Automatic Commands File.

```
*
* AUTOMATIC IPL COMMANDS
*
R RDR,CICSPROD
R RDR,VTAMSTRT
```

Note: It can reside in any DASD. However, be sure to protect this file to avoid possible tampering.

3.13 Step 11: Tailor the Parameter File

The CA-Top Secret control options that define an installation's environment can be placed in the CA-Top Secret Parameter File.

The download procedure creates a Parameter File; its member name is **TSSPARMS.Z** Edit the TSSPARMS control options to agree with your site's standards.

Note: Be sure to protect this file to avoid possible tampering.

File Characteristics: The Parameter File is a standard sequential data set or member of a library with the following DCB attributes:

LRECL=80
RECFM=F(B)

The Parameter File can contain any number of records. It is identified to CA-Top Secret by the CAIPRM1 statement in the STD label area.

Note: The order in which control options are placed in the Parameter File is important. Control options are processed **sequentially.** See the heading entitled "Sequential Processing", later in this chapter.

3.13.1.1 Rules for Creating the Parameter File

The following paragraphs outline the rules for creating a Parameter File.

```
*
* SECURITY CONTROL OPTIONS FOR TEST MACHINE
*
MODE(WARN)                    * SELECT PROCESSING MODE
BACKUP(0400)                  * 4 IN THE MORNING
NEWPW(MIN=5,MASK=CVCVC)       * NEW PASSWORD MASKING
```

3.13.1.2 Contents of a Typical Parameter File

Comment Statements: Comment statements are identified by an asterisk, **(*),** in the first position. For example, the following line (extracted from "Rules for Creating the Parameter File" above, is a comment statement:

*** SECURITY CONTROL OPTIONS FOR TEST MACHINE**

An asterisk also ends a control statement and allows a comment to be placed on the line (following the asterisk). The following line demonstrates this:

MODE(WARN) * SELECT PROCESSING MODE

Where:

MODE(WARN)	is the control option
*	ends the control options and marks the beginning of a comment,
SELECT PROCESSING MODE	is the comment

Control Options: Control options may be placed anywhere between positions 1 through 70, generally, one per line. Free format can be used, with or without commas as separators. Control options are processed sequentially making the order in which they are listed important (see the heading entitled "Sequential Processing").

There are defaults for control options. If a control option is coded incorrectly, the default is used. Be certain you understand the syntax of control options, especially MODE which defaults to FAIL. A maximum of 2000 entries can be made in the Parameter File.

Your initial review of control options should include those listed in *Table 2-8: Control Options Initial Review*, below.

Table 3-1 (Page 1 of 2). Control Options Initial Review

Option	Description	Default
MODE	Sets global security mode.	(FAIL)
LOG	Controls recording of security events.	(MSG,SEC9,SMF,INIT)
DOWN	Selects DOWN options for TSS.	(SB,TW,BW,OW)
FACILITY	Selects options per facility.	See the *Control Options Guide*
JOBACID	Locates ACID on BATCH job card.	(U,7)

Table 3-1 (Page 2 of 2). Control Options Initial Review		
Option	**Description**	**Default**
NEWPW	Sets specifications for new passwords.	(MIN=4,NR,NV,ID,RS,TS MINDAYS=1,WARN=3)
SUBACID	Controls derivation of BATCH job ACIDs submitted through the internal reader.	(U,7)

3.13.1.3 Sequential Processing

Control options are processed sequentially in the order they are placed in the Parameter File. Options can override previous options. For example, to override specific facilities, the FACILITY option must be placed **after** control options such as MODE and LOG.

The control option MODE appears first and sets CA-Top Secret into WARN mode for all facilities. However, the FACILITY control option specifies that, for the facility Batch, MODE=IMPLement. The FACILITY statement, because it is processed *after* MODE, overrides the CA-Top Secret mode for Batch.

Therefore, CA-Top Secret is in WARN mode for all facilities *except* Batch. Batch is in IMPLementation mode.

For more information about sequential processing, see the *Control Options Guide*.

Overrides: While we're on the subject of overrides, it seems appropriate to mention two things:

1. other ways to specify/override control options, and
2. the hierarchy of control option overrides.

In addition to the Parameter File, there is one other way to specify control options. Using the TSS MODIFY command. For example:

```
TSS MODIFY('control option')
```

Because there are two ways to enter control options, CA-Top Secret uses a hierarchy approach to determine an installation's setting.

- The Parameter File settings, which are overridden by the TSS MODIFY commands.

Parameter File control option settings remain in effect even after CA-Top Secret is reinitialized. MODIFY settings are temporary, that is, they are canceled when CA-Top Secret is brought down.

Specifying control options and the hierarchy of overrides is discussed in further detail in the *Control Options Guide.*

3.14 Step 12: Assign a Unique Customer Encryption Key

An encryption key, unique to each customer site, is used to encrypt user-oriented information on the Security File. Each installation must supply their own key; no default key is assigned.

You cannot start CA-Top Secret until you have assigned a customer encryption key. It is important that the identical encryption key is maintained in all future versions of CA-Top Secret.

SAFEGUARD YOUR KEY FOREVER !!!

Note: If it is necessary, CA-Top Secret does provide a means to change the encryption key. Refer to the TSSXTEND utility in Appendix B, “TSSXTEND and TSSRECVR” on page B-1, for more details.

3.14.1 Encryption Key Format

The Encryption Key is a 16-hex digit string. For example, CAC2C4C5A1A2B1B2 is a valid encryption key.

3.14.2 Installing the Key

TSSKEY.Z is contained in member TSSKEY.Z. It installs an inline ZAP to assign your encryption key to the CA-Top Secret library.

Caution

If your site does not protect job output, you should delete the output from this step, to reduce the chance of the key value being viewed.

Edit the JCL to conform to your site standards and supply your company's unique Key in the ???????????????? field **before** submitting the job.

3.14.3 Required JCL

```
// JOB TSSKEY
// EXEC    MSHP,SIZE=800K
PATCH SUB=LIB.SUBLIB
AFF  PHASE=TSSKTPRC
ALT  000000 /8/00:????????????????
/*
/&
* $$ EOJ
```

CA-Top Secret will not initialize if the key is not supplied.

3.15 Step 13: Create the Security File

The Security File is the data base containing all security-related information about users, profiles, departments, divisions, zones, and resources. The CA-Top Secret utility, **TSSMAINT,** creates and formats the Security File.

3.15.1 File Characteristics

The size of the Security File will **not** affect CA-Top Secret's performance. Based on the supplied input parameters, TSSMAINT will automatically allocate enough space to accommodate present and future security information.

The Security File requires contiguous space on DASD. It is organized for direct access and can be located on FBA, 3350, 3375, 3380, 3390, or 9345 devices.

Based on the type of device the file resides on, TSSMAINT will determine a default block size if one is not specified. However, it is recommended that each site chose a specific blocksize to match their own environment. The default blocksize must be a multiple of 256. When calculating the optimal blocksize for your Security File, remember that the file contains keyed records with a 17-byte key.

The default parameters allow for 5,000 ACIDs and 1000 (DASD) volumes. These defaults take future requirements for the average installation into consideration.

3.15.2 Multiple CPUs

If you are using more than one CPU, place the Security File on a shared DASD volume accessible to all systems.

If sharing Security Files between multiple CPUs, the control option SHRFILE(YES) must be specified on each CPU.

3.15.3 Required JCL

You must run two JCL procedures to create the Security File. Both JCL procedures are provided as Z copybooks, named **TSSMAIND.Z** and **TSSMAINS.Z**.

3.15.3.1 TSSMAIND

Allows TSSMAINT to determine the number of blocks needed to create the Security File, you must first do a "dummy" run. The JCL for the dummy run is in member TSSMAIND.Z.

Before running TSSMAIND, edit the procedure to conform to your site's standards. The run of TSSMAIND will determine the number of blocks needed to create the Security File.

The Security File parameters must be entered one per line, starting in column one.

Enter your site's standards for the following Security File parameters: ACCESSORS, VOLUMES, and SCA.

Parameter **Function**

ACCESSORS=????? Indicates the maximum number of user, profile, department, division and zone ACIDs that will be defined to CA-Top Secret. The value entered for **?????** determines the amount of Security File space allocated to hold ACID-related security information. A minimum of 5000 is enforced. The default is 5000.

VOLUMES=?????? Indicates the number of volumes/prefixes that will be defined to CA-Top Secret. The value entered for **??????** determines the amount of Security File space allocated to hold volume-related security information. The default is 1000.

Device Type	Default BLOCKSIZE
3350	6144
3375	3840
3380	6144
3390	6144

RESBLOCKS=????? This parameter is optional. It allows you to specify the number of general resource blocks to be allocated by the system. Valid numbers are 10 - 99. The default is 20. Each owned general resource prefix requires one 16 byte entry in the index.

SDTBLOCKS=????? This parameter is optional. It allows you to specify the number of blocks allocated for the SDT. Valid numbers are 2 to 99. The default is 2. You must use this parameter to change the number of blocks to be used by the SDT.

SCA=msca/password Supplies the name and password of the Master Central Security Administrator ACID (MSCA).

msca A one- to seven-character name for the MSCA.

password A 4 to 8 character password assigned to the MSCA. The password will expire upon initial signon. The default is: **SCA=TSSSEC/TORONTO**

3.15.3.2 The TSSMAIND.Z Copybook

A copy of member TSSMAIND.Z is printed below:

```
// JOB TSSMAIND
// ASSGN SYS0XX,DISK,VOL=YYYYYY,SHR
// DLBL CAISECF,'CAI.TOP.SECRET.SECURITY.FILE',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,X
// EXEC TSSMAINT
CREATE SECURITY
ACCESSORS=?????
VOLUMES=????
RESBLOCKS=????
SDTBLOCKS=????
SCA=MSCA/MSCA
/*
```

The normal job termination for TSSMAIND is a CANCEL abend. Its only purpose is to check the size of the allocated disk extent for the Security File based on the input parameters.

3.15.3.3 TSSMAINS

After you have run the TSSMAIND procedure and determined the number of blocks needed to create the Security File, you are ready to create the Security File.

TSSMAINS.Z must first be edited to conform to your site's standards. The Security File parameters must be entered one per line, starting in column 1. Valid parameter options for Security File creation are provided below.

Parameter **Function**

CREATE SECURITY

This parameter is required. It requests Security File initialization.

ACCESSORS=????? Indicates the maximum number of user, profile, department, division and zone ACIDs that will be defined to CA-Top Secret. The value entered for ????? determines the amount of Security File space allocated to hold ACID-related security information. A minimum of 5000 is enforced. The default is 5000.

VOLUMES=?????? Indicates the number of volumes/prefixes that will be defined to CA-Top Secret. The value entered for ?????? determines the amount of Security File space allocated to hold volume-related security information. The default is 1000.

Device Type	Default BLOCKSIZE
3350	6144
3375	3840
3380	6144
3390	6144

RESBLOCKS=????? This parameter is optional. It allows you to specify the number of general resource blocks to be allocated by the system. Valid numbers are 10 - 99. The default is 20. Each owned general resource prefix requires one 16 byte entry in the index.

SDTBLOCKS=????? This parameter is optional. It allows you to specify the number of blocks allocated for the SDT. Valid numbers are 2 to 99. The default is 2. You must use this parameter to change the number of blocks to be used by the SDT.

SCA=msca/password Supplies the name and password of the Master Central Security Administrator ACID (MSCA).

msca A one- to seven-character name for the MSCA.

password A four- to eight-character password assigned to the MSCA. The password will expire upon initial signon. The default is: SCA=TSSSEC/TORONTO.

ID=iiiiiiii This parameter option is required. The characters entered for iiiiiii (up to eight characters) are placed in the master Security File and become the file's identifier. The identifier will distinguish the master Security File from the Backup File.

For the Master File, CA suggests: ID=**PRIMARY**

For the Backup File, CA suggests: ID=**BACKUP**

3.15.3.4 The TSSMAINS JCL

A copy of member TSSMAINS.Z is printed below:

```
// JOB TSSMAINS
// ASSGN SYS0XX,DISK,VOL=YYYYYY,SHR
// DLBL CAISECF,'CAI.TOP.SECRET.SECURITY.FILE',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,X
// EXEC TSSMAINT
CREATE SECURITY
ACCESSORS=?????
VOLUMES=????
RESBLOCKS=????
SDTBLOCKS=????
SCA=MSCA/MSCA
ID==PRIMARY
/*
```

3.16 Step 14: Create a Backup Security File on DASD

It is recommended that the built-in automatic backup feature be used to backup the Security File.

The Backup file should not be shared between CPUs. Place the Backup File on a DASD that is NOT the same volume, unit, channel path, etc., as the Security File.

The Backup File is an identical copy of the Security File having the same type DASD and extend size attributes.

To create the Backup Security File use the JCL supplied in member name **TSSMAINB.Z.** The values for the JCL parameters must be **exactly** the same as the parameter values used for TSSMAINS.Z with the exception of the **ID= parameter.**

The **ID=** parameter should clearly state that this is the *Backup* Security File. It is suggested that the ID= parameter be set to **BACKUP.**

Back up the Security File from one CPU only. For more details, refer to 4.1, “Multi-CPU Environments” on page 4-2.

3.16.1 Required JCL

The JCL needed to create the Backup Security File is supplied on the distribution tape in member name **TSSMAINB.Z.**

3.16.1.1 The TSSMAINB.Z JCL

The following is a copy of member TSSMAINB.Z:

```
// JOB TSSMAINB
// ASSGN SYS0XX,DISK,VOL=YYYYYY,SHR
// DLBL CAIBKCF,'CAI.TOP.SECRET.BACKUP.FILE',99/365
// EXTENT SYS0XX,YYYYYY,1,0,Z,X
// EXEC TSSMAINT
CREATE SECURITY
ACCESSORS=?????
VOLUMES=????
RESBLOCKS=????
SDTBLOCKS=????
SCA=MSCA/MSCA
ID=BACKUP
/*
```

3.17 Step 15: Create the Recovery File

The Recovery File records changes made to the Security File. This file is a "wraparound" file; when the file is full, recording continues at the beginning of the file, overlaying existing data.

