CA-Top Secret®

Troubleshooting Guide
Release 3.0
VSE

SP1

Second Edition, September 2000

Contents

About This Guide

Purpose

This guide enables CA-Top Secret administrators to accomplish the following:

- Verify that a problem actually exists, categorize the problem, and eliminate obvious causes;
- Diagnose and resolve specific problem categories by providing step-by-step procedures or checklists for each category;
- Conduct clear, streamlined discussions with CA-Top Secret Technical Support by providing guidelines on when to call technical support, and what information to provide to the CA-Top Secret support analyst.

Obviously, this guide alone will not solve all of your CA-Top Secret problems. It can, however, show you how to use your CA-Top Secret documentation and product features to make troubleshooting more effective and efficient. This should allow you to solve many problems on your own.

If you must take advantage of CA-Top Secret support, the checklists and job aids provided in this guide will help you get the most out of your conference with the support analyst, and help to resolve your problems quickly and efficiently.

Intended Audience

This guide is intended for individuals who are charged with the maintenance of CA-Top Secret, or for any security administrator, operator, auditor, or programmer who must diagnose and resolve CA-Top Secret related problems.

Prerequisites

This guide assumes that the reader:

1. Has attended the CA-Top Secret Basics class or is familiar with the TSS commands, control options, CA-Top Secret utilities, CA-Top Secret concepts and facilities, and CA-Top Secret documentation.
2. Is familiar with the following guides:
   - *Planning Guide*
   - *User Guide*
3. Has a basic knowledge of data processing and data security concepts.

Note: CA-Top Secret is easy to install, customize, administrate, and use. Since this *Troubleshooting Guide* is written explicitly to simplify users' tasks, help us make CA-Top Secret even better by using the forms in the back of this guide to offer us your feedback on the functionality of this guide. Your help will be greatly appreciated.

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. How to Use This Guide

This chapter presents an overview of the CA-Top Secret troubleshooting process, and explains how to use this guide to direct the process.

1.1 General Approach to Troubleshooting

There are probably as many approaches to troubleshooting as there are things to troubleshoot. While any specific approach will depend on the system under consideration, almost all diagnostic approaches follow a basic procedure:

Figure 1-1. Basic Troubleshooting Process

1.2 CA-Top Secret Approach to Troubleshooting

CA-Top Secret contains features and utilities which allow this basic troubleshooting approach to be incorporated into a simple three phase procedure:

Phase 1	Verify and categorize the problem.
Phase 2	Perform diagnostic or checkout procedure(s) to determine the cause of the problem. If a resolution is found, implement it, and then test it. If it works, your problem is solved.
Phase 3 (optional)	Organize diagnostic information, and call Technical Support for assistance with diagnosis and resolution of the problem.

1.3 Using This Guide

The following explanations describe how this guide will support each Troubleshooting Phase, and how the user should go about using the documentation.

1.3.1 Phase 1—Problem Identification

This is where you should begin your diagnostic effort. You should be trying to answer the following questions: "Is this problem really a problem, what type of problem is it, and where do I go for help?"

Phase 1 assists you by providing:

1. A simple procedure for categorizing problems. The supported general categories are:
   a. Abends
   b. Customization
   c. Facility Authorization
   d. Logging of Events
   e. Message Production/Suppression
   f. Resource Authorization
2. Directions and references to other CA-Top Secret guides and utilities which allow you to diagnose and resolve the problem.

1.3.2 Phase 2—Diagnostic Procedures

Once you have been able to verify that there is indeed a problem, and have been able to isolate it to one of the supported categories (listed under section 1.3.1, "Phase 1—Problem Identification") you should refer to the appropriate diagnostic procedure or checklist which is provided for each general category.

Each procedure is designed to either allow you to determine the cause and resolution of the problem yourself, or to assemble as much information as you can about the problem. This will enable you to make effective use of any discussions you may have with CA-Top Secret Technical Support.

1.3.3 Phase 3—CA Technical Support

This phase provides a checklist of items and information which will greatly facilitate any discussions with the CA-Top Secret support analyst. This is actually the easiest phase of your investigation since you will have assembled most of this data as part of the prior troubleshooting phases.

Phase 3 also describes how CA-Top Secret's Technical Support Center is organized, and how your call will be processed.

1.3.4 TSSSIM Documentation

Chapter 5 of this guide contains complete documentation for the TSSSIM simulation utility. The simulator gives administrators the ability to test CA-Top Secret permissions on their Security File without causing a violation against the user ACID. Thus, the administrator may simulate access attempts to determine if CA-Top Secret will grant a user access to a resource, and display the permission which granted or denied access.

The documentation contains a complete overview of the utility, plus instructions and examples of each TSSSIM command. An explanation of how to interpret simulator trace information is included, as well as instructions on how to use the TSSSIM menus.

1.3.5 Appendixes

The Phase 2 procedures will often direct you to one or more of the following trouble-shooting aids:

A. Reports

B. Diagnostic Traces

C. Message Display/Suppression Algorithm

Chapter 2. Phase 1 — Problem Identification

STEP 1 Determine nature of the problem.

Has anything changed recently?

YES If the ACID has changed lately, run the TSSAUDIT changes() report.

If there were any recent software upgrades, investigate the changes.

NO GO TO STEP 2.

STEP 2 Attempt to recreate the problem on the system, or use TSSSIM to simulate the problem.

Did the problem reoccur?

YES GO TO STEP 3.

NO This indicates a user error which may involve user training or some other administrative action. Take the appropriate action.

GO TO STEP 5

STEP 3 Did the system display a TSS message, DRC code, or other abend code?

YES Refer to *Messages and Codes Guide* to determine the cause of the problem. Take the corrective action which corresponds to the message or code.

GO TO STEP 5.

NO GO TO STEP 4.

STEP 4 Categorize the problem, if possible, and go to the appropriate Phase 2 procedure.

The following list contains general categories only. If you cannot locate an exact category in the list, try to apply the basic troubleshooting principles of a related category before contacting the CA-Top Secret Support Center. If the problem does not correspond to one of the categories listed next, or appears to be caused by CA-Top Secret internals, go directly to the Phase 3 documentation in this guide.

Category	Procedure
Abends	Abends
Customization	Customization
Facility Access	Access Incorrectly Allowed
	Signon Passwords Not Being Checked
	CPU Restriction Not Honored
	Access Incorrectly Denied
Logging	CA-Top Secret Not Logging Violations
	CA-Top Secret Not Logging Access or Initiations
Messages	Users Not Receiving CA-Top Secret Messages
	Console Not Receiving CA-Top Secret Messages
Resource Access	Data Set or Volume Access Incorrectly Denied
	Data Set Passwords Not Being Checked

Does the problem pertain to or is it related to a listed category?

YES GO TO STEP 5

NO Go to Phase 3

STEP 5 Before proceeding to Phase 2, let's review the CA-Top Secret security validation algorithm to ensure that the problem is not due to a misunderstanding of how CA-Top Secret works. Consult the *User Guide* for a more detailed explanation of the algorithm.

The Security Validation Algorithm

The CA-Top Secret algorithm takes the following factors into consideration when processing a request to access a resource.

- **Ownership vs. Authorization**

  Ownership overrides authorization. If CA-Top Secret finds two TSS entries, one giving ownership of a resource to an ACID and the other giving authorization, ownership will prevail, allowing the ACID total access.

- **Order of Search**

  CA-Top Secret searches the security records in the following order:

  1. ACID's security record
  2. Profile(s) attached to the ACID
  3. The ALL Record

- **LCF Lists**

  LCF lists are used to restrict the use of a facility's commands on a user-by-facility basis. There are two types of LCF lists: inclusive and exclusive. Within a particular facility, either the inclusive or exclusive restriction (not a combination of both) should be used. If both types are used within a single facility, inclusive is checked first, then exclusive. For a further discussion of LCF lists, refer to the *User Guide.*

- **Best Match Criteria**

  When more than one relevant PERMIT is encountered in a security validation search, CA-Top Secret employs the "best match" criteria. "Best match" refers to the closeness of the matchup between the resource identifier in the PERMIT and the resource identifier for which access is being requested. Generic prefixing and masking are often the culprits responsible for (un)authorized access discrepancies.

- **AUTH Control Option**

  The AUTH control option governs whether or not CA-Top Secret will merge the results of the security record search.

- **Volume vs. Data Set Authorization**

  When determining access to a particular DASD data set, CA-Top Secret must evaluate both volume and data set access authorizations. Volume level checking can optionally be bypassed. However, in situations where both volume and data set level checking is being done, CA-Top Secret performs volume—level checking first. Thus, a request to access a data set can be granted or failed strictly on the basis of the user's volume access authorization.

- **MODE Control Option**

  The mode setting specifies the level of security at which security validation will be performed. MODE options range from DORMANT to FAIL. CA-Top Secret uses this setting to determine its response to security validation requests. See the CA-Top Secret*User Guide* for further information on the modes and how they affect security calls.

Based on the CA-Top Secret security validation algorithm, does a problem still exist?

YES Go to Phase 2.

NO Review the CA-Top Secret documentation and examine your existing security definitions. Remember that CA-Top Secret offers customer assistance in the form of education classes, on-site security consulting, and professional services.

Chapter 3. Phase 2 — Diagnostic Procedures

Phase 2 of the troubleshooting procedure describes the steps that should be taken to diagnose the root of your problem. The procedures in Phase 2 are broken down by categories. (Prior to reaching this phase you should have placed your problem into a specific category.) To locate the proper procedure, find the exact category heading or the heading that most closely resembles the type of problem you are experiencing. The general categories are:

Problem	**Procedure**
Abends	Abends
Customization	Customization
Facility Access	Access Incorrectly Allowed
	Signon Passwords Not Being Checked
	CPU Restriction Not Honored
	Access Incorrectly Denied
Logging	CA-Top Secret Not Logging Violations
	CA-Top Secret Not Logging Access or Initiations
Messages	Users Not Receiving CA-Top Secret Messages
	Console Not Receiving CA-Top Secret Messages
Resource Access	Data Set of Volume Access Incorrectly Denied
	Data Set Passwords Not Being Checked

3.1 TSSSIM

TSSSIM (explained in detail in Chapter 5) is a valuable tool when trying to find the underlying problems in (un)authorized access discrepancies. If you are unsure of how CA-Top Secret reacts to certain authorization definitions, use TSSSIM to test those authorizations in a simulated security environment. TSSSIM uses the CA-Top Secret security validation algorithm and immediately displays whether access is granted or denied. If denied, TSSSIM also displays the reason for denial in the form of a DRC and CA-Top Secret messages.

3.2 Abends

Most Ca-top Secret abend messages and dump titles contain and document the abend code, program name , and program offset associates with the error. When a CA-top Secret dump exists in a VSE DUMP library or in SYSLST, various methods can be used to gather further details about the error. Similary, when a CA-Top Secret dump is being viewed via INFOANAL, the PRINT FORMAT command can be used to obtain an excellent summary of the error. Whenever possible please try to provide this output to Top Secret technical support. When reproting an abend of CA-top Secret since this information will allow faster and more accurate problem isolation and resolution.

Please see the IBM provided VSE/ESA Diagnosis Tools , Document Number SC33-6614 for further detail on VSE problem determination and Dump handling.

STEP 1 If the CA-Top Secret partition abends during processing, gather the following information.

1. Provide an actual system dump from the dump data set onto either a 6250 BPI magnetic tape or cartridge. Please ensure the dump includes all of the system getvis areas (SVA).
2. Provide all copies of the system log data from 15 minutes before and 5 minutes after the abend occurred. Hardcopy is preferred.
3. If the abend was produced by a batch job, provide a copy of the JCL for that batch job and the VSE joblog.
4. If TSS message TSS9999E indicated the abend, record the text of the message, including all register and offset information which follow the message.
5. Copy all other CA-Top Secret messages that appear with the job or session in the order in which they occur.
6. Refer to the Phase 3 procedure for a checklist of general information required by technical support, and for instructions on how to call technical support.

STEP 2 Proceed to Phase 3—Contact CA-Top Secret Technical Support.

3.3 Customization

Customization problems can stem from any one of the three supported CA-Top Secret customization techniques. Namely,

- the VSE Security Interface,
- the Installation Exit, or
- the Application Interface.

Problems usually occur within the parameter lists that are passed to CA-Top Secret by the customized code. There are a number of tools that can be used for diagnostics:

- Information Feedback Areas containing DRCs
- CA-Top Secret Messages, and IBM Messages
- Traces containing DRCs (see Appendix B)
- Dumps (forced or from DIAGTRAP)

3.3.1 Tips and Checkpoints

Customization code can get tricky at times. This section outlines those little things that can easily be overlooked when developing the customization code and can lead to problems. (The old forgetting a semicolon!) Locate the customization facility you are using and check that the listed items have been incorporated into your customization code.

- Parameter lists
  - Is the correct format being used?
  - Have you supplied a length?
- RACDEF and RACLIST have NOT been used for customization.
- The INSTLN operand of the security macro is being used to obtain information feedback.
- The ACEE= parameter is present for multi-user address space on all requests to security (RAC macros or RACROUTE), except in task per user environments (TCBSENV).

- RACROUTE
  - Used by current releases of VSE (state-of-the-art).
  - Preferred to the usage of RACINIT, RACHECK, and FRACHECK.
  - Your call is one that is supported by CA-Top Secret—that is, REQUEST=AUTH, REQUEST=FASTAUTH REQUEST=VERIFY, REQUEST=EXTRACT
  - A valid Class Name is being used. See the *User Guide* for valid Class Names.
  - If using the Dynamic Extract/Update Facility, you are not trying to extend or reduce the size of the INSTDATA (Installation Data) field.
  - If using RACROUTE REQUEST=VERIFY to build a facility, review the facility entries to make sure they were entered correctly.
  - RACROUTE REQUEST=VERIFY, ENVIR=DELETE - ensure ACEE parameter points to the address if the address of the ACEE is returned with RACROUTE REQUEST=VERIFY, ENVIR=CREATE.
  - Program issuing RACROUTE REQUEST=VERIFY (top RB) matches a valid facility.
  - If the facility does not have the NOAUTHINIT attribute, ensure that the program is issuing RACROUTE REQUEST=VERIFY APF authorized in system Key 0 or Supervisor state.

    Note: NOAUTHINIT is only valid for STC.
  - If using RACROUTE REQUEST=VERIFY use the installation feedback area for returned DRCs.

3.3.1.1 Installation Exit

- The Installation Exit is executed in Key 0, Supervisor State, when live; therefore, it must be coded and tested with great care.
- The activation matrix contains a nonzero entry for the desired function.
- The customized code is in the appropriate section of the TSSINSTX module.
- The EXIT control option is ON. TSS,STATUS will list the status of EXIT.
- Under TSO, the installation exit code can be tested in Key 8 by:
- Loading TSSINSTX
- Building your own parameter list as expected by the exit. Parameter lists are described in the TSSINSTX code found in the Optional Materials file on the CA-Top Secret distribution tape.
- Passing control to TSSINSTX
- Examining the parameter list, feedback areas, and return codes

3.3.1.2 Application Interface

- All the necessary Request Record fields are supplied.
- Samples of the Application Interface can be found in the Optional Materials file on the CA-Top Secret distribution tape and in the *User Guide.*

3.3.2 Troubleshooting—Procedure

The general procedure that can be used to troubleshoot customization problems follows.

STEP 1 Write down all message numbers and text displayed by CA-Top Secret or MVS. Refer to the *Messages and Codes Guide* to determine the meaning of the CA-Top Secret messages and the appropriate action that should be taken to correct the problem.

Did the messages expose the problem?

YES GO TO STEP 4

NO GO TO STEP 2

STEP 2 Turn on the CA-Top Secret TRACE:

```
F TSS,SECTRACE(ACT,WTL)
TSS ADD(acid) TRACE
```

Retry request and examine the TRACE information. The TRACE will show the information that is being passed to CA-Top Secret along with DRCs and other valuable diagnostic information. Appendix B explains how to read the CA-Top Secret TRACE.

Did the TRACE expose the problem?

YES GO TO STEP 4

NO GO TO STEP 3

STEP 3 Obtain a dump. Either force a dump through your code using: **DC F'00'** or use the CA-Top Secret DIAGTRAP control option. Examine the parameter list that is being sent to CA-Top Secret.

Did the DUMP expose the problem?

YES GO TO STEP 4

NO GO TO STEP 5

STEP 4 Implement the solution to the problem and retest the customization code.

Was the problem resolved?

YES Stop

NO GO TO STEP 5

STEP 5 Gather all pertinent information. Go to Phase 3—contacting Technical Support.

3.4 Facility Access

This section details the procedures for investigating the following conditions:

- Access Incorrectly Allowed
- Signon Passwords Not Being Checked
- CPU Restrictions Not Honored
- Access Incorrectly Denied.

These procedures assume that the reader is familiar with the TSS LIST and TSS MODIFY commands and with the FACILITY, LOG, and MODE control options.

Remember that TSSSIM can be a valuable aid to determine the cause of access discrepancies.

3.4.1 Access Incorrectly Allowed

STEP 1 Determine if the user or profile has been explicitly granted access to the facility.

```
TSS LIST(acid) DATA(BASIC,PROF)
```

Does the facility in question appear within the user or profile ACID's Security Record?

YES TSS REMOVE the facility from the Security Record or remove the user from the profile.

GO TO STEP 8.

NO GO TO STEP 2.

STEP 2 Determine the security mode for the facility (or site) and for the user or profile ACID.

FOR	ENTER
Facility	TSS MODIFY(FAC(fac))
ACID(s)	TSS LIST(acid) DATA(XAUTH)

Is the user or profile ACID or the facility in either DORMANT or WARN MODE?

YES DORMANT MODE may allow access without security checking. WARN MODE may also allow access, but CA-Top Secret should be sending messages to the user's terminal. Consider using the TSS PERMIT command to move the user to a more restrictive security mode (IMPL or FAIL), or set WARNPW for the FACILITY control option. This will prohibit a user from accessing a facility unless he is explicitly authorized.

GO TO STEP 3 if WARN MODE; GO TO STEP 6 if DORMANT.

NO GO TO STEP 4.

STEP 3 List the logging options to determine if warning messages are being issued to the user. Ask the user to enter TSS WHOAMI at his terminal.

Is MSG displayed in the LOG field of the WHOAMI response?

YES Take administrative action. GO TO STEP 8.

NO The MSG option is not specified, therefore no messages are being sent to the user. Review LOG control options.

