CA Top Secret® for z/OS

Troubleshooting Guide

r15

Fourth Edition

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

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

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

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

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

The manufacturer of this Documentation is CA.

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

CA Technologies Product References

This documentation set references the following CA products:

- CA ACF2™ for z/OS (CA ACF2)
- CA Common Services for z/OS (CA Common Services)
- CA Distributed Security Integration Server for z/OS (CA DSI Server)
- CA LDAP Server for z/OS (CA LDAP Server)
- CA Top Secret® for z/OS (CA Top Secret)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- TYPE Operand—Identify the Type of Event to Process (see page 112)—Added a description for HFS, which specifies to trace internal functions of CA SAF HFS security.

The following documentation updates were made in the previous release of this documentation:

- (TSO) Invoke a Simulation Session to Test Security File Permissions (see page 41)—Consolidated logon information into one procedure.
- Perform a Simulated Resource Check (see page 44)—Added this consolidated procedure (applicable to multiple environments) that describes how to invoke simulation resource commands to test permissions on the security file without affecting the production environment. You can also view details about resource classes (and commands) that are available for simulation.

Contents

Chapter 1: Troubleshooting 9

Common Troubleshooting Procedures 9

Abends Occurring 10

Facility Access Incorrectly Allowed 11

Passwords not Checked 12

CPU Restrictions not Honored 13

Facility Access Incorrectly Denied 14

CA Top Secret Not Logging Violations 15

CA Top Secret Not Logging Access or Initiations 15

Users not Receiving CA Top Secret Messages 16

Console not Receiving CA Top Secret Messages 16

Data Set or Volume Access Incorrectly Denied 17

Data Set Passwords not Checked 18

Troubleshoot Customization Problems 19

Customization Tips and Checkpoints 20

Installation Exit 21

Chapter 2: Performance Tuning Considerations 23

Performance Related Statistics 24

Statistics Gathering 25

CPF Statistics 26

Security File Location 26

Recovery File Location 27

CA Top Secret Address Space 27

Data Integrity 27

Sharing Considerations 27

ACID Index Performance Setting 28

File Sharing and the Coupling Facility 28

CA Top Secret CACHE 28

CACHE and CF Size 29

Security File Cache 30

Activate the Cache 31

Security File Cache Size 31

SECCACHE in a Shared Security File Environment 32

Tuning CPF 32

CICS Performance Considerations 33

CICS Signon Sub-Tasking 33
Shared Profile Table 34
CICS / DB2 Considerations....................34

Chapter 3: Using the TSSSIM Utility 35

Introduction to TSSSIM 35
Authority and Scope....................36
TSSSIM Under TSO in a Non-SPF Environment....................36
Use TSSSIM Under TSO in an SPF Environment 37
TSSSIM TSO Online Tutorial 38
TSSSIM in TSO/E-Background, TSO/E-Foreground, or TSO/E-Batch....................39
Execute the Simulator in a Batch Environment....................40
TSSSIM and CA Roscoe 41
(TSO) Invoke a Simulation Session to Test Security File Permissions....................41
LOGOFF/SIGNOFF....................43
HELP Command—List and Describe Commands 43
ENVIRON Command—Change Simulated Environment....................43
END/QUIT Command—Exit Simulator 44
Perform a Simulated Resource Check 44
TSSSIM Special Commands....................48
EJECT Command—Page Eject....................48
STATUS Command—Current Environment's Status....................49
TSS Command—Performing Security Administration Functions 49

Chapter 4: Using the TSSFAR Utility 51

About the TSSFAR Utility....................51
Required Authority....................51
Set Up MEMLIMIT 52
TSSFAR JCL....................53
TSSFAR Control Statements 53

Chapter 5: TSSRECVR - Security File Recovery 59

About TSSRECVR....................59
Recovery Procedure for Automatic Backup 60
Manual Recovery Procedure....................63

Chapter 6: Using the TSSXTEND Utility with WHOHAS 65

About the TSSXTEND Utility with WHOHAS 66
Required Authority for TSSXTEND Utility with WHOHAS....................67

Run the TSSXTEND Utility with WHOHAS 67

Chapter 7: TSSXVSDT - Digital Certificate Backout 69

About TSSXVSDT 69
VSAM Digital Certificate Backout 69

Chapter 8: PDS Member Level Protection Utilities 75

About PDS Member Level Protection 75
Security Ownership Commands 76

Appendix A: CA Top Secret Diagnostic Trace 79

Introduction 79
Trace Destinations 80
Trace Messages 80
Trace Detail 1 81
Trace Detail 2 90
Trace Detail 3 94
Trace Detail 4 99
Trace Detail 5 100
Trace Detail 6 102
Diagnostic Traces Examples 103

Appendix B: Tracing SAF Requests 109

About the SAF SECTRACE Command 109
SAF SECTRACE Command 110
SECTRACE Command Use 111
SAF SECTRACE SET and MODIFY Operands 111
TYPE Operand—Identify the Type of Event to Process 112
OMVS Operands—Define Callable Services 113
TRACING Operands—Event Matching 114
ENVIRONS Operands—Environment Filter 115
STATUS Operands—Define Trap Status 116
EVENTS Operand—Maximum Number of Events 117
ROUTE TO Operands—Output Format and Routing 118
END and CANCEL Operands—Complete Trap Specification 121
Stop and Restart SECTRACE 121
Enable SAF SECTRACE Traps 121
Disable SAF SECTRACE Traps 121
Delete SECTRACE Traps 122

Display SAF SECTRACE Operands 122
SAF SECTRACE at SAF Initialization 122
    Example: Installation Defaults 123
    Example: Prompted Implementation 124
Tracing UNIX System Services (OMVS) 125
    Stop SECTRACE for OMVS 125

Index 127

Chapter 1: Troubleshooting

This section contains the following topics:

Common Troubleshooting Procedures (see page 9)
Troubleshoot Customization Problems (see page 19)

Common Troubleshooting Procedures

Use these procedures to determine the cause and resolution of the problem or to assemble as much information as you can about the problem.

Abends Occurring

Symptom:

Abends are being generated.

Solution:

Do the following:

- Provide a system dump COPY from the dump data set onto a 6250 BPI magnetic tape or cartridge. Sample JCL, is shown below:

```
//DUMP          JOB
//              EXEC            PGM=IEBGENER
//SYSPRINT      DD              SYSOUT=*
//SYSIN         DD              DUMMY
//SYSUT1        DD              DISP=SHR,DSN=SYS1.DUMPxx
//SYSUT2        DD              DISP=(,KEEP),UNIT=TAPE,DSN=xxx
```

- Provide all copies of the system log data from 15 minutes before and 5 minutes after the abend occurred. Hardcopy is preferred. Include the issue number in the subject field.
- If the abend was produced by a batch job, provide a copy of the JCL for that batch job and the JES joblog.
- If TSS message TSS9999E indicated the abend, record the text of the message, including all register and offset information which follow the message.
- Copy all other CA Top Secret messages that appear with the job or session in the order in which they occur.

Most CA Top Secret abend messages and dump titles contain and document the abend code, program name, and program offset associated with the error. When an CA Top Secret dump exists in a SYS1.DUMPxx data set, the z/OS operator command “D D,E” can be used to gather further details about the error. Similarly, when an CA Top Secret dump is being viewed via IPCS, the “ST FAILDATA IPCS command can now be used to obtain a summary of the error. Whenever possible provide “ST FAILDATA” (or “D D,E”) output to CA when reporting an abend.

Facility Access Incorrectly Allowed

Symptom:

Users are being granted incorrect access to facilities.

Solution:

Do the following:

- Determine if the facility appears within the user or profile ACID's Security Record:

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

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

  Remove the facility from the Security Record or remove the user from the profile.

- Determine if the user ACID, profile ACID or facility are in DORMANT or WARN MODE:

  ```
  TSS MODIFY(FACILITY(fac))
  ```

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

  ```
  TSS LIST(ALL) DATA(XAUTH)
  ```

  Consider moving the user to a more restrictive security mode or set WARNPW in the FACILITY control option.

- Determine if MSG is displayed in the LOG field in response to a WHOAMI command.

  If not, the MSG option is not specified.

- Determine if the user possesses the NORESCHK attribute:

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

  Remove the NORESCHK attribute from the ACID.

- Determine if the user is allowed to bypass security by locating BYPASS message TSS9530I on the STATUS response:

  ```
  TSS MODIFY(STATUS)
  ```

  Reset BYPASS:

  ```
  TSS BYPASS(RESET).
  ```

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

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

  Reset the DRC NOVIOL attribute.

- Activate the diagnostic trace for the user.

  If trace records do not appear for facilities other than TSO, BATCH, or STC the site might not have a working interface with CA Top Secret.

Passwords not Checked

Symptom:

Users passwords are not being checked.

Solution:

Do the following:

- Determine if the user's or profile's ACID are in DORMANT or WARN MODE:

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

  Consider moving the user to a more restrictive security mode.

- Determine the facility's ACID is in DORMANT or WARN MODE.

  ```
  TSS MODIFY(FACILITY(fac))
  ```

  Assign the WARNPW attribute to the facility to force defined users and jobs to enter their correct passwords while in WARN mode.

- Enter TSS WHOAMI command at the user's terminal.

  If MSG is not displayed in the LOG field, LOG options are not specified and no messages are being sent to the user.

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

  ```
  TSS MODIFY(STATUS)
  ```

  If the user's ACID, jobname or started task name appear in this list remove the applicable bypass attribute from the user's ACID, jobname or started task name.

- Determine if trace records appear for facilities other than TSO, BATCH, or STC. If not, the site might not have a working interface with CA Top Secret.

CPU Restrictions not Honored

Symptom:

CPU restrictions are not being honored.

Solution:

Do the following:

- Determine if the CPU restriction exists for the user(s) and CPU:

  ```
  TSS LIST(acid) DATA(BASIC,XAUTH,PROFILE)
  ```

  ```
  TSS LIST(ALL) DATA(XA)
  ```

  If permissions exist which allow the user to access the CPU use the TSS REVOKE and PERMIT commands to define the access.

- Determine if the CPU is owned.

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

  If not owned, see the CPU keyword in the *Command Functions Guide*. If a resource is not owned, CA Top Secret cannot restrict access.

- Determine if security mode for the facility (or site) and for the user or profile ACID(s) is in DORMANT or WARN MODE:

  ```
  TSS MODIFY(FACILITY(fac))
  ```

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

  ```
  TSS LIST(ALL) DATA(XAUTH)
  ```

  The mode might allow access without security checking. Consider moving the user to a more restrictive security mode.

- Enter TSS WHOAMI at the user's terminal.

  If MSG is not displayed in the LOG field, LOG options are not specified and no messages are being sent to the user.

- Determine if the ACID possesses the NORESCHK attribute:

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

  If yes, enter:

  ```
  TSS REMOVE(acid) NORESCHK
  ```

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

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

  If yes, reset the DRC.

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

  ```
  TSS MODIFY(STATUS)
  ```

  If the user's ACID appears in this list remove the applicable bypass attribute from the user's ACID.

- Activate the diagnostic trace for the user.

  If trace records do not appear for facilities other than TSO, BATCH, or STC the subsystem might not have a working interface with CA Top Secret.

Facility Access Incorrectly Denied

Symptom:

Users are being denied access to facilities.

Solution:

Do the following:

- If there is a Detailed Violation Reason code (DRC) displayed with the violation, see the *Messages and Codes* Guide to determine if the correct authorizations have been made.
- Determine the facilities the user is allowed to access through ADD or PERMIT authorizations.

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

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

  If authorizations do not exist which permit the user to access the facility add the facility to the user's ACID.

- Activate the diagnostic trace for the user.

  If trace records do not appear for facilities other than TSO, BATCH, or STC the facility might not have a working interface with CA Top Secret.

CA Top Secret Not Logging Violations

Symptom:

Violations are not being logged.

Solution:

Do the following:

- Determine if the site is in dormant mode:

  ```
  TSS MODIFY(STATUS)
  ```

  If the site or facility is in dormant mode CA Top Secret does not perform logging. Upgrade the mode of user or facility.

- Determine if the DRC set to NOVIOL:

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

  If the DRC is set to NOVIOL enter:

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

- 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. Ensure that the Audit/Tracking File exists.

  If SMF and/or //AUDIT are specified, enter:

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

CA Top Secret Not Logging Access or Initiations

Symptom:

Security events are not being logged.

Solution:

Do the following:

- Determine if the site or facility are in dormant mode:

  ```
  TSS MODIFY(STATUS)
  ```

  ```
  TSS MODIFY(FACILITY(fac))
  ```

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

  ```
  TSS LIST(ALL) DATA(XAUTH)
  ```

  If the site or facility are in dormant mode CA Top Secret does not perform logging. Upgrade the mode of the user or facility..

- If ACCESS or INIT are not specified as suboptions of the LOG control option, specify them.

Users not Receiving CA Top Secret Messages

Symptom:

Users are not receiving CA Top Secret messages.

Solution:

Do the following:

- Determine if the site or facility is in dormant mode:

  ```
  TSS MODIFY(STATUS)
  ```

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

  ```
  TSS LIST(ALL) DATA(XAUTH)
  ```

  ```
  TSS MODIFY(FACILITY(fac))
  ```

  CA Top Secret does not perform logging in DORMANT MODE. Upgrade the mode of the user or facility.

- Determine if the MSG suboption of the LOG control option is specified:

  ```
  TSS MODIFY(FACILITY(fac))
  ```

  If not specified, see the *Control Options Guide*.

Console not Receiving CA Top Secret Messages

Symptom:

The console is not receiving CA Top Secret messages..

Solution:

Do the following:

- Determine if the site or facility is in dormant mode:

  ```
  TSS MODIFY(STATUS)
  ```

  ```
  TSS MODIFY(FACILITY(fac))
  ```

  CA Top Secret does not perform logging in dormant mode. Upgrade the security mode if logging is required.