The CA-Top Secret utility, TSSRECVR, uses the recorded changes in the Recovery File to restore a lost Security File.

Use the TSSMAINT utility to create and format the Recovery File. Allocate enough space for a minimum of 250 blocks. Do NOT allocate the Recovery File on the same volume as the Security File (in case of loss due to hardware malfunction).

3.17.1 File Characteristics

The Recovery File's default attributes are: RECFM=FB, LRECL=1280, DSORG=PS, BLKSIZE=1280. The default size Recovery File can hold approximately 2,000 changes before a wraparound occurs.

There is no maximum to the number of blocks that may be specified for the Recovery File. However, a large Recovery File will result in a delay during the initialization of the CA-Top Secret address space every time it is started. This delay can be as much as a few minutes if the Recovery File is several thousand blocks in size. The exact length of the delay depends on the speed of the hardware device on which the Recovery Filke is located. In general, the delay will be proportional to the size of the recovery file.

The size of the Recovery File depends upon the interval between Security File backups. The file should be made large enough to record two to three days of changes for every day in the Security File backup period.

For example, if the Security File is backed up at the end of each day, the Recovery File should be large enough to accommodate at least two days of changes.

The default size Recovery File can hold approximately 2,000 changes before wraparound.

3.17.2 Multiple CPUs

The Recovery File may be shared between systems. However, identical sharing must exist for **both** the Recovery and Security Files. The systems which share a Recovery File must be using the same Security File to ensure proper serialization and management of the shared Recovery File.

3.17.3 Required JCL

The JCL required to create the Recovery File is in member name **TSSMAINR.Z.** on the distribution tape. Before running TSSMAINR.Z, it must be edited to conform to your site's standards.

3.17.4 JCL Parameters for the Recovery File

The Recovery File parameters must be entered one per line, starting in column 1.

Valid JCL parameters for Recovery File creation are: CREATE RECOVERY and BLOCKS.

Parameter	Function
CREATE RECOVERY	Requests Recovery File initialization.
BLOCKS=????	Specifies the number of blocks to be used for the Recovery File. The default is 250.

3.17.4.1 The TSSMAINR JCL

The following is a copy of the TSSMAINR JCL:

```
// JOB TSSMAINR
// EXEC TSSMAINT
CREATE RECOVERY
BLOCKS=????
/*
```

3.18 Step 16: Create the Audit/Tracking File

The Audit/Tracking File is an optional online file that records security incidents.

The Audit/Tracking File provides administrators and auditors with a current, online record of system security activity from all CPUs.

The Audit/Tracking File is a wraparound file; when the file is full, recording continues at the beginning of the file, overlaying existing data. Optionally, two Audit/Tracking Files may be used. If this is the case, when the first Audit/Tracking File is full, CA-Top Secret will automatically switch to the alternate Audit/Tracking File. When the alternate Audit/Tracking File is full, recording continues at the beginning of the first Audit/Tracking File, overlaying existing data.

The procedures for creating an alternate Audit/Tracking File are detailed in Step 17.

3.18.1 File Characteristics

The Audit/Tracking File's attributes are fixed at: RECFM=FB, LRECL=256.

3.18.2 Required JCL

The JCL required to create the Audit/Tracking File is in its member name, **TSSMAINA.Z**. Before running TSSMAINA.Z, it must be edited to conform to your site's standards.

3.18.2.1 JCL Parameters for the Audit/Tracking File

The Audit/Tracking File parameters must be entered one per line, starting in column 1. Valid JCL parameters for Audit/Tracking File creation are: CREATE AUDIT and BLOCKS.

Parameter	Function
CREATE AUDIT	Requests Audit/Tracking File initialization.
BLOCKS=????	Specifies the number of blocks to be used for the Audit/Tracking File.
ID=AUDIT	Distinguishes one Audit/Tracking File from the other when using alternating Audit/Tracking Files. You can only specify one of the following values. No other values will be accepted. For the first Audit/Tracking File you ***must*** use: ID=AUDIT. For the alternate Audit/Tracking File you ***must*** use: ID=AUDIT2.

Note:

- If you are using an alternate Audit/Tracking File, you ***must*** code a Dataset name of CAIAUD2 in the CA-Top Secret IPL procedure.
- If you plan to use a new Audit/Tracking File for Release 3.0 and above, note that the new file cannot be shared with any CA-Top Secret MVS system prior to Release 5.0, or any CA-Top Secret VM system prior to Release 1.4. Only CA-Top Secret MVS 5.0 and CA-Top Secret VM 1.4 systems support the new 256 byte resource name capability.

3.18.2.2 The TSSMAINA.Z JCL

The following is a copy of the TSSMAINA.Z JCL:

```
 // JOB TSSMAINA
 // EXEC TSSMAINT
 CREATE AUDIT
 ID=AUDIT
 /*
// EXEC TSSMAINT
CREATE AUDIT
ID=AUTDIT2
/*
```

3.19 Step 17: Create Alternate Audit/Tracking File

In addition to the primary Audit/Tracking File, CA-Top Secret supports the use of an alternate Audit/Tracking File. Once the primary file has been filled, CA-Top Secret will then write to the alternate file.

3.19.1 Required JCL

The JCL required to create an alternate Audit/Tracking File, is found under the member name of **TSSMAINA.Z.** It is the same JCL required to create the initial Audit/Tracking File. In addition to editing TSSMAINA.Z to conform to your site's standards, you must also edit the DLBL and ID= statements so the DTF name points to CAIAUD2 and ID=AUDIT2. The TSSMAINA.Z job assumes these file labels are already placed into Standard labels using the TSSLABEL.Z sample.

3.19.1.1 JCL Parameters for the Audit/Tracking File

The Audit/Tracking File parameters must be entered one per line, starting in column 1. Valid JCL parameters for Audit/Tracking File creation are: CREATE AUDIT and BLOCKS.

3.19.1.2 The TSSMAINA.Z JCL

The following is a copy of the TSSMAINA.Z JCL:

```
 // JOB TSSMAINA
 // EXEC TSSMAINT
 CREATE AUDIT
 ID=AUDIT
 /*
// EXEC TSSMAINT
CREATE AUDIT
ID=AUDIT2
/*
```

Note: If you are using Audit/Tracking, you ***must*** code a Dataset name of CAIAUD1 in the CA-Top Secret IPL procedure.

3.20 Step 18: Create CPF Recovery File

The CPF Recovery File is used by CPF to save transmitted commands until responses to those commands are received from remote machines.

If you intend to use CPF, the CPF Recovery File must be formatted through TSSMAINT. The size of each record in the CPF Recovery File is 2048 bytes (the blocksize should be a multiple of this value). One record is written to the CPF Recovery File for each node to which the command is sent. The space is reused when a response to the command is received.

Note: The CPF Recovery File cannot be shared across multiple systems.

3.20.1 Required JCL

Sample JCL follows on the next page, and lowercase type must be replaced with the appropriate parameters for your site. The sample JCL is located in TSSMAINC.Z.

3.20.1.1 The TSSMAINC.Z JCL

The following is a copy of the TSSMAINC.Z JCL:

```
// JOB TSSMAINC
// EXEC TSSMAINT
CREATE CPF RECOVERY
/*
```

The blocksize for the file can be changed by the installation. The value chosen should be one that provides efficient utilization of the track capacity for the device on which the file resides. This varies from device to device. An exact value is not necessary as TSSMAINT will round the blocksize down to a multiple of its logical record length. The actual blocksize may be less than specified on the JCL, but it should be approximate.

A DLBL statement must be inserted into the CA-Top Secret TSS JCL to define the CPF Recovery File. A sample DLBL statement follows:

```
// DLBL CAICPFR,'CAI.TSS30.CPFROCV',,99/365
```

If the CPF Recovery File is not defined, command routing through CPF can still occur, but there will be no retransmission of unresponded commands.

If the Recovery File should happen to become temporarily filled, then message TSS9803I is written to the job LOG each time CPF needs to write a message to the file but cannot. The CPF operation continues but, in case of failure, the command cannot be recovered.

3.21 Step 19: Setup Backup, Restore, and Recovery

The importance of setting up backup, restore, and recovery procedures at installation time can not be over-stressed. The CA-Top Secret distribution tape contains backup, restore, and Security File recovery JCL procedures. The JCL procedures described in this step must be tailored to conform to your site's requirements and placed into your RDR queue so that they are available when they are needed.

3.21.1 Automatic Backup to DASD

The Security File is a critical resource and should be integrated into your standard backup process. The loss of the Security File data base would necessitate running your system without security until the Security File was rebuilt or recovered resulting in countless security exposures. To minimize the possibility of a loss of the data base, we recommend a **daily** backup of the Security File along with the use of the CA-Top Secret RECOVERY feature.

To ensure that a daily backup of the Security File is performed, CA strongly suggests that you take advantage of the CA-Top Secret Automatic Backup feature. At the time designated by the BACKUP control option, CA-Top Secret will backup the Security File to DASD. When using this feature, you can be assured that your Security File is backed up daily. Automatic Backup requires only:

1. the presence of the Recovery File and Backup Security File on DASD and,
2. the control option settings, RECOVER(ON) and BACKUP(hhmm) where hhmm represents the time the backup will occur.

3.21.2 Setup Tape Backup Procedures

All CA-Top Secret files must be copied BIT-BY-BIT. Because some standard backup and restore facilities do not copy files bit-by-bit, they **cannot** be used to backup CA-Top Secret files to tape.

Member TSSBCKUP can be used to backup the:

- Security File (not recommended, see Automatic Backup),
- Recovery File, and
- the entire Audit/Tracking File.

  Note: The Audit/Tracking File can be backed up in its entirety or it can be archived daily, weekly, monthly, etc., using the TSSARCHI JCL procedure.

To perform the backup, member TSSBCKUP employs the FCOPY utility.

TSSBCKUP should be setup to backup the Security File. If it becomes necessary to backup other CA-Top Secret files, simply specify the correct file when executing the task.

3.21.2.1 TSSBCKUP

```
TSSBCKUP
* $$ JOB JNM=TSSBCKUP,CLASS=0,DISP=D
// JOB TSSBCKUP
// UPSI 001
*
*THIS FUNCTION USES A TAPE FOR OUTPUT
*MOUNT TAPE TSS001 ON DEVICE 181
*MOUNT ALTERNATE TAPE ON DEVICE 182
*THEN CONTINUE. IF NOT POSSIBLE CANCEL THIS JOB.
// PAUSE
// ASSGN SYS005,181,00
// ASSGN SYS005,182,ALT
// ASSGN SYS004,DISK,VOL=XXXXXX,SHR
// EXEC FCOPY
DUMP FILE='CAI.TOP.SECRET.SECURITY.FILE'-
OPTIMIZE=1 NOVERIFY
/*
```

3.21.3 Setup Restore Procedures

As previously mentioned, CA-Top Secret files must be copied bit-by-bit. Although all CA-Top Secret files can be backed up, only the Security File and Recovery File can be restored. CA-Top Secret will abend if an old copy of the Audit/Tracking File is restored. If damage to the Audit/Tracking File is encountered, a new file must be initialized using the TSSMAINT Utility.

Two JCL procedures which will restore the Security File (and Recovery File) from tape to DASD are shipped with TSS/VSE.

TSSRESTR, Restore Security or Recovery File to DASD over the old file.

TSSRESTN, For use with the Security File only. Allocates space on DASD, then restores the Security File to DASD in the newly allocated space.

Member TSSRESTR employs the FCOPY utility.

TSSRESTR should be setup to restore the Security File. If it becomes necessary to restore the Recovery File, simply specify the correct file when executing the task.

Note: FBA users: The provided sample copybooks assumes the usage of CKD DASD. Please replace the FCOPY utility statement with FCOPYB for FBA. Refer to IBM FCOPY documentation about the usage of FCOPY.

3.21.3.1 TSSRESTR

```
* $$ JOB JNM=TSSRESTR,CLASS=0,DISP=D
 // JOB TSSRESTR
 // UPSI 100
 *
 *THIS FUNCTION USES A TAPE FOR INPUT
 *MOUNT TAPE TSS001 ON DEVICE 181
 *MOUNT ALTERNATE TAPE ON DEVICE 182
 *THEN CONTINUE. IF NOT POSSIBLE CANCEL THIS JOB.
 // PAUSE
 // ASSGN SYS004,181
 // ASSGN SYS004,182,ALT
 // ASSGN SYS005,DISK,VOL=XXXXXX,SHR
 // PAUSE MOUNT BACKUP DATA TAPE ON 181 FOR RESTORE OF DISK JXC356
 // EXEC FCOPY
 RESTORE FILE='CAI.TOP.SECRET.SECURITY.FILE'-
 NOVERIFY
 /*
```

The CA-Top Secret recovery file can be backed up or restored using the same copybook examples. Simply replace the name of the Security File with that of the Recovery File and catalog these under a new member name in the power reader queue.

3.21.4 Setup Recovery Procedures

The TSSRECVR Utility provides recovery processing services for the Security File. TSSRECVR uses the records kept by the Recovery File to recover all changes made to the Security File. In order for TSSRECVR to be effective, the recovery control option, RECOVER(ON), must have been in effect. RECOVER(ON) should be specified now, during installation. It can, however, be specified by an operator O/S MODIFY or TSS MODIFY command.

TSSRECVR recovers the Security File in two phases: First, TSS commands are generated from the Recovery File, then BATCH TMP executes the TSS commands. TSSRECVR requires that two JCL procedures are setup:

TSSRCVR1 Retrieves changes from the Recovery File which occurred after the time and date specified in the EXEC PARM.