GO TO STEP 8.

STEP 4 Determine if the user possesses the NORESCHK attribute:

```
TSS LIST(acid)
```

Does the user possess the NORESCHK attribute?

YES TSS REMOVE the NORESCHK attribute from the ACID.

NO GO TO STEP 5

STEP 5 Determine if user is being allowed to bypass security by locating BYPASS message TSS9530I, on the STATUS response:

```
TSS MODIFY(STATUS)
```

Does the user's ACID, jobname or started task name appear in this list?

YES You must reset BYPASS, TSS BYPASS(RESET). GO TO STEP 8.

NO GO TO STEP 6.

STEP 6 Determine if the DRC control option is set to NOVIOL for the returned DRC code:

```
TSS MODIFY(DRC(drc#))
```

Is the DRC set to NOVIOL?

YES See the *Control Options Guide* to reset the DRC NOVIOL attribute.

NO GO TO STEP 7

STEP 7 Activate the diagnostic trace for the user. Refer to TSS ADD(acid)TRACE in the *Command Functions Guide* for instructions.

Did trace records not appear for facilities other than TSO, BATCH, or STC?

YES The site may not have a working interface with CA-Top Secret. Ask the site systems programmer to check/install the correct interface. Refer to the appropriate facility implementation guide for installation guidelines.

GO TO STEP 8.

NO Refer to Appendix B for instructions on how to interpret trace information.

GO TO STEP 8 when the examination of the trace is completed.

STEP 8 Implement the solution to the problem if this has not been done. If no solution has been determined, GO TO STEP 10.

STEP 9 Attempt to sign on to the facility in question using the ACID in question.

Was access denied?

YES Stop

NO GO TO STEP 10.

STEP 10 Gather all information obtained during previous steps, or gather any JCL procedures that are believed to be involved with the problem.

Go to Phase 3 for instructions on how to call Technical Support.

3.4.2 Signon Passwords not being Checked

STEP 1 Determine user's or profile's security mode:

```
TSS LIST(acid) DATA(XAUTH)
```

Is ACID in either DORMANT or WARN MODE?

YES DORMANT MODE may allow access without security checking. WARN MODE may also allow access, but CA-Top Secret should be sending messages to the user's terminal. Consider using the TSS PERMIT command to move the user to a more restrictive security mode (IMPL or FAIL). This will prohibit a user from accessing a facility unless he is explicitly authorized. Or assign the WARNPW attribute to the facility: TSS MODIFY((fac=WARNPW)). This will force defined users and jobs to enter their correct passwords while in WARN MODE.

If user is in WARN, GO TO STEP 3. If in DORMANT, GO TO STEP 7.

NO GO TO STEP 2.

STEP 2 Determine the facility's mode and attributes.

```
TSS MODIFY(FAC(fac))
```

Is facility in WARN MODE with the NOWARNPW attribute?

YES WARN MODE, in combination with NOWARNPW, will allow a user to bypass password security checking. Consider using the TSS PERMIT command to move the user to a more restrictive security mode, IMPL or FAIL. This will prohibit a user from accessing a facility unless he is explicitly authorized. Or assign the WARNPW attribute to the facility and TSS MODIFY(FAC(fac= WARNPW)). This will force defined users and jobs to enter their correct passwords while in WARN MODE.

GO TO STEP 7.

NO GO TO STEP 3.

STEP 3 Is facility in DORMANT MODE?

YES DORMANT MODE may allow access without security checking. Consider using the TSS PERMIT command to move the user to a more restrictive security mode (IMPL or FAIL). This will prohibit a user from accessing a facility unless he is explicitly authorized.

GO TO STEP 7.

NO GO TO STEP 5.

STEP 4 List the user's logging options to determine if warning messages are being logged. Ask the user to enter TSS WHOAMI at his terminal.

Is MSG displayed in the LOG field of the WHOAMI response?

YES Take administrative action. GO TO STEP 7.

NO LOG options are not specified, therefore no messages are being sent to the user. Review LOG control options.

GO TO STEP 7.

STEP 5 Determine if user is being allowed to bypass security by locating the BYPASS message TSS9530I, on the STATUS response:

```
TSS MODIFY(STATUS)
```

Does the user's ACID, jobname or started task name appear in this list?

YES TSS REMOVE the applicable bypass attribute from the user's ACID, jobname or started task name.

GO TO STEP 7.

NO GO TO STEP 6.

STEP 6 Activate the diagnostic trace for the user. Refer to TSS ADD(acid) TRACE in the *Command Functions Guide* for instructions.

Did trace records not appear for facilities other than TSO, BATCH, or STC?

YES The site may not have a working interface with CA-Top Secret. Ask the site systems programmer to check/install the correct interface. Refer to the appropriate facility implementation guide for installation guidelines.

GO TO STEP 7.

NO Refer to Appendix B for instructions on how to interpret trace information.

GO TO STEP 7 when the examination of the trace is completed.

STEP 7 Implement the solution to the problem if this has not been done. If no solution has been determined, GO TO STEP 9.

STEP 8 Attempt to sign on to the facility in question using the ACID and an invalid password.

Was access denied?

YES Stop

NO GO TO STEP 9.

STEP 9 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem.

Go to Phase 3 for instructions on how to call Technical Support.

3.4.3 CPU Restrictions not Honored

STEP 1 Determine if the CPU restriction exists for the user(s) and CPU in question.

```
TSS LIST(acid) DATA(BASIC,XA)
TSS LIST(ALL) DATA(XA)
```

Note: Examine rules for time of day, day of week, and program restrictions, along with any ACTIONS which may be in the PERMITs for the CPU.

Do permissions exist which allow the user to access the CPU?

YES TSS REVOKE the permissions, or TSS PERMIT the user access to the CPU with ACTION(FAIL,DENY).

GO TO STEP 9.

NO GO TO STEP 2.

STEP 2 Ensure that the CPU is owned.

```
TSS WHOOWNS CPU(cpu)
```

Is the CPU owned?

YES GO TO STEP 3.

NO Refer to TSS ADD - CPU in *Command Functions Guide.* If resource is not owned, CA-Top Secret cannot restrict access.

GO TO STEP 9.

Note: If the DEFPROT attribute is attached to a particular resource class, then the resource class is protected by default, even if it is not owned.

STEP 3 Determine the security mode for the facility (or site) and for the user or profile ACID(s).

FOR	ENTER
Facility	TSS MODIFY(FAC(fac))
ACID(s)	TSS LIST(acid) DATA(XA)

Is the ACID(s) or the facility in either DORMANT or WARN MODE?

YES DORMANT MODE may allow access without security checking. WARN MODE may also allow access, but CA-Top Secret should be sending messages to the user's terminal. Consider using the TSS PERMIT command to move the user to a more restrictive security mode (IMPL or FAIL). This will prohibit a user from accessing a facility unless he is explicitly authorized.

GO TO STEP 4 if WARN MODE; STEP 9 if DORMANT.

NO GO TO STEP 4.

STEP 4 List the user's logging options to determine if warning messages are being logged. Ask the user to enter TSS WHOAMI at his terminal.

Is MSG displayed in the LOG field of the WHOAMI response?

YES GO TO STEP 5.

NO LOG options are not specified, therefore no messages are being sent to the user. Review LOG control options.

GO TO STEP 9.

STEP 5 Determine if the ACID possesses the NORESCHK attribute:

```
TSS LIST(acid)
```

Does ACID possess the NORESCHK attribute?

YES TSS REMOVE(acid) NORESCHK

GO TO STEP 9

NO GO TO STEP 6

STEP 6 Determine if the DRC control option is set to NOVIOL for the returned DRC:

```
TSS MODIFY (DRC(drc#))
```

Is the DRC set to NOVIOL?

YES See the *Command Functions Guide* to reset the DRC.

GO TO STEP 9.

NO GO TO STEP 7.

STEP 7 Determine if user is being allowed to bypass security by locating the BYPASS message on the STATUS response:

```
TSS MODIFY(STATUS)
```

Does the user's ACID appear in this list?

YES TSS REMOVE the applicable bypass attribute from the user's ACID.

GO TO STEP 9.

NO GO TO STEP 8.

STEP 8 Activate the diagnostic trace for the user. Refer to TSS ADD(acid) TRACE in the *Command Functions Guide* for instructions.

Did trace records not appear for facilities other than TSO, BATCH, or STC?

YES The subsystem may not have a working interface with CA-Top Secret. Ask the site systems programmer to check/install the correct interface. Refer to the appropriate facility implementation guide for installation guidelines.

GO TO STEP 9.

NO Refer to Appendix B for instructions on how to interpret trace information.

GO TO STEP 9 when the examination of the trace is completed.

STEP 9 Implement the solution to the problem if this has not been done.

GO TO STEP 10.

Note: If no solution has been determined, GO TO STEP 11.

STEP 10 Attempt to sign on to the facility in question using the ACID in question.

Was access denied?

YES Stop

NO GO TO STEP 11.

STEP 11 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem.

Go to Phase 3 for instructions on how to call Technical Support.

3.4.4 Facility Access Incorrectly Denied

STEP 1 Is there a Detailed Violation Reason code (DRC) displayed with the violation?

YES Refer to "Detailed Reason Codes" in the *Messages and Codes Guide* to determine if the correct authorizations have been made to ensure access.

NO GO TO STEP 4.

STEP 2 Was the problem resolved during Step 1?

YES GO TO STEP 5

NO GO TO STEP 3

STEP 3 Determine the facilities the user is allowed to access through ADD or PERMIT authorizations.

```
TSS LIST(acid) DATA(BASIC)
```

```
TSS LIST(ALL) DATA(BASIC)
```

Do authorizations exist which permit the user to access the facility?

YES GO TO STEP 4.

NO TSS ADD the facility to the user's ACID.

GO TO STEP 5.

STEP 4 Activate the diagnostic trace for the user. Refer to TSS ADD(acid) TRACE in the *Command Functions Guide* for instructions.

Did NOTRACE appear for facilities other than TSO, BATCH, or STC?

YES Facility may not have a working interface with CA-Top Secret. Ask the site systems programmer to check/install the correct interface. Refer to the appropriate facility implementation guide for installation guidelines.

GO TO STEP 5.

NO Refer to Appendix B for instructions on how to diagnose the trace information.

GO TO STEP 5.

STEP 5 Implement the solution to the problem if this has not already been done.

GO TO STEP 6.

STEP 6 Attempt to sign on to the facility in question using the ACID in question.

Was access allowed?

YES Stop

NO GO TO STEP 7.

STEP 7 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem. Go to Phase 3 for instructions on how to call Technical Support.

3.5 Logging of Events

Whether or not security events are properly logged is a function of the LOG option which can be issued globally or by facility.

Although the following procedure provides basic checks which should be made to ensure that the LOG options are set correctly, the actual settings for the LOG option will vary from site to site, and from facility to facility. Refer to Chapter 15, Determine Violation Logging and Reporting in the *Planning Guide* and the LOG control option in the *Control Options Guide* to ensure the proper implementation and use of the LOG control option.

These procedures assume that the reader can use TSSUTIL,TSSTRACK, and TSSAUDIT to obtain logging reports. Refer to the *Report and Tracking Guide* if unfamiliar with any of these utilities.

3.5.1 CA-Top Secret Not Logging Violations

STEP 1 Enter:

```
TSS MODIFY(STATUS)
```

Is the site or facility in DORMANT MODE?

YES CA-Top Secret does not perform logging in DORMANT MODE. Upgrade mode of user or facility if logging is desired.

GO TO STEP 4.

NO GO TO STEP 2.

STEP 2 Examine the DRC control option setting.

```
TSS MODIFY(DRC(drc#))
```

Is DRC set to NOVIOL?

YES Reset the DRC NOVIOL setting. Enter:
TSS MODIFY(DRC(drc#,VIOL))

GO TO STEP 4.

NO GO TO STEP 3.

STEP 3 Determine if SMF suboption of LOG control option is specified, and examine the CA-Top Secret started task procedure to determine if the //AUDIT... statement is included in the procedure. Also, ensure that the Audit/Tracking File exists. Refer to *Installation Guide.*

Are SMF and/or //AUDIT specified?

YES GO TO STEP 5.

NO Enter:

```
TSS MODIFY(LOG(SMF...))
```

and/or refer to *Installation Guide* for instructions on how to add the //AUDIT statement to the CA-Top Secret started task procedure, and how to create the Audit/Tracking File.

GO TO STEP 4.

STEP 4 Simulate a violation to the facility in question using the ACID in question.

Was the violation logged?

YES STOP

NO GO TO STEP 5.

STEP 5 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem. Go to Phase 3 for instructions on how to call Technical Support.

3.5.2 CA-Top Secret not Logging Access or Initiations

STEP 1 Enter:

```
TSS MODIFY(STATUS)
```

or

```
TSS MODIFY(FAC(fac))
```

Is the site or facility in DORMANT MODE?

YES CA-Top Secret does not perform logging in DORMANT MODE. Upgrade the mode of the user or facility if logging is required.

GO TO STEP 3.

NO GO TO STEP 2.

STEP 2 Are ACCESS or INIT specified as suboptions of the LOG control option?

YES GO TO STEP 4.

NO Refer to the LOG or FACILITY control options in the *Control Options Guide* to enter ACCESS or INIT suboptions.

GO TO STEP 3.

STEP 3 Simulate the access and initiation attempt using the ACID in question.

Was the access or initiation logged?

YES STOP

NO GO TO STEP 4.

STEP 4 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem. Go to Phase 3 for instructions on how to call Technical Support.

3.6 Messages

The behavior of CA-Top Secret Messages is controlled by the MSG suboption of the FACILITY or LOG control option.

3.6.1 Users not Receiving CA-Top Secret Messages

STEP 1 Enter:

```
TSS MODIFY(STATUS)
```

or

```
TSS LIST(acid) DATA(XA)
```

Is the site, facility, or user in DORMANT MODE?

YES CA-Top Secret does not perform logging in DORMANT MODE. Consider upgrading the mode of the user or facility if logging is desired.

GO TO STEP 4.

NO GO TO STEP 2.

STEP 2 Determine if the MSG suboption of the LOG control option is specified by having the user enter TSS WHOAMI, or enter:

```
TSS MODIFY(FAC(fac))
```

Is MSG specified?

YES GO TO STEP 3.

NO Refer to LOG or FACILITY in the *Control Options Guide* and enter the MSG suboption.

GO TO STEP 4.

STEP 3 TSS may suppress messages due to entries made through the MSG control option. Refer to the *Control Options Guide* for instructions on how to check the characteristics of messages through the MSG control option.

Are messages being suppressed unnecessarily?

YES Make the appropriate entries for the MSG control option.

GO TO STEP 4.

NO GO TO STEP 5.

STEP 4 Perform access and initiation attempts using the ACID in question.

Were messages received?

YES STOP.

NO GO TO STEP 5.

STEP 5 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem. Go to Phase 3 for instructions on how to call Technical Support.

3.6.2 Console not Receiving CA-Top Secret Messages

STEP 1 Enter:

```
TSS MODIFY(STATUS)
```

Is the site or facility in DORMANT MODE?

YES CA-Top Secret does not perform logging in DORMANT MODE. Upgrade the security mode if logging is required.

GO TO STEP 4.

NO GO TO STEP 2.

STEP 2 Determine if the SEC9 suboption of the LOG control option is specified.

Is SEC9 specified?

YES GO TO STEP 3.

NO Refer to LOG or FACILITY in the *Control Options Guide* and enter the SEC9 suboption.

GO TO STEP 5.

STEP 3 CA-Top Secret may suppress certain messages due to entries made through the MSG control option (such as NOSEC9). Refer to the *Control Options Guide* for instructions on how to check the characteristics of messages through the MSG control option.

Are messages being suppressed unnecessarily?

YES Make the appropriate entries for the MSG control option.

GO TO STEP 4.

NO GO TO STEP 5.

STEP 4 Perform access and initiation attempts using the ACID in question.

Were messages received?

YES STOP.

NO GO TO STEP 5.

STEP 5 Gather all information obtained during previous steps, or gather any procedures, JCL, or STC procedures that are believed to be involved with the problem. Go to Phase 3 for instructions on how to call Technical Support.

3.7 Resource Access Problems

This procedure provides instructions on how to diagnose access authorization problems for:

- Data Sets
- Volumes
- Other ownable resources

CA-Top Secret uses an access algorithm to determine whether or not to grant a user the access that they requested to the resource. This algorithm is detailed in the *User Guide.* A thorough understanding of the algorithm, and the control options and commands which affect the algorithm, are prerequisites to the effective use of the following procedures and guidelines.

Use of the TSS LIST command function, and the TSSSIM utility are also prerequisite skills. If you are unfamiliar with any of the aforementioned items, refer to the appropriate guide for details. Refer to Appendix A, *Diagnostic Tools,* for references.

3.7.1 Data Set or Volume Access Incorrectly Denied

STEP 1 Determine the user's and the facility's mode.

To determine the user's mode, enter:

```
TSS LIST(acid) DATA(XA)
```

To determine the facility's mode, enter:

```
TSS MODIFY(FAC(fac))
```

Is user in a more restrictive mode than the facility?

Example: Facility is in WARN MODE, but user is in FAIL.

YES Permit the user to a less restrictive mode or, PERMIT the user explicit access to the resource.

GO TO STEP 7.

NO GO TO STEP 2.

STEP 2 Determine if default data set protection is in effect for IMPLement or WARN MODEs.

Enter:

```
TSS MODIFY(STATUS)
```

Is the DEFPROT attribute attached to the data set?

YES PERMIT the user explicit access to the resource.

GO TO STEP 7.

NO GO TO STEP 3.

STEP 3 Access TSSSIM to simulate a data set access attempt by entering TSSSIM parameters as shown below:

For	Enter
ACID	User's ACID
FACILITY	Facility user had access to when the problem occurred
MODE	The facility's mode
TRACE	YES
ACCESS	The access which the user requested when access was denied.

Note: Refer to Chapter 5, *Using the TSSSIM Utility*, in this guide.

STEP 4 Press Enter.

Was the requested access granted or denied?

DENIED GO TO STEP 5.

GRANTED This indicates a probable user error, or system error. Take administrative action.

STEP 5 TSS LIST the user's profiles and user Security Record.

```
TSS LIST(acid) DATA(ALL,PROFILE)
```

Note: Be sure to examine ownership and XAUTH fields, and the security records of any connected profiles. Also, be aware that data set access problems may be caused by volume access rules. CREATE access for data sets may, for example, be overridden by a volume authorization which only allows UPDATE access to data sets on the volume.