- Ensure that the SEC9 suboption of the LOG control option is specified. For information, see the *Control Options Guide.*

Data Set or Volume Access Incorrectly Denied

Symptom:

Users cannot access a resource.

Solution:

Do the following:

- Determine the user's and the facility's mode:

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

  ```
  TSS LIST(ALL) DATA(XAUTH)
  ```

  ```
  TSS MODIFY(FACILITY(fac))
  ```

  If the user has a more restrictive mode than the facility, permit the user to a less restrictive mode or permit explicit access to the resource.

- Determine if the DEFPROT attribute is attached to the data set:

  ```
  TSS LIST(RDT) RESCLASS(DATASET)
  ```

  Is the DEFPROT attribute is attached to the data set, permit the user explicit access to the resource.

- Access TSSSIM to simulate a data set access attempts by entering the TSSSIM parameters:

  - ACID—User's ACID
  - FACILITY—Facility user had access to when the problem occurred
  - MODE—The facility's mode
  - TRACE—Yes
  - ACCESS—The access that the user requested when access was denied

- List the user's profiles and user Security Record:

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

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

  Examine ownership and XAUTH fields and the security records of any connected profiles. Data set access problems might be caused by volume access rules.

  If the display does not show authorizations at the requested access level and to the resource permit the user access to the resource through the user record, ALL Record, or through connection to a profile which has the appropriate access.

- Check the settings of the AUTH control option:

  ```
  TSS MODIFY(STATUS)
  ```

  If AUTH been changed after the Security File was originally established attempt to reconstruct Security File to match the required access definitions.

Data Set Passwords not Checked

Symptom:

Users are being granted access to a resource without being checked.

Solution:

Do the following:

- Determine if the data set has the ACTION(PASSWORD) attribute:

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

  If the ACTION(PASSWORD) is attached, permit the data set attaching the ACTION(PASSWORD) attribute.

- Determine if the ACID has the NODSNCHK attribute:

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

  If the ACID has NODSNCHK attribute remove authority with:.

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

- Determine if the user has ACTION(NODSNCHK) authority to the volume:

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

  ```
  TSS LIST(ALL) DATA(XAUTH)
  ```

  If the volume is authorized with ACTION(NODSNCHK) CA Top Secret will not perform data set level checking.

- Turn on the TSS TRACE for the ACID:

  ```
  TSS ADDTO(acid) TRACE
  ```

  If the acid is signed on, issue:

  ```
  TSS REFRESH(acid)
  ```

  Activate the TRACE from the console:

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

- Ask the user to access the data set in question. Examine the trace information.

  - If 04 DRC is being returned, the data set is authorized with ACTION(PASSWORD).

  - If 04 DRC is not being returned, 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.

Troubleshoot Customization Problems

Customization problems can stem from any one of the three supported CA Top Secret customization techniques:

- The z/OS Security Interface
- The Installation Exit
- The Application Interface

To determine the cause of customization problems

1. Write down all message numbers and text displayed by CA Top Secret. See the *Messages and Codes Guide* to determine the meaning of the CA Top Secret messages.
2. Turn on the CA Top Secret TRACE:

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

   ```
   TSS ADDTO(acid) TRACE
   ```

3. If the acid is signed on, either issue TSS REFRESH(acid) JOBNAME(*) or have the acid signoff and back on again.
4. To turn off the TRACE:

   ```
   TSS MODIFY(SECTRACE(OFF))
   ```

   ```
   TSS REMOVE(acid) TRACE
   ```

5. Examine the TRACE information. The TRACE shows the information being passed to CA Top Secret along with DRCs and other diagnostic information.
6. Obtain a dump using DC F'00' or the CA Top Secret DIAGTRAP control option. Examine the parameter list being sent to CA Top Secret.

Customization Tips and Checkpoints

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 z/OS (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.
  - 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.

Installation Exit

Installation exit checks:

- 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 non zero entry for the desired function.
- The customized code is in the appropriate section of the TSSINSTX module.
- The EXIT control option is ON. F TSS,STATUS lists the status of EXIT.
- Under TSO, the installation exit code is tested in Key 8 by:
  - Loading TSSINSTX
  - Building your own parameter list as expected by the exit
  - Passing control to TSSINSTX
  - Examining the parameter list, feedback areas, and return codes

To test your installation exit code set up debug code which executes the exit only for a particular ACID, jobname:

- Locate the correct branch point in the Installation Exit.
- Check for the ACID, jobname, and so on:
- To locate the jobname or TSO session name, use the following code:

```
L    R2,PSATOLD-PSA(0)    TCB
L    R2,TCBTIO-TCB(R2)    TIOT
      CLC  TIOCNJOB-TIOT1(8,R2)=CL8"jobname or TSO session"
*
      BE   OURJOB   it's our job so perform the exit code
 *
       BNE  NOTJOB   it's not our job, exit so we don't interfere
       with other users
*
.
.
.
IHAPSA     PSA expansion
KJTCB      TCB expansion
IEFTIOT1   TIOT expansion
```

  Since the acidname is passed into a particular field in the installation exit, another way to locate the ACID is to examine the field.

- If running under ESA or XA, refresh LLA after linking a new exit:
  `F LLA,REFRESH`

Chapter 2: Performance Tuning Considerations

This section contains the following topics:

Performance Related Statistics (see page 24)
Statistics Gathering (see page 25)
CPF Statistics (see page 26)
Security File Location (see page 26)
Recovery File Location (see page 27)
CA Top Secret Address Space (see page 27)
Data Integrity (see page 27)
Sharing Considerations (see page 27)
ACID Index Performance Setting (see page 28)
File Sharing and the Coupling Facility (see page 28)
CA Top Secret CACHE (see page 28)
Security File Cache (see page 30)
Tuning CPF (see page 32)
CICS Performance Considerations (see page 33)
Shared Profile Table (see page 34)
CICS / DB2 Considerations (see page 34)

Performance Related Statistics

To expedite a resolution for performance related issues, CA requests you automatically issue the TSS MODIFY(STATS) command on an hourly basis.

The output of the command lists the current statistics (at the time the command was issued):

```
INIT    Job initiations validated.
XREQ    Cross memory requests processed.
SMF     SMF security records logged.
CHNG    Number of changes made to the security file.
AUD     Number of Audited events recorded in the ATS.
READ    Number of Secfile reads.
WRIT    Number of Security file writes
#REQ    Number of Requests pending execution
HWM     High water mark is the highest amount reached for waits on the queue.
LOCK    Number of times lock was requested
LWT     Number of times we couldn’t get the lock
Divide LWT by LOCK = Percentage of time waiting for the lock.
```

In addition, CA will want to see RMF Reports against the device the security file is on. Benchmark several transactions for response time and volume trends. Usually, performance problems creep up because of some added workload or system configuration change. Having a significant history of benchmark data available will:

- Predict and avoid performance problems before they occur
- Expedite resolving performance issue

Statistics Gathering

You can set up CA Top Secret to gather statistics for the:

- Sysplex coupling facility
- Cache facility
- Command processor
- Command processor workload balance
- Command Propagation Facility (CPF)
- Input/Output
- SAF RACROUTE calls
- Seccache facility

The collected statistics are written to SMF at defined time intervals. You can then:

- Use the TSSRPTSG report to display the saved statistics
- Specify a pre-allocated dataset to browse the statistics without running the report

To collect statistics

1. Enter the command:

   ```
   TSS MODIFY STATSLOG(DSNAME)
   ```

   DSNAME

   Specifies the name of a pre-allocated dataset statistics are written to. The dataset must with the format of RECFM=FB, LRECL=100, DSORG=PS.

   Default: SMF

2. Enter the command:

   ```
   TSS MODIFY STATREC(statrec)
   ```

   STATREC

   Specifies the statistics to be collected. Valid entries are CACHE, CPF, RACROUTE, SYSPLEX, COMMAND, WORKLOAD, IOSTATS, SECCACHE, and ALL.

   Default: ALL

3. Enter the command:

   ```
   TSS MODIFY STATGINT(nn)
   ```

   NN

   Specifies the time interval (in minutes) where statistics are gathered and SMF records are created.

   Default: 15 minutes

4. Enter the command:

   ```
   TSS MODIFY STATG(ON)
   ```

   Statistics gathering is activated.

Upon de-activation, a final statistics update occurs. To deactivate statistics gathering, enter the command:

```
TSS MODIFY STATG(OFF)
```

CPF Statistics

CPF related event statistics are collected for the number of:

- Inbound command requests for a particular node
- Inbound password requests for a particular node
- Outbound command requests for a particular node
- Outbound password requests for a particular node
- Returned outbound requests for a particular node

To display CPF related event statistics, enter:

```
TSS MODIFY(STATS)
```

Note: CPF Statistics are displayed only if CPF is active and the control option STATG(ON) is specified.

Security File Location

Isolate the security file on a device with nothing else on the disk. This device should be high performance DASD. Fill the remaining DASD space on the volume with a dummy dataset to prevent someone with access to the volume utilizing the free space. Protect this data from access. If isolating the security file is not an option, make sure that any other data placed on the device is not utilized during prime time business hours.

Recovery File Location

The recovery file should not be put on the same volume as the system catalogs or any other system files. A reserve on volume could cause a “system hang” situation resulting from a deadly embrace. This is also true for the audit file.

Avoid putting the RECOVERY file on the same spindle as the SECFILE. If you lose the single device, you will lose both of these files.

CA Top Secret Address Space

The CA Top Secret address space acts as a file server for other address spaces to access the security file. This includes address spaces beyond the z/OS operating system such as LINUX. The CA Top Secret database is accessible, both inbound and outbound, from any LDAP compliant directory.

Communication between address spaces uses CSA and ECSA. The amount of memory utilized is dependent on how you configure CA Top Secret.

Data Integrity

Data integrity must be maintained in a multi-CPU environment. The security file and ATF serialize using the same logic. This logic involves the use of a LOCK record that is stored in the SECFILE. The LOCK record assures that the updating system has complete control of the SECFILE. CA Top Secret does not use a MVS hardware reserve. The Lock record is stored in HDR1 (Header Record One) of the Security File.

Sharing Considerations

Best performance from the security file is achieved when SHRFILE(NO) is set in the CA Top Secret parm file.

Important! Only set SHRFILE(NO) when the security file and audit tracking file are not shared among systems. SHRFILE(NO) causes CA Top Secret to obtain the LOCK record from HDR1 and place it in memory until CA Top Secret is quiesced. With the lock record in memory, CA Top Secret will not need to obtain the lock record before accessing the security file. This could reduce I/O to the security file by as much as 66 percent.

ACID Index Performance Setting

The CA Top Secret ACID index is a directory of the users defined on the security file. It identifies the location of the security record on the SECFILE for each ACID defined to the database. When CA Top Secret initializes, the index is read into memory. Any time an ACID record is updated using an administration command, the index is refreshed.

The AINDXPER, activated via the SHRFILE control option, provides a file I/O performance improvement when the security file is shared between multiple systems and security administration is being done.

For example, if three systems are sharing a security file and the administrator changes or creates a new ACID, it causes the index on that system to be updated both in core and on the file and to set flags that the file has changed. When the other systems attempt to access this file, if AINDXPER is not set, it will have to read the entire index. For larger systems, CA Top Secret reads one or two tracks rather than several cylinders.

File Sharing and the Coupling Facility

The coupling facility is part of the SYSPLEX technology. CA Top Secret allows you to store all or a portion of the security file in an XES structure. Performance is improved because the coupling facility is faster than DASD access.

CA Top Secret OPTIONS(61), allows maintaining the security file lock record in coupling facility. Instead of having to go out to the SECFILE to read the lock record, CA Top Secret uses the faster access speed available with the coupling facility.

CA Top Secret CACHE

The CA Top Secret CACHE reduces I/O against the security file and increases system performance. The CACHE control option can be modified on the fly or through a permanent update in CA Top Secret PARMS. This includes setting the cache size, turning the cache off, clearing the cache, and listing out the cache statistics.

CACHE and CF Size

To determine the approximate size for the CACHE and CF XES structure use the TSSFAR UTILITY. When TSSFAR is run with the SFSTATS parameter, the recommended sizes for both the CACHE and XES is listed.

You may need to adjust the size. For instance, if message TSS1301I CACHE HAS BEEN CLEARED occurs too often, increase the cache size.

To determine if the CACHE has cleared, issue a TSS MODIFY CACHE(STATUS). The output line TSS1307I CLEARED (nnnnn) increments once each time the cache is cleared. This value is cleared when you recycle CA Top Secret.

The coupling facility XES structure gets cleared when the structure becomes 95% full. A message is sent to the console when the structure hits 75% full. The message is resent every additional 5%.

Security File Cache

SECCACHE provides a cache for CA Top Secret security records that reflect the status of a user following a RACROUTE VERIFY request. The cache is managed in a common data space that can be accessed from all address spaces.

Use SECCACHE to:

- Reduce the CPU cycles required to process RACROUTE VERIFY requests in both the user and security manager address spaces. In many cases, multiple RACROUTE VERIFY requests are performed to complete a single user logon. After the first RACROUTE VERIFY request is cached, subsequent requests by the same user in the same or different address space do not queue the security manager address space.
- Reduce file I/O in an environment where the security file is shared between multiple systems.
- Generate a comprehensive status display. Use the status display run-time information to tune SECCACHE by balancing the commitment of virtual storage with optimal system performance.

To recover unused space, SECCACHE record entries can be set to automatically expire at regular intervals or they can be force purged by command. The longer a record remains in SECCACHE, the more future requests it can handle. The percentage of requests satisfied by SECCACHE indicates how well the SECCACHE is performing.

Once activated, the SECCACHE data space remains intact during a temporary shutdown and restart of the security manager address space. This preserves the security records already cached.

Use the status display to monitor both your data and index area usage. Set a threshold full warning level during initialization that automatically attempts to recover space by clearing expired entries. The warning level applies to both the data and index areas independently. If the automated recovery process cannot recover sufficient space a warning message is sent to the operator console and remains there until sufficient space is recovered, the SECCACHE is deactivated, or the security address space is terminated. Records can be added to SECCACHE until the areas become 100% full.