You must add TIME or DATE to the EXEC PARM field before executing TSSRCVR1. One or both of these control selections can be used. The correct format is:

```
// EXEC TSSRECVR,PARM='TIME(hhmm),DATE(yyddd)'
- or -
// EXEC TSSRECVR,PARM='TIME(hhmm),DATE(-nn)'
```

Where:

hhmm Is the hour and minute for selecting recovery records. This should be the time of the last Security File backup.

yyddd Is the earliest date (in Julian format) for selecting recovery records.

-nn Is the number of previous days for which you wish to retrieve changes from the Recovery File.

To use the contents of the entire Recovery File, specify TIME(0000),DATE(00000).

TSSRCVR2 Applies the changes to the Backup Security File.

Note: During installation, the TSSRCVR1 JCL and TSSRCVR2 JCL should be setup. The TSSRECVR Utility and its use is explained in further detail in Appendix B, “TSSXTEND and TSSRECVR” on page B-1.

3.21.4.1 TSSRCVR1

```
* $$ PUN CLASS=A,RBS=12000,DEST=(*,CHRJO04)
// JOB TSSRCVR1
// ID USER=MSCA,PWD=TORONTO
// DLBL TSSCMDS,'CAI.TOP.SECRET.RECV.WORK',,0
// EXTENT SYS0XX,XXXXXX,1,0,X,Y  <-- REPLACE WITH CORRECT EXTENT INFO
// EXEC TSSRECVR,PARM='TIME(0000),DATE(00000)'
/*
```

3.21.4.2 TSSRCVR2

```
* $$ PUN CLASS=A,RBS=12000,DEST=(*,CHRJO04)
// JOB TSSRCVR2
// ID USER=MSCA,PWD=TORONTO
// DLBL TSSCMDS,'CAI.TOP.SECRET.RECV.WORK',,0
// EXTENT SYS0XX,XXXXXX,1,0,X,Y  <-- REPLACE WITH CORRECT EXTENT INFO
// UPSI 01
// EXEC TSSCMNDB
/*
```

3.22 Step 20: Tailor TSS.PROC and TSSB.PROC Startup Procedures

CA-Top Secret uses Procedure copybooks to start the security manager, using either the primary security database (TSS.PROC) or the backup security file (TSSB.PROC).

Correct the TSS.PROC copybook supplied with the product to reflect the location and extent information used when TSSMAINT ran to create the primary and backup datasets.

```
// JOB TSS
// EXEC LIBR,SIZE=768K,PARM='MSHP'
ACC S=CAILIB.TSSVSE30    catalog into CAILIB.TSSVSE30
CATALOG TSS.PROC R=Y
ASSGN SYSIN,FEC,PERM
ASSGN SYSPCH,FED
ASSGN SYSLST,FEE
ASSGN SYSxxx,DISK,VOL=xxxxxx,SHR
ASSGN SYSyyy,DISK,VOL=yyyyyy,SHR
ASSGN SYSzzz,DISK,VOL=zzzzzz,SHR    point to DASD holding AUDIT, PARM, and other TSS files
// DLBL CAISECF,'CAI.TOP.SECRET.SECURITY.FILE',99/365
// EXTENT SYS0xx,xxxxxx,1,0,x,x
// DLBL CAIBKUP,'CAI.TOP.SECRET.BACKUP.FILE',99/365
// EXTENT SYS0yy,yyyyyy,1,0,y,y
// EXEC TSSMNGR,SIZE=128K
/*
/+
```

After the procedure has been altered, it should be placed back into a VSE procedure library, using LIBR procedure.

Ths same step now has to be repeated for the TSSB.PROC copybook, which is used for recovery purposes.

```
// JOB TSS
// EXEC LIBR,SIZE=768K,PARM='MSHP'
ACC S=CAILIB.TSSVSE30    catalog into CAILIB.TSSVSE30
CATALOG TSSB.PROC R=Y
ASSGN SYSIN,FEC,PERM
ASSGN SYSPCH,FED
ASSGN SYSLST,FEE
ASSGN SYSxxx,DISK,VOL=xxxxxx,SHR
ASSGN SYSyyy,DISK,VOL=yyyyyy,SHR
ASSGN SYSzzz,DISK,VOL=zzzzzz,SHR    point to DASD holding AUDIT, PARM, and other TSS files
// DLBL CAISECF,'CAI.TOP.SECRET.BACKUP.FILE',99/365
// EXTENT SYS0yy,yyyyyy,1,0,y,y
// EXEC TSSMNGR,SIZE=128K
/*
/+
```

Notice how TSSB.PROC uses the backup security file dataset as the primary security file (CAISECF). Also, notice that all references to CAIBKUP have been removed. This missing CAIBKUP statement forces TSS to ignore automatic backups.

Again, correct the sample procedure and recatalog it into a VSE procedure library.

Chapter 4. VSE Considerations

This chapter describes special considerations for a variety of processes and subsystems, some of which may apply to your installation.

4.1 Multi-CPU Environments

The following considerations were mentioned in the installation steps but are repeated here as reminders:

1. In a multiple CPU environment, place the following CA-Top Secret files on a shared DASD volume: Security File, Audit Tracking File, and Recovery File.

   If sharing a Security File between multiple CPUs, the control option SHRFILE(YES) must be specified on each CPU.

2. Back up the Security File from one CPU only. Omit the CAIBKUP DLBL statement in STDLABELS or code the BACKUP(OFF) control option on all systems but one.

Chapter 5. Startup and Shutdown

CA-Top Secret can be started two ways, which are determined by the release of VSE/ESA.

If CA-Top Secret for VSE was purchased from IBM, then only VSE/ESA release 2.4 and above is supported.

Running on VSE/ESA release 1.4 up to VSE/ESA release 2.3: CA-Top Secret should be initialized in a static partition, prior to VSE/POWER activiation. The suggested partition used for this purpose should be fixed partition FB.

A new control option command, TSSIPLA, should be inserted in the VSE IPL procedure, immediately following the execution of CASATUIL. For example:

```
// UPSI 00
// EXEC CASAUTIL,SIZE=800K (Start CA-CIS and load CA-Top Secret and other CA products)
// EXEC TSSIPLA,PARM='SERVPART=FB' (Instruct TSS to initiate in static partition FB)
/*
START F1 (Start power)
Stop
```

Note: SERVPART=xx can be substituted with the value of another static partition, such as BG, F4, etc.

In the VSE IPL procedure, make the following changes to the $$JCLFB.PROC startup book, or whatever name your site calls the procedure that initiates FB.

```
// LIBDEF PHASE,SEARCH=(...CAILIB.TSSVSE30)
// SETPFIX LIMIT=100K,PERM
// EXEC PROC=TSS
/*
// PAUSE  execute PROC=TSS to restart TSS.  Use TSSB for emergency restart.
```

During IPL, TSSIPLA will initiate static partition FB, which will invoke the TSS address space using TSSMNGR. When TSS startup has completed, TSSMNGR will post TSSIPLA and the main IPL will continue. At this point it is assumed VSE Power initializes in the presetermined Power partition.

Once Power has completely activated itself and all Power controlled partitions, TSS will start to execute the commands (if any) defined in the TSSAUTO.Z dataset.

Starting CA-Top Secret on VSE/ESA 2.4 and above On VSE/ESA 2.4 and above systems, CA-Top Secret becomes the IBM External Security Manager (ESM).**:** In the IPL procedure, after SYS SEC=YES, an additional statement should be added to define the startup of ESM, instead of the IBM provided Basic Security Manager (BSM).

This statement should look like this:

```
SYS ESM=CAKSESM (Defines what ESM security should be used)
SYS SERVPART=FB (Defines what partition is used for external security)
```

In the IBM provided JCL procedures, locate the BSM server routine BSTPSTS and replace this with a // EXEC PROC=TSS statement instead. Please notice TSSIPLA is **not** required for VSE/ESA 2.4 and above.

It is very important that all assigns to TSS database files and LIBDEFs are resolved prior to executing TSSMNGR.

```
/.SECPART
* ID USER=FORSEC     !!NO PWD REQUIRED!!
* EXEC BSTPSTS       not needed anymore
// EXEC PROC=LIBDEF   SET LIBDEF SEARCH CHAINS
// LIBDEF PROC,SEARCH=(IJSYSRS.SYSLIB,CAILIB.TSSVSE30)
// EXEC PROC=TSS
/*
// SETPARM XSTAT&XPARTPW=' '
// EXEC PROC=CPUVAR&XNCPU,XSTAT&XPARTPW
/*
// IF XSTAT&XPARTPW = INACTIVE THEN
// GOTO FINISH
// PAUSE TO RESTART THE SECURITY SERVER ENTER '//EXEC PROC=TSS' USE TSSB FOR EMERGENCY

/.FINISH
```

CASAUTIL must be inserted in the VSE/ESA 2.4 and above IPL procedure, in a place prior to BSSINIT, see 3.9.6, “Task 7F: Insert EXECUTE For CASAUTIL” on page 3-21. It is recommended that CASAUTIL should be inserted before the first execute of IBM phase BSSINIT in the BG partition.

Make sure all libraries containing CA-CIS services and CA-Top Secret for VSE are accessible before executing CASAUTIL. Any prior // LIBDEF DROP should be removed.

As in the previous VSE releases, standard labels must be updated with correct CA-Top Secret dataset names, and permanent assignments to the datasets must be inserted in all partitions.

5.1 Activating CA-Top Secret

At this point, CA-Top Secret can be started by system IPL, once it has been assured that all previous tasks have completed.

CA-Top Secret has been successfully implemented if the following message displays:

```
TSS9000I  CA-Top Secret SECURITY 30yymmAKDnn ACTIVE
```

CA-Top Secret should initially be brought up in DORMANT mode. In this particular mode, you will lose password protection for password protected data sets. To maintain password protection you must permit each password protected data set for all facilities and accesses with ACTION(PASSWORD,FAIL). For example:

```
TSS PER(ALL) DSN('password.protected.dsname')
    FAC(ALL) ACCESS(ALL) ACTION(PASSWORD,FAIL)
```

5.2 Verifying Installation

Once CA-Top Secret has started, logon or sign on to the system.

CA-Top Secret will require a new password. If you are not prompted for a new password, be certain that you have used the ACID that was specified for in the TSSMAINT Security File creation JCL, for the SCA= parameter.

Use the following JCL as a sample:

```
// JOB TSS TEST
// ID USER=MSCA,PWD=MSCAPW,NEWPW=TORONTO
// PAUSE
/*
/&
```

Note: In this sample it was assumed SCA=MSCA with a password of MSCAPW used in the TSSMAINT utility, and that a new password of TORONTO was wanted.

5.2.1 Define User to CA-Top Secret

CA-Top Secret is now active, so we now have to add some userid definitions to the database. Copybook member TSSINIT.Z contains a sample JCL needed to define the first set if initial userid's and departments. Please notice this member is only meant as a guide line, carefully read the Command Functions guide and Users guide for further information. Some of the userid's in TSSINIT.Z such as CICSPROD and CICSTEST are required for CICS to initiate properly. Please refer to the CICS implementation guide for details on this. Please notice all of these initial definitions should be done under the MSCA userid.

5.3 Restarting CA-Top Secret

There are three ways to restart CA-Top Secret after it has been brought down:

- IPL
- Executed TSSMNGR from the outstanding reply in the predefined non power controlled security partition.
- Execute either // EXEC PROC=TSS or // EXEC PROC=TSSB

5.4 CA-Top Secret Shutdown

The CA-Top Secret address space can be deactivated with the following command:

```
80 SHUTDOWN
```

CA-Top Secret displays the two types of shutdowns in the following messages:

```
FB 0080 TSS9300I  ** SELECT TYPE OF SHUTDOWN ** <I> TO IGNORE
FB 0080 TSS9301I  <Z>  END OF DAY;  **RE-IPL** WILL BE REQUIRED
FB 0080 TSS9302D  <T>  TEMPORARY;   MAY IMPACT THROUGHPUT
FB-0080
```

5.4.1 End-Of-Day Shutdown

A normal end-of-day shutdown is accomplished by entering **Z** in response to the displayed messages. End-of-day shutdown prohibits new initiations in all modes other than DORMANT. This means new users will not be able to sign on to any facility and new batch jobs will not execute.

Provided the operator has the authority, an end-of-day shutdown can be reset using the RESETEOD control option.

If you entered **Z** inadvertently, refer to the previous section, *Restarting CA-Top Secret*.

5.4.2 Temporary Shutdown

A mid-day shutdown of security is accomplished by replying with an entry of **T** to the displayed messages. To prevent unauthorized mid-day shutdowns, specification of the MSCA's *previous* password or an ACID/password which has the CONSOLE attribute is required. If security is shutdown from any mode other than DORMANT, the DOWN options become active.

Chapter 6. SNTL2TSS - Conversion Utility

The SNTL2TSS program converts the CA-Top Secret Release 2.3 catalog to the format of a CA-Top Secret Release 3.0 security database.

The SNTL2TSS program can be used:

- only by the Data Security Administrator and
- only in a VSE environment.

CA-Top Secret Release 3.0 provides a hierarchy of TSS administrator types with varying scopes of authority. This hierarchy is as follows:

Administrative Type	**Scope Of Authority**
Master Security Administrator (MSCA)	The entire installation
Central Security Administrator (SCA)	The entire installation
Divisional Security Administrator (VCA)	An entire division
Departmental Security Administrator (DCA)	An entire department
User	Resources that are owned by the user

As there are different *scopes* of authority in CA-Top Secret Release 3.0, there are also varying *types* of authority, which correspond to authorization levels found in CA-Top Secret Release 2.3. Please refer to your CA-Top Secret Release 3.0 documentation for complete information about the database organization.