Does the display show authorizations at the requested access level and to the resource in question?

YES GO TO STEP 6.

NO PERMIT the user access to the resource through the user record, ALL Record, or through connection to a profile which has the appropriate access.

GO TO STEP 7.

STEP 6 Check the settings of the AUTH control option:

```
TSS MODIFY(STATUS)
```

Has AUTH been changed after the Security File was originally established?

YES Attempt to reconstruct Security File to match the required access definitions.

GO TO STEP 7.

NO GO TO PHASE 3 for instructions on how to call Technical Support.

STEP 7 Refer to Step 3 for instructions on how to use TSSSIM to perform a simulated data set access attempt.

Was access denied?

GRANTED Stop.

DENIED Obtain all information gathered during the previous steps. GO TO PHASE 3 for instructions on how to call Technical Support.

3.7.2 Data Set Passwords not being Checked

STEP 1 Determine if the data set has the ACTION(PASSWORD) attribute.

```
TSS WHOHAS DSNAME(data set name or prefix)
```

Is the ACTION(PASSWORD) attached?

YES GO TO STEP 2

NO To password protect the data set you must PERMIT the data set attaching the ACTION(PASSWORD) attribute. If the attribute is not attached, password protection is overridden by CA-Top Secret. See the *Command Functions Guide.*

GO TO STEP 7

STEP 2 Determine if the ACID has the NODSNCHK attribute.

```
TSS LIST(acid) DATA(BASIC)
```

Does the ACID have the NODSNCHK attribute?

YES Remove the NODSNCHK authority.

```
TSS REMOVE(acid) NODSNCHK
```

GO TO STEP 7

NO GO TO STEP 3

STEP 3 Determine if the user has ACTION(NODSN) authority to the volume.

```
TSS LIST(acid) DATA(XAUTH)
```

Is the volume authorized with ACTION(NODSN)?

YES CA-Top Secret will not perform data set level checking.

```
TSS REVOKE(acid) VOLUME(volser or prefix)
    ACTION(NODSN)
```

GO TO STEP 7

NO GO TO STEP 4

STEP 4 Turn on the TSS TRACE for the ACID in question:

```
TSS ADD(acid) TRACE
```

GO TO STEP 5

STEP 5 Activate the TRACE from the console:

```
TSS MODIFY(SECTRACE(ACT,WTL))
```

GO TO STEP 6

STEP 6 Ask the user to access the data set in question. Examine the trace information that is generated.

Is the 04 DRC being returned?

YES The data set is authorized with ACTION(PASSWORD).

GO TO STEP 8.

NO An overriding authorization was found in the user acid, connected profiles, or ALL Record. Determine the source of the overriding authorization and REVOKE it from the ACID.

GO TO STEP 7.

STEP 7 Implement the correction described in the previous step. Ask the user to attempt access to the data set.

Was the ACID prompted for the data set password?

YES The problem is resolved.

NO GO TO STEP 8

STEP 8 Gather all previously collected information. Proceed to Phase 3.

Chapter 4. Phase 3—CA Technical Support

This chapter lists the information that you must gather before you call Technical Support. It also describes how the support center is organized and the type of help you can expect to receive. Note: the following describes the general guidelines for contacting Computer Associates technical support. If CA Top-Secret was purchased from IBM and installed from the IBM External products tape, all technical support should be obtained directly from IBM support. Please refer to your IBM provided guidelines for support and contacts on your IBM purchased Security product. This chapter can be used as general guidelines for some of the questions and date your IBM service center will ask you for.

4.1 Pre-Call Preparation

Before contacting Technical Support, attempt to use the Phase 2 diagnostic procedure. If you cannot solve your problem by using the applicable Phase 2 diagnostic procedure, you must prepare the necessary information prior to placing your call. What follows is a list of the information which the CA-Top Secret Technical Support technician may want to discuss with you.

4.1.1 Required Information

Prior to calling CA-Top Secret Technical Support, obtain the following:

1. All information obtained from the Phase 2 diagnostic process, including violation codes, message numbers, TRACE results, operator console log from the time of the error, TSS LIST results, TSSSIM results, and everything else you know about the problem.
2. CA-Top Secret's current maintenance level as obtained through the VERSION control option (refer to the *Control Options Guide* for details).
3. Maintenance levels of the software environment, specifically the VSE release and PUT levels, This site-dependent information is crucial.
4. Any information regarding special exits or customization strategies.
5. Any information about when the specific problem started and what changes occurred in your system.

4.1.2 Enhancing Communication With Technical Support

The information listed above allows you to be as specific as possible about the problem. Symptoms such as "it doesn't work", or "getting some message", or "some weird abend", only lead to delays in solving the problem. Please have all required information available. If a dump is produced, please have a system programmer present to give dump details. Dumps sent to Technical Support must be unformatted, and on tape. See the 4.3.6, "Mailing Diagnostic Information", for more details.

4.2 Contacting Technical Support

Computer Associates provides 24-hour support, 365 days a year. If you need technical assistance with your CA product, there are several ways to obtain it:

1. During normal business hours, call the support number for the product with which you are having a problem, and you will be connected with a support representative. If a technician is not available, your call will be logged and a technician will return your call as soon as possible.

   If you are calling with a severity one problem and do not get a technician immediately, the receptionist will ask you if you would like to stay on hold until a technician is available. The support number for CA-Top Secret for VSE is **908-874-9605.** (The individual support numbers are listed in the *Product Support Directory* located at http://www.cai.com.)

2. A toll free number **1-800-645-3042** is also available to you for support calls. Calls logged there will be returned in a timely fashion. This number is also to be used to request after-hours emergency support for all products. If you are calling from outside of North America or if the 800 number is not accessible to you, please contact your local Computer Associates Support Center.

3. If you wish to contact the center by FAX, you may do so using the following number: **908-874-9178.**

   Note: Only your local Computer Associates Support Center can provide native language assistance. Please use English when contacting any North American center.

4. CA-Total Client Care clients can dispatch problem-related questions and receive their answers electronically.

There are two kinds of technical support available: primary and emergency. Primary service is provided for all CA products during normal business hours. **Emergency service is available after primary service hours for severity one problems only.**

When contacting Technical Support, have the following information available:

- your site ID (a six digit ID which uniquely identifies your site). This number can be found on the labels of most mailings received from CA, or may be obtained from your account manager.
- the product name, release number, operating system and genlevel
- your name, telephone number, and extension (if any)
- your company name

- the *severity code*. This is a number from one to four that you assign to the problem. Use the following guidelines when determining the severity of the problem:

  1. a "system down" or inoperative condition
  2. a suspected high-impact condition associated with the product
  3. a question concerning product performance or an intermittent low-impact condition associated with a product
  4. a question concerning general product utilization or implementation

- any documentation that may help in resolving the problem, including dumps, compiler listings, and so on.

Refer to the *Product Support Directory* for the individual primary service support numbers for each of your CA products. For severity one calls during emergency service hours, you should always call **1-800-645-3042** so that a technician on call can be paged to return your call.

Note: Requests for services such as: orders for documentation, maintenance tapes, requests for products, information about education, on-site assistance, or requests for new features or design changes to CA-Top Secret may be directed to your Regional Account Representative.

4.3 Generating a Problem Report

Once a Computer Associates Technical Services representative has determined that your problem requires further investigation, there are two tools you can use to generate a problem report. These are: the CA-Activator problem reporting facility and the CAISERV utility. Either of these generate the problem report required by Computer Associates.

4.3.1 CA-Activator Problem Reporting Facility

The CA-Activator problem reporting facility allows you to provide a description of the problem. It automatically produces a report on

- your current system environment
- CA product installation and maintenance information
- the product runtime options in effect

You can produce a hard copy of this report or upload it with CA-TCC.

Note: You must have a contact number assigned to you by Computer Associates Technical Services before using CA-TCC to upload your problem report.

4.3.2 CAISERV Utility

The CAISERV diagnostic facility produces a problem report for you to fill out and send in with all problem documentation.

CAISERV also produces a short report on the Computer Associates VSE products that you have installed. This information should also be sent to aid Technical Support in solving your problem.

To invoke CAISERV, execute the CAISERV procedure in your sample JCL library:

```
// JOB CAISERV
// EXEC CAISERV
/*
/&
```

Edit the JCL to your installation's standards and submit the job.

The messages you may encounter when running CAISERV are:

CAPP999E INSUFFICIENT STORAGE TO PROCESS CAISERV

Reason: Sufficient storage was not allocated to execute CAISERV.

Action: Use at least 100k of storage for executing CAISERV.

****PRODUCT CAISERV MODULE 'modulename' NOT ACCESSIBLE *****

Reason: Libraries are not properly concatenated.

Action: Review and modify the JCL. Execute CAISERV.

4.3.3 Problem Escalation

If you wish to change the severity code of a reported problem, contact the support center working on your problem and request the change.

If you feel that your problem is not being adequately addressed by CA, you can escalate your concerns by contacting the following:

- Level one or level two support manager
- Product owner
- Senior Vice President of Research and Development
- Corporate Help Desk in Islandia

4.3.4 Level One Support

The level one support team handles problems as follows:

- If the problem appears to be caused by CA-Top Secret internals, the technician determines if corrective maintenance tapes have been distributed. If the maintenance tapes have been distributed, the technician asks the customer to apply the maintenance. If no fix exists, the problem is escalated to the level two support team for further examination.
- If the problem appears to be caused by customer error, then the technician attempts to help the customer diagnose and resolve the error. The technician asks many of the same questions which were asked by the Phase 2 procedures. If no resolution can be reached, then the problem is placed in the Problem Control System for review by a level two technician.
- Before escalating a problem to level two support, the level one technician ensures that the customer has prepared information as documented in this guide.
- If a client calls asking for the status of a case, any technician can offer assistance provided the client has a contact number. The customer should specify that he or she merely wants the status of a particular case.

4.3.5 Level Two Support

The level two support team is responsible to locate the internal errors and provide a fix. These fixes are later incorporated into the standard maintenance tapes. If the level two technician is unable to locate the internal error, or if the error requires major modifications, the request is referred to management for review by development.

4.3.6 Mailing Diagnostic Information

Diagnostic information, such as dump tapes, trace records, system logs, and so on, may be mailed to the following address:

Computer Associates International, Inc.
One Computer Associates Plaza
Islandia, NY 11788-7000
ATTN: PCS Desk Contact # _____

Note: Your CA-Top Secret case number and representative must be clearly marked on the package.

SVC dumps must be sent in the form of SYSDUMPS on the 6250-bpi magnetic tape or cartridge. SVC dumps **must not be edited in any way, and do not** use the INFOANAL utility program to format dumps. Do **not** send SYSDUMPS when an SVC dump is available; under **no circumstances** should an ABEND-AID or DUMP-MASTER dump be sent.

Please mark the following information on the tape's external label:

- Tape label information
- Contact number associated with diagnostic materials
- Data set name(s)
- Dump Type/Format
- Release of operating system on which material was produced (for example, VSE/ESA 2.1), as well as the maintenance level of the operating system (for example, 9806).
- Return address if the tape needs to be returned to your data center.

Materials that are not in the proper form delay problem resolution.

Chapter 5. Using the TSSSIM Utility

The Security Simulation Utility (TSSSIM) gives the security administrators the ability to test permissions on the Security File without affecting the "real" production environment. When the administrator invokes the security simulator and performs a simulated LOGON/SIGNON for a specified user, he can test the Security File permissions (even simulating FAIL mode processing while the system may still be in DORMANT mode).

Testing of the security permissions consists of invoking a simulation resource command for the desired resource. TSSSIM will report whether the currently active simulated ACID has access to the resource under the conditions specified by the administrator. Resource qualifying conditions that may be simulated include SVC-in-control, access level, and privileged program.

In addition, the simulator can also be used as a diagnostic tool when the administrator needs to "debug" errors in the Security File permissions. Often, when a user has several profiles attached to his ACID, it may become difficult to isolate which permission has allowed or denied access to a particular resource. By interpreting trace information generated by the security algorithm, TSSSIM can isolate the exact permission or ownership as well as indicate which record (user, profile, or all) contained the permission.

TSSSIM can only be executed under the BATCH facility. See section 5.8, where TSSSIM under BATCH is discussed.

Error messages and abend codes for TSSSIM can be found in the *Messages and Codes Guide*.

5.1 Authority and Scope

TSSSIM is only available for use by administrators having the following authority:

```
TSS ADMIN(acidname) MISC1(TSSSIM)
```

A simulated signon can be initiated for any ACID in the Security File, including those not within the administrator's scope. However, the resource checks that the administrator may issue depend on the following:

- If the ACID lies within the administrator's scope, he may issue resource checks against any resource in the Security File.
- If the ACID does not lie within the administrator's scope, he may issue resource checks against only those resources that lie within his scope.

5.2 TSSSIM Commands in General

TSSSIM is basically a command-driven facility. Once TSSSIM is invoked, simply enter the appropriate commands and any qualifying parameters, if applicable. TSSSIM will simulate a logon for a specific user, perform resource checking, as well as act as a diagnostic tool.

TSSSIM commands can be divided into three discrete types: system environment, resource, and special.

For your convenience, all commands are listed according to type and are accompanied by a brief description. The commands then appear individually with more detailed information.

System Environment Commands:

LOGON/SIGNON	Initiates a simulated ACID session.
LOGOFF/SIGNOFF	Terminates a current simulated session.
HELP	Generates a help listing.
ENVIRON	Changes the simulated ACID environment.
END/QUIT	Exits the Security Simulator.

Simulated Resource Commands:

All simulated resource commands are prefixed by a dollar sign ($) with the exception of user-defined resources, which are prefixed by an at sign (@).

$ABS	Checks Abstracts.
$ALT-ACID	Checks acid cross-authorization.
$APPCLU	Checks logical units.
$APPCPORT	Checks the VTAM logical unit name.
$APPCSI	Checks access authorization to APPC side information files.
$APPCTP	Checks use of transaction programs and profile.
$APPL	Checks IMS Application Group Names.
$AREA	Checks IDMS Areas.
$CAADMIN	Checks CA Administrative functions.
$CACCFDSN	Checks CA-Librarian data sets.
$CACCFMEM	Checks CA-Librarian CCF members.
$CACMD	Checks CA Commands.

$CALIBMEM	Checks CA-Librarian members.
$CAPCLU	Checks for CA-Top Secret/PC ports.
$CAREPORT	Checks CA-Dispatch reports.
$CATAPE	Checks CA-1 Resources.
$CAVAPPL	Checks CA VTAM applications.
$CIMS	Checks security for IMS commands.
$CPCMD	Checks VM CP commands.
$CPU	Checks CPU resources.
$DASDVOL	Checks DASD volume accesses.
$DATABAS	Checks ABABAS databases.
$DBD	Checks IMS Database Descriptors.
$DB2	Checks DB2 databases.
$DB2BUFFP	Checks DB2 buffer pools.
$DB2COLL	Checks DB2 collections.
$DB2DBASE	Checks DB2 databases.
$DB2PKG	Checks DB2 packages.
DB2PLAN	Checks DB2 plans.
$DB2STOGP	Checks DB2 storage groups.
$DB2SYS	Checks DB2 system privileges and authorities.
$DB2TABLE	Checks DB2 tables.
$DB2TABSP	Checks DB2 table spaces.
$DCSS	Checks VM Discontiguous Saved Statements (DCSS).
$DCT	Checks CICS Destinations.
$DEVICES	Checks MVS Allocation of UR, graphic, or teleprocessing devices.
$DIAG	Checks VM Diagnose instructions.
$DLFCLASS	Checks DLF data sets.
$DSN	Checks MVS data sets.
$FCT	Checks CICS Files.
$FIELD	Checks user-defined resources.
$IBMFAC	Checks IBM facility resources.
$IBMGROUP	Checks for IBM Group resources.
$IUCV	Checks IUCV Communications targets.

$JCT	Checks CICS Journal Control Tables.
$JESINPUT	Checks JES job sources.
$JESJOB	Checks JES job names.
$JESSPOOL	Checks JES Spool Data Sets (SYSIN and SYSOUT).
$JOBNAME	Checks MVS JES2/JES3 job names.
$LCF	Checks LCF commands and transactions.
$MDISK	Checks VM minidisks (for the VMMDISK resource class).
$MGMTCLAS	Checks SMS management classes.
$OPCMD	Checks system operator commands.
$OPERCMD	Checks both JES and MVS operator commands.
$OTRAN	Checks OTRAN (ownable transactions).
$PANEL	Checks panel authorizations for CA products.
$PGM	Checks MVS programs.
$PPT	Checks CICS programs.
$PROPCNTL	Checks automatic propagation on batch jobs.
$PSB	Checks IMS PSBs.
$PSFMPL	Checks PSF security functions.
$RECIPID	Checks CA-Dispatch recipient IDs.
$SCHEDULE	Checks CA-Scheduler schedules.
$SDSF	Checks the SDSF interface.
$SMESSAGE	Checks TSO message sending.
$SPI	Checks CICS system functions.
$STATION	Checks CA-Scheduler stations.
$STORCLAS	Checks SMS storage classes
$SUBMIT	Checks ACIDs job submissions.
$SUBSCH	Checks IDMS subschemas.
$SYSCONS	Checks System consoles.
$TAPEVOL	Checks tape volumes.
$TERM	Checks network terminal IDs.
$TOTAL	Checks ENVIRON/1 totals.
$TSAF	Checks VM TSAF connections.
$TSOACCT	Checks TSO account codes.

$TSOAUTH	Checks TSO user attributes.
$TSOPRFG	Checks TSO performance groups.
$TSOPROC	Checks PROCs used for TSO logon.
$TST	Checks temporary CICS storage names.
$UR1	Checks for general UR1 resource names.
$UR2	Checks for general UR2 resource names.
$USER	Checks for user non-ownable resources.
$USRCLASS	Checks SDSF USRCLASSES.
$VMANAPPL	Checks CA-VMAN applications.
$VMCF	Checks VMCF communication targets.
$VMDIAL	Checks virtual machines with DIAL access security.
$VMMACH	Checks virtual machine logons.
$VMNODE	Checks VM nodes.
$VMRDR	Checks virtual machine readers.
$VSELIB	Checks access to VSE library.
$VSESLIB	Check access to VSE Sublibrary.
$VSEMEMBR	Checks access to member in VSE Sublibrary.
$VSEUSER	Checks access to VSE User resource.
$VSEPART	Check access to specific VSE partition resource.
$VTAMAPPL	Checks ACB open authority from non-APF programs.
$VXDEVICE	Checks VAX devices.
$VXFILE	Checks VAX files.
$WRITER	Checks processing of output to specific devices.
@cccccccc	Checks user-defined resources added to the RDT Record.