The SECCACHE control option can be modified online or through permanent update in the CA Top Secret PARMS. This includes setting the SECCACHE initialization options, turning SECCACHE off, clearing records from SECCACHE, and listing SECCACHE statistics.

The caching of security records in an data space improves system performance by re-using the security records on subsequent user log ons.

Activate the Cache

To activate the security record cache, enter:

```
TSS MODIFY SECCACHE(nnnn)
```

nnnn

Specifies the number of megabytes of available memory allocated as a data space for security record caching.

Maximum: 2 gigabytes

To deactivate the security record cache, enter:

```
TSS MODIFY SECCACHE(OFF)
```

Security File Cache Size

SECCACHE requires virtual storage in the form of a common data space. The maximum size of the SECCACHE is 2 gigabytes. Start with less and use the status display to tune the size. To estimate the optimum size for SECCACHE use the formula:

```
SECCACHE Size = 1.01 * (ACID#(AVGRECSZ + 16))
```

ACID#

The number of *user* ACIDs that "log on".

AVGRECSZ

The average record size from the SECCACHE(STATUS) display.

Example: calculating SECCACHE size

In this example, the number of ACIDs is 25,000 and the average record size is 4200:

```
1.01 * (25,000(4200 + 16 )) = 102 megabyte
```

SECCACHE in a Shared Security File Environment

In a shared security file environment, it is important that policy changes made on one system are reflected within any in-core processing tables on all remote systems as soon as possible. This is accomplished internally when a security file I/O is performed on the remote system, for example when a user logs on.

The SECCACHE control option eliminates much of the security file I/O associated with a user log on event. To improve the synchronization an internal processing event performs any in-core table refresh, if required, at the end of two TIMER cycles. This keeps the tables up to date even though the security file has not been accessed. You can manually synchronize the in-core tables with the MODIFY SYNCH control option. This allows you to immediately see the effects of remote changes to selected SDT records with the TSS LIST command.

Tuning CPF

The Command Propagation Facility (CPF) provides synchronization of Security Files across the enterprise. This lets you take advantage of the performance benefit you get when not sharing the Security File.

Running with the SHRFILE control option set to NO eliminates lock file contention.

Notes:

- Digital Certificate related commands are not CPF.
- CPF is designed to synchronize normal processing. It is *not* meant for massive command updates. Massive updates should be run in batch mode using TARGET (LOCAL).

CICS Performance Considerations

The facility control option LOCKTIME assigns the amount of time in minutes after which a terminal connected to a specific facility will lock if CA Top Secret does not detect activity.

When running with LOCKTIME active and LTLOGOFF set to NO, the USER remains active in the CICS region taking up CICS resources.

- LOCKTIME with LTLOGOFF set to YES, causes the user to be signed off
- PCLOCK parm on the facility controls the handling of the locktime prompting
- PCLOCK= YES the prompting is Pseudo conversational
- PCLOCK=NO the prompting is conversational

The recommended setting for PCLOCK is:

```
TOR - PCLock = YES
AOR/FOR = PCLock = NO - eliminates exit point
```

Set your TOR to PCLOCK = YES. This ensures that the TOR locktime is Pseudo conversational. Since the AOR and FOR types are not terminals, we recommend PCLOCK = NO. This eliminates the exit point needed to manage lock time.

Conversational CICS TCA slot is tied up until password is entered Pseudo conversational does not tie up CICS resources waiting for a reply. Replace with an CA Top Secret task freeing up the CICS TCA slot.

CICS Signon Sub-Tasking

CICS signons are sub tasked. Five signon subtasks are allocated by default. CA Top Secret allocates five subtasks when the CICS facility is initialized.

To determine the maximum number of signon subtasks controlled by CA Top Secret facility parms setting use:

```
MAXSIGN (10, retry) - facility defaults
```

Extra subtasks (6 through 10) are allocated as needed automatically until the MAXSIGN is hit. Specifying RETRY means that Signon/Signoff requests that exceed threshold are requeued.

Notes:

- Do not set MAXSIGN to KILL since this can cause unwanted system dumps.
- On average, each signon subtask takes up to 40K of High Private below the line for MVS control block.

Shared Profile Table

The shared profile table allows CA Top Secret profile sharing in multi-user address space environments such as CICS and IMS. CA Top Secret validates authorization checks without leaving the region's address space. PROFILE sharing is only available at the facility level. To activate or deactivate this option, specify the facility control option SHRPRF or NOSHRPRF.

The number specified in the PRFT Facility Option is in PAGES. Each page holds 256 profiles that are stored in above the line storage storage.

Limit profile administration to off-peak hours for profiles you expect to be shared. Excessive administration for profiles in the table during peak use of the facility can result in having multiple copies of the same profiles stored in the table. The following message is posted when profile table is full:

TSS0960E SHARE PROFILE TABLE FULL – jobname

For example, a region supporting 250 users, each sharing three common profiles, where each user also has one unique profile, must have a shared profile table with no less than 253 entries: PRFT=1.

The default is PRFT=3. It supports profile sharing of up to 768 unique, active profiles within the region. If this value is changed using the TSS MODIFY command, the region must be recycled for the change to take effect.

CICS / DB2 Considerations

The userid used when CICS accesses DB2 is determined by the setting of AUTHTYPE parm in the RCT or CSD:

- AUTHTYPE = USERID, DB2 is passed from CICS userid only and must issue the SAF call to sign on the user.
- AUTHTYPE = GROUP, DB2 is passed on to both the userid and a pointer to the ACEE which DB2 can use without issuing a signon.

The DB2 subsystem has two CICS exit points for authorization routines:

- DSN3@SGN is for sign-on processing
- DSN3@ATH is for connection processing

Chapter 3: Using the TSSSIM Utility

This section contains the following topics:

Introduction to TSSSIM (see page 35)
Authority and Scope (see page 36)
TSSSIM Under TSO in a Non-SPF Environment (see page 36)
Use TSSSIM Under TSO in an SPF Environment (see page 37)
TSSSIM in TSO/E-Background, TSO/E-Foreground, or TSO/E-Batch (see page 39)
Execute the Simulator in a Batch Environment (see page 40)
TSSSIM and CA Roscoe (see page 41)
(TSO) Invoke a Simulation Session to Test Security File Permissions (see page 41)
LOGOFF/SIGNOFF (see page 43)
HELP Command—List and Describe Commands (see page 43)
ENVIRON Command—Change Simulated Environment (see page 43)
END/QUIT Command—Exit Simulator (see page 44)
Perform a Simulated Resource Check (see page 44)
TSSSIM Special Commands (see page 48)

Introduction to TSSSIM

Use TSSSIM to test permissions on the Security File without affecting the production environment. For example simulate FAIL mode processing while the system is in DORMANT mode.

Testing of the security permissions consists of invoking a simulation resource command (or one of the simulator's TSO/SPF panels) for the desired resource. TSSSIM reports 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.

The simulator can "debug" errors in the Security File permissions. Often, when a user has several profiles attached to his ACID, it is difficult to isolate which permission has allowed or denied access to a 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 be executed under:

- TSO (both SPF and non-SPF environments)
- BATCH
- CA-Roscoe®

Authority and Scope

TSSSIM is available for administrators with the authority:

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

A user with no administrative authority may use TSSSIM if given USE access to entity TSSUTILITY.TSSSIM in the CASECAUT resource class. This access may be granted by an administrator using the following command:

```
TSS PERMIT(user) CASECAUT(TSSUTILITY.TSSSIM) ACCESS(USE)
```

A simulated signon can be initiated for any ACID in the Security File, including those not within the administrator's scope.

If an ACID:

- Is within the administrator's scope, he may issue resource checks against any resource in the Security File
- Is not within the administrator's scope, he may issue resource checks against only those resources within his scope

TSSSIM Under TSO in a Non-SPF Environment

Under TSO in a non-SPF environment, TSSSIM is exclusively command-driven. You can perform the following actions:

- Invoke a command to conduct the desired resource check, invoke a command to gather information about the simulator, or invoke commands to modify tables within CA Top Secret.
- Invoke the TSS administration command from directly within the simulator.

Note: When you run TSSSIM in a non-SPF environment, single TSSSIM commands can be up to 512 characters.

To receive all messages from TSSSIM at the TSO session terminal, the TSO session PROFILE WTPMSG must be set prior to initiating TSSSIM. If PROFILE NOWTPMSG is set, messages may be routed to the operator console.

Use TSSSIM Under TSO in an SPF Environment

You can use TSSSIM under TSO in an SPF environment.

The following restrictions apply:

- The LOGON, LOGOFF, QUIT, END, and ENVIRON commands are not honored.
- To change the environment of a signed-on simulated ACID, return to the logon panel and specify the ACID name and facility.
- On all panels after the logon panel, the FACILITY field must be clear before you press Enter.

Follow these steps:

1. Enter the TSSSIM option on the OPTION line of the SPF main menu.

   The logon panel appears.

2. Complete the logon panel:

   a. Specify required logon information in the ACIDNAME and Facility fields.

   b. (Optional) Specify values for the other fields.

   Press Enter to attempt the logon.

   TSSSIM displays a message describing the status of the session.

3. Press the PF3 key to clear the message.

   The CA Top Secret Resource Selection List panel appears. This panel displays resource classes on which the simulator can perform a security check. You can scroll the resource class list if needed.

4. Enter **S** next to a resource to display an associated resource class panel where you can complete fields and perform the security check.

   To display more information about any resource class panel, enter **HELP** on the OPTION line of the panel. (To return to the original panel, press the PF3 key.)

   Special commands (EJECT, STATUS, and TSS) generate output to an ISPF table. This process is useful for commands (such as TSS LIST) that generate lengthy output, because the output resides in a normal ISPF table. Pressing the PF3 key destroys the output and returns control to the previously displayed panel.

   Note: CA Top Secret recognizes special commands only when you enter the commands on the COMMAND line from one of the panels. The commands do not work on COMMAND lines within a panel that was triggered by another command.

TSSSIM TSO Online Tutorial

To use the tutorial for TSSSIM, enter 16 on the CA Top Secret Selection Menu.

On the OPTION line of the Logon panel, enter HELP. A menu with all the topics contained in the tutorial is displayed.

Note: Each resource is documented on a separate panel.

TSSSIM in TSO/E-Background, TSO/E-Foreground, or TSO/E-Batch

If you run TSSSIM in TSO/E-Native mode and TSO/E-ISPF mode under TSO/E-Foreground, the output is the same in each mode.

To invoke TSSSIM in TSO/E-Background and produce output, use the following JCL:

```
//stepname EXEC PGM=IKJEFT01
//SYSIN    DD DUMMY
//SYSTSPRT DD SYSOUT=A
//SIM$$OUT DD SYSOUT=A
//SYSTSIN  DD *
  TSSSIM
//SIM$$IN  DD *
  (TSSSIM commands)
//
```

Note: When you invoke TSSSIM through JCL, single TSSSIM commands cannot exceed 80 characters. To run a simulation command that is longer than 80 characters, use TSSSIM under a non-SPF environment.

You can also invoke the simulator in a TSO/E-Batch environment. For example, you can use the following JCL to invoke the simulator in a TSO/E-Batch environment to perform resource checking for data set SYS1.LINKLIB with or without the ACCESS parameter:

```
//stepname EXEC PGM=IKJEFT01
//SYSIN    DD DUMMY
//SYSTSPRT DD SYSOUT=A
//SIM$$OUT DD SYSOUT=A
//SYSTSIN  DD *
  TSSSIM
//SIM$$IN  DD *
  LOGON ACID(USER01) FAC(TSO) TRACE
  $DSN('SYS1.LINKLIB')
  $DSN('SYS1.LINKLIB') ACC(UPDATE)
  $DSN('SYS1.LINKLIB') ACC(UPDATE) PRIVPGM(IEBCOPY)
  END
//
```

More information:

TSSSIM Under TSO in a Non-SPF Environment (see page 36)

Execute the Simulator in a Batch Environment

Instead of entering commands each time the test is run, you can save the commands and then submit batch tests when required.

The format of the commands follows the general simulator command format.

All commands are supported in the batch environment, although the simulator trace information is condensed. The EJECT command is a batch-only command and causes a page eject before the simulator executes the next command.

To execute the simulator in a batch environment, use the following JCL:

```
//SIMULATE        EXEC         PGM=TSSSIM
//SIM$$LOG        DD           SYSOUT=A
//SIM$$IN         DD *
                    .
                    .
                    .
              (TSSSIM commands)
/*
//
```

Note: When you invoke TSSSIM through JCL, single TSSSIM commands cannot exceed 80 characters. To run a simulation command that is longer than 80 characters, use TSSSIM under a non-SPF environment.

Example: Invoke the Simulator in a Batch Environment

This example invokes the simulator in a batch environment to perform resource checking on data set SYS1.PROCLIB with UPDATE access. The data set can be accessed only through privileged program IEBUPDTE. This example also checks whether USER01 has CREATE access on the DASD volume SYSRES. The administrator signs on with the simulated ACID USER01 in a TSO facility. The TRACE function is also activated.

```
//SIMULATE        EXEC           PGM=TSSSIM
//SIM$$LOG        DD             SYSOUT=A
//SIM$$IN         DD *
LOGON ACID(USER01) FACILITY(TSO) TRACE
$DSN('SYS1.PROCLIB') ACC(UPDATE) PRIVPGM(IEBUPDTE)
$DASDVOL(SYSRES) ACC(CREATE)
END
/*
```

More information:

EJECT Command—Page Eject (see page 48)

TSSSIM and CA Roscoe

To access the security simulator while signed on to CA Roscoe, invoke the CA Roscoe monitor command SIM. TSSSIM under CA Roscoe is exclusively a command-driven utility. Usually both input and output are directed at the terminal; however, alternate input sources and output destinations are:

SIM

Input read from the terminal, output directed to the terminal.

SIM-I

Input read from the AWS, output directed to the terminal.