When converting your security database with the SNTL2TSS program, you specify administration and ownership criteria using control statements. You can enter these control statements from either the SYSLOG or SYSIPT device. If you initiate SNTL2TSS from the systems console, control statements are accepted from SYSLOG. If you initiate SNTL2TSS from the SYSRDR device, control statements are accepted from SYSIPT. All output is directed to SYSLST and SYSPCH. Error return codes and the conversion completion message also display on the systems console.

Note: This program will process the entire security catalog and create a CA-Top Secret Release 3.0 security database. This function can take a very long time to complete, depending on the content of the security catalog.

6.1 Storage Requirements for SNTL2TSS

The Conversion program has the following storage requirements:

- The SNTL2TSS program must be executed in a partition of at least 260K.
- The program also requires the use of disk work space for a sequential file. Use the following as a guideline for estimating the size of the file needed (an example follows).

Note: The DTF name of the sequential file is CAIDRES.

To estimate the storage needed, enter the approximate value on each line of the table below then enter these values into the equation that follows.

Number of Resources per Authorization group = ____ = A

Number of Authorization groups per User = ____ = B

Number of Users on the Database = ____ = C

Next, enter these values into the equation below and compute the file storage needed.

```
((A * B) * C) * 55 = number bytes or D
  D/19,069 = number of tracks or E
E/30 = number of cylinders on disk device
```

6.1.1 Example for Determining Needed File Size

This example uses 3350 disk devices.

A DSA fills in the chart according to her VSE security database:

Number of Resources per Authorization group = 100 = A

Number of Authorization groups per User = 10 = B

Number of Users on the Database = 300 = C

Then the DSA substitutes these values into the equation, as follows:

```
((100*10)*300)*55 = 16,500,000
   16,500,00/19,069 = 865.279
      865.279/30 = 28.845
```

From this, the DSA calculates that approximately 30 (28.845) cylinders should be allocated for the file.

6.2 Command Syntax

Follow these rules when coding control statements:

- Code between columns 1 and 71. Columns 72-80 are ignored.
- Begin each control statement with a command, followed by one or more operands.

6.3 Summary of Commands

The SNTL2TSS commands must be entered in a specific order. The following summary presents the commands in their proper order.

Command	Function
ADMINISTRATION	Specifies how to create divisions and departments.
DIVISION	Specifies division ACIDs and names.
DEPARTMENT	Specifies department ACIDs.
END	Specifies that the user has finished supplying division or department ACIDs.
DEFAULT	Specifies that the conversion program should supply the department ACIDs.
OWNERSHIP	Specifies who will be assigned ownership of the protected resources.
FCT	Converts an FIL resource type as a CISS FCT, otherwise the FIL resource type is converted as a VSE DSN.
PPT	Converts a PGM resource type as a CISS PPT, otherwise the PGM resource type is converted as a VSE PROGRAM.

If you initiate the SNTL2TSS program from SYSLOG, command prompts display on the systems console. If you initiate the SNTL2TSS program from SYSRDR, obtain the output from SYSLST, check for errors, correct and re-execute the job.

6.4 ADMINISTRATION

Purpose: The ADMINISTRATION command specifies how divisions and departments will be created, who is assigned to them, and who will administer them. It also specifies how the user's password is to be converted.

The ADMINISTRATION command must be specified.

Format

```
ADMINistration(MSCA|DIVIsion|DEPArtment,FORCE|SAME)
```

Keyword Descriptions: MSCA|DIVISION|DEPARTMENT

Specify which levels of administrative authority to include in the new database.

Valid levels are:

MSCA

- Creates one department with

  ```
  ACID = SNTL2TSS
  Name = CORPORATE
  ```

- Assigns all users to this department (except DSA and DSOs).
- Converts the DSA into an SCA with ALL authority.
- Creates DSOs based on authorization level:
  - Level 1 receives AUDIT authority.
  - Level 2 receives AUDIT + MAINTAIN + XAUTH authority.
  - Levels 3-9 receive ALL authority.
- Creates no divisions, DCAs, or VCAs.
- Allows processing without DIVISION or DEPARTMENT commands.

DIVISION

- Creates departments based on user profile information.
- Creates divisions for DSOs with multiple departments.
- Assigns users without departments to "CORPORATE." Assigns all other users to their corresponding departments.
- Converts the DSA and level 9 DSOs to SCAs with ALL Authority.
- Defines DSOs as appropriate (DCA or VCA):
  - Level 1 receives AUDIT authority.
  - Level 2 receives AUDIT + MAINTAIN + XAUTH authority.
  - Levels 3, 4, and 5 receive ALL authority.

To create a division, enter the ADMIN command followed by the DIVISION command to supply the conversion program with an 8-character division ACID and a 20-character division name. If these values are not given, the DSO being processed becomes a DCA.

To create more than one division, enter a DIVISION command for each one. If you want to define every DSO with multiple departments as a DCA, be sure to enter sufficient DIVISION commands. (Up to 999 divisions are supported.)

DEPARTMENT

- Creates departments based on user profile information.
- Assigns users to "CORPORATE." Assigns all other users to their corresponding departments.
- Converts the DSA and level 9 DSOs to SCAs with ALL authority.
- Defines DSOs with multiple departments as SCAs:
  - Level 1 receives AUDIT authority.
  - Level 2 receives AUDIT + MAINTAIN + XAUTH authority.
  - Levels 3, 4, 5, and 9 receive ALL authority.
- Defines DSOs with a single department as DCAs:
  - Level 1 receives AUDIT authority.
  - Level 2 receives AUDIT + MAINTAIN + XAUTH authority.
  - Levels 3, 4, and 5 receive ALL authority.
- Creates no divisions or VCAs.

The user may either supply 8-character department ACIDs or let the conversion program generate them using defaults.

To supply the program with department ACIDs, issue commands in this order: ADMINISTRATION, DIVISION (if any), and then DEPARTMENT. You can define up to 999 departments.

6.4.1 SAME|FORCE

Use these options to specify how each user's password is to be converted:

SAME Defines the user to CA-Top Secret Release 3.0 with the same password defined on the CA-Top Secret Release 2.3 database.

FORCE Defines the user to CA-Top Secret Release 3.0 with the user's ACID. At first signon, the user must supply a new password to successfully complete the signon.

6.5 DIVISION

Purpose: Use the DIVISION command to supply an ACID ID and name for each division created in the conversion.

Format

```
DIVISION(acid,division-name)
```

For "acid" specify the 8-character division ACID.

For "division-name" specify the 20-character division name.

Note: Specify DIVISION commands before DEPARTMENT commands.

6.6 DEPARTMENT

Purpose: Use the DEPARTMENT command to supply ACIDs for the departments that are created in the conversion.

Format

```
DEPARTMENT(acid)
```

For "acid" specify the 8-character department ACID.

6.7 END

Purpose: Use this command to end DIVISION processing or DEPARTMENT processing.

Format

```
END
```

6.8 DEFAULT

Purpose: Use this command to allow the conversion program to generate department ACIDs.

Format

```
DEFAULT
```

The default IDs are created in the following format: DEPTnnnn.

Note: If an insufficient quantity of department IDs are specified to handle the number of departments, SNTL2TSS will first use the specified IDs and then generate default department IDs.

6.9 OWNERSHIP

Purpose: Use this command to assign ownership of the protected resources. In CA-Top Secret Release 3.0, resources can only be owned by a single ACID. Consequently, resources that are owned by multiple users on the CA-Top Secret VSE database, must be defined with a single ownership level.

The OWNERSHIP command is required.

Format:

```
OWNership(MSCA|DIVIsion|DEPArtment|USER)
```

This command creates a department of resources with:

```
ACID = RESOURCE
ACID name = RESOURCES
```

Ownership of all resources (except terminals) is distributed in the following manner.

MSCA All resources are owned by department "RESOURCES."

DIVISION The division owns only resources it can access. All other resources are owned by department "RESOURCES."

DEPARTMENT
The department owns only resources it can access. The division owns only resources the division can access. All other resources are owned by department "RESOURCES."

USER The user owns only the resources he can access. The department owns only resources it can access. The division owns only resources the division can access. All other resources are owned by department "RESOURCES."

All terminals are owned by one department with:

```
ACID = TERMINAL
ACID name = TERMINALS
```

OWNERSHIP is the last command you need to issue. Once you issue this final command, the SNTL2TSS program then converts authorization groups automatically with no further instructions.

6.10 FCT

Purpose: Use this command to allow the conversion program to distinguish between a CICS file (FCT entry) and a VSE batch dataset for resource type FIL.

Format:

```
FCT(name)|FCT(nn*)
```

Use as many FCT commands as needed. "name" is 1-8 chracter resource type FIL name that is also a CICS FCT entry name. You can use a generic name by adding an "*" to indicate any FIL name that begins with nn.

The conversion program will convert a FIL resource type into VSE batch dataset name, unless FIL name has a matching FCT name (specifically or generically), then it is converted into a CICS FCT entry.

6.11 PPT

Purpose: Use this command to allow the conversion program to distinguish between a CICS program (PPT entry) and a VSE batch program for resource type PGM.

```
PPT(name)|PPT(nn*)
```

Use as many PPT commands as needed. "name" is 1-8 chracter resource type PGM name that is also a CICS PPT entry name. You can use a generic name by adding an "*" to indicate any PGM name that begins with nn.

The conversion program will convert a PGM resource type into VSE batch PROGRAM name, unless PGM name has a matching PPT name (specifically or generically), then it is converted into a CICS PPT entry.

6.12 Using SNTL2TSS: A Sample Conversion

A DSA wants to convert according to the following plan:

A Create divisions and departments. All users' passwords should remain the same.

B Create up to three divisions. All others should become DCAs.

C End division processing.

D Generate default department IDs.

E Bring ownership down to the user level whenever possible.

To accomplish these goals, the DSA codes the following JCL:

```
     // JOB SNTL2TSS
     // ID USER=DSA,PWD=INITIAL
     // ASSGN SYS005,DISK,VOL=WORK01
     // DLBL CAIDRES,'RESOURCE.FILE'
     // EXTENT SYS005,WORK01,1,0,90,300
     // LIBDEF PHASE,SEARCH=CAILIB.TSSVSE30
     // EXEC SNTL2TSS,SIZE=SNTL2TSS
A         ADMINISTRATION(DIVISION,SAME)
        DIVISION(DIVI01,DIVISION01)
B         DIVISION(DIVI02,DIVISION02)
        DIVISION(DIVI03,DIVISION03)
C         END
D         DEFAULT
E         OWNERSHIP(USER)
            FCT(nn*)
            PPT(nn*)
     /*
     /&
```

Note: The conversion job **must** be run on a system where CA-Top Secret 2.3 is active, as the conversion routine uses the TSSVSE runtime system to access the security catalog. But, the conversion routine must be executed from the CA-Top Secret **3.0** sub-library, as is noted by the LIBDEF statement in the above sample JCL. There is a conversion routine with the same name in the CA-Top Secret 2.3 product sub-library, it should **not** be used.

The conversion job produces two forms of output at completion:

1. On SYSLST:
   detailed information on what has and has not been converted. The listing will include all FILES, SYSTEM LIBRARIES, SUB LIBRARIES, and MEMBERS that you must manually convert from VSE to MVS. Also, throughout the listing will be notes on what is not supported by CA-Top Secret Release 3.0.
2. On SYSPCH:
   all the commands that you should submit to CA-TOP SECRET MVS. To submit these from batch to TSO, you can use the MVS utility program (IKJEFT01).

6.13 Unsupported MVS Functions

The following CA-Top Secret VSE security information is not supported in the conversion process to CA-Top Secret Release 3.0:

User Profile fields: EXPINT, EXPDT, MIN DAYS, MAX TIMES, LOGONOPT, CALID, TIMID, VIOLATIONS, SIGNON

Facilities/Resources: PARTITION, DESCRIPTION, LOGOP

6.14 After The Conversion

CA-Top Secret Release 3.0 is generic in nature. All resources are owned and defined by prefixes, causing everything to be protected. In the CA-Top Secret Release 2.3 environment, only resources defined to the database are protected. Because of the nature of TSS, users may encounter security violations where they never received them before. Therefore, we recommend that you run CA-Top Secret Release 3.0 in WARN mode after the conversion is completed.

Chapter 7. ALRT2TSS - Alert Instructions

ALRT2TSS is batch program that reads the ALERT files and generates CA-Top Secret batch statements to create users, define resources, and permit users acccess to those resources. Users can make changes to the statements generated by ALRT2TSS, with an editor of their choice, before running the statements throughCA-Top Secret

In some cases, statements must be edited before submitting them to CA-Top Secret. For example, if an ALERT/CICS terminal record is converted to a CA-Top Secret ACID using the four character terminal ID as the ACID, that four character terminal ID must be changed to the corresponding VTAM applid for that terminal. When the generated statements is submitted to CA-Top Secret for execution, it must be run with the MSCA ACID, since it may create an administrator ACID.

ALERT/CICS users may have more than one security file, one for each CICS partition or for different CICS environments (test, production, etc.). ALRT2TSS should be run multiple times, once for each CICS security file, as well as once for each table ID. ALERT/VSE users may define multiple table IDs in the same security file, one for each VSE batch environment (test, production, etc.). Each run of ALRT2TSS would be for a different CA-Top Secret facility corresponding to the different ALERT/CICS security files or ALERT/VSE table IDs.

When users are permitted access to resources in CA-Top Secret, that permission will only be for that FACILITY. The same userid may exist multiple times in the ALERT CICS or VSE security files. Even though ALRT2TSS will attempt to define the same user to CA-Top Secret multiple times, only the first definition will succeed and the additional definitions will fail as duplicate users. However, the users will be (with some exceptions) authorized to all the resources that they had access to in all files or under different table IDs.