Special Commands:

EJECT	Generates a page eject for a batch listing.
STATUS	Displays current simulation status.
TSS	Invokes the TSS command for administrators.

5.3 System Environment Commands

The system environment commands allow administrators to log on and off of TSSSIM, issue help requests for any command, and exit the simulator. Each command is discussed on the following pages and includes this information: function, the security macro invoked, any parameters that may be used to qualify the command, an example, and any additional comments.

5.3.1 LOGON/SIGNON

Function Initiates a simulated ACID session.

Security Macro RACROUTE REQUEST=INIT

Parameters ACID, CPU, FACILITY, MODE, PRIVPGM, SVC, TERMINAL, TRACE

Example After invoking the Security Simulation Facility by entering TSSSIM, an administrator logs on with the simulated ACID in a TSO facility. He enters the following:

```
LOGON ACID(acid) FAC(TSO)
```

or

```
SIGNON ACID(acid) FAC(TSO)
```

TSSSIM will return a message telling the administrator that a successful simulated session has been established. If there are any restrictions, the simulator will return a message indicating that an ACID is suspended, a facility is currently deactivated, and so on.

Comments The SIGNON or LOGON command has the same effect, and can be used interchangeably.

5.3.2 LOGOFF/SIGNOFF

Function Terminates a current simulated session.

Parameters None

Example When ready to end a simulated session, the administrator enters one of the following commands:

```
LOGOFF
```

or

```
SIGNOFF
```

Comments The SIGNOFF or LOGOFF command has the same effect and can be used interchangeably.

The administrator must be logged on to execute the LOGOFF/SIGNOFF command.

5.3.3 HELP

Function Without any accompanying TSSSIM command or asterisk, it will generate a listing of all commands and a brief description of each.

When accompanied by a command, it will give a brief description of the command as well as its command class, attributes, and qualifying parameters.

Parameters Any valid TSSSIM command names.

Example When an administrator wishes to acquire a listing of all TSSSIM commands, she enters:

```
HELP
```

If an administrator wants to know more about the $DSN command, she enters:

```
HELP $DSN
```

She then receives the following information concerning $DSN:

```
COMMAND NAME = $DSN
DESCRIPTION  = RES CHECK - OS DATASETS
COMMAND CLASS= DATASETX (C4)
ATTRIBUTES   = LOGON-REQUIRED,NEWRB
PARAMETERS   = ACCESS,NEWDSN
PARAMETERS   = PRIVPGM,SVC,TRACE,VOLUME,XACCESS
```

The NEWRB indicates that if PRIVPGM is specified, TSSSIM will generate another RB to simulate the one in the CA-Top Secret PRIVPGM.

COMMAND CLASS indicates the security class in which this resource resides.

If an administrator enters:

```
HELP *
```

She will receive the same information as illustrated above, but for every command.

5.3.4 ENVIRON

Function Changes the simulated ACID environment.

Parameters CPU, MODE, OWN, PRIVPGM, SVC, TERMINAL, TRACE

Example An administrator originally logged on with a simulated ACID in FAIL mode.

He now wishes to change the environment of the simulated ACID to WARN mode and put a trace on all resource checks. He enters the following:

```
ENVIRON MODE(WARN) TRACE
```

5.3.5 END/QUIT

Function Exits the Security Simulator Facility.

Parameters None.

Example When an administrator wishes to exit the simulator, he enters:

```
END
```

or

```
QUIT
```

Comments You may enter either of these commands at any time during a simulated session.

If you enter QUIT or END before entering the LOGOFF or SIGNOFF command, it will terminate the session as well as exit the simulator.

5.4 Simulated Resource Commands

The simulated resource commands allow the administrator to choose the type (or class) of resource he wishes to check. Resource checks for the specific resource are passed to CA-Top Secret on behalf of the simulated ACID. Summary information regarding whether the resource access was allowed or denied is displayed and includes the following:

- Return codes and reason codes from the respective security routine.
- Violation messages (which would be received by the user if the user was in IMPL or FAIL mode).
- A status message indicating if access would have been actually allowed or denied, based on the simulated mode.

Basically, all of the specific commands provide a similar function in that they pass a resource-check command to the simulator which, in turn, issues a security request to CA-Top Secret. The only real difference between the individual commands is the different parameters that are allowed. For example, certain resources allow access-level checks, while a few do not.

Note: Only one resource check can be performed for each command. Generic prefixing is not honored.

Each command is discussed in detail, and includes the following information: function, the security module invoked, any parameters that may be used to qualify the command, an example, and any additional comments.

Note: To perform a resource check for a facility different from the one you are signed on to, log off and then log on to the new facility using the LOGON command (explained earlier in this chapter).

The following is a brief description of the resource qualifying parameters. The discussion of individual resource commands will indicate which parameters are valid.

ACCESS Specifies the name of the access level for the resource check.

OWN Valid for resources that use the RACROUTE REQUEST=FASTAUTH security module; the default is NOOWN. If OWN is specified, it indicates that CA-Top Secret should assume the resource is owned, and not check its Global Resource Table to see if the resource entity (or prefix) is defined.

Whether OWN is specified or not, if the resource is not owned, the access will be denied.

PRIVPGM Specifies the program in control when the resource check is to be issued.

SVC Specifies the SVC in control when the resource check is to be issued. The following list of SVC names are allowed by TSSSIM:

ALLOCATE
CATALOG
CREATE
FEOV
OPEN
RENAME
SCRATCH

The default is OPEN.

TRACE If TRACE is specified, it indicates that the trace feature is to be enabled so that the exact permission causing the resource access or denial can be located. The default is NOTRACE.

XACCESS Specifies a two-byte hexadecimal code equating to specific access level(s). If specified, this overrides the default access value used in conjunction with the specified SVC name. This is especially convenient if you need to check a combination of access levels. For example, in an SPF environment, you can not enter more than one access level with the ACCESS keyword. For more detailed information, see the examples for the $DBD and $TST commands in this section.

5.4.1 $ABS (ABSTRACT)

Function Issues a security check for abstract resources.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters PRIVPGM, OWN, TRACE

Example An administrator wishes to issue a security check to see if the current simulated ACID can access account numbers beginning with the prefixes RCS and TCS. He enters:

```
$ABS(RCS,TCS)
```

Comments $ABS is also used in issuing a security check for TMS security bypass control XTD98000. This resource class is also used, with additional site-written code, to extend security to other "system" type resources, for example, when defining job execution classes to CA-Top Secret.

5.4.2 $APPCLU (APPC Logical Units)

Function Issues a security check to identify which logical units (LUs) can establish a link for the purpose of processing APPC transactions and conversations.

Security Macro RACROUTE REQUEST=LIST

Parameters TRACE, OWN, ACCESS, XACCESS

Example An administrator wishes to issue a security check to see which logical units are allowed to process APPC transactions.

```
$APPCLU(LU1,LU2)
```

5.4.3 $APPCPORT (APPC Ports)

Function Issues a security check to identify the VTAM logical unit name of the logical unit from which an APPC conversation request must originate. The APPL resource class is used to indicate which logical unit a conversation request can be made.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCES

Example An administrator wishes to issue a security check of the current simulated ACID for a port name of LU01. He enters:

```
$APPCPORT(LU01)
```

5.4.4 $APPCSI (APPC Side Information)

Function Issues a security check to identify who can administer the APPC side information files.

Security Macro RACROUTE REQUEST=AUTH

Parameters TRACE, OWN, ACCESS, XACCESS

Example An administrator wishes to maintain all side information files assigned to the RESEARCH database token. He enters:

```
$APPCSI(RESEARCH.SYS1)
```

5.4.5 $APPCTP (APPC Transaction Programs and Profiles)

Function Issues a security check on the use of the APPC transaction programs and the transaction profiles.

Security Macro RACROUTE REQUEST=AUTH

Parameters TRACE, OWN, ACCESS, XACCESS

Example An administrator is responsible for maintaining TPB (which has a database token of PAYROLL and will be accessible to all users for LU02). He enters:

```
$APPCTP(PAYROLL.SYS1.TPB)
```

5.4.6 $APPL (IMS Application)

Function Issues a security check for an IMS Application Group Name (AGN).

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator needs to perform a security check on an IMS Application named TEMPAY and activate the TRACE facility to locate the exact permission.

```
$APPL(TEMPAY) TRACE
```

5.4.7 $AREA (IDMS Database Areas)

Function Issues a security check for an IDMS data base Area.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, TRACE, PRIVPGM, XACCESS

Example An administrator wants a security check on an IDMS Data Base Area (AFRPER3) with an access of UPDATE. He also specifies the OWN parameter so that CA-Top Secret will assume the resource is owned. He enters:

```
$AREA(AFRPER3) ACCESS(UPDATE) OWN
```

Comments Resource name lengths of up to 44 characters are supported with the $AREA command.

5.4.8 $CAADMIN (CA Administrative Functions)

Function Issues a security check for CA administrative functions

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wishes to issue a security check to determine who is authorized to perform the CA-ASM2 $CI administrative function. He enters:

```
$CAADMIN($CI)
```

5.4.9 $CACCFDSN (CA-Librarian/CCF Data Sets)

Function Issues a security check for CA-Librarian/CCF data sets.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check for the "LIB.TEST.DATA" data set. He enters:

```
$CACCFDSN(LIB.TEST.DATA)
```

5.4.10 $CACCFMEM (CA-Librarian/CCF Members)

Function Issues a security check for CA-Librarian/CCF members.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check of the CCFMEM02 module. He enters:

```
$CACCFMEM(CCFMEM02)
```

5.4.11 $CACMD (CA Command Authorization)

Function Issues a security check for CA command authorization.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check of the CA-1 LOEXTEND command. He enters:

```
$CACMD(LOEXTEND)
```

5.4.12 $CALIBMEM (CA-Librarian Member)

Function Issues a security check for CA-Librarian members.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check for the LIBMEM01 member. He enters:

```
$CALIBMEM (CA-Librarian Member)
```

5.4.13 $CAPCLU (CA-Top Secret/PC Ports)

Function Issues a security check for LUs assigned as CA-Top Secret/PC ports.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to determine whether LU10 is a secured CA-Top Secret/PC port. He enters:

```
$CAPCLU(LU10)
```

5.4.14 $CAREPORT (CA-Dispatch reports)

Function Issues a security check for CA-Dispatch reports.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to determine who can run the PYRLLUPD report under CA-Dispatch. He enters:

```
$CAREPORT(PYRLLUPD)
```

5.4.15 $CATAPE (CA-1 Resource)

Function Issues a security check for CA-1 members.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check for the CA-1 BLPNORES resource. He enters:

```
$CATAPE(BLPNORES)
```

5.4.16 $CAVAPPL (CA VTAM Applications)

Function Issues a security check for CA products using VTAM ACBs.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check to determine who has access to the ACB123 VTAM ACB. He enters:

```
$CAVAPPL(ACB123)
```

5.4.17 $CIMS (PC Port Logical Unit(LU) Names)

Function Issues a security check for IMS 4.1 commands.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wishes to issue a security check to determine if a user can issue the IMS/START command. He enters:

```
$CIMS(STA)
```

5.4.18 $CPU (Central Processing Unit)

Function Issues a security check for a CPU resource.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, PRIVPGM, LIBR

Example An administrator wants a security check on a CPU (SYSA) with an access of UPDATE. He also specifies the TRACE parameter to locate the permission for the access. He enters:

```
$CPU(SYSA) TRACE
```

5.4.19 $DASDVOL (DASD Volume Level)

Function Issues a security check for a DASD volume. This is a volume-only check. No data set level check is performed.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wishes to issue a security check on a DASD volume with ALL access. She enters:

```
$DASDVOL(24T921) ACCESS(ALL)
```

5.4.20 $DATABAS (ADABAS Database)

Function Issues a security check for an ADABAS database.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, ACCESS, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wishes to issue a security check on an ADABAS database with UPDATE access. He enters:

```
$DATABAS(EMPDATA) ACCESS(UPDATE)
```

5.4.21 $DBD (IMS Database Descriptor)

Function Issues a security check for an IMS Database Descriptor name.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wants a security check on an IMS DBD (TSTPDA) and issues an XACCESS of 88 which simulates access levels of UPDATE and DELETE. He enters:

```
$DBD(TSTPDA) XACCESS(88)
```

Alternately, he could have specified:

```
$DBD(TSTPDA) ACCESS(UPDATE,DELETE)
```

5.4.22 $DB2 (DB2 Access)

Function Issues a security check for a DB2 database.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check on a DB2 subsystem PRD1 from CICS. He enters:

```
$DB2(DSNR.PRD1.SAS)
```

Comments The $DB2 command supports resource name lengths of up to 44 characters.

5.4.23 $DB2BUFFP (DB2 Buffer Pools)

Function Issues a security check for DB2 buffer pools.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wants a security check on buffer pool BP1. He enters:

```
$DB2BUFFP(BP1)
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.24 $DB2COLL (DB2 Collections)

Function Issues a security check for DB2 collections.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wants a security check on the collection PAY04 with PACKADM access. He enters:

```
$DB2COLL(PAY04) ACCESS(PACKADM)
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.25 $DB2DBASE (DB2 Databases)

Function Issues a security check for DB2 databases.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wishes to issue a security check on the database ACCTING with DROP access. He enters:

```
$DB2DBASE(ACCTING) ACCESS(DROP)
```

Comments DB2DBASE has been designed for use with the CA-Top Secret for DB2 interface.

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.26 $DB2PKG (DB2 Packages)

Function Issues a security check for DB2 packages.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wishes to issue a security check on the package PAY04.PQUERY with COPY access. He enters:

```
$DB2PKG(PAY04.PQUERY) ACCESS(COPY)
```

Comments DB2PKG has been designed for use with the CA-Top Secret for DB2 interface.

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.27 $DB2PLAN (DB2 Plans)

Function Issues a security check for DB2 plans.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wants a security check on a DB2 plan named SR19052P with EXECUTE access. He enters:

```
$DB2PLAN(SR19052P) ACCESS(EXECUTE)
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.28 $DB2STOGP (DB2 Storage Groups)

Function Issues a security check for DB2 storage groups.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wants a security check on the TESTDASD storage group. He enters:

```
$DB2STOGP(TESTDASD)
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.29 $DB2SYS (DB2 System)

Function Issues a security check for DB2 System privileges and authorities.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE

Example An administrator wants a security check on the SYSADM privilege, and activates the TRACE parameter so that he can locate the exact permission. He enters:

```
$DB2SYS(SYSADM) TRACE
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.30 $DB2TABLE (DB2 Tables)

Function Issues a security check for DB2 tables or columns.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wants a security check on the table or column USER.MIKE.PAYR with UPDATE access. He enters:

```
$DB2TABLE(USER.MIKE.PAYR) ACCESS(UPDATE)
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.31 $DB2TABSP (DB2 Table Spaces)

Function Issues a security check for DB2 table spaces.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wants a security check on the table space known as TSPACE1, residing on the ACCTING database, with USE access. He enters:

```
$DB2TABSP(ACCTING.TSPACE1) ACCESS(USE)
```

Note: TSSSIM does not take into account hierarchical checking for the DB2 resources.

5.4.32 $DCSS (Discontiguous Saved Statements)

Function Issues a security check for DCSS (Discontiguous Saved Statements).

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check on the program product SAS and to activate the TRACE facility in order to locate the exact permission. He enters:

```
$DCSS(SAS) TRACE
```

5.4.33 $DCT (CICS Destination Control Table)

Function Issues a security check for a CICS Destination Control Table name, limited to a maximum length of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wishes to issue a security check on a DCT PRT6. He can only access this DCT via the privileged program ACDCT5. He enters:

```
$DCT(PRT6) PRIVPGM(ACDCT5)
```

5.4.34 $DEVICES (Hardware Devices)

Function Issues a security check for MVS allocation of UR, graphic and tele-processing devices.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, TRACE, XACCESS

Example An administrator wishes to issue a security check for TP 2703. If the sysid is XAD1 and the address is E80, he enters:

```
$DEVICE(XAD1.TP.2703.E80)
```

5.4.35 $DIAG (VM Diagnose Codes)

Function Issues a security check for VM Diagnose codes.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE

Example An administrator wishes to issue a security check on diagnose code number 84 (Directory Update in Place). He enters:

```
$DIAG(84)
```

Comments The $DIAG command appears on the TSSSIM Security Resource Selection panel as DIAGNOSE.

5.4.36 $DLFCLASS (DLF Data Set)

Function Issues a security check for DLF data sets.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, PRIVPGM, XACCESS, ACCESS, LIBR

Example An administrator wishes to issue a security check on the "VSAM01.CYCLE1" DLF data set. He enters:

```
$DLFCLASS(VSAM01.CYCLE1)
```

5.4.37 $DSN (MVS Data Set)

Function Issues a security check for an MVS data set name. Data set names should be enclosed in single quotes. If the data set is not enclosed in single quotes, the userid is prefixed to the data set which may not provide the desired simulation.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, PRIVPGM, SVC, TRACE, XACCESS, NEWDSN, VOLUME, LIBR, FACILITY

Example An administrator wishes to issue a security check on a data set named ACCTPAY.MASTER with UPDATE access. He enters:

```
$DSN('ACCTPAY.MASTER') ACCESS(UPDATE)
```

Comments If SVC is set to RENAME, the NEWDSN parameter can be used to rename the simulated data set. The $DSN command appears on the TSSSIM Security Resource Selection panel as DATASET.

5.4.38 $FCT (CICS File Control Table)

Function Issues a security check for a CICS File Control Table (FCT) name, limited to a maximum length of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wants a security check on an FCT named PENSION with ALL access. She enters:

```
$FCT(PENSION) ACCESS(ALL)
```

5.4.39 $FIELD (Database Field)

Function Issues a security check for general user-defined resource names. This resource class supports entities with a maximum of 44 characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator needs to issue a security check on a data base field SALARY. This particular database field can only be accessed through a privileged program called PRTFLD. He also wishes to activate the trace facility for this resource. He enters:

```
$FIELD(SALARY) PRIVPGM(PRTFLD) TRACE
```

Comment CA-Top Secret allows the definition of user-defined resources and the validation of access to these resources by user programs or transactions. This resource is intended for those sites that wish to further customize security checks within their environment.