(TSO) Invoke a Simulation Session to Test Security File Permissions

In TSO (non-SPF environment), you specify the LOGON command to invoke the simulator. This command uses security macro RACROUTE REQUEST=VERIFY.

Follow these steps:

1. Enter TSSSIM at the READY prompt to invoke the simulator.
2. Entering the LOGON command to begin a simulation session, specifying at least a user ACID and facility, as shown in the following syntax:

   ```
   LOGON ACID(acid) FACILITY(facility_name)
   ```

   TSSSIM returns a message that a successful simulated session has been established. If there are any restrictions, the simulator returns a message accordingly.

You can use the following parameters with the LOGON command:

ACID(*acid*)

Specifies the ACID whose session you will simulate with TSSSIM. A password is not used with a TSSSIM signon.

CPU

Specifies any four-character SMF CPU ID.

Default: CPU to which the administrator is currently signed on

FACILITY(*facility_name*)

Specifies the facility that you are simulating for your TSSSIM session. To test maskable resources with TSSSIM, the facility name must have the RES suboption; otherwise, the simulation might not accurately reflect the permissions for maskable resources.

MODE

Specifies the security mode in which the simulated session operates.

Options: DORM, WARN, IMPL, or FAIL

Default: FAIL

PRIVPGM

Specifies any valid z/OS program name. This program is automatically passed to all simulated resource commands to simulate proper PRIVPGM restrictions unless explicitly overridden on individual simulated resource commands.

QUALIFIER

Specifies any high-level qualifier to be issued as part of the data set name. This qualifier is automatically passed as part of the data set name for all issued resource checks unless specifically overridden on individual resource commands.

SVC

Specifies the valid SVC name that must be among the list of SVC names allowed by TSSSIM. This SVC is automatically passed as the SVC in control for all resource checks unless explicitly overridden on individual resource commands.

Default: OPEN

TERMINAL

Specifies any valid VTAM or TCAM network terminal ID.

Default: Terminal to which the administrator is currently signed on

TRACE

Controls the simulation trace facility and can be specified as TRACE or NOTRACE. When activated, each resource check will pinpoint the exact reason for resource access or denial.

Default: NOTRACE

During a simulation, you can override some of the parameters that were specified during logon. For example, you can include a PRIVPGM specification on many resource commands that overrides the PRIVPGM logon parameter (which is useful when checking security permissions that use program pathing).

Example: Invoke a Simulation Session and Simulate a TSO Facility for Your Session

After invoking the Security Simulation Facility by entering TSSSIM at the READY prompt, an administrator logs on with the simulated ACID in a TSO facility:

```
LOGON ACID(MYACID) FACILITY(TSO)
```

LOGOFF/SIGNOFF

Use this command to terminate a current simulated session.

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.

Example: LOGOFF command

This example ends a simulated session:

```
LOGOFF
```

HELP Command—List and Describe Commands

Use this command to generate a list of all commands and a brief description of each.

When accompanied by a command, HELP gives a brief description of the command as well as its command class, attributes, and qualifying parameters.

Examples: HELP command

This example lists all TSSSIM commands:

```
HELP
```

This example lists information about the $DSN command:

```
HELP $DSN
```

ENVIRON Command—Change Simulated Environment

Use this command to change the simulated ACID environment.

This command uses the:parameters CPU, MODE, OWN, PRIVPGM, SVC, TERMINAL, TRACE

Example: ENVIRON Command

This example changes the environment of the simulated ACID to WARN mode and put a trace on all resource checks:

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

END/QUIT Command—Exit Simulator

Use this command to exit the Security Simulator Facility.

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, terminates the session.

Example: END command

This example exits the simulator:

```
END
```

Perform a Simulated Resource Check

Simulated resource commands allow the administrator to choose the type (or class) of resource check. Resource checks for the specific resource are passed to CA Top Secret on behalf of the simulated ACID. Invoking simulation resource commands allows the administrator to test permissions on the security file without affecting the production environment.

Follow these steps:

1. Log on to TSSSIM.
2. (Optional) View the resource classes (and commands) that are available for simulation:
   - (non-SPF) Enter **HELP** on the TSSSIM command line.
   - (SPF) View the resource class list on the Resource Selection List panel.
   - (Batch) Include HELP in the JCL, then view the output after execution.
   - (CA Roscoe) Enter **HELP** at the appropriate input source for the environment (terminal or AWS).

   The SPF environment includes predefined classes ($ prefix) and user-defined classes (@ prefix) in the list of available classes; however, the output for most other environments does *not* include user-defined classes.

   Note: If you have authority to list the contents of the RDT, you can issue the TSS LIST(RDT) command to see output that shows predefined *and* user-defined resource classes.

3. (Optional) Access details for a resource class:

   - (non-SPF) Enter **HELP** *resource_class_name* on the TSSSIM command line.
   - (SPF) Select the resource class on the Resource Selection List panel, then enter **HELP** on the command line of the Resource Simulation Panel for the resource.
   - (Batch) Include **HELP** *resource_class_name* in the JCL, then view the output after execution.
   - (CA Roscoe) Enter **HELP** *resource_class_name* at the appropriate input source for the environment (terminal or AWS).

   The details include the qualifying parameters that are available for the resource class name.

4. (Optional) Determine which access levels and maximum name length are permitted for a resource:

   a. Issue the following command:

   ```
   TSS LIST(RDT) RESCLASS(resource_class_name)
   ```

   b. Review the MAXPERMIT field (for maximum name length) and ACCESS field (for access level permissions).

5. Perform a resource check through one of the following actions:

   - (Predefined resource class) Issue the following command (including applicable qualifying parameters if necessary):

     ```
     $resource_class_name(resource_entity_name) [qualifying_parameter_list]
     ```

   - (User-defined resource class) Issue the following command (including applicable qualifying parameters if necessary):

     ```
     @resource_class_name(resource_entity_name) [qualifying_parameter_list]
     ```

   Qualifying parameters are as follows.

   Note: To see the qualifying parameters that are applicable for a resource class, access the detailed help for the resource.

   ACCESS

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

   LIBRARY

   Specifies the library in which a privileged program must reside.

   NEWDSN

   Simulates a DADSM RENAME function for data sets.

   Important! To use this parameter, you must have SVC set to RENAME.

   Any rename involves authorization checks for the “old” data set ($DSN) and the “new” data set (NEWDSN). The old data set must have ACCESS(READ,SCRATCH); the new data set must have ACCESS(WRITE,CREATE).

OWN

Specifies that CA Top Secret should assume the resource is owned and not check the Global Resource Table to see if the resource entity (or prefix) is defined.

Note: If the resource is not owned, access is denied (*regardless* of whether OWN is specified).

Default: NOOWN

PRIVPGM

Specifies the program that is in control when the resource check occurs.

SVC

Specifies the SVC that is in control when the resource check occurs. Values are as follows:

- ALLOCATE
- CATALOG
- CREATE
- FEOV
- OPEN
- RENAME
- SCRATCH

Default: OPEN

TRACE

Specifies to enable the trace feature to locate the exact permission that is causing the resource access or denial.

Default: NOTRACE

VOLUME

Specifies the volume on which the data set resides.

Note: A $DSN simulation with no VOLUME operand can produce unpredictable results.

XACCESS

Specifies a two-byte hexadecimal code equating to a specific access level or levels. This specification overrides the default access value that is used with the specified SVC name. Using XACCESS is convenient if you need to check a combination of access levels. For example, in an SPF environment, you cannot enter more than one access level with the ACCESS keyword.

Example: (Non-SPF Environment) Perform a Simulated Resource Check

This example shows how to perform a simulated resource check in a non-SPF environment:

After you log on to TSSSIM, enter HELP on the TSSSIM command line to produce a list of available commands, such as the commands shown in the following sample excerpt:

```
COMMAND DESCRIPTION

---------        ------------------------------------
$ABS             RES CHECK - ABSTRACT RESOURCES
$ABSTRACT        RES CHECK - ABSTRACT RESOURCES
$ACID            RES CHECK - ACID JOB SUBMISSION
$ALT-ACID        RES CHECK - ACID JOB SUBMISSION
$APPCLU          RES CHECK - APPC LOGICAL UNITS
$APPCPORT        RES CHECK - APPC PORT OF ENTRY
$APPCSI          RES CHECK - APPC SIDE INFORMATION
```

Enter HELP $DSN at the TSSSIM command line to produce the following detailed help for the $DSN resource class:

```
COMMAND NAME   =  $DSN
DESCRIPTION    =  RES CHECK - OS DATASETS
COMMAND CLASS  =  DATASET     (00C4)
RESCLASS-INDEX =  PIE,SIMULATIVE  (RV#=04)
ATTRIBUTES     =  LOGON-REQUIRED
PARAMETERS     =  ACCESS,NEWDSN
PARAMETERS     =  PRIVPGM,LIBRARY,SVC,TRACE,VOLUME,XACCESS
```

You can then perform a resource check on a data set named ACCTPAY.MASTER with UPDATE access:

```
$DSN('ACCTPAY.MASTER') VOLUME(ACCTP1) ACCESS(UPDATE)
```

Example: Use NEWDSN to Simulate a DADSM RENAME Function for a Data Set Name

This example uses NEWDSN to simulate a DADSM RENAME function for a z/OS data set name (with SVC set to RENAME):

```
$DSN('ACCTPAY.MASTER') NEWDSN('UPAY.MASTER') VOLUME(ACCTP1) SVC(RENAME)
```

The rename involves authorization checks for the “old” data set ($DSN) and the “new” data set (NEWDSN). The ACCTPAY.MASTER data set has the required access level of ACCESS(READ,SCRATCH); the UPAY.MASTER data set has the required access level of ACCESS(WRITE,CREATE).

Note: For other $DSN accesses, the ACCESS level is specified without reference to a specific SVC.

Example: Use OWN to Issue a Security Check for an IUCV Communication Target

(Valid on z/VM only) This example checks on an IUCV connection to virtual machine APP17 and specifies the OWN parameter so that CA Top Secret assumes the resource is owned:

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

IUCV communication targets control the ability of users to issue IUCV connection to the virtual machine designated by the resource name.

Example: Use PRIVPGM to Issue a Security Check for a CICS Destination Control Table (DCT)

This example issues a security check on DCT PRT6 through the privileged program ACDCT5 that resides in program library PRIVPROG.LIB:

```
$DCT(PRT6) PRIVPGM(ACDCT5) LIBRARY(PRIVPROG.LIB)
```

Example: Use TRACE to Perform a Security Check on an IMS Application

This example performs a security check on an IMS application named TEMPAY and activates the TRACE facility to locate the exact permission:

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

Example: Use XACCESS to Issue a Security Check for an IMS Database Descriptor (DBD) Name

This example checks on an IMS DBD (TSTPDA) and issues an XACCESS of 88, which simulates access levels of UPDATE and DELETE:

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

TSSSIM Special Commands

The special simulator commands perform no resource checks. They provide the administrator with information about the simulator within CA Top Secret.

EJECT Command—Page Eject

Use this batch-only command to cause a page eject before the next command is executed by the simulator.

Note: These commands are entered as control statements in the JCL when executing TSSSIM in batch. For information, see Using TSSSIM/Batch.

STATUS Command—Current Environment's Status

Use this command to provide the current status of the simulation environment:

- 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

Example: STATUS command

This example views the current status of the logged on simulated ACID:

```
STATU
```

TSS Command—Performing Security Administration Functions

This command is not executed by the simulator. It is 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.

This command uses TSS command functions and parameters.

Example: TSS command

This example views information on a profile:

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

Chapter 4: Using the TSSFAR Utility

This section contains the following topics:

About the TSSFAR Utility (see page 51)
Required Authority (see page 51)
Set Up MEMLIMIT (see page 52)
TSSFAR JCL (see page 53)
TSSFAR Control Statements (see page 53)

About the TSSFAR Utility

Use the File Analysis Routine (TSSFAR) to review the permissions and assignments recorded in the Security File. The type of security information displayed by TSSFAR depends upon the control statements selected.

TSSFAR is used to:

- 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

Important! The file TSSFAR executes on is continuously accessed for the duration of the job. Running TSSFAR against the Security File can cause degradation to system performance. We recommend that TSSFAR always run against a Backup File at the direction of CA support staff.

Required Authority

The following users may run TSSFAR:

- The MSCA
- A user with no administrative authority may use TSSFAR if given USE access to entity TSSUTILITY.TSSFAR in the CASECAUT resource class.

  This access may be granted by an administrator using the following command:

  ```
  TSS PERMIT(user) CASECAUT(TSSUTILITY.TSSFAR) ACCESS(USE)
  ```

Set Up MEMLIMIT

The IARV64 macro obtains and releases the 64 bit memory object. For this macro to work:

- You must have z/OS r1.6 or higher
- The MEMLIMIT keyword must be increased in either in the SMFPRMxx member during IPL time or on the jobcard of the JCL running TSSFAR

To increase MEMLIMIT, add the the following line to SYS1.PARMLIB(SMFPRM*xx*):

```
MEMLIMIT(4G)
```

Example: Setting up MEMLIMIT

This sample sets MEMLIMIT to 4Gb:

```
ACTIVE                    /*ACTIVE SMF RECORDING*/
DSNAME(SYS1.MANx)         /* DATA SET */
NOPROMPT                  /*DO NOT PROMPT OPERATOR FOR OPTIONS*/
REC(PERM)                 /*TYPE 17 PERM RECORDS ONLY*/
BUFNUM(4,9)               /* 4 - 4096 BUFFERS ALWAYS AND
                             ALLOW UP TO 9 BEFORE SUSPENDING
                             A USER FOR BUFFER SHORTAGE*/
MAXDORM(3000)             /* WRITE AN IDLE BUFFER AFTER 30 MIN*/
STATUS(010000)            /* WRITE SMF STATS AFTER 1 HOUR*/
JWT(0300)                 /* 522 AFTER 60 MINUTES*/
SID(XExx)                 /* SYSTEM ID*/
MEMLIMIT(4G)              /* 64 BIT STORAGE */
SYS(TYPE(0:255),EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,
    IEFUSI,IEFUJI,IEFUTL,IEFU29),NOINTERVAL,NODETAIL)
```

TSSFAR JCL

Use the following sample JCL to run TSSFAR:

```
//TSSFAR JOB USER=msca,CLASS=A
//FAR EXEC PGM=TSSFAR
//SECFILE DD DSN=top.secret.backup.file,DISP=SHR
//
//SYSPRINT DD SYSOUT=*
//INPUT DD *
KEY=ENCRYPTION KEY
HEADER
ARLBMAP
ALLOC
ACIDCHAN
ACIDLINK
RESINDEX
ACIDRES
RESOURCE
SFSTATS
WHOHAS (resource owning ACID)
/*
```

TSSFAR Control Statements

The following selection criteria are used in generating TSSFAR reports:

KEY

(Required) Displays customer encryption key in 16-byte hexadecimal or 8 EBCDIC characters:

KEY=*hhhhhhhhhhhhhhhh*|'*cccccc*

ARLBMAP

(Optional) Prints the ARLB allocation map from the header record.