ALRT2TSS execution is controlled by the following control statements that are read from SYSIPT. The statements must start in column one and continue as shown with no spaces:

ALTC=aaaaaaa:

This statement is used to convert the ALERT/CICS file, where "aaaaaaa" must be a one to seven character VTAM applid. For the CICS partition that the ALERT/CICS security file controls, this VTAM applid will be used to define the CA-Top Secretfacility for that CICS partition, and it will also be used to to permit users access to resources in this facility.

ALTV=tt,FAC=ffffffff:

This statement is used to convert the ALERT/VSE file, where "tt" is the table ID in the VSE security file being converted and "ffffffff" is a one to eight character name used to define a CA-Top Secret facility for the ALERT/VSE table ID batch environment. This name will also be used to permit users access to VSE resources in that facility.

LOC=(C,3,2,D,3,2,T,2,3):

This parameter is used to map ALERT location information (C=Company, D=Division, T=Department) into CA-Top Secret Zone, Division and Department ACIDs, respectively. "C,3,2" allocates the third position for two columns of the company number to create the TOP SECRET zone ACID for that company. "D,3,2" allocates the third position for two columns of the ALERT division number, which is appended to the company code to construct the CA-Top Secret division ACID. "T,2,3" allocates the second position for three columns of the department code, which is appended to the division ACID to construct the CA-Top Secret department ACID.

Note: The total length of the three segments must not exceed 8 characters, which is the CA-Top Secret ACID length limit. Also, the Section part of the ALERT location is not used since there is no match for it in CA-Top Secret.

This parameter is required if the user desires to convert the existing location information into CA-Top Secret format in order to move into decentralized security adminstration. If the parameter is omitted, no location information is converted. The converted department ACID is not guaranteed to be unique, and some duplicate ACIDs may result, which must then be corrected manually.

CPUID=cccccccc,VSEc:

This statement limits users access to resources located on a particular CPU. ALERT/VSE uses the ENVERS CPUID:cccccccc, where "cccccccc" is the eight hexadecimal digit hardware CPU IDentifer. CA-Top Secret defines each CPU as a FACILITY VSEc, where "c" is the system adapter CPU ID. This control statement is used to map the hardware CPU ID to the one character adapter CPU ID. One control statement must be provided for each CPU in a multiple CPU environment.

LANG=l,cc:

This statement identifies the one character language code that ALERT supports, "l" represents the following language codes: E=English, F=French, G=German, and I=Italian. "cc" represents the two character message suffix defined on the ALERT User Profile records. This message suffix will be converted to the CICS Language code.

PASSWORD= (FORCE, SAME) Default: FORCE:

This statement identifies if the conversion program should attempt to FORCE a new password, or keep the SAME as Alert when creating the new CA-Top Secret ACID's. The FORCE option requires the user to supply a new password to successfully complete the first signon.

After validating the input control statements, ALRT2TSS also ensures that the ALERT input files are ALERT Release 4.9, otherwise it terminates with an error message. The program then generates a CA-Top Secret statement to create a TSSCONV ACID as a department ACID. This ACID will own all of the existing resources that are converted.

```
TSS CREATE(TSSCONV ) NAME('MASTER ADMINISTRATOR', TYPE(DEPARTMENT)
```

Next, ALRT2TSS generates statements to create Zone, Division, and Department using ALERT Location records only if the LOC=(....) was specified.

```
TSS CREATE(cc      ) NAME('company name        ') TYPE(ZONE      )
TSS CREATE(ccdd    ) NAME('division name       ') TYPE(DIVISION  )
TSS CREATE(ccddttt ) NAME('department name     ') TYPE(DEPARTMENT)
       SUSPEND
```

The SUSPEND keyword is generated only if the location status code is equal to "D" Disabled.

If the ALTC parameter was specified, ALRT2TSS converts the CICS file, and first creates the CICS FACILITY from the ALTC=aaaaaaa control statement.

```
TSS ADDTO(TSSCONV ) FACILITY(aaaaaaa )
```

It then converts the Transactions records, and generates statements to add existing transactions to the TSSCONV department.

```
TSS ADDTO(TSSCONV.) OTRAN(....)
```

The statement is only generated if the transaction status code is not equal to "D" Disabled. ALRT2TSS then converts the Program records, and generates statements to add existing programs to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   PPT(........)
```

The statement is only generated if the program status code is not equal to "D" Disabled. ALRT2TSS then converts the FL File records, and generates statements to add existing files to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   FCT(........)
```

The statement is only generated if the file status code is not equal to "D" Disabled.

ALRT2TSS does not convert the terminal records since ALERT uses the four character physical terminal ID, while CA-Top Secret needs the VTAM applid for the terminal in order to secure the termial as a resource. However, ALRT2TSS issues a warning message for each terminal that it finds so the users can manually add those terminals to CA-Top Secret if desired. ALRT2TSS also converts terminal records that are defined with Operator Sign On Exempt = Yes, and Terminal Sign on Exempt = Yes. A user ACID is created for each terminal with those two attributes and those ACIDS will also be permitted access to any resources defined on the terminal records.

ALRT2TSS first creates the ACID records.

```
TSS CREATE(ACID) NAME('    ') TYPE(USER)   -
      INSTDATA(......... ......)
      SUSPEND
      LANGUAGE(..)
      LTIME(....)
      DEPARTMENT(........)
      PASSWORD(.....)
```

The terminal ID is used as the ACID, and the terminal adminstrator ID is used in the NAME( ). Those ACIDs are always created as TYPE(USER).

The INSTDATA(.....) keyword is generated if either TUSER field exists (not equal to hex 00s, not equal to blanks, and not equal to underscores).

The SUSPEND keyword is generated only if the terminal record status is equal to "D" Disabled.

The LANGUAGE keyword is generated only if the terminal record TEMSG exists.

The DEPARTMENT keyword is only generated if the LOC=( ) parm was specified and the terminal record location information exists.

The Password is generated with the NOEXP No Expiration option.

ALRT2TSS then permits users access to resources, and generates a permit statements for OTRAN, PPT and FCT according to the bit map defined in those terminal records.

```
TSS PERMIT(term) OTRAN(....)
TSS PERMIT(term) PPT(........)
TSS PERMIT(term) FCT(........)
```

If the ALTV=tt,ffffffff statement is provided, ALRT2TSS starts converting the VSE, and first creates the facility from the ALTV=tt,ffffffff statement.

```
        TSS ADDTO(TSSCONV ) FACILITY(ffffffff)
```

It also adds DITTO program to TSSCONV

```
TSS ADDTO(TSSCONV ) PROGRAM(DITTO)
```

ALRT2TSS then starts converting the secid record type 40 and 35 into CA-Top Secret PROFILE ACIDs.

```
  TSS CREATE(secid   )            TYPE(PROFILE) DEPARTMENT(TSSCONV)
              SUSPEND
```

The SUSPEND keyword is generated only if the secid record status is not equal to "D" Disabled.

ALRT2TSS then converts the OP and UP Operator records, and it generates statements to create an ACID for each user.

```
  TSS CREATE(opnumber) NAME('operator name') TYPE(USER)
  TSS CREATE(opnumber) NAME('operator name') TYPE(SCA)
  TSS CREATE(opnumber) NAME('operator name') TYPE(LSCA)
              INSTDATA(......... ......)
              SUSPEND
              UNTILL(....)
              OPIDENT(...)
              LANGUAGE(..)
              LTIME(....)
              DEPARTMENT(........)
              PASSWORD(,dur,EXP)
```

Type(USER) is generated if the operator record OPTYPE is equal to regular user. Type(SCA) is generated if the operator record OPTYPE is equal to main administrator, otherwise TYPE(LSCA) is generated.

The INSTDATA(..... .....) keyword is generated if either OUD1 or OUD2 exists (not equal to hex 00s, not equal to blanks, and not equal to underscores). An 18 character field is always generated even if only one of the two fields existed.

The SUSPEND keyword is generated only if the operator record status is not equal to "D" Disabled

The UNTIL keyword is generated only if the operator record OEXP expiration date exists.

The OPIDENT keyword is generated only if the operator record CICS operator ID exists.

The LANGUAGE keyword is generated only if the operator record ULANG exists (disabled until implemented in TSS/VSE).

The LTIME keyword is generated only if the operator record OTMLT Inactive time exists.

The DEPARTMENT keywords are only generated if the LOC=( ) parm was specified and the operator record location information exists.

The PASSWORD keyword is always generated, and no password is converted. Instead the ACID ID will be the password for the first sign on. The duration for the password is taken from the global OPTION record, and the password is forced to expire after the first sign on.

ALRT2TSS then ADDTO facilities and profiles to users.

```
TSS ADDTO(........)
      FAC(CICS)
      FAC(BATCH)
      FAC(CICS,BATCH)
      PROFILE(--------,--------,--------,--------,--------)
```

The FAC(CICS) is generated if the operator record OCIC is equal to "Y."

The Facility name is taken from the ALTC=aaaaaaaa control statement.

The FAC(BATCH) is generated if the operator record OBAT is equal to "Y."

The Facility name is taken from the ALTv=tt,fac=aaaaa control statement.

The FAC(CICS,BATCH) is generated if both were equal to "Y."

The PROFILE(...,..) statement is generated if any of the five batch secids are defined on the OP record.

ALRT2TSS then ADMINs users that are converted to SCA or LSCA main administrators and sub adminstrators.

```
TSS ADMIN(........)ACID(ALL)DATA(ALL)MISC1(ALL)MISC2(ALL)MISC9(ALL)
            FAC(CICS)
            FAC(BATCH)
            FAC(CICS,BATCH)
```

The FAC(CICS) is generated if the operator record OCIC is equal to "Y."

The Facility name is taken from the ALTC=aaaaaaaa control statement.

The FAC(BATCH) is generated if the operator record OBAT is equal to "Y."

The Facility name is taken from the ALTv=tt,fac=aaaaa control statement.

The FAC(CICS,BATCH) is generated if both were equal to "Y."

ALRT2TSS then permits users access to CICS resources, and generates permit statements for OTRAN, PPT and FCT according to the bit map defined in those records. If the OMDEL field in the op record exists, the user will be permitted access to resource defined on the model record.

```
TSS PERMIT(........) OTRAN(....) ACTION(AUDIT)
```

Up to six statements may be generated if any ???? of the ALERT transaction list exists.

```
TSS PERMIT(operator) OTRAN(....)
TSS PERMIT(operator) PPT(........)
TSS PERMIT(operator) FCT(........)
```

ALRT2TSS also permits users access to terminals, only if the ALERT user is permited accees to "ALL" terminals, and it generates up to 4 permit statements per user if any of the operator record OPRMT, OALT1, OALT2, or OTRMG are specified "ALL."

```
TSS PERMIT(........) TERMINAL(*ALL*)
```

ALRT2TSS then converts the group records, and first generates a statement to create a CA-Top Secret profile from the group GR records.

```
TSS CREATE(groupid) NAME('group desc') TYPE(PROFILE)
             DEPARTMENT(TSSCONV)
             SUSPEND
```

The SUSPEND keyword is generated only if the group record status is equal to "D" Disabled.

ALRT2TSS then generates statements to permit the group access to resources according to the bit map defined in the GR record.

```
TSS PERMIT(group   ) OTRAN(....)
TSS PERMIT(group   ) PPT(........)
TSS PERMIT(group   ) FCT(........)

TSS PERMIT(........) OTRAN(....) ACTION(AUDIT)
```

Up to six statements may be generated if any ???? of the ALERT transacion list exists.

ALRT2TSS then adds the list of groups assigned to an operator in the UG records to that operator.

```
TSS ADDTO(operator)GROUP(--------,--------,--------,--------,--------)
```

ALRT2TSS then permits the operator access to the resources included in the UG record.

```
TSS PERMIT(operator) OTRAN(....)   ACC(....)
TSS PERMIT(operator) PPT(........) ACC(.....)
TSS PERMIT(operator) FCT(........) ACC(.....)
```

ALRT2TSS also denys the operator access to the resources excluded in the UG record.

```
TSS PERMIT(operator) OTRAN(....)      ACC(NONE)
TSS PERMIT(operator) PPT(........)    ACC(NONE)
TSS PERMIT(operator) FCT(........)    ACC(NONE)
```

ALRT2TSS then converts the VSE file DASD dataset record type 50 and first adds them to the TSSCONV ACID.

```
TSS ADDTO(TSSCONV.)
       DSNAME(.......................)
       VOLUME($$VSAM)
```

The VOLUME keyword is generated only if it exists on the DASD record.

ALRT2TSS then PERMITS access to the dataset to the secid reference record type 70, and generates up to 4 PERMIT statements per secid, one for each access level defined for the data set.

```
TSS PERMIT(........) ACCESS(NONE/READ)     ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/UPDATE)   ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/WRITE)    ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/ALL)  ACTION(NOTIFY/AUDIT)
       DSNAME(.......................)
       VOLUME(........)
```

Four access levels are converted:

- R is converted to ACC(READ)
- U is converted to ACC(UPDATE)
- W is converted to ACC(WRITE)
- A is converted to ACC(ALL) ??

ACC(NONE) is generated if action code is equal to "C" Cancel. The ACTION(NOTIFY/AUDIT) is decided based on the action code:

- uresp.code "L" log is converted to ACTION(AUDIT)
- uresp.code "W" wto is converted to ACTION(NOTIFY)

otherwise, the ACTION( ) keyword is not generated at all.

The VOLUME keyword is generated only if it exists on the DASD record.

ALERT/VSE supports generic secid's to be permitted to a resource, while CA-Top Secret does not. Multiple sets of PERMIT statements are generated for every secid on the file that matches a generic secid.

ALRT2TSS then converts the VSE file TAPE dataset record type 55, and first adds them to the TSSCONV ACID.

```
TSS ADDTO(TSSCONV.)
       DSNAME(.......................)
       VOLUME(........)
```

The VOLUME keyword is generated only if it exists on the tape record.