5.4.40 $IBMFAC (IBM Facilities)

Function Issues a security check to determine who has ownership or control over catalog, IDCAMS, SMS, DFDSS and other IBM facilities.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS, LIBR

Example An administrator wishes to issue a security check on the IBM facility IDC.DIAG. He enters:

```
$IBMFAC(IDC.DIAG)
```

5.4.41 $IBMGRP (IBM Products)

Function Issues a security check to determine who has access to or ownership of identifiers which are examined by DB2, DF/HSM and other IBM products.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE

Example An administrator wants a security check on the DB2 resource PAYROLL. He also activates the TRACE facility to locate the exact permission. He enters:

```
$IBMGRP(PAYROLL) TRACE
```

5.4.42 $IUCV (Inter User Communication Vehicle(IUCV))

Function Issues a security check for IUCV communication targets.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE

Example An administrator wants a security check on an IUCV connect to virtual machine APP17 and he also specifies the OWN parameter so that CA-Top Secret will assume the resource is owned. He enters:

```
$IUCV(APP17) OWN
```

Comments **For VM:** IUCV communication targets control the ability of users to issue IUCV connection to the virtual machine designated by the resource name.

5.4.43 $JCT (CICS Journal Control Tables)

Function Issues a security check for a CICS Journal Control Table (JCT) name, limited to a maximum length of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wishes to issue a security check on a JCT named JRN03 with WRITE access. He enters:

```
$JCT(JRN03) ACCESS(WRITE)
```

5.4.44 $JESINPUT (JES Source)

Function Issues a security check for JES job entry sources checked during job initiation.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants to issue a security check to determine if a user can execute jobs which originate from JES node SRCNODE. He enters:

```
$JESINPUT(SRCNODE)
```

Comments Only JES2 SP 3.1.3 and JES3 SP 3.1.3 releases and above offer support for JESINPUT resources.

5.4.45 $JESJOBS (JES Job Submit and Cancel)

Function Issues a security check to determine which job names users can use when submitting or cancelling jobs. JESJOBS can also control which users can submit jobs.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants to issue a security check to determine if a specific user can submit a job using MYJOB as a jobname. He enters:

```
$JESJOBS(SUBMIT.MYNODE.MYJOB.MYACID)
```

Comments Only JES2 SP 3.1.3 and JES3 SP 3.1.3 releases and above offer support for JESJOBS resources.

5.4.46 $JESSPOOL (JES Spool Data Sets)

Function Issues a security check to determine who has access to or ownership of JES Spool data sets (SYSIN and SYSOUT)

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to see if a user may update the JESNEWS (JES2) data set. He enters:

```
$JESPOOL(MYNODE.JES2.$JESNEWS.STC00000.D01.JESNEWS)
```

Comments Only JES2 SP 3.1.3 and JES3 SP 3.1.3 releases and above offer support for all JESPOOL resources.

5.4.47 $LCF (Limited Command Facility)

Function Issues a security check for commands or transactions that are protected through the Limited Command Facility.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters TRACE

Example An administrator wishes to issue a security check on an LCF, and wants to put a trace on the transaction DDYA. She enters:

```
$LCF(DDYA) TRACE
```

Comment The $LCF command appears on the TSSSIM Security Resource Selection panel as CMD/TRAN.

5.4.48 $MDISK (VM Minidisks (VMMDISK))

Function Issues a security check for VM minidisks.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check on the SCAMID.191 mini disk. He enters:

```
$MDISK(SCAMID.191)
```

Comment The $MDISK command appears on the TSSSIM Security Resource Selection panel as VMDISK.

5.4.49 $MGMTCLASS (SMS Management Class)

Function Issues a security check for SMS management classes defined by the storage administrator.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check on the management class named PRODMGMT. He enters:

```
$MGMTCLAS(PRODMGMT)
```

5.4.50 $OPCMD (System Operator Command)

Function Issues a security check for a system operator command that was issued by programs using SVC 34; for example, SDSF, TSO OPER, or installation-written programs.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wishes to issue a security check on an operator command (DISPLAY). He enters:

```
$OPCMD(DISPLAY)
```

5.4.51 $OPERCMD (MVS and JES Operator Commands)

Function Issues a security check on JES and MVS operator commands

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wishes to issue a security check to determine if a user can issue the MVS "D A,L" console command. He enters:

```
$OPERCMD(MVS.DISPLAY.ACTIVE)
```

Comments The OPERCMDS resource class is only used for MVS SP 3.1.3, JES2 SP 3.1.3 or JES3 SP 3.1.3 release or higher.

5.4.52 $OTRAN (Owned Transactions(OTR))

Function Issues a security check for an owned transaction name (OTRAN). IMS and IDMS names can have a maximum of eight characters; CICS names a maximum of four characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, XACCESS, ACCESS

Example An administrator wishes to issue a security check on an owned transaction (OTRAN) named PAYR. He enters:

```
$OTRAN(PAYR)
```

5.4.53 $PANEL (CA Product Panels)

Function Issues a security check for user access to CA product panels.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wants a security check to determine if a user can access panel KOTSTPNL. He enters:

```
$PANEL(KOTSTPNL)
```

Comments Resource names for this class are defined in the corresponding CA product documentation.

5.4.54 $PGM (MVS Program Module Name)

Function Issues a security check for MVS program module names. The names can have a maximum of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE, LIBR

Example An administrator wishes to issue a security check on an OS program named AMASPZAP. He then enters:

```
$PGM(AMASPZAP)
```

Comment The $PGM command appears on the TSSSIM Security Resource Selection panel as PROGRAM.

5.4.55 $PPT (CICS Program Table)

Function Issues a security check for CICS Program Table name (PPT). The names can have a maximum of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE, LIBR, ACCESS, XACCESS

Example An administrator wishes to issue a security check on a PPT named SRSIP. This resource can only be accessed through a privileged program CPROPX. He enters:

```
$PPT(SRSIP) PRIVPGM(CPROPX)
```

5.4.56 $PROPCNTL (Automatic Propagation Control)

Function Issues a security check for those ACIDs not subject to automatic propagation on batch jobs.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wants a security check on the IMS ACID to ensure that it cannot be automatically propagated. In the event that it can, he also wants to activate the TRACE facility to locate the exact permission. He enters:

```
$PROPCNTL(IMS) TRACE
```

5.4.57 $PSB (IMS Program Specification Block)

Function Issues a security check for IMS Program Specification Block (PSB) names. The names may contain a maximum of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE, ACCESS, XACCESS, LIBR

Example An administrator wishes to issue a security check on an IMS PSB named LRLPP. He also wishes to put a trace on the resource to locate the permission for the access. He enters:

```
$PSB(LRLPP) TRACE
```

5.4.58 $PSFMPL (Print Services Facility Modifiable Page Labeling)

Function Issues a security check for output processed by PSF to allow for user suppression of output labeling.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, TRACE

Example An administrator wants a security check to see if a user can suppress PSF identification labels in printed output. He enters:

```
$PSFMPL(PSF.DPAGELBL)
```

5.4.59 $SDSF (Spool Display and Search Facility)

Function Issues a security check for SDSF 1.3 commands, functions, and other resources.

Security Macro RACROUTE REQUEST = AUTH

Parameters ACESS, OWN, TRACE

Example An administrator issues a security check to see if a user can issue MVS and JES2 commands through SDSF. He enters:

```
$SDSF(ISFOPER.SYSTEM)
```

5.4.60 $SMESSAGE (TSO Message Transmission)

Function Issues a security check to determine if a TSO user is allowed to transmit messages to other TSO users via the TSO SEND command.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wishes to issue a security check to determine if TGTUSER is allowed to transmit messages. He enters:

```
$SMESSAGE(TGTUSER)
```

5.4.61 $SPI (CICS System Functions)

Function Issues a security check for the following:

- CEMT INQUIRE, SET and PERFORM
- EXEC CICS INQUIRE, and CICS
- EXEC CICS ENABLE, DISABLE and EXTRACT
- EXEC CICS SPOOLOPEN

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wants a security check on the SPI CONNECTION resource with SET access. He enters:

```
$SPI(CONNECTION) ACCESS(SET)
```

5.4.62 $STORCLASS (SMS Storage Class)

Function Issues a security check for an SMS storage class as defined by the storage administrator.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator issues a security check on the PRODSTOR storage class and he also includes the OWN parameter so that CA-Top Secret will assume that the resource is owned. He enters:

```
$STORCLASS(PRODSTOR) OWN
```

5.4.63 $SUBMIT (ACID Job Submission)

Function Issues a security check to determine if the simulated ACID is allowed to submit a job that will run with the specified ACID name.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters PRIVPGM, TRACE, LIBR

Example An administrator wants a security check to determine if the simulated ACID is allowed to submit a job with the specified ACID of PAYPROD. He enters:

```
$SUBMIT(PAYPROD)
```

5.4.64 $SUBSCH (IDMS Subschema)

Function Issues a security check for IDMS subschema names. These names may consist of up to eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, PRIVPGM, TRACE

Example An administrator wants a security check on an IDMS subschema called PARTA. He also wants CA-Top Secret to assume that the resource is owned. He enters:

```
$SUBSCH(PARTA) OWN
```

Comment The $SUBSCH command appears on the TSSSIM Security Resource Selection panel as SUBSCHEM.

5.4.65 $SYSCONS (System Console)

Function Issues a security check for System Consoles. This is equivalent to the IBM CONSOLE resource class.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check on the 03 console. He enters:

```
$SYSCON(03)
```

5.4.66 $TAPEVOL (Tape Volume)

Function Issues a security check for tape volumes. This is a volume-only check. No data set-level check is performed.

Security Macro RACROUTE REQUEST=AUTH

Parameters ACCESS, TRACE, XACCESS, PRIVPGM, LIBR

Example An administrator wishes to issue a security check on a tape volume, 30T112, with access levels of READ and WRITE. She enters:

```
$TAPEVOL(30T112) ACCESS(READ,WRITE)
```

Comment The $TAPEVOL command appears on the TSSSIM Security Resource Selection panel as VOLUME.

5.4.67 $TERM (Network Terminal ID)

Function Issues a security check for terminal ID names. The names should correspond to actual logical unit IDs as follows:

VTAM eight-character VTAM terminal name

TCAM eight-character TCAM terminal name

BTAM four-character CICS terminal name

Security Macor RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wishes to issue a security check on a terminal known to VTAM by the name T0001012. He enters:

```
$TERM(T0001012)
```

Comment The $TERM command appears on the TSSSIM Security Resource Selection panel as TERMINAL.

5.4.68 $TSAF (Transparent Services Access Facility)

Function Issues a security check for TSAF resources

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check for the FTPSERV TSAF resource. He also wishes to specify OWN so that CA-Top Secret will assume that the resource is owned. He enters:

```
$TSAF(FTPSERV) OWN
```

5.4.69 $TSOACCT (TSO Account Codes)

Function Issues a security check for those account codes that are to be used during TSO logon.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, PRIVPGM

Example An administrator wants a security check on an account code named 044221. He enters:

```
$TSOACCT(044221)
```

5.4.70 $TSOAUTH (TSO Authorities)

Function Issues a security check for TSO user attributes.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, PRIVPGM

Example An adminstrator issues a security check on a TSO MOUNT authority. He also activates the TRACE facility so that he can find the exact permission that allows access to the resource. He enters:

```
$TSOAUTH(MOUNT) TRACE
```

5.4.71 $TSOPRFG (TSO Performance Groups)

Function Issues a security check for TSO performance groups.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check on a performance group named 001. He enters:

```
$TSOPRFG(001)
```

5.4.72 $TSOPROC (TSO Procedures)

Function Issues a security check for PROCs used for TSO logon

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check on a logon PROC of SMPEPROC. He enters:

```
$TSOPROC(SMPEPROC)
```

5.4.73 $TST (Temporary Storage Table)

Function Issues a security check for CICS Temporary Storage Table (TST) names. The names are limited to a maximum length of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wants a security check for a TST named FOLDER. He also wants to use the one-byte hexadecimal code (61) in place of specifying READ, WRITE and PURGE accesses. He enters:

```
$TST(FOLDER) XACCESS(61)
```

Alternately, he could have specified:

```
$TST(FOLDER) ACCESS(READ,WRITE,PURGE)
```

5.4.74 $UR1 (Owned General UR1 Resource)

Function Issues a security check for general UR1 resource names. These names can have a maximum of 44 characters; however, ownership of these resources is based on a maximum prefix length of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wishes to issue a security check for a general resource class 1 with the name ARCH. This resource is accessed through a privileged program ACUPDATE with an access level of READ. He enters:

```
$UR1(ARCH) PRIVPGM(ACUPDATE) ACCESS(READ)
```

5.4.75 $UR2 (Owned General UR2 Resource)

Function Issues a security check for general UR2 resource names. These names can have a maximum of 44 characters; however, ownership of these resources is based on a maximum prefix length of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, PRIVPGM, TRACE, XACCESS

Example An administrator wishes to issue a security check for a general resource class 2 with the name BARG. This particular resource can only be accessed through a privileged program called GABX5. The OWN parameter is specified so that CA-Top Secret will assume the resource is owned. She enters:

```
$UR2(BARG) PRIVPGM(GABX5) OWN
```

5.4.76 $USER (Unowned User Resource)

Function Issues a security check for a non-ownable USER resource of the given one-character class code (A-Z, 0-9). Each name is limited to a maximum of eight characters.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters TRACE, PRIVPGM, LIBR

Example An administrator wants a security check for a non-ownable USERA resource named OIL. The TRACE parameter is specified so that the exact permission causing the access is located. She enters:

```
$USER(A,OIL) TRACE
```

5.4.77 $VMANAPPL (VMAN Applications)

Function Issues a security check for VMAN application names.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator issues a security check on a VMAN application. He also specifies the OWN parameter so that CA-Top Secret will assume that the resource is owned. He enters:

```
$VMANAPPL(TESTTSO) OWN
```

5.4.78 $VMCF (VMCF Communication Targets)

Function Issues a security check for VMCF communication targets.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check on a VMCF communication target. He enters:

```
$VMCF(SVCMACH)
```

Comments **For VM:** The communication targets control the ability of users to issue a VMCF send to the virtual machine designated by the resource name.

5.4.79 $VMDIAL (VMDIAL access)

Function	Issues a security check for virtual machines with DIAL access security
Security Macro	RACROUTE REQUEST=FASTAUTH
Parameters	OWN, TRACE
Example	An administrator wants a security check on virtual machine OPER17. He enters: `$VMDIAL(OPER17)`
Comments	For VM.

5.4.80 $VMMACH (VM Machines)

Function	Issues a security check to verify who can allow a virtual machine to be autologged or logged on using an alternate ACID.
Security Macro	RACROUTE REQUEST=FASTAUTH
Parameters	ACCESS, OWN, TRACE, XACCESS, PRIVPGM
EXAMPLE	An administrator wants a security check on virtual machine BATCH1. He enters: `$VMMACH(BATCH1)`
Comments	For VM.

5.4.81 $VMNODE (VM Nodes)

Function	Issues a security check for VM nodes (RSCS Network Node prefixes).
Security Macro	RACROUTE REQUEST=FASTAUTH
Parameters	OWN, TRACE
Example	An administrator wants a security check on the VMNODE MIAMI. He enters: `$VMNODE(MIAMI)`
Comments	For VM only.

5.4.82 $VMRDR (VM Readers)

Function Issues a security check for VM readers.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check on a CMSBATCH machine. He enters:

```
$VMRDR(CMSBATCH)
```

Comments For VM.

5.4.83 $VSELIB (VSE Library Access)

Function Issues a security check to determine who has access to or ownership of a VSE library.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to see if a user may access the PRD2 library.

```
$VSELIB(VSE.PRD2.LIBRARY.PRD2)
```

5.4.84 $VSESLIB (VSE Sublibrary Access)

Function Issues a security check to determine who has access to or ownership of a VSE Sublibrary library.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to see if a user may access the PRD2.SAVE library.

```
$VSESLIB(PRD2.SAVE)
```

5.4.85 $VSEMEMBR (VSE Member Level Access)

Function Issues a security check to determine who has access to or ownership of a member in a VSE library.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to see if a user may access the member called TSS.PROC in the PRD2.SAVE library.

```
$VSEMEMBR(PRD2.SAVE.TSS.PROC)
```

5.4.86 $VSEPART (VSE Partition Access)

Function Issues a security check to determine who has access to or ownership of a VSE partition.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to see if a user may access the BG partition.

```
$VSEPART(BG)
```

5.4.87 $VSEUSER (VSE User Resource Access)

Function Issues a security check to determine who has access to or ownership of a VSE specific user resource.

Security Macro RACROUTE REQUEST=AUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator wants a security check to see if a user may access the ACNT9999 resource.

```
$VSEUSER(ACNT9999)
```

5.4.88 $VTAMAPPL (VTAM Applications)

Function Issues a security check for authorization to acess VTAM ACB's under a non-APF authorized program.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE

Example An administrator wants a security check for access to the VTAM application SA32SESN. He enters:

```
VTAMAPPL(SA32SESN)
```

5.4.89 $VXDEVICE (VAX Devices)

Function Determines the relationship of volumes to nodes that can access them for the ACID or NDT Record. A volume name must be a device name and a volume must be owned by a dpartment or division.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, TRACE, XACCESS, PRIVPGM

Example An administrator wants a security check on the ZEUS$DUA0 device. He enters:

```
$VXDEVICE(ZEUS$DUA0)
```

5.4.90 $VXFILE (VAX Files)

Function Issues a security check to determine who may transfer ownership of VMS Files.

Note: A VMS file can either be a data file or directory file. Data files are equivalent to MVS data sets and directory files are equivalent to MVS catalogs.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters ACCESS, OWN, TRACE, XACCESS, PRIVPGM

Example An administrator wants a security check on the PEGASUS$DUA0.HORSE VAX file. He enters:

```
$VXFILE (PEGASUS$DUA0.HORSE)
```

5.4.91 $WRITER (JES Writer Control)

Function Issues a security check to monitor routing and processing of job output to local devices, RJE stations or NJE nodes.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters OWN, TRACE, ACCESS, XACCESS

Example An administrator issues a security check to see if a user is authorized to route output to another JES node. He enters:

```
$WRITER(JES3.NJE.SITE2)
```

5.4.92 @cccccccc (User-Defined Resource)

Function Issues a security check for a resource added to the RDT record by the installation. The at sign (@) is followed immediately by the one- to eight-character resource class name.