ALLOC

(Optional) Prints a cross-reference of all mismatches between 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. Exceptions exist if an ARLB is chained and the first byte was used to add an x'00' to the end of an extension element in the ACID record. This setup makes the key appear to be wrong because the key is allocated but the ARLB is empty. These exceptions are resolved by using the ACIDCHAN function.

ACIDCHAN

(Optional) Analyzes ARLB logical record chains and reports empty chains. ACIDCHAN performs the following activities:

- Runs the ALLOC function.
- Reads all ARLB logical records that are connected (chained) to the first ARLB for each ACID.
- Verifies that the actual number of ARLBs in the chain match what the ACID listed in its File Accessorid Record (FACTREC).
- Lists any ARLBs that are chained but empty. If this number of empty ALRBs matches the number of ARLBs that the ALLOC function reported as key errors, the allocation maps are correct.

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

ACIDLINK

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

ACIDSIZE[(*nnn*)]

(Optional) Reviews all ACIDs and reports any ACID whose record size reaches a certain percentage of the maximum allowed size.

nnn

Specifies a percentage of maximum allowed size. When the ACID record size reaches this threshold, notification occurs. If a value is not specified, the default value is 80 (80%).

HEADER

(Optional) Prints the first 256 bytes of the header record.

RESINDEX

(Optional) Verifies that all resource ownership indexes match the owning ACID.

ACIDRES

(Optional) Verifies that all resources claiming to be owned by an ACID have a correct matching owning ACID in the resource index.

RESOURCE

(Optional) Verifies that the following conditions exist:

- All permits have associated WHOHAS entries on the owning ACID.
- All WHOHAS information on the owning ACID has a corresponding permit.

Note: When you run this utility, CA Top Secret might respond with messages identifying permit, owner, or resource issues, because multiple reference pointers might not be up-to-date. These messages include as much information as is available in the security file.

The following message is a sample message:

```
Permit is not owned
ACID holding permit: userid1    Owning ACID: userid2
Resclass: DATASET
Resname: USER.DATASET.ONE
```

If the list includes discrepancies, you can perform the applicable CA Top Secret administration to correct each instance by revoking and repermitting the authorizations. If the TSSFAR utility flags too many discrepancies to fix through manual CA Top Secret administration, you can run the TSSXTEND utility with the WHOHAS option (see page 67).

SFSTATS

(Optional) Searches the security file and prints statistics for the following conditions:

- ACID index entries allocated
- ACID index entries defined
- Next available ACID number
- Last available ACID number
- ACID blocks allocated
- ACID blocks used

- Volume entries allocated

  % Used % Deleted

- PIE blocks allocated

  % Used % Deleted

- RES blocks allocated

  % Used % Deleted

- SDT blocks allocated

  % Used

- *XREF* ACIDs present/not present
- Features available on the SECFILE:
  - Large ACID support—YES/NO
  - Large ORG ACID support—YES/NO
  - RDT2BYTE support—YES/NO
  - New password support—YES/NO
  - SDT in VSAM support—YES/NO
  - DIGICERTS in VSAM support—YES/NO
  - AES encryption—YES/NO
  - Large VSAM records—YES/NO
- Version of the SECFILE—*x.x*
- CA Top Secret version when SECFILE was created—*xx.x*
- Recommended TSS cache size
- Recommended XES structure size
- Active ACID count Average size *xxxxxxx* bytes
- Number of SCAs
- Number of LSCAs
- Number of ZONEs
- Number of ZCAs
- Number of DIVs
- Number of VCAs
- Number of DEPTs
- Number of DCAs

- Number of USERs
- Number of PROFs
- Number of GROUPs

WHOHAS (*resource-owning ACID*)

(Optional) Lists all ACIDs that hold a permit to resources that are owned by the specified ACID. CA Top Secret permits only one WHOHAS control statement per run.

resource-owning ACID

Specifies an ACID that owns resources.

Chapter 5: TSSRECVR - Security File Recovery

This section contains the following topics:

About TSSRECVR (see page 59)
Recovery Procedure for Automatic Backup (see page 60)
Manual Recovery Procedure (see page 63)

About TSSRECVR

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

Security file recovery is performed by applying the TSSRECVR routines to the backup security file. An up-to-date security file is reconstructed by applying, from the recovery file to the backup security file, all the changes that occurred since the last backup of the security file.

Recovery Procedure for Automatic Backup

Use this procedure if the primary security file is lost or damaged and you have been using the automatic backup feature.

To recover the security file

1. Enter the command:

   ```
   STOP TSS
   ```

   CA Top Secret stops.

2. (Optional) If you are using the VSAM/r14 dataset and the security file is shared, modify and run CAIJCL member VSAMDEF7 to copy the VSAMBKUP file and build the alternate index and path files.

3. Enter the command:

   ```
   START TSSB
   ```

   CA Top Secret restarts.

   TSSB was created at installation and has the following characteristics:

   - The SECFILE DD statement points to the backup security file.
   - If using VSAM/r14 with a non shared security file, the VSAMFILE DD statement points to the backup VSAM file.
   - If using VSAM/r14 with a shared security file:
     - VSAMFILE DD statement points to the VSAMCOPY file created by VSAMDEF7
     - VSAMAIX DD statement points to the AIXCOPY file created by VSAMDEF.
     - VSMPATH DD statement points to the PATHCOPY file created by VSAMDEF7
   - Automatic backup is turned OFF (no BACKUP or VSAMBKUP DD statements).

   You now have a backup security file that is at most, 24 hours out-of-date.

4. If you are using your only copy of the backup security file and suspect that damage to the security file was caused by a command function update, make a copy of the backup security file with the TSSBCKUP JCL.

5. Enter the command:

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

   Recovery is turned off to avoid duplication of TSS command functions on the recovery file resulting from the recovery process.

6. Enter the command:

   ```
   START TSSRCVR1,DATA=DATE(yyddd) [,TIME=TIME(hhmm)]
   ```

   TSSRCVR1 executes.

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

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

   The above example will show up in the output of TSSRCVR1 as:

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

8. Enter the command:

   ```
   TSS ADDTO(STC) PROCNAME(TSSRCVR2) ACID(msca) STCACT
   ```

   TSSRCVR2 is run under the authority of the MSCA. This ensures that commands will not FAIL due to insufficient authority. The STCACT keyword is optional and has the effect of prompting the operator console for a userid and password when the procedure is started. Finally, this information is written to the Audit file.

9. Enter the command:

   ```
   START TSSRCVR2
   ```

   The changes collected in Step 5 are applied to the backup security file.

10. Use the backup security file to test that the file was properly recovered. Use the TSSB STC and ensure that Automatic Backup and RECOVER are OFF.

11. Enter the command:

    ```
    STOP TSS
    ```

    CA Top Secret stops.

12. If the primary security file still exists on DASD (damage was not a result of hardware problems), skip to Step 12; otherwise, a new security file must be initialized using the TSSMAINT utility. Besides the ID= parameter, the new primary security file that is created must have the same parameter values as the original security file. The id= parameter should be set to ID=PRIMARY.

13. Create a new started task modeled after the TSS STC in which the SECFILE DD statement points to the backup security file, and the BACKUP DD statement points to the primary security file. You can use the following TSSB JCL:

    ```
    //TSSB  PROC PARMS='sys1.parmlib',
    //           HL='CAI.TSSC0',
    //           PRINT='*'
    //*
    //*
    //* CA Top Secret SECURITY (TSS) STARTED TASK FOR USE
    //* DURING RECOVERY PROCEDURE ONLY
    //*
    //*
    //TSSB       EXEC   PGM=TSSMNGR4,DPRTY=(15,14),
    //                  TIME=1440,REGION=500K
    //SECFILE    DD     DISP=SHR,DSN=&HL..BACKUP
    //BACKUP     DD     DISP=SHR,DSN=&HL..SECFILE
    //VSAMFILE   DD     DISP=SHR,DSN=&HL..VSAMBKUP
    //VSAMBKUP   DD     DISP=SHR,DSN=&HL..VSAMFILE
    //RECFILE    DD     DISP=SHR,DSN=&HL..RECFILE
    //AUDIT      DD     DISP=SHR,DSN=&HL..AUDIT
    //PARMFILE   DD
      DISP=SHR,FREE=CLOSE,DSN=&PARMS(TSSPARM0).
    //AUTOCMDS   DD
      DISP=SHR,FREE=CLOSE,DSN=&PARMS(TSSAUTO0).
    //SYSUDUMP   DD     SYSOUT=&PRINT.
    //PEND
    ```

14. Enter the command:

    ```
    START TSSB
    ```

    CA Top Secret starts using the newly created procedure, TSSB.

15. Enter the command:

    ```
    F TSS,BACKUP
    ```

    An automatic backup is forced.

16. Enter the command:

    ```
    STOP TSS
    ```

    CA Top Secret stops.

CA Top Secret can be restarted using the TSS STC. The primary security file is recovered. Make sure that after restarting, RECOVER is ON.

Manual Recovery Procedure

If the security file is lost or damaged and your site is not using the CA Top Secret automatic backup feature you can do a manual recovery.

Note: For a manual recovery RECOVER(ON) must have been in effect.

To manually recover the security file

1. Enter the command:

   ```
   STOP TSS
   ```

   CA Top Secret stops.

2. The current backup copy of the security file must be on DASD. Run the TSSRESTN JCL procedure found in PROCLIB

   The backup copy of the security file is restored to DASD.

3. Enter the command:

   ```
   START TSS
   ```

   CA Top Secret starts.

4. Enter the command:

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

   Recovery is turned off.

5. Enter the command:

   ```
   START TSSRCVR1
   ```

   TSSRCVR1 reads the security file timestamp and retrieves changes from the recovery file after that time to avoid problems of double-updating. If it is necessary to override the timestamp enter the TIME(hhmm) and DATE(yyddd) in the EXEC PARM field before executing TSSRCVR1.

6. Enter the command:

   ```
   START TSSRCVR2
   ```

   The changes are applied and the security file is recovered.

7. Enter the command:

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

   The recovery file is turned back on.

Chapter 6: Using the TSSXTEND Utility with WHOHAS

This section contains the following topics:

About the TSSXTEND Utility with WHOHAS (see page 66)
Required Authority for TSSXTEND Utility with WHOHAS (see page 67)
Run the TSSXTEND Utility with WHOHAS (see page 67)

About the TSSXTEND Utility with WHOHAS

When you run the TSSFAR utility with the RESOURCE option, the possibility exists that the utility may identify multiple reference pointers that may not be current.The TSSXTEND utility with the WHOHAS option lets you resolve these pointers in an automated fashion.

This section details the requirements to run the TSSXTEND utility with the WHOHAS option and its behaviors in response to the output of the TSSFAR utility using the RESOURCE option.

The TSSXTEND utility WHOHAS option flags the following:

- Permits for resources that are *not* owned.
- Target ACIDs (owning ACIDs) that will not hold the added WHOHAS information.

Important!: The job runs to completion even if it flags issues as previously detailed; however, before using the extended security file, you must continue to run the job until it does not flag any ownership concerns. This means you may need to run TSSXTEND WHOHAS multiple times. TSSXTEND will fail if it detects that the same output file used in a previous TSSXTEND WHOHAS run is specified without reinitializing the output file between the two runs. If you need to run TSSXTEND utility with the WHOHAS option more than once, *you must delete and reallocate the output SECFILE each time*. Until you correct the ownership errors, do not implement the TSSXTEND WHOHAS output security file.

Before you run TSSXTEND WHOHAS, note the following ownership concerns:

- Run TSSXTEND with WHOHAS against the backup file. We recommend that you execute a TSS BACKUP prior to running the TSSXTEND WHOHAS utility. This ensures that the newly extended security file will be current.
- Run TSSXTEND with WHOHAS using the same system where the security backup file is active. A backup file for a shared security file can be used as input on the TSSXTEND WHOHAS option on any of the sharing systems. You can only use a CPF-synchronized file as input on TSSXTEND WHOHAS under the system where the file was active.
- TSSXTEND with WHOHAS takes significantly longer to execute than running a normal TSSXTEND without the WHOHAS option because the this option causes CA Top Secret to rebuild the WHOHAS data for the entire security file. Expect the added runtime to be ten times as long. For example, a TSSXTEND without WHOHAS that typically runs for 20 minutes will likely take approximately 3 hours when run with WHOHAS.

Required Authority for TSSXTEND Utility with WHOHAS

The following users may run TSSXTEND utility with WHOHAS:

- The MSCA
- A user with USE access to the entity TSSUTILITY.TSSXTEND in the CASECAUT resource class.

  This access may be granted by an administrator using the following command:

  ```
  TSS PERMIT(user) CASECAUT(TSSUTILITY.TSSXTEND) ACCESS(USE)
  ```

Run the TSSXTEND Utility with WHOHAS

Use this procedure to resolve reference pointers using the TSSXTEND utility with the WHOHAS option.