ALRT2TSS then PERMITS access to the dataset to the secid reference record type 70, and it generates up to 4 PERMIT statements per secid, one for each access level defined for the data set.

```
TSS PERMIT(........) ACCESS(NONE/READ)     ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/UPDATE)   ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/WRITE)    ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/ALL)  ACTION(NOTIFY/AUDIT)
       DSNAME(.......................)
       VOLUME(........)
```

Four access levels are converted:

- R is converted to ACC(READ)
- U is converted to ACC(UPDATE)
- W is converted to ACC(WRITE)
- A is converted to ACC(ALL) ??

ACC(NONE) is generated if action code is equal to "C" Cancel.

The ACTION(NOTIFY/AUDIT) is decided based on the action code:

- uresp.code "L" log is converted to ACTION(AUDIT)
- uresp.code "W" wto is converted to ACTION(NOTIFY)

otherwise the ACTION( ) keyword is not generated at all.

The VOLUME keyword is generated only if it exists on the DASD record.

ALERT/VSE supports generic secids to be permitted to a resource, while CA-Top Secret does not. Multiple sets of PERMIT statetments are generated for every secid on the file that matches a generic secid.

ALRT2TSS then converts the VSE file OTHER resources record 60, and it first maps the ENVERS CPUID to FACILITY VSEx according to the CPUID=cccccccc,VSEc control statement.

ALRT2TSS maps the ENVERS PTNID to FACILITY ptnid, and it also maps the LIBRARY to VSELIB, SUBLIB to VSESLIB, and LIBMEM to VSEMEMBR. ALRT2TSS also maps user defined classes into ALRTUCLS resource class.

For DITTO EXECUTE: pgmname resource, ALRT2TSS permits the PROFILE the ability to execute program DITTO but only from PRIVPGM pgmname.

```
TSS PERM(ASIBG   ) PROGRAM(DITTO) FAC(BATCH   ) PRIVPGM(DFHSIP  )
```

ALRT2TSS then adds them to the TSSCONV ACID.

```
TSS ADDTO(TSSCONV.)
       otherclass(.......................)
```

The resource name in ALERT can be up to 52 bytes long, which is truncated to 44 bytes. However, the only resource that may have a problem is the LIBRARY resource, and the program will scan the resource name and issue a warning message if the name excedes 44 bytes.

ALRT2TSS then PERMITS access to the resource to the secid reference record type 70, and it generates up to 4 PERMIT statements per secid, one for each access level defined for the data set.

```
TSS PERMIT(........) ACCESS(NONE/READ)     ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/UPDATE)   ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/EXECUTE)  ACTION(NOTIFY/AUDIT)
TSS PERMIT(........) ACCESS(NONE/ALL)  ACTION(NOTIFY/AUDIT)
       DSNAME(.......................)
       VOLUME(........)
```

Four access levels are converted:

- R is converted to ACC(READ)
- U is converted to ACC(UPDATE)
- E is converted to ACC(EXECUTE)
- A is converted to ACC(ALL) ??

ACC(NONE) is generated if action code is equal to "C" Cancel.

The ACTION(NOTIFY/AUDIT) is decided based on the action code:

- uresp.code "L" log is converted to ACTION(AUDIT)
- uresp.code "W" wto is converted to ACTION(NOTIFY)

otherwise the ACTION( ) keyword is not generated at all.

ALERT/VSE supports generic secids to be permitted to a resource, while CA-Top Secret does not. Multiple sets of PERMIT statetments are generated for every secid on the file that matches a generic secid.

For DITTO COMMAND: cmdname resource, ALRT2TSS permits the PROFILE the ability to execute program DITTO in addition to the command permit.

```
TSS PERM(ASIBG   ) PROGRAM(DITTO) FAC(BATCH   )
```

Chapter 8. XSMA2TSS - Conversion Utility

The XSMA2TSS utility is a batch program that reads the IBM CICS migration file DFHXSMA and generates CA-Top Secret batch statements to create users, define resources, and permit users access to those resources.

The user can make changes to the statements generated by XSMA2TSS before running them through CA-Top Secret. Any editor can be used to make changes.

In some cases, the program generates a CA-Top Secret statement to create a TSSCONV ACID as a department ACID. This ACID will own all of the existing resources that are converted.

```
TSS CREATE(TSSCONV ) NAME('MASTER ADMINISTRATOR', TYPE(DEPARTMENT)
```

Next, XSMA2TSS generates statements to cause all the CICS resources to be owned by the TSSCONV department. It first converts the Transactions records to record type PC. It then generates statements to add existing transactions to the TSSCONV department.

```
TSS ADDTO(TSSCONV.) OTRAN(....)
```

XSMA2TSS then converts the Program records to record type PP. It generates statement to add existing programs to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   PPT(........)
```

XSMA2TSS then converts the File records to record type FC It generates statements to add add existing files to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   FCT(........)
```

XSMA2TSS then converts the Journal records to record type JC It generates statements to add add existing Journals to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   JCT(........)
```

XSMA2TSS then converts the Transient Data records to record type DC It generates statements to add add Transient data to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   DCT(........)
```

XSMA2TSS then converts the Temp Stoarage records to record type TS It generates statements to add add Temp storage to the TSSCONV department.

```
TSS ADDTO(TSSCONV.)   TST(........)
```

XSMA2TSS converts the RCF Auhorized Printers by creating an ACID for the printertype(user) using the four character id as the ACID ID.

```
TSS CREATE(acid) NAME('    ') TYPE(USER)   -
    DEPARTMENT(TSSCONV )
    PASSWORD(.....)
```

The Password is generated with the NOEXP No Expiration option. XSMA2TSS then converts the User Profile records to record type SU. It generates statements to create USER type acids.

```
TSS CREATE(acid) NAME('    ') TYPE(USER)   -
    DEPARTMENT(TSSCONV )
    OPIDENT(...)
    OPPRTY(...)
    LANGUAGE(.)
    LTIME(...)
    PASSWORD(.....)
```

Next, XSMA2TSS generates statements to add the facility for the CICS region that the user was defined for. This is taken from the VTAM applid for the CICS region from the user profile record. Those FACILITY values must reside in the site's Facility Matrix Table. XSMA2TSS also genrates statements to add the Operator classes and TRANSEC keys that the user is authorized for.

```
TSS ADDTO(acid) FACILITY(........) -
    OPCLASS(....) -
    SCTYKEY(....)
```

XSMA2TSS then permits users access to transactions if the transaction record is defined for the same CICS region as the User Profile record and the SECKEY on the transaction records matches one of the keys defined on the TRANSEC KEY on the user profile record.

```
TSS PERMIT(user) OTRAN(....) FACILITY(.....)
```

The FACILITY keyword value is taken from the VTAM applid for the CICS region for the transaction. The user will be permitted access to the transaction only when run in that CICS region.

XSMA2TSS then permits users access to other resourses if the transaction record is defined for the same CICS region as the User Profile record and the RSL KEY on that resource record matches one of the RSL keys defined the the user profile record.

```
TSS PERMIT(user) PPT(....) FACILITY(.....)
TSS PERMIT(user) FCT(....) FACILITY(.....)
TSS PERMIT(user) JCT(....) FACILITY(.....)
TSS PERMIT(user) DCT(....) FACILITY(.....)
TSS PERMIT(user) TST(....) FACILITY(.....)
```

The FACILITY keyword value is taken from the VTAM applid for the CICS region for the resource The user will be permitted access to the transaction only when run in that CICS region. Those FACILITY values must reside in the site's Facility Matrix Table.

Appendix A. Initial CAI Product Installation

CAINSTB1: The initial product installation JCL creates the production libraries and history file, and installs the product into the newly created libraries and history file. Use this JCL only to install the first product tape utilizing this standard.

After the product tape has been successfully installed, proceed with the steps in this Installation Guide.

Use the following JCL to extract CAINSTB1 from the installation tape.

```
// JOB     CAINSTB0          CATAL INSTALL JCL TO LIBRARY
// SETPARM LIBNAME=@LIBNAME  replace @LIBNAME w/ library name
// SETPARM SUBNAME=@SUBNAME  replace @SUBNAME w/ sublibrary name
// SETPARM TAPECUU=@TAPECUU  replace @TAPECUU w/ install tape address
   MTC     FSF,&TAPECUU,7
// ASSGN   SYSIPT,&TAPECUU
// EXEC    LIBR,SIZE=256K,PARM='ACCESS SUBLIB=&LIBNAME..&SUBNAME'
// RESET   SYSIPT
/&
```

CAINSTB1 Execution: CAINSTB1 performs the following functions:

1. Open SYSPCH using the extents of the history files. This is a precautionary measure to avoid errors when the history file extent resides on a newly defined VM minidisk.
2. Create the CAI production MSHP history file and the CAI production libraries.
3. Install the product tape to the production history file and libraries.

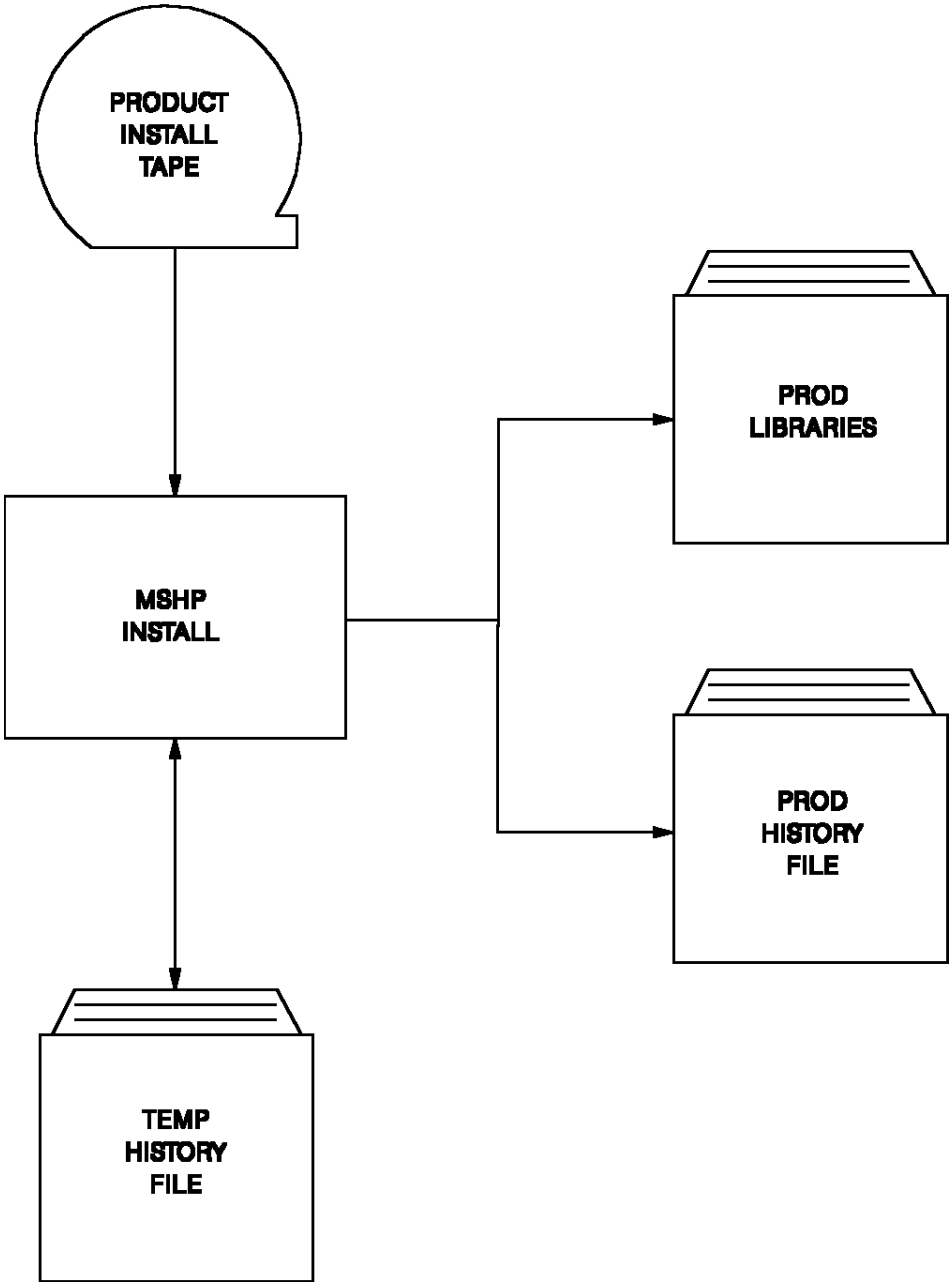


Figure A-1: CAINSTB1 Execution

A.1 Subsequent CAI Product Installation

CAINSTB2: CAINSTB2 is used to install a product into test libraries to allow installation verification and testing prior to migration into the production libraries.

After the product tape has been successfully installed, proceed with the steps in this Installation Guide.

CAINSTB2 Execution: CAINSTB2 performs the following functions:

1. Open SYSPCH using the extents of the history file. This is a precautionary measure to avoid errors when the history file extent resides on a newly defined VM minidisk.
2. Create the CAI installation libraries and history file for the product tape being installed.
3. Install the product tape to the installation history file and libraries.