Security Macro RACROUTE REQUEST=FASTAUTH

Parameters PRIVPGM, OWN, TRACE, ACCESS, XACCESS, SVC

Example An administrator wants a security check for a JOBNAME called TDD022 with UPDATE access. This resource class has already been defined to the RDT record. He enters:

```
@JOBNAME(TDD022) ACC(UPDATE)
```

Comments The tested resource class name (that is, the value of "cccccccc" in @cccccccc) must have been defined previously to CA-Top Secret by means of the ADDTO(RDT) command function. Access levels are relevant only if they were specified when the resource class was defined.

The user-defined resource field appears on the TSSSIM Security Resource Selection panel as DYNAMIC.

5.5 Special Commands

The special simulator commands (EJECT, STATUS, TSS) perform no resource checks. They are primarily designed to provide the administrator with information about the simulator within CA-Top Secret. Each command is discussed in detail, and includes the following information: function, any parameters that may be used to qualify the command, an example, and any additional comments.

The EJECT command is used for batch-only processing of TSSSIM, and causes a page eject before the next command is processed. The STATUS command provides the current status of the simulation environment. The TSS command is "passed through" the simulator to the CA-Top Secret administration processor and supports all of the TSS administration command functions.

5.5.1 EJECT (BATCH Page Eject)

Function A batch-only command that causes a page eject before the next command is executed by the simulator.

Parameters NONE

Example An administrator issues the $DSN and $TERM commands. Before he enters any other commands, he issues EJECT in order to effect a page break.

Note: These commands are entered as control statements in the JCL when executing TSSSIM in batch. For more detailed information, refer to 5.10.2, "Using TSSSIM/BATCH".

5.5.2 STATUS (Status of Environment)

Function Provides the administrator with the current status of the simulation environment, and includes the following:

```
Simulation ACID, data set qualifier, simulation mode
Simulation facility, terminal, CPU
Default SVC name, SVC number, and associated access
Default privileged program PRIVPGM
Default TRACE setting
```

Parameters None

Example An administrator wishes to view the current status of the logged on simulated ACID. He enters:

```
STATUS
```

A screen is then displayed with the following information:

```
TSS8410I    SIMULATED ACID = USER01 QUALIFIER = USER01 MODE = FAIL
TSS8410I    SIMULATED FAC = TSO TERMINAL = T100010 CPU = XA81
TSS8410I    DEFAULT SVC = TSO NUMBER = 13 (HEX) ACCESS = 40
TSS8410I    DEFAULT PRIVPGM = <NONE>
TSS8410I    SECURITY TRACE = NO
```

5.5.3 TSS (TSS Command Functions)

Function This command is not really executed by the simulator. Instead it is simply passed on to the CA-Top Secret administration processor module. All of the administration command functions are supported. This gives the administrator the capability of performing any security administration function without the need to leave, and then subsequently re-enter the simulator.

Parameters TSS command functions and parameters

Example An administrator wishes to view some pertinent information on one of his profiles. He enters:

```
TSS LIST(SYSPROF1) DATA(ALL)
```

5.6 Simulator Trace Information

When activated, the trace may consist of several messages. Each message is described, below.

```
TSS8390I             RESOURCE = (c) xxxxxxxx
```

c resource class code in hexadecimal.

xxxxxxxx resource entity name.

This trace message is displayed to assure the administrator that CA-Top Secret checked the proper resource entity name.

```
TSS8391I        TSS SVC=ss RC=rr DRC=dd VDRC=vv XSW=xx ALG=aa
```

ss security SVC called for the resource check (00=FRACHECK, 82=RACHECK, 83=RACINIT, 85=RACDEF).

rr CA-Top Secret's security return code.

dd CA-Top Secret's detailed reason code for the resource.

vv CA-Top Secret's detailed reason code for volumes.

xx Algorithm results (00=auth, 04=not found, 08=unauthorized).

aa Algorithm selection (00=override, 80=merge, 40=allmerge).

```
TSS8392I      REQUESTED ACCESS = nnnnnnnn,nnnnnnnn,...
TSS8392I      ALLOWED ACCESS = nnnnnnnn,nnnnnnnn,...
TSS8392I      VOLUME ACCESS = nnnnnnnn,nnnnnnnn,...
```

nnnnnnnn access level name for the resource class.

There may be several TSS8392I messages in the trace. The "requested" access assures the check was for what the administrator asked. The "allowed" and "volume" accesses indicate the security accesses that CA-Top Secret allowed the resource and the volume, respectively.

```
TSS8393I        OVERRIDES = xxxxxxxx,xxxxxxxx,...
```

xxxxxxxx one of the security overrides:

DSN	data set security bypass
LCF	limited command facility bypass
RES	general resource security bypass
SUB	job submission security bypass
VOL	volume security bypass
USR-BYP	complete security bypass for user
EMG-BYP	system is in emergency bypass mode

```
TSS8394I            RES ORIGIN = rrrrrrrr - llllllll
```

rrrrrrrr one of the following:

OWNED	resource found to be owned by this ACID
PERMITTED	resource found permitted to this ACID

Note: This permission may be one of denial. Check return codes for allowance or denial.

llllllll one of the following:

USER	contained in user's record
PROFILE=(profile-name)	contained in this profile
ALL	contained in the ALL Record

```
TSS8395I            RES RULE # = nnn
```

nnn consecutive permission number in the record in message TSS8394I.

Note: This message is only present for resource checks where a masking character is not contained in the matching PERMIT that allowed or denied the access. This message is unavailable for resource classes that do not support masking characters.

```
TSS8397I          ------SECURITY PERMISSION------
TSS8397I    RESOURCE = xxxxxxxx
TSS8397I    ACCESS = xxxxxxxx,xxxxxxxx,...
TSS8397I    FACILITY = xxxxxxxx,xxxxxxxx,...
TSS8397I    PRIVPGM = xxxxxxxx,xxxxxxxx,...
                    .
                    .
                    .
TSS8397I          --------------------------------
```

The above represents the security permission that allowed or denied access to the resource. The format is identical to that of a TSS LIST function, so it is easy to compare against a listing of a Security Record.

```
TSS8398I          VOLUME FEEDBACK: f1 f2 f3 f4
```

f1,f2,f3,f4 represent bit-mapped tape volume information. The information presented in these flags goes beyond the scope of user problem diagnosis; however, this information may be requested by Technical Support.

```
TSS8399I         SUBRTN FLAGS: f1 f2 f3 f4 f5
```

f1,f2,f3,f4,f5 represent bit-mapped security subroutine information. The information presented in these flags goes beyond the scope of user problem diagnosis. It may be requested by Technical Support.

```
         TSS8400I         PARENT SVC = pp   CALLING SVC = cc
```

cc the SVC hexadecimal number of the module that called a security routine.

pp the SVC hexadecimal number of the security caller.

```
TSS8401I  CURRENT/PARENT/BOTTOM PRIVPGM(S) = cccccccc pppppppp bbbbbbb  b
```

cccccccc current PRIVPGM program name used for the security check

pppppppp name of the calling program from the parent MVS tcb

bbbbbbbb name of the program called from the current MVS tcb (bottom/last PRB)

An example of trace information follows using the data set name QASCA.LIB.TEXT as the resource class.

```
TSS8390I  RESOURCE = (C4) QASCA.LIB.TEXT
TSS8391I  TSS SVC = 82 RC = 00 DRC = 00 VDRC = 00 XSW = 04 ALG = 00
TSS8392I  REQUESTED ACCESS = READ
TSS8392I  ALLOWED ACCESS = NONE
TSS8392I  VOLUME ACCESS = NONE
TSS8393I  OVERRIDES = DSN SUB VOL
TSS8398I  VOL FEEDBACK: 00 00 00 00
TSS8399I  SUBRTN FLAGS: 80 00 04 00 01
  TSS8400I  PARENT SVC = 85 CALLING SVC = 13
  TSS8401I  CURRENT/PARENT/BOTTOM PRIVPGM(S) = ISPTASK TSSSIM00 ISPTASK
```

Note: These fields contain values initialized by the security algorithm. In this case, access was granted due to the NODSNCHK attribute.

5.7 TSSSIM Facilities

The TSSSIM utility can be executed under the BATCH environment.

5.7.1 Using TSSSIM/BATCH

Since the security administrator is meant to test security permissions resident on the Security File, the administrator may choose to design a series of tests that should be run after major changes are made to the Security File. Rather than enter lengthy commands repeatedly each time the test is run, the administrator can save the commands, and subsequently submit batch tests when required.

The format of the commands should follow the general simulator command format as described in the previous discussions and examples. All commands are supported in the batch environment, although the simulator trace information is somewhat condensed, but very straightforward. In addition, the EJECT command is a batch-only command, and causes a page eject before the next command is executed by the simulator.

The necessary JCL to execute the simulator in a batch environment is described below:

```
// JOB SIMULATE
// EXEC TSSSIM
   .
   .
   .
(TSSSIM Commands)
/*
/&
```

The following JCL illustrates how the simulator can be invoked in a batch environment to perform resource checking on a data set 'VSE.PRD1.LIBRARY' with UPDATE access. Also a DASD volume DOSRES is checked with UPDATE access. The administrator will sign on with the simulated ACID USER01 in a BATCH facility. The TRACE function is also activated.

```
// JOB SIMULATE
// EXEC TSSSIM
LOGON ACID(USER01) FAC(BATCH) TRACE
$DSN('VSE.PRD1.LIBRARY') ACC(UPDATE)
$DASDVOL(DOSRES) ACC(UPDATE)
/*
```

Chapter 6. Using the TSSFAR Utility

The CA-Top Secret File Analysis Routine (TSSFAR) allows security administrators to review the permissions and assignments that are recorded in the Security File. The type of security information displayed by TSSFAR depends upon the control statements selected. Each control statement is discussed in detail following a discussion of the JCL necessary to execute TSSFAR.

TSSFAR can be used to perform the following tasks:

- Provide a cross reference of ALRB block keys with the ARLBs in the block and verify the count against what is in the header block map.
- Review mismatched ARLB chains.
- Review connections between ACIDs and PROFILEs.
- Review connections between owned and owning ACIDs.
- Review resource ownership between ACIDs and resources.

Caution

The file upon which TSSFAR executes will be continuously accessed for the duration of the job. Therefore, running TSSFAR against the Security File could cause significant degradation to system performance. Because of this, we **strongly recommend** that TSSFAR always run against a Backup File.

Only run TSSFAR at the direction of the CA-Top Secret support staff.

6.1 TSSFAR JCL

The following sample JCL can be used to run TSSFAR.

```
// JOB TSSFAR
// ID USER=MSCA,PWD=TORONTO
// EXEC TSSFAR
KEY=C1C1C1C1C1C1C1C1
HEADER
ARLMAP
ALLLOC
ACIDCHAN
ACIDLINK
RESINDEX
```

WHOHAS (resource owning ACID) /*

6.2 Control Statements

The selection criteria used in generating TSSFAR reports are listed below. They are described on the following pages.

Mandatory control statements are:

KEY

Option control statements are:

ARLBMAP
ALLOC
ACIDCHAN
ACIDLINK
HEADER
RESINDEX
WHOHAS (resource owning ACID)

6.2.1 KEY=

Displays customer encryption key in either 16 byte hexadecimal or 8 EBCDIC characters. The KEY control statement is mandatory.

```
KEY=hhhhhhhhhhhhhhhh|'cccccc'
```

6.2.2 ARLBMAP

Prints the ARLB allocation map from the header record.

6.2.3 ALLOC

Prints a cross reference of all ARLB block keys and the actual ARLBs in the block. ALLOC also verifies the number of ARLBs against what is in the header block map.

This routine assumes that if an ARLB is allocated, it should have something in it. Exceptions exist if an ARLB is chained and the first byte was used to add an x'00' to end an XE. This makes the key appear to be wrong because the key is allocated but the ARLB is empty. These cases can be resolved using the ACIDCHAN function.

6.2.4 ACIDCHAN

ACIDCHAN runs the ALLOC function and then uses the ACID index and chase ARLB chains to build a second allocated ACID map. While chasing the chains, TSSFAR will confirm the actual number of chained ARLBs against what the ACID had listed in its FACTREC. ACIDCHAN then lists any ARLBs that are chained but empty. If these match the ARLB numbers reported as key errors in the ALLOC function, the allocation maps are correct.

```
All FACTREC header ARLB counts match actual chains -
```

This message shows that the value found as the number of ARLBs in the FACTREC header matches the number of ARLBs chained together for all ACIDs.

Note: There will be a discrepancy of 12 in the total number of ACIDs reported in your system by TSSFAR compared to a TSS LIST command. This is due to TSSFAR having eight User ACIDs that are reserved and four Department ACIDs that are dynamically built.

ACIDCHAN provides additional analysis of the contents as follows:

- The number of ACID index entries allocated and used.
- The number of ACID index blocks allocated and used.
- The next and last available ACID number.
- Recommended CA-Top Secret cache size.
- Recommended XES structure size.

6.2.5 ACIDLINK

ACIDLINK reviews all ACIDs for connections to other ACIDs. If a connection exists, ACIDLINK verifies whether it should exist. If an ACID shows a profile attached, ACIDLINK verifies that the profile reflects the same information.

6.2.6 HEADER

Prints the first 256 bytes of the header record.

6.2.7 RESINDEX

RESINDEX verifies that all resource ownership indexes match the owning ACID.

6.2.8 WHOHAS (Resource Owning ACID)

WHOHAS (resource owning ACID) lists PERMITS to all resources owned by the specified ACID.

6.3 Sample TSSFAR Output

A sample TSSFAR output is printed below.

```
KEY=
HEADER
ARLBMAP
ALLOC
ACIDCHAN
ACIDLINK
RESINDEX
WHOHAS(FJADEPT)
   ******* HEADER   *******

         +000000   C8C4D940 00001800 000016B8 F88DCF5D   E73B600F ED864B51 F61EB78F C2709644   * HDR ........8..)X.-..f..6...B.o.
         +000020   F7306C1C 48E52F93 14808000 48E52F93   CBA33F60 00000000 00001027 0098240F   * 7.%..V.l.....V.l.t.-.........q..
         +000040   11001335 00000000 00000000 00000000   00000052 00000014 00000004 00000001   * ...............................
         +000060   000001C2 00001480 00000000 00000000   00000018 00000180 00000180 00000180   * ...B...........................
         +000080   000000AF 0000001E 0000000A 00000070   00000008 00000074 00000236 0000006F   * ............................?
         +0000A0   0000001D 00000073 00000008 00000235   000016B5 00007AFF 00004A4A 000005FF   * ......................:.........
         +0000C0   00000000 0001339D 0001EBFF 00000180   00230010 00100100 00000000 00000000   * ...............................
         +0000E0   C2C1C3D2 E4D74040 439B95A0 28068EEE   FA000000 00000000 00000000 00000000   * BACKUP  ..n....................

   ******* ARLB MAP *******

         +000000   00000000 00000000 00000000 00000000   00010000 00000000 00000000 00000000   * ...............................
         +000020   00000000 00000000 00050001 00000200   00000200 00000000 00000000 00020000   * ...............................
         +000040   00000000 00000002 00000102 01040100   00000200 01000000 00000200 00000100   * ...............................
         +000060   00000000 00000000 01010009 03020500   00000000 00000000 00000004 00000000   * ...............................
         +000080   08050000 00000605 02030307 01020001   01040300 00020303 010B0000 000B0A18   * ...............................
         +0000A0   0003060B 000B0003 00001809 01000B00   07180000 00000000 18181818 18181818   * ...............................
         +0000C0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0000E0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000100   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000120   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000140   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000160   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000180   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0001A0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0001C0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0001E0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000200   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000220   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000240   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000260   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +000280   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0002A0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0002C0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
         +0002E0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ...............................
```

Figure 6-1. Sample Output of TSSFAR Utility

```
           C A - T O P   S E C R E T  V e r s i o n   3.0  --  Security File Analysis Utility                    08/28/98

          +000300   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +000320   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +000340   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +000360   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +000380   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +0003A0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +0003C0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +0003E0   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +000400   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................
          +000420   18181818 18181818 18181818 18181818   18181818 18181818 18181818 18181818   * ................................

 *** Allocation Map Validation Begins ***

 RBA 00567 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00005 KEY/ARLB mismatches: 00000

 RBA 00569 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00005 KEY/ARLB mismatches: 00000

 RBA 00570 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00008 KEY/ARLB mismatches: 00000

 RBA 00580 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00002 KEY/ARLB mismatches: 00000

 RBA 00588 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00003 KEY/ARLB mismatches: 00000

 RBA 00596 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00597 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00598 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00599 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00600 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00601 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00602 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00603 RESULTS:  HEADER FREE ARLBS: 00000 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 RBA 00609 RESULTS:  HEADER FREE ARLBS: 00001 KEY/ARLB Matched free: 00012 KEY/ARLB mismatches: 00000

 RBA 00715 RESULTS:  HEADER FREE ARLBS: 00002 KEY/ARLB Matched free: 00013 KEY/ARLB mismatches: 00000

 RBA 00733 RESULTS:  HEADER FREE ARLBS: 00003 KEY/ARLB Matched free: 00024 KEY/ARLB mismatches: 00000

 *** Allocation Map Validation Ends ***
```

Figure 6-2. Sample Output of TSSFAR Utility

```
           C A - T O P   S E C R E T  V e r s i o n   3.0  --  Security File Analysis Utility                    08/28/98

*** ACID Chain Validation Begins ***

All FACTREC header ARLB counts match actual chains

ACID:  RBA 00719   ARLB 0003672      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003673      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003674      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003675      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003676      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003677      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003678      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003679      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003680      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003681      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003682      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003683      Key: USED      Chain: FREE
ACID:  RBA 00719   ARLB 0003684      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003912      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003913      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003914      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003915      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003916      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003917      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003918      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003919      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003920      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003921      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003922      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003923      Key: USED      Chain: FREE
ACID:  RBA 00729   ARLB 0003924      Key: USED      Chain: FREE


Acid index entries allocated:     31,487  Acid index entries defined:      1,900

Next available acid number:       19,018  Last available acid number:     31,487

Acid blocks allocated:             5,248  Acid blocks used:                  173


Recommended TSS cache size:          1033K

Recommended XES structure size:      4236K
```