To obtain a valid SECFILE using TSSFAR and TSSXTEND with WHOHAS

1. Run the TSSFAR utility using the RESOURCE option.

   The utility flags any ownership issues.

2. Run TSSXTEND WHOHAS to clean up the ownership problems flagged with each run, which includes:

   - Defining ownerships.
   - When an ACID is not large enough to handle the added WHOHAS information, you need to redefine ownership at multiple levels.

   The utility resolves the out-of-date reference pointers.

3. Re-run the TSSFAR utility using the RESOURCE option.

   After you do not have any un-owned resources permitted or target ACIDs (owning ACIDs) that exceed the maximum size, activate and leverage the extended security file using your standard, established site procedures.

Chapter 7: TSSXVSDT - Digital Certificate Backout

This section contains the following topics:

About TSSXVSDT (see page 69)
VSAM Digital Certificate Backout (see page 69)

About TSSXVSDT

TSSXVSDT is a batch utility that assists in backing out of the VSAM digital certificate feature.

If the VSAM digital certificate feature is active, digital certificates and keyrings are loaded in the VSAM file. To determine if the feature is active, see the appendix 'TSSXTEND Extend the Security File'. If you entered INITVSAM=DIGICERT on the control statement as input to the TSSMAINS program and completed the steps in this section your certificates and keyrings have been migrated to VSAM.

Important! If you maintain multiple security files through CPF, prevent CPF from sending the backout commands to multiple nodes. Work with each system and security file as a single entity.

VSAM Digital Certificate Backout

This procedure is done on a system where the VSAM digital certificate feature is active and digital certificates and keyrings are loaded in the VSAM file.

Follow these steps:

1. Enter the command:

   ```
   TSS LIST(ACIDS) DIGICERT(ALL)
   ```

   A list of all digital certificates added to all users is displayed.

2. Count and record the number of certificates.

3. For each user displayed in the previous list, enter the command:

   ```
   TSS LIST(user) DIGICERT(ALL)
   ```

   Detailed information for all certificates belonging to the user is displayed.

4. Enter the command:

   ```
   TSS LIST(ACIDS) KEYRING(ALL)
   ```

   A list of all keyrings added to all users is displayed.

5. Count and record the number of keyrings.

6. For each user displayed in the previous list, enter the command:

   ```
   TSS LIST(user) KEYRING(ALL)
   ```

   Detail information for all keyrings belonging to the user is displayed.

7. Use the BACKUP control option to create a backup of both the current BDAM security file and the VSAM certificate file. The backups can be used to restore the files in the event of an emergency.

8. Edit the TSSXVSDT batch utility. Enter:

   - The existing VSAM file containing the digital certificate and keyring records.
   - A new VSAM file that will be defined by IDCAMS and populated by the batch utility.
   - A sequential output file that will contain TSS EXPORT commands.
   - A sequential output file that will contain TSS ADD commands.
   - A SYSIN input statement with the format:

     ```
     DCDSN(xxxxxxx.xxxxxxxx.xxxxxxxxx)
     ```

     The name specified is used as a prefix to create the DCDSN operand on both the TSS EXPORT and TSS ADD commands created by the utility. The prefix can have a maximum length of 26 characters and must conform to standard MVS data set naming conventions.

   - A SYSIN input statements with the format:

     ```
     PKCSPASS(pppppppp)
     ```

     The password specified is used to create the PKCSPASS operand on both the TSS EXPORT and TSS ADD commands created by the utility. The password can have a maximum length of 32 characters.

9. Run the TSSXVSDT batch utility:

   The utility generates:

   - A new VSAM file with all digital certificate records and keyring records removed, leaving only KERBEROS records if they exist.
   - File CMDEXPT containing TSS EXPORT commands for all digital certificate records found in the existing VSAM file.
   - File CMDADD containing TSS ADD commands for all digital certificate records and keyring records found in the existing VSAM file.

- A summary report listing the execution results of the utility and any errors found during processing. A non-zero completion code is accompanied by an error message that should be self-explanatory. Correct the error and rerun the utility as often as necessary.

  The summary report contains number of:

  - VSAM input records.
  - VSAM output records.
  - Digital certificate records deleted. This should match the number of certificates recorded in step 2.
  - Keyring records deleted. This should match the number of keyrings recoded in step 5.

10. Edit TSSXVTMP, enter the CMDEXPT file name created by the batch utility TSSXVSDT. This file holds the TSS EXPORT commands to be executed.

11. Run the batch job TSSXVTMP.

    The batch job executes IKJEFT01 to read the TSS command file as input and execute the TSS EXPORT commands. The existing VSAM file is used as input to generate the DCDSN data sets with the certificate data required by the TSS ADD process. A unique data set is allocated and cataloged for each TSS EXPORT command executed, the data set names have the format:

    `DCDSN(`*`xxxxxxxx.xxxxxxxx.xxxxxxxx.aaaaaaaa.dddddddd`*`)`

    xxxxxxxx

    The prefix specified on the input DCDSN statement.

    aaaaaaaa

    Specifies the ACID that owns the certificate.

    dddddddd

    Specifies the certificate name.

12. Edit batch job TSSXVOFF, enter:

    - The name of the BDAM security file found on the SECFILE DD statement in your current TSS started task procedure.
    - A SYSIN input statement with the format:

      `OFF` vvvvvvvv

      vvvvvvvv

      Set to either VSAMDCRT or VSAMALL.

To disable all VSAM processing specify VSAMALL. To determine if VSAM is needed review the count of VSAM output records in step 9. If the count is 1 VSAM can be disabled. If the number of output records is greater than 1 you have KERBEROS records stored in VSAM that require continued VSAM processing and you should only disable certificate and keyring VSAM processing.

To disable VSAM digital certificate and keyring processing specify VSAMDCRT on the input statement. This allows the continued VSAM handling of KERBEROS records that have been migrated to VSAM.

13. Run batch job TSSXVOFF

    The batch job turns off the appropriate VSAM feature flags located in the BDAM security file to disable VSAM processing.

14. Edit the TSS and TSSB started task procedures to reflect the new processing requirements:

    - If you are disabling VSAM processing completely, remove all VSAM related DD statements from the procedures, including VSAMFILE, VSAMBKUP, VSAMAIX, and VSMPATH.
    - If you are only disabling VSAM certificate and keyring processing, remove the DD statements for VSAMAIX and VSMPATH and modify the DD statement VSAMFILE to point at the new VSAM file generated by TSSXVSDT.
    - If you are sharing the security file make the same modifications to the started task procedures on all systems.

15. Shut down and restart the CA Top Secret address space using the updated procedure. The restart should include the startup parameter:

    ```
    REINIT (S TSS,,,REINIT)
    ```

    If you are sharing the security file shut down and restart the CA Top Secret address space with the updated procedure on all systems as soon as possible to prevent the creation of new certificates and keyrings or the update of existing certificates and keyrings in VSAM that will not be reflected in the backout process.

    When the TSS address space is restarted there will be *no* certificates or keyrings available for processing. Any product or process requiring a digital certificate should be quiesced until the certificates are completely restored.

16. Edit TSSXVTMP. Enter the CMDADD file name created by TSSXVSDT, this file holds the TSS ADD commands to be executed.

17. Run TSSXVTMP.

    This job executes IKJEFT01 to read the TSS command file as input and execute the TSS ADD commands. The commands use the DCDSN data sets created by the TSS EXPORT commands as input to add the digital certificates to the appropriate users, add digital certificates to keyrings, and add keyrings to users where required.

18. Review the list output from the execution of the commands and make sure they completed successfully.

19. Shut down and restart the CA Top Secret address space using the procedure from step 15. The restart should include the startup parameter:

    ```
    REINIT (S TSS,,,REINIT)
    ```

20. Repeat the TSS LIST commands in steps 1 to 6. The TSS LIST(ACIDS) is entered as TSS LIST(SDT) commands since the data is no longer in VSAM.

    The commands provide a new directory of digital certificate and keyring objects after they have been restored to the BDAM security file.

21. Compare the TSS LIST command output to verify that all certificates and keyrings have been correctly restored to the BDAM security file. The digital certificate and keyring counts from both steps should match.

22. (Optional) Discard the command files and the DCDSN files generated to support the backout process.

Note: For information on the TSS EXPORT and TSS ADD commands for digital certificates and keyrings, see the *Command Functions Guide* and the *Cookbook*.

Chapter 8: PDS Member Level Protection Utilities

This section contains the following topics:

About PDS Member Level Protection (see page 75)
Security Ownership Commands (see page 76)

About PDS Member Level Protection

Three utility programs are provided to assist the resolution of problems with PDS member level protection. These programs are executed under the direction of CA Technical Support. To prevent their misuse, secure these programs using "CAPDSSEC" resources within the IBMFAC resource class.

Security Ownership Commands

The following example commands define security ownership over these resources and permit their use by "user1" alone:

```
TSS ADDTO(anydept) IBMFAC(CAPDSSEC)
```

```
TSS PERMIT(user1) IBMFAC(CAPDSSEC.STATUS)
```

```
TSS PERMIT(user1) IBMFAC(CAPDSSEC.TRACE)
```

```
TSS PERMIT(user1) IBMFAC(CAPDSSEC.TERM)
```

CAS4STAT

Displays overall status of the PDS member level protection component, location of key modules, and other statistics. This program can only be executed using an ACID which has access to the CAPDSSEC.STATUS resource within the IBMFAC resource class.

Sample execution JCL:

```
//PDS      EXEC PGM=CAS4STAT,REGION=512K
```

CAS4TRCE

Enables PDS member level protection tracing. Trace messages are issued to the system console. This program can only be executed using an ACID which has access to the CAPDSSEC.TRACE resource within the IBMFAC resource class. For information on the output messages from this trace, see the *Messages and Codes Guide*.

Sample execution JCL:

```
//PDS      EXEC PGM=CAS4TRCE,REGION=512K,PARM='opt'
```

Where opt can be:

ON

Enables general tracing.

SEC

Enables tracing of security processing.

CCW

Enables general tracing and also traces CCW I/O activity against a protected data set. This setting can potentially generate a lot of output and, depending on how many data sets are protected, impact system performance.

MEM

Enables general tracing and also generates tracing showing Using the TSSFAR UtilityUsing the TSSFAR UtilityPDS member information including disk address translation. This setting can potentially generate a lot of output.

ALL

Enables all tracing.

OFF

Disables all tracing.

CAS4TERM

Removes PDS member level protection intercepts from the system. In all normal situations, TSS MODIFY PDSPROT(ON/OFF) commands should be used to enable or disable PDS member level protection. This program can only be executed using an ACID which has access to the CAPDSSEC.TERM resource within the IBMFAC resource class.

Sample execution JCL;

```
//PDS      EXEC PGM=CAS4TERM,REGION=512K
```

To re-enable PDS member level protection, enter:

```
TSS MODIFY PDSPROT(ON)
```

Appendix A: CA Top Secret Diagnostic Trace

This section contains the following topics:

Introduction (see page 79)
Trace Destinations (see page 80)
Trace Messages (see page 80)
Trace Detail 1 (see page 81)
Trace Detail 2 (see page 90)
Trace Detail 3 (see page 94)
Trace Detail 4 (see page 99)
Trace Detail 5 (see page 100)
Trace Detail 6 (see page 102)
Diagnostic Traces Examples (see page 103)

Introduction

The trace is used to diagnose access problems. It is activated via any one of several control options specified through the O/S 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

Trace Destinations

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

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

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,keep tracing to a minimum and be specific about who or what is being traced.

Trace Messages

Trace messages begin with:

```
TSS-?
```

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

- 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 ....
- TSS-5 optional trace data if the event is a RACROUTE EXTRACT

Trace 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
- 14-signon failed
- 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.

*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):

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

rr

RACF SVC Flags (bit mapped):

RACINIT:

- 00ENVIR=CREATE
- 04STAT=NO
- 08PASSCHK=NO
- 40ENVIR=CHANGE
- 80ENVIR=DELETE
- C0ENVIR=VERIFY

RACHECK:

- 00RACFIND not specified

01ENTITY=(,CSA)

- 02LOG=NONE
- 0831-bit parameters
- 10VSAM dataset
- 80RACFIND=NO
- C0RACFIND=YES

RACDEF:

- 00RACFIND not specified
- 20CHKAUTH=YES
- 80RACFIND=NO
- C0RACFIND=YES

G/

Algorithm Data

sw

Algorithm Switch:

- 00access allowed
- 04authorization not found
- 08access denied
- 0Cvolume access is create; force DSN checking
- 10volume 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 ADDTO)
- 20 = authorized (via TSS PERMIT)
- 80 = tape owned

aa

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

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

L/

LOGGING INDICATORS

l1

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

l2

FLAGS (bit mask):

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

ee

EVENT CODE:

- 01job initiation
- 02resource check
- 32TSS command
- 33program change
- 34change control option
- 39DUF update
- 40operator 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 = z/OS 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-compatible
- 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:

- n01 = default ACID
- n02 = 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

Trace 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

Trace 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

Reserved

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:

- n02 = SYSLIB
- n01 = 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

fa01 = XDEF

- 02 = MULTIUSER
- 04 = NOABEND
- 10 = ASUBM
- 20 = NSHRPRF
- 40 = INACTIVE
- 80 = facility in use

fb01 = NOINITAUTH

- 02 = RNDPW
- 04 = NOINSTDATA
- 08 = unused
- 10 = PSEUDO
- 20 = SIGNS
- 40 = STMSG
- 80 = LUMSG

fc01 = transactions being used

- 02 = TSOC
- 04 = WARNPW
- 08 = unused
- 10 = NORES
- 20 = AUDIT
- 80 = TSOPWP

fd01 = VM SIO checking in effect

- 02 = IMS extended support
- 04 = new password reverification
- 08 = honor password in dormant mode
- 40 = TRACE
- 80 = ALWAYSCALL

Trace 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

Trace Detail 5

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

Audit Indicator Information

a1

First Audit Indicator

- 01=action of audit
- 02=user is audited

Can be result of:

  - facility has AUDIT attribute FACILITY(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
- 2=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

OC

The original resource class if a translation was performed

Trace Detail 6

```
TSS-5  fieldname/pf  fieldname/pf  fieldname/pf  fieldname/pf  fieldname/pf
fieldname/pf
```

Fieldname

Name of field to be extracted

pf

Relative security record from which the field was extracted:

- 00 = User record
- 01-FE = Profile 1-254
- FF = Default value supplied
- NF = Field not found

Diagnostic Traces Examples

Example: 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.