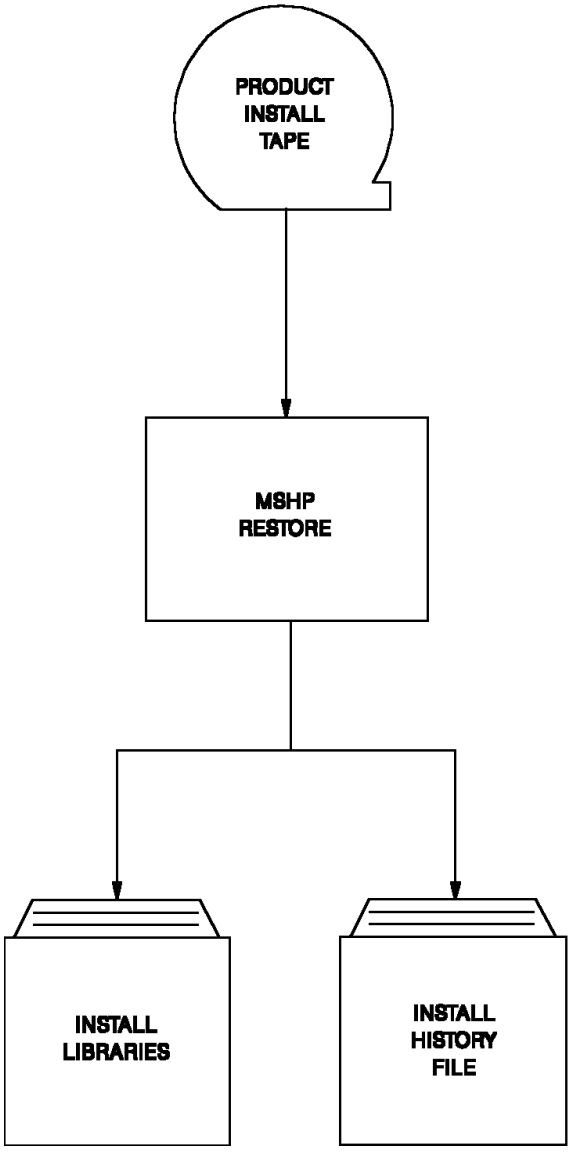


Figure A-2: CAINSTB2 Execution

A.2 Migration of CAI Products into Production

CAINSTB3: CAINSTB3 is used to migrate a product into the production libraries and history file when a product was installed to an installation library and history file. This step is executed only after product installation, customization, verification and testing are complete.

The sample JCL members are located on the restored source library in the Zòsublibrary.

After this process is complete, the installation libraries and history file can be deleted.

CAINSTB3 Execution: CAINSTB3 merges the tested product or products into the production libraries and history file.

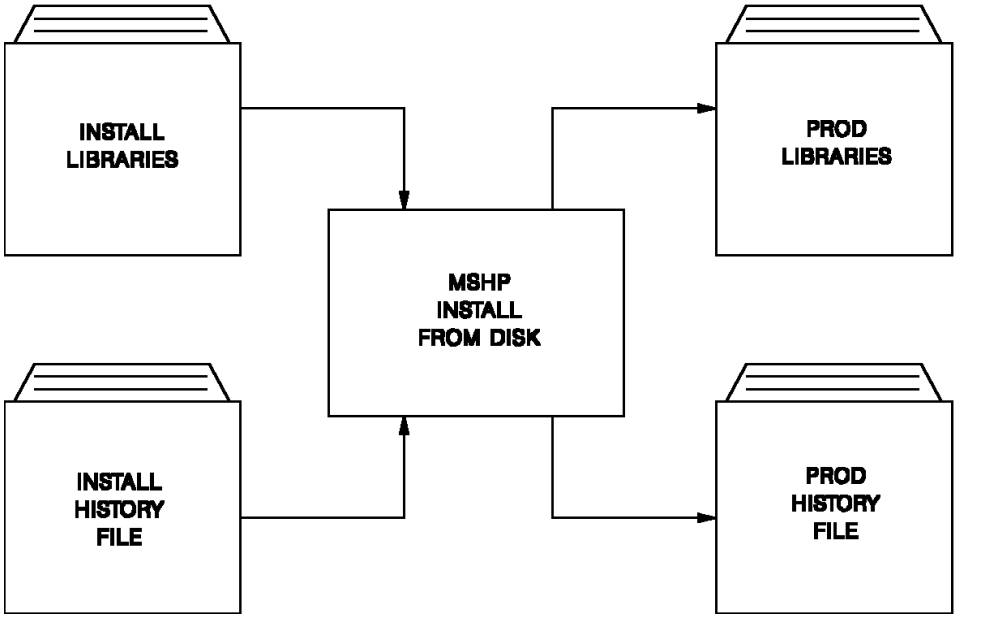


Figure A-3: CAINSTB3 Execution

A.3 MSHP Installation JCL Customization

The MSHP installation JCL contains variables that must be modified to tailor the JCL to your specific environment. The variables are specified in a worksheet contained in this Installation Guide. These variables will be manually changed via a user editor.

Variable symbols present in the MSHP installation JCL follow:

FOR JOB CAINSTB1

Variable	Description
@CUSTNME	Customer name used to personalize the MSHP history files.
@CUSTADD	Customer address used to personalize the MSHP history files.
@CUSTPHN	Customer phone # used to personalize the MSHP history files.
@PROGNME	Customer programmer name used to personalize the history files.
@HISTVOL	Volume serial # where the CAI production history file will reside.
@HISTREL	Relative track or block where the CAI production history file is to begin.
@HISTEXT	Number of tracks or blocks to be allocated to the CAI production history file.
@INSTVOL	Volume serial # where the temporary installation history file will reside.
@INSTREL	Relative track or block where the temporary installation history file is to begin.
@INSTEXT	Number of tracks or blocks to be allocated to the temporary installation history file.
@DLIBVOL	Volume serial # where the CAI production library will reside.
@DLIBREL	Relative track or block where the CAI production library is to begin.
@DLIBEXT	Number of tracks or blocks to be allocated to the CAI production library.
@TAPECUU	The device address where the product tape will be mounted.

FOR JOB CAINSTB2

Variable	Description
@CUSTNME	Customer name used to personalize the MSHP history files.
@CUSTADD	Customer address used to personalize the MSHP history files.
@CUSTPHN	Customer phone # used to personalize the MSHP history files.
@PROGNME	Customer programmer name used to personalize the history files.
@INSTVOL	Volume serial # where the CAI product installation history file will reside.
@INSTREL	Relative track or block where the CAI product installation history file is to begin.
@INSTEXT	Number of tracks or blocks to be allocated to the CAI product installation history file.
@ILIBVOL	Volume serial # where the CAI product installation library will reside.
@ILIBREL	Relative track or block where the CAI product installation library is to begin.
@ILIBEXT	Number of tracks or blocks to be allocated to the CAI product installation library.
@PRODCDE	Product code for the product tape to be installed. For example, YD540 for CA-DYNAM release 5.4.
@PRODUCT	Product name. For example DYNAM for the CA-DYNAM product.
@TAPECUU	The device address where the product tape will be mounted.

FOR JOB CAINSTB3

Variable	Description
@HISTVOL	Volume serial # where the CAI production history file will reside.
@HISTREL	Relative track or block where the CAI production history file is to begin.
@HISTEXT	Number of tracks or blocks to be allocated to the CAI production history file.
@DLIBVOL	Volume serial # where the CAI production library will reside.
@INSTVOL	Volume serial # where the CAI product installation history file will reside.
@INSTREL	Relative track or block where the CAI product installation history file is to begin.
@INSTEXT	Number of tracks or blocks to be allocated to the CAI product installation history file.
@ILIBVOL	Volume serial # where the CAI product installation library will reside.
@PRODCDE	Product code for the product tape to be installed. For example, YD540 for CA-DYNAM release 5.4.
@TAPECUU	The device address where the product tape will be mounted.

Appendix B. TSSXTEND and TSSRECVR

Once you have installed CA-Top Secret, you can modify your Security File by altering its size or by changing the Security Encryption Key. Both of these functions use the TSSXTEND utility and both can only be performed by the MSCA.

The TSSRECVR utility is used for Security File recovery, regardless of whether your site has selected the Automatic Backup feature (which is recommended) or whether your site prefers to manually backup the Security File onto tape.

B.1 TSSXTEND - Extend Security File

The TSSXTEND Utility allows an MSCA to enlarge or reduce the size of the CA-Top Secret Security File, or change the Security Encryption Key. TSSXTEND is used in combination with TSSMAINT (refer to 3.15, “Step 13: Create the Security File” on page 3-33 for more information on TSSMAINT). Enlarging or reducing the size of the Security File is a three-step process that must be performed by the **MSCA.**

1. Create a new Security File using the TSSMAINT utility.
2. Issue a TSS MODIFY(BACKUP) command to copy the new Security File to the backup Security File.
3. Use the TSSXTEND Utility to copy the backup Security File to a new Security File.

Note: You can check how full the security file is by issuing the TSS MODIFY command and reviewing the message TSS9501I SECURITY FILE (NNN%). (nnn%) represents the percentage of the security file being used.

B.1.1 Special Considerations

There are some important considerations in making a smooth transition from the existing backup file to the new Security File.

- Renaming new files:

  Both the new Security File and the new backup Security File must be created with names other than those of the old files.

- Creating a new backup Security File:

  Once you have created a new Security File, you must create a Backup File (with the same space attributes assigned to the new Security File) using TSSMAINT (see 3.15, “Step 13: Create the Security File” on page 3-33). To preserve the existing Backup File, use a name other than the name of the existing file.

- Changing the JCL statements:

  Once the new Security and Backup Files are created, you must change the PROC statements in your JCL procedures (namely, TSS STC, TSSB STC, and backup and recovery procedures) to point to the new files.

- Changing the encryption key:

  TSSXTEND has the capability of changing your company's Security File encryption key. Ordinarily, your company's encryption key should **never** be changed. However, if it is suspected that the integrity of your key has been violated, a new key can be supplied using TSSXTEND. If you must change the Security File encryption key, it is necessary to run TSSKEY using the APPLY CRYPTKY control statement, and supply the new key so that CA-Top Secret can operate correctly.

- Deletion of old files:

  Computer Associates recommends that the old Security File and Backup files are not deleted until your installation is sure that the file enlargement/replacement was successful. Provided that you have supplied the names of the new files in all the necessary JCL PROCs, both the new and old files can reside on disk without causing CA-Top Secret to malfunction.

B.1.2 Creating the New Security File

As previously mentioned, the first step in enlarging the Security File is to create a new larger one using the TSSMAINT utility. To create the new file, follow the procedures and use the JCL as explained in 3.15, “Step 13: Create the Security File” on page 3-33. Keep in mind that the name specified for the **ID=** parameter must be different than the name of the existing Security File.

B.1.3 Updating the Backup Security File

Next, you must update the backup Security File to have it match the new Security File. Issue a TSS MODIFY(BACKUP) command to do this.

B.1.4 Run TSSEXTEND Utility

The following JCL can be used to copy the contents of the old backup Security File into the new Security File. It is assumed that the new Security File was already created (via TSSMAINT).

```
* $$ JOB JNM=TSSEXTEND,CLASS=0,DISP=D
// JOB TSSXTEND
// ID USER=MSCA,PWD=TORONTO
// ASSGN SYS0XX,DISK,VOL=XXXXXX,SHR
// ASSGN SYS0YY,DISK,VOL=YYYYYY,SHR
// DLBL SECFNEW,'CAI.TOP.SECRET.SECURITY.FILE',99/365
// EXTENT SYS0XX,XXXXXX,1,0,XXX,XXX <-REPLACE WITH CORRECT EXTENT
// DLBL CAIBKUP,'CAI.TOP.SECRET.BACKUP.FILE',99/365
// EXTENT SYS0YY,YYYYYY,1,0,YYY,YYY <-REPLACE WITH CORRECT EXTENT
// EXEC TSSXTEND
COPY BACKUP
OLDKEY=????????????????
NEWKEY=????????????????
WHOHAS
/*
```

The following keywords are valid for TSSXTEND:

COPY BACKUP	Use backup security as input for expand job (recommended).
COPY SECURITY	Use primary security file as input for expand job.
WHOHAS	Reorganize WHOHAS records as a part of the expand job (optional).
OLDKEY	Old file encryption key.
NEWKEY	New file encryption key (can be identical to OLDKEY).

The OLDKEY and NEWKEY fields must be a 16-character hexadecimal number. Comments cannot be added to these fields.

!!!!!!SAFEGUARD YOUR KEY FOREVER!!!!!!!

B.1.5 Messages and Codes

The User Abend codes that are generated by an unsuccessful execution of TSSXTEND are listed in the *Messages and Codes Guide.*

B.2 TSSRECVR

This section presents two methods and procedures for using TSSRECVR for Security File recovery. The first method is for sites using the Automatic Backup feature, in accordance with the suggested installation procedure. That is, a backup Security File exists on DASD, RECOVERY is ON, and CA-Top Secret automatically performs Security File backup via the BACKUP control option. The second method is for sites who manually backup the Security File onto tape.

Note: If the Security File is damaged or lost, and CA-Top Secret must be brought down, in order to start the recovery procedures, the ACID and password given after the P TSS command must belong the the MSCA (since it's the only ACID residing in storage).

B.2.1 Recovery Procedure for Automatic Backup

If the primary Security File is lost or damaged and you have been using the Automatic Backup feature, follow this procedure to recover the Security File.

Step 1: Stop CA-Top Secret by replying SHUTDOWn to the outstanding reply in the security service (see 5.4, “CA-Top Secret Shutdown” on page 5-6 for details).

Step 2: After TSS has completed the shutdown process, restart CA-Top Secret using the backup procedure TSSB.

```
reply id EXEC PROC=TSSB
```

TSSB was created at installation and has the following characteristics:

- The SECFILE DLBL statement points to the Backup Security File.
- Automatic Backup is turned OFF (no CAIBKUP DLBL statement).

Step 3: At this point, you will be using a backup Security File that should be, at most, 24 hours out-of-date. Security File recovery is performed by applying the TSSRECVR routines to the backup Security File. An up-to-date Security File is reconstructed by applying, from the Recovery File to the backup Security File, all the changes that occurred since the last backup of the Security File.

If you are using your only copy of the Backup Security File and suspect that damage to the Security File was caused by a command function update, make a copy of the backup Security File before proceeding. The TSSBCKUP JCL can be used (TSSBCKUP was set up at installation and placed into the JCL library).