Figure 6-3. Sample Output of TSSFAR Utility

```
           C A - T O P   S E C R E T  V e r s i o n   3.0  --  Security File Analysis Utility                    08/28/98

Active Acid count:     1,888  Average size:       477 bytes

            SCAs:         81
           LSCAs:         16
           ZONEs:         20
            ZCAs:          4
            DIVs:         48
            VCAs:         92
           DEPTs:        185
            DCAs:         30
           USERs:        817
           PROFs:        579
          GROUPs:         16

*** ACID Chain Validation Ends ***


*** Acid Link Validation Begins ***

ACID RPGDCA1  claims Profile RPGP1    but the Profile doesn't claim the ACID
ACID RPGDCA1  claims Profile RPGP2    but the Profile doesn't claim the ACID

Profile FJAP1    claims ACID SYSOPR   but the  ACID doesn't claim the Profile
Profile HARBE1P1 claims ACID HARBPROF but the  ACID doesn't claim the Profile
ACID RPGD1    claims to be owner of HARBE07  but he denies it
ACID TCSMSCA  claims to be owner of KOTPA01  but he denies it
ACID TCSMSCA  claims to be owner of ZONECA   but he denies it

*** Acid Link Validation Ends ***


*** Resource Index Validation Begins

ACID IJMDEPT1 denies owning RIE  rescode(D0) resource IJMUSER1
ACID CICSDEPT denies owning RIE  rescode(C6) resource FILER

     7,680 RIE  Type Entries allocated      4,862 RIE   Type Resources defined

ACID TCSWKS   denies owning PIE  rescode(C4) resource A1234567.B1234567.C1234567
ACID ELLPH01  denies owning PIE  rescode(E1) resource DLF2.TEST.P001092.D15699V1
ACID PASJE01  denies owning PIE  rescode(E1) resource DLF2.TEST.P001093.D15699V1
ACID ELLPH01  denies owning PIE  rescode(E1) resource DLF2.TEST.P001091.D15699V1
ACID PASJE01  denies owning PIE  rescode(C4) resource PASJE06.PASJE03.PASJE02.PA
```

Figure 6-4. Sample Output of TSSFAR Utility

```
           C A - T O P   S E C R E T  V e r s i o n   3.0  --  Security File Analysis Utility                    08/28/98

ACID ACCTDP   denies owning PIE  rescode(C4) resource REIPA02.SMS.DATA
ACID TOPDINV  denies owning PIE  rescode(C4) resource TOP.INVEST.MASTER.WITH.CRU
ACID RPGP25   denies owning PIE  rescode(9B) resource ZZZZZZZZ.BBBBBBBB.CCCCCCCC

    78,750 PIE  Type Entries allocated     2,553 PIE   Type Resources defined

ACID BUTJO04  denies owning VPIE volume G

       384 VPIE Type Entries allocated         27 VPIE  Type Resources defined

ACID TOPDSFT  denies owning VIE  volume TOPMVS
ACID VOLDEPT  denies owning VIE  volume TSS001
ACID TOPDSFT  denies owning VIE  volume TOPTSO
ACID TOPDSFT  denies owning VIE  volume TOPWK2
ACID HZSP1S   denies owning VIE  volume TSS002
ACID CICSDIV  denies owning VIE  volume TED
ACID TOPDSFT  denies owning VIE  volume TOPPAG
ACID TOPDSFT  denies owning VIE  volume TOPUSR
ACID TOPDSFT  denies owning VIE  volume TOPWK3
ACID TOPDSFT  denies owning VIE  volume TOPIPL
ACID TOPDSFT  denies owning VIE  volume TOPWK1
ACID MDADEPT  denies owning VIE  volume TIM001
ACID TOPDSFT  denies owning VIE  volume TOPRES

     1,536 VIE  Type Entries allocated       202 VIE   Type Resources defined

*** Resource Index Validation Ends

WHOHAS information for ACID: FJADEPT
NRNP5     DATASET   HH.+++++++P
   ACCESS(READ)

NRNP5     DATASET   HH.PY+++++P
   ACCESS(NONE)

ROSCOE    IBMGROUP  ARCCATGP

DINERO    DATASET   HH.+++++++X
   ACCESS(UPDATE)

DINERO    DATASET   HH.LO+++++X
   ACCESS(READ)

VENTURE   DATASET   HH.+++++++X
   ACCESS(ALL)
```

Figure 6-5. Sample Output of TSSFAR Utility

```
           C A - T O P   S E C R E T  V e r s i o n   3.0  --  Security File Analysis Utility                    08/28/98

VENTURE   DATASET   HH.LO+++++X
   ACCESS(READ)

TCSLFM    DB2       DSNR.SSS

FOXYD     DB2       DSNR.SSSS.BATCH

FOXYD     DB2       DSNR.DB2E.BAT

*ALL*     DATASET   'FRANKS.TEST.DSN'
   ACCESS(READ)

SYSADM    DB2       DSNR.DSN.BATCH

TDGRPG    OPCCLASS  AD
   ACCESS(READ)

TCSFJA    ARANK     BINKY
   ACCESS(READ)
   ACTION(FAIL)

TCSFJA    DB2       DSNR.SSSS.BATCH

TCSFJA    ARANK     L2U
   ACCESS(READ)

TDGRPG    DATASET   DB2SYS
   ACCESS(CREATE)

MCCRA01   DATASET   DSNTEST.RENAME
   ACCESS(ALL)
   ACTION(FAIL)

DB2ACID   DATASET   DB2
   ACCESS(ALL)

DB2ACID   DATASET   DSN
   ACCESS(ALL)

*ALL*     DB2       DSNR.DB2T.
   ACTION(AUDIT)

MATGA01   OTRAN     FLOG
   ACCESS(ALL)
```

Figure 6-6. Sample Output of TSSFAR Utility

```
           C A - T O P   S E C R E T   V e r s i o n   3.0  --  Security File Analysis Utility                    08/28/98
HARBE30   TSOPROC   $HARBE03

HARBE30   TSOPROC   $HARB350

ACCTUS3   TSOPROC   $TSO(G)

ACCTUS2   TSOPROC   $TSOAC(G)

ACCTUS5   TSOPROC   $TSOAC

DABAD01   TSOPROC   $DABDB2(G)

DABAD01   TSOPROC   $DABXDC(G)

DABAD01   TSOPROC   $DB2510(G)

*ALL*     TSOPROC   $DABDB2

ADAM01    TSOPROC   $DABDB2

*ALL*     TSOPROC   $DB2510

*ALL*     TSOPROC   $

BERLA02   TSOPROC   $BERLA02

WHOHAS info list complete
***File Analysis Complete***
```

Figure 6-7. Sample Output of TSSFAR Utility

Appendix A. Diagnostic Tools

This section outlines the many tools available to you to determine the cause of authorization problems. Each type of problem requires that only a subset of the tools be used to isolate problems or understand CA-Top Secret's behavior.

Tool	Use or Importance
TSS LIST	Lists ACID information including that contained in attached profiles. Lists global (ALL) authorizations related to the accessed resource. Refer to the *Command Functions Guide.*
TSSSIM utility	Simulates resource access requests made by users. Pinpoints the rule that allows or denies access. Most access problems or behaviors can be effectively diagnosed by TSSSIM. Refer to Chapter 5, *Using the TSSSIM Utility*, in this guide.
TSS WHOHAS	Lists the ACIDs which have access to the resource specified in the command. Refer to the *Command Functions Guide.*
TSSUTIL program	Used to obtain event reports. Refer to the *Report and Tracking Guide.*
TSS WHOAMI	Describes a logged-on user's security environment and important controls such as bypass attributes, log and message display options, and user's mode. Useful for "on-the-spot" debugging. Refer to the *Command Functions Guide.*
TSSOERPT program	Used to obtain a report on OpenEdition security events. Refer to the CA-Top Secret *Report and Tracking Guide*.

MODE	The mode of the actual event, not always the user's or facility's mode. You must know the actual mode, be it facility, user, or event level. The only sure way of knowing the mode of a security event is through a trace. Refer to the *Control Options Guide.*
Authorization Algorithm	Basic understanding of the algorithm is mandatory for definition of proper authorizations. Refer to the *User Guide.*
AUTH control option	Incorrect setting or changing of AUTH dramatically affects whether CA-Top Secret will grant access to a resource. Refer to the *Control Options Guide.*
LOG control option	LOG and FACILITY(LOG=) control CA-Top Secret violation recording and message display. Refer to the *Control Options Guide.*
Message Algorithm	Explains when and why messages are displayed or suppressed. Appendix C describes the algorithm.
MSG control option	Controls message characteristics, such as when they are to be suppressed. Refer to the *Control Options Guide* for details.
TRACE	The diagnostic trace can be employed to provide event-related details that cannot be uncovered by other tools. Appendix B explains how to interpret trace information. See TRACE in the ADD/REMOVE chapter of the *Command Functions Guide* guide and SECTRACE or FACILITY in the *Control Options Guide* for instructions on how to apply the trace.
STATUS control option	Status of various control options and understanding of how they relate to or affect CA-Top Secret's behavior. Refer to the *Control Options Guide* for details.
FACILITY control option	Attributes and status of facility controls. Refer to the *Control Options Guide* guide for details.
DRC control option	Controls or displays violation attributes, especially if your site has changed the effect of some violations. See DRC in the *Control Options Guide* and DRC in the *Messages and Codes Guide.*
TSSAUDIT utility	Generates attribute cross-reference and audits MVS bypasses. See the *Report and Tracking Guide.*

DIAGTRAP control option	Diagnostic dump production. Refer to the *Control Options Guide* for details.
DUMP control option	Generates internal control block dump. See DUMP in the *Control Options Guide.*
TSSCHART program	Diagrams your Security File organization and structure See the *User Guide.*
TSSCFILE program	Formats the Security File records to be used for generating customized reports. See the *Report and Tracking Guide.*

Appendix B. CA-Top Secret Diagnostic Trace

The trace is used to diagnose access problems. It is activated via any one of several control options specified through the MODIFY TSS command or the TSS MODIFY in combination with the TSS ADDTO commands:

Trace Level	Activation
SYSTEM-WIDE	SECTRACE(ON) SECTRACE(WTO) SECTRACE(WTL)
FACILITY-WIDE	FACILITY(TSO=TRACE)
GROUP OF USERS	TSS ADDTO(profile) TRACE
SPECIFIC USER	TSS ADDTO(user) TRACE

B.1 Trace Destinations

TSO traces go to both the user's screen and the system log. CICS, and other online traces go to SYSLOG.

Note: For CICS, you can also write diagnostic trace records into the CICS main trace table; see the *Implementation: CICS Guide* for details.

For batch, traces go to SYSLOG if SECTRACE(ACT,WTL) is specified, or to the security console if SECTRACE(ACT,WTO) is specified.

The trace provides abundant information: 3 or 4 lines worth per event. Therefore keep tracing down to a minimum and be specific about who or what is being traced.

B.1.1 Trace Messages

A sample trace message is shown below. A table that explains what each term in the message indicates appears in the following sections.

Trace messages begin with:

```
TSS-?
```

The ? indicates the type of trace record, as shown below. A trace for a single event comprises four or five trace messages with the headers shown below.

TSS-x-trace data .....
TSS-1 trace data .....
TSS-2 trace data .....
TSS-3 optional trace data if LIBRARY was specified in rule for event
TSS-4 trace data .....

B.1.2 Detail 1

```
TSS-x-rcdr*acid init fcmmrr G/swr1r2dhvh,pfdovoaa L/l1l2ee F/f1f2f3f4,
  c1c2c3,aabb,iijjkk
```

x Event Code:

A=Abend
C=RACROUTE REQUEST=AUTH
D=RACDEF access
E=termination
F=RACROUTE REQUEST=FASTAUTH
I=initiation/signon
L=RACLIST
O=RACROUTE REQUEST=AUDIT
P=Control option
T=TSS command/program
V=password verify (from JES)
X=RACXTRT
Y=VERIFYX call from JES

rc Security Interface Return Code (hex):

00 access allowed

04 resource not owned / ACID not defined / ACTION(PASSWORD) on data set PERMIT

08 access denied / signon password incorrect

0C password expired

10 new password invalid

18 initiation failed by site security exit

1C initiation access failed (see DRC for explanation)

20 force TSO UADS password security

28 OID card required

2C OID card not valid

30 terminal access rejected

34 application access denied

50 surrogate check failed

54 JESJOBs not authorized

dr Detail Violation Reason Code (hex): Specific violation code that denied access or operation (see *Messages and Codes Guide).*

***** If present, indicates that the return code 'rc' was actually passed to the caller. If blank, then a real return code of 00 was returned. A real return of 00 indicates that the user is not in FAIL MODE.

acid The name of the ACID associated with this event.

init A batch jobname, STC procname, or online userid with this event.

f Facility Code:

From the Facility Matrix entry for this facility. Identifies the facility:

T=TSO
C=CICSPROD
B=BATCH
I=IMSPROD
S=STC
K=CICSTEST
N=NCCF
R=ROSCOE

See FACILITY(ID) in the *Control Options Guide.*

c Resource Class or Event:

Identifies the type of resource being accessed, or the operation being attempted. For example: PROGRAM, CPU, TERMINAL, DATASET, ABSTRACT, VOLUME.

mm User or Facility Mode (bit mask):

80=DORMANT
40=WARN
20=FAIL
30=IMPL,
01=CA-Top Secret HAS EXPIRED

rr RACF SVC Flags (bit mapped):

RACINIT:

00 ENVIR=CREATE
04 STAT=NO
08 PASSCHK=NO
40 ENVIR=CHANGE
80 ENVIR=DELETE
C0 ENVIR=VERIFY

RACHECK:

00 RACFIND not specified
01 ENTITY=(,CSA)
02 LOG=NONE
08 31-bit parameters
10 VSAM dataset
80 RACFIND=NO
C0 RACFIND=YES

RACDEF:

00 RACFIND not specified
20 CHKAUTH=YES
80 RACFIND=NO
C0 RACFIND=YES

G/ Algorithm Data

sw Algorithm Switch:

00 access allowed
04 authorization not found
08 access denied
0C volume access is create; force DSN checking
10 volume access is (none)

r1 RELATIVE RULE that allowed or denied data set (first rule is 01, second rule is 02, and so on.)

r2 RELATIVE RULE that allowed or denied volume access

dh ALGORITHM HIGH LENGTH for data sets or resources

vh ALGORITHM HIGH LENGTH for volume access

pf RELATIVE SECURITY RECORD that allowed or denied access:

00 = USER record
FF = ALL record
01-FE = PROFILE 1-254

do DATA SET AUTHORIZATION ORIGIN: (see vo below)

vo VOLUME AUTHORIZATION ORIGIN:

10 = owned (via TSS ADD)
20 = authorized (via TSS PERMIT)
80 = tape owned

aa ACTION from rule that authorized or allowed access (bit mask):

02 PASSWORD or NODSN or DENY
08 NOTIFY
10 EXIT
20 AUDIT
80 FAIL

L/ LOGGING INDICATORS

l1

01 do not write to SMF
02 force message to user
04 send specific message by id
08 do not perform I/O
10 audited event
20 real return code passed
40 forced log-out
80 violation

l2 FLAGS (bit mask):

01 delay after message
02 audit update/alteration
04 audit access if successful
08 audit access or LOG=NOFAIL
10 initiating control ACID
20 reserved
40 do not update feedback area
80 LOG=NONE

ee EVENT CODE:

- **01** job initiation
- **02** resource check
- **32** TSS command
- **33** program change
- **34** change control option
- **39** DUF update
- **40** operator accountability check

F/ FLAG INDICATOR

f1

01 = change propagation
02 = rename
04 = RACROUTE REQUEST=FASTAUTH logging
08 = JES early verify
10 = trace this call
20 = always caller
40 = tape data set request
80 = VSAM data set access

f2

01 = ACTION(EXIT)
02 = no initiation resource checks
04 = FETCH protection required
08 = VTHRESH exceeded
10 = this is a "non" violation
20 = mask search performed
40 = resource is audited
80 = ACTION(PASSWORD)

f3

01 = environment data obtained
02 = MVS system
04 = feedback area validated
08 = do not perform logging
10 = audit this event
20 = simulator trace
40 = TSSSIM simulation
80 = AMODE(31) storage used

f4

01 = PLIST is not RACF-compatiable
02 = this is TCBSENV
04 = third party RACHECK
08 = RACHECK invoked by FRACHECK
10 = at least ONE TSO message issued
20 = priv/exempt caller
40 = initiator in control
80 = JES2 or JES3 in control

c1

01 = password change required
02 = exit continues without checks
04 = no password checking
08 = user mode used
10 = STCACT
20 = VMRDR submission
40 = password violation
80 = security bypass /job

c2

01 = abend switch
02 = address space termination
04 = CHKAUTH=YES
08 = ACEE= supplied with SVC
10 = non 3270 device
20 = GAR retry
40 = undefined ACID
80 = OID card prompt

c3 UNDEFINED ACID RETRY SWITCH:

01 = default ACID
02 = exit called

aa PROCESS/ABEND STATE (bit mapped):

01 = TSSERASE
02 = TSSKGAR
04 = logging interface
20 = installation exit
40 = invalid feedback area
80 = invalid parameter

bb MODULE/ABEND STATE (bit mapped)

01 = TSSKROUT
02 = TSSKEXTR
04 = TSSKCHG
08 = TSSKIXI
10 = TSS utility or command
20 = TSSKSEC
40 = TSSFRACK
80 = TSSKID

ii INSTALLATION EXIT FLAG (bit mapped):

01 = RESERVED
02 = RESERVED
04 = RESERVED
08 = User in BYPASS mode
10 = RESERVED
20 = Exit requested ACID to be suspended
40 = Exit requested auditing
80 = Exit requested MODE change

jj Return Code From Installation Exit

kk Function Code For Entry To Installation Exit

B.1.3 Detail 2

```
TSS-1 ravada v1v23v400 T/t1t2t3t4t5 volser resourcenamenewdsname
```

ra Requested Access Level (see da below)

va Allowed Access Level For Volume (see da below also)

8000 = BLP

da Allowed Access Level For Data Set:

0100 = NOCREATE (volume level access only)
0400 = CONTROL
0800 = SCRATCH
1000 = CREATE
1C00 = ALTER (ALL) (control, scratch, create)
2000 = WRITE
4000 = READ
6000 = UPDATE
8000 = FETCH
FFFF = ALL

v1 Tape Volume Information:

If tape VOLUME protection is active
01 = volume not authorized
08 = scratch requested
10 = volume not owned
20 = volume is owned
40 = volume defined as SCRATCH
80 = not defined to CA-Top Secret

v2 Tape Volume Switch

v3 Tape Volume Disposition (bit mapped):