Example: 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 indicate password violation.

Trace (11) shows password required but not validated.

Example: 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.

Example: 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).

Example: 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).

Example: 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.

Example: 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.

Example: 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).

Example: 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 B: Tracing SAF Requests

This section contains the following topics:

About the SAF SECTRACE Command (see page 109)
SAF SECTRACE Command (see page 110)
SAF SECTRACE SET and MODIFY Operands (see page 111)
Stop and Restart SECTRACE (see page 121)
Enable SAF SECTRACE Traps (see page 121)
Disable SAF SECTRACE Traps (see page 121)
Delete SECTRACE Traps (see page 122)
Display SAF SECTRACE Operands (see page 122)
SAF SECTRACE at SAF Initialization (see page 122)
Tracing UNIX System Services (OMVS) (see page 125)

About the SAF SECTRACE Command

Use the SAF SECTRACE command to trace any security request made to the System Authorization Facility (SAF). Any program using SAF automatically interfaces with CA Top Secret.

SAF SECTRACE is used to supply information to technical support.

SAF SECTRACE enables you to:

- Capture, format, and display the RACROUTE parameter list passed by requests for SAF services
- Display additional environmental information, such as job name, user ID, and the program issuing the SAF call.

The batch utility program, TSSRPTST, processes and displays the output written to SMF by the SAF SECTRACE command. For information, see the *Report and Tracking Guide*.

Because SAF SECTRACE can supply sensitive information, limit the use of the SAF SECTRACE command.

SAF SECTRACE Command

This command has the format:

```
SECTRACE [SET operand][MODIFY operand][ENABLE|DISABLE][DELETE][DISPLAY]
id=trapid|ALL
```

SET|T operand

Defines a SAF SECTRACE trap. By default defines the trap as enabled.

MODIFY|F operand

Changes one or more operands of a named SAF SECTRACE trap. A SAF SECTRACE SET command must have previously defined the trap.

ENABLE

Activates the named SAF SECTRACE trap. A SAF SECTRACE SET command must have previously defined the trap.

DISABLE

Deactivates the named SAF SECTRACE trap. The trap definition is retained and can be reactivated by a subsequent SAF SECTRACE ENABLE command.

DELETE

Removes the named SAF SECTRACE trap definition.

D|DISPLAY

Displays the status and operands of the named SAF SECTRACE trap.

ID=trapid|ALL

Names the SAF SECTRACE trap. Define multiple trace requests by specifying a unique ID for each. Required with all SAF SECTRACE commands except the SAF SECTRACE SET command. If you do not supply an ID value with the SET command, the SAF SECTRACE command assigns a value. The format of the default trace ID is TRACE (*nnn*),where (nnn), is the next available sequential number.

ALL

Specifies that all IDs are to be traced.

Note: You must enter the ID on the MODIFY command before any other operand.

SECTRACE Command Use

The protection table of the SECTRACE command is shown below:

Command	OPERCMDS Resource Name	Access
SECTRACE SET	TRCE.SECTRACE.SET	UPDATE
SECTRACE DELETE	TRCE.SECTRACE.DELETE	UPDATE
SECTRACE MODIFY	TRCE.SECTRACE.MODIFIED	UPDATE
SECTRACE DISPLAY	TRCE.SECTRACE.DISPLAY	READ
SECTRACE ENABLE	TRCE.SECTRACE.ENABLE	UPDATE
SECTRACE DISABLE	TRCE.SECTRACE.DISABLE	UPDATE

The ST command starts the SECTRACE address space and this address space remains up (but not active) until the next IPL. The only time the address space does anything is when the SAF SECTRACE is active. No attempt should be made to cancel the SECTRACE address space as unpredictable result might occur.

You can enter the SECTRACE command at an operator's console, at a terminal that acts as a console, or through SYS1.PARMLIB. When the first SAF SECTRACE command is entered, CA Top Secret automatically starts the SAF SECTRACE address space.

When you enter commands at a console, you can enter multiple keywords and operands by pressing ENTER after each line. SAF SECTRACE prompts you at the operator's console issuing the command for more operands until you type:

```
END             or
CANCEL.
```

This example shows the WTOR message asking for continuance of the SAF SECTRACE operands:

```
nn CAS21100 CONTINUE SECTRACE SPECIFICATION, CANCEL, OR END

R nn,JOBNAME=jobname,END
```

SAF SECTRACE SET and MODIFY Operands

You must identify the trap through the ID operand when you issue the SAF SECTRACE MODIFY command. You can, however, invoke the SAF SECTRACE SET command without the ID operand. SAF SECTRACE assigns the value TRACE(nnn), where (nnn) is the next available sequential number.

TYPE Operand—Identify the Type of Event to Process

Use the TYPE operand to identify the type of security event to be processed. TYPE lets you specify RACROUTE parameters that act as additional filters on specific SAF requests.

This operand has the following format:

```
TYPE=SAF|SAFP|OMVS|HFS
```

SAF

Specifies that the security event to be traced is from the z/OS SAF facility. Any SAF event that matches the specified environment qualifies for tracing. This value is the default value.

SAFP

Specifies that the security event to be traced is from the z/OS SAF facility. Use this option to enter additional SAF information to further filter out SAF events. When you enter TYPE=SAFP, you are prompted by SAF SECTRACE for the RACROUTE parameters. Only single values for parameters are accepted. The maximum length for the parameter keyword, operator, and value is 64 characters. To specify parameters that are unique to a specific RACROUTE request, first enter REQUEST= to specify the type of RACROUTE request. SECTRACE continues to prompt you until you enter END or CANCEL.

You must adhere to RACROUTE macro coding conventions when you specify RACROUTE parameters for SAF SECTRACE filtering. You can specify special operators to indicate the presence of a particular value (for example, ENVIR=CREATE) or the presence of a pointer address (for example, ACEE=>). You can use the following operators, depending on your type of keyboard:

- = (equal to)
- ^= (not equal to)
- <> (not equal to)
- != (not equal to)
- => (pointer address specified)
- ^=> (no pointer address specified)
- !=> (no pointer address specified)

Pointer operators are valid only if the parameter is specified as a pointer to a data area or a data structure (for example, an ACEE). When you specify a pointer operator, you cannot specify a value for the parameter. A pointer operator indicates the presence or absence of a pointer. For example, you cannot specify ACEE=>address, but you can specify ACEE=> or ACTINFO^=>. Using standard CA Top Secret masking characters, you can mask character data types of parameters. You can mask other types of data only if the mask is complete. A complete mask indicates that the trap will match all values. For example, USERID=- indicates that this parameter matches all values of USERID. USERID^= indicates that the USERID option is not coded on the RACROUTE request.

OMVS

Specifies that the security event to be traced is from the USS facility. Any SAF event that matches the specified function qualifies for tracing.

HFS

Specifies to trace internal functions of CA SAF HFS security. The trace output might be requested by CA Support.

OMVS Operands—Define Callable Services

Use the OMVS operands group to define callable services. You cannot specify more than one function, however, multiple SECTRACES can be set. Other parameters can be specified on the command, but they are ignored. The OMVS SECTRACE output is written to the console only. DEST does not redirect the output.

This operand has the format:

```
FUNC=ALL|MAKE|INIT|SET|GET|CHECK|CHANGE|MISC
```

ALL

(Default) Defines all callable services.

MAKE

make_fsp, make_root_fsp, make_ISP

INIT

initUSP, deleteSUP, fork_exit, init_ACEE

SET

set_file_creation_mask, setuid, set_effective_uid, setgid, set_effective_gid, R_exec_set, clear_setid, R_admin

GET

getUMAP, getGMAP, get_supplemental_groups, get_users_groups, get_effective_uid_gid_supplemental_groups, R_dceinfo, R_dcekey, R_dceruid

CHECK

check_access, check_privilege, check_process_owner, check_file_owner, check_owner_2_files, R_dceauth

CHANGE

change_owner_group, change_file_mode, change_audits_options

MISC

audit, query_file_options, query_security_options

TRACING Operands—Event Matching

Use the TRACING operands group to specify which address spaces, jobs, or users must be in control for an event to match the SAF SECTRACE specifications. By specifying USERID= - or JOBNAME= -, you can trap events regardless of the address space they occur in. However, you can create more specific traps to filter out individual users or started tasks by ASID or job name. These traps enable you to use SAF SECTRACE in a production environment with little impact. When you specify one or more operands from this group, a match occurs only when all of the operands are matched (the operands are ANDed).

These operands have the format:

```
[ASID=nn][JOBname=mask][USERid=mask]
```

ASID=nn

Specifies the address space number.

JOBname=mask

Indicates the job name or job name mask of the target address space as determined by the ACUCB (address space level ACEE). Use this operand to target batch, TSO, started task, and MOUNT address spaces, even if they are not secured by CA Top Secret.

Use an asterisk (*) as a masking character to display all possible combinations of a specific jobname. For example, specify ABC* to display all jobnames beginning with ABC.

USERid=mask

Specifies the logonid or logonid mask of the target address space as determined by the ACUCB (address space level ACEE).

Use an asterisk (*) as a masking character to display all possible combinations of a specific userid. For example, specify ABC* to display all userids beginning with ABC.

ENVIRONS Operands—Environment Filter

Use the ENVIRONS operations group as filters on the environment when a security event occurs. If one or more fields from this group are specified, all fields must match for SAF SECTRACE to trap the event.

These operands have the format:

```
[RB=mask][ProGraM=mask][RETcode=nn][RSNcode=nn]
```

RB=mask

Specifies the request block (RB) name that the security event must occur in. When an event occurs directly under a PRB, the name of the program specified in that block is used to match what you specify in this field. If an event occurs under a supervisor call request block (SVRB), the RB name is assigned SVCnnn, where nnn is the decimal SVC number. If this RB is the only RB on the active RB chain under an SVRB, the interrupt code (SVC number) cannot be determined. Therefore, another RB name is assigned. If the program manager indicator is set, the assigned RB name is *PMSVRB*. If this indicator is not set, the RB name is *SYSTEM*. If the security event occurs under the control of a service request block (SRB), the assigned RB name is *SRB*.

ProGraM=mask

Specifies the program name of the newest PRB on the active RB chain. If no PRB exists on the active RB chain when a monitored event occurs, the name used for the RB field is also used for PROGRAM.

RETcode=nn

Specifies the return code to be matched against the return code from the security event. If a return code of 0 (the default) is specified, all return codes from security events match. If a nonzero return code is specified and the return codes match (for TRACE=AFTER requests only), the specified SAF SECTRACE action takes place.

RSNcode=nn

Indicates the reason code to be matched against the reason code from the security event. If a reason code of 0 (the default) is specified, all reason codes from security event match. If a nonzero reason code is specified and the reason codes match (for TRACE=AFTER requests only), the specified SAF SECTRACE action takes place.

STATUS Operands—Define Trap Status

Use the STATUS group operands to define:

- The status of the SAF SECTRACE trap
- Which action to take if an event is trapped and when to apply the trap

If an event matches the criteria for more than one defined SAF SECTRACE trap, SAF SECTRACE generates trace output in the order that the SAF SECTRACE traps were entered until a trap specifying ACTION=IGNORE is encountered.

These operands have the format:

```
[ENABLE|DISABLE][ACTION=IGNORE|TRACE][,TRACE=[PRE|BEFORE]][POST|AFTER]][ALL]
```

ENABLE|DISABLE

Specifies the initial status of the SECTRACE trap.

ENABLE

(Default) Indicates that the SECTRACE trap identified by the ID operand is enabled. All events that match the SECTRACE trap are trapped.

DISABLE

Indicates that the SECTRACE trap identified by the ID operand is disabled.

ACTION=IGNORE|TRACE

Specifies the action to perform when a SECTRACE trap is matched.

IGNORE

Indicates that the SECTRACE trap is ignored and the search for other defined SECTRACE traps that might match the event is halted.

TRACE

Specifies that the trace output is sent to all specified destinations. See the DEST operand for the destinations that you can specify.

TRACE=PRE|BEFORE|POST|AFTER|ALL

Specifies when the trap is applied if all criteria are matched before security validation or processing occurs or after it occurs.

PRE|BEFORE

Specifies that the event is trapped before security validation or processing occurs.

POST|AFTER

Specifies that the event is trapped after security validation or processing occurs.

ALL

Indicates that the event is trapped before and after all security validation or processing occurs.

EVENTS Operand—Maximum Number of Events

Use the EVENTS operand to specify the maximum number of events to be traced before the trap is automatically disabled.

MATCHLIM=1000|0|*nn*

Specifies the maximum number of trace events to occur before SAF SECTRACE is automatically disabled.

To reduce overhead, terminate SAFTRACE after a finite number of events.For an unlimited trace, set the match limit to zero.

When a disabled SAFTRACE is modified to ENABLE, the event counter for the trap is reset to zero and the MATCHLIM remains unchanged (unless specifically modified by the command).

Maximum: 9999

ROUTE TO Operands—Output Format and Routing

Use the ROUTE TO operands group to describe how the trace output is formatted and routed to a console.

These operands have the format:

```
[CONSid=nn][TSoUser=id][LINELEN=nn]
                       [DEST=CONSOLE|JOBLOG|SMF|SYSLOG|TSOUSER|ALL|DATASET]
                       [ForMaT[DUMP|LABEL|NOPACK|PACK]
                       [IMSGid|NOMSGid]
                       [WAIT|NOWAIT]
```

CONSid=nn

Identifies the console that receives the trace output when DEST=CONSOLE is specified. This operand avoids problems caused by flooding crucial consoles in a production environment and preserves the existing SAF SECTRACE function.

TSoUser=id

Identifies the ID of the time-sharing (TSO) user who receives the trace output when DEST=TSOUSER is specified.

LINELEN=nn

Specifies the length of the variable output line for the display of the RACROUTE macro parameters.

DEST=CONSOLE|JOBLOG|SMF|SYSLOG|TSOUSER|ALL|DATASET

Specifies where the trace output is delivered when ACTION=TRACE is specified:

CONSOLE

Indicates that trace output is sent to the console identified by the CONSID operand. SAF SECTRACE may discard trace output when control is received with LOCKs held or in SRB mode. When control is received again, a message is issued stating how many trace events were lost.

JOBLOG

Specifies that trace output is sent to the JOBLOG (ROUTCDE=11). SAF SECTRACE may discard trace output when control is received with LOCKs held or in SRB mode. When control is received again, a message is issued stating how many trace events were lost.

SMF

Indicates that trace output is journaled to the System Management Facility (SMF). SMF is the only trace output destination where output is guaranteed. Information is retrieved and formatted with TSSRPTST.

SYSLOG

Specifies that trace output is sent to the system log file (WTL). SAF SECTRACE may discard trace output when control is received with LOCKs held or in SRB mode. When control is received again, a message is issued stating how many trace events were lost.

TSOUSER

Specifies that trace output is sent to the time-sharing (TSO) user identified by the TSOUSER operand. See the WAIT|NOWAIT operand for additional information. SAF SECTRACE may discard trace output when control is received with LOCKs held or in SRB mode. When control is received again, a message is issued stating how many trace events were lost.

ALL

Indicates that all of the above destinations will receive trace output.

DATASET

Specifies that trace output is sent to a data set. The data set can be a sequential data set or a member of a partitioned data set(PDS). SECTRACE might discard trace output when control is received with locks held in SRB mode. DATASET is not included in DEST=ALL. A separate SECTRACE is required to specify DATASET as a destination.

Note: DEST=DATASET also requires the DSName= operand to specify the data set that is to receive the trace output. If a member of a partitioned data set is to be used, the MEMBER= operand is required in addition to the DSName= operand.

This console may also receive messages regarding the failure to deliver trace messages to any of the other destinations. The default for CONSID is the console that initially set the SAF SECTRACE trap.

DSName=dsn

Identifies the DASD data set that receives the trace output. The data set must be allocated and available before the SECTRACE is enabled.

This must be a fully-qualified data set name. Do not include the member name in the DSName= operand. Use the MEMBER= operand to specify the member name.

Specifying this operand automatically sets DEST=DATASET provided that the DEST= operand does not follow this operand when entering the SECTRACE command.

You can use the INITSECD member included in the sample JCL library to allocate the data set.

ForMaT=DUMP|LABEL|NOPACK|PACK

Specifies the format option of the trace output. Format options are valid only for output destinations of CONSOLE, JOBLOG, SYSLOG, and TSOUSER:

DUMP

Specifies that the external data structures identified in the RACROUTE macro definition are to be displayed following the RACROUTE macro parameters. These external data structures are shown in both hexadecimal and EBCDIC formats.

Note: DUMP is only valid when used with the PACK operand.

LABEL

Indicates that the RACROUTE parameter format extension tags are displayed. This field is for internal use and diagnostic purposes only.

Note: LABEL is only valid when used with the PACK operand.

NOPACK

Specifies that the output of the RACROUTE macro is not to be packed. Only one parameter per line is displayed.

PACK

Indicates that the output of the RACROUTE macro is to be packed with multiple parameters per line up to the line length specified by the LINELEN operand of the SAF SECTRACE command.

MEMBER=member

Identifies the member of a partitioned data set that receives the trace output.

MSGid|NOMSGid

Indicates whether the message ID of the trace output is displayed on the output. This option is valid only for CONSOLE, JOBLOG, SYSLOG, and TSOUSER destinations.

Default: MSGID.

WAIT|NOWAIT

Specifies whether the task control block (TCB) being traced is to wait if trace output destined for a TSO user cannot be displayed until TPUT buffers are available for the TSOUSER destination. When NOWAIT is specified, trace messages may be lost. When TPUT buffers are available, a message tells the TSO user of the loss of trace messages. Use WAIT with caution because you may halt system-level processing by waiting for the availability of TPUT buffers. To view messages and free buffers for use, the TSO user must press the Enter key.

Default: NOWAIT

END and CANCEL Operands—Complete Trap Specification

Use one of these operands to complete your SAF SECTRACE specifications:.

END

Indicates that the SAF SECTRACE specifications for this trap are finished.

CANCEL

Specifies that the SAF SECTRACE specifications for this trap are ignored.

Stop and Restart SECTRACE

To disable the SECTRACE for OMVS, enter:

```
ST DISABLE,ID=tracennn,END
```

To start a disabled trace, enter:

```
ST ENABLE,ID=tracenn,END
```

To stop the SECTRACE for OMVS, enter:

```
ST DEL,ID=tracennn,END
```

tracennn

The identifier assigned to the SECTRACE.

Enable SAF SECTRACE Traps

To activate a SAF SECTRACE trap that has been previously defined by the SAF SECTRACE SET command, enter:

```
SECTRACE ENABLE id=trapid|ALL
```

Disable SAF SECTRACE Traps

To deactivate a SAF SECTRACE trap, enter:

```
SECTRACE DISABLE id=trapid|ALL
```

The trap definition is retained and can be reactivated using a subsequent SAF SECTRACE ENABLE command.

Delete SECTRACE Traps

To delete a SECTRACE trap that has been previously defined, enter:

```
SECTRACE DELETE id=trapid|ALL
```

Display SAF SECTRACE Operands

With the SAF SECTRACE DISPLAY command, you can display the operands of a specified trap and the number of events that matched the specified trap.

The format of the DISPLAY command is:

```
SECTRACE D|DISPLAY id=trapid|ALL
```

SAF SECTRACE at SAF Initialization

SAF SECTRACE may be set during the SAF initialization process when problems are encountered by creating the member, CAISEC00, in SYS1.PARMLIB. This member may contain one of the one-line entries below:

```
PROMPT
  or
TRACE(zz,{NOSTART|START})
```

The first entry indicates that the message CAS2070I is generated at the system console.

The second entry indicates that ST SAF SECTRACE commands are read from member CAITRCzz; the operand, NOSTART or START, is used to indicate if SAF initialization should suppress or start the commands in the member.

Example: Installation Defaults

The *Installation Guide* provides the following default member CAISEC00 and CAITRC00.

```
CAISEC00:
        TRACE(00,NOSTART)
```

This member associates SAF initialization with ST command member CAITRC00, but does not start the commands in that member. The commands provided in CAITRC00 are:

```
CAITRC00:
        ST SET,ID=SAFINIT,DEST=SMF,END
```

Note: The SAF SECTRACE at SAF initialization should be directed to SMF, because it is probable that JES SYSTEM LOFG is not available at the time the trace begins.

Example: Prompted Implementation

When directed by support, implement a prompted SAF SECTRACE for OMVS with members:

```
CAISEC00:
        PROMPT
```

This member automatically generates the message, CAS2070I, prompting the console operator to choose at IPL whether to initiate a SAF SECTRACE specification or to default.

- If the operator responds with:

  ```
  R 01,6
  ```

  the default command goes into effect

  ```
  TRACE(00,NOSTART)
  ```

  so that the member, CAITRC00 (if it exists), is referenced but not processed by SAF SECTRACE.

- If the operator responds with:

  ```
  R 01,TRACE(55,START)
  ```

  the TRCE command is entered directly from the console. The member CAITRC55 is referenced for its ST command, and the trace is started immediately.

- If the operator responds with:

  ```
  R 01,SEC=01
  ```

  the member CAISEC01 is invoked:

  ```
  CAISEC01
  TRACE(99,START)
  ```

  This member points you to the member CAITRC99:

  ```
  ST SET,TYPE=OMVS,ID=OMVS,DEST=SMF,END
  ```

Tracing UNIX System Services (OMVS)

Setting a SECTRACE for USS directs the trace to the SMF file. There is no choice (as there is with TYPE=SAF or TYPE=SAFP) of:

- ID
- DEST
- FORMAT
- ACTION
- TRACE
- ENABLE|DISBALE

Note this ID, it is needed later to disable the SECTRACE.

The format to set an OMVS trace is:

```
SECTRACE SET,FUNC=function,END
```

For a list of valid functions, see OMVS Group Operands. The ID for the TRACE request is provided on the MVS console using the non-rollable message

```
CAS2110I SECTRACE SET ON mm.ddd hh:mm:ss ID=TRACE001
```

The SECTRACE remains active until it is disabled. In general, setting or enabling a SECTRACE should be performed only under the direction of CA Top Secret Support, as this can add considerable overhead to USS/OMVS operations.

Stop SECTRACE for OMVS

To stop he SECTRACE for OMVS, enter:

```
ST DEL,ID=XXXX,END
```

xxxx

The identifier assigned to the SECTRACE.

Index

(

(TSO) Invoke a Simulation Session to Test Security File Permissions • 4, 41

A

Abends Occurring • 10
About PDS Member Level Protection • 75
About the SAF SECTRACE Command • 109
About the TSSFAR Utility • 51
About the TSSXTEND Utility with WHOHAS • 66
About TSSRECVR • 59
About TSSXVSDT • 69
ACID Index Performance Setting • 28
Activate the Cache • 31
Authority and Scope • 36

C

CA Technologies Product References • 3
CA Top Secret Address Space • 27
CA Top Secret CACHE • 28
CA Top Secret Diagnostic Trace • 79
CA Top Secret Not Logging Access or Initiations • 15
CA Top Secret Not Logging Violations • 15
CACHE and CF Size • 29
CICS / DB2 Considerations • 34
CICS Performance Considerations • 33
CICS Signon Sub-Tasking • 33
Common Troubleshooting Procedures • 9
Console not Receiving CA Top Secret Messages • 16
Contact CA Technologies • 3
CPF Statistics • 26
CPU Restrictions not Honored • 13
Customization Tips and Checkpoints • 20

D

Data Integrity • 27
Data Set or Volume Access Incorrectly Denied • 17
Data Set Passwords not Checked • 18
Delete SECTRACE Traps • 122
Diagnostic Traces Examples • 103
Disable SAF SECTRACE Traps • 121
Display SAF SECTRACE Operands • 122
Documentation Changes • 4

E

EJECT Command—Page Eject • 48
Enable SAF SECTRACE Traps • 121
END and CANCEL Operands—Complete Trap Specification • 121
END/QUIT Command—Exit Simulator • 44
ENVIRON Command—Change Simulated Environment • 43
ENVIRONS Operands—Environment Filter • 115
EVENTS Operand—Maximum Number of Events • 117
Example
  Installation Defaults • 123
  Prompted Implementation • 124
Execute the Simulator in a Batch Environment • 40

F

Facility Access Incorrectly Allowed • 11
Facility Access Incorrectly Denied • 14
File Sharing and the Coupling Facility • 28

H

HELP Command—List and Describe Commands • 43

I

Installation Exit • 21
Introduction • 79
Introduction to TSSSIM • 35

L

LOGOFF/SIGNOFF • 43

M

Manual Recovery Procedure • 63

O

OMVS Operands—Define Callable Services • 113

P

Passwords not Checked • 12
PDS Member Level Protection Utilities • 75
Perform a Simulated Resource Check • 4, 44

Performance Related Statistics • 24
Performance Tuning Considerations • 23

R

Recovery File Location • 27
Recovery Procedure for Automatic Backup • 60
Required Authority • 51
Required Authority for TSSXTEND Utility with WHOHAS • 67
ROUTE TO Operands—Output Format and Routing • 118
Run the TSSXTEND Utility with WHOHAS • 55, 67

S

SAF SECTRACE at SAF Initialization • 122
SAF SECTRACE Command • 110
SAF SECTRACE SET and MODIFY Operands • 111
SECCACHE in a Shared Security File Environment • 32
SECTRACE Command Use • 111
Security File Cache • 30
Security File Cache Size • 31
Security File Location • 26
Security Ownership Commands • 76
Set Up MEMLIMIT • 52
Shared Profile Table • 34
Sharing Considerations • 27
Statistics Gathering • 25
STATUS Command—Current Environment's Status • 49
STATUS Operands—Define Trap Status • 116
Stop and Restart SECTRACE • 121
Stop SECTRACE for OMVS • 125

T

Trace Destinations • 80
Trace Detail 1 • 81
Trace Detail 2 • 90
Trace Detail 3 • 94
Trace Detail 4 • 99
Trace Detail 5 • 100
Trace Detail 6 • 102
Trace Messages • 80
TRACING Operands—Event Matching • 114
Tracing SAF Requests • 109
Tracing UNIX System Services (OMVS) • 125
Troubleshoot Customization Problems • 19
Troubleshooting • 9
TSS Command—Performing Security Administration Functions • 49
TSSFAR Control Statements • 53
TSSFAR JCL • 53
TSSRECVR - Security File Recovery • 59
TSSSIM and CA Roscoe • 41
TSSSIM in TSO/E-Background, TSO/E-Foreground, or TSO/E-Batch • 39
TSSSIM Special Commands • 48
TSSSIM TSO Online Tutorial • 38
TSSSIM Under TSO in a Non-SPF Environment • 36
TSSXVSDT - Digital Certificate Backout • 69
Tuning CPF • 32
TYPE Operand—Identify the Type of Event to Process • 4, 112

U

Use TSSSIM Under TSO in an SPF Environment • 37
Users not Receiving CA Top Secret Messages • 16
Using the TSSFAR Utility • 51
Using the TSSSIM Utility • 35
Using the TSSXTEND Utility with WHOHAS • 65

V

VSAM Digital Certificate Backout • 69