Step 4: In order to avoid duplication of TSS command functions on the Recovery File resulting from the recovery process, turn RECOVER OFF by entering:

```
F TSS,RECOVER(OFF)
```

Step 5: You are now ready to begin phase 1 of the recovery. You will be using the TSSRCVR1 JCL to avoid the problem of double updating by:

- Specifying the time and date in the EXEC PARM of the last **completed** backup of the Security File.
- Creating a file of TSS command functions.

To execute TSSRCVR1, type:

```
R RDR, TSSRCVR1
```

TSSRCVR1 was created during installation and placed into the JCL library. While both DATE and TIME should be coded in the EXEC PARM field before executing TSSRCVR1, failure to code the DATE parameter will result in an abend. Refer to 3.21.4, “Setup Recovery Procedures” on page 3-49, for more information on coding DATE and TIME. For example,

```
// JOB TSSRCVR1
// ID USER=MSCA,PWD=TORONTO
// DLBL TSSCMDS,'CAI.TOP.SECRET.RECV.WORK',,0
// EXTENT SYS0XX,XXXXXX,1,0,X,Y <--REPLACE WITH CORRECT EXTENT INFO
// EXEC TSSRECVR,PARM=TIME(0000),DATE(00000)'
/*
/&
```

Note: If the TSS command contains the keyword TARGET, when it is placed in the recovery file on the system it was entered, the TARGET keyword will be commented out and replaced with TARGET(=). This is to prevent duplicate permits on remote nodes when recovery is done on one system. For example:

```
TSS TARGET(=,NODE2) PERMIT(USER1) DSN(ABC.) ACCESS(READ)
```

will show up in the output of TSSRCVR1 as:

```
TSS TARGET(=) /*RGET(=,NODE* / PERMIT(USER1) DSN(ABC.) ACCESS(READ)
```

Step 6: Apply the changes collected in Step 5 to the backup Security File (phase 2 of recovery) by typing:

```
R RDR, TSSRCVR2
```

Note: TSSRCVR2 should run under the authority of the MSCA. This ensures that commands will not FAIL due to insufficient authority. To accomplish this, enter:

```
TSS ADD(STC) PROC(TSSRCVR2) ACID(msca) STCACT
```

The STCACT keyword is optional and has the effect of prompting the operator console for a userid and password when the procedure is started. Finally, this information is written to the Audit File.

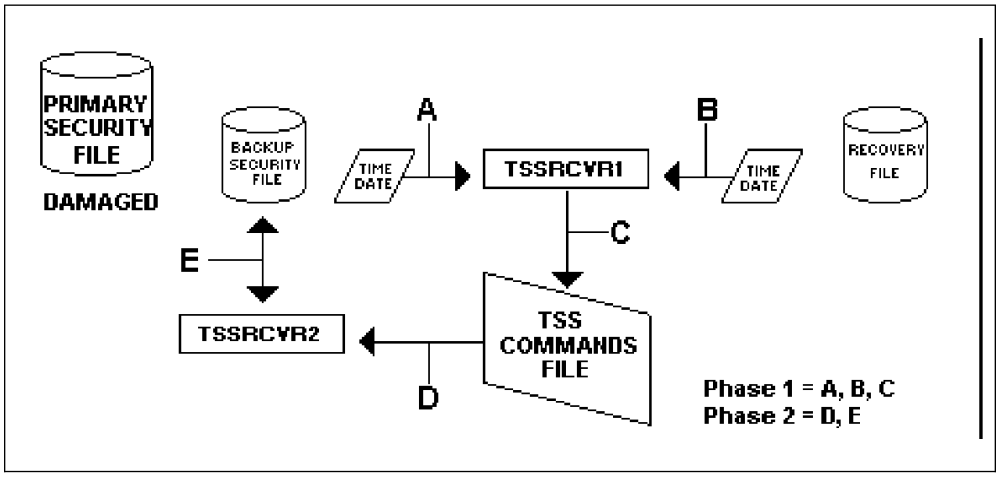


Figure B-1. TSSRECVR Phase 1(Step 5) & 2(Step 6)

After completing Step 6, the backup Security File has been recovered. You can use the backup Security File to test that the file was properly recovered. Remember, however, that you must use the TSSB STC, Automatic Backup is turned OFF, and RECOVER is OFF. When you are sure that the recovery was successful, continue to Step 7.

Step 7: Stop CA-Top Secret by typing:

```
reply id SHUTDOWN
```

Step 8: The remaining steps of this recovery procedure copy the updated data from the backup Security File into the primary Security File so that the TSS task can be restarted.

If the primary Security File still exists on DASD (damage was not a result of hardware problems), skip to Step 9; otherwise, a new Security File must be initialized using the TSSMAINT utility explained in Chapter 2. Besides the ID= parameter, the new primary Security File that is created must have the same parameter values as the original Security File. The ID= parameter should be set to ID=PRIMARY.

Step 9: Create a new started task modeled after the TSS.PROC in which the CAISECF.DLBL statement points to the backup Security File, and the CAIBKUP DLBL statement points to the primary Security File. You can use the following TSSC JCL.

```
ACC S=CAILIB.TSSVSE30
CATALOG TSSC.PROC R=Y DATA=YES
ASSGN SYSLST,IGN
ASSGN SYS0XX,DISK,VOL=XXXXXX,SHR
ASSGN SYS0YY,DISK,VOL=YYYYYY,SHR
// ASSGN SYS026,DISK,VOL=JXC356,SHR
// DLBL CAIBKUP,'CAI.TOP.SECRET.SECURITY.FILE',99/365
// EXTENT SYS0XX,XXXXXX,1,0,XX,XXX
// DLBL CAISECF,'CAI.TOP.SECRET.BACKUP.FILE',99/365
// EXTENT SYS0YY,YYYYYY,1,0,YY,YYY
// EXEC TSSMNGR,SIZE=128K
/*
```

Step 10: Start CA-Top Secret using the newly created procedure, TSSC by typing:

```
reply id EXEC PROC=TSSC
```

Step 11: Force an Automatic Backup by typing the following:

```
reply id BACKUP
```

Step 12:

Stop the TSS task by typing:

```
SHUTDOWN
```

After completing Step 12, CA-Top Secret can be restarted using the TSS PROC. The primary Security File is recovered. Make sure that after restarting TSS, RECOVER is ON.

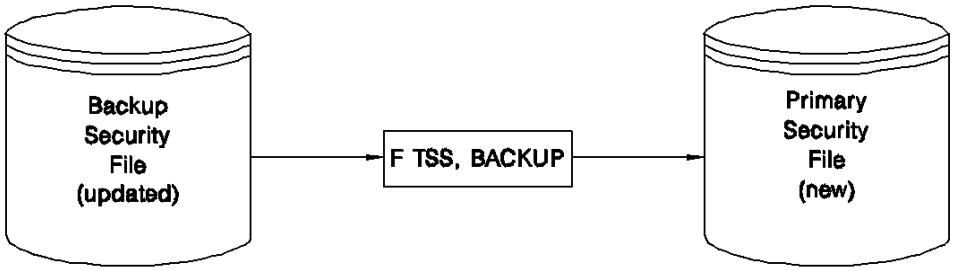


Figure B-2. Restoring Primary Security File, Step 11

B.2.2 Manual Recovery Procedure

If the Security File is lost or damaged and your site is not using the CA-Top Secret Automatic Backup feature, follow this procedure to manually recover the Security File. Note that for a manual recovery, the recovery feature, RECOVER(ON) must have been in effect.

Step 1: Stop CA-Top Secret by typing the following console command:

```
SHUTDOWN
```

Step 2: The most current backup copy of the Security File must be on DASD, not tape, to proceed. You must manually restore the file. The JCL library should contain two JCL procedures that perform file restoration: TSSRESTR and TSSRESTN. The two procedures were set up at installation and perform the following tasks:

TSSRESTR Restores a file from tape to DASD.

TSSRESTN Allocates new space on DASD and restores a file from tape to the newly allocated DASD space.

Choose either TSSRESTR or TSSRESTN and restore the backup copy of the Security File to DASD.

Step 3: Restart CA-Top Secret by typing the following console command:

```
EXEC PROC=TSS
```

Step 4: To avoid duplication of TSS command functions, type the following TSS command from a Console reply to TSSMNGR:

```
F TSS,RECOVER(OFF)
```

Step 5: You are now ready to retrieve the changes from the Recovery File using the TSSRCVR1 JCL. TSSRCVR1 reads the Security File time stamp and retrieves changes from the Recovery File after that time to avoid problems of double-updating. TSSRCVR1 was set up at installation and placed into the JCL library. Note that if it is necessary to override the time stamp, you must enter the TIME(hhmm) and DATE(yyddd) in the EXEC PARM field before executing TSSRCVR1. To execute TSSRCVR1, type:

```
R RDR, TSSRCVR1
```

Step 6: Apply the changes (collected in Step 5) to the Security File using the TSSRCVR2 JCL by typing:

```
R RDR, TSSRCVR2
```

After completing Step 6, the Security File is recovered.

Step 7: Turn the Recovery File back on by typing:

```
TSS,RECOVER(ON)
```

Appendix C. Installation Checklist

The following checklist should be used to ensure that all the necessary installation steps have been taken.

C.1.1 C.1.1 After Downloading CA-Top Secret

[ ] If converting from CA Top-Secret 2.3 or Alert/CICS/VSE, point libdef to new library and run conversion utilities while old security systems are still active.

[ ] CA-CIS is installed and your site's LMP key or IBMKEY is defined

[ ] Sample JCL Z copybooks are tailored to your site's standards

[ ] Apply Encryption key with TSSKEY.Z copybook

[ ] Encryption key stored in safe place

[ ] Program library is shared across multiple CPUs (if required)

[ ] Add CA-CIS and CA-Top Secret library to LIBDEF search chains in IPL procedure

[ ] Add Standard label information from TSSLABEL.Z copybook to IPL procedure

[ ] Ensure a permanent ASSGN for each DASD in TSSLABEL exists in IPL JCL procedure for each partition.

[ ] Make sure any modules from prior release of CA Top-Secret 2.3 are not in searched libraries

[ ] Define startup of CASAUTIL and CA Top-Secret in IPL procedure

[ ] Perform First IPL

After first IPL:

[ ] Define Auto Commands file with TSSAUTO.Z copybook

[ ] Edit control options from TSSPARM.Z copybook and submit jcl to define options.

[ ] Define master Security file with TSSMAINS.Z copybook

[ ] Define backup Security file with TSSMAINB.Z copybook

[ ] Ensure CAIBKUP DLBL is only in STDLABEL on one CPU or BACKUP(OFF) defined on other CPU's if files are shared.

[ ] Define Recovery file with TSSMAINR.Z copybook

[ ] Define Audit files with TSSMAINA.Z copybook

[ ] Define CPF recovery file with TSSMAINC.Z copybook.

[ ] Correct and catalog TSS.PROC and TSSB.PROC into IJSYSRS.SYSRES library

[ ] Modify IPL procedure to start FB partition outside of power and place // EXEC PROC=TSS in JCL procedure for FB.

[ ] IPL second time

After Second IPL

[ ] Define Default Security values using TSSINIT.Z copybook

[ ] Modify CICS JCL to support new Security. See CICS Implementation guide.

Index

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

User Registration Form

If you are interested in registering as a user of Computer Associates software, please complete the information below. Send it to the following address:

Computer Associates International, Inc.
ATTN: User Registration
One Computer Associates Plaza
Islandia, NY 11749

Registration entitles you to receive such items as:

- newsletters
- product release announcements
- information about User Groups
- information about CA conferences
- client questionnaires

You *do not* have to be the primary contact for CA software within your company or organization. All you have to be is interested in CA software, how it is used, and where it is headed.

Product(s): _____

_____

_____

_____

Site ID: _____
(Enter UNKNOWN if you do not know your Site ID.)

Name: _____

Position: _____

Company or organization: _____

Address: _____

Address: _____

Phone No.: _____

Date: _____

I would like additional information on: _____

Demand Analysis Request Form

Please use this form to make suggestions for product improvement. Send it to the following address.

Computer Associates International, Inc.
ATTN: Demand Analysis Requests
One Computer Associates Plaza
Islandia, NY 11749

Be sure to print your name and address below if you would like a reply.

Name: _____

Position: _____

Company or organization: _____

Address: _____

Address: _____

Phone No.: _____

Date: _____

Is DAR from a User Group? (Yes/No) ___ User Group ID: _____

Priority: _____

Description of Requested Item:

_____

_____

_____

_____

_____

_____

_____

_____

_____

CA Account Manager:	CA Office:
Input by:	Date:

Reader Comment Form

CA-Top Secret Installation and Maintenance Guide

Release 3.0 VSE

Document Number: I101TS300NE

Please use this form to tell us how you use this manual and to make suggestions for improvement. Send it to the following address. (To request additional publications, call 1-800-841-8743 or check the CA home page (http://www.cai.com) on the Internet. To ask questions about the functionality of CA products, contact your CA representative.)

Computer Associates International, Inc.
ATTN: Reader Comment Form
One Computer Associates Plaza
Islandia, NY 11749

Be sure to print your name and address below if you would like a reply.

Name: _____

Position: _____

Company or organization: _____

Address: _____

Address: _____

Phone No.: _____

Date: _____

Years of experience with this CA product: _____

Overall Rating

	Good	Fair	Poor	Why?
Accuracy				
Organization				
Clarity				

Reference Aids

	Good	Fair	Poor	Why?
Contents				
Index				
Glossary				
Reference to Other Manuals				

Clarity

	Good	Fair	Poor	Why?
Instructions and Procedures				
Examples				
Graphics				

How Manual Is Used:

How do you use this manual in your job?

How often do you use this manual in a week?

Suggestions:

What do you like about this manual?

What do you dislike about this manual?

What would you like us to change?

Additional Comments:

DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO660W POSTSCRIPT FILE'S LINE LENGTH VIOLATES ADOBE STRUCTURING CONVE
NTIONS.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S