40 = DISP=OLD
80 = DISP=MOD
C0 = DISP=NEW

v4 Tape Volume Disposition (bit mapped):

00 = RESERVED
01 = UNCATALOGED
02 = CATALOGED
04 = DELETE
08 = KEEP
10 = PASS

00 Reserved

T/	Trace Information
t1	01 = password required for this ACID
	02 = default ACID assigned (INIT) or restricted CA-Top Secret data set accessed
	04 = userid extracted from NJE header
	08 = DRC(FAIL) alteration
	10 = password not validated or vol scan
	20 = exit invoked
	40 = CVOL (CHECK) or random password generation
	80 = resource audit
t2	01 = DINFO occurred
	02 = unused
	04 = TAPE dsname derived or DASDVOL scratch dsname derived
	08 = unused
	40 = RESERVED
	80 = MUAS c/b switch occurred (INIT) or VINFO obtained
t3	01 = data set prefix found 02 = unused 04 = NOxxxCHK access allowed 08 = data set rule found 10 = library scan found library 20 = library scan occurred (CHECK) or forced password change 40 = TMP CALL active 80 = unused
t4	01 = extra restrictions in rule 02 = VSAM access level change 04 = tasklib present 08 = TMP active 10 = no library 20 = fetch-only rule found 40 = extended mask found 80 = floating mask found

t5 01 = KSEC/KID entered
02 = automatic logon occurred
04 = SECREC built
08 = dummy SECREC built
10 = ACEE built
10 = RECVR allowed access (if call is not a RACINIT)
20 = 31-bit XA GETMAIN
40 = ACEE freed
80 = SECREC freed

volser Volume Serial

resourcename
Data Set or Resource Name

newdsname
NEW Data Set Name

B.1.4 Detail 3

```
TSS-2 s1s2cp R/ssffaa terminal S/i1i2i3,a1a2a300 A/afasai P/pgminfo,
    p1,c1c2,p2,p3 F/fafbfcfd
```

s1 SVC in control

s2 SVC above SVC in control

cp CVOL/VSAM flag

R/ RACVT Data

ss STATUS (bit mapped):

02 = AUTOCMDS issued
10 = reserved
20 = emergency bypass active
40 = reserved
80 = TSS address space is DOWN

ff FLAGS (bit mapped):

01 = DEBUG(ON)
02 = reserved
04 = RESERVED
08 = MSUSPEND(YES)
10 = AUTOERASE(YES)
20 = RESERVED
40 = ADSP(ALL) in effect
80 = recovery active

aa ALGORITHM (default AUTH(OVERRIDE,ALLOVER)):

40 = ALL RECORD MERGE
80 = PROFILE MERGE

terminal Online Terminal or Batch Reader Name

S/ SECREC Indicators and Attributes

i1 01 = master SECREC
02 = multi-user address space
04 = reserved
08 = reserved
10 = exclusive LCF present
20 = inclusive LCF present
40 = executing under TMP
80 = bypass active

i2
01 = defined user
02 = TSS address space
04 = in-core DUF update occurred
08 = unused
10 = unused
20 = password verified by JES
40 = reserved
80 = default SECREC

i3
01 = unused
02 = unused
04 = TSO password supplied
08 = error in SECREC
10 = locked due to excessive violations
20 = TSO retry pending
40 = terminal locked
80 = initialization complete

a1
01 = NOSUBCHK
02 = TRACE
04 = OIDCARD required
08 = AUDIT
10 = NOPWCHG
20 = NOADSP
40 = TSOMPW attribute
80 = multiple passwords/fac

a2
01 = unused
02 = NOLCFCHK
04 = NODSNCHK
08 = NOVOLCHK
10 = NORESCHK
20 = SUSPEND attribute
40 = record in error
80 = library(s) permitted

a3
01 = unused
02 = unused
04 = DUFUPD
08 = DUFXTR
10 = reserved
20 = CONSOLE
40 = reserved
80 = MRO attribute

00	Resevered
A/	ACEE Indicators CA-Top Secret
af	01 = ACID defined to CA-Top Secret 40 = ADSP active for user
as	01 = DLI initiation complete 02 = TONE initiation complete 04 = VAM/SPF initiation complete 08 = IDMS/DC initiation complete 10 = multi-user address space 20 = MRO environment 40 = IMS initiation complete 80 = CICS initiation complete
ai	20 = TSB password updated 80 = ACEE completed
P/	Programs in control
pgminfo	Name of program currently in control
p1	Program from current TCB, TOP PRB
c1	CDE ATTR1
c2	CDE ATTR2: 02 = SYSLIB 01 = APF (from LINK or ATTACH)
p2	Program from current TCB, BOTTOM PRB (from LINK SVC)
p3	Program from higher TCB, TOP PRB (mother program)
F/	Facility (Matrix Entry) Attributes
fa	01 = XDEF 02 = MULTIUSER 04 = NOABEND 10 = ASUBM 20 = NSHRPRF 40 = INACTIVE 80 = facility in use

fb
01 = NOINITAUTH
02 = RNDPW
04 = NOINSTDATA
08 = unused
10 = PSEUDO
20 = SIGN(S)
40 = STMSG
80 = LUMSG

fc
01 = transactions being used
02 = TSOC
04 = WARNPW
08 = unused
10 = NORES
20 = AUDIT
80 = TSOPWP

fd
01 = VM SIO checking in effect
02 = IMS extended support
04 = new password reverification
08 = honor password in dormant mode
40 = TRACE
80 = ALWAYSCALL

B.1.5 Detail 4

```
TSS-3 pgm bkbzbt librarydata setname
```

BLDL information only when data set rule specifies library.

pgm BLDL program name

bk BLDL concatenation number of task library

bt BLDL type

bz BLDL module origin:

01 = linklist
02 = tasklib
03+ = higher TCB
03+ = higher TCB tasklib

B.1.6 Detail 5

```
TSS-4 aiai2a3a4 aaaaaaaa ssssssss REQ/reqstor SUB/subsys
```

Audit Indicator Information

a1 First Audit Indicator
01=action of audit
02=user is audited

Can be result of:

- facility has AUDIT attribute FAC(xxx=AUDIT)
- check facility flags
- in trace CA-Top Secret-2 message
- user bypassing security
- installation exit requesting audit of user

04=TEMPDSN(NO)
08=CA-Top Secret file access
10=Bypass DSN check (NODSNCHK)
20=DSN/VOL exit RC=8
40=ACEEOPER set
80=security bypass set

a2 Second Audit Indicator

01=RESERVED
02=DRC(AUDIT)
04=audit of update
08=audit if successful
10=signon—terminal/CPU audited
20=resource audit
40=DSN audit
80=volume audit

a3 Third Audit Indicator

01=Reserved
02=RESERVED
04=RESERVED
08=environmental error
10=user authentication
20=user accountability
40=lock request
80=no authority for function

a4 Fourth Audit Indicator

RESERVED

aaaaaaaa	Address of ACEE
ssssssss	Address of SECREC
reqstor	The REQSTOR= parameter from the RACROUTE macro
subsys	The SUBSYS= parameter from the RACROUTE macro

B.2 Example Diagnostic Traces and Meanings

B.2.1 Example 1. CPU Validation During TSO Logon

```
TSS-F-0000 NLSMPK   NLSMPK   T U4000 G/0000000000,00000000 L/00A002
F/04000F20,000800,0001,000000
TSS-1 000000 0000000000 T/0000000000        CPU.XA81
TSS-2 830000 R/108A00 S/000100,02002000 T0001013 A/010080 P/IKJEFLC,
B512,        ,IDJEFLA F/80C20600
```

- TSS-F indicates RACROUTE REQEST=FASTAUTH validation.
- Return code 00 and no violation (00).
- ACID and jobname both NLSMPK.
- T indicates TSO.
- U indicates Abstract.
- 4000 shows Warn MODE.
- Logging indicates Log=None (normal for CPU check by CA-Top Secret).
- Resource is CPU.XA81.
- SVC in control is 83 (Racinit).
- User (ACID) is defined. ACID has TRACE and CONSOLE attributes.
- Terminal is T0001013. Program in control is TSO scheduler (IKJEFLC), running.
- Facility using LastUsed and Status, RndPw, WarnPw options.

B.2.2 Example 2. Password Violation during TSO Logon

```
TSS-I-0809*NLSMPK   NLSMPK   T   4000 G/0000000000,00000000 L/F01001
F/00000330,400000,0081,000040
TSS-1 000000 0000000000 T/1100000015
TSS-2 008000 R/108A00 S/000100,02002000 T0001013 A/010080 P/IKJEFLC
B512,        ,IDJEFLA F/80C20600
```

- TSS-I shows initiation call. Real return code passed to TSO (*).
- Return code violation code is 09.
- Logging indicates violation, forced logging, real return and event being audited.
- Flags indicates password violation.
- Trace (11) shows password required but not validated.

B.2.3 Example 3. Reader Access Violation for Batch Job

```
TSS-I-1C88 NLSMPK BJOB2   B TERMINAL 4000 G/0000000000,00000000
L/801001, F/00000330,000000,0081,000040
TSS-1 000000 0000000000 T/0100000015        R5.R1
TSS-2 008000 R/108A00 S/000100,02002000 T0001013 A/010080 P/IEFIIC ,
B512,IEFIB600,IEESB605 F/80C00400
```

- Another initiation call. Return code passed to initiator is 00, but would be 1C if this had been a real failure instead of a warning.
- Violation is 88 (136).
- B indicates Batch facility.
- TERMINAL indicates batch job performing reader violations.
- Trace (01) shows required password has been validated.
- Source of origin is remote 5.
- Program in control is the initiator.

B.2.4 Example 4. Authorized Access to data set

```
TSS-C-0000 NLSMPK   LINKMVS  B DATASET 4081 G/0009080906,00202000
L/100002 F/00000320,000000,0021,000040
TSS-1 60FF10 0000000000 T/0000090001 SRVC01 SYS2.TSS.TEST.LOAD
TSS-2 160000 R/108A00 S/000180,02002000 INTRDR   A/0100A0
P/IEWL   ,
0B22,        ,IEFIIC F/80C00400
```

- Job LINKMVS has accessed SYS2.TSS.TEST.LOAD for update access (60).
- ACID NLSMPK has All access (FF).
- Algorithm data shows access allowed (00).
- Ninth data set permission in user record.
- Data set rule had a permission of length nine (actually, SYS2.TSS.), volume rule had six characters (SRVC01).
- Open-J was in control (16).

B.2.5 Example 5. Failure Due to Bad Access Level with ACTION(FAIL)

```
TSS-D-0866*NLSMPK   NLSMPK   T DATASET 20A0 G/08070A1106,00202080
L/B00002 F/00000330,000000,0021,000040
TSS-1 486010 0000000000 T/0000010801 STRG02 NLSMPK.SYSLOG.DATA
TSS-2 1E0000 R/108A00 S/440180,02002000 T0001013 A/0100A0 P/RENAME
3512,        ,IKJEFT09 F/80C20600
```

- A real return code of 08 (*) was passed to the rename (1E) SVC.
- 66 indicates wrong access level attempt. T DATASET shows TSO and data set resource.
- A0 indicates RACFIND=NO (no RACF bit).
- Algorithm shows access illegal (08), seventh data set rule, tenth volume rule.
- Action is FAIL (80), which changes running mode to FAIL (20) for this event.
- TSO command is RENAME, running directly under the TMP (IKJEFT09).

B.2.6 Example 6. Job Submission (Authorized)

```
TSS-C-0000 NLSMPK   NLSMPK   T ALT-ACID 4081 G/0000000000,00000000
L/100002 F/00000330,000800,0021,000040
TSS-1 400000 0000000000 T/0000000001        NLSMPK  TSSRACL
TSS-2 000000 R/108A00 S/440180,02002000 T0001013 A/0100A0 P/IKJEFF04,
3122,        ,IDJEFF76 F/80C20600
```

- TB indicates job submission (B) from TSO (T).
- ACID for job TSSRACL is NLSMPK.
- Submit command (IKJEFF04) was used.

B.2.7 Example 7. Program Violation

```
TSS-F-0888 IMSRG2   IMSB   S PROGRAM 4000 G/0000000000,00000000
L/802002 F/04000720,000800,0001,000040
TSS-1 000000 0000000000 T/0000000400        DFSFDLD0
TSS-2 2A0000 R/108A00 S/000180,02000000     A/0100A0 P/DFSXDSP0,0B22,
        ,DFSMVRC0 F/80C00000
```

Here a program (DFSXDSP0) which is running as started task IMSB is attempting to access program DFSFDLD0.

- 0888 indicates resource not accessible.
- S PROGRAM shows STC with program check.
- 4000 shows WARN MODE, therefore the security driver receives a real return code (0) and continues normally.

B.2.8 Example 8. CICS Transaction Violation

```
TSS-F-0888*DANTEST  CICS50   K LCF 2000 G/0000000000,00000000 L/A02002
F/04000720,000800,0001,000040
TSS-1 000000 0000000000 T/0000000400        CSMT
TSS-2 000000 R/108A00 S/200180,0A008800 T0001008 A/01B0A0 P/DFHSIP,0B23 ,
        ,IEESB605 F/16C21500
```

- TSS-F is a FRACHECK-processed CICSTEST,LCF (K LCF) event.
- 0888* indicates that a real return code was passed back to CICS because the ACID is in FAIL MODE (20).

B.2.9 Example 9. CICS Resource Violation

```
TSS-F-0888*DANTEST  CICS50   K PPT     2000 G/0000000000,00000000
L/A02002 F/04000720,000800,0001,000040
TSS-1 000000 0000000000 T/0000000400        DFHEMTP
TSS-2 000000 R/108A00 S/200180,0A008800 T0001008 A/01B0A0 P/DFHEMTP,  3
0B23,        ,IEESB605 F/16C21500
```

- A real violation again (*) for PPT (Q) resource DFHEMTP.

Appendix C. Message Display/Suppression Algorithm

The checks that are made to determine whether or not the user or security console should receive a message are documented in the flowchart on the following page.

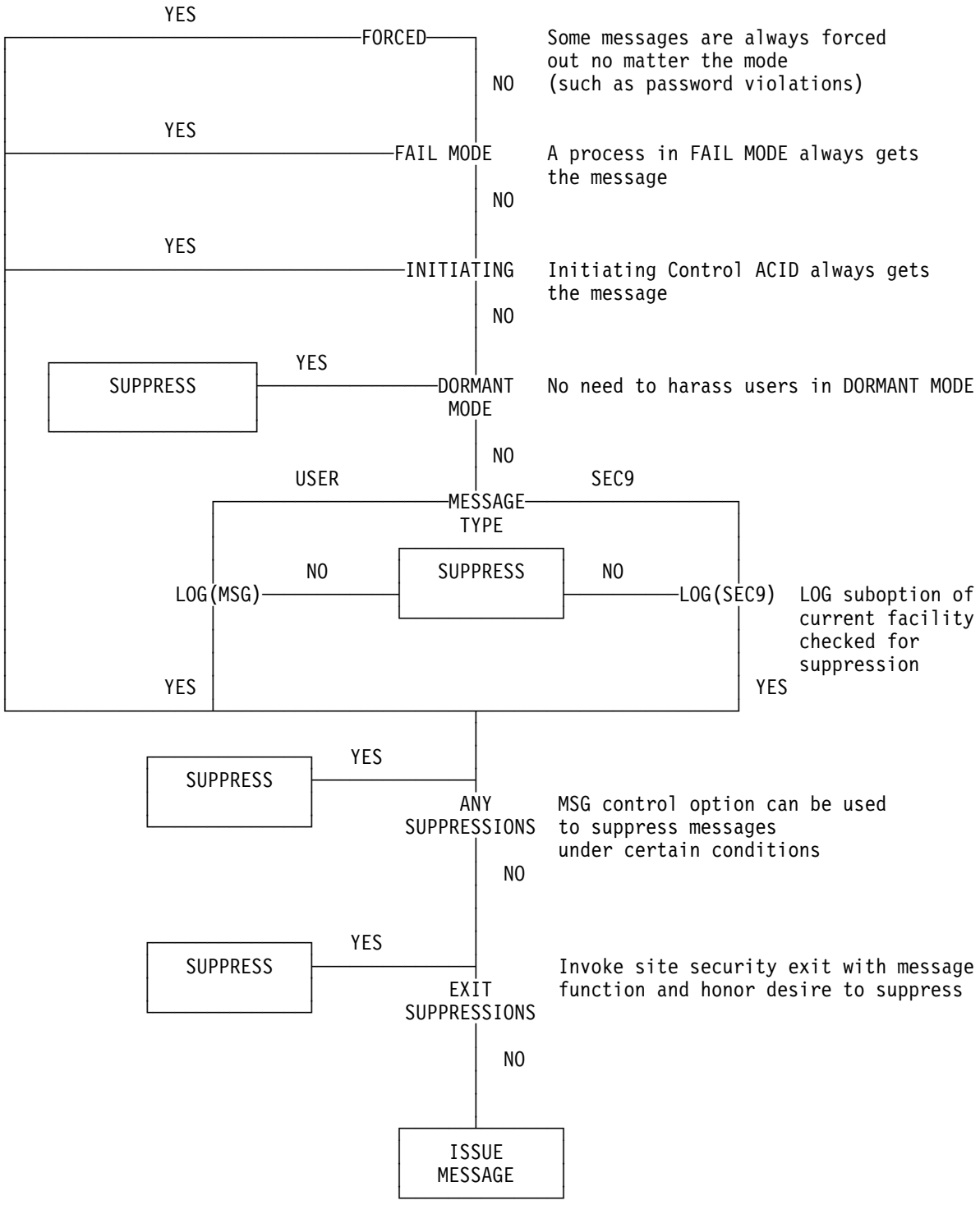

Figure C-1. CA Top Secret Message Algorithm

Index

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 Troubleshooting Guide

Release 3.0 VSE

Document Number: R101TS30TSE

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:

DSMKPO653E POSTSCRIPT FILE '@E@P@S' NOT FOUND.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO653E POSTSCRIPT FILE '@E@P@S' NOT FOUND.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO653E POSTSCRIPT FILE '@E@P@S' NOT FOUND.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S
DSMKPO653E POSTSCRIPT FILE '@E@P@S' NOT FOUND.
DSMMOM395I '.EDFPO' LINE 70: .po @E@P